УДК 004.94

В.А. Резниченко

60 лет базам данных

Приводится обзор исследований и разработок баз данных с момента их возникновения в 60-х годах прошлого столетия и по настоящее время. Выделяются следующие этапы: возникновение и становление, бурное развитие, эпоха реляционных баз данных, расширенные реляционные базы данных, пост-реляционные базы данных и большие данные. На этапе становления описываются системы IDS, IMS, Total и Adabas. На этапе бурного развития освещены вопросы архитектуры ANSI/X3/SPARC, предложений КОДАСИЛ, концепции и языков концептуального моделирования, моделей данных. На этапе эпохи реляционных баз данных раскрываются результаты научной деятельности Э. Кодда, теория зависимостей и нормальных форм, языки запросов, экспериментальные исследования и разработки, оптимизация и стандартизация, управление транзакциями. Этап расширенных реляционных баз данных посвящен описанию темпоральных, пространственных, дедуктивных, активных, объектных, распределенных и статистических баз данных, баз данных массивов, машин баз данных и хранилищ данных. На следующем этапе раскрыта проблематика постреляционных баз данных, а именно, NOSQL, ключ-значение, документные, колоночные, графовые, NewSQL, онтологические, мультимедийные, изображений, видео, полнотекстовые, мультимодельные. На шестом этапе раскрываются причины возникновения, характеристические свойства, классификация, принципы работы, методы и технологии больших данных. Наконец, в последнем разделе дается краткий обзор исследований и разработок по базам данных в Советском Союзе.

Ключевые слова. Типы баз данных: иерархическая, сетевая, реляционная, навигационная, темпоральная, пространственная, пространственно-временная, пространственно-сетевая, перемещающихся объектов, дедуктивная, активная, объектно-ориентированная, объектно-реляционная, распределенная, параллельная, массивов, статистическая, многомерная, машина баз данных, хранилище данных, NoSQL, ключ-значение, колоночная, документо-ориентированная, графовая, мультимодельная, мультимедийная, облачная, научная, многозначная, XML, NewSQL, онтологическая, векторная, потоковая, полнотекстовая, видео, большие данные.

В.А. Резніченко. 60 років базам даних

Наводиться огляд досліджень і розробок баз даних з моменту їх виникнення в 60-х роках минулого століття і по теперішній час. Виділяються наступні етапи: виникнення і становлення, бурхливий розвиток, епоха реляційних баз даних, розширені реляційні бази даних, постреляційні бази даних і великі дані. На етапі становлення описуються системи IDS, IMS, Total і Adabas. На етапі бурхливого розвитку висвітлені питання архітектури баз даних ANSI/X3/SPARC, пропозицій КОДАСИЛ, концепції і мов концептуального моделювання. На етапі епохи реляційних баз даних розкриваються результати наукової діяльності Е. Кодда, теорія залежностей і нормальних форм, мови запитів, експериментальні дослідження і розробки, оптимізація та стандартизація, управління транзакціями. Етап розширених реляційних баз даних присвячений опису темпоральних, просторових, дедуктивних, активних, об'єктних, розподілених та статистичних баз даних, баз даних масивів, машин баз даних і сховищ даних. На наступному етапі розкрита проблематика постреляціонних баз даних, а саме: NOSQL, ключ-значення, стовпчикові, документні, графові, NewSQL, онтологічні, мультимедійні, зображень, відео, повнотекстові, мультимодельні. Шостий етап присвячений розкриттю причин виникнення, характеристичних властивостей, класифікації, принципів роботи, методів і технологій великих даних. Нарешті, в останньому розділі дається короткий огляд досліджень і розробок по базах даних в Радянському Союзі.

Ключові слова. Типи баз даних: ієрархічна, мережева, реляційна, навігаційна, темпоральна, просторова, просторово-темпоральна, просторово-мережева, об'єктів, що переміщуються, дедуктивна, активна, об'єктно-орієнтована, об'єктно-реляційна, розподілена, паралельна, масивів, статистична, багатовимірна, машина баз даних, сховища даних, NoSQL, ключ-значення, стовпчикова, документо-орієнтована, графова, мультімодельна, мультімедійна, хмарна, наукова, багатозначна, XML, NewSQL, онтологічна, векторна, потокова, повнотекстова, відео, великі дані.

V.A. Reznichenko. 60 Years of Databases

The article provides an overview of research and development of databases since their appearance in the 60s of the last century to the present time. The following stages are distinguished: the emergence formation and rapid development, the era of relational databases, extended relational databases, post-relational databases and big data. At the stage of formation, the systems IDS, IMS, Total and Adabas are described. At the stage of rapid development, issues of ANSI/X3/SPARC database architecture, CODASYL proposals, concepts and languages of conceptual modeling are highlighted. At the stage of the era of relational databases, the results of E. Codd's scientific activities, the theory of dependencies and normal forms, query languages, experimental research and development, optimization and standardization, and transaction management are revealed. The extended relational databases phase is devoted to describing temporal, spatial, deductive, active, object, distributed and statistical databases, array databases, and database machines and data warehouses. At the next stage, the problems of post-relational databases are disclosed, namely: NOSQL, key-value, column, document, graph, NewSQL, ontological, multimedia, images, video, full-text, multimodel databases. The sixth stage is devoted to the disclosure of the causes of occurrence, characteristic properties, classification, principles of work, methods and technologies of big data. Finally, the last section provides a brief overview of database research and development in the Soviet Union.

Keywords. Database types: hierarchical, network, relational, navigational, temporal, spatial. spatio-temporal, spatio-network, moving objects, deductive, active, object-oriented, object-relational, distributed, parallel, arrays, statistical, multi-dimensional, database machines, data warehouse, NoSQL, key-value , triple store, column-oriented, document- oriented, graph-oriented, multimodel, multimedia, cloud, scientific, multi-valued, XML, NewSQL, ontological, vector, streaming, video, full-text, Big Data.

Содержание

Список сокращений

Далее приводится список сокращений, которые используются для именования баз данных различного типа.

АБД	—	активная БД
БД	—	база данных
БДВ	—	БД видео
БДИ	—	БД изображений
БДКЗ	—	БД ключ-значение
БДМ	—	БД массивов
БДПО	—	БД перемещающихся объектов
ВБД	—	векторная БД
ГБД	—	графовая БД
ГМБД	—	гипермедийная БД
ДБД	—	дедуктивная БД
ДОБД	—	документо-ориентированная БД
ИБД	—	интенсиональная БД
КБД	—	колоночная БД
МБД	—	машина БД
ММБД	—	мультимодельная БД
ММеБД	—	мультимедийная БД
МнБД	—	многомерная БД
ОБД	—	объектная БД
ОнБД	—	онтологическая БД
ООБД	—	объектно-ориентированная БД
ОРБД	—	объектно-реляционная БД
ПБД	—	пространственная БД
ПВБД	—	пространственно-временная БД
ПВСБД	—	пространственно-временная сетевая БД
Поли-БД	—	поли-БД
ПСБД	—	пространственно-сетевая БД
ПТБД	—	полнотекстовая БД
ПтБД	—	потоковая БД
РаБД	—	распределенная БД
РБД	—	реляционная БД
СБД	—	статистическая БД
ТБД	—	темпоральная БД
ФБД	—	федеративная БД
ЭБД	—	экстенсиональная БД

Введение

Концепция баз данных, зародившаяся шестьдесят лет назад, оказалась весьма плодотворной. Базы данных стали важной составляющей компьютерных наук и информационных технологий. Они проникли во все сферы человеческой деятельности: науку, инженерное дело, экономику, бизнес, образование, медицину, культуру и многие другие отрасли. И даже вершина современной информационной технологии, без которой немыслимо существование человечества - веб, это, по сути, всемирная база данных.

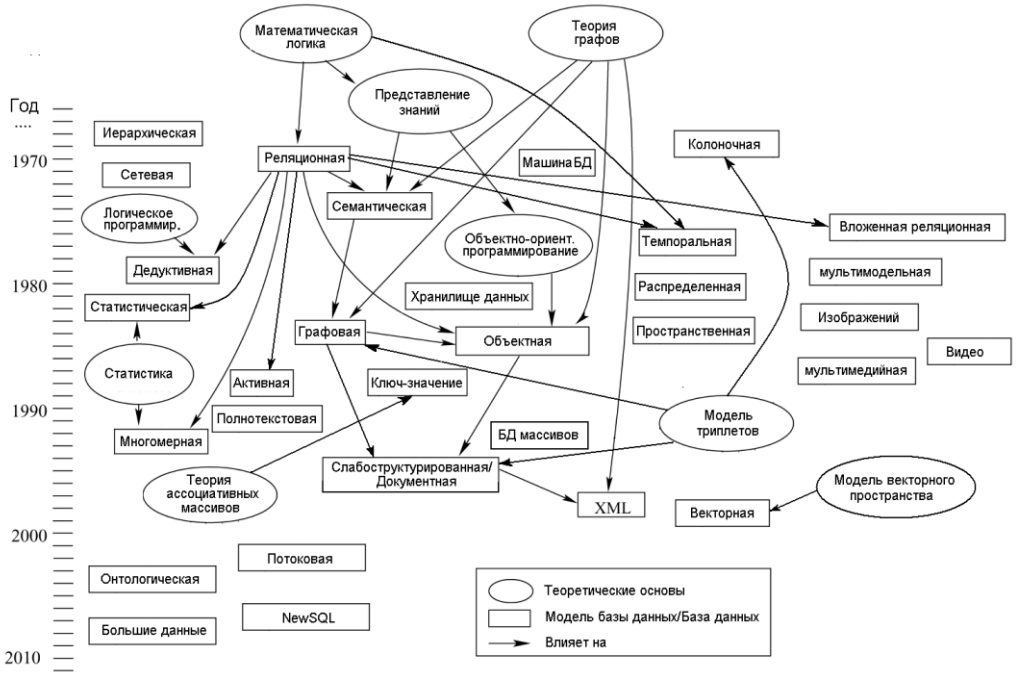
В статье описывается история зарождения, становления и развития баз данных. Особое внимание обращается на модели данных, собственно базы данных и системы управления базами данных. Также обращается внимание на языки баз данных, их проектирование, оптимизацию, стандартизацию. За основу изложения были взяты типы баз данных с точки зрения моделей данных, которые они поддерживают. С исторической точки зрения мы чисто условно разделили весь охватываемый период на десятилетние этапы, хотя многие исследования и разработки выходят за границы десятилетий, в которых они активно развивались.

Мы не претендовали на детальный анализ истории развития баз данных с полным охватом всех аспектов. Достаточно сказать, что по состоянию на начало 2022 года на сайте https://dblp.org, содержащем довольно полную библиографию в области компьютерных наук, было расположено более 56 000 библиографических ссылок со словом "database"

Наша цель заключалась в том, чтобы обрисовать общую картину истории развития рассматриваемых типов баз данных, выполнить некоторую систематизацию, описать наиболее значимые направления и вехи развития, привести библиографические ссылки для более подробного ознакомления с тем или иным вопросом.

В работе были использованы многочисленные статьи, аналитические и исторические обзоры, научно-технические отчеты, монографии ведущих ученых в области баз данных, труды международных конференций, материалы стандартов.

На следующем рисунке дается наглядное представление истории развития баз данных. Здесь в овалах представлены используемые теоретические основы. Их расположение не привязывается к годам. В прямоугольниках представлены либо модели баз данных, либо типы баз данных, если они не имеют собственных моделей, например, активные БД, хранилища данных, распределенные БД и др.



Этап 1. Становление баз данных (1960-1970)

60-е гг. — это период осознания необходимости отделения данных от программ, кристаллизации требований к такой независимой совокупности данных и, как следствие, зарождения и успешного становления технологий баз данных, формирования их методологических основ, становления концепции модели данных и появления первых двух классических моделей - иерархической и сетевой, рождения индустрии программного обеспечения систем баз данных и организационного оформления сообщества специалистов, работающих в этой области.

В начале 60-х годов компьютеры начали внедряться на производстве. Это были крупные компании, которые были в состоянии приобрести дорогостоящие оборудование. Компьютеры начали использоваться для автоматизации производственных процессов, включая учет получаемого сырья и деталей, производимой продукции, персонала и т.д. Компьютеры становились инструментом хранения и обработки больших объемов данных. При этом очень скоро стало очевидным, что технология создания автоматизированных систем, при которой существовала тесная связь между данными и программами, которые их используют, является не жизнеспособной, так как любые маломальские изменения в структуре данных приводили к необходимости переписывать программы. По мере усложнения структуры данных и роста их объема, увеличения количества пользователей и интенсивности их использования такой подход приводил к краху систем. Это привело к осознанию того факта, что надо разорвать эту связь и предоставить возможность независимого существования данных от программ. Это и послужило основой появления в информатике направления, которое со временем получило название «базы данных».

Чтобы понять в каких условиях зарождались базы данных, отметим, что это было время компьютеров практически без операционной системы, с 64 КБ оперативной памяти, в качестве носителей ввода данных использовались перфокарты и перфоленты, в качестве внешней памяти - в основном магнитные ленты и только немногие компании могли позволить себе магнитные диски объемом на 5 МБ и размером, превышающим трехстворчатый шкаф вместе с антресолями, наконец, общение человека с компьютером проходило через пульт управления или, в лучшем случае, пишущую машинку и АЦПУ (алфавитно-цифровое печатающее устройство).

Погрузка жесткого диска на 5 МБ компании IBM, 1956 г.

Система IDS. В 1960 году небольшая команда из General Electric, занимающаяся автоматизацией бизнес-процессов, приступила к проектированию системы Integrated Data Store (IDS) - интегрированное хранилище данных - под руководством Чарльза Бахмана (Charles William Bachman). В конце 1962 году был закончен прототип этой системы, а в начале 1964 года была выпушена первая промышленная версия IDS [1–3]. С ее появлением началась эра баз данных и звездный путь Бахмана.


Чарльз Бахман

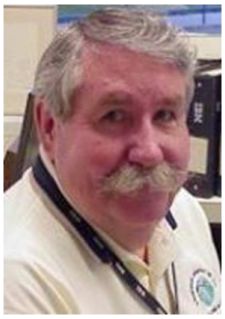
В IDS было впервые воплощено то, что сейчас считается основными функциями системы управления базами данных. IDS выполняла функцию посредника между прикладными программами и файлами, в которых хранились данные. Программы не могли напрямую манипулировать данными.

Вместо этого они должны были обращаться к IDS, чтобы она выполняла соответствующие действия от их имени. Как и современные системы управления базами данных, IDS позволяет явно создавать, хранить и манипулировать метаданными, хотя делается это слишком примитивным образом. В IDS были реализованы в простейшем виде функциональные возможности, которые впоследствии получили название независимости данных от программ. Бахман разработал в IDS инновационную на то время систему «Диспетчер проблем» (Problem Controller), которая стала прообразом системы управления транзакциями. В IDS также была спроектирована и реализована система резервного копирования и восстановления данных на магнитных лентах. Наконец, была предусмотрена функция запрещения доступа к определенным частям базы данных конкретным пользователям. IDS стала прообразом системы управления базами данных, поддерживающей сетевую модель данных.

IDS развивалась, совершенствовалась и использовалась многие десятки лет. В настоящее время IDS используется в ряде компаний, где показывает отличные результаты производительности на терабайтных массивах данных.

В 1973 году Бахман был награжден самой престижной в области информатики премией Алана Тьюринга за выдающийся вклад в технологию баз данных. Он был первым лауреатом премии Тьюринга без степени доктора философии, первым с опытом работы в области техники, а не науки, и первым, кто провел всю свою карьеру в промышленности, а не в научных кругах. Он также был первым, кто получил эту премию за работы по базам данных.

Система IMS. В 1965 году фирма IBM получила заказ на создание автоматизированной системы для учета огромного количества изделий, деталей и материалов, которые должны были использоваться при выполнении космической программы НАСА "Аполлон" - полета человека на Луну. Эта система первоначально получила название Information Control System - ICS, которая по завершению разработки была переименована в Information Management System – IMS (система управления информацией). Согласно [4] в IMS за основу была принята модель данных, разработанная в середине 60-х годов компанией North American Rockwell. В 1968 году IMS была предъявлена заказчику, а уже в 1969 года стала доступной в мире информационных технологий [5–7]. С тех пор и практически по настоящее время фирма IBM развивает IMS, переносит на различные платформы и операционные системы, расширяет функциональные возможности. Это, по сути, была первая успешная попытка создания промышленного варианта СУБД, хотя она так и не называлась в то время. Главным архитектором IMS был Верн Уоттс (Vern Watts). Он возглавлял эту работу с момента её проектирования и вплоть до своей кончины в 2009 году.


Верн Уоттс

IMS поддерживает иерархическую модель данных. Она состоит из схемы и экземпляров. На схемном уровне основным строительным блоком является сегмент, состоящий из совокупности полей. Сегменты связываются направленными бинарными связями, сегмент, из которого выходит связь, называется родительским, а в которого поступает - дочерним. Каждый сегмент может иметь не более одного родительского сегмента и множество дочерних. Сегмент без родителя называется корневым, а без дочерних сегментов – листьями. На уровне экземпляров связь между сегментами означает, что один экземпляр родительского сегмента связывается со многими экземплярами дочернего сегмента. Экземпляр иерархической структуры содержит один экземпляр корневого сегмента. Таким образом, иерархическая модель естественным образом представляет связи один-ко-многим. Следует отметить, что отсутствует строгая формальная спецификация иерархической модели данных и она, как правило, освещается так, как это было определено в IMS.

Система Total. В 1968 г. Томас Нис (Thomas Nies), Клод Богардус (Claude Bogardus) и Том Ричли (Tom Richley) основали компанию Cincom Systems, а уже в 1969 г. была выпущена первая версия СУБД Total [8]. С точки зрения многих пользователей и специалистов система Total являлась серьезным конкурентом IMS на компьютерах IBM. В отличие от IMS и большинства других СУБД того времени Total не ограничивалась одним типом компьютеров. По сравнению с IMS управлять Total довольно легко и более и эффективно.

Базовой структурой данных Total является двухуровневая иерархия, содержащая одну запись-владельца (master) и множество записей-членов (details). Эти типы записей могут быть связаны таким образом, чтобы создавать сложные структуры данных. Эта структура напоминала сетевую структуру первых версий IDS.


Томас Нис

В Total поддерживалось обращение из Cobol, Fortran, PL/1 и Assemler. Язык манипулирования напоминал спецификацию Codasyl. Был реализован механизм защиты базы данных, который включал динамическую регистрацию, периодическое резервное копирование (дамп) и рестарт, предотвращение одновременного обновления данных. Поддерживался режим одновременной работы многих прикладных программ. Был реализован механизм независимости на уровне отдельных элементов данных. Для каждой программы можно было выделить доступное подмножество базы данных помощью механизма, подобного подсхемам.

На начало 70-х годов Total имела самое большое количество пользователей среди всех действующих в то время СУБД. Считается, что на начальном этапе компания Cincom Systems внесла существенные вклад в развитие СУБД.

Система Adabas. Adabas (Adaptable Database System - адаптивная система баз данных) — система управления базами данных компании Software AG, Германия [9]. Впервые выпущена для мейнфреймов IBM в 1971 году. Изначальная модель данных — на базе инвертированного индекса. Подход Adabas отличается от сетевой модели данных, однако обеспечивает возможность поддержки полной сетевой структуры за счет неявных отношений. В момент создания язык манипулирования Adabas представлял собой расширение языков программирования КОБОЛ и ПЛ/1. В 1980-е годы дополнена элементами реляционной модели. В период взлёта популярности в середине 1980-х годов реляционных СУБД, была одной из самых продаваемых систем управления базами данных. Эксплуатируется по настоящее время.

Поддерживает многозначные атрибуты и согласно сайта DB-Engines (https://db-engines.com/en/ranking/multivalue+dbms) является самой популярной до сих пор среди многозначных СУБД.

IDS, IMS, Total, Adabas относятся к классу так называемых *навигационных баз данных*. Этот термин был введен Чарльзом Бахманом в своей статье [10], приуроченной получению премии Тьюринга. Суть этого класса заключается том, что записи данных могут связываться между собой различными ссылками, создавая тем самым сложную структуру данных, а язык манипулирования позволяет осуществлять произвольную навигацию по этим ссылкам для получения доступа к требуемым записям. Идея навигационных систем была порождена появлением магнитных дисков, которые, в отличие от магнитных лент, перфолент и перфокарт, предполагающих только последовательный доступ, предоставляли прямой доступ.

В заключение этого раздела отметим, что сам термин база данных (database) появился в начале 1960-х годов. По мнению Уильяма Олле (T. William Olle) [4] этот термин впервые был введён в употребление на симпозиумах, организованных компанией System Development Corporation (SDC) в 1963 и 1965 годах, хотя понимался сначала в довольно узком смысле. В широкое употребление в современном понимании термин вошёл лишь в начале 1970-х годов [11].

Литература

1) Bachman Charles W. Integrated Data Store - The Information Processing Machine That We Need! General Electric Computer

Users Symposium. Kiamesha Lake. New York May 17-18, 1962
2) IDS Reference Manual GE 625/635, GE Inform. Sys. Div., Pheonix, Ariz., CPB 1093B, Feb. 1968.
3) Bachman Charles W. “The Origin of the Integrated Data Store (IDS): The First Direct-Access DBMS,” IEEE Annals of the History of Computing, Vol. 31, Num. 4, Oct-Dec 2009, pp. 42-54.
4) Olle T. William. The CODASYL Approach to Data Base Management. Chichester, England: Wiley-Interscience, 1978, 287p.
5) History of IMS: Beginnings at NASA. - https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.imsintro.doc.intro/ip0ind0011003710.htm#ip0ind0011003710
6) Long R., Harrington M., Hain R., Nicholls G. IMS Primer. - http://www.redbooks.ibm.com/redbooks/pdfs/sg245352.pdf
7) Information Management System/360, Application Description Manual H20-0524-1. IBM Corp., White plains, N.Y., July 1968.
8) Nies T. Cincom Systems' Total. Annals of the History of Computing, IEEE. 2009, vol. 31, No 4, pp. 55-61.
9) ADABAS. - https://en.wikipedia.org/wiki/ADABAS
10) Bachman Charles W. "The programmer as navigator". Communications of the ACM, November 1973, Vol. 16 No. 11, pp. 653-658
11) Haigh T. How Data Got its Base: Information Storage Software in the 1950s and 1960s // IEEE Annals of the History of Computing. 2009, Vol. 31, No. 4, pp. 6-25

Этап 2. Бурное развитие (1970-1980)

70- годы - это годы бурного развития баз данных, создание основ технологии баз данных. Они ознаменовались, прежде всего, исследованиями рабочей группы CODASYL по базам данных (CODASYL DBTG), которая специфицировала сетевую модель, языки определения и манипулирования данными. В этот период было определено и изучено множество моделей данных, включая семантические. В 1976 г. Петер Чен определил ER- модель. Специфицирована трехуровневая архитектура баз данных ANSI/X3/SPARC, которая стала классической, проведены исследования по концептуальному моделированию предметных областей. Заложены основы индустриального производства СУБД и другого программного обеспечения баз данных. Наконец, было реализовано большое количество промышленных СУБД, которые были востребованными последующие несколько десятков лет. В 1973 году Чарльз Вильям Бахман был награжден самой престижной в области информатики премией Алана Тьюринга за выдающийся вклад в технологию баз данных.

К концу 60-х годов научное сообщество пришло к осознанию того факта, что системы управления базами данных (СУБД) становятся центральным звеном в автоматизированных информационных системах. Однако к этому времени еще не было ясного понимания того, что собой представляет СУБД, каким требованиям она должна удовлетворять, какие модели данных должна поддерживать, каким архитектурным решениям должна соответствовать. Но уже в начале 70-х годов появились первые отчеты и статьи, в которых давались предложения по конкретным системам [1], а также формулировались требования к СУБД [2, 3].

Инфологическая и даталогическая модели

Уже в 60-х годах ученые, работающие в области информационных систем, пришли к пониманию того, что в компьютерной системе должны быть представлены не только данные, но и их семантика.

В середине 60-х годов шведский ученый Бордже Лангефорс (Börje Langefors) ввел понятия инфологической и даталогической моделей (infological and datalogical models)., которые он развивал на протяжении 15 лет [4–6]. Даталогическая модель – это совокупность структурированных и взаимосвязанных данных и способы оперирования ими. Инфологическая модель – это модель представления информации (то есть семантики) о данных. Эти термины используются по настоящее время, хотя со временем появился термин «семантическая модель» как модель предметной области, предназначенная для представления семантики предметной области на самом высоком уровне абстракции.


Бордже Лангефорс

В 1999 году Б. Лангефорс получил престижную премию LEO за выдающиеся достижения в области информационных систем Международной ассоциации по информационным системам. А в 2010 году Шведская академия по информационным системам учредила премию Б. Лангефорса за лучшую докторскую диссертацию Швеции в области информатики и информационных систем.


Б. Сундгрен

Идеи Лангефорса в дальнейшем были развиты и адаптированы к технологиям баз данных шведским ученым Б. Сундгреном (Bo Sundgren) [7]

Архитектура баз данных ANSI/X3/SPARC

С появлением первых СУБД возникло новое понятие – схема данных (описание данных), которое отсутствует при файловой организации данных. Спецификация этой схемы и манипулирование данными выполняется уже языковыми средствами СУБД – ЯОД (язык описания данных) и ЯМД (язык манипулирования данными). Взаимодействие СУБД с прикладной программой осуществляется с помощью разработки специального интерфейсного модуля, в котором специфицируются объекты базы данных, требуемые этой программе, и необходимые операции над этими объектами, как это делается, например, в СУБД Adabas. Прикладная программа обращается к этому модулю через соответствующую точку входа и передает ему определенные параметры, уточняющие запрос. В ответ программа получает требуемые данные. Это так называемая *одноуровневая архитектура*. Этот единственный уровень составляет схема базы данных. Следующим шагом к усовершенствованию было введение *двухуровневой архитектуры*. Суть ее заключается в том, что помимо уровня схемы вводится уровень подсхемы - фрагмента общей схемы, создаваемый для каждого приложения и описывающий данные, которые требуются этому приложению. Двухуровневая архитектура была принята в IMS. Наконец, в ANSI была определена *трехуровневая архитектура* баз данных, которая стала классической на многие десятилетия и о которой речь пойдет далее.

В ноябре 1972 году подкомитет SPARC (Standard Planning and Requirements Committee) комитета X3 (Committee on Computers and Information Processing) Американского Национального Института Стандартов (ANSI) создал рабочую группу ANSI/X3/ SPARC DBMS для исследования возможностей и выработке рекомендаций по стандартизации СУБД. Сначала группу возглавил Томас Стил (Thomas B. Steel, Jr), а затем - Дионисиос Цикритзис (Dionysios Tsichritzis).


Дионисиос Цикритзис

Первоначальной задачей группы было исследование вопроса, следует ли вообще решать проблему стандартизации СУБД, и если да, то что именно должно быть стандартизировано.

В результате группа пришла к выводу, что стандартизации могут быть подвергнуты только интерфейсные составляющие СУБД [8].

В связи с этим была поставлена задача определения множества компонент, из которых должна состоять СУБД, интерфейсы между которыми могли бы стать кандидатами на стандартизацию. В основу выявления этих компонент были положены следующие концептуальные положения. Во-первых, существует реальный мир, информационная модель которого должна найти свое отражения в базе данных. Во-вторых, с учетом конкретных потребностей, в сознании людей отражаются их личные представления о том, что собой представляет реальный мир. Наконец, этот реальный мир материализуется в виде совокупности символов, в текстовом или электронном виде. Именно это триединство нашло отражение в предложенной этой группой покомпонентной структуре баз данных, которая была названа трехуровневой архитектурой баз данных ANSI-SPARC и которая получила всеобщее признание в среде разработчиков СУБД. Данная архитектура является актуальной по настоящее время. Она предполагает наличие концептуального, внешнего и внутреннего уровней. *Концептуальный уровень* предназначен для описания концептуальной информационной модели предметной области (ПО). *Внешний уровень* определяет пользовательское представление БД. Это та часть БД, которая соответствует потребностям конкретного пользователя, причем эта часть представляется в том виде, который удобен пользователю. *Внутренний уровень* предназначен для описания физического хранения БД. Между этими уровнями существуют отображения концептуальный-внешний и концептуальный-внутренний, Эта трехуровневая архитектура обеспечивает необходимые условия достижения логической и физической независимости данных от программ. В свою очередь мощность механизмов описания отображений определяет степень достаточности достижения упомянутых двух видов независимостей. Результаты деятельности этой рабочей группы были представлены в отчетах [9, 10].

В 1977 году Томас Стил получил «Награду за выдающиеся заслуги» (Distinguished Service Award) ассоциации ACM.

Предложения КОДАСИЛ

Вклад CODASYL в технологию баз данных связывают с созданием сетевой модели данных. В 1967 г. в КОДАСИЛ (CODASYL - Conference on Data Systems Languages) была учреждена специальная Рабочая группа по базам данных (CODASYL Data Base Task Group — DBTG). Одна из первоочередных задач Рабочей группы состояла в создании средств управления базами данных для языка Кобол. В дальнейшем эта задача была существенно расширена и сформулирована как разработка концепции, архитектуры и языковых спецификаций баз данных общего назначения. В 1971 году, осознавая важность исследований по спецификации языковых средств баз данных, был создан Комитет КОДАСИЛ по языку описания данных (CODASYL Data Description Language Committee). В результате деятельности этих двух групп были опубликованы отчеты [1, 11, 12], которые вызвали значительный резонанс, были заслуженно признаны специалистами по базам данных и на долгие годы стали образцом спецификации баз данных. В этих отчетах, исходя из единых позиций и в тесной взаимосвязи, впервые были строго специфицированы:

- сетевая модель данных, идеи которой были заложены Чарльзом Бахманом в системе IDS, и которая получила название модели данных КОДАСИЛ (CODASYL Data Model);
- трехуровневая архитектура баз данных, которая впоследствии была принята и развитие в ANSI/X3/SPARC DBMS;
- языки описания данных (ЯОД) на всех трех уровнях (язык схемы, язык подсхемы, язык схемы хранения);
- в ЯОД также включены такие функции администрирования, как проверка достоверности, управление доступом, настройка, распределение ресурсов, защита данных, целостность данных;
- отображения между схемой и подсхемой, а также схемой и схемой хранения;
- язык манипулирования данными, предназначенный для навигации по сетевой структуре с целью спецификации тре-

буемой записи для ее обновления, удаления, либо для вставки новой записи.

По результатам работы Комитетов КОДАСИЛ было опубликовано множество материалов, среди которых отметим монографию Уильяма Олле (T. William Olle) [13].


Уильям Олле

Следует отметить, что предложения КОДАСИЛ были специфицированы для систем с включающим языком, то есть они предполагали, что работа с базой данных осуществлялась через язык программирования. Это полностью соответствовало принятой в то время технологии обработки данных и поэтому способствовало эффективной реализации в существующей вычислительной среде.

Сетевые СУБД. Согласно спецификациям КОДАСИЛ было реализовано ряд СУБД, среди которых: IDMS (Integrated Database Management System) компании Cullinane Database Systems, которая стала основной сетевой СУБД для мейнфреймов и самой популярной в 70-80-е годы прошлого столетия, DMS1100 (UNIVAC), IDS/II (Honeywell), DBMS10/20 (DEC).

Концептуальное моделирование

В ноябре 1977 г. комитет ISO по языкам программирования принял решение о создании рабочей группы по исследованию различных аспектов использования концептуальных схем в системах управления базами данных с целью обеспечения основы для стандартизации в данной области. Сначала эту группу возглавил Томас Бревард Стил-младший (Thomas Brevard Steel Jr.), а затем Дональд Жардин (Donald A. Jardine). В результате деятельности этой группы в 1982 был выпущен отчет [14] под редакцией Дж. Грийтусена (Joost J. Van Griethuysen).


Дж. Грийтусен

В отчете описывается роль и содержание концептуальной схемы, а также определяется связь концептуальной схемы с информационным моделированием и семантикой данных. Отмечается, важность точного определения как статических, так и динамических правил в концептуальной схеме. Обсуждается архитектурная роль концептуальной схемы и то, как системы управления базами данных вписываются в такую архитектуру. В этом отчете впервые были четко сформулированы следующие требования к концептуальной схеме:

- это единая основа однозначного понимания сути предметной области (ПО) всеми заинтересованными лицами;
- она включает только концептуально релевантные аспекты ПО;
- это средство определения допустимой эволюции информационной базы данных и разрешенного манипулирования информацией о ПО;
- это базис для интерпретации внешних и внутренней схем;
- это основа отображения внешних схем во внутреннюю и наоборот.

Модели данных

Согласно [15] термин "модель данных" стал использоваться в начале 70-х годов после публикации фундаментальной работы Эдгара Кодда (E. Codd) [16]. Однако еще во второй половине 60-х гг. стали появляться первые модели данных. В результате развития технологии баз данных было предложено множество средств и методологий концептуального моделирования, в частности, к ним относятся описываемые далее модели.

Модель «объектов-ролей»


Екхард Фолкенберг

Модель объектов-ролей (ORM - Object-Role Model) Екхарда Д. Фолкенберга (Falkenberg, Eckhard D) [17, 18], которая была развита другими учеными (С. Нейссен, Р. Меерсман, Д. Вермейр, Т. Халпин - Sjir Nijssen, Robert Meersman, Dirk Vermeir, Terry Halpin). ORM предполагает представление информационной модели в виде объектов (сущностей), которые играют те или иные роли (представляемые в виде связей между объ-

ектами). В отличие от объектно-ориентированного подхода и подхода сущность-связь ORM не предполагает существование атрибутов, они представляются в виде ролей фактов, которые вместе с правилами моделируются в виде естественных предложений, легко понимаемых и проверяемых пользователями.

Модель данных, основанная на бинарных связях

У истоков происхождения модели бинарных связей (BR - Binary Relations) стоят работы таких авторов, как Абриаль [19] (семантическая бинарная модель), Браччи [20], Дурхольц [21]. Суть этого подхода к моделированию заключается в том, что любой «элемент» информации представляется с помощью экземпляров бинарных ассоциаций, то есть высказываний, в состав которых входят только два терма. В частности, М. Сенко в рамках проекта DIAM (Data Independence Access Method) определил бинарную сетевую модель, разработал на базе этой модели язык FORAL и исследовал возможности базирующегося на нем пользовательского интерфейса [22-24].

Семантические модели

Отметим работы Дж. Смита и Д. Смит по моделям абстракции, агрегации и обобщения данных [25, 26], а также семантическую модель данных SDM Хаммера и МакЛеода [27]. В статье [28] приводится перечень около 20 семантических моделей баз данных.


Фред Лоховски

Итоги развития моделей данных к началу 80-х гг. подведены в широко известной монографии Д. Цикритзиса и Фреда Лоховски (F. Lochovsky) [29].

ER-модель

Вместе с тем, наибольшую популярность заслужено приобрёл подход сущность–атрибут-связь, называемый как подход сущность-связь (ER-подход). Свое начало он берет от диаграмм структур данных Бахмана [30], а также модели Инглеса [31].

Наиболее полно впервые эту модель описал П.П. Чен (Петер Пин-Шен Чен - Peter Pin-Shan Chen) [32]. ER-модель данных стала общепризнанной в мире и служит основой многих методик системного анализа, концептуального моделирования и проектирования баз данных. Она базируется на простой идее, что структурная составляющая концептуальной модели предметной области может быть представлена в виде сущностей, атрибутов и связей. *Сущность* – это любой реальный ли абстрактный объект произвольной природы, который представляют самостоятельный интерес. *Атрибут* – это свойство сущности, способствующее качественному или количественному ее описанию, идентификации, классификации или отражению ее состояния. Наконец *связь* – это некоторая, представляющая интерес, ассоциация между различными сущностями (классами сущностей).


Петер Чен

После публикации статьи Чена появилось множество статей, посвященных исследованию различных аспектов ER-моделировния предметных областей. Например, в общем случае предполагается существование n-арных связей, а Ричард Баркер (Barker Richard) предложил ER-модель только с бинарными связями [33], которая имеет определенные преимущества.

В связи с широким использованием ER-модели [34] было предложено множество различных ее расширений и обобщений [35–38], которые в конечном итоге привели к определению иерархической ER-модели (ER-модели более высокого порядка) [38].

В статье [39] ER-модель расширена включением элементов семантизации данных. Также была предложена темпорально-расширенная ER-модель [40], которая предоставляет возможность включать темпоральную информацию в концептуальную информационную модель и представлять ее в реляционной модели Для поддержания темпоральных запросов язык SQL был расширен возможностями определения, поиска и управления историческими отношениями.

Со временем было предложено еще несколько темпоральных ER-моделей, обзор которых приведен в [41]. Наконец, существует пространственная ER-модель (см. «Пространственные базы данных»).

Литература

1) CODASYL: "Data Base Task Group Report", ACM (New York 1971).
2) GUIDE-SHARE: "Data Base Management System Requirements", SHARE Inc. (New York 1970)
3) CMSAG Joint Utilities Project: "Data Management System Requirements", CMSAG (Orlando, FL 1971)
4) Langefors B. Theoretical Analysis of Information Systems. 402 S. m. Fig. Lund/Kopenhagen/Oslo 1966. Akademisk Forlag/Universitetsforlaget
5) Langefors B. Information systems theory. Inf. Syst. 1977, Vol. 2, No. 4, pp. 207-219
6) Langefors B. Infological models and information user views. Information Systems Volume 5, Issue 1, 1980, pp. 17-32
7) Sungren Bo. An Infological Approach to Data Bases. National Central Bureau of Statistics, Sweden, Stokholm, 1973. 294 p.
8) SPARC: "Outline for Preparation of Proposals for Standardization", Document SPARC/90, CBEMA (Washington, DC 1974).
9) ANSI/X3/SPARC, 'Study Group on Data Base Management Systems: Interim Report 75-02-08' // Newsletter ACM SIGMOD Record, FDT, Vol 7, No. 2, 1975. – P. 1-140
10) Tsichritzis D.C., Klug A. "The ANSI/X3/SPARC DBMS Framework". Report of the Study Group on a Database Management System". Information Systems, Vol. 3, No. 4, 1978.
11) CODASYL/Data Description Language Committee (DDLC), "June 73 Report". CODASYL Data Description Language Committee Journal of Development, June 1973
12) CODASYL Data Description Language Committee Journal of Development, 1978.
13) Olle T. William. The CODASYL Approach to Data Base Management. Chichester, England: Wiley-Interscience, 1978, 287p.
14) Concepts and Terminology for the Conceptual Schema and the Information Base, van Griethauzen, J.J., Ed., ISO TC97/SC5/WG3, 1982, Publ. 695.
15) Kogalovsky M. R. Encyclopedia of databases technologies (Rus). Moscow, Finance and Statistics,. 2005. — 800 p.
16) Codd E.F. “A Relational Model of Data for Large Shared Data Banks,” Communications of the ACM, Vol. 13, No. 6 (June 1970), pp. 377-397
17) Falkenberg E,D. Structuring and Representation of Information at the Interface Between Data Base User and Data Base Management System. Diss. Univ. Stuttgart (1975).
18) Falkenberg E., Concepts of Modelling Information, Proc. of the IFIP Working Conf. on Modelling in Data Base Management Systems, Nijssen, G.M., Ed., NorthHolland, 1976, p. 95-109.
19) Abrial Jean-Raymond, Data Semantics, In: J. W. Klimbie, K. L. Koffeman (eds.), Database Management, Proceedings IFIP TC2 Conference. Grgese, 1974., North-Holland Publishing Company, pp.1-60.
20) Bracchi G., Paolini P., Pelagatti G. “Binary Logical Associations in Data Modelling,” in J. M. Nijssen (ed.), Modelling in Database Management Systems (Proc. IFIP TC2 Conference, Freudenstadt), North-Holland Publishing Company, Amsterdam, The Netherlands, 1976.
21) Durchholz R. and Richter G., "Concepts for data base management systems". In: Data Base Management, J. W. Klimbie and K. L. Koffeman, (eds.),
22) Senko, M.E., Conceptual Schemas, Abstract Data Structures, Enterprise Descriptions, In: International Computing Symposium, Liege, Belgium, 1977, North-Holland Publishing Company.
23) Senko M.E., Altman E.B., Astrahan MM., Fehder P.L. Data Structures and Accessing in DataBase Systems. IBM System J., v. 12, no. 1 (1973).
24) Senko M.E., "The DDL in the Context of Multilevel Structured Description: DIAM II with FORAL". Proc. of the IFIP TC-2 Special Working Conference on Data Base Description, pp.239-257, Jan. 1975

25) Smith J.M. and Smith D.C.P. Database Abstractions : Aggregation and Generalization. ACM Trans, on Database Syst, v. 2, no. 2, 1977, pp. 105133
26) Smith J.M. and Smith D.C.P. Database Abstractions: Aggregation. Comm. of the ACM, v. 20, no. 6, 1977, pp. 405-413
27) Hammer, M. and McLeod, D., Database Description with SDM: A Semantic Database Model, ACM Transactions on Database Systems,1981, Vol. 6, No. 3, pp. 351-386.
28) Abiteboul, S., Hull, R., IFO: A Formal Semantic Database Model, ACM Trans. Database Syst. 12, 4 (1987), 525-565.
29) Tsichritzis D.C., Lochovsky F.H., Data models, Prentice-Hall, Englewood Cliffs, N.J., 1982, 381 p.
30) Bachman, C. W., "Data Structure Diagrams", Data Base, 1969, No 1, 2, pp. 4-10.
31) Engles R.W. A Tutorial on Database Organization, Annual review in automatic programming, Vol 7. Part I, Pergamon Press, 1972, 93 p.
32) Chen P.P. The Entity-Relationship Model — Toward a Unified View of Data // ACM Transactions on Database Systems (TODS), 1976. — Vol. 1, No. 1. — P. 9–36.
33) Barker R. Case*Method: Entity Relationship Modelling Publisher: Addison-Wesley, 1990, 240 p.
34) Embley D., Thalheim B., editors. Handbook of conceptual modelling: its usage and its challenges. Springer; Berlin 2011
35) Gogolla M. An extended entity-relationship model – fundamentals and pragmatics. LNCS, vol. 767. Berlin: Springer; 1994.
36) Hartmann S. Reasoning about participation constraints and Chen's constraints. In: The Fourteenth Australian Database Conference, Adelaide, Australia. Conferences in Research and Practice in Information Technology; 2003. p. 105–113.
37) Hohenstein U. Formale Semantik eines erweiterten Entity-Relationship-Modells. Stuttgart: Teubner; 1993.
38) Thalheim B. Entity-relationship modeling – foundations of database technology. Berlin: Springer; 2000.
39) Teorey, T.J., Yang, D. and Fry, J.P. A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model, ACM Computer Surveys, 1986, Vol.18, No. 2. pp. 197-222
40) Vincent S. Lai, Jean Pierre Kuilboer, Jan Lucille Guynes. Temporal databases: model design and commercialization prospects. ACM SIGMIS Database: the DATABASE for Advances in Information Systems, 1994, Vol. 25, No 3, pp. 6-18
41) Gregersen H., Jense C.S. Temporal Entity-Relationship Models—a Survey. IEEE Transactions on Knowledge and Data Engineering, 1999, Vol. 11, No. 3, pp. 464 - 497

Этап 3. Эпоха реляционных баз данных (1970-1990+)

В начале 80-х годов появились первые промышленные реляционные СУБД, которые к концу 80-х гг. быстро завоевали рынок и стали господствующими практически на всех распространенных аппаратно-программных платформах и не утратили свое преимущество по настоящее время. Тем не менее, основы реляционной модели данных и реляционных СУБД были заложены в предыдущем десятилетии, родоначальником которых стал Эдгар Франк Кодд, определивший реляционную структуру данных, алгебру и исчисление, заложивший основы теории зависимостей и нормальных форм, сформулировавший требования реляционности баз данных. Эти и другие исследования в конечном итоге привели к созданию теории реляционных баз данных. Базы данных превратились из описательной науки в формальную. В 1981 Эдгар Франк Кодд был награжден премией Тьюринга за фундаментальный и продолжительный вклад в теорию и практику систем управления базами данных, в особенности реляционного типа. Было открыто множество исследовательских проектов по исследованию и созданию экспериментальных СУБД, предложено множество языков запросов реляционных баз данных, изучены вопросы оптимизации выполнения запросов, структуры хранения, методы доступа, защиты, обеспечения целостности. В 1986 г. проявился первый стандарт SQL и с тех пор он стал единственным официальным языком внешнего интерфейса реляционных СУБД. Были проведены обширные исследования по управлению транзакциям, за которые в 1998 г. Джеймс Николас Грей был награжден премией Тьюринга.

Реляционные базы данных

В 70- годах ученые, занимающиеся базами данных, были уверенны, что будущее баз данных лежит в создании все более и более сложных структур данных, которые бы позволяли адекватно представлять информационную модель данных произвольных предметных областей. Высказывались мнения, что в ближайшее время структуры будут настолько сложными, что в базах данных соотношение полезной информации и той, которая ее поддерживает, будет 1:30.

Вклад Э.Ф. Кодда в реляционные базы данных

И вот на этом фоне в 1970 году публикуется статья [1] малоизвестного на то время британского ученого Эдгара Франка Кодда (Edgar Frank Codd), работающего в компании IBM, в которой он предложил наиболее простую структуру данных, представляющую собой одномерную, плоскую, нормализованную таблицу.


Эдгар Франк Кодд

Одномерность означает, что имеется только одна, горизонтальная, шапка и не может быть вертикальной шапки, как, например, в учебных планах ВУЗа. Плоскость свидетельствует о том, что в шапке не может быть полей, состоящих из множества подполей, например, чтобы поле ФИО состояло из подполей Фамилия, Имя и Отчество. И наконец, нормализованность свидетельствует о том, что в ячейках таблицы может быть только атомарное (единственное) значение. Такая структура была названа *реляционным отношением*, так как она напоминает математическое понятие отношения. Также было принято считать, что такие отношения находятся в первой нормальной форме (First Normal Form – 1NF). В этой же работе он обосновал существование двух семейств реляционных языков, которые впоследствии были названы реляционным исчислением и реляционной алгеброй.

В 1971 году Кодд публикует статью [2], в которой он приводит пример того, как логика исчисления предикатов может быть использована для создания высокоуровневого языка реляционной базы данных. Описанный им язык ALPHA был первым языком класса реляционного исчисления. Хотя ALPHA не был реализован, однако он оказал серьезное влияние на создание последующих коммерческих реляционных языков.

В 1972 году Кодд публикует следующую замечательную статью [3], в которой он:

- дает формальное определение реляционной алгебры и реляционного исчисления (кортежно-ориентированного);
- формулирует тезис реляционной полноты селективных возможностей языков запросов к реляционной базе данных на основе реляционного исчисления. Он был единодушно воспринят в ученом мире баз данных и в дальнейшем все создаваемые языки запросов проверялись на реляционную полноту;
- приводит алгоритм редукции произвольного выражения реляционного исчисления в семантически эквивалентное выражение реляционной алгебры, тем самым устанавливая ее реляционную полноту. Этот результат впоследствии был назван теоремой Кодда. (Впоследствии Палермо (Palermo) [4] улучшил этот алгоритм с точки зрения повышения его эффективности.)

Реляционная модель с самого начала подвергалась критике за простоту ее структуры. Это, в частности, отразилось на конференции 1974 года «SIGMOD Workshop on Data Description, Access, and Control», на которой развернулись дебаты между сторонниками реляционного и сетевого подхода, главными спикерами которых выступили Кодд и Бахман. Позиция Кодда на этих дебатах отражена в статье [5]. В конце концов, реляционная модель получила всеобщее признание. Это объясняется тем фактом, что в ней удалось сформулировать языки высокого уровня (алгебра, исчисление), что позволило наиболее полно решить ту основную проблему, которая была поставлена перед базами данных, а именно, достижение независимости данных от программ. В свою очередь, повышение сложности структуры данных приводит к неминуемому снижению уровня языка манипулирования, что снижает возможности по достижению такой независимости.

Стремясь придать дополнительные возможности, в работе [6] Э. Кодд предложил повысить семантику реляционной модели, идеи которой используются до сих пор в коммерческих реляционных СУБД. Также следует отметить, что Э. Кодд в статье [7] определил понятие "модель данных" как тройку: структура данных, операции и ограничения целостности. С тех пор это определение модели данных используется в проблематике баз данных.

Теория зависимостей и нормальных форм

Реляционная модель дала серьезный толчок в развитии проектирования баз данных. Впервые задача логического проектирования БД приобрела строго формальный подход. Сущность этой теории заключалась в том, что на основе анализа различных видов зависимостей (ограничений целостности), которые существуют внутри реляционных отношений и между ними, выявлять нежелательные ситуации и устранять их с помощью обоснованных процедур эквивалентных преобразований. Как правило, такой процедурой является декомпозиция отношений, то есть разбиение отношения на несколько. Основоположником этой теории стал Э.Ф. Кодд, опубликовав работы [8–10]. В этих работах он определил понятие функциональной зависимости (Functional Dependency - FD) в реляционном отношении, сформулировал так называемые аномалии манипулирования отношениями, выявил две нежелательные разновидности FD, которые порождают эти аномалии, а именно, неполные FD и транзитивные FD, и предложил процедуру декомпозиции, которая устраняет эти разновидности FD в результирующих отношениях. Отношения, в которых отсутствуют неполные FD, получили название отношений во второй нормальной форме (2NF), а в которых отсутствуют неполные и транзитивные FD, - в третьей нормальной форме (3NF).

В 1981 Кодд был награжден премией Тьюринга за фундаментальный и продолжительный вклад в теорию и практику систем управления базами данных, в особенности реляционного типа. Кристофер Дейт написал книгу [11] - исторический обзор научного вклада Кодда в реляционную технологию.

С точки зрения структуры функциональных зависимостей 3NF все же обладала определенными аномалиями. В связи с этим

в 1974 г. Кодд вместе Раймондом Бойсом (Raymond F. Boyce) предложили усилить 3NF. Результирующая нормальная форма получила название нормальной формы Бойса-Кодда (Boyce–Codd normal form – BCNF) [12].


Раймонд Бойс

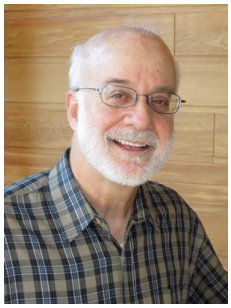
По справедливому замечанию Дейта, первоначально эту нормальную форму определил Ян Хит (Ian Heath) в статье [13]. Также отметим, что в этой статье он также доказал теорему о декомпозиции без потерь реляционного отношения при наличии FD, то есть декомпозиции, которая является эквивалентной по данным. Эта теорема была названа его именем (теорема Хита). Она используется при приведении отношений в 2NF, 3NF и BCNF.


Ян Хит

В 1974 году Вильям Армстронг (William Ward Armstrong) в статье [14] предложил систему аксиом FD (минимально полный набор правил вывода новых FD из заданных). Они получили название аксиом Армстронга. Они позволили определить и исследовать такие понятия, относящиеся к FD, как выводимость, полнота, замыкание, (минимальное) покрытие, эквивалентность. Полученные в этом направлении результаты способствовали решению задачи автоматизации проектирования баз данных.

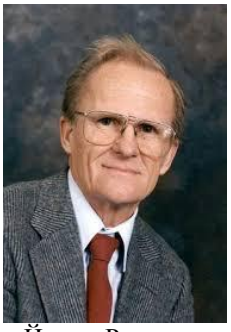

Вильям Армстронг

В 1977 году Рональд Феджин (Ronald Fagin) в статье [15] определил новый вид зависимости – многозначную зависимость (multivalued dependency MVD), наличие которой в отношении также вызывает аномалии манипулирования. Предложенная им форма, устраняющая эту ситуацию, была названа четвертной нормальной формой (Fourth Normal Form – 4NF), а алгоритм приведения в 4NF базировался на доказанной им теореме (теорема Феджина). В последующей статье [16] была предложена полная система аксиом MVD, а также две аксиомы, связывающие FD и MVD (выводимость MVD из FD и наоборот).

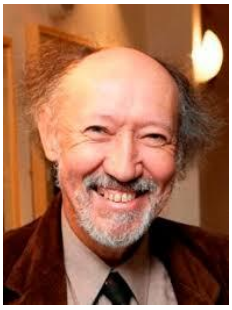

Рональд Феджин

Отметим, что независимо от Феджина многозначную зависимость также исследовал Заниоло [17]. Кроме того, Делобел [18] определил понятие "иерархической декомпозиции первого порядка", которое также связано с концепцией многозначной зависимости.

В 1978 г. Йорма Риссанен (Jorma Rissanen) определил зависимость по соединению (join dependency - JD) [19], которая явилась обобщением MVD (MVD является бинарной JD). На ее основе Феджин в статье [20] определил и исследовал проекционно-соединительную нормальную форму (Projection-Join Normal Form – PJ/NF), которая со временем получила название пятой нормальной формы (Fifth Normal Form – 5NF).


Йорма Риссанен

Наконец, Кристофер Дейт (Christopher J. Date) определил шестую нормальную форму (Sixth Normal Form – 6NF), как форму, в которой отсутствуют нетривиальные зависимости по соединению. Как отмечают многие исследователи, эта нормальная форма оказалась полезной в темпоральных базах данных. По утверждению Дейта [21] 6NF равносильна доменно-ключевой нормальной форме (DK/NF) Феджина (см. далее).


Кристофер Дейт

Приведенные до сих пор зависимости и нормальные формы относятся к так называемым классическим. Приведем еще несколько определенных и исследованных видов зависимостей, с подробным их анализом и структурой взаимосвязей между ними можно познакомиться в работе [22]:

– улучшенная 3NF (Improved 3NF) [23];
– нормальная форма элементарного ключа (Elementary Key Normal Form – EKNF) [24];
– нормальная форма суперключа (Superkey Normal Form – SKNF) [25];
– приведенная 5NF (reduced-5NF – 5NFR) [26];
– нормальная форма без избыточности (Redundancy Free Normal Form-RFNF) [27];
– нормальная форма с существенными кортежами (Essential Tuple Normal Form – ETNF) [28];
– доменно-ключевая нормальная форма Феджина (Domain-Key Normal Form – DK/NF) [29];
– иерархическая зависимость [30] и ее связь с иерархической структурой данных [31];
– зависимость по включению (Inclusion dependency) и нормальные формы по включению (Inclusion Normal Forms) [32–35].

В заключение отметим, что приведено только незначительное количество исследованных зависимостей. В книге [36] приведен перечень более 600 статей, посвященных теории зависимостей и нормальных форм, а в монографии [37] анализируются около 90 зависимостей, расклассифицированные на 6 основных типов.

Языки запросов реляционной модели

Реляционная модель дала существенный толчок исследованиям по созданию языков запросов. В своем обзоре [38] Дональд Чемберлин (Donald D. Chamberlin) предложил следующую классификацию языков реляционных баз данных: языки реляционной алгебры, языки реляционного исчисления, графические языки и языки, ориентированные на отображение. Дадим краткий обзор языков этих классов.


Дональд Чемберлин

Языки реляционной алгебры

Были предложены и экспериментально апробированы следующие языки/системы, базирующиеся на реляционной алгебре: система MACAIMS [39], разработанная в MIT, системы IS/1 [40] и PRTV (Peterlee Relational Test Vehicle) [41], разработанные в научном центе IBM в Питерли, Англия, система RDMS [42], созданная в исследовательской лаборатории General Motors. Во многих системах расширяется набор операций реляционной алгебры, вводя специфические. Параллельно проводились исследования по оптимизации выполнения выражений реляционной алгебры [43–47]. В [48] приводится обширный обзор исследований по анализу сложности операций и оптимизации запросов в реляционных базах данных.

Языки реляционного исчисления

Как мы уже отметили выше, первым языком запросов реляционной модели явился язык ALPHA Кодда [2], который непосредственно основывается на реляционном исчислении. ALPHA и позволяет пользователю, используя такие понятия, как переменные и кванторы, формулировать непроцедурные запросы. Впоследствии были предложены другие языки, которые, как и ALPHA, базировались на реляционном исчислении. К ним относятся QUEL [49], созданный в рамках научно-исследовательского проекта Ingres в Калифорнийском университете в Беркли, COLARD (**C**alculus **O**riented **LA**nguage for **R**elational **D**ata) [50], RIL [51].

Графические языки

В языках, относящихся к графическим, формулировка запросов производится не с использование традиционного линейного синтаксиса, а заполнением ячеек в бланках таблиц. Язык CUPID (Casual User Pictorial Interface Design - работа с графическим интерфейсом непрофессионального пользователя) [52–55] предоставляет пользователю графический язык запросов. CUPID содержит высокоуровневый меню-образный подъязык, который является внешним интерфейсом к системе INGRES.

Идея языка QBE (Query By Example - запрос по образцу) [56–60], который был разработан Моше М. Злуфом (Moshe M. Zloof), заключается в следующем. Пользователю предоставляются пустые бланки таблиц базы данных. Формулировка запроса - это заполнение бланков одним правильным ответом, а задача системы - на основании этого примера вывести все возможные правильные строки таблиц. Несмотря на простоту, было доказано [56], что QBE является реляционно полным языком. Разновидности этого языка были реализованы в СУБД PARADOX, DBASE IV, ACCESS. Последняя входит в состав Microsoft Office. М.М. Злуф также разработал язык OBE (Office-by-Example - офис по образцу) [61], который явился расширением QBE для офисных приложений.


Моше М. Злуф

Языки, ориентированные на отображение

В 1973 г. коллеги Кодда из лаборатории IBM в Сан Хосе Раймонд Бойс, Дональд Чемберлин и Вильям Кинг (William F. King) разработали язык SQUARE (**S**pecifying **QU**eries **A**s **R**elational **E**xpressions - спецификация запросов в виде реляционных выражений) [62, 63]

Использование SQUARE-подобного языка для описания множественных представлений (взглядов), а также управления целостностью данных и их авторизации описано в статье [64]. В отличие от реляционного исчисления SQUARE не использует кванторов и связанных переменных и поэтому не требует соответствующей математической подготовки. В языке запросы выражаются в виде естественных примитивных операций, которыми пользуются люди при поиске информации в таблицах. Большинство семантически **простых** запросов выражаются в языке просто и лаконично. Вместе с тем SQUARE является реляционно полным языком [63].

В 1974 г. Бойс и Чемберлин представили язык SEQUEL (**S**tructured **E**nglish **QUE**ry **L**anguage - структурированный английский язык запросов) [65], который явился усовершенствованным вариантом языка SQUARE. Измененный синтаксис языка был назван блочно-структурированным синтаксисом ключевых слов английского языка. SQUARE и SEQUEL были декларативными языками, то есть в них формулируется «что» надо найти, а не «как» это сделать, что характерно для процедурных языков. В 1975 г. был реализован экспериментальный вариант SEQUEL на базе разработанного интерпретатора [66]. Задача интерпретатора – минимизировать выполнение операций доступа к данным при выполнении запросов за счет сужения пространства поиска. Для этого были исследованы специальные оптимизирующие алгоритмы. Сам интерпретатор SEQUEL базируется на XRM (Extended n-ary Relational Memory [67, 68]) – системе, разработанной для хранения и поиска данных, представленных в виде n-арных отношений. XRM, в свою очередь, реализована на базе RM (Relational Memory) [69, 70], предоставляющей эффективный ассоциативный доступ к бинарным отношениям. Наконец, в 1976 г. был представлен язык SEQUEL2 [71], в котором уже были включены все основные средства для оперирования базами данных: определение, манипулирование и управление.

Оригинальный подход был предложен в языке APPLE (Access Path Producing Language – язык, порождающий путь доступа) [72], одним из авторов которого явился Карл Роберт Карлсон (Carl Robert Carlson). Язык предполагает использование в запросе только имен атрибутов отношений базы данных. Задача системы – на основании структуры базы данных определить множество отношений, необходимых для выполнения запроса, и определить путь доступа к ним.


Карл Карлсон

Экспериментальные исследования и разработки

Уже в начале 70-х гг. был реализован ряд ранних реляционных систем — MacAIMS (1970 г.), IS/1 (1972 г.) и PRTV, RENDEZVOUS (1974 г.) и др.

IS/1 и PRTV. IS/1 была первой в мире экспериментальной реляционной системой баз данных с ограниченными возможностями, реализованной в научном центре IBM в Питерли, Великобритания в 1970–1972 гг. [73]. С учетом результатов, полученных при реализации IS/1, была разработана СУБД PRTV (Peterlee Relational Test Vehicle) [74], которая позволяла оперировать большими объемами данных, имела свой собственный язык запросов ISBL уровня реляционной алгебры и была однопользовательской.

System/R и DB2. В 1974 году в исследовательской лаборатории в Сан-Хосе компании IBM был инициирован проект System/R по созданию экспериментальной СУБД. Задача проекта – продемонстрировать возможность создания высокопроизводительных промышленных реляционных СУБД. За основу был взят язык SEQUEL, который в процессе разработки был переименован в SQL исходя из юридических соображений. К 1975 году был реализован пользовательский интерфейс упрошенного варианта языка [66]. Затем была реализована полнофункциональная многопользовательская версия System/R [75]. Наконец, на протяжении 1978-1979 годов System/R прошла всестороннюю практическую апробацию [76, 77], результаты которой продемонстрировали, что реляционные СУБД могут обеспечить высокую производительность. В 1979 г. проект System/R был завершен. Впоследствии краткая история экспериментальных исследований по проекту System/R была изложена Чемберлином и его коллегами в статье [78]. Используя полученный опыт, компания IBM в 1980 г. приступила, а в 1982 г. выпустила промышленную реляционную СУБД году под названием SQL/DS, которая впоследствии была переименована в DB2 и поддерживается по настоящее время на различных платформах и в различных конфигурациях. Она стала стратегическим программным продуктом компании IBM.

Oracle. В 1977 году трое молодых программистов из американской электронной компании Ampex Corporation, Ларри Эллисон (Larry Ellison), Боб Майнер (Bob Miner) и Эд Оутс (Ed Oates), вдохновленные идеями Кодда, основали компанию Software Development Laboratories (SDL) по созданию реляционной СУБД и приступили за разработку и маркетинг программы. В 1979 году компания была переименована в Relational Software Inc. В этом же году компания выпустила Oracle, первую коммерческую реляционную СУБД, в которой использовался язык SQL. Программа очень скоро стала популярной. В 1982 году компания была переименована в Oracle Systems Corporation. С тех пор Oracle является крупнейшим поставщиком реляционных СУБД на базе SQL.


Ларри Эллисон

Ingress. В 1973 г. два ученых исследовательской лаборатории Калифорнийского университета в Беркли Майкл Стоунбрейкер (Michael Ralph Stonebraker) и Юджин Вонг (Eugene Wong), заинтересовавшись исследованиями Кодда и результатами своих коллег из IBM по созданию System R, решили начать свой собственный проект по созданию реляционной СУБД. Разрабатываемая экспериментальная СУБД была названа INGRES (**IN**teractive **G**raphics and **RE**trieval **S**ystem). Последующие два года были проведены экспериментальные исследования и разработки. Были приняты проектные решения [79, 80], разработаны структуры хранения и методы доступа [81] Был разработан оптимизационный алгоритм выполнения операций соединения отношений, получивший название алгоритма Вонга-Юсефи (Wong-Youssefi algorithm) [82], исследован механизм предоставления альтернативных взглядов (view) путем подстановки в запросы пользователей их определений взглядов [83]. Авторизация и контроль целостности обеспечивался добавлением дополнительных предикатов к запросам пользователя [84]. Реализован механизм безопасного одновременного обновления


Майкл Стоунбрейкер


Юджин Вонг

базы данных [85], а также система защиты [86]. К 1976 году была реализована экспериментальная версия INGRES [87], которая поддерживала язык QUEL. В 1980 году часть сотрудников этой лаборатории организовали фирму Relational Technology, которая в 1981 году выпустила промышленную СУБД INGRES. В 1986 году INGRES была переведена на SQL. Ряд ключевых идей, заложенных в INGRES, до сих пор широко используются в реляционных системах, например, в NonStop SQL, Sybase и Microsoft SQL Server.

Postgres. После основания Relational Technology Стоунбрейкер вместе с Лоуренсом А. Роу (Lawrence A. Rowe) приступили к исследованиям по устранению ограничений реляционной модели. Новый проект получил название Postgres (POST inGRES). Были разработаны концептуальные проектные решения [88], предложена объектно-реляционная модель со сложными типами данных [89], разработаны структура хранения данных [90] и система правил [91] (триггеров), которая позволяет определять дополнительные действия, инициируемые при выполнении операций вставки, обновления или удаления в таблицах базы данных. Изначально языком запросов Postgres был PostQUEL. Язык был разработан в 1985 году в Калифорнийском университете в Беркли под руководством Майкла Стоунбрейкера. PostQUEL основывался на языке запросов QUEL. В 1987 г. была реализована первая версия СУБД Postgres, которая на протяжении последующих несколько лет усовершенствовалась [92]. Postgres стала широко использоваться экономике, промышленности, медицине, финансовом деле, астрономии и во многих других областях. Также использовалась в учебном процессе. В 1994 году был добавлен интерпретатор языка SQL, а в 1996 г. программный продукт был переименован на PostgreSQL.


Лоуренс Роу

В 2014 г. Майкл Стоунбрейкер стал лауреатом премии Тьюринга за фундаментальный вклад в концепции и методы, лежащие в основе современных систем баз данных [93].

СУБД для ПК. До 80-х г. исследования, экспериментальные и промышленные разработки СУБД велись для больших и средних компьютеров. В начале 80-х гг. появились IBM PC и совместимые с ними ПК, оснащенные ОС MS-DOS, что привело к появлению СУБД для ПК. В 1981 г. компания Ashton-Tate выпустила dBase II для ПК. Ее нельзя было назвать настоящей СУБД, так как многие важные функции не поддерживались, но для ПК того времени это было большим событием. dBase II получила большую популярность. В 1984 г. была выпущена более совершенная версия dBase III, в 1986 - ее расширенный вариант dBase III+, а в 1998 - dBase IV. Они стали доминирующими СУБД для IBM PC. Успех dBase III+ предопределил появление на рынке многочисленных аналогов, которые были совместимы по языку и структуре файлов базы данных. К ним относятся FoxBASE (1984), FoxPro (1990) компании Fox Software, Clipper (1985) компании Nantucket Corporation, Со временем они были объединены прижившимся среди профессионалов понятием «xBase». Тенденция создания продуктов-аналогов и большая популярность xBase активизировала деятельность по созданию стандарта. Были сделаны две попытки стандартизации языка xBase в 1987-1988 и 1992 гг., но они завершились безрезультатно. Итак, в 80-х годах доминирующая роль на рынке СУБД для IBM PC была за семейством СУБД xBase.

В 1985 г. компания Ansa Software выпустила СУБД Paradox. Этот высокопроизводительный продукт для создания реляционных баз данных стал примечательным своим языком QBE (Query By Example) и языком разработки приложений. Он был популярен в конце 80-х - начале 90-х годов и конкурировал с семейством xBase,

Оптимизация

Реляционные системы базируются на высокоуровневом непроцедурном интерфейсе, их языки запросов являются декларативными. В связи с этим в таких системах принципиально важным является вопрос оптимизации выполнения запросов. В 70-80

годы прошлого столетия были проведены многочисленные исследования и опубликовано громадное количество статей по этому вопросу. Отметим некоторые из них. Одной из первых статей по оптимизации запросов в РМД была [94], описывающая методы оптимизации в System R, которая по сути положила начало исследований в этом направлении и результаты которой впоследующем были использованы во многих коммеческих СУБД. Оригинальный на конец 80-х г. подход для оптимизации запросов был предложен в системе Starburst [95, 96] использованием правил эквивалентных преобразование запросов с целью более эффективного их выполнения. Отметим также системы Volcano [97] и Cascades [98] , использующие нисходящую трансформационную оптимизацию с запоминанием. Хорошие обзоры представлены в статьях [99-101]. Также рекомендуем прекрасный обзор С.Д. Кузнецова [102].

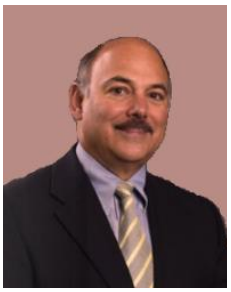
Стандартизация

В мае 1979 была создана рабочая группа по реляционным базам данных (RTG) ANSI/X3/SPARC DBS-SG под руководством Майкла Броди (Michael L. Brodie)

Майкл Броди

для проведения исследований по обоснованию возможности создания стандарта по реляционным базам данных. В 1982 году эта группа выпустила отчет [103], в котором подтверждалась такая необходимость. Для последующего содействия в работе по созданию такого стандарта был разработан «Каталог функций реляционных концепций, языков и систем», который должен был помочь выявлению и установлению тех аспектов, как самой реляционной модели, так и реляционных баз данных, которые могут рассматриваться кандидатами для стандартизации.

К началу 80-х годов в связи с широким распространением реляционных СУБД появилась необходимость анализа возможной стандартизации языка для управления реляционными базами данных и разработки такого стандарта, если это будет признано целесообразным. В связи с этим в 1982 году Американский национальный институт стандартов (American National Standards Institute — ANSI) создал комитет X3H2, перед которым была поставлена эта задача. На протяжении 11 лет комитет возглавлял Дональд Р. Дойч (Donald R. Deutsch). Комитет принял к рассмотрению различные реляционные языки, которые были описаны и реализованы к тому времени. Однако, учитывая широкую распространенность SQL в промышленных СУБД и тот факт, что он фактически уже стал стандартом к тому времени, комитет остановился на этом языке. Взяв за основу его диалект, реализованный в СУБД DB2, комитет постарался его обобщить, учитывая реализованные в других реляционных СУБД возможности. После четырех лет работы, в 1986 году предложенный комитетом вариант SQL был официально утвержден как стандарт ANSI, а в 1987 году он был принят в качестве стандарта Международной организацией стандартов (International Standards Organization — ISO). Затем стандарт ANSI/ISO приняло правительство США как федеральный стандарт в области обработки информации (Federal Information Processing Standard — FIPS). В 1989 году стандарт был незначительно изменен и получил название SQL-89 (или SQL1).



Дональд Дойч

С этого момента SQL был принят единственным языком внешних интерфейсов реляционных баз данных. ANSI/ISO ведет постоянную работу по его усовершенствованию и выпуску новых версий. За 35 лет было выпущено 10 версий SQL (1986, 1989, 1992, 1999, 2003, 2006, 2008, 2011, 2016, 2019).

В заключение данного раздела отметим, что к началу 80-х годов на рынке появились две промышленные реляционные СУБД: Oracle и DB2. Затем появились Postgress, Informix и другие. Началась эра реляционных СУБД, которые до сих пор являются наиболее популярными на рынке баз данных.

Литература

1) Codd E.F. “A Relational Model of Data for Large Shared Data Banks,” Communications of the ACM, Vol. 13, No. 6 (June 1970), pp. 377-397
2) Codd E.F. “A data base sublanguage founded on the relational calculus,” Proc. 1971 ACM-SIGFIDET Workshop on Data Description, Access, and Control, Nov. 1971. ACM. New York, 1971, DP. 35-68
3) Codd E. F. Relational Completeness of Data Base Sublanguages. In Courant Computer Science Symposium 6 on Data Base Systems. R. Rustin, Ed., 1971, pp. 65-98.
4) Palermo F.P. “A data base search problem”, Proceedings 4th Computer and Information Science Symposium (COINS IV), Miami Beach, Dec. 1972, Plenum Press, New York, 1972. pp. 67–101
5) Codd E.F. “Interactive Support for Nonprogrammers: The Relational and Network Approaches,”Proceedings of the ACM SIGMOD Workshop on Data Description, Access, and Control, Vol. II, Ann Arbor, Michigan, May 1974.
6) Codd E.F. Extending the database relational model to capture more meaning. ACM Trans. on Database Syst., vol. 4, No. 4, 1979, pp. 397-434
7) Codd E.F. Data models in database management. In Proceedings of the 1980 Workshop on Data abstraction, Databases, and Conceptual Modeling. ACM Press, 1980, pp. 112–114.
8) Codd E. F. "The Second and Third Normal Forms for the Relational Model", IBM technical memo (October 6th. 1970).
9) Codd E.F. "Further Normalization of the Database Relational Model", in Data Base Systems, Courant Inst. Comput.Sci. Symp. Series 6 (New York, 1971), Englewood Cliffs, N.J.: Prentice Hall, 1972, pp. 33-64.
10) Codd E.F. “Normalized Data Base Structure: A Brief Tutorial", Proc. 1971 ACM SIGFIDET Workshop on Data Description, Access. and Control. San Diego. Calif. 1971, p. 1-17
11) Date C.J. The database relational model : a retrospective review and analysis. - Addison-Wesley Educational Publishers Inc., 2000, 152 p.
12) Codd E.F. "Recent Investigations in Relational Database Systems," Information Processing 74, pp.1017–1021.
13) Heath I.J. Unacceptable File Operations in a Relational Data Base. Conference: Proceedings of 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control, San Diego, California, November 11-12, 1971, pp. 19–33
14) Armstrong William Ward. "Dependency structures of data base relationships". In Jack L. Rosenfeld and Herbert Freeman, editors, Proceedings of IFIP Congress 74, pp. 580-583, North Holland, 1974
15) Fagin R. Multivalued Dependencies and a New Normal Form for Relational Databases / R. Fagin // ACM Transactions on Database Systems. – 1977. – Vol. 2, № 1. – P. 262-278.
16) Beeri C., Fagin R., Howard J.H. A complete axiomatization for functional and multivalued dependencies in database relations. Proc. ACM SIGMOD Conf., D.C.P. Smith, Ed., Toronto, Canada, August 1977, pp. 47-61.
17) Zaniolo C. Analysis and design of relational schemata for database systems. Ph.D. Diss., Tech. Rep. UCLA-ENG-7669, U. of California, Los Angeles, Calif., July 1976.
18) Delobel C., Leonard M. The decomposition process in a relational model. Proc. Int. Workshop on Data Structure Models for Information Systems, Presses U. de Namur, Namur, Belgium, May 1974, pp. 57-80.
19) Rissanen J. Theory of relations for databases - a tutorial survey, in "Proc. 7th Sympos. on Math. Found. of Computer Science," 1978, pp. 537-551, Lecture Notes in Computer Science No. 64, Springer-Verlag, Berlin
20) Fagin R. Normal Forms and Relational Database Operators / R. Fagin // Proceedings of the ACM SIGMOD International Conference on Management of Data (Boston, Mass., May 30-June 1), ACM, New York, 1979, p. 153-160

21) Date Chris J. "On DK/NF normal form". - https://web.archive.org/web/20120406123712/http://www.dbdebunk.com/page/page/621935.htm
22) Buy B., Puzikova A. V. Some nonclassical normal forms in relational databases (Rus) // Bulletin of Taras Shevchenko National University of Kyiv. Series Physics & Mathematics, 2015, No 1, pp. 65-74
23) Ling T. W. An Improved Third Normal Form for Relational Databases / T. W. Ling, F. W. Tompa, T. Kameda // ACM Transactions on Database Systems. – 1981. – Vol. 6, № 2. – P. 329-346.
24) Zaniolo C. A New Normal Form for the Design of Relational Database Schemata / C. Zaniolo // ACM Transactions on Database Systems. – 1982. – Vol. 7, № 3. – P. 489-499
25) Normann R. Minimal lossless decompositions and some normal forms between 4NF and PJ/NF / R. Normann // Information Systems. – 1998. – Vol. 23, № 7. – P. 509-516.
26) Vincent M. W. A corrected 5NF definition for relational database design / M. W. Vincent // Theoretical Computer Science (TCS). – 1997. – Vol. 185, № 2. – P. 379-391.
27) Vincent M.W. Redundancy Elimination and a New Normal Form for Relational Database Design / M. W. Vincent // In Semantics in Databases (Libkin, L., Thalheim, B., eds.), vol. 1358 of LNCS. – 1998. – P. 247-264.
28) Darwen H. A Normal Form for Preventing Redundant Tuples in Relational Databases / H. Darwen, C. Date, R. Fagin // Proceedings of the 15th International Conference on Database Theory – ICDT'2012, March 26– 30, 2012, Berlin, Germany. – P. 114-126.
29) Fagin R. A Normal Form for Relational Databases That Is Based on Domains and Keys / R. Fagin // Communications of the ACM. – 1981. – Vol. 6. – P. 387-415.
30) Delobel C. Normalization and hierarchical dependencies in the relational data model. ACM TODS, 1978, Vol. 3, No. 3, 201-222.
31) Pasichnik V.V., Stogniy A. A. Relational models of data bases (Rus). - M.: CNIIATOMINFORM, 1983, 268 p.
32) Casanova M.A. Inclusion dependencies and their interaction with functional dependencies / M. A. Casanova, R. Fagin, C. H. Papadimitriou // Journal of Computer and System Sciences. – 1984. – № 28. – P. 29-59.
33) Nicolas J.M. Mutual dependencies and same results on indecomposable relations / J. M. Nicolas // Proceedings of the fourth international conference on Very Large Data Bases, 1978. – Vol. 4. – P. 360-367.
34) Ling T.W. Logical Database Design with Inclusion Dependencies / T. W. Ling, C. H. Goh // In Proceedings of the Eighth International Conference on Data Engineering, Tempe, Arizona, 1992. – P. 642-649.
35) Levene M. Justification for Inclusion Dependency Normal Form / M. Levene, M. W. Vincent // IEEE Transactions on Knowledge and Data Engineering, 2000, Vol. 12, No. 2, pp 281-291.
36) Thalheim B. Bibliographie zur Theorie der Abhangigkeiten in relationalen Datenbanken, 1970-1984, TU Dresden 566/85, Dresden 1985.
37) Thalheim B. Dependencies in Relational Databases, 1991, Teubner-Texte zur Mathematik, 214 p.
38) Chamberlin D.D. "Relational Data-Base Management Systems," Computing Surveys, Vol. 8, No. 1, p. 43-66, March 1976
39) Goldstein R.C., Strnad A.L. The MACAIMS Data Management System. Proceedings of the ACM-SIGFIDST Workshop on Data Description, Acess and Control, Nov. 1970. ACM, New York, 1970, pp. 201-229.
40) Notley M.G. The Peterlee IS/1 system. IBM UK Scientific Centre Report UKSC-0018, March 1972.
41) Todd S.J.P. Peterlee relational test vehicle PRTV, a technical overview. IBM Scientific Centre Report UKSC 0075, Peterlee, England, July 1975.
42) Whitney V.K.M. "RDMS: A Relational Data Management System," Proceedings of the Fourth International Svmposium

on Computer and Information Sciences (COINS IV), Dec. 1972, Plenum Press, New York, 1972.
43) Pecherer R.M. Efficient evaluation of expressions in a relational algebra. Proc. ACM Pacific 76 Regional Conf., April 1975, ACM, New York, 1975, pp. 44-49.
44) Gotlieb L.R. Computing joins of relations. Proc. ACM-SIGMOD International Conference on Management of Data (San Jose, Calif., May 14-16, 1975), ACM. New York, 1975, pp. 55-63
45) Smith J.M., Chang P. Optimizing the performance of a relational algebra data base interface. Comm. ACM, 1975, Vol. 18, No. 10, pp. 568–579.
46) Hall P. A. V. Optimisation of a single relational expression in a relational data base system, IBM Scientific Centre Renort UKSC 0076. Peterlee, England, July 1975.
47) Palermo F.P. An APL environment for testing relational operators and data base search algorithms. Proc. APL 75 Conf., June 1975, ACM. New York, 1975, pp. 249-256
48) Bui D.B., Skobelev V.G. Complexity of operations in database systems (a survey), Radioelectronic and computer systems, 2014, No 6(70). pp. 53-59
49) Held G.D., Stonebraker M.R., Wong E. “INGRES: a relational data base system,” Proc. AFZPS h’ational Computer Conf., May 1975, Vol. 44, AFIPS Press, Montvale, N.J., 1975, pp. 409-416.
50) Bracchi G., Fedeli A., Paolini P. A language for a relational data base management system. Proc. Sixth Annual Princeton Conf. on Information Science and Systems, March 1972, Princeton Univ., N.J., 1972. pp. 84-92.
51) Fehder P.L. The representation-independent language. Res. Rep. RJ 1121, IBM Research Laboratory, San Jose, Calif., Nov. 1972
52) McDonald N.,Stonebraker M. "CUPID — The Friendly Query Language," University of California, Berkeley, Technical Report No. UCB/ERL M487, October 1974.
http://www2.eecs.berkeley.edu/Pubs/TechRpts/1974/ERL-m-487.pdf
53) McDonald N.,Stonebraker M. Cupid — The friendly query language. Proc. ACM Pacific-75, San Francisco, Calif., April 1975, pp. 127-131.
54) McDonald N. Cupid: A Graphics Oriented Facility for Support of Non-Programmer Interactions with a Data Base. University of California, Berkeley, Technical Report No. UCB/ERL M563, November 1975.
55) McDonald N., Stonebraker M. CUPID: the friendly query language. Proc. ACM Pacific 75 Regional Conf., Auril 1975. ACM. New York. 1975, pp, 127-131.
56) Zloof M.M. Query by example. RC4917, IBM T. J. Watson Research Center. York-town Heights, N. Y., July 1974.
57) Zloof M.M. Query by Example. Proc. AFIPS National Computer Conf., May 1975, Vol. 44, AFIPS Press, Montvale, N.J., 1975, pp 431-438.
58) Zloof M.M. Query by Example: the invocation and definition of tables and forms. Proc. Internatl. Conf. on Very Large Data Bases, Sept. 1975, ACM, New York, 1975, pp. 1-24.
59) Zloof M.M. Query-by-Example: a data base language. IBM System J., 1977, Vol. 16, No. 4, pp. 324–343
60) Thomas J. C., Gould J.D. A psychological study of Query by Example. Proc. AFIPS National Computer Conf., May 1975, Vol. 44, AFIPS Press, Montvale, N.J., p 439-445.
61) Zloof M.M. Office-by-Example: A business language that unifies data and word processing and electronic mail. IBM Systems Journal, 1982, Vol. 21, No. 3, pp. 272 - 304
62) Boyce R.F., Chamberlin D.D., King W.F., Hammer M.M. Specifying queries as relational expressions. Proc. ACM SIGPLAN/SIGIR Interface Meeting, Gaithersburg, Md., Nov. 1973.
63) Boyce R.F., Chamberlin D.D., King W.F., Hammer M.M. Specifying queries as relational expressions: the SQUARE data sublanguage. Communications of the ACM, 1975, Vol. 18, No. 11, pp. 621-628
64) Boyce, R.F., Chamberlin D.D. Using a structured English query language as a

data definition facility. Res. Report RJ 1318, IBM Res. Lab., San Jose, Calif., Dec. 1973.

65) Chamberlin D D., Boyce R.F. SEQUEL: A structured English query language. SIGFIDET '74: Proceedings of the 1974 ACM SIGFIDET (now SIGMOD. workshop on Data description, access and control. May 1974, pp, 249–264.

66) Astrahan M.M., Chamberlin D.D. Implementation of a structured English query language. Communications of the ACM, 1975, Vol. 18, No. 10, pp. 580-588

67) Lorie R.A. XRM-an extended (n-ary) relational memory. Tech. Report G320-2096, IBM Scientific Center, Cambridge, Mass., Jan. 1974.

68) Astrahan M.M., Lorie R.A. SEQUEL-XRM: a relational system. Proc. ACM Pacific 76 Regional Conf., April 1975, ACM, New York, 1975, pp. 34-38.

69) Symonds A.J., Lorie, R. A. A schema for describing a relational data base. Proc. ACM-SIGFIDET Workshop on Data Description and Access, Nov. 1970, ACM, New York, 1970, pp. 230-245.

70) Lorie R.A., Symonds, A.J. A relational access method for interactive applications. Courant Computer Science Symposia, 6, Data Base Systems, Prentice-Hall, New York, 1971, pp. 99-124.

71) Chamberlin D D., Astrahan M.M., Eswaran K.P., Griffiths P.P., Lorie R.A., Mehl J.W., Reisner Ph., Wade B.W. SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control. IBM Journal of Research and Development. 1976, Vol. 20, No. 6, pp. 560-575

72) Carlson C.R., Kaplan R.S. A Generalized Access Path Model and Its Application to a Relational Data Base System. SIGMOD '76: Proceedings of the 1976 ACM SIGMOD international conference on Management of data. June 1976, pp. 143–154

73) Notley M, Peterlee IS/1 System. UKSC Report 18, 1972

74) Todd S. The Peterlee Relational Test Vehicle - A System Overview. IBM Systems Journal. 1976, Vol. 15, No. 4, pp. 285–308.

75) Astrahan M.M., et al. System R: A relational approach to database management. ACM Trans. Database Syst. 1976, Vol. 1, No. 2, pp. 97-137

76) Chamberlin D.D. A summary of user experience with the SQL data sublanguage. Proc. Internat. Conf. Data Bases, Aberdeen, Scotland, July 1980, pp. 181-203

77) Chamberlin D.D., et al. Support for repetitive transactions and adhoc queries in System R. ACM Trans. Database Syst. 1981. Vol. 6, No 1, pp. 70-94.

78) Chamberlin D.D., Gilbert, A.M., Yost, R.A. A history of System R and SQL/data system. VLDB '81: Proceedings of the seventh international conference on Very Large Data Bases. Vol. 7, September 1981, pp. 456–464

79) McDonald N., Stonebraker M., Wong E. Preliminary design of INGRES: Part I. Electronics Research Lab. Report ERL-M435, Univ. of California, Berkeley, April 1974.

80) McDonald N., Stonebraker M., Wong E. Preliminary design of INGRES: Part II. Electronics Research Lab. Report ERL-M436, Univ. of California, Berkeley, April 1974.

81) Held G., Stonebraker M. Storage structures and access methods in the raational data base management system INGRES. Proc. ACM Pacific 75 Regional Conf., April 1975, ACM, New York, 1975, pp. 26-33.

82) Wong E., Youssefi K. Decomposition - A strategy for query processing. ACM Trans. on Database Systems I, 3 (Sept. 1976), pp. 223-241

83) Stonebraker M. Implementation of integrity constraints and views by query modification. Proc. ACM-SIGMOD Conf. May 1975, ACM, New York, 1975, pp. 65-78.

84) Stonebraker M., Wong E. Access control in a relational data base management system by query modification. Proc. 1974 ACM Nat. Conf., San Diego, Calif., Nov. 1974, pp. 180-187.

85) Stonebraker M. High level integrity assurance in relational data base management systems. Electronics Research Lab.

Report ERL-M473, Univ. of Calif. at Berkeley, August 1974.
86) Stonebraker M., Rubinstein P. The INGRES protection system. Proc. 1976 ACM National Conf., Houston, Tex., Oct. 1976
87) Stonebraker M., Held G., Wong E., Kreps P. The Design and Implementation of INGRES. ACM Transactions on Database Systems. 1976, Vol.1, No 3, pp. 189–222.
88) Stonebraker M., Rowe L. The design of POSTGRES. Proc. 1986 ACM-SIGMOD Conf., Washington, DC, June 1986.
89) 1Rowe L.A., Stonebraker M. The POSTGRES data model. In Proc. 13th Intl. Conf. on Very Large Data Bases, P. M. Stocker, W. Kent, P. Hammersley, Eds., San Francisco, CA: Morgan Kaufmann Publishers Inc., 1987, pp. 83-96.
90) Stonebraker M. The design of the POSTGRES storage system. In Proc. 1987 VLDB Conf:, Brighton, England, Sept. 1987.
91) Stonebraker M., Hanson E., Hong C. H. The design of the POSTGRES rules system. Proc. IEEE Conference on Data Engineering, Feb. 1987.
92) Stonebraker M., Rowe L.A., Hirohama M. The Implementation Of Postgres IEEE Transactions on Knowledge and Data Engineering, 1990, Vol. 2, No. 1, pp. 125-142
93) ACM Turing Award Goes to Pioneer in Database Systems Architecture: MIT's Michael Stonebraker Brought Relational Database Systems from Concept to Commercial Success. - https://www.prweb.com/pdfdownload/12607207.pdf
94) Selinger P.G., Astrahan M.M., Chamberlin D.D., Lorie R.A., Price T.G. Access path selection in a relational database management system. SIGMOD '79: Proceedings of the 1979 ACM SIGMOD international conference on Management of data. 1979, pp. 23–34
95) Haas L.M., Freytag J.C., Lohman G.M., Pirahesh H. Extensible query processing in Starburst. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. 1989, pp. 377–388.
96) Pirahesh H., Hellerstein J.M., Hasan W. Extensible/rule based query rewrite optimization in Starburst. SIGMOD '92: Proceedings of the 1992 ACM SIGMOD international conference on Management of data, 1992, pp. 39–48
97) Graefe G., McKenna W.J. The Volcano optimizer generator: extensibility and efficient search. In: Proceedings of the 9th International Conference on Data Engineering. 1993, p. 209–218.
98) Graefe G. The Cascades Framework for Query Optimization. IEEE Bulletin of the Technical Committee on Data Engineering,1995, Vol. 18, No. 3, pp. 19–29.
99) Jarke M., Koch J. Query optimization in database systems. ACM Comput Surv. 1984, Vol. 16, No. 2, pp.111–152
100) Chaudhuri S. An overview of query optimization in relational systems. In: Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems; 1998. pp. 34–43.
101) Neumann Th. Query Optimization (in Relational Databases). In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 3009-3015. Springer, New York, 2018
102) Kuznetsov S.D. Methods for optimization of query execution in relational DBMS (Rus) // "Vychislitelnye nauki. Vol. 1 (Itogi nauki i tekhniki VINITI AN USSR" M.; VINITI AN USSR, 1989, pp. 76-153. - http://masters.donntu.org/2002/foreign/aswad/lib/mpbd
103) Brodie M.L., Schmidt J.W. Final Report of the ANSI/X3/SPARC DBS-SG Relational Database Task Group. SIGMOD Record, 1982, Vol. 12, No. 4, pp. 1-62

Управление транзакциями

Важной функцией функционирования БД является управление транзакциями.

Транзакция (transaction) — это логическая единица работы, представляющая собой группу последовательных операций над данными базы данных, которая может быть выполнена либо целиком и успешно, соблюдая целостность данных и независимо от других параллельно работающих транзакций, либо не выполнена вообще, и тогда она не должна произвести никакого эффекта. Понятие транзакции впервые было введено и обсуждено Джимом Греем (Jim Gray) и его коллегами в работах [1, 2, 3].

Правила ACID. Одним из наиболее распространённых наборов требований к транзакциям и транзакционным системам является набор ACID (Atomicity, Consistency, Isolation, Durability).

- *Atomicity* - *атомарность*. Транзакция либо выполняется полностью, либо не выполняется вообще. С точки зрения внешнего восприятия, она не имеет никаких промежуточных состояний.
- *Consistency* - *согласованность*. Транзакция сохраняет ограничения целостности базы данных. По завершению работы транзакция оставляет базу данных в целостностном состоянии.
- *Isolation* - *изолированность*. Транзакция работает так, как если бы не было никаких одновременно работающих транзакций.
- *Durability – долговременность*. По завершению работы все произведенные транзакцией изменения сохраняются в базе данных на долговременной основе.

Требования ACID были в основном сформулированы в начале 1980-х годов Джимом Греем [1]. Вместе с тем существуют специализированные системы с ослабленными транзакционными свойствами [4].

Изолированность транзакции - ситуация, в которой транзакция защищена (заизолирована) от действий других выполняющихся одновременно с ней транзакций. Другими словами, изоляция гарантирует, что промежуточные состояния транзакции являются невидимыми другим одновременно работающим транзакциям. Степень изолированности транзакции определяется уровнями изоляции. Чтобы добиться изолированности транзакций, следует использовать методы управления совместным выполнением транзакций. План выполнения набора транзакций называется сериальным, если результат совместного выполнения транзакций эквивалентен результату некоторого последовательного выполнения этих же транзакций.

Сериализация. *Сериализация транзакций* - это механизм такого совместного выполнения транзакций, при котором результат эквивалентен результату некоторого последовательного выполнения этих же транзакций.. Обеспечение такого механизма является основной функцией управления транзакциями. Система, в которой поддерживается сериализация транзакций, обеспечивает реальную изолированность пользователей.

Концепция сериализуемости была сформулирована и исследована Греем и его коллегами в работах [2 3]. Кроме того, в работе [5] был определен двухфазный протокол блокирования, и исследована техника предикатного блокирования. Вопросы области действия блокирования (гранулярности) обсуждены в статьях [2, 6].

Модели транзакций. Следует отметить, что все модели транзакций определялись, как правило, с учетом классов прикладных систем, в которых они находят применение [7]. Были предложены две фундаментальные модели транзакции - модель страницы и модель объекта. Первая из них - исполнительная модель, а вторая - концептуальная.

Модель страниц (модель Read/ Write). Основывается на предположении, что основные операции базы данных - это запись и чтение на страницы, которые передаются между внешней и оперативной памятью. Страничная модель транзакции берет свое начало во второй половине 70-х годов со статей Джима Грея [1, 8] и Капали Эсваран (Kapali Eswaran) [158]. В это же время возникло родственное понятие - атомарное действие [9, 10]. Концепция страничной модели транзакции стала предметом интенсивных теоретических исследований в 80-х

годах [11–14] и является действенной по настоящее время, хотя приобрела ряд расширений и вариаций [15]. С обзором исследований и разработок этой модели можно познакомиться в работах [7, 16]

Все приводимые далее модели относятся к классу так называемых моделей объектов.

Плоские транзакции (flat transaction) обладают единственным уровнем управления для произвольного количества элементарных действий. Они не обладают внутренней структурой. Плоские транзакции - основные строительные блоки для реализации принципа атомарности. В плоских транзакциях атомарность и долговременность поддерживается механизмом восстановления, который обычно обеспечивается ведением журналов операций обновления, в связи с чем операции типа "отменить", "повторить" можно выполнять по мере необходимости. Изолированность обеспечивается механизмом управления параллелизмом (concurrency control), который реализуется с помощью блокировок. Обзор исследований по управлению параллелизмом приведен в работе [17]. Согласованность обеспечивается механизмом управления целостностью. Было предложено два подхода по управлению целостности в транзакциях: включение этого механизма в СУБД [18] и поддержание целостности за счет усилий разработчиков приложений [7].

Плоские транзакции соблюдают в полной мере все принципы ACID и являются вполне достаточными для многих традиционных приложения баз данных, в которых время выполнения транзакции относительно непродолжительное, количество параллельных транзакций достаточно небольшое и база данных не является распределенной. Однако такие ACID-транзакции не в состоянии поддерживать долговременные транзакции и транзакции со сложной внутренней структурой и распределенными базами данных.

Точки сохранения. Это такие моменты в вычислительном процессе, начиная с которых возможен перезапуск вычислений при возникновении каких-либо проблем. Они впервые были определены в 1976 г. в System R [19]. При возникновении сбоя происходит откат к последней сохраненной точке с освобождением всех сделанных после этой точки блокировок. Хотя механизм точек сохранения широко используется в плоских транзакциях, однако он приобрел новое звучание в расширенных моделях транзакций, которые появились в 80-х годах.

Все приводимые далее расширенные модели транзакций приводят к ослаблению тех или иных составляющих ACID.

Модель многозвенных транзакций (chained transactions) подобна модели плоской транзакции с точками сохранения, но она предоставляет не только возможность пометить какую-либо точку для возможного повторного выполнения, но и фиксацию той части работы, которая была выполнена в момент достижения этой точки, причем откат может быть выполнен только к последней контрольной точке. В этом подходе была заложена идея декомпозиции больших транзакций на более мелкие последовательно выполняемые субтранзакции, которые соответствуют интервалам между точками. При сбое текущей субтранзакции предыдущая транзакция уже была зафиксирована и ее результаты были сохранены в базе данных, поэтому откат производится к этой точке сохранения. Отметим, в этой модели атомарность и изолированность не гарантируется для всей транзакции. Согласно [7] идея многозвенной транзакции впервые реализована в системе IMS компании IBM.

Вложенные транзакции (Nested Transactions). Важным шагом в развитии базовой модели транзакции явилось расширение плоской (одноуровневой) модели в многоуровневую структуру. Вложенная транзакция впервые была определена в 1981 г. Моссом (Moss) [20], а затем в [21]. Ее концепция базировалась на понятии сфер контроля (spheres of control) [22]. Вложенная транзакция - это множество субтранзакций, которые могут содержать другие субтранзакции, образуя таким образом транзакционное дерево. Дочерняя транзакция запускается после родительской, а родительская транзакция заканчивается только после завершения работы всех ее дочерних транзакций. При аварийном завершении родительской транзакции все ее дочерние транзакции также завершаются аварийно. При аварий-

ном завершении дочерней транзакции ее родитель может выбрать альтернативный вариант (contingency subtransaction - транзакция на непредвиденный случай). Вложенные транзакции обеспечивают полную изоляцию на глобальном уровне. Для вложенных транзакций ослаблено свойство долговременности ACID.

Модель вложенных транзакций хорошо подходит для активных баз данных, поскольку иерархическая структура модели позволяет легко согласовывать связь между основной транзакцией и той, которая запускается с помощью триггера. В статье [23] предложены следующие варианты синхронизации запуска субтранзакции триггера и по отношению к основной транзакции:

– *немедленно (immediate)* - субтранзакция запускается разу же после наступления события триггера;
– *откладывается (deffered)* - запуск субтранзакции откладывается до завершения основной транзакции;
– *причинно-независимая (causally independent)* - субтранзакция триггера запускается как полностью самостоятельная транзакция;
– *причинно-зависимая (causally dependent)* - субтранзакция триггера запускается как самостоятельная транзакция, но ее успешное завершение зависит от успешного завершения основной транзакции.

Открытые вложенные транзакции (Open Nested Transactions) [24] ослабляют требование изолированности в связи с тем, что результаты зафиксированных субтранзакций становятся видимыми другим одновременно работающим вложенным транзакциям. При этом достигается высокий уровень параллельности.

Многоуровневые транзакции представляют собой наиболее общий вариант вложенных транзакций [24, 5] Субтранзакции многоуровневой транзакции могут произвести фиксацию и освобождение свих ресурсов до завершения работы глобальной транзакции Если глобальная транзакция завершается аварийно, то для поддержания атомарности следует произвести откат субтранзакция запуском их компенсирующих субтранзакций. Однако все же возможно нарушение целостности в связи с тем, что некоторая другая транзакция имела доступ к результатам завершенных субтранзакций, которые затем были откатаны компенсирующими субтранзакциями. Были предложены решения этой ситуации, например, введением горизонтальных компенсаторов [26]. В монографии [27] приводятся отличительные признаки многоуровневых и вложенных транзакций.

Распределенные транзакции. (Distributed transactions). Представляют собой совокупность субтранзакций, которые привязываются к локальным базам данных, и общей глобальной транзакции. В статье [28] приводится обзор распределенных транзакций, а также рассмотрена "модель базисной транзакции" (base transaction model) и ее расширения. В 1996 г. была предложена модель X/Open Distributed Transaction Processing (X/Open DTP) [29]. Эта модель является стандартом для протокола двухфазной фиксации (2PC - Two Phase Commit).

Гибкие транзакции (Flexible Transactions) [30, 31] - были предложены для среды распределенных баз данных. В этом случае глобальная транзакция представляет собой набор субтранзакций, каждая из которых осуществляет доступ к данных на одном локальном узле. Модель гибкой транзакции поддерживает гибкое управление вычислениями путем спецификации зависимостей двух типов между субтранзакциями: 1) зависимости порядка вычислений между двумя субтранзакциями, 2) зависимости альтернатив между двумя поднаборами субтранзакций. Было разработано несколько конкретных моделей гибких транзакций: ConTracts, FlexTransactions, SplitTransactions S-transactions и другие [30–34]. Был предложен язык IPL [35] для спецификации гибких транзакций с определяемой пользователем атомарностью и изолированностью

Длительные транзакции, компенсаторы и модель Saga. Идея компенсирующих транзакций (compensation transaction) впервые была высказана Греем в работе [8], затем она была формализована в [36, 37] и, наконец, использована в модели Saga [38] Эта модель имеет отношение к длительным транзакциям (Long-Running Transactions)

[39]. Модели распределенных транзакций хорошо справляются с кратковременными транзакциями, однако являются неприемлемыми для длительных транзакций. В модели Saga предлагается разбивать длительные транзакции на более короткие. Saga состоит из совокупности упорядоченных ACID субтранзакций и совокупности компенсирующих субтранзакций, по одной на каждую из основных субтранзакций. Координация всего процесса осуществляется с помощью сообщений и временных отметок. Saga завершается успешно, если успешно зафиксированы все субтранзакции. Если какая-то из субтранзакций завершается аварийно, то все предварительно завершенные субтранзакции откатываются выполнением, так называемых компенсирующих субтранзакций. Saga ослабляет требование к изолированности и увеличивает межтранзакционный параллелизм. В работе [40] предложена усовершенствованная модель - вложенная Saga, которая позволяет представлять линейную структуру долговременных транзакций в виде иерархической транзакционной структуры.

Транзакции Split/Join (разделить/ соединить). Концепция разделения/соединения транзакций была впервые описана в [41] и затем тщательно проработана в [42] для таких долгосрочных видов деятельности, как автоматизированное проектирование, инженерное проектирование, проектирование и разработка программного обеспечения и др. Операция Split разделяет транзакцию на две сериализуемые транзакции, которые фиксируются или завершаются аварийно независимо друг от друга. Операция Join объединяет две транзакции в одну. Split используется, например, чтобы пораньше зафиксировать результаты работы части транзакции, или чтобы распределить работу среди нескольких исполнителей. В свою очередь Join равносилен передаче всей работы одному исполнителю [42]. Впоследствии операции Split и Join были включены во вложенные транзакции для создания комбинированных моделей транзакций [43]

Кооперативные транзакции были предложены в [44] для использования в системах, в которых ярко выражена потребность во взаимодействии между транзакциями, в кооперативной интерактивной рабочей среде. Фундаментальная проблема, связанная с кооперативными транзакциями, - это отсутствие для них четких критериев согласованности. Для кооперативных транзакций была предложена их структуризация в виде дерева, называемого иерархией кооперативных транзакций (Cooperative Transaction Hierarchy). В частном случае иерархия ограничена тремя уровнями: корень, одна или более транзакционных групп и несколько кооперативных транзакций. Кооперативные транзакции образуют листья иерархии, которые объединяются в группы. Члены группы транзакций работают вместе, выполняя некоторую логическую единицу работы, называемую задачей, которая может быть разбита на подзадачи. Каждая кооперативная транзакция ответственна за конкретную подзадачу. Поскольку требование атомарности ослаблено, кооперативная транзакция не обязана сохранять глобальную непротиворечивость базы данных, то есть изменения, сделанные.кооперативной транзакцией становятся сразу видимыми другими кооперативными транзакциями этой группы. Чтобы результаты были видны вне группы, используются точки сохранения. В иерархии может быть больше трех уровней, т.е. допускается несколько уровней вложенности групп. Кооперативные транзакции не обязательно должны быть сериализуемыми. Из-за своей интерактивной природы, кооперативные транзакции длятся значительно дольше обычных. Далее, в отличие от традиционных транзакций, кооперативные необязательно должны быть полностью изолированными.

ACTA и ее производные. ACTA [43, 45, 46] - это метамодель, которая облегчает спецификацию, анализ и синтез расширенных моделей транзакций. Формализм ACTA основывается на логике первого порядка с отношением предшествования, который позволяет разработчику транзакций специфицировать как высокоуровневые свойства (требования) модели, так и низкоуровневые аспекты поведения в терминах аксиом. Помимо поддержки спецификации и анализа существующих моделей транзакций, ACTA предоставляет возможность специфицировать требования новых транзакционных

приложений и синтезировать модели, удовлетворяющие этим требованиям. В работе [47] авторы предложили упрощенное средство разработки расширенных транзакций, названное ASSET. Оно основывается на транзакционных примитивах, позаимствованных у ACTA и может использоваться на уровне программирования для спецификации специализированных для конкретных приложений моделей транзакций, которые позволяют поддерживать кооперацию и взаимодействие.

В 90-е гг. уделялось большое внимание транзакциям в системах реального времени [48, 49, 50], и в мобильных системах баз данных [51, 52].

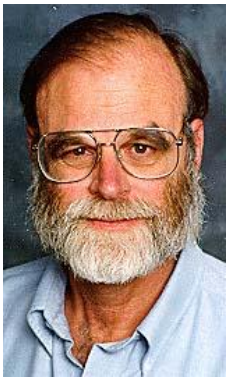
Транзакции веб-сервисов. Начиная с 2000-х годов все больше и больше уделяется внимание использованию транзакций для слабосвязываемых веб-сервисов с целью обеспечения согласованности и надежности веб-сервисных приложений. В настоящее время разработано три стандарта, имеющие отношение к транзакциям веб-сервисов.

Business Transaction Protocol (BTP) [53, 54]. Первая версия разработана в 2004 году в OASIS, Имеет отношение как к веб-сервисам, так и к произвольным бизнес-процессам. Это базирующийся на языке XML протокол для описания и управления сложными многошаговыми B2B-транзакциями (B2B - business-to-business) в Интернете. Он предоставляет возможность координировать транзакции между многими автономными сервисами, а использование XML делает его подходящим для веб-сервисных архитектур [55].

Web Services Transactions (WS-Tx). [56, 57] Спецификация, одобренная в 2007 г., состоит из WS-Coordination (WS-C), WS-AtomicTransaction (WS-AT), WS-BusinessActivity (WS-BA), и разработана в Microsoft, IBM и BEA. WS-Tx определяет механизмы транзакционной интероперабельности между веб-сервисами и обеспечивает внедрение качественных транзакционных сервисов в веб-сервисные приложения. WS-Tx определяет последовательность сообщений, передаваемых в между участвующими сторонами в (краткосрочных) атомарных транзакциях (WS-AT) и (длительных) бизнес-транзакциях (WS-BA). WS-C определяет координационные протоколы сообщений, которыми обмениваются стороны, участвующие в транзакции. WS-C поддерживает различные координационные модели [58].

WS Composite Application Framework (WS-CAF) [59]. Стандарт разработан в OASIS с участием компаний SUN, Oracle, Arijuna и др. Цель стандарта - разработка интероперабельных и простых в использовании составных веб-сервисных приложений.

Были проведены сравнительные исследования приведенных выше трех стандартов, с результатами которых можно познакомиться в статьях [60, 61].


Джеймс Грей

В 1998 г. Джеймс Николас Грей (James Nicholas Gray) был награжден премией Тьюринга за основополагающие идеи в области баз данных, исследования по обработке транзакций и техническое лидерство в реализации систем.

Литература

1) Gray J. The Transaction Concept: Virtues and Limitations. In: Proceedings of the 7th International Conference on Very Large Databases, 1981. pp. 144—154, IEEE, Cannes, France,
2) Gray J., Lorie R., Putzulo G. "Granularity of Locks and Degrees of Consistency in a Shared Data Base," In Modelling in Data Base Management Systems. G.M. Ni]ssen, (ed.) North Holland Publishing Company, 1976, pp.365-394
3) Eswaran K.P, Gray J, Lorie R.A, Traiger I.L. The notions of consistency and predicate locks in a database system. Commun ACM. 1976;19(11):624–633.
4) Advanced Transaction Models and Architectures. Sushil Jajodia and Larry Kerschberg (eds.) Springer Scienee+Business Media New York. 1997

5) Ullman J.D. Principles of Database and Knowledge-Based Systems.Maryland: Com-puter Sciences Press Inc., 1989
6) Reis D.R., Stonebraker M. Effect of locking granularity in a database management systems. ACM Trans. on Database Syst., 2:3, 1977, pp. 233-246.
7) Gray J., Reuter A. Transaction Processing: Concepts and Techniques. Morgan Kauf-mann, San Francisco. 1993
8) Gray J.N. Notes on data base operating systems. In: Bayer R., Graham R.M., Seegmüller G. (eds) Operating Systems. Lecture Notes in Computer Science, vol 60. Springer, Berlin, Heidelberg. 1978. p. 393–481.
9) Lampson B.W. Atomic transactions. In: Lampson B.W, Paul M, Siegert H.J, editors. Distributed systems – architecture and implementation: an advanced course, LNCS, vol. 105. Berlin: Springer; 1981. p. 246–285.
10) Lomet D.B. Process structuring, synchronization, recovery using atomic actions. ACM SIGPLAN Not. 1977; 12(3):128–137.
11) Bernstein P.A, Shipman D.W, Wong W.S. Formal aspects of serializability in database concurrency control. IEEE Transactions on Software Engineering. 1979, Vol. SE-5, No.3, pp. 203–216
12) Bernstein P.A, Hadzilacos V, Goodman N. Concurrency control and recovery in database systems. Reading: Addison-Wesley; 1987.
13) Papadimitriou C.H. The serializability of concurrent database updates. J ACM. 1979;26(4):631–653.
14) Papadimitriou C.H. The theory of database concurrency control. Rockville: Computer Science; 1986.
15) Weikum G, Vossen G. Transactional information systems – theory, algorithms, the practice of concurrency control and recovery. San Francisco: Morgan Kaufmann; 2002.
16) Shasha D, Bonnet P. Database tuning – principles, experiments, and trouble-shooting techniques. San Francisco: Morgan Kaufmann; 2003.
17) Ramamritham, K., Chrysanthis, P. K., (1997). Advances in Concurrency Control and Transaction Processing. IEEE Computer Society Press, Los Alamitos, California.
18) Grefen P., Apers P. (1993). Integrity Control in Relational Database Systems - An Overview. Journal of Data & Knowledge Engineering (10)2: 187-223.
19) Astrahan M.M., et al. System R: A relational approach to database management. ACM Trans. Database Syst. Vol. 1, No 2 (June 1976), 97-137
20) Moss J.E.B. Nested transactions: an approach to reliable distributed computing. Technical Report. PhD Thesis. UMI Order Number: TR-260: Massachusetts Institute of Technology; 1981. p. 178.
21) Been C, Bernstein P.A., Goodman N, Lai M.Y., Shasha D.E. A concurrency control theory for nested transactions. Proc. of Second ACM Symposium on Principles of Database Systems (PODS), 1983, pp. 45-62
22) Davies C.T. Data processing spheres of control. IBM Syst J. 1978;17(2):179–198.
23) Dayal U., Hsu M., Ladin R. A generalized transaction model for long-running activities and active databases. IEEE Data Engineering Bulletin, March 1991, Vol. 14, No. 1, pp. 4-8
24) Weikum G. and Schek H. Concepts and applications of multilevel transactions and open-nested transactions. In Elmagarmid A., editor. Database Transaction Models for Advanced Applications. Morgan Kaufmann Publishers. San Mateo. CA., 1992, pp. 515–553.
25) Weikum G. Principles and realization strategies of multilevel transaction management. ACM Transactions on Database Systems. 1991;16(1):132–180.
26) Krychniak P., Rusinkiewicz M., Chichocki A., Sheth A., Thomas G. Bounding the Efects of Compensation under Relaxed Multi-Level Serializability. Distributed and Parallel Database Systems, 1996, 4(4), pp. 355-374
27) Lewis, P. M., Bernstein A. J., Kifer M. (2002). Databases and Transaction Processing: An Application-Oriented Approach. Addison-Wesley, United States

28) Breitbart Y., Garcia-Molina H., Silberschatz A. Overview of multidatabase transaction management. VLDB Journal, 1992, vol. 1, No 2, pp. 181-240.
29) X/Open Company Ltd., (1996). Distributed Transaction Processing: Reference Model, version 3. X/Open Company Ltd., U.K.
30) Elmagarmid A.K., Leu Y., Litwin W.,Rusinkiewicz M. (1990) A Multidatabase Transaction Model for InterBase. In Proc. of the 16th. Intl. Conference on Very Large Data Bases, pp. 507-518, Brisbane. Australia
31) Zhang A., Nodine M., Bhargava B., Bukhres O. Ensuring Relaxed Atomicity for Flexible Transactions in Multidatabase Systems. In Proc/ 1994 SIGMOD International Conference on Management of Data, 1994, pp. 67-78
32) Zhang A, Nodine M, Bhargava B. Global scheduling for flexible transactions in heterogeneous distributed database systems. IEEE Trans Knowl Data Eng. 2001;13(3):439–450.
33) Wächter H, Reuter A. The ConTract model. In: Elmagarmid A.K., editor. Database transaction models for advanced applications. Los Altos: Morgan Kaufmann; 1992. pp 39-43
34) Veijalainen J., Eliassen F. The S—transaction Model. In: Elmagarmid A.K., editor. Database transaction models for advanced applications. Los Altos: Morgan Kaufmann, 1992, pp. 55-59
35) Chen J., Bukhres O., Elmagarmid A. K. (1993). IPL: A Multidatabase Transaction Specification Language. In Proc. of the 13th Intl. Conference on Distributed Computing Systems - ICDCS '93. 1993, pp. 439-448
36) Garcia-Molina H. Using Semantic Knowledge for Transaction Processing in a Distributed Database. ACM Transactions on Database Systems, 8(2):186-213, June 1983.
37) Korth H., Levy E., Silberschatz A. A Formal Approach to Recovery by Compensating Transactions. In Proceedings of the 16th International Conference on Very Large Data Bases, Brisbane, Australia, 1990, pp. 95–106
38) Garcia-Molina H., Salem K. Sagas. In Proc. of ACM SIGMOD International Conference on Management of Data, 1987, pp 249-259 San Francisco, CA.
39) Bancilhon F., Kim W., Korth H. A model of CAD Transactions. VLDB '85: Proceedings of the 11th international conference on Very Large Data Bases - Volume 11, 1985, pp. 25–33
40) Garcia-Molina. H., Salem K., Gawlick D., Klein J., Kleissner K., Modeling Long-Running Activities as Nested Sagas, IEEE Data Engineering Bulletin, 1991, 14(1) pp 14-18
41) Pu C., Kaiser G.E., Hutchinson N.C. Split-transactions for open-ended activities. In: Proceedings of the 14th International Conference on Very Large Data Bases; 1988. p. 26–37.
42) Kaiser G.E., Pu C. Dynamic restructuring of transactions. In: Elmagarmid AK, editor. Database transaction models for advanced applications. Burlington: Morgan Kaufmann Publishers; 1992. p. 265–295.
43) Chrysanthis P.K, Ramamritham K. Synthesis of extended transaction models using ACTA. ACM Trans. Database Syst. 1994;19(3):450–491.
44) Nodine M.H., Zdonik S.B. Cooperative transaction hierarchies: Transaction support for design applications. VLDB Journal, 1(1):41–80, 1992.
45) Chrysanthis P.K., Ramamritham, K., (1990). ACTA: A Framework for Specifying and Reasoning about Transaction Structure and Behavior. Proceedings of the ACM SIGMOD International Conference on Management of Data: 194-203.
46) Chrysanthis P.K., Ramamritham K. (1992). ACTA: The SAGA Continues. In Elmagarmid A., editor. Database Transaction Models for Advanced Applications. Morgan Kaufmann Publishers. San Mateo. CA., 1992, pp. 349-397
47) Biliris A., Dar S., Gehani N., Jagadish H., Ramamritham K. (1994). ASSET: A System for Supporting Extended Transactions. In Proc. of ACM SIGMOD Conference on Management of Data, pp. 44-54, Minneapolis, M.N.
48) Abbott R., Garsia-Molina H. Scheduling real-time transactions: a performance

evaluation. ACM Trans, on Database Syst, 17(3), September 1992, pp. 513-560
49) Agrawal D., El Abbadi A., Jeffers R. Using Delayed Commitment in Locking Protocols for Real-Time Databases. SIGMOD Conference 1992: 104-113
50) Hong D., Johnson T., Chakravarthy S. Real-Time Transaction Scheduling: A Cost Conscious Approach. SIGMOD Conference 1993: 197-206.
51) Alonso R., Korth H. Database System Issues in Nomadic Computing. SIGMOD Record, Vol. 22, No 2,1993, pp.388-392.
52) Imelinski T., Badrinath B.R. Data Management for Mobile Computing. SIGMOD Record, Vol. 22, No. 1, 1993
53) Ceponkus A., Dalal S., Fletcher T., Furniss P., Green A., Pope B. Business transaction protocol, Version 1.1, 2002
54) Business transaction protocol. - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=businesstransaction [2004]
55) Stevens M., Mathew S., McGovern J., Tyagi S. Java Web Services Architecture. San Francisco: Morgan Kaufmann Publishers, 2003.
56) WSTx (Web Services Transactions). - https://searchapparchitecture.techtarget.com/definition/WSTx-Web-Services-Transactions
57) IBM, BEA Systems, Microsoft, Arjuna, Hitachi, IONA, \Web Services Transactions specifations," IBM Developer Works, IBM, 2004.
58) Curbera F., Khalaf R., Mukhi N., Tai S.,Weerawarana S. The Next Step in Web Services," Communications of the ACM, October 2003, Vol. 46, No. 10, Pages 29-34
59) OASIS Web Services Composite Application Framework (WS-CAF), OASIS, 2006. - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-caf
60) Little M., Freund Th. J.. A comparison of web services transaction protocols: A comparative analysis of WS-C/WS-Tx and OASIS BTP," IBM, 2003. Available: http://www-128.ibm.com/developerworks/webservices/library/ws-comproto/. [Accessed May 2008].
61) Jin T., Goschnick S. (2004) Utilizing Web Services in an Agent Based Transaction Model. In: Cavedon L., Maamar Z., Martin D., Benatallah B. (eds) Extending Web Services Technologies. Multiagent Systems, Artificial Societies, and Simulated Organizations (International Book Series), vol 13. Springer, Boston, MA. pp 273-291
62) Kratz B., Protocols For Long Running Business Transactions. Technical Report 17, Infolab Technical Report Series, 2004, 48 p.

Этап 4. Расширенные реляционные базы данных (1980+-2000+)

С момента возникновения реляционная модель подвергалась критике в связи с простотой ее структуры данных. В связи с этим предлагались более развитые модели, которые позволяли более адекватно представлять информационные модели различных предметных областей. Тем не менее, их характерной особенностью было то, что все они строились на базе реляционной модели и получили название расширенных реляционных баз данных. Подавляющее большинство перечисленных в следующем абзаце БД либо создавались на основе реляционных БД, либо имели варианты такой реализации.

В этот период активизировались исследования по взаимопроникновению технологий искусственного интеллекта (ИИ) и БД. В 1988 г. состоялись два отдельных симпозиума по интеграции ИИ и БД. Отвечая на эти потребности, в БД возникли два направления по представлению в них правил, порождающих новые данные из существующих, и в результате возникли БД двух типов: активные и дедуктивные. Кроме того, потребности включения в БД времени и пространства привело к появлению темпоральных и пространственных БД, а потребность применения объектной технологии к БД привела к появлению объектных БД. Стремление существенно повысить производительность баз данных для работы с большими объемами данных привело к исследованиям и разработкам машин баз данных, а успехи в создании компьютерных сетей привели к появлению распределенных и параллельных баз данных. Наконец, в этот же период пришло осознание того, факта, что БД должны использоваться не только для "рутинной" работы по сбору, хранению и поиску тщательно отобранных и проверенных данных, но и для их систематизации, обобщению, статистической и аналитической обработки. Так появились статистические БД, БД для работы с массивами, многомерные БД и, наконец, хранилища данных. Отказ от 1NF привел к появлению ненормализованных (вложенных) БД.

Темпоральные базы данных

Темпоральные базы данных – это базы данных, хранящие данные, привязанные ко времени. Время, как отдельный тип данных, присутствует во всех СУБД, но это не является основанием считать их темпоральными, так как интерпретация времени и семантика взаимосвязи между временем и данными остается за разработчиком. В темпоральных базах данных должны существовать правила интерпретации времени и возможности по раскрытию семантики взаимосвязи данных со временем.

Исследования по использованию понятия времени в информационных системах были предприняты уже в 60-х годах прошлого столетия. Считается [1], что впервые идеи фиксации изменяющейся во времени информации в базах данных появились в 1976 г. в работе Яниса А. Бубенко мл. (Janis A. Bubenko, Jr) [2]. Впоследствии были предприняты активные исследования по раскрытию семантики времени на концептуальном уровне [3–7], созданию зависимых от времени моделей данных для статических реляционных баз данных [8–11] и разработке темпоральных языков запросов [12–16].


Янис Бубенко мл.

Следует отметить, что понятие времени было включено во многие модели данных, включая объектно-ориентированную, сущность-связь, семантическую, дедуктивную, а также модели, базирующиеся на знаниях. Ссылки на статьи с этими моделями данных можно найти в обзоре [17]. Однако подавляющее большинство работ по темпоральным базам данных основываются на реляционной модели.

Основные понятия

Уже к середине 80-х годов сложились основополагающие положения темпоральных БД (ТБД). Суть их заключается в следующем.

Темпоральный домен (temporal domain) в самом общем случае определяется как множество темпоральных индивидов

(temporal individuals), на которых заданы темпоральные отношения (temporal relations). Темпоральный домен характеризуется следующими аспектами: структурным (линейное время, ветвящееся время), дискретным (непрерывное время, дискретное время), граничным (ограниченное время, бесконечное время) и относительностью (абсолютное, относительное).

В качестве темпоральных индивидов могут выступать моменты времени (временные точки) или временные интервалы. Для моментов времени задано отношение линейного порядка (linear order), а для интервалов - отношения Аллена [18]

Ассоциация темпоральных индивидов с данными базы данных производится с помощью временных отметок (timestamps.)

Линия времени - это временная ось, заданная на конкретном временном домене и предназначенная для ассоциации временных отметок с данными. Выделяют две линии времени:

- Линия действительного времени. *Действительное время* (valid time) определяет период времени, на протяжении которого имеет место (истинен) тот или иной факт моделируемой реальности.
- Линия транзакционного времени. *Транзакционное время* (transaction time) - это период времени, на протяжении которого информация о факте хранится в базе данных.

Базы данных, поддерживающие обе линии времени, называются *битемпоральными*.

Кроме того, предполагается существование пользовательского времени (user - defined time) - время (интервал времени), которое привязывается в базе данных факту самим пользователем.

На линии времени присутствует специальный момент времени, который называется СЕЙЧАС, обладающий специфическими особенностями [19]

В темпоральной реляционной модели время может привязываться либо к атрибутам, либо к кортежам.

Темпоральные модели данных

Темпоральная модель данных - это модель, в которой предоставляется возможность привязывать данные ко времени. Эти модели отличаются тем, какие линии времени они поддерживают (модель действительного времени, модель транзакционного времени, битемпоральная модель), какие темпоральные индивиды используют (моменты времени - точечная модель, или интервалы - интервальная модель) и к каким данным привязывается время (к значениям атрибутов или значениям кортежей).

Пик исследований по темпоральным моделям данных приходится на 80-е годы. Приведем далеко не полный список работ, посвященных темпоральным моделям данных:

- Яков Бен-Зви (Jacob Ben-Zvi) [13, 14] - битемпоральная интервальная модель;
- Джонс и Мейсон (Jones and Mason) [15] - интервальная модель действительного времени;
- Ричард Снодграс (Richard Snodgrass) [16] - битемпоральная точечная и интервальная модель;
- Лоренцос и Джонсон (Lorentzos and Johnson) [20, 21] - точечная и интервальная модель действительного времени.

Все эти модели предполагали привязку времени к кортежам. Также были предложены темпоральные модели с привязкой времени к атрибутам [22–27].

С обзором исследований по темпоральным моделям данных можно познакомиться в работах [28, 29–31].

Темпоральные зависимости

Теория проектирования баз данных базируется на понятии зависимостей. Из них фундаментальным понятием является функциональная зависимость.

Динамический вариант функциональной зависимости (DFD) впервые был предложен Виану (Vianu) [32]. Предполагалось, что такая FD должна выполняться в текущем кортеже и в его обновленном варианте. В этой же работе была исследована взаимосвязь между DFD и статическими FD. Еще один вид темпоральной FD был предложен Вийсеном (Wijsen) [33, 34]. В этом случае

требуется, чтобы FD выполнялась в объединении старого и нового отношения. Вийсен также определил тренд-зависимость (trend dependency) [35], которая является обобщением определенной им же темпоральной FD.

Еще одна разновидность TFD была определена в работе [36]. В этой статье было предложено расширение теории нормализации с учетом избыточности, которая порождается TFD. В работе [37] была определена зависимость, порожденная ограничениями (constraint generating dependency - CGD). CGD означает, что каждый атрибут отношения принимает значения из домена, определенного ограничением. Это характерно для атрибутов временных отметок темпоральных моделей данных. Наконец, в работе [38] были исследованы темпоральные расширения отношений специализации и обобщения.

Темпоральные языки

Темпоральный язык запросов базы данных – это язык, который обладает встроенными возможностями манипулирования темпоральныи данными, а также спецификации утверждений и ограничений, накладываемых на такие данные. Такой язык обычно тесно связан с темпоральной моделью данных.

Были проведены исследования и предложены темпоральные реляционные алгебры [20–22, 26, 27, 39–41] и исчисления [23, 26] для различных темпоральных моделей данных. Была также определена вложенная битемпоральная модель данных и соответствующая ей алгебра [24]

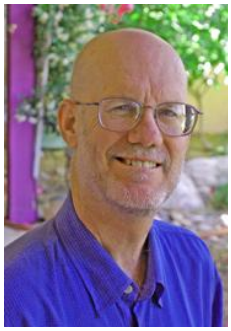
Тансел и Аркун разработали язык HQUEL [42], представляющий собой расширение языка QUEL "историческими" данными. Они же предложили язык TBE (Time-By-Example) [43], в котором воспользовались идеей графического реляционного языка QBE. Снодграс также предложил темпоральный вариант QUEL, которые был назван TQUEL [16, 44].

Были предложены различные варианты темпорального расширения SQL [25, 45–49].

Обширный перечень темпоральных реляционных и объектных языков запросов приведен в [28], а в [50] приводится обзор некоторых из них. Обзору темпоральных языков запросов также освящены работы [51, 52].

TSQL2

Одним из ключевых периодов в области исследований темпоральных баз данных, временем ее «официального» представления можно считать 1992–1995 гг. Сначала Ричард Снодграс (Richard T. Snodgrass) высказал идею о возможном темпоральном расширении стандарта SQL-92, а затем в 1993 г. был проведен семинар [53], продемонстрировавший заинтересованность научного сообщества в разработке темпорального расширения стандарта SQL-92. В результате был учрежден комитет по созданию такого языка, который получил название Temporal Structured Query Language TSQL2. Ведущую роль в работе комитета сыграл Снодграс. Уже в сентябре 1993 г. был выпущен первый черновой вариант языка, а в декабре - второй. В результате плодотворной работы, в марте 1994 г. появилась первая предварительная версия спецификации языка [54], а в сентябре - учебное пособие [55]. Наконец, в 1995 г. была опубликована окончательная спецификация языка запросов TSQL2 [56].


Ричард Снодграс

Последующая деятельность была связана с включением и расширением основных идей TSQL2 в SQL3. Этот язык был назван SQL/Temporal. Были проработаны вопросы поддержки в SQL/Temporal действительного и транзакционного времени [57, 58]. Окончательные предложения перехода от TSQL2 в SQL3 были сформулированы в [59].

Темпоральный SQL: 2011

В 1995 году в ANSI/ISO было принято решение о разворачивании работ по созданию нового стандарта SQL, который бы включал темпоральные свойства. В связи с этим США внесло предложение по расширению соответствующих возможностей SQL, которые базировались на пионерских

исследованиях коллектива под руководством Снодграсса.

Эти предложения базировались на детально проработанных к тому времени группой Снодграсса спецификациях языка TSQL2, являющегося темпоральным расширением SQL-92, а также на предложениях переноса TSQL2 в SQL3. Однако некоторые члены ISO выразили сомнения по поводу этих предложений США в связи с существованием в них серьезных проблем и противоречий. В вою очередь Великобритания внесла предложение, которое было сформулировано на основе исследований Никоса Лоренцоса (Nikos Lorentzos) из университета Афины, Греция. США не согласились позицией ISO по отношению к их предложению и не поддержали предложение Великобритании. В связи с этим ANSI и ISO решили отложить дальнейшую работу по темпоральному SQL до официальной публикации версии SQL-99.

После публикации SQL-99 ни США ни Великобритания не внесли никаких новых предложений, которые бы разрешали возникшие ранее разногласия. В связи с этим в 2001 году ANSI и ISO решили прекратить деятельность по созданию стандарта темпорального SQL. Вторая попытка по добавлению темпоральных свойств в SQL была предпринята в 2008 году. Она началась с обсуждения, введения и принятия предложений двух комитетов INCITS DM32.2 и ISO/IEC JTC1 SC32 WG3 по «системно-версионным таблицам» (systemversioned tables). Еще одна темпоральная черта была добавлена в SQL в 2010 году в виде «таблиц с прикладными периодами» (application-time period tables). Эти два понятия и разработанные для них соответствующие языковые средства были включены в стандарт SQL: 2011. С темпоральными особенностями SQL: 2011 можно познакомиться в [60, 61].

Литература

1) Snodgrass R.T., Ahn I. A taxonomy of time databases. ACM SIGMOD Record, 1985,Vo. 14, No 4, pp. 236-246
2) Bubenko J.A, Jr. The temporal dimension in information modeling. Technical Report RC 6187 #26479, IBM Thomal J. Watson Research Center, Nov. 1976
3) Bubenko J.A. Jr. The Temporal Dimension in Information Processing. In: Proceedings of IFIP WG 2.6 Working Conference on Architecture and Models in Data Base Management Systems, G M Nijssen, Ed, North Holland, 1977, pp. 93-118
4) Breutmann B., Falkenberg E., Mauer R. "CSL: a language for defining conceptual schemas". in Proceedings of the Database Architecture Conference, Venice, June 1979, pp. 237-256
5) Hammer M., McLeod D. Database Descitption with SDM A Semantic Database Model ACM Transactions on Database Systems, 6, No 3, Sep 1981, pp 351-386
6) Klopprogge M.R. TERM: An Approach to Include the Time Dimension in the Entity-Relationship Model. In: Proceedings of the Second International Conference on the Entity Relationship Approach, Washington, DC, pp. 477–512 (October 1981)
7) Anderson,T.L. Modeling Time at the Conceptual Level. In Improving Database Usability and Responsiveness, Ed. P. Scheuermann Jerusalem, Israel Academic Press, 1982, pp. 273-297
8) Codd, E.F. Extending the database relational model to capture more meaning. ACM Transactions on Database Systems, Vol. 4, No. 4, Dec 1979, pp 397-434
9) Sernadas A Temporal aspects of logical procedure definition. Information Systems, 1980, vol. 5, No 3, pp. 167-187
10) Clifford, J. and Warren D.S. Formal semantics for time in databases. ACM Transactions on Database Systems, vol. 8, No 2, June 1983, pp. 214-254
11) Ariav G. A temporally oriented data model. ACM Transactions on Database Systems (TODS), 1986 vol. 11, No 4, pp. 499-527
12) Ariav G., Morgan H.L., Zisman M.D. MDM: Em
, Technical Report 82-03-01 Department of Decision Sciences, Wharton School, University of Pennsylvania, 1982

13) Ben-Zvi J. “The Time Relational Model,” PhD thesis, Computer Science Dept., UCLA, 1982
14) Gadia S. Ben-Zvi's Pioneering Work in Relational Temporal Databases. In: Tansel A. et al. Temporal Databases: Theory, Design, and Implementation (Redwood City, CA: The Benjamin/Cumming s Publishing Company, 1993). pp. 202-207
15) Jones S., Mason P.J. Handling the Time Dimension in a Data Base. In Proceedings of the Inlernational Conference on Data Bases, Eds. S.M. Deen and P Hammersley British Computer Society University of Aberdeen, Heyden, July 1980 pp 65-83
16) Snodgrass R. The temporal query language TQuel. In PODS '84: Proceedings of the 3rd ACM SIGACT-SIGMOD symposium on Principles of database systems.
17) Ozsoyoglu G., Snodgrass R.T.. Temporal and Real-Time Databases: A Survey. IEEE Transactions for Knowledge and Data Engi-neering 7(4):513-532, 1995.
18) Allen J.F. "Maintaining knowledge about temporal intervals". Communications of the ACM, Nov. 1983, 26(11. pp.832-843
19) Clifford J., Dyreson C.E., Isakowitz T., Jensen C.S., Snodgrass R.T. On the semantics of “now” in databases. ACM Trans Database Syst. 1997; 22(2), pp. 171–214.
20) Lorentzos N.A., Johnson R.G. TRA a model for a temporal relational algebra. In: Rolland C, Bodart F, Leonard M, editors. Temporal aspects in information systems. North-Holland; 1988. p. 203–215.
21) Lorentzos N.A., Johnson R.G. Extending relational algebra to manipulate temporal data. Inf Syst. 1988; 13(3): 289–296.
22) Tansel A.U. Adding time dimension to relational model and extending relational algebra. Information Systems. 1986; 11(4):343–355.
23) Tansel A.U. Temporal relational data model. IEEE Transactions on Knowledge and Data Engineering, 1997, 9(3): 464-479
24) Tansel A.U., Atay C.E. Nested bitemporal relational algebra. Conference: Computer and Information Sciences - ISCIS 2006, 21th International Symposium, Istanbul, Turkey, November 1-3, 2006, Proceedings, pp. 622-633
25) Navathe S.B., Ahmed R. A temporal relational model and a query language. Information Sciences: an International Journal. 1989;49(1-3):147–175.
26) Gadia S.K. A homogeneous relational model and query languages for temporal databases. ACM Trans Database Syst. 1988;13(4):418–448.
27) Clifford J, Croker A. The historical relational data model (HRDM. and algebra based on lifespans. In: Proceedings of the 3rd International Conference on Data Engineering; 1987. p. 528–537.
28) Zaniolo C., Ceri S., Faloutsos Ch., Snodgrass R.T., Subrahmanian V.S., Zicari R. Advanced Database Systems. The Morgan Kaufmann Series in Data Management Systems. 1997, 574 p.
29) Gregersen H., Jense C.S. Temporal Entity-Relationship Models—a Survey. IEEE Transactions on Knowledge and Data Engineering, 1999, Vol. 11, No. 3, pp. 464 - 497
30) Arora S. A comparative study on temporal database models: A survey, 2015 International Symposium on Advanced Computing and Communication (ISACC), 2015, pp. 161-167,
31) Gandhi L. Literature survey of temporal data models. International Journal of Latest Trends in Engineering and Technology. 2017, Vol. 8 No. 4-1, pp.294-300
32) Vianu V. Dynamic functional dependencies and database aging. J ACM. 1987;34(1):28–59.
33) Wijsen J. Design of temporal relational databases based on dynamic and temporal functional dependencies. In: Clifford J, Tuzhilin A, editors. Temporal databases. Workshops in computing. Berlin/Heidelberg/New York: Springer; 1995. p. 61–76.
34) Wijsen J. Temporal FDs on complex objects. ACM Trans Database Syst. 1999;24(1):127–176.

35) Wijsen J. Reasoning about qualitative trends in databases. IInformation Systems. 1998;23(7):463–487.
36) Wang X.S., Bettini C., Brodsky A., Jajodia S. Logical design for temporal databases with multiple granularities. ACM Trans Database Syst. 1997;22(2):115–170.
37) Baudinet M., Chomicki J., Wolper P. Constraint generating dependencies. J Comput Syst Sci. 1999;59(1):94–115.
38) Jensen C.S., Snodgrass R.T. Temporal specialization and generalization. IEEE Trans Knowl Data Eng. 1994;6(6):954–974
39) Sarda N.L. Algebra and query language for a historical data model. The Computer Journal. 1990;33(1):11–18
40) Lorentzos N.A., Johnson R.G. Extending relational algebra to manipulate temporal data. Information Systems. 1988;13(3):289–296.
41) Tuzhilin A, Clifford J. A temporal relational algebra as basis for temporal relational completeness. In: Proceedings of the 16th International Conference on Very Large Data Bases; 1990. p. 13–23.
42) Tansel A.U., Arkun M.E. HQUEL, a Query Language for Historical Relational Databases. SSDBM'86: Proceedings of the 3rd international workshop on Statistical and scientific database management,1986 pp. 135-142
43) Tansel A.U., Arkun M.E, Ozsoyoglu G. Time-by-example query language for historical databases. IEEE Trans Softw Eng. 1989;15(4):464–478.
44) Snodgrass S. The temporal query language TQUEL. ACM Trans Database Syst. 1987;12(2): 247–298.
45) Lorentzos N.A., Mitsopoulos Y.G. SQL extension for interval data. IEEE Trans Knowl Data Eng. 1997;9(3):480–99.
46) Sarda N.L. Extensions to SQL for historical databases. IEEE Trans Knowl Data Eng. 1990;2(2):220–230.
47) Navathe S.B., Ahmed R. TSQL: a language interface for history databases. In: Rolland C, Bodart F, Leonard M, editors. Temporal aspects in information systems. North-Holland; 1988. p. 109–122.
48) Toman D. Point-based temporal extensions of SQL and their efficient implementation. In: Etzion O, Jajodia S, Sripada S, editors. Temporal databases: research and practice. Springer,; 1997, p. 211–237.
49) Böhlen M.H., Jensen C.S., Snodgrass R.T. Temporal statement modifiers. ACM Trans Database Syst. 2000;25(4):407–456.
50) Chomicki J. Temporal query languages: A survey. H.J. Ohlbach and D.M. Gabbay, eds., Proc. First Int'l Con$ Temporal Logic. Lecture Notes in Artificial Intelligence 827, Springer-Verlag, pp. 506-534, July 1994.
51) McKenzie E., Snodgrass R.T. An evaluation of relational algebras incorporating the time dimension databases. ACM Computing Surveys, 1991, vol. 23, No. 4, pp. 501-543
52) Snodgrass R.T. "Temporal object-oriented databases: A critical comparison. Chap. 19, Modern Database Systems: The Object Model, Interoperability and Beyond, W. Kim, ed., Addison-WesleytACM Press, pp. 386-408,1995.
53) Snodgrass R.T. editor. In: Proceedings of the ARPA/NSF International Workshop on an Infrastructure for Temporal Databases, 1993.
54) Snodgrass R.T., Ahn I., Ariav G., Batory D.S., Clifford J., Dyreson C.E., Elmasri R., Grandi F., Jensen C.S., Käfer W., Kline N., Kulkarni K., Leung T.Y.C., Lorentzos N., Roddick J.F., Segev A., Soo M.D., Sripada S.M. TSQL2 language specification. ACM SIGMOD Rec. 1994;23(1):65–86.
55) nodgrass R.T., Ahn I., Ariav G., Batory D.S., Clifford J., Dyreson C.E., Elmasri R., Grandi F., Jensen C.S., Käfer W., Kline N., Kulkarni K., Leung T.Y.C., Lorentzos N., Roddick J.F., Segev A., Soo M.D., Sripada S.M. A TSQL2 tutorial. ACM SIGMOD Rec. 1994;23(3):27–33
56) Snodgrass R.T. Editor. The TSQL2 temporal query language. Kluwer Academic; 1995.
57) Snodgrass R.T., Böhlen M.H., Jensen C.S., Steiner A. Adding valid time to

SQL/temporal. Change proposal, ANSI X3H2-96-501r2, ISO/IEC JTC1/SC21/WG3 DBL MAD-146r2, Nov 1996.
58) Snodgrass R.T., Böhlen M.H., Jensen C.S., Steiner A. Adding transaction time to SQL/temporal. Change proposal, ANSI X3H2-96-502r2, ISO/IEC JTC1/SC21/ WG3 DBL MAD-147r2, Nov 1996.
59) Snodgrass R.T., Böhlen M.H., Jensen C.S., Steiner A. Transitioning temporal support in TSQL2 to SQL3. In: Ezion O, Jajodia S, Sripada SM, editors. Temporal databases: research and practice. Berlin: Springer; 1998. p. 150–194.
60) Kulkarni K, Michels J-E. Temporal features in SQL:2011. ACM SIGMOD Rec. 2012;41(3):34–43.
61) Reznichenko V.A. Temporal SQL:2011 (Rus). Software Engineering, 2013, vol. 15, No 3-4, pp. 48-65

Пространственные базы данных

Пространственная база данных (ПБД) *— это база данных, предназначенная для хранения, манипулирования и выполнения запросов к данным о пространственных объектах, представленных некоторыми абстракциями.* В то время как традиционные БД предназначены для хранения и обработки числовой и символьной информации, ПБД предоставляющие возможности работы с целостными пространственными объектами, объединяющими как традиционные виды данных (описательная часть или атрибутивная), так и геометрические (данные о размерах и положении объектов в пространстве).

Еще в начале 1970-х г. понятием обработки пространственных данных пользовались для обозначения деятельности, связанной с электронной обработкой данных для повышения производительности при составлении и редактировании карт, картографических измерениях и анализе пространственных данных. Идея компьютерного хранения геометрических данных появилась в конце 70-х гг. в связи с возрастающим успехом реляционных баз данных.

Модели пространственных данных

Двумя видами моделей пространственных данных являются: полевая и объектная.

Полевая модель

Используется для представления непрерывных или аморфных явлений, например, температуры или облачности. Эта модель поддерживает функциональную точку зрения, когда базисная система отсчета (пространственная система координат, например, широта и долгота) функционально отображается в заданную область значений, например, в градусы для температуры. Компьютерной реализацией полевой модели является растровая структура данных - равномерная решетка, наложенная на базисное пространство. Другими популярными структурами данных для представления полей являются триангулированная нерегулярная сеть (triangulated irregular network TIN), линии контура, точечные решетки.

Операции полевой модели делятся на три типа [1]:

– *локальные операции* - значение функции в данной точке зависит только от значения аргумента в этой точке (поточечная сумма, разность, максимум, среднее значение);
– *фокальные операции* - значение функции в данной точке зависит от значений в малой окрестности этой точки (наклон, средневзвешенное значение в окрестности);
– *зональные операции* - полевая функция задается не на точках или их малых окрестностях, а на областях (многоугольниках) целиком (сумма, среднее, максимальное, минимальное полевое значение каждой зоны).

Объектная модель

Трактует информационное пространство как совокупность дискретных, идентифицируемых, пространственных сущностей. Она в компьютерах представляется так называемой векторной структурой данных. Главный вопрос объектной модели - выбор базового множества типов пространственных данных. Было проведено множество исследований, в результате которых был разработан стандарт OGC [2], который определил следующие типы (с некоторым упрощением):

– простые объекты - точки, кривые, поверхности,
– наборы объектов - набор точек, набор кривых, набор поверхностей.

Операции

Было проведено множество исследований по определению пространственных операций. Статья [3] стала одной из первых попыток общего описания операций над картами (алгебра карт) с позиций растрового анализа и стала базовым языком для работы с полевыми моделями. В дальнейшем эта алгебра была уточнена в [4]. В [5] предложен расширяемый язык запросов для географических баз данных. В статье [6] впервые было предложено расширение алгебры реляционной модели введением пространственных объектов и операций. В статье [7] предложен язык Spatial SQL, в котором в язык SQL включены пространственные операции и отношения. Алгебра ROSE (**RO**bust Spatial **E**xtension) [8] базируется на реляционной модели, использует типы данных для представления точек, линий и областей и предлагает исчерпывающий набор операций; семантика типов и операций определена формально. В работе [9] представлена пространственная логика, которая может использоваться для рассуждений относительно топологических и пространственных взаимосвязей между объектами. Преимущество данного подхода - строго определенная семантика и использование механизма логического вывода.

Для объектной модели были определены следующие операции/предикаты [10]:

– *операции на множествах* (равно, не равно, является членом, является пустым, пересечение, объединение, разность, кардинальность, ...);
– *топологические операции* (является границей, внутренняя часть, внешняя часть, замыкание, касаются, пересекаются, находится внутри, находится снаружи, охватывает, содержится в, ...);
– *метрические операции* (расстояние, угол, длина, площадь, периметр, ...);
– *операции направления* (на север, на восток, слева, сверху, спереди, между, ...);
– *сетевые операции* (предшественник, последователь, соединены, путь, ...);
– *динамические операции* (повернуть, масштабировать, сдвинуть, разделить, слить, ...).

В терминах объектно-реляционных баз данных пространственная модель данных реализуется определением пространственных типов данных и операций над объектами этих типов. В связи с этим многие работы по ПБД были направлены на разработку абстрактных типов данных (АТД) и их внедрению в языки запросов. Благодаря созданию консорциума Open Geospatial Consortium Inc. (OGC) удалось серьезно продвинуться в области создания стандартов по геопространственным технологиям [11, 12]. В частности OGC представил спецификацию [12] встраивания в SQL двумерных геопространственных АТД на основе объектной

модели, и предложил исчерпывающий список операций. В монографии [10] дается глубокий анализ проблематики ПБД.

Пространственные типы данных

Пространственные типы данных предоставляют возможность моделировать объекты в пространстве, а также их взаимосвязи, свойства и операции. Они представляют особый интерес в ПБД [10, 13 14]. Большинство наиболее популярных абстракций пространственных объектов относятся к классу структурных пространственных типов данных. Эти типы данных представляют пространство в виде точек, линий, областей, поверхностей, объемов, пространственных разбиений (spatial partitions), пространственных сетей и других подобных объектов. То есть пространственные объекты рассматриваются с точки зрения их структурной формы и пространственных размеров. Пространственные типы данных для точек, линий и областей исследуются в [6, 8, 15–19], для поверхностей и объемов в [20], для пространственных разбиений в [21] и для пространственных сетей в [22]. Оригинальный подход для определения пространственных типов данных был предложен в [22], который был назван Realm. Realm - это конечное множество точек и непересекающихся линейных отрезков, которые могут располагаться в узлах равномерно распределенной сетки. На основе этих примитивных понятий определяются более сложные структуры и операции над ними.

Отношение главного направления

Понятие "направление" является одной из важных характеристик пространственных систем. Алгоритмы вычисления пространственных направлений лежат в основе ПБД и ГИС.

Отношение главного направления - *ОГН* (cardinal direction relationships) - это пространственное отношение, указывающее расположение одного объекта относительно другого. Оно имеет как количественное, так и качественное значение. Было предложено ряд моделей ОГН. На начальном этапе предлагались модели, представляющие точками пространственные объекты, а направление определялось согласно наносимой сетке [23, 24]. В проекционных моделях сетка наносилась параллельно осям координат, а в конусной модели - под углом. В последующих моделях объекты аппроксимировались так называемыми "представительными" областями, среди которых наиболее часто предлагались ограничивающие прямоугольники [25, 26]. Однако этот метод давал неправильное направление, когда объекты наложены друг на друга, переплетены или подковообразные [27]. Затем были предложены более точные модели ОГН, в которых исходные объекты представлены своими точными фигурами, а ссылочные объекты аппроксимируются ограничивающим прямоугольником. Опять же, в зависимости от наносимой сетки различают модели проекционных направленных связей (projection-based directional relationships) [28, 29, 30] и конусообразных направленных связей) [10]. Было также предложено моделировать ОГН тернарными отношениями [31]. Относительно перечисленных выше моделей ОГН исследовались следующие задачи:

- эффективное определение отношений, которые имеются между множествами объектов [26, 31, 32, 33],
- вычисление инверсных отношений [24, 26, 31, 34, 35],
- вычисление композиции двух или боле отношений [24, 26, 29, 35],
- проверка согласованности множества отношений [24, 26, 30, 36, 37].

Оригинальное решение по моделированию направления в виде пространственного объекта было предложено в работе [38].

Концептуальное моделирование

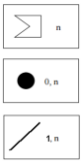
Общепризнанным средством концептуального моделирования является ER-язык. Было предложено много расширений этого языка для представления в нем пространственных объектов с пространственными характеристиками [39]. Одним из таких расширений является ER-схема с пиктограммами [40]. В ней пиктограммы используются для указания типов пространственных сущностей и пространственных связей. Ба-

зовыми типами геометрических фигур являются: точка, (ломаная) линия и многоугольник. Они имеют следующие пиктограммы объектов:

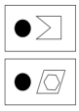

точка


линия


многоугольник


Мультифигуры - это множества базовых фигур. Предоставляется возможность указать количество элементов в мультимножестве по аналогии с указанием мощности окончаний связей в обычном ER-языке (M : N - не менее M и не более N)


Производные фигуры - фигура объекта является производной от фигур других объектов, например, многоугольник страны является производным от многоугольников областей.


Альтернативные фигуры - объект может быть представлен несколькими фигурами, например, в зависимости от масштаба город представляется точкой или многоугольником.


Любая возможная фигура - объект может быть представлен любой допустимой фигурой.

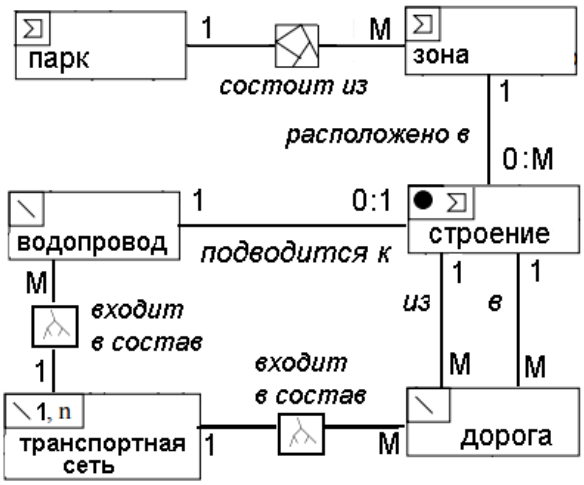

Определяемая пользователем фигура - объект представляется не стандартной фигурой, а определенной пользователем.

Пиктограммы связей:


Является частью в смысле иерархической структуры.


Является частью в смысле отношения разбиения (partition).

На рисунке ниже приведен пример пространственной ER-схемы.



Детальное описание этого расширения ER-языка приводится в работах [10, 41].

Геопространственные онтологии

Были высказаны предложения по использованию геопространственных онтологий для концептуального моделирования геоинформационных систем [41], которые давали бы возможность представлять в модели ПО более глубокую пространственную семантику. Кроме того, предлагается расширить геопространственные онтологии темпоральными характеристиками с использованием OWL-Time [43].

Все исследования по пространственным онтологиям разделяются на две категории:

– онтологии для интеграции структур хранения пространственных данных. В этом случае решатся задача интеграции различных ПБД с целью решения задачи обмена данными между ними [44];
– онтологии для более точного представления семантики данных. В этом случае пространственная онтология создается либо для отражения семантики конкретной предметной области с характерным только для нее набором объектов и понятий [45], либо создается некая универсальная онтология, охватывающая максимально широкий круг понятий и характеристик геопространственных объектов [46–48].

Огромный объем геоинформационных ресурсов в интернет инициировал исследования по созданию геопространственного семантического веба [49, 50], так, например,

было предложено геопространственное расширение RDF (GeoRDF) [51].

Многомерные методы доступа

Поисковые операции в БД требуют специальной поддержки на физическом уровне. Это имеет мест о как в обычных БД, так и в пространственных БД, в которых обычные поисковые операции включают точечные запросы (point query), например, найти все объекты, которые содержат данную точку, и пространственные запросы (region query), например, найти все объекты, которые пересекаются с данной областью. За многолетнюю историю развития пространственных БД было разработано множество многомерных методов доступа (ММД) для поддержки таких операций.

С точки зрения используемых методов доступа ПБД и БД изображений (БДИ) очень близки между особой. ПБД содержат многомерные данные с учетом наличия явных знаний о том, что собой представляют объекты, какова их форма, место расположения и взаимосвязь. В свою очередь БДИ содержит исходные изображения, которые представлены в виде графических данных, которые в результате специальных методов анализа преобразуются в многомерные данные, представимые с помощью тех же методов доступа, что и пространственные данные. Далее мы кратко проанализируем ММД, применимые в ПБД, а их использование в БДИ описано в разделе «Базы данных изображений».

Выделяются при категории методов доступа:

– основные;

– точечные;

– пространственные.

К основным методам доступа относятся одномерные методы и методы основной памяти

Классические *одномерные методы доступа* являются фундаментом практически для всех ММД. На практике наиболее общепризнанными одномерными методами доступа являются линейное хеширование, расширяемое хеширование и B-деревья.

На заре становления ММД они не учитывали страничную организацию внешней памяти, поэтому были мало пригодными для больших ПБД. В связи с этим они получили название *методов оперативной памяти*. Тем не менее, со временем они были адаптированы и включены во многие ММД. К ним относятся K-D-, BSP-, BD-деревья и разновидности квадродеревьев.

Точечные методы доступа (Point Access Methods - PAM) – структуры и методы, предназначенные для представления и поиска точек в многомерном пространстве. Выделяют две разновидности PAM - *многомерное хеширование* и *иерархические методы доступа*. Идея многомерного хеширования заключается в том, чтобы использовать такие функции хеширования, которые бы позволяли располагающиеся близко друг к другу объекты в реальном пространстве хранить близко друг к другу на диске. Иерархические методы доступа предполагают построение такой древовидной структуры, листья которой содержат точечные блоки (bucket).

Пространственные методы доступа предназначены для представления и поиска пространственных объектов. Для оперирования такими объектами PAM были модифицированы с использование одного из следующих методов:

– трансформация (отображение объектов);

– перекрывающиеся области;

– разрезание (дублирование объекта);

– многослойность.

Трансформация. Одномерные методы доступа могут использоваться для оперирования пространственными объектами при условии, что объекты предварительно трансформируются в другое представление. Предлагается два варианта такого преобразования: либо объект трансформируется в точку более высокой размерности, либо в совокупность одномерных интервалов.

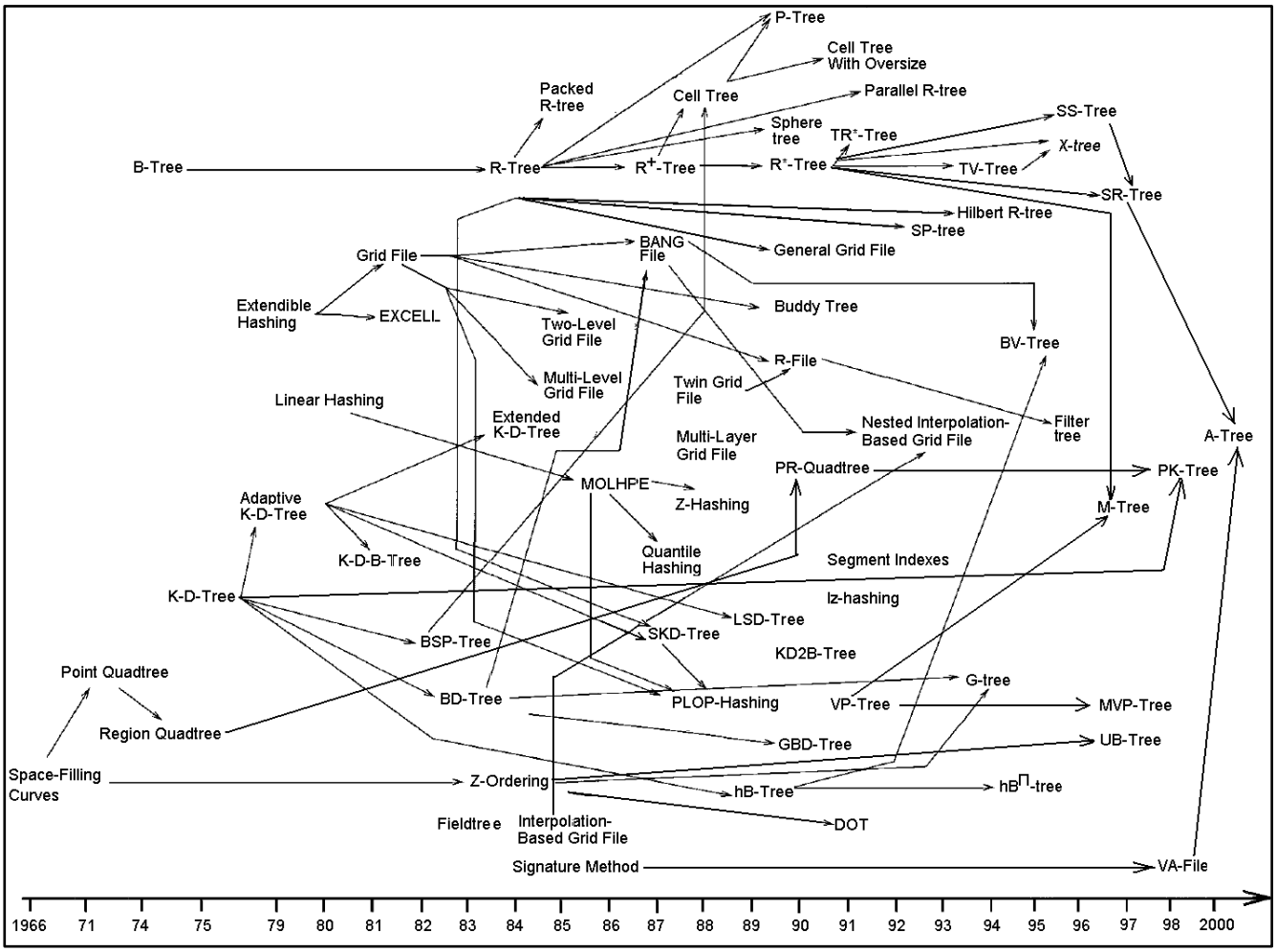
Перекрывающиеся области. Эти методы предполагают декомпозицию пространства иерархическим образом Объекты хранятся в листьях иерархии, а промежуточные вершины повышают эффективность поиска. Вершины одного уровня могут перекрывать друг друга.

Разрезание. Этот метод не допускает каких-либо пересечений между областями блоков. Любой объект, располагающийся в

нескольких областях блока, разрезается вдоль разделяющих гиперплоскостей.

Многослойность. Пространство разбивается на разделы, которые называются слоями. Слои организованы иерархическим образом. Области раздела одного уровня не пересекаются. Объекты запоминаются на самом нижнем слое без разрезания

На рисунке ниже, являющимся интеграцией двух подобных рисунков из [52, 53], приведено около 70 хронологически упорядоченных методов доступа, опубликованных до 2001 г., а в последующей таблице – распределение этих методов по перечисленных выше категориям. В этих двух статьях все эти методы систематизированы, кратко описаны, указаны библиографические ссылки их первоисточников и дан краткий сравнительный анализ. В статье [54] дается более детальный сравнительный анализ по сравнению с предыдущими двумя. Всесторонний анализ пространственных методов доступа проведен в монографии [55].



Категории	Подкатегории	Методы
Основные методы доступа	Одномерные методы доступа	Linear Hashing, Extendible Hashing, B-Tree, BD-Tree
	Методы основной памяти	K-D-Tree, BSP-Tree, Quadtree (Point Quadtree, Region Quadtree)
Точечные методы доступа	Многомерное хеширование	Grid File, EXCELL, Two-Level Grid File, Twin Grid File, Multidimensional Linear Hashing. (MOLHPE, Quantile Hashing, PLOP-Hashing, Dynamic Z-Hashing)

	Иерархические методы доступа	K-D-B-Tree, LSD-Tree, Buddy Tree, BANG File, G-tree, hB-Tree, hB$^{\Pi}$-tree, Π -tree, BV-Tree
Пространственные методы доступа	Трансформация (отображение объектов)	Mapping to Higher-Dimensional Space, Space-Filling Curves for Extended Objects, Z-ordering
	Перекрывающиеся области	R-Tree, Packed R-Tree, Sphere Tree, Hilbert R-Tree, R*-Tree, P-Tree, SKD-Tree, GBD-Tree, PLOP-Hashing
	Разрезание	Extended K-D-Tree, R^{+}-Tree, Cell Tree
	Многослойность	Multilayer Grid File, R-File, MX-CIF Quadtree, Filter Tree

Пространственно-сетевые базы данных (ПСБД)

Они предназначены для поддержки пространственных сетей путем предоставления необходимых модели данных, языка запросов, структуры хранения и методов индексирования. Модель пространственной сети может быть представлена в виде графа, вершины которого являются точки в пространстве. Характерными задачами, решаемыми в ПСБД, является нахождение пути между двумя вершинами, который удовлетворяет указанным ограничениям.

Как и в обычных БД, в ПСБД выделяют три уровня моделирования: концептуальный, логический и физический.

Задача *концептуальной модели* - адекватно представить множество объектов, их связей, свойств и ограничений. Для этого предлагается использовать пиктографическую ER-модель [40] с нагруженными необходимой информацией вершинами и ребрами. Более совершенной является транспортная модель данных UNETRANS [56].

Логическая модель данных предполагает использование модели конкретной коммерческой СУБД. Обычно использует объектно-реляционную модель. В работе [10] описываются специальные операции на графах, которые используются в ПСБД. Еще одна логическая модель системы GraphDB описана в работе [57].

Физическая модель данных связана с конкретной реализацией ПСБД. На этом уровне решаются задачи структур хранения, методов индексирования и доступа, управление памятью и другие. Здесь используют такие структуры, как сетевой граф, матрица смежности, список смежности [10]. В работе [58] предложен метод доступа CCAM. В статьях [59, 60] обсуждаются вопросы выполнения таких стандартных запросов в ПСБД, как "кратчайший путь", "ближайший сосед", "ближайшие пары". Важным транспортным ограничением для пространственной сети является так называемые "ограничения на выполнение поворотов". Если их не учитывать, то можно строить невыполнимые пути. Вопросу построения путей с учетом ограничений на повороты посвящены статьи [61–64].

Пространственно-временные сетевые БД

Практически все транспортные задачи на сетях зависят от того, на какое время суток они решаются. Другими словами, пространственно-сетевые модели являются зависимыми от времени. В связи с этим для решение транспортных задач следует использовать пространственно-временные модели сетей. В этом направлении также проводятся исследования. Так, например, для моделирования пространственно-временных сетей в статье [65] предлагается создавать копии всей пространственной сети для каждого необходимого момента времени, а в работах [66, 67] связывать со всеми вершинами и ребрами изменяющиеся во времени атрибуты.

Пространственно-временные базы данных (ПВБД)

Они представляют эволюцию во времени пространственных объектов. Такая

эволюция может быть дискретная или непрерывная во времени. В случае непрерывного времени говорят о двигающихся объектах и в связи с этим вводят понятия двигающихся точках, линиях, многоугольниках. Перемещающиеся объекты имеют свои операции, функции и предикаты.

Было предложено несколько подходов для моделирования дискретных изменений пространственных объектов. Один из них - внедрение в темпоральные БД пространственных типов данных. [68]. Другой подход [69] - оставить пространственные объекты как есть, но дополнить каждую из компонент объекта (например, точку или сегмент) темпоральной характеристикой.

Пространственно-временные типы данных позволяют описывать динамическое поведение пространственных объектов во времени [70].

Был предложен пространственно-временной язык запросов STQL [71] на базе SQL. В [7] также предлагается вариант пространственного SQL. В [71–73] предлагаются пространственно-временные предикаты. В работе [74] дается обзор современного состояния исследований по ПВБД.

БД перемещающихся объектов (БДПО)

Это пространственно-временная база данных, предназначенная для фиксации и отслеживания местоположения двигающихся объектов. Исследования по БДПО были инициированы в конце 90-х годов прошлого столетия [70, 75, 76, 77, 78]. Как правило, БДПО используют плоскую пространственно-сетевую модель данных [79, 80].

Два исследования дали жизнь этому направлению. Во-первых, была предложена модель MOST (Moving Objects Spatio-Temporal) [75, 76], которая позволила отслеживать в БД набор зависящих от времени местоположений, например, движение транспортного средства. Было введено понятие динамического атрибута и определен язык запросов FTL (Future Temporal Logic), который позволял специфицировать изменяющиеся во времени взаимосвязи между предполагаемыми местоположениями движущихся объектов. Наконец, были предложены решения по учету неопределенности при отработке запросов.

Вторым важным событием этого времени было открытие в 1996 г. европейского проекта Chorochronos [77], в котором сделана попытка интегрировать концепции пространственных и временных баз данных. В рамках этого проекта для представления перемещающихся объектов была предложена так называемая модель ограничений (constraint model) [81, 82] и разработан прототип DEDALE [78].

Различают два вида БДПО: в первом случае БДПО моделирует, представляет и дает возможность формулировать запросы к предыстории перемещения для проведения последующего пространственно-временного анализа [83, 84]. Вторая разновидность предоставляет возможность моделировать, прогнозировать и запрашивать текущее и будущее перемещение [75, 85]. Во втором случае приходится выбирать между неточностью прогнозных результатов и затратами на обновление БД [76], что приводит к решению задачи управления неопределенностью [86].

Распространенным подходом в исследованиях по БДПО является создание специальных типов данных (перемещающиеся точки и перемещающиеся области (многоугольники), специальных операций и предикатов. Так, например, в [70] предлагаются типы данных, которые позволяют задавать зависимые от времени пространственные объекты и операции над ними. Система типов данных для двигающихся объектов была строго определена в работе [87].

В заключение отметим, что в монографиях [72, 88, 89] детально освещены практически все вопросы, имеющие отношение к двигающимся объектам.

Пространственные СУБД

Многие из распространённых коммерческих СУБД поддерживают работу с пространственными данными.

Среди реляционных СУБД к ним относятся: Oracle Database Spatial, MS SQL Server 2008, DB2 Spatial Extender, Informix Spatial Blade, MySQL Spatial, Spatial Query Server корпорации Boeing, расширение

PostGIS СУБД PostgreSQL, расширение SpatiaLite для SQLite,

Среди NoSQL-систем поддержка пространственных данных реализована в MongoDB, RethinkDB, Cassandra.

Литература

1) Worboys M.F., Duckham M. GIS: a computing perspective. Boca Raton: CRC press; 2004.
2) Open Geospatial Consortium Inc. Date: 2011-05-28 Editor: John R. Herring OpenGIS® Implementation Standard for Geographic information - Simple feature access- Part 1: Common architecture. 93 p.
3) Tomlin C.D. A map algebra. In: Proceedings of the Harvard Computer Graphic Conference; 1983
4) Chan K.K.L., Tomlin C.D. Map Algebra as a Spatial Language. In D. M. Mark and A. U. Frank, editors, Cognitive and Linguistic Aspects ofGeographic Space, pp. 351–360.Kluwer Academic Publishers, Dordrecht, 1991
5) Scholl M., Voisard A. Thematic map modeling. In: Proceedings of the 1st International Symposium on Advances in Spatial Databases; 1989. p. 167–190.
6) Güting R.H. Georelational algebra: a model and query language for geometric database systems. In: Advances in Database Technology, Proceedings of the 1st International Conference on Extending Database Technology; 1988. p. 506–527.
7) Egenhofer M.J. Spatial SQL: a query and presentation language. IEEE Trans Knowl Data Eng. 1994;6(1): 86–95.
8) Güting R.H, Schneider M. Realm-based spatial data types: the ROSE algebra. VLDB J. 1995;4(2):243–286.
9) Cui Z., A.G. Cohn & D.A. Randell, Qualitative and Topological Relationships in Spatial Databases. 3rd Int. Symp. on Advances in Spatial Databases (SSD'93), LNCS 692, 296-315, 1993.
10) Shekar S., Chawla S. Spatial databases: a tour. Englewood Cliffs: Prentice-Hall; 2003
11) The Open Geospatial Consortium Date: 2011-12-19 OGC Reference Model. 44 p.
12) Open Geospatial Consortium Inc. Date: 2010-08-04 Editor: John R. Herring OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option. 111 p.
13) Güting RH. An introduction to spatial database systems. VLDB J. 1994;3(4):357–99.
14) Rigaux P, Scholl M, Voisard A. Spatial databases - with applications to GIS. San Francisco: Morgan Kaufmann Publishers; 2002.
15) Clementini E., Di Felice P. A model for representing topological relationships between complex geometric features in spatial databases. Inf Sci. 1996; 90(1–4):121–136.
16) Schneider M. Spatial data types for database systems - finite resolution geometry for geographic information systems, vol. LNCS 1288. Berlin/New York: Springer; 1997.
17) Schneider M, Behr T. Topological relationships between complex spatial objects. ACM Trans Database Syst. 2006; 31(1): 39–81
18) Worboys M.F, Bofakos P. A canonical model for a class of areal spatial objects. In: Proceedings of the 3rd International Symposium on Advances in Spatial Databases; 1993. p. 36–52.
19) 286) Egenhofer M.J. & R.D. Franzosa, Point-Set Topological Spatial Relations. Int. Journal of Geographical Information Systems, 5(2), 161-174, 1991.
20) 287) Schneider M.,Weinrich B. An abstract model of three dimensional spatial data types. In: Proceedings of the 12th ACM International Symposium on Geographic Information Systems; 2004. p. 67–72.
21) 288) Erwig M., Schneider M. Partition and conquer. In: Proceedings of the third international conference on spatial information theory; 1997. p. 389–408.
22) 289) Güting R.H., Schneider M. Realms: A Foundation for Spatial Data Types in Database Systems. 3rd Int. Symp. on Advances in Spatial Databases, LNCS 692, 14-35, 1993.
23) 292) Freksa C. Using orientation information for qualitative spatial reasoning. In: Proceedings of the International Confer-

ence on Spatial Information Theory; 1992. p. 162–78.
24) 295) Ligozat G. Reasoning about cardinal directions. J Visual Lang Comput. 1998;9(1):23–44.
25) 298) Mukerjee A, Joe G. A qualitative model for space. In: Proceedings of 7th National Conference on AI; 1990. p. 721–727.
26) 299) Papadias D. Relation-based representation of spatial knowledge. PhD Thesis, Department of Electrical and Computer Engineering, National Technical University of Athens; 1994.
27) 293) Goyal R. Similarity assessment for cardinal directions between extended spatial objects. PhD Thesis, Department of Spatial Information Science and Engineering, University of Maine; 2000.
28) 294) Hernández D. Qualitative representation of spatial knowledge. LNCS, vol. 804. Berlin: Springer; 1994.
29) 301) Skiadopoulos S, Koubarakis M. Composing cardinal direction relations. Artif Intell. 2004;152(2):143–71
30) 302) Skiadopoulos S, Koubarakis M. On the consistency of cardinal directions constraints. Artif Intell. 2005;163(1):91–135.
31) Clementini E., Billen R. Modeling and computing ternary projective relations between regions. IEEE Trans Knowl Data Eng. 2006;18(6):799–814.
32) Peuquet D.J., Ci-Xiang Z. An algorithm to determine the directional relationship between arbitrarilyshaped polygons in the plane. Pattern Recognit. 1987;20(1):65–74.
33) Skiadopoulos S, Giannoukos C, Sarkas N, Vassiliadis P, Sellis T, Koubarakis M. Computing and managing cardinal direction relations. IEEE Trans Knowl Data Eng. 2005;17(12):1610–23.
34) Cicerone S., Di Felice P. Cardinal directions between spatial objects: the pairwise-consistency problem. Inf Sci. 2004;164(1–4):165–188.
35) Skiadopoulos S, Sarkas N, Sellis T, Koubarakis M. A family of directional relation models for extended objects. IEEE Transactions on Knowledge and Data Engineering, vol.17, No. 12, 2005, pp 1610–1623
36) Liu W., Li S. Reasoning about cardinal directions between extended objects: the NP-hardness result. Artif Intell. 2011;175(18): 2155–2169.
37) Liu W., Zhang X, Li S., Ying M. Reasoning about cardinal directions between extended objects. Artif Intell. 2010;174(12–13):951–983
38) Shekhar S., Liu X. Direction as a Spatial Object: A Summary of Results. In R. Laurini, K. Makki, and N. Pissinou, editors, ACM-GIS ’98, Proceedings of the 6th international symposium on Advances in Geographic Information Systems, November 6-7, 1998, Washington, DC, USA, pp. 69–75. ACM, 1998.
39) Thanasis Hadzilacos, Nectaria Tryfona. An Extended Entity-Relationship Model for Geographic Applications. ACM SIGMOD Record, Vol. 26, No. 3, 1997, pp. 24–29
40) Shekhar S., Vatsavai R.R., Chawla S., Burke T.E. Spatial pictogram enhanced conceptual data models and their translation to logical data models. In: ISD '99: Selected Papers from the International Workshop on Integrated Spatial Databases, Digital Inages and GIS, 1999 pp. 77–104
41) Gandhi V., Kang J., Shekhar S. Spatial Databases. Technical Report; 07-020, 2007. Retrieved from the University of Minnesota Digital Conservancy, https://hdl.handle.net/11299/215734
42) Frederico T. Fonseca, Max J. Egenhofer. Ontology-driven geographic information systems. In Claudia Bauzer Medeiros, editor, ACM-GIS ’99, Proceedings of the 7th International Symposium on Advances in Geographic Information Systems, November 2-6, 1999, Kansas City, USA, pages 14–19. ACM, 1999.
43) Simon Jonathan David Cox, Chris Little. Time Ontology in OWL. Technical Report, July 2016. - https://www.researchgate.net/publication/305810003_Time_Ontology_in_Owl
44) Bennacer N., Aufaure M.A., Cullot N., Sotnykova A., Vangenot C. (2004). Representing and reasoning for spatiotemporal ontology integration. In R. Meersman, Z. Tari, & A. Corsaro (Eds.), OTM int. conf. on the move to meaningful internet systems (pp. 30–31). Springer.

45) Baglioni M., Masserotti M.V., Renso C., Spinsanti L. (2007). Building geospatial ontologies from geographical databases. In F. Fonseca, M. A. Rodríguez, & S. Levashkin (Eds.), International conference on geospatial semantics (pp. 195–209). Springer.
46) Hogenboom F., Borgman B., Frasincar, F. & Kaymak U. (2010). Spatial knowledge representation on the semantic web. Proceedings of the IEEE 4th International Conference on Semantic Computing (ICSC 2010), pp. 252-259, September 2010.
47) Parent C., Spaccapietra S., Zimányi E. (2006). Conceptual modeling for traditional and spatio-temporal applications: The MADS approach. Springer.
48) Spaccapietra S., Cullot N., Parent C., Vangenot, C (2004). On spatial ontologies. Database Laboratory, Swiss Federal Institute of Technology. 9 p.
49) Egenhofer M.J. Toward the semantic geospatial web. Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems, 2002, pp. 1–4.
50) Fonseca F., Rodriguez M.A. From geopragmatics to derivation ontologies: New directions for the geospatial semantic web. Transactions in GIS, 2007, vol. 11, No. 3, pp. 313-316.
51) Subbiah G., Alam A., Khan L. Thuraisingham B. An integrated platform for secure geospatial information exchange through the semantic web. Proceedings of ACM Workshop on Secure Web Services (SWS), 2006б George Mason University, Fairfax, VA, USA
52) Gaede V., Günther O. Multidimensional access methods. ACM Comput Surv. 1998;30(2):170–231.
53) Venkateswaran J. A Survey of Recent Multidimensional Access Methods. Technical Report, University of Missouri-Rolla. -2004. 162 p.
54) Ahn H.K., Mamoulis N., Wong H.M. A survey on multidimensional access methods. Technical report, Utrecht University (2001)
55) Samet H. Foundations of Multidimensional and Metric Data Structures, Morgan Kaufman Series in Dala Management. Morgan Kaufman Publishers. San Francisco, CA. USA, 2006, 993 p.
56) Curtin K., Noronha V., Goodchild M., Grise S. ARCGIS transportation model (UNETRANS), UNETRANS data model reference, 2003.
57) Guting R.H. GraphDB: modeling and querying graphs in databases. In: VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases, 1994, pp.297–308
58) Shekhar S., Liu D.R. CCAM: a connectivity-clustered access method for networks and network computations. IEEE Trans Knowl Data Eng. 1997;9(1): 102–119.
59) Jensen C.S., Kolar J., Pederson T.B., Timko I. Nearest neighbor queries in road networks. In: GIS '03: Proceedings of the 11th ACM International Symposium on Advances in Geographic Information System; 2003. pp. 1–8
60) Papadias D, Zhang J, Mamoulis N, Tao Y. Query processing in spatial network databases. In: VLDB '03: Proceedings of the 29th international conference on Very large databases - Vol. 29, 2003, pp. 802–813
61) Miller H.J., Shaw S.L. GIS-T data models, geographic information systems for transportation: principles and applications. Oxford: Oxford University Press; 2001.
62) Anez J., de la Barra T., Perez B. Dual graph representation of transport networks. Transp Res. 1996;30(3):209–216.
63) Winter S. Modeling costs of turns in route planning. GeoInformatica. 2002; 6(4):345–361.
64) Hoel E.G., Heng W.L., Honeycutt D. High performance multimodal networks. In: Proceedings of the 9th International Symposium on Advances in Spatial and Temporal Databases; 2005, pp. 308-327.
65) Kohler E., Langtau K., Skutella M. Time-expanded graphs for flow-dependent transit times. In: Proceedings of the 10th Annual European Symposium on Algorithms; 2002, pp. 599–611
66) George B., Shekhar S. Spatio-temporal network databases and routing algorithms: a summary of results. In: Proceedings of the 10th International Symposium on Ad-

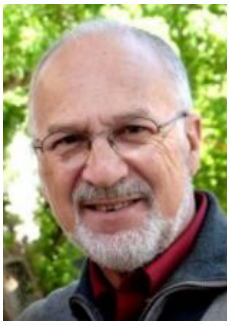
vances in Spatial and Temporal Databases; 2007. p. 460–477.
67) George B., Shekhar S. Time-aggregated graphs for modeling spatio-temporal networks - an extended abstract. In: Proceedings of the 25th International Conference on Conceptual Modeling; 2006 p. 85–99.
68) Tansel A.U, Clifford J., Gadia S., Jajodia S., Segev A., Snodgrass R.T, editors. Temporal databases: theory, design, and implementation. Benjamin-Cummings Publishing Co., 1993, 633 p.
69) Worboys M.F. A unified model for spatial and temporal information. Comput J. 1994;37(1): 25–34.
70) Erwig M., Güting R.H., Schneider M., Vazirgiannis M. Spatio-temporal data types: an approach to modeling and querying moving objects in databases. Geoinformatica. 1999;3(3):265–291.
71) Erwig M., Schneider M. Developments in spatiotemporal query languages. In: Proceedings of the IEEE International Workshop on Spatio-Temporal Data Models and Languages; 1999. p. 441–449.
72) Güting R.H., Schneider M. Moving objects databases. San Francisco: Morgan Kaufmann; 2005
73) Erwig M., Schneider M. Spatio-temporal predicates. IEEE Trans Knowl Data Eng. 2002; 14(4):1–42.
74) Jitkajornwanich K., Pant N., Fouladgar M., Elmasri R. A survey on spatial, temporal, and spatio-temporal database research and an original example of relevant applications using SQL ecosystem and deep learning, Journal of Information and Telecommunication, 2020, 4:4, 524-559,
75) Sistla A. P., Wolfson O., Chamberlain S., Dao S. Modeling and Querying Moving Objects. ICDE '97: Proceedings of the Thirteenth International Conference on Data Engineering, 1997, pp. 422–432
76) Wolfson O., Chamberlain S., Dao S., Jiang L., Mendez G. Cost and imprecision in modeling the position of moving objects. In: Proceedings of the 14th International Conference on Data Engineering; 1998. p. 588–596.
77) Frank A., Grumbach S., Güting R.H., Jensen C.S., Koubarakis M., Lorentzos N., Manolopoulos Y. Chorochronos: a research network for spatiotemporal database systems. ACM SIGMOD Record, Vol. 28I, No. 3,. 1999, pp 12–21
78) Grumbach S., Rigaux Ph., Segoufin L. The DEDALE system for complex spatial queries SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data, 1998 pp. 213–224
79) Vazirgiannis M., Wolfson O. A Spatiotemporal Model and Language for Moving Objects on Road Networks. SSTD '01: Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, 2001, pp. 20–35
80) Güting R.H., Victor Teixeira de Almeida, Zhiming Ding. Modeling and querying moving objects in networks The International Journal on Very Large Data Bases, 2006, vol. 15, No. 2, pp 165–190
81) Belussi A., E. Bertino & B. Catania, Manipulating Spatial Data in Constraint Databases. 5th Int. Symp. on Advances in Spatial Databases (SSD'97), LNCS 1262, 115-141, 1997.
82) Rigaux P., Scholl M., Segoufin L., Grumbach S. Building a constraint-based spatial database system: model, languages, and implementation. Inf Syst. 2003;28(6):563–595.
83) Erwig M., Güting R.H., Schneider M., Vazirgiannis M. Spatio-temporal data types: an approach to modeling and querying moving objects in databases. Geoinformatica. 1999;3(3):265–291.
84) Güting R.H., Böhlen M.H., Erwig M., Jensen C.S., Lorentzos N.A., Schneider M., Vazirgiannis M. A foundation for representing and querying moving objects. ACM Trans Database Syst. 2000;25(1): pp. 1–42.
85) Sistla A.P., Wolfson O., Chamberlain S., Dao S. Querying the uncertain position of moving objects. In: Etzion O, Jajodia S, Sripada S, editors. Temporal databases: research and practice, LNCS, vol. 1399. Berlin: Springer; 1998. p. 310–37.
86) Trajcevski G., Wolfson O., Hinrichs K., Chamberlain S. Managing uncertainty in moving objects databases. ACM Trans Database Syst. 2004;29(3): 463–507.
87) Güting R.H., Böhlen M.H., Erwig M., Jensen C.S., Lorentzos N.A., Schneider M.,

Vazirgiannis M. A foundation for representing and querying moving objects in databases. ACM Trans Database Syst. 2000;25(1):1–42.
88) Pelekis N., Theodoridis Y. Mobility data management and exploration. New York: Springer; 2014.
89) Renso C., Spaccapietra S., Zimányi E. Mobility data: modeling, management, and understanding. Cambridge, UK: Cambridge University Press; 2013.

Дедуктивные базы данных

По мере роста объемов информационных ресурсов все острее встает проблема их понимания и интерпретации, особенно если это относится к сложным предметным областям. Для решения этой проблемы необходимо обладать механизмами поддержки рассуждений с тем, чтобы делать сложные выводы. Для этого начали привлекать математическую логику.


Ашок Чандра


Давид Харел

В конце 70-х гг. начали формироваться подходы по использованию аппарата логики в базах данных [1, 2]. В 1982 г. Чандра и Харел [3] опубликовали статью, которую считают первой работой в области теории дедуктивных баз данных (ДБД). ДБД, как одно из направлений теории баз данных, начали активно развиваться в середине 80-годов прошлого столетия. Исходным пунктом появления и становления ДБД стала теория логического программирования и, в частности, Prolog.

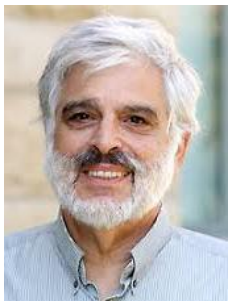
ДБД - это результат объединения логического программирования с реляционными базами данных. ДБД более выразительны, чем реляционные базы данных, но менее выразительные, чем системы логического программирования. ДБД - это система баз данных, которая может делать выводы на основе правил и фактов хранящихся в базе данных. ДБД представляется как база фактов и база правил. Первая из них в теории ДБД называется экстенсиональной базой данных (ЭБД), а вторая – интенсиональной базой данных (ИБД). ЭБД представляется в виде реляционных отношений, а ИБД представляет собой подмножество Prolog: без функциональных символов и специальных предикатов типа cut и var. ИБД - это множество правил, которые с логической точки зрения представляются в виде хорновских дизъюнктов. Такие правила имеют форму "если A то B", где A называется "те-

лом", В - "головой". Тело состоит из конъюнкции литералов (подцелей). Литерал - это атом или его отрицание. Атом - это предикат, содержащий переменные или константы.

Во многих статьях и энциклопедиях понятие ДБД и язык Datalog рассматриваются как синонимы. Считается, что самое раннее опубликованное упоминание термина "Datalog " было сделано в 1985 г. в рукописи [4]. Затем термин "Datalog" использовался в работах [5, 6] Однако все же принято считать, что официально язык Datalog впервые был исследован в книге Майера (David Maier) и Уоррена (David S. Warren) [7] как упрощенный вариант Prolog без функциональных символов. Авторы это название объяснили тем, что предикаты без функциональных символов напоминают отношения базы данных. Отметим, то в 1985 г. термин Datalog также был использован в качестве языка запросов к базе данных в системе, поддерживающей естественный язык [8]. Это термин был взят в качестве сокращения от “database dialogue" и никакого отношения к Prolog не имел.


Давид Майер

В любом случае, к середине 80-х годов термин Datalog утвердился как язык дедуктивных баз данных, а не как язык логического программирования. В 1989 г. была опубликована замечательная монография Джеффри Ульмана (Jeffrey D. Ullman) [2], в которой отдельная глава посвящена детальному изложению сущности Datalog как логической модели данных. Многое из этой главы используется далее при изложении принципов Datalog.


Джеффри Ульман

Безопасное правило. Чтобы Datalog-правила интерпретировались операциями над конечными отношениями, переменные правила должны быть ограничены, то есть они должны содержаться в по крайней мере одном атоме (литерале без отрицания). Правило безопасно, если все его переменные ограничены На этот аспект было обращено внимание в работах [9, 10].

Различают три вида правил:

– простые - без рекурсии и отрицаний;
– рекурсивные - содержащие рекурсивные определения;
– с отрицаниями - содержащие атомарные формулы с отрицаниями.

Простые правила. Для простых правил существует способ их преобразования в выражения реляционной алгебры (см., например, [2]). Полученные выражения определяют отношения для предикатов ИБД и представляют собой единственную минимальную модель.

Рекурсивные правила. Для рекурсивных Datalog -программ, которые не содержат отрицаемых подцелей, существует единственная минимальная модель, которая содержит заданные ЭБД-отношения, и эта модель является единственной минимальной неподвижной точкой относительно ЭБД-отношений соответствующих Datalog-правил. Известно, что если функция монотонна, то ее рекурсивное вычисление приводит к неподвижной точке. В реляционной алгебре все операции, за исключением разности, являются монотонными. Поэтому в реляционной алгебре, расширенной циклами можно построить алгоритм, вычисляющий наименьшую неподвижную точку рекурсивной Datalog –программы. В [2, 11] даны обзоры по оптимизации рекурсивных запросов в ДБД.

Правила с отрицаниями. Правила могут содержать в теле отрицаемые подцели. В этом случае возникают две проблемы. Во-первых, отрицания могут порождать бесконечные интерпретации. Поэтому было расширено требование существования безопасных правил с отрицаниями - переменные, которые встречаются в отрицаемых подцелях, должны обязательно присутствовать в подцелях без отрицаний. Вторая проблема связана с тем, что при наличии отрицаний Datalog-программа может иметь множество минимальных моделей (минимальных неподвижных точек) В этом случае смысл программы с отрицаниями задается выбором некоторой "предпочитаемой" модели [12-19]. Отрицания также вызывают

проблемы в рекурсии. В связи с этим было введено понятие стратифицированного отрицания [12, 13, 20, 21] и стратифицированных программ, которые имеют интуитивно понятную семантику [22-25]. Стратификация не гарантирует существование наименьшей неподвижной точки. Однако это ограничение дает возможность выбора среди множества минимальных неподвижных точек такой "предпочитаемой", которая будет интерпретацией смысла Dtalog-программы. Были также определены более специфические классы стратифицируемых программ, например, локально стратифицированные программы [26] модульно стратифицированные программы [27].

Оптимизация. Одной из наиболее сложных проблем создания ДБД является оптимизация. Для не рекурсивных правил проблема оптимизации аналогична традиционной реляционной оптимизации, а при наличии рекурсии и отрицания возникают дополнительные проблемы и возможности. Были проведены многочисленные исследования в этом направлении. Среди них можно отметить метод магических множеств (magic-sets) [2], а также ряд базирующихся на нем методов [28], алгоритм подсчета (counting algorithm) [29], факторинговая оптимизация (factoring optimization) [30], метод удаления избыточных правил и литералов [31], метод оптимизации экзистенциальных запросов [32], метод "конвертов" (envelopes) [33] и другие. В работе [34] дается аналитический обзор различных стратегий оптимизации и сравнительный анализ их производительности.

Дедуктивные системы баз данных. Что касается создания дедуктивных систем баз данных, то можно выделить два направления. С одной стороны проводились исследования, открывались проекты и разработки по созданию самостоятельных систем, с обзором которых можно познакомиться в [35, 36]. С другой стороны, в 1999 г. в очередную версию SQL (SQL99) была ведена возможность формулировать и выполнять рекурсивные запросы. Как мы уже отмечали, простые правила и специальные правила с отрицаниями полностью выразимы в реляционной алгебре, а значит, и в стандартном (без рекурсии) SQL. Включение рекурсии в SQL привело к тому, что все возможности дедуктивных Datalog-программ стали выразимы в SQL99. Дополнительно предоставляются возможности указывать направление поиска (в ширину, в глубину), фиксации и предотвращения бесконечных циклов, создавать рекурсивные представления, определять прямую и взаимную рекурсию, линейную и нелинейную рекурсию. С рабочим вариантом стандарта рекурсивного SQL можно познакомиться в [37], а с кратким описанием практического использования - в работе [38]

Следует также отметить, что к концу 80-х годов сформировалось направление объектно-дедуктивных баз данных. Для них были разработаны специальные языки, такие как O-Logic, F-Logic, ROL, IQL [39, 40]. В статье [41] дается аналитический обзор объектно-дедуктивных баз данных.

Литература

1) Gallaire H., Minker J., eds. Logic and Databases. New York: Plenum. 1978.
2) Ullman J.D. Principles of Database and Knowledge-Based Systems.Maryland: Computer Sciences Press Inc., 1989
3) Chandra A.K., Harel D. Horn clauses and the fixpoint hierarchy Proc. ACM Symp. on the Principles of Database Systems (PODS) (1982), pp. 158-163
4) Porter H.H., Oct. 1985. Optimizations to Earley deduction for DATALOG programs. - www.cs.pdx.edu/~harry/earley/datalog.pdf
5) Afrati C.H., Papadimitriou Ch., Papageorgiou G., Roussou A., Sagiv Y, Ullman J.D. 1986. Convergence of sideways query evaluation. In ACM Symposium on Principles of Database Systems, pp. 24–30
6) Bancilhon, R. Ramakrishnan. 1986. An amateur’s introduction to recursive query processing strategies. In Proc. of the 1986 ACM SIGMOD International Conference on Management of Data, SIGMOD ’86, pp. 16–52.
7) Maier D., Warren D.S. 1988. Computing with Logic: Logic Programming with Prolog. Benjamin-Cummings Publishing Co., Inc.Subs. of Addison-Wesley Longman Publ. Co390 Bridge Pkwy. Redwood City, CA United States. 535 p.

8) Hafner C.D., Godden K. Portability of syntax and semantics in DATALOG. ACM Trans. on Information Systems, 1985, 3(2):141–164.
9) Zaniolo, C. [1986]. "Safety and compilation of nonrecursive Horn clauses," Proc. First Intl. Conf. on Expert Database Systems, pp. 167-178, Benjamin-Cummings, Menlo Park, CA.
10) Ramakrishnan R., Bancilhon F., Silberschatz A. [1987]. "Safety of recur sive Horn clauses with infinite relations," Proc. 5ixth ACM Symp. on Principles of Database Systems, pp. 328-339.
11 Bancilhon F., Ramakrishnan R. An amateur's introduction to recursive query processing strategies. SIGMOD Record, v. 15, no.2, 1986, pp. 16-52
12 Apt K.R., Blair H., Walker A., Towards a Theory of Declarative Knowledge, in: J. Minker (ed.), Foundations of Deductive Databases and Logic Programming, Morgan Kaufmann, San Mateo, CA, 1988, pp. 89-148.
13) Chandra A.K., Harel D. Horn Clause Queries and Generalizations, J. Logic Programming 2(1):1-15 (Apr. 1985).
14) Gelfond M., Lifschitz V. The Stable Model Semantics for Logic Programming, in: Proceedings of the Fifth International Conference and Symposium on Logic Programming, 1988.
15) Przymusinska H., Przymusinski T.C. Weakly Perfect Model Semantics for Logic Programs, in: Proceedings of the Fifth International Conference/Symposium on Logic Programming, 1988.
16) Przymusinski, T.C. On the Declarative Semantics of Stratified Deductive Databases in: J. Minker (ed.), Foundations of Deductive Databases and Logic Programming, 1988, pp. 193-216.
17) Przymusinski T.C. Extended Stable-Semantics for Normal and Disjunctive Programs, in: Seventh International Conference on Logic Programming, 1990, pp. 459-477.
18) Ross K. Modular Stratification and Magic Sets for DATALOG Programs with Negation, in: Proceedings of the ACM Symposium on Principles of Database Systerns, 1990, pp. 161-171.
19) Van Gelder A., Ross K., Schlipf J.S. The Well-Founded Semantics for General Logic Programs, Journal of he ACM 38(3):620- 650 (1991)
20) Naqvi S. A Logic for Negation in Database Systems, in: J. Minker (ed.), Proceedings of the Workshop on Foundations of Deductive Databases and Logic Programming, 1986, pp. 378-387.
21) Van Gelder A. Negation as Failure Using Tight Derivations for General Logic Programs, Journal of Logic Programming 6(1):109-133 (1989).
22) Balbin I., Port G.S., Ramamohanarao K., Meenakshi K. Efficient Bottom-Up Computation of Queries of Stratified Databases, Journal of Logic Programming 11:295-345 (1991).
23) Bayer R. Query Evaluation and Recursion in Deductive Database Systems, Technical Report 18503, Teehnisehe Universitaet Muenchen, Feb. 1985.
24) Beeri C., Naqvi S., Ramakrishnan R., Shmueli O., Tsur S. Sets and Negation in a Logic Database Language, in: Proceedings of the ACM Symposium on Principles
25) Kerisit J.M., Pugin J.M. Efficient Query Answering on Stratified Databases, in: Proceedings of the International Conference on Fifth Generation Computer Systems, Tokyo, Japan, Nov. 1988, pp. 719-725.
26) Przymusinski T. On the Declarative Semantics of Stratified Deductive Databases, in J. Minker (ed.), Foundations of Deductive Databases and Logic Programming, 193-216, Morgan-Kaufmann, Los Altos, 1988.
27) Ross K.A. Modular Stratification and Magic Sets for Datalog Programs with Negation. Proceedings of the ACM Symposium on Principles of Database Systems, 161-171, Nashville, 1990.
28) Warren D.S. Memoing for Logic Programs, Communzcations of the ACM 35 (3): 93-111 (Mar. 1992)
29) Sacca D., Zaniolo C. The Generalized Counting Methods for Recursive Logic Queries, in: Proceedings of the First International Conference on Database Theory, 1986.

30) Naughton J.F., Ramakrishnan R., Sagiv Y., Ullman J.D. Argument Reduction Through Factoring, in: Proceedings of the Fifteenth International Conference on Very Large Databases, Amsterdam, The Netherlands, Aug. 1989, pp. 173-182.
31) Sagiv Y., Optimizing Datalog Programs, in: J. Minker (ed.), Foundations of Deductive Databases and Logic Programming, Los Altos, CA, Morgan Kaufmann, 1988, pp. 659-698.
32) Ramakrishnan R., Beeri C., Krishnamurthy R. Optimizing Existential Datalog Queries, in: Proceedings of the ACM Symposium on Principles of Database Systems, Austin~ TX, Mar. 1988, pp. 89-102.
33) Sippu S., Soisalon-Soinen E. An Optimization Strategy for Recursive Queries in Logic Databases, in: Proceedings of the Fourth International Conference on Data Engineering, Los Angeles, CA, 1988.
34) Bancilhon F., Ramakrishnan R. An amateur's introduction to recursive query processing strategies. SIGMOD Record, Vol. 15, No.2, 1986, pp. 16-52
35) Gallaire H., Minker J. and Nikolas J.M. Logic and databases: a deductive approach. Computing Surveys, 16:1, 1984, pp. 154-185
36) Ramakrishnan R., Ullman J.D. A survey of deductive database systems. The Journal of Logic Programming, 1995, Vol. 23, No 2, pp. 125-149
37) Finkelstein S.J., Mattos N., Mumick I., Pirahesh H. Expressing Recursive Queries in SQL SO/IEC JTC1/SC21 WG3 DBL MCI-X3H2-96-075 Tech. Rep., March, 1996
38) Reznichenko V.A. Recursive SQL (Rus). Software Engineering, 2010, vol. 4, No 4, pp. 48-65.
39) Kifer M., Lausen G. F-Logic: A Higher Order Language for Reasoning about Objects, Inheritance, and Schema. SIGMOD Record, v. 18, no.2,1989, pp. 139- 146.
40) Liu M. Deductive Database Languages: Problems and Solutions. ACM Computing Surveys, v. 31, no. 1, 1999. pp. 27-62
41) Falcone Sampaio P.R., Paton N.W. (1997) Deductive objectoriented database systems: A survey. In: Geppert A., Berndtsson M. (eds) Rules in Database Systems. RIDS 1997. Lecture Notes in Computer Science, vol 1312. Springer, Berlin, Heidelberg. pp 1-19

Активные базы данных

Традиционные базы данных являются пассивными. Данные помещаются, обновляются, переносятся и выбираются из БД под воздействием внешних источников (человек или программа). Бизнес-правила, применяемые к содержимому базы данных, также, как правило, управляются внешними источниками. Короче говоря, традиционные базы данных не являются активными участниками функционирования информационной системы и обеспечивают только функцию хранения данных. Для преодоления этого недостатка была введена концепция активных баз данных.

Активная база данных (АБД) - *это база данных, по отношению к которой СУБД выполняет не только те действия, которые явно указывает пользователь, но и дополнительные действия в соответствии с правилами, заложенными в саму БД.*

Зарождение идей АБД связывают с появлением концепции триггера - механизма, впервые предложенного в исследовательском проекте System R компании IBM. Поддержка концепции триггера предусматривалась в языке этой системы SEQUEL. Однако, следует отметить, что идея триггера ранее была воплощена в языке определения данных CODASYL [1, 2] (хотя сам термин "триггер" еще не использовался). В языке предусматривалась поддержка концепции процедуры базы данных, которая может ассоциироваться с различными объектами базы данных в спецификации схемы. Процедура базы данных запускается автоматически в случае, если над объектом, с которым она ассоциирована, выполняется одна из приведенных в спецификации операций. При этом выполнение процедуры может предшествовать выполнению заданной операции, следовать за ней или иметь место в случае возникновения ошибки.

АБД должна предусматривать поддержку следующих возможностей:

- содержать логику обработки данных (бизнес-правила) в самой базе данных с тем, чтобы она управлялась через СУБД, а не прикладными программами или пользователями;
- обеспечить мониторинг событий и условий, которые воздействуют на данные и могут инициировать обработку данных, управляемую СУБД;
- содержать средство, с помощью которого эти события и условия могли бы запускать логику обработки данных внутри базы данных.

ECA-правило

Для поддержки перечисленных выше возможностей в активной базе данных было ведено понятие ECA-правила - это конструкция, включающая три составляющих: событие, условие и действие (Event-Condition-Action). Впервые оно было определено в проекте HiPAC (High Performance ACtive database system) [3]. Семантика правила простая: при наступлении события проверяется условие и если оно истинно, выполняется действие.

Условием ECA-правила могут быть: запрос к базе данных, логическое выражение, вызов подпрограммы (процедуры или функции, возвращающей логическое значение).

Действие ECA-правила - произвольный код, вызываемый при наступлении события и при истинности условия. Действие выполняется либо в виде составной части транзакции ECA-правила, либо в виде самостоятельной транзакции в зависимости от режима связывания. В пределах одного ECA-правила могут выполняться несколько действий одновременно, поэтому следует отслеживать конфликтные ситуации. Тело действия может инициировать события, которые вызывают выполнение другого правила и.т.д. Наконец, цепочка последовательно инициируемых вложенных правил может быть рекурсивной.

Модели ECA-правил

Для описания ECA-правил были предложены две модели: *модель знаний* (knowledge model) и *модель исполнения* (execution model) [3, 4]. Модель знаний описывает, что собой представляют активные правила, а модель исполнения специфицирует, каким образом интерпретируются ECA-правила в процессе их выполнения.

Модель знаний ECA-правил

В работе [5] были обобщены и классифицированы характеристики модели знаний, которые ранее были представлены в литературе [6–8]. Эти характеристики относятся ко всем составляющим ECA-правил и они приводятся далее.

Характеристики модели знаний события

К ним относятся следующие:

- *Источники событий* (event sources): операции над структурными элементами базы данных (structure operations), поведение внешней среды (behaviour invocation - действия пользователей или прикладных программ), команды транзакций (begin, abort, rollback commit), временные характеристики;
- *Обнаружение событий* - это процесс анализа потока событий для выявления событий, соответствующих заданному образцу. Обычно выявление событий включает процедуры фильтрации и агрегации. Основополагающие исследования по выявлению событий были проведены при выполнении проектов HiPAC [3, 9], Snoop [10, 11], ODE [12], SAMOS [13]. В [14] дается обзор исследований по обнаружению событий, а в статье [15] - спецификаций событий.
- *Гранулярность событий* (event granularity) - указывает, определяется ли событие для каждого объекта из множества, или изданного подмножества или конкретных объектов множества.
- *Простые и составные события.* Событие, объединяющее в себе несколько событий, называется составным. Пионерской работой в области составных событий считается проект HiPAC [3]. Для описания составных событий в рамках разработанных систем были определены различные алгебры, например, [7, 10, 11, 13, 16]. В работе [10] для составных событий введена характеристика "политика потребления" (consumption policy), определяющая ситуацию, в которой считается, что составное событие произошло. В работе [17] предлагается общий метод и язык EPL спецификации семантики составных событий.

Характеристики модели знаний условия

К ним относятся следующие:

- *факультативность* - является ли условие обязательным в ECA-правиле, или нет. Считается, что в правиле должно присутствовать либо событие, либо условие.
- *контекст* - в целом в качестве контекста ECA-правила предлагаются следующие варианты: состояние базы при: запуске транзакции (DBT), инициировании события (DBE), проверке условия (DBC), выполнении действия (DBA), а также привязка условия к событию (BindE) и действия к условию (BindC). В качестве контекста условия выступают DBT, DBE, DBC и BindE.

Характеристики модели знаний действия

К ним относятся следующие:

- *виды действий*. Предлагаются следующие: работа со структурой базы данных, инициирование внешней среды, информирование, аварийное завершение, выполнение иного действия чем то, что инициировало событие ("do-instead" Стоунбрейкера [6])
- *контекст* - аналогично контексту условия специфицирует, какие именно данные доступны действию. Допустимыми значениями являются DBT, DBE, DBC, DBA и BindC

Модель исполнения ECA-правил

Модель исполнения (execution model) специфицирует, каким образом трактуются ECA-правила в процессе их выполнения [4]. Она касается событий, условий и действий ECA-правил.

Модель исполнения событий

Если активная база данных поддерживает выявление составных событий, то необходимы правила их обнаружения и отработки. Для решения этой ситуации были предложены так называемые "режимы потребления событий" (event consumption modes) [10, 16, 18].

Далее описываются характеристики модели исполнения условий и действий ECA-правил.

Правила проверки условий и выполнения действий

Режимы связывания (Coupling modes). Режимы связывания определяют, как инициируется проверка условия в ответ на происшедшее событие и как планируется, диспетчируется и выполняется действие ECA-правила при положительно результате проверки условия.. Они впервые были исследованы в проекте HiPAC [9]. Они определяются для пар событие-условие и условие-действие. Связывание событие-условие определяет, когда следует проверить условие относительно события, а условие-действие - когда следует выполнить действие относительно условия, Для обоих видов связываний были предложены одни и те же варианты: связывания:

- *немедленно* (immediately) - условие/действие проверяется/выполняется сразу же после события/условия
- *отсрочено* (deferred) - проверка/выполнение условия/действия откладывается до завершения транзакции (до выполнения Commit), в которой инициирован триггер. Также были предложены варианты, когда отсрочка задавалась определяемым пользователем временем отсрочки [19] или выполнением специальных команд [8].
- *отдельно* (detached) - проверка/выполнение условия/действия производится в другой транзакции, чем событие/условие, причем выполнение действия может зависеть или быть независимым от завершения транзакции, в которой произошло событие или было проверено условие.

В работе [9] было установлено, что в триггере не все варианты пар значений режимов связывания являются допустимыми. Также отметим, что в исследовательском проекте REACH (REal-time ACtive Heterogeneous System) [20] были предложены еще два варианта режима связывания для поддержки побочных эффектов необратимых действий ECA-правил.

Запуск событием нескольких правил

Возможна ситуация, когда событие инициирует запуск нескольких правил. В этом случае были предложены механизмы планирования порядка выполнения правил [4, 6, 21].

Политика итогового эффекта

Для случая, когда в пределах одного правила выполняется несколько действий относительно одних и тех же данных предложна политика итогового эффекта (net effect policy) [4], когда выполняется только одно действие или даже не выполняется ни одного действия.

Вызов правилом другого правила

Правило может инициировать вызов другого правила и т.д. При этом возникают ситуации, когда выполнение внутреннего правила противоречит выполнению внешнего правила, а также возможны циклы, когда правило инициирует выполнение самого себя. Эти ситуации также рассматриваются в литературе [4].

Системы активных баз данных

Разработаны системы активных реляционных баз данных (РБД) и объектно-ориентированных баз данных (ООБД).

Активные РБД

Включение активных механизмов в РБД не является чем-то новым, подавляющее большинство коммерческих систем поддерживают механизмы триггеров. Кроме того, проводятся исследования по разработке более развитых средств поддержки активных правил. Предложения по включению активного поведения в реляционные системы, как правило, ограничиваются возможностями традиционных пассивных реляционных систем. Например, событиями, которые инициируют правила, являются только операции над данными (вставка, удаление, замена). Как правило, в реляционных системах не рассматриваются составные события, они не обладают развитыми режимами связывания и язык описания правил встраивается в язык запросов. Среди "ранних" реляционных систем, которые обладали механизмами активизации, можно отметить Starburst [8, 22], Postgress [23], Ariel [24]. Примерами активного рас-

ширения реляционной модели данных являются работы [25–31]. Среди них особый интерес представляют те, которые исследуют взаимосвязь активных и дедуктивных баз данных [27–31].

В стандарт SQL 1999 г. были включены триггеры [32]. С тех пор все промышленные реляционные СУБД в качестве механизма активных правил включают как минимум триггеры SQL.

Активные ООБД

Что касается ООБД, то в отличие от реляционных они всегда поддерживали тесную связь между поведением пользователей и данными БД. Такое поведение представляется методами, приписываемыми классам данных БД Этот факт, а также инкапсуляция структуры объекта указывают, что некоторые аспекты, которые могут быть представлены в РБД с помощью активного поведения, в ООБД поддерживаются с помощью методов. Тем не менее, исследования по активному расширению ООБД начались практически одновременно с появлением самих ООБД и их существенное отличие заключатся в том, что в активных ООБД примитивные события часто ассоциируются с вызовами методов, а не с операциями над структурными элементами БД. Разработано множество систем активных ООБД, среди которых можно отметить HiPAC [3, 7, 33], EXACT [19], NAOS [34], Chimera [35], Ode [36], SAMOS [13], Sentinel [18], REACH [20]. С кратким описанием этих систем можно познакомиться в [4], в этой же работе имеются ссылки на другие работы по активному расширению ООБД.

Варианты использование ECA-правил

Активные правила могут использоваться для расширения различных функциональных возможностей БД. Например, ECA-правила используются для поддержки ограничений целостности [37, 38], материализованных представлений [39, 40], производных данных [41], координации распределенных вычислений [42, 43], моделей транзакций [44], развитых возможностей моделирования данных [45], автоматического обновления экрана при изменении базы данных [46, 47].

Система HiPAC

В заключение отметим, что HiPAC стала одной из лидирующих систем активных баз данных своего времени и единственной, которая ориентировалась на потребности приложений реального времени, что привело к созданию инновационной модели триггера. Модель ECA-правил, представленная в HiPAC, теперь широко применяется в активных вычислительных системах, в системах обработки сложных событий и в распределенных системах. Руководитель проекта HiPAC Умешвар Дайал (Umeshwar Dayal) в 2010 году стал лауреатом инновационной премии SIGMOD имени Эдгара Ф. Кодда за пионерские работы и существенный вклад в распределенные гетерогенные базы данных, высокопроизводительные активные базы данных, модели долговременных транзакций и исследования в области бизнес-процессов.


Умешвар Дайал

Литература

1) CODASYL/Data Description Language Committee (DDLC), "June 73 Report". CODASYL Data Description Language Committee Journal of Development, June 1973
2) "CODASYL Data Description Language Committee Journal of Development", 1978.
3) Chakravarthy S, Blaustein B, Buchmann A.P, Carey M, Dayal U, Goldhirsch D, Hsu M, Jauhuri R. Ladin R, Livny M, McCarthy D, McKee R, Rosenthal A. HiPAC: a research project in active, time-constrained database management. Technical report. CCA-88-02. Cambridge, MA: Xerox Advanced Information Technology; 1988
4) Paton N.W., Díaz O. Active database systems. ACM Computing Surveys. 1999, vol. 31, No 1, pp. 63–103
5) Paton N., Diaz O., Williams M., Campin J., Dinn A., Jaime A. Dimensions of active behaviour. In N. Paton and M. Williams

Eds., Proc. 1st Int. Workshop on Rules In Database Systems, Springer-Verlag., 1994, pp. 40-57.
6) Stonebraker M., Jhingran A., Goh J., Potamianos S. On rules, procedures, caching and views in database systems. In Proc. ACM SIGMOD 1990, pp. 281-290
7) Dayal U., Buchmann A., McCarthy D. Rules are objects too: A knowledge model for an active object oriented database system. In K. Dittrich Ed., Proc. 2nd Inti Workshop on OODBS, Volume 334, 1988, pp. 129-143. Springer-Verlag. Lecture Notes in Computer Science
8) Widom J., Finkelstein S. Set-Oriented Production Rules in Relational Database Systems. In Proceedings of the ACM SIGMOD International Conference on Management of Data,1990, pp. 259-270.
9) Dayal U, Blaustein B, Buchmann A, Chakravarthy S, Hsu M, Ladin R, McCarty D, Rosenthal A, Sarin S, Carey M.J, Livny M, Jauhari R. The HiPAC project: combining active databases and timing constraints. ACM SIGMOD Rec. 1988;17(1):51–70.
10) Chakravarthy S, Krishnaprasad V, Anwar E, Kim S.K Composite events for active database: semantics, contexts, and detection. In: Proceedings of the 20th International Conference on Very Large Data Bases; 1994. p. 606–617.
11) Chakravarthy S, Mishra D. Snoop: an expressive event specification language for active databases. Data Knowl Eng. 1994;14(1):1–26.
12) Gehani N.H., Jagadish H.V., Schmueli O. Gehani N., Jagadish H.V., Shmueli O. COMPOSE: A system for composite specification and detection. In: Adam N.R., Bhargava B.K. (eds) Advanced Database Systems. 1993, pp. 3-15. Lecture Notes in Computer Science, vol 759. Springer, Berlin.
13) Gatziu S., Dittrich K. Events in an active object-oriented database. In N. Paton and M. Williams Eds., Proc. 1st Int. Workshop on Rules in Database Systems, 1994, pp. 23-39. Springer-Verlag
14) Mellin J., Berndtsson M. Event Detection. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 1361-1366. Springer, New York, 2018
15) Mellin J., Berndtsson M. Event Specification. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 1389-1393. Springer, New York, 2018
16) Gehani N., Jagadish H.V., Smueli O. Event specification in an active object-oriented database. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 1992. p. 81–90
17) Motakis I., Zaniolo C. Composite Temporal Events in Active Databases: A Formal Semantics. In: Clifford J., Tuzhilin A. (eds) Recent Advances in Temporal Databases. Workshops in Computing. Springer, London, 1995), 332- 352
18) Chakravarthy S., Anwar E., Maugis L., Mishra D. Design of Sentinel: an object-oriented DBMS with event-based rules. Information and Software Technology, 1994, 36, 9, 555-568.
19) Diaz O., Jaime, A. EXACT: an EXtensible approach to ACTive object-oriented databases. VLDB Journal 1997,.6, 4, 282-295
20) Branding H, Buchmann A, Kudrass T, Zimmermann J. Rules in an open system: the REACH rule system. In: Proceedings of the 1st International Workshop on Rules in Database Systems, Workshops in Computing; 1994. p. 111–126
21) Agrawal R., Cochrane R., Lindsay B. On maintaining priorities in a production rule language. In G. Lohman, A. Sernadas, and R. Camps Eds., Proc. 17th VLDB, 1991, pp. 479-487. Morgan-Kaufmann.
22) Widom J. The Starburst Rule System: Language Design, Implementation, and Applications. In: IEEE Data Engineering Bul letin, Special Issue on Active Databases, 1992, 15(4): 15-18
23) Stonebraker M., Kemnitz, G. The POSTGRES Next generation Database Management System. Communications of the ACM 1991, Vol. 34, No.10, pp. 78 92
24) Hanson E.N. The Design and Implementation of the Ariel Active Database Rule System. IEEE Trans. Knowl. Data Eng. 1996, 8(1): 157-172
25) Kotz A., Dittrich K., Mulle J. Supporting semantic rules by a generalized event/trigger mechanism. In Advance in Database Technology, EDDT, Venicеб 1988, pp. 76 91.

26) Reddi S., Poulovassilis A., Small C. Extending a Functional DBPL With ECA-Rules. In T. Sellis Ed., Proc. 2nd Int. Wshp. on Rules in Database Systems 1995, pp. 101-115. Springer-Verlag.
27) Kiernan G., de Maindreville C., Simon E. Making Deductive Databases a Practical Technology: a step forward. In II. Garcia-Molina and II. Jagadish Eds., Proc. ACM SIGMOD Conf. 1990., pp. 237-246
28) Zaniolo C. A Unified Semantics for Active and Deductive Databases. In: Paton N.W., Williams M.H. (eds) Rules in Database Systems. Workshops in Computing. Springer, London. 1994, pp 271-287
29) Harrison J., Dietrich. S. Integrating active and deductive rules. In N. Paton and M. Williams Eds., Proc. 1st Int. Workshop on Rules In Database Systems, 1994, pp. 288 305. Springer-Verlag.
30) Widom J. Deductive and Active Databases: Two Paradigms or Ends of a Spectrum? In N. Paton and M. Williams Eds., Proc. 1st Int. Workshop on Rules In Database Systems 1994, pp. 306 315. Springer-Verlag.
31) Bayer P., Jonker W. A framework for supporting triggers in deductive databases. In N. Paton and M. Williams Eds., Proc. 1st Int. Workshop on Rules In Database Systems 1994, pp. 316 330. Springer-Verlag.
32) Kulkarni K., Mattos N., Cochrane R. Active Database Features in SQL3. In: Paton N.W. (eds) Active Rules in Database Systems. 1999, pp. 197-219 Monographs in Computer Science. Springer, New York, NY.
33) Chakravarthy S. Rule management and evaluation: an active DBMS perspective. SIGMOD RECORD 1989, 18, 3, 20-28.
34) Collet C., Coupaye T. and Svensen T. NAOS: Efficient and modular reactive capabilities in an object-oriented database system. In J. Bocca, M. Jarke, and C. Zaniolo Eds., Proc. 20th VLDD Conf, 1994, pp. 132 -143. Morgan-Kaufmann.
35) Ceri S., Fraternali P., Parabosciii S., Tanca, L. Active Rule Management in Chimera. In J. Widom and S. Ceri Eds., Active Database Systems: Triggers and Rules for Active Database Processing, 1996, pp. 151-175. Morgan Kaufmann.
36) Gehani N. and Jagadish H. ODE as an Active Database: Constraints and Triggers. In R. C. G.M. Loiiman. A. Sernadas Ed., 17th Intl. Conf. on Very Large Data Bases, Barcelona,1991, pp. 327-336. Morgan Kaufmann
37) Ceri S., Gottlob G., Tanca L. 1990. Logic Programming and Databases, Springer-Verlag, Berlin.
38) Diaz O. Deriving rules for constraint maintenance in an object-oriented database. In Proceedings of the International Conference on Databases and Expert Systems DEXA, I. R. A. M. Tjoa, Ed., Springer-Verlag, 1992, pp. 332–337.
39) Stonebraker M., Jhingran A., Goh J., Potamianos S. On rules, procedures, caching and views in database systems. In Proceedings of ACM SIGMOD, 1990., pp. 281–290.
40) Widom J., Cochrane R., Lindsay B. Implementing set-oriented production rules as an extension to Starburst. In Proceedings of the Seventeenth International Conference on Very Large Data Bases (Barcelona), R. C. G. M. Lohman and A. Sernadas, Eds., Morgan-Kaufmann, San Mateo, CA, 1991, pp. 275–286.
41) Etzion O. PARDES—a data-driven oriented active database model. ACM SIGMOD Record, 1993, Vol. 22, No. 1, pp, 7–14.
42) Dayal U., Hsu M., Landin R. Organising long-running activities with triggers and transactions. In Proceedings of the SIGMOD Conference, ACM, New York, 1990, 204–214.
43) Ceri S., Widom J. Managing semantic heterogeneity with production rules and persistent queries. In Proceedings of the Nineteenth International Conference on Very Large Data Bases, R. Agrawal, S. Baker, and D. Bell, Eds., Morgan-Kaufmann, San Mateo, CA, 1993, 108–119.
44) Geppert A., Dittrich K. Rule-based implementation of transaction model specifications. In Proceedings of the First International Workshop on Rules in Database Systems, N. Paton and M. Williams, Eds., Springer-Verlag, 1994, 127–142.
45) Paton et al. 1993] Paton N., Diaz O., Barja M. Combining active rules and metaclasses for enhanced extensibility in object-

oriented systems. Data Knowl. Eng. 1993, 10, 45–63.
46) Diaz et al. 1994] Diaz O., Jaime A., Paton N., Al Qaimari G. Supporting dynamic displays using active rules. ACM SIGMOD Record, 1994, Vol. 23, No. 1, pp. 21–26.
47) Paton et al. 1996]. Paton N., Doan D., Diaz O. Jaime A. Exploitation of object-oriented and active constructs in database interface development. In Proceedings of the Third International Workshop on Interfaces to Database Systems, J. Kennedy and P. Barclay, Eds., Springer-Verlag. 1996.

Объектные базы данных

Возникновение направления объектных баз данных (ОБД) определялось, прежде всего, потребностями практики: необходимостью разработки сложных прикладных систем, для которых технология предшествующих систем баз данных не была вполне удовлетворительной. Исследования в области ОБД были начаты в связи необходимостью разработать эффективный механизм, которые бы позволял объектно-ориентированным приложениям сохранять объекты после завершения своей работы и ими пользоваться при последующем запуске. То есть необходимо было объектно-ориентированной среде предоставить прозрачный механизм сохранения и выборки объектных данных из базы данных.

ОБД возникли не на пустом месте. Соответствующий базис обеспечивался работами в области баз данных, направлениями языков программирования с абстрактными типами данных и объектно-ориентированных языков программирования.

Первые объектные СУБД

В начале 80-х гг. многие исследовательские группы из университетов, научных институтов, ведущих компьютерных компаний и небольших начинающих компаний приступили к созданию ООСУБД. Были выпущены первые промышленные ООСУБД G-Base (1986 г), Gemstone (1987 г.), IRIS (1987), Statice (1988 г.), Vbase (1988 г.), ObjectStore (1988 г.), Versant (1988 г.), O2 (1988 г.), ORION (1989 г.),

Два направления в ОБД

К концу 80-х годов определились два направления в создании объектных баз данных: объектно-ориентированные базы данных (ООБД) и объектно-реляционные базы данных (ОРБД). ООБД использует объектно-ориентированный язык программирования в качестве языка базы данных и обеспечивает сохраняемость объектов с предоставлением всех функциональных возможностей, присущих для традиционных баз данных. ООБД предполагают создание самостоятельных объектно-ориентированных

систем управления базами данных (ООСУБД). ООСУБД реализует гибкую модель, которая базируется на той же парадигме, что и объектно-ориентированные языки программирования. ООСУБД обеспечивают более глубокую интеграцию с объектно-ориентированными приложениями, чем реляционная база данных.

В свою очередь ОРБД расширяет возможности реляционных баз данных средствами поддержки объектов.

Манифест объектно-ориентированных систем баз данных

В 1989 г. группа ведущих специалистов и исследователей в области баз данных написали "Манифест объектно-ориентированных систем баз данных" [1] (так называемый Первый манифест). Это был первый документ, в котором была предпринята попытка дать определение системам объектно-ориентированных баз данных. Были описаны основные свойства и характеристики, которыми должна обладать технология ООБД.

В нем также отмечается, что текущее состояние дел в проблематике ООБД характеризуется: отсутствием общепринятой модели данных, отсутствием единой формальной теории и активной экспериментальной деятельностью.

Общепринятая объектно-ориентированная модель данных отсутствовала не потому, что не было ни одной разработанной полной модели, а по причине отсутствия общего согласия о принятии какой-либо модели. Что касается формальной теории, то для ООБД нужно было нечто подобное тому, что создал Ковальский для логического программирования. Необходимость такой теории очевидна: формальная семантика основных понятий ООБД определена слабо. Ее отсутствие делало практически невозможным достижение консенсуса относительно модели данных.

Исследования в области объектных баз данных, исключительно активно развивались в 80-е годы. Это привело в конце 80-х к образованию промышленных компаний и рынка систем управления объектными базами данных (СУОБД). Вместе с тем, рынок объектных баз данных остро нуждался в стандарте. Решающее слово в этом отношении, как впрочем и в других проблемах, связанных с объектными базами данных, было сказано в стандарте ODMG.

Стандарт на хранение объектов ODMG 3.0

Летом 1991 г. в США была образована Object Data Management Group (ODMG) - Группа Управления Объектными Данными - как консорциум производителей СУОБД и других заинтересованных участников для разработки стандарта СУОБД. ODMG возглавил Рик Кеттелл (Rick Cattell). Задачей группы являлась разработка стандарта на хранение объектов в базах данных. В период с 1993 по 2001 год ODMG опубликовала пять версий своей спецификации, последняя из них была версия 3.0 [2], после чего группа завершила свою работу.


Рик Кеттелл

Стандарт на хранение объектов ODMG 3.0 разработан на основе трех существующих стандартов: управление базами данных (SQL), стандарты OMG - Object Management Group и стандарты на объектно-ориентированные языки программирования (C++, Smalltalk, Java). ODMG добавляет возможности взаимодействия с базами данных в объектно-ориентированные языки программирования. Стандарт состоит из следующих частей:

- *Объектная модель* - унифицированная основа всего стандарта. Она расширяет объектную модель консорциума OMG.
- *Язык определения объектов* (ODL - Object Definition Language) - средство определения типов объектов, которые соответствуют объектной модели данных ODMG. ODL используется для поддержки переносимости объектных схем между соответствующими системами управления объектными данными (СУОД).
- *Язык объектных запросов* (OQL - Object Query Language) - SQL - подобный декларативный язык, который предоставляет

эффективные средства для извлечения объектов из базы данных,

- *Формат обмена объектами* (OIF - Object Interchange Format) - язык описания загрузки и выгрузки текущего состояния СУОД в/из файлов и используются для обмена хранимыми объектами между СУОД.
- *Связывание с ОО-языками***.** Стандарт связывания с C++, Smalltalk и Java определяет Object Manipulation Language (OML) - язык манипулирования объектами, который расширяет базовые ОО-языки средствами манипулирования и хранения объектов.

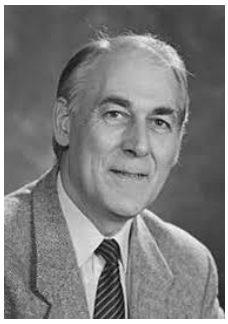
Второй манифест. Сообщество исследователей реляционных баз данных ответило на активность в области ООБД своим манифестом в поддержку ОРБД. В 1990 г. М. Стоунбрекер и его коллеги по комитету перспективных систем БД опубликовали "Манифест систем баз данных третьего поколения" [3] (так называемый Второй манифест), в котором они утверждают, что СУБД третьего поколения, то есть те, которые придут за реляционными, должны быть созданы на основе реляционных технологий. Сторонники этого направления придерживаются принципа эволюционного развития возможностей СУБД без коренной ломки предыдущих подходов и с сохранением преемственности с системами предыдущего поколения. Этот принцип был поддержан при создании дедуктивных и темпоральных баз данных, на этом же пути развивалось создание объектно-реляционных баз данных.

На основе этой идеи под руководством М. Стоунбрейкера в университете Беркли (Калифорния, США) была разработана СУБД Postgres [4]. Это была первая практически реализованная объектно-реляционная система баз данных, в которой был продемонстрирован подход по интеграции объектных и реляционных концепций. Следует также отметить Вона Кима (Won Kim), который в 1991 г. выпустил систему UniSQL [5], также считающуюся одной из первых объектно-реляционных СУБД.

Третий манифест

В марте 1995 г. была опубликована статья Хью Дарвена (Hugh Darwen) и Кристофера Дж. Дейта (Christopher J. Date), названная авторами “Третьим манифестом” [6]. В ней излагался взгляд авторов на относительно будущих систем управления базами данных и подход по интеграции реляционной и объектной технологий. Проблема, поднятая в манифесте, как решить задачу несоответствия между объектно-ориентированными языками программирования и системами управления реляционными базами данных. Авторы предлагают взять за основу реляционную базу данных и расширить ее поддержкой объектов в виде определяемых пользователем типов.

Кристофер Дейт



Хью Дарвен

Схемы реализации ОРБД

Были предложены различные схемы реализации ОРБД. К ним можно отнести следующие:

Объектно-реляционный шлюз (Object-Relational Gateway). Объектно-ориентированное приложение работает как обычный пользователь с использованием объектного языка, а шлюз выделяет и заменяет все объектно-ориентированные элементы этого языка на их реляционные компоненты. Несмотря на снижение производительности, такой вариант позволяет программистам целиком сконцентрироваться на объектно-ориентированной разработке.

Объектно-реляционный интерфейс (Object-Relational Interface). Между ООБД и ОРБД располагается промежуточный интерфейс, который отображает объектные конструкции в реляционные и наоборот. Объектно-ориентированное приложение работает с ООСУБД, которая через интерфейс взаимодействует с реляционной СУБД.

Унифицированные СУБД (unified DBMS). Еще одним решением является создание гибридных объектно-реляционных СУБД, которые могут хранить и традиционные табличные данные, и объекты.

Объектно-реляционные СУБД

Во второй половине 90-х гг. лидирующие компании начали выпуск СУБД, поддерживающие объектно-реляционную модель данных. Первой в 1996 г. вышла на рынок СУБД Informix, созданная на основе системы Illustra Стоунбрекера. В 1997 г. была выпущена объектно-реляционная версия СУБД DB2 компании IBM, в основу которой был положен исследовательский прототип Starburst IBM Almaden. В этом же году компания Oracle выпустила продукт этого же класса Oracle 8. В настоящее время практически все современные реляционные СУБД являются объектно-реляционными. Все они расширяют реляционную базу данных средствами представления объектов.

Позиции объектно-реляционного подхода упрочились благодаря принятию в 1999 г. версии стандарта SQL-3, в котором была введена поддержка объектно-ориентированной концепции (определяемые пользователем структурные типы данных, типизированные таблицы, объекты, методы, наследование).

Литература

1) Atkinson M., Bancilhon F., DeWitt D., Dittrich K., Maier D., Zdonik S. The object-oriented database system manifesto. In: Proceedings of the 1st International Conference on Deductive and Object-Oriented Databases; 1989. p. 223–240.
2) Cattell R.G.G., Barry D.K.(eds.). The Object Data Standard: ODMG 3.0. — San Francisco, Calif.: Morgan Kaufmann, 2000.
3) Stonebraker M., Rowe L.A., Lindsay B., Gray, Carey M., Brodie M., Bernstein Ph., Beech D. Third-Generation Database System Manifesto. SIGMOD Record 19(3), September, 1990. pp 31-43
4) Rowe L, Stonebraker M. The Postgres data model. In: Proceedings of the 13th International Conference on Very Large Data Bases; 1987. p. 83–96.
5) Won Kim. UniSQL/X unified relational and object-oriented database system. SIGMOD '94: Proceedings of the 1994 ACM SIGMOD international conference on Management of dataMay 1994, p. 481
6) Darwen H., Date C.J. (March 1995). "The third manifesto". ACM SIGMOD Record. New York, NY, USA: ACM Press. 24 (1): 39–49.

Распределенные базы данных

Распределенная база данных (РаБД) - *это интегрированная совокупность баз данных, которые физически распределены по компьютерной сети, а распределенная система управления базами данных (РаСУБД) - это программная система, которая так управляет распределенной базой данных, что аспекты распределения являются прозрачными (невидимыми) для пользователей.* РаСУБД может иметь общий интерфейс для доступа к распределенным данным [1].

Появление РаБД обусловлено тем, что здесь естественно представляется организационная структура данных предприятия, повышается надежность, доступность и локальный контроль, увеличивается производительность, облегчается процедура расширения системы.

Выработка концепции и исследования в области РаБД начались во второй половине 70-х годов. Среди многочисленных исследовательских систем наиболее известны следующие три: система SDD-1 [2–5], созданная в научно-исследовательском отделении корпорации Computer Corporation of America в конце 1970-х и начале 1980-х годов, система System R* [6–9], распределенная версия системы-прототипа System R, созданная в исследовательском отделе компании IBM в начале 1980-х годов, и система Distributed Ingres [10–12], распределенная версия прототипа системы Ingres, созданная также в начале 1980-х годов в Калифорнийском университете в Беркли. Также можно отметить проект POLIPHEME во Франции [13]. В проектах 70-х гг. был выявлен круг ключевых проблем, связанных с разработкой систем распределенных баз данных, предложены подходы к их решению. Тот факт, что всего за несколько лет в этой области были получены значимые результаты, подтверждается появлением в конце 70-х гг. обзоров на эту тему [14-16].

К концу 80-х годов были проведены многочисленные исследования, экспериментальные разработки и стали появляться первые промышленные РаБД. Было обращено внимание на создание мультибаз данных и на предоставление большей автономности индивидуальным системам [17, 18].

В 1986-87 гг. были представлены первые промышленные РаСУБД Ingres/STAR, Oracle 7 и DB2. В связи с этим появилась необходимость в формулировке основных принципов, требований и функциональных возможностей РаБД. В ответ на эти потребности в 90-м году появилась статья Дейта [19], в которой были сформулированы 12 правил РаБД, главное из которых - прозрачность для пользователей распределенной структуры баз данных. Эти правила были восприняты научным сообществом и ими до сих пор пользуются при разработке РаСУБД.

Типы РаБД

Имеется два основных типа РаБД: *однородные* (homogenous) и *неоднородные* (heterogeneous)

Однородные РаБД

В них все узлы находятся под управлением РаСУБД одного типа (и возможно под управлением одной операционной системы). Имеется два типа однородных РаБД: автономные и неавтономные. *Автономные* РаБД работают независимо, передавая и принимая сообщения друг другу для совместного обновления данных. *Неавтономные* РаБД предполагают существование центральной (главной) РаСУБД, которая координирует доступ к данным и их обновление в сети. Обычные распределенные (regular distributed) и параллельные базы данных относятся к однородным РаБД.

Неоднородные РаБД

Они работают под управлением различных операционных системах и типах РаСУБД. Имеется четыре типа неоднородных РаБД:

- федеративные (federated),
- с посредниками (mediators),
- мультибазы данных (multi-databases),
- одноранговые базы данных.

Федеративные РаБД

Представляют собой объединение БД различных типов, которыми владеют различные пользователи и которые объединя-

ются для облегчения совместного использования данных. Федеративная БД предполагает определение глобальной интеграционной схемы, содержащей отображения в схемы участвующих баз данных. Впервые федеративная БД была определена Маклеод и Хаймбигнер (McLeod and Heimbigner) в 1985 г. [20] и исследовалась во многих работах [21, 22]. В работе [23] приводится обзор по федеративным БД.

При существенном увеличении интегрируемых баз данных становится трудно, а иногда и невозможно определить глобальную интеграционную схему. *Мультибазы данных* [24] не предполагают существования глобальной схемы, но вместо этого язык запросов предоставляет возможность специфицировать выражения, позволяющие производить поиск по объединяемым базам данных.

Посредники

Посредники (mediators) [25, 26] занимают положение между системами с одной глобальной схемой и вообще без схем. Вместо этого пользователи определяют взгляды-посредники, которые объединяют и согласовывают данные из различных источников. Для таких взглядов необходим язык запросов, который позволяет формулировать запросы по многим базам данных наподобие языка запросов мультибаз данных.

Одноранговые БД

Одноранговые БД (peer-to-peer databases - P2PDB) [27–29] представляют собой совокупность автономных локальных репозиториев/баз данных, которые взаимодействуют друг с другом на равноправной основе, Основная задача P2PDB - распространять запросы между гетерогенными узлами в большой распределенной сети. Такое распространение может остановиться через несколько шагов. Это допускается для некоторых современных систем, которые не требуют высокой точности результатов, например, в поисковых машинах Интернета.

Распределение данных. Фрагментация

В РаБД стоит задача распределения логически целостной БД по узлам распределенной структуры так, чтобы оптимизировать целевую функцию. Существуют два фундаментальных метода решения этой задачи: фрагментация и репликация.

Фрагментация (сегментация, декомпозиция) предполагает разбиение данных на непересекающиеся сегменты (фрагменты) данных для их привязки к узлам сети. Репликация предполагает запоминание на различных узлах идентичных копий всей или части логической базы данных. РаСУБД гарантирует для пользователей прозрачность такого распределения. Помимо этого существует задача размещения фрагментированных/реплицированных данных в узлах сети.

Существует два вида фрагментации: горизонтальная и вертикальная. При горизонтальной фрагментации отношение разбивается на группы строк, которые распределяются по узлам. При вертикальной фрагментации отношение разбивается на группы столбцов, Также допускается гибридная фрагментация, которая предполагает одновременное использование двух предыдущих фрагментаций.

Основные исследования по фрагментации были проведены в начале 80-х годов [1, 30, 31, 32]. Одна из основных задач вертикальной фрагментации - определение наборов атрибутов, которые должны быть объединены в одну группу. В работах [33, 34] был предложен алгоритм энергетического связывания (bond energy algorithm) для группирования атрибутов, на основании которого производится вертикальная фрагментация. В работе [35] предложен модифицированный вариант этого алгоритма.

Что касается задачи размещения данных, то работы в этом направлении начались еще в конце 60-х годов, когда исследовалась проблема размещения файлов [36]. В работах [37–39] была исследована проблема сложности задач размещения. В работах [40, 41] были исследованы динамические алгоритмы размещения данных, которые предполагают возможность изменения первоначального размещения для учета изменений в методах доступа и рабочих нагрузках. Также были предложены методы интеграции фрагментации и размещения [30, 42].

Распределение данных. Репликация

Работы по репликации БД датируются началом 80-х годов, когда были проведены исследования по доступности данных и большинство предлагаемых решений обеспечивали согласованность данных. Хорошим обзором исследований этого времени является статья [43].

Основная проблема репликации данных заключается в том, что обновление любого заданного логического объекта должно распространяться по всем хранимым копиям этого объекта. Грей в 1996 г. продолжил исследования в этой области [44] и предложил варианты немедленного (eager) и отсроченного (lazy) обновлений. Один из вариантов отстроченного обновления - использование первичной копии (master copy), когда основная копия обновляется оперативно, а обновление вторичных копий откладывается до удобного времени, причем синхронная репликация предполагает завершение распространение изменений до завершения транзакции, а асинхронная репликация - допускается распространение изменений после завершения транзакции. Наконец, Греем была предложена двухуровневое (two-tier) обновление транзакций.

Эта статья активировала дальнейшие исследования по репликации. Одно из направлений исследований - уменьшение накладных расходов на коммуникацию и координацию за счет задержки обновлений удаленных копий. Однако в этом случае копии могут содержать устаревшие или даже несогласованные данные. В связи с этим были предложены решения по избеганию несогласованности [45], установлению ограничений на "устаревание" данных и обнаружению и устранению несогласованности [46].

Другое направление исследований, проводимых в контексте масштабируемой кластерной репликации, связано с разработкой методов обеспечения надежной согласованности при приемлемых затратах [47, 48]. С появлением облачных систем хранения в исследования были вовлечены внутриоблачные репликации, концептуально похожие на кластерную репликацию, а также междуоблачные и гео-репликации [49, 50]. Что касается распределенных систем, то сначала исследования были сконцентрированы на репликационных файловых системах [51, 52], затем на веб-серверных репликациях [53] и файловых репликациях одноранговых систем [54], Также были получены результаты по отказоустойчивым репликациям объектов [55, 56]. Как и в области баз данных, самые последние результаты, касающиеся репликации распределенных систем, относятся к облачной инфраструктуре, а именно, репликации в системах хранения [57], таких как HDFS и Cassandra, а также глобальная (wide-area) репликация [47, 58].

Тупики в РаБД

В БД, которые используют протокол блокировок для доступа к совместно используемым данным, возможны тупиковые ситуации (deadlock), когда транзакция ожидает наступления события, которое может произойти в результат последующих действие самой транзакции, например, когда две транзакции ждут друг друга.

Имеются следующие категории алгоритмов обнаружения тупиков в РаБД [1]: централизованные, иерархические, распределенные. Централизованные алгоритмы [59, 60] используют центральный узел для обнаружения тупиков. Как отмечается в [2], централизованная двухфазная блокировка (two-phase locking - 2PL) и обнаружение тупиков являются хорошей естественной комбинацией. Централизованное определение тупиков впервые было реализовано в Distributed INGRES [61]. Иерархически алгоритмы [60, 62] для обнаружения тупиков полагаются иерархическую структуру узлов РаБД. Распределенные алгоритмы [62, 63] полагаются на кооперацию всех узлов РаБД для обнаружения тупиков. Распределенные определение тупиков впервые было реализовано в System R* [63].

В монографии [1] представлен обзор методов управления распределенными тупиками. В обзорах [64-67] также обсуждаются различные распределенные алгоритмы обнаружения тупиков. В работе [68] приводятся сравнительный анализ дополнительных алгоритмов обнаружения тупиков: продвижение пути (path-pushing) [62], зондовый

(probe-based) [69], глобального состояния (global state) [70, 71].

Распределенная обработка запросов

Это процедура выполнения запроса в распределенной среде, где данные размещены в различных узлах компьютерной сети. Она предполагает преобразование запроса, сформулированного в высокоуровневом языке (например, SQL) в выражение процедурного языка низкого уровня (например, реляционная алгебра), которое получило название план выполнения запроса. Затем этот план оптимизируется с учетом распределенности данных и, наконец, производится последовательное выполнение операторов полученного оптимального плана.

Исследования по распределенной обработке запросов начались в конце 70-х годов. В этот период были разработаны три экспериментальные системы, в которых были заложены фундаментальные методы распределенной оптимизации и обработки запросов: SDD-1 [72] (1976), Distributed INGRES [73, 74] (1977) и System R* [75, 76] (1981). Считается, что первым дистрибутивным алгоритмом оптимизации запросов является "скалолазание" (hill climbing) Вонга [77], который затем был улучшен в SDD-1 включением операции полусоединения. Оптимизационный алгоритм SDD-1 является статичным и направлен на уменьшение суммарных коммуникационных затрат и не поддерживает фрагментацию и репликацию. Дистрибутивный алгоритм оптимизации запросов Distributed Ingres [73] на каждом шаге детерминировано анализирует пространство возможных планов и принимает решение по локальной оптимизации. Он поддерживает горизонтальную фрагментацию. Целевая функция оптимизации является взвешенной комбинацией стоимости суммарного времени и времени реакции. Алгоритм валяется динамическим.

Дистрибутивный алгоритм оптимизации запросов System R* [76] всесторонне анализирует пространство поиска всех возможных планов выполнения запросов. Алгоритм не поддерживает фрагментацию и репликацию. Целевая функция оптимизации учитывает локальную обработку и коммуникационные затраты. Алгоритм является статическим.

Были проведены исследования по оптимизации выполнения выражений реляционной алгебры, включая распределенную среду. В статье [78] приведен обзор этих результатов. Было предложено несколько подходов по динамической оптимизации запросов для параллельных и распределенных баз данных [79]. Алгоритм в [80] предполагает изменение плана отработки запроса в процессе его выполнения, чтобы учитывать непредвиденные обстоятельства. В системе Mariposa [81] впервые был предложена экономическая модель оптимизации распределенного запроса. В монографиях [1, 82] детально освещаются результаты в области технологий распределенных баз данных и оптимизации распределенных запросов, полученные в 80-90-х года. Статья [79] является более свежим обзором в этой области.

Управление параллелизмом

Управление параллелизмом (concurrency control) - это процедура такого управления одновременной бесконфликтной работой многих транзакций, при которой транзакции корректно выполняют свою работу без нарушения ограничений целостности БД (принципа ACID).

Исследования по управлению параллелизмом в распределенных системах зародились в начале 80-х годов. Они полагались на широко известную в то время статью [83] по управлению параллелизмом в централизованных БД. Грей развил эти идей для транзакций [84], а Спектор и Шварц [85] исследовали транзакции в распределенной среде.

Было предложено три механизма управления параллелизмом: блокировка, оптимистический протокол и упорядочение по временным отметкам.

Блокировка

Блокировка - это ограничение доступа к совместно используемым ресурсам (данным) при одновременном выполнении многих транзакций.

Первым широко известным механизмом блокировок стала *двухфазная блокировка (*Two-Phase Locking - 2PL), которая

впервые была описана в [83]. Впоследствии было определено множество ее разновидностей: строгая (strict), консервативная (conservative), первичной копии (primary copy), распределенная (distributed), точная (rigorous). В [86] описан вариант 2PL, учитывающий использование старых значений. Также были предложены гибридные блокировки, которые предполагают использование методов, отличных от 2PL [87–89].

Оптимистический протокол

Следующий тип управления параллелизмом получил название *оптимистический* в том смысле, что создаются локальные копии данных транзакции и обновляются именно они, а не сами данные. Впервые этот метод был предложен в работе [90], и с тех пор было исследовано множество его разновидностей [91, 92]

Упорядочение по временным отметкам

Наконец, *упорядочение по временным отметкам* (timestamp odering) использует Системное Время или некоторый логический счетчик в качестве временных отметок для упорядочения выполнения параллельных транзакций. Транзакции присваивается временная отметка, как правило, на основании времени запуска транзакции. Более старая транзакция имеет больший приоритет. При возникновении конфликтов предпочтение отдается более приоритетной транзакции. Этот протокол описывается в [93-95]. В [95, 96] также описываются многоверсионные временные отметки. Хорошим обзором по методам управления параллелизмом является статья [97].

Литература

1) Özsu M.T., Valduriez P. Principles of Distributed Database Systems, Fourth Edition, Springer, 2020
2) Rothnie J.B. Jr, Bernstein P.A., Fox S., Goodman N., Hammer M., Landers T.A., Reeve C.L., Shipman D.W., Wong E. Introduction to a system for distributed databases (SDD-1). ACM Trans. on Database Syst. 1980;5(1):1–17.
3) Bernstein P.A., Shipman D.W., Rothnie J.B. Concurrency Control in a System for Distributed Databases (SDD-1) ACM Transactions on Database Systems, Vol. 5, No. 1, March 1980, Pages 19-51.
4) Hammar M., Shipman D. Reliability mechanism for SDD-1: a system for distributed database. ACM Trans. Database Syst., 5 (4) (Dec. 1980), pp. 431-466
5) Bernstein P.A., Goodman N., Wong E., Reeve C.L., Rothnie J.B. Query Processing in a System for Distributed Databases (SDD-1). ACM Transactions on Database Systems, Vol. 6, No. 4, December 1981, Pages 602-625
6) Selinger P.G. An architectural overview of R*: a distributed database management system. In: Proceedings of the 5th Berkeley Workshop on Distributed Data Management and Computer Networks; 1981, p. 187.
7) Williams R., Daniels D., Haas L., Lapis G., Lindsay B., Ng P. Obermarck R., Selinger P., Walker A., Wilms P., Yost R. R*: An Overview of the Architecture.IBM Research Report RJ3325, IBM Research Laboratory, San Jose, CA, Dec. 1981.
8) Lohman G.M., Mohan C., Haas L.M., Daniels D., Lindsay B.G., Selinger P.G., Wilms P.F. Query Processing in R*. Lohman G.M. et al. (1985) Query Processing in R*. In: Kim W., Reiner D.S., Batory D.S. (eds) Query Processing in Database Systems. Topics in Information Systems. Springer, Berlin, Heidelberg, pp. 31-47
9) Daniels D. et al. An Introduction to Distributed Query Compilation in R*. Distributed Data Bases (ed. H.-J. Schneider): Proc. 2nd Int. Symposium on Distributed Data Bases. - New York, N.Y.: North-Holland, 1982.
10) Stonebraker M.R., Neuhold E.J. A Distributed Data Base Version of INGRES // Proc. 2nd Berkley Conf. On Distributed Data Management and Computer Networks. — Lowrence Berkley Laboratory, May 1977.
11) Epstein R., Stonebraker M., Wong E. Distributed Query Processing in a Relational Database System // Proc. 1978 ACM SIGMOD Int. Conf. on Management of Data. — Austin, Tex. — May-June 1978.
12) Stonebraker M. The design and implementation of distributed INGRES. The

INGRES Papers, Reading; 1986, p. 187–196.
13) Adiba M.E., Andrade J.M., Fernandez F., Gia Toan Nguyen. POLIPHEME: An experience in distributed database system design and implementation. Proc. of Int. Simposium on Distributed Data Bases. Paris, France, 1980, pp. 475-479
14) Epstein R., Stonebraker M., Wong E. Distributed Query Processing in a Relation Data Base System. SIGMOD '78: Proceedings of the 1978 ACM SIGMOD international conference on management of dataMay 1978 Pages 169–180.
15) Davenport R.A. Distributed database technology — a survey. Computer Networks (1976), Volume 2, Issue 3, 1978, Pages 155-167,
16) Rothnie J.B., Goodman N. A Survey of Research and Development in Distributed Database Management. VLDB '77: Proceedings of the third international conference on Very large data bases - Volume 3, October 1977 Pages 48–62
17) Sheth A.P., Larson J.A. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Comput Surv. 1990;22(3):183–236.
18) Stonebraker M., Aoki P.M., Pfeffer A, Sah A, Sidell J, Staelin C, Yu A. Mariposa: a widearea distributed database system. VLDB J. 1996;5(1):48–63.
19) Date C.J. What is a Distributed Database System? In: Date C. J. Relational Database Writings 1985-1989. — Reading, Mass.: Addison-Wesley, 1990.
20) Heimbigner D., McLeod D. "A Federated Architecture for information management". ACM Transactions on Information Systems, 1985,.Volume 3, Issue 3. pp. 253–278.
21) Sheth A.P., Larson J.A. "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases". ACM Computing Surveys, 1990, Vol. 22, No.3. pp. 183–236.
22) Masood N., Eaglestone B. "Component and Federation Concept Models in a Federated Database System". Malaysian Journal of Computer Science, 2003, 16 (2): 47–57.
23) Sheth A., Larson J.A. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. ACM Computing Surveys 1990, 22(3):183-236
24) Litwin W., Mark L., Roussopoulos N. Interoperability of multiple autonomous databases. ACM Comput. Surv. 1990; 22(3):267–293.
25) Wiederhold G. Mediators in the architecture of future information systems. Computer, 1992, Vol. 25, No. 3, pp. 38–49.
26) Risch T., Josifovski V., Katchaounov T. (2004) Functional Data Integration in a Distributed Mediator System. In: Gray P.M.D., Kerschberg L., King P.J.H., Poulovassilis A. (eds) The Functional Approach to Data Management. Springer, Berlin, Heidelberg. pp. 211-238
27) Gribble S.D., Halevy A.Y., Ives Z.G., Rodrig M., Suciu D. What Can Database Do for Peer-to-Peer? In Processing of Int'l Workshop on the WEB and Databases (WebDB), 2001, pp. 31-36
28) Bonifati A., Chrysanthis P.K., Ouksel A.M., Sattler K.-U. Distributed databases and peer-to-peer databases: Past and present, ACM SIGMOD Record, 2008, 37(1): 5-11
29) Beng Chin Ooi, Kian-Lee Tan, guest editors. Introduction: special section on peer-to-peer-based data management. IEEE Trans Knowl Data Eng. 2004;16(7):785–786.
30) Sacca D., Wiederhold G. Database partitioning in a cluster of processors. ACM Trans Database Syst. 1985;10(1):29–56.
31) Ceri S., Negri M., Pelagatti G. Horizontal data partitioning in database design. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 1982. p. 128–136.
32) Ceri S., Pelagatti G. Distributed databases: principles and systems. New York: McGraw-Hill; 1984.
33) Navathe S., Ceri S., Wiederhold G., Dou J. Vertical partitioning of algorithms for database design. ACM Trans Database Syst. 1984;9(4):680–710.
34) McCormick W.T., Schweitzer P.J., White T.W. Problem decomposition and data re-

organization by a clustering techniqiue. Oper Res. 1972;20(5): 993–1009
35) Shikha Mehta, Parul Agarwal, Prakhar Shrivastava, Jharna Barlawala, Differential bond energy algorithm for optimal vertical fragmentation of distributed databases, Journal of King Saud University - Computer and Information Sciences, 2018,
36) Chu W.W. Optimal file allocation in a multiple computer network. IEEE Trans Comput. 1969;- 18(10):885–889.
37) Apers P.M. Data allocation in distributed database systems. ACM Trans Database Syst. 1988;13(2): 263–304.
38) Bell D.A. Difficult data placement problems. Comput J. 1984;27(4):315–320.
39) Chang C.C, Shieh J.C. On the complexity of file allocation.problem. In: Proceedings of the International.Conference on the Foundations of Data Organization; 1985. p. 177–181.
40) Brunstrom A., Leutenegger S.T, Simha R. Experimental evaluation of dynamic data allocation strategies in a distributed database with changing workloads. In: Proceedings of the 40) 4th International Conference on Information and Knowledge Management; 1995. p.395–402.
41) Karlapalem K., Ng M.P. Query-driven data allocation algorithms for distributed database systems. In: Proceedings of the 8th International Conference Database and Expert Systems Applications; 1997. p. 347–356.
42) Yoshida M., Mizumachi K., Wakino A., Oyake I., Matsushita Y. Time and cost evaluation schemes of multiple copies of data in distributed database systems. IEEE Trans. Softw. Eng. 1985; 11(9) :954–958.
43) Bernstein P.A., Hadzilacos V., Goodman N. Concurrency control and recovery in database systems. Reading: Addison Wesley; 1987.
44) Gray J., Helland P., O'Neil P., Shasha D. The dangers of replication and a solution. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 1996. p. 173–182.
45) Breitbart Y., Komondoor R., Rastogi R., Seshadri S., Silberschatz A. Update propagation protocols for replicated databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 1999. p. 97–108.
46) Saito Y., Shapiro M. Optimistic replication. ACM Comput Surv. 2005;37(1):42–81.
47) Lin Y., Kemme B., Patiño-Martínez M., Jiménez-Peris R. Middleware based data replication providing snapshot isolation. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 2005. p. 419–430.
48) Wiesmann M., Schiper A. Comparison of database replication techniques based on total order broadcast. IEEE Trans Knowl Data Eng. 2005;17(4):551–566
49) Corbett J.C., Dean J., Epstein M., Fikes A., Frost C., Furman J.J., Ghemawat S., Gubarev A., Heiser C., Hochschild P., Hsieh W.C., Kanthak S., Kogan E., Li H., Lloyd A., Melnik S., Mwaura D., Nagle D., Quinlan S., Rao R., Rolig L., Saito Y., Szymaniak M., Taylor C., Wang R., Woodford D. Spanner: Google's globally distributed database. ACM Trans Comput Syst. 2013;31(3):8:1-8:22.
50) Mahmoud H.A., Nawab F., Pucher A., Agrawal D., El Abbadi A. Low-latency multi-datacenter databases using replicated commit. Proc VLDB Endow. 2013;6(9):661–672.
51) Satyanarayanan M., Kistler J.J., Kumar P., Okasaki M.E., Siegel E.H., Steere D.C. Coda: a highly available file system for a distributed workstation environment. IEEE Trans Comput. 1990;39(4):447–459.
52) Terry D.B., Theimer M., Petersen K., Demers A.J., Spreitzer M., Hauser C. Managing update conflicts in Bayou, a weakly connected replicated storage system. In: Proceedings of the 15th ACM Symposium on Operating System Principles; 1995. p. 172–183.
53) Sivasubramanian S., Szymaniak M., Pierre G., van Steen M. Replication for web hosting systems. ACM Comput Surv. 2004;36(3):291–334.
54) Lv Q., Cao P., Cohen E., Li K., Shenker S. Search and replication in unstructured peer-to-peer networks. In: Proceedings of the 16th Annual International Conference on Supercomputing; 2002. p. 84–95.

55) Budhiraja N, Marzullo K, Schneider FB, Toueg S. The primary-backup approach. In: Mullender S, editor. Distributed systems. 2nd ed. Reading: Addison Wesley; 1993. p. 199–216.
56) Schneider F.B. Replication management using the state-machine approach. In: Mullender S, editor. Distributed systems. 2nd ed. Reading: Addison Wesley; 1993. p. 169–198.
57) Almeida S., Leitão J., Rodrigues L.E.T. Chainreaction: a causal+ consistent datastore based on chain replication. In: Proceedings of the 8th ACM SIGOPS/EuroSys European Conference on Computer Systems; 2013. p. 85–98.
58) Sovran Y., Power R., Aguilera M.K., Li J. Transactional storage for geo-replicated systems. In: Proceedings of the 23rd ACM Symposium on Operating System Principles; 2011. p. 385–400.
59) Gray J. Notes on data base operating systems. In: Advanced Course: Operating Systems; 1978. p. 393–481.
60) Ho G.S, Ramamoorthy C.V. Protocols for deadlock detection in distributed database systems. IEEE Trans Softw Eng. 1982;8(6):554–557.
61) Stonebraker M. The design and implementation of distributed ingres. In: The INGRES papers: anatomy of a relational database system; 1986. p. 187–96.
62) Menascé D.A., Muntz R. Locking and deadlock detection in distributed data bases. IEEE Trans Softw Eng. 1997;5(3):195–202.
63) Mohan C., Lindsay., Bruce G., Obermarck R. Transaction management in the R* distributed database management system. ACM Trans Database Syst. 1986;11(4):378–396.
64) Abonamah A.A,. Elmagarmid A. A survey of deadlock detection algorithms in distributed database systems. In: Advances in distributed and parallel processing. System paradigms and methods, vol. 1; 1994. p. 310–341.
65) Elmagarmid A.K. A survery of distributed deadlock algorithms. ACM SIGMOD Rec. 1986;15(3):37–45.
66) Knapp E. Deadlock detection in distributed databases. ACM Comput Surv. 1987;19(4): 303–328.
67) Singhal M. Deadlock detection in distributed systems. Computer. 1989;22(11):37–48.
68) Krivokapic N, Kemper A, Gudes E. Deadlock detection in distributed database systems: a new algorithm and a comparative performance analysis. VLDB J. 1999;8(2):79–100.
69) Roesler M., Burkhard W.A., Cooper K.B. Efficient deadlock resolution for lock-based concurrency control schemes. In: Proceedings of the 18th International Conference on Distributed Computing Systems; 1998. p. 224–233.
70) Bracha G., Sam T. Distributed deadlock detection. Distrib Comput. 1985;2(3):127–138.
71) Chandy K.M., Lamport L. Distributed snapshots: determining global states of distributed systems. ACM Trans Comput Syst. 1986;3(1):63–75.
72) Bernstein P.A., Goodman N., Wong E., Reeve C.L., Rothnie Jr.J.B. Query processing in a system for distributed databases (SDD-1). ACM Trans Database Syst. 1981;6(4):602–625.
73) Epstein RS, Stonebraker M, Wong E. Distributed query processing in a relational data base system. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 1978. p. 169–180.
74) Stonebraker M. The design and implementation of distributed INGRES. In: Stonebraker M, editor. The INGRES papers. Reading: Addison-Wesley; 1986.
75) Williams R., Daniels D., Hass L., Lapis G., Lindsay B., Ng P., Obermarck R., Selinger P., Walker A., Wilms P., Yost R. R*: an overview of the architecture. IBM Research Lab, San Jose, Technical Report RJ3325; 1981.
76) Haas L.M., Selinger P.G., Bertino E., Daniels D., Lindsay B.G., Lohman G.M., Masunaga Y., Mohan C., Ng P., Wilms P.F., Yost R.A. R*: a research project on distributed relational DBMS. IEEE Database Eng Bull. 1982;5(4):28–32.

77) Wong E. Retrieving dispersed data from SDD-1: a system for distributed databases. In: Proceedings of the 2nd Berkeley Workshop on Distributed Data Management and Computer Networks; 1977. p. 217–235.
78) Yu C.T. and Chang C.C. Distributed query processing. ACM Comput. Surv., 16(4):399–433, 1984.
79) Kossmann D. The state of the art in distributed query processing. ACM Comput Surv. 2000;32(4):422–469.
80) Urhan T., Franklin M.J., Amsaleg L. Cost based query scrambling for initial delays. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 1998. p. 130–141.
81) Stonebraker M., Devine R., Kornacker M., Litwin W., Pfeffer A., Sah A., Staelin C. An economic paradigm for query processing and data migration in Mariposa. In: Proceedings of the 3rd International Conference Parallel and Distributed Information Systems; 1994. p. 58–67.
82) Ceri S., Pelagatti G. Distributed databases principles and systems. New York: McGraw-Hill; 1984.
83) Eswaran K.P., Gray J.N., Lorie R.A., Traiger I.L. The notion of consistency and predicate locks in a database system. Commun ACM. 1976;19(11):624–633.
84) Gray J.N. The transaction concept: virtues and limitations. In: Proceedings of the 7th International Conference on Very Data Bases; 1981. p 144–154.
85) Spector A.Z., Schwarz P.M. Transactions: a construct for reliable distributed computing. ACM Operat Syst Rev. 1983;17(2):18–35
86) Stearns R.E., Rosenkrantz D.J. Distributed database concurrency controls using before-values. SIGMOD '81: Proceedings of the 1981 ACM SIGMOD international conference on Management of data, 1981, pp. 74–83
87) Boral H., Gold I. Towards a self-adapting centralized concurrency control algorithm. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 1984. p. 18–32.
88) Lausen G. Concurrency control in database systems: a step towards the integration of optimistic methods and locking. In: Proceedings of the ACM Annual Conference; 1982. p. 64–68.
89) Salem K., Garcia-Molina H., Shands J. Altruistic locking. ACM Trans Database Syst. 1994;19(1):17–165.
90) Kung H.T. Robinson J.T. "On Optimistic Methods for Concurrency Control". ACM Transactions on Database Systems, Vol. 6, No. 2, 1981, pp. 213-226.
91) Rahm E. Concepts for Optimistic Concurrency Control in Centralized and Distributed Database Systems. IT Informationstechnik, (in German), 1988, vol. 30, no. 1, pp. 28-47.
92) Thomasian A. Distributed optimistic concurrency control methods for high-performance transaction processing IEEE Transactions on Knowledge and Data Engineering, 1998, 10(1):173-189
93) Bernstein P., Goodman N. [1980] “Timestamp-Based Algorithms for Concurrency Control in Distributed Database Systems,” in VLDB '80: Proceedings of the sixth international conference on Very Large Data Bases - Volume 6, October, 1980, pp. 285–300
94) Bernstein P.A., Goodman N., Rothnie J.B. Jr., Papadimitriou C.H. Analysis of serializability of SDD1: a system of distributed databases (the fully redundant case). IEEE Trans. On Software Engineering, SE-4: 3 (1978), pp. 154-168.
95) Reed D.P. “Implementing Atomic Actions on Decentralized Data,” TOCS, 1:1, February 1983, pp. 3–23.
96) Reed D.P. Naming and synchronization in a decentralized computer system. Ph. D. Thesis, MIT, Cambridge, Mass., 1977
97) Thomasian A. Concurrency Control: Methods, Performance, and Analysis, ACM Computing Surveys, 1998, 30(1):70-119

Машины баз данных

В общем случае *машиной базы данных* (МБД) принято называть аппаратно-программный мультимикропроцессорный комплекс, предназначенный для выполнения всех или некоторых функций СУБД. Это направление баз данных начало развиваться в начале 70-годов. На первом этапе, который длился 10-12 лет, основная идея исследований и разработок МБД была направлена на создание специализированных вычислительных устройств и разработки архитектур, в которых процесс вычисления базы данных размещался ближе к дискам с тем, чтобы достичь значительного увеличения производительности. В эти годы было реализовано более 50 проектов. Основными критериями оценки того или иного проекта были полнота выполняемых функций СУБД и ожидаемое повышение производительности при их выполнении. На основе экспериментальных прототипов впоследствии во многих странах мира сформировалось производство различных образцов машин баз данных [1].

Процессоры фильтров

В этот период были предложены решения, получившие названия *процессоров фильтров,* задача которых эффективно проверять передаваемые данные с дисков на внешний сервер. В работах [2, 3] процессоры фильтров были разделены на следующие группы:

- процессор на дорожку (Processor-per-Track - PPT)
- процессор на головку (Processor-per-Head - PPH)
- процессор на диск (Processor-per-Disk - PPD)
- мультипроцессорный кэш (Multi-Processor Cache - MPC).
- процессор на ячейку пузырьковой памяти (Processor-per-Bubble-cell - PPB)

Процессор на дорожку - PPT

Согласно [4] пионером исследований в области МБД был Даниэль Слотник (Daniel L. Slotnick), опубликовавший в 1970 г. статью [5], в которой предлагается архитектура с процессором на каждую дорожку. В этой архитектуре запоминающее устройство состоит из большого количества ячеек, каждая из которых имеет дорожку данных, с которой связан процессор, выполняющий "на лету" функцию поиска требуемых данных. Координацию работы с ячейками выполняет управляющий процессор. Основная идея Слотника заключалась в том, чтобы производить поиск в базе данных непосредственно на запоминающем устройстве, ограничивая тем самым объем данных, передаваемых на основной процессор. В дальнейшем подход Слотника развили Паркер (Parker) [6], Мински (Minsky) [7], Пархами (Parhami) [8]. На основе этой архитектуры были реализованы МБД RAP [9], CASSM [10], RARES [11].


Даниэль Слотник

Процессор на головку - PPH

К этому классу относятся МБД, в которых логика обработки данных привязывается к каждой головке в диске с подвижными головками. В PPH данные параллельно передаются от головок ко множеству процессоров. Каждый процессор применяет функцию отбора к выходящему потоку данных и размещает выбранные данные в выходном буфере. При такой организации каждый цилиндр диска с подвижной головкой анализируется за один оборот. К этому классу относятся МБД DBC [12], SURE [13].

Процессор на диск - PPD

В отличие от PPT и PPH данная архитектура предполагает использование стандартных дисководов. Процессор (или множество процессоров) помещается между диском и запоминающим устройством, в которое передаются отобранные данные. Процессор действует как фильтр, передавая в основной процессор только те данные, которые соответствуют критерию отбора.

Мультипроцессорный кэш - MPC

К этому классу относятся МБД, в которых специализированные процессоры отделяются от устройств хранения большим

дисковым кэшем. Цель этого архитектурного решения - поддерживать параллелизм обработки при использовании традиционных устройств хранения. Перед обработкой данные должны быть перемещены из диска в кэш и после этого они становятся доступными процессорам в параллельном режиме. Более того, промежуточные результаты выполнения запроса помещаются процессорами в кэш и к ним предоставляется быстрый доступ для выполнения последующих операций запроса. Реализовано много МБД этого класса, включая RAP.2 [14], DIRECT [15], INFOPLEX [16], RDBM [17], DBMAC [18]

Процессор на ячейку пузырьковой памяти - PPB

С каждой ячейкой внешней пузырьковой памяти ассоциируется процессор.

Следует отметить, что в этот период большинство проектов разработки МБД концентрировалось вокруг специализированного аппаратного обеспечения, находящегося еще в стадии разработки, такого как CCD-память (charge-coupled device, устройство с зарядовой связью), пузырьковая память (bubble memory), диски с фиксированными головками на каждую дорожку (head-per-track disks) и оптические диски (optical disks). Ни одна из этих технологий себя не оправдала в полной мере. В связи с этим по истечении двенадцати лет активности в этом направлении будущее МБД выглядело неопределенным даже для самых верных их сторонников. Так, например, в 1983 г. статья [4] предрекала скорое исчезновение МБД.


Эсен Озкарахан

Наиболее известными монографиями по машинам баз данных первого этапа были написанные в 1986 г. Эсен Озкараханом [1], а также в 1990 г. Калиниченко Л.А. и Рывкиным В.М. [19].

Параллельные базы данных

Несмотря на пессимистические настроения, направление МБД выжило и успешно развивается благодаря параллельным системам баз данных.

Как отмечается в [20], успех параллельных баз данных объясняется широким распространением реляционных баз данных. В 1983 году они только еще появлялись на рынке, сегодня же доминируют. Реляционные запросы как нельзя лучше подходят для параллельного выполнения; они состоят из однородных операций над однородным потоком данных. Каждая операция образует новое отношение, так что из операций могут быть составлены высокопараллельные графы потоков данных. Две операции могут работать последовательно, если направить вывод одной операции на вход другой. Это так называемый конвейерный параллелизм (pipelined parallelism). Если разделять вводимые данные между несколькими процессорами и памятью, часто оказывается возможным разбить операцию на несколько независимых операций, каждая из которых работает с частью данных. Такое разделение данных и обработки называется раздельным параллелизмом (partitioned parallelism)

Таким образом, история показывает, что узкоспециализированные машины баз данных оказались несостоятельными, в то время как параллельные системы баз данных достигли огромных успехов. Успешные параллельные системы баз данных строятся на обычных процессорах, памяти и дисках. Именно в этих системах в основном отразились идеи высоко параллельных архитектур.

Массивные параллельные вычисления

В 1980-х годах исследования по машинам баз данных были сосредоточены на массивных параллельных вычислениях (massive parallel computing). Процессоры общего назначения и дисководы были соединены в узлы, и такие узлы затем были объединены в высокоскоростные межблочные связи [20, 21]. Некоторые из этих типов машин баз данных достигли большого успеха в промышленности.

Классификация параллельных мультипроцессорных систем

В середине 80-х годов Стоунбрейкер предложил следующую простую классификацию параллельных мультипроцессорных систем [22]:

- *Совместно используемые память и диски* (shared-everything - SE). Все процессоры имеют прямой доступ к общей глобальной памяти и ко всем дискам. Взаимодействие между процессорами осуществляется с использованием общей памяти. Примерами подобных систем являются XPRS [23], DBS3 [24], Volcano [25].
- *Совместно используемые диски* (shared disks - SD). Каждый процессор имеет свою собственную память и прямой доступ ко всем дискам. Все процессоры связаны друг с другом через высокоскоростную сеть для передачи данных Примерами параллельных систем баз данных SD-архитектруры являются IBM IMS [26], Oracle Parallel Server [27], nCUBE [28], VAXclusters [29], IBM Parallel Sysplex [30].
- *Отсутствие совместного использования ресурсов* (shared-nothing - SN). Каждая память и диск находятся в распоряжении одного процессора, который работает как сервер хранящихся в них данных. Массовое запоминающее устройство в таких архитектурах распределено между процессорами посредством соединения одного или более дисков. Как и SD-архитектуре, все процессоры связаны друг с другом через высокоскоростную сеть. Отсутствие совместного использования ресурсов характерно для систем баз данных, используемых в проектах Teradata [31], Gamma [32], Tandem [33], Bubba [34], Arbre [35] и nCUBE [36]. Примерами коммерческих систем SN-архитектуры являются NonStop SQL [37], Informix PDQ [38], NCR/Teradata DBC [39], IBM DB2 PE [40].

Вопросу анализа архитектур параллельных систем баз данных также посвящена статья Соколинского Л.Б. [41].

Со временем появились мультипроцессорные системы, которые сочетали характеристики SE- и SN-архитектур, поэтому Коупленд и Келлер [42] предложили следующим образом расширить классификацию Стоунбрейкера:

- *кластеризовано все* (clustered everything - CE) - кластеры, имеющие SE-архитектуру, объединяются по принципу SN-архитектуры
- *кластеризованы диски* (clustered-disk - CD) - кластеры, имеющие SD-архитектуру, объединяются по принципу SN-архитектуры

Такие архитектуры получили название иерархических [43]. Предложения Коупленда позволяются строить двухуровневые иерархии (ISE/SD-кластеры первого уровня объединяются в SN-кластеры второго уровня). Двухуровневая архитектура Коупленда может быть легко расширена до архитектур с тремя или более иерархическими уровнями. Двухуровневая иерархическая архитектура была исследована в работах [42, 44–47].

Отметим, что во второй половине 90-х годов появились многопроцессорные системы, имеющие компоненты сложной конструкции и вобравшие в себя различные архитектурные решения, которые не подпадают под классификацию Стоунбрейкера/ Коупленда. К ним можно отнести мультипроцессорную систему серии МВС-100/1000 [48], мультипроцессорную систему SP2 [49] компании IBM, компьютеры на основе технологии ServerNet компании Tandem [50], гибридную архитектуру CDN [51].

Современные МБД

Согласно [52] первым шагом на пути создания современных МБД была презентация в 2000 г. технологии InfiniBand - высокоскоростной коммутируемой компьютерной сети компании Voltaire (партнер Oracle, начиная с 2001 года), которая была использована в Oracle RAC (Real Application Cluster), начиная с версии Oracle Database 9i. В 2009 г. среди Top 500 суперкомпьютеров мира 29% использовали InfiniBand. Oracle Exadata V1 стала первой современной МБД, созданной Oracle HP (Hewlett-Packard) в

2008 г. Тестирование этой МБД в CERN [53] показало высокую эффективность по времени и памяти при выгрузке данных большого объема. В 2009 г. Sun и Oracle создали МБД Exadata Database Machine Version 2. Благодаря использованию современных технологий этих двух компаний МБД работает в два раза эффективнее, чем Oracle Exadata V1,

Teradata Database — это система массовой параллельной обработки (MPP), имеющая коллективную распределённую архитектуру. Задача равномерно распределяется по всем процессам и параллельно обрабатывается. Поддерживает архитектуру без совместного использования ресурсов. Обладает высокой горизонтальной масштабируемостью Имеет один из самых развитых оптимизаторов на рынке. Автоматически равномерно распределяет данные по дискам. Поддерживает стандарт SQL**.**

Литература

1) Ozkarahan E. Database Machines and Database Management. Englewood Cliffs, N.J.; Prentice-Hall, 1986 - 636 p.
2) DeWitt D.J., Hawthorn P.B. A performance evaluation of data base machine architectures. In: Proceedings of the 7th International Conference on Very Data Bases; 1981. p. 199–214.
3) Boral H., DeWitt D.J., Wilkinson W.K. Performance evaluation of four associative disk designs Information Systems, Volume 7, Issue 1, 1982, Pages 53-64
4) Boral H., DeWitt D. Database machines: An idea whose time has passed? A critique of the future of database machines. In Proceedings of the 1983 Workshop on Database Machines. H.-O. Leilich and M. Missikoff, Eds., Springer-Verlag, 1983, pp. 16-187
5) Slotnik D.L. "Logic per Track Devices" in Advances in Computers, Vol. 10., Frantz Alt, Ed., Academic Press, New York, 1970, pp. 291-296. TODS, Vol 1, No. 3., 1976.
6) Parker J.L. "A Logic per Track Retrieval System," IFIP Congress, 1971. J.L. Parker, “A Logic per Track Retrieval System”, Proc. IFIP Congress 1971, pp. TA-4-146 to TA-4-150
7) Minsky N., "Rotating Storage Devices as Partially Associative Memories" Proc. 1972 FJCC. N. Minsky: Rotating Storage Devices as Partially Associative Memories, FJCC 1972, AFIPS Conf. Proc., pp. 587–595
8) Parhami B. "A Highly Parallel Computing System for Information Retrieval" Proceedings of the Fall Joint Computer Conference, 1972. pp. 681-690
9) Ozkarahan E.A., Schuster S.A., Smith K.S. RAP: An Associative Processor for Data Base Management. Proc. AFIPS 44, NCC, 1975, pp. 379-387.
10) Su S.Y.W., Lipovski G.J."CASSM: A Cellular System for Very Large Data Bases", VLDB '75: Proceedings of the 1st International Conference on Very Large Data BasesSeptember 1975 Pages 456–472
11) Lin S.C., Smith D.C.P., Smith J.M. "The Design of a Rotating Associative Memory for Relational Database Applications," TODS Vol. 1, No. 1, pages 53 - 75, Mar. 1976.
12) Kannan K. "The Design of a Mass Memory for a Database Computer," Proc. Fifth Annual Symposium on Computer Arcliecture. Palo Alto, CA. April 1978, pp. 44-51
13) Leilich H.-O., Stiege G., Zeidler H.Ch. "A Search Processor for Data Base Management Systems" VLDB '78: Proceedings of the fourth international conference on Very Large Data Bases - Volume 4, September 1978, Pages 280–287
14) Schuster S.A., Nguyen, H.B., Ozkarahan, E.A. K.C. Smith, "RAP.2 - An Associative Processor for Databases and its Applications," IEEE Transactions on Computers, C-28, No. 6, June 1979. pp. 446-458
15) DeWitt D.J., "DIRECT - A Multiprocessor Organization for Supporting Relational Database Management Systems," IEEE Transactions on Coiputers. June 1979, pp. 395-406.
16) Madnick S.E. "The Infoplex Database Computer: Concepts and Directions," Proceedings of the IEEE Computer Conference, Feb. 1979l, pp. 168-176
17) Hell W. "RDBM - A Relational Database Machine: Archtecture and Hardware Design," Proceedings of the 6th Workshop on

Computer Architecture for Non-Numeric Processing, June 1981,
18) Missikoff M. "An Overview of the project DBMAC for a relational machine," Proceedings of the 6th Workshop on Computer Architecture for Non-Numeric Processing, Hyeres, France, June 1981.
19) Kalinichenko L.A., Ryvkin V.M. Database and Knowledge base Machines (Rus). Moscow, Nauka, 1990, 296 p.
20) DeWitt D.J., Gray J. Parallel database systems: the future of high performance database systems. Commun ACM. 1992;36(6):85–98.
21) Hurson A.R., Miller L.L., Pakzad S.H., Eich M.H., Shirazi B. Parallel architectures for database systems. Advances in Computers, Vol. 28, 1989, pp. 107-151.
22) Stonebraker M. "The Case for Shared Nothing," Database Engineering, Vol. 9, No. 1, 1986. pp. 4-9
23) Stonebraker M., Katz, R.H., Patterson, D.A., Ousterhout, J.K., The Design of XPRS, Fourteenth Int. Conf. on Very Large Data Bases, (Los Angeles, 1988), Morgan Kaufmann, 1988, pp. 318–330.
24) Bergsten B., Couprie, M., Lopez, M., DBS3: A Parallel Data Base System for Shared Store (Synopsis), in Issues, Architectures, and Algorithms (Proc. of the 2nd Int. Conf. on Parallel and Distributed Information Systems (PDIS 1993), San Diego, 1993), IEEE Comput. Soc., 1993, pp. 260–262.
25) Graefe G., Volcano—An Extensible and Parallel Query Evaluation System, IEEE Trans. Knowledge Data Engineering, 1994, vol. 6, no. 1, pp. 120–135.
26) Strickland J.P., Uhrowczik, P.P., Watts, V.L., IMS/VS: An Evolving System, IBM Systems J., 1982, vol. 21, no. 3, pp. 490–510.
27) Linder B., Oracle Parallel RDBMS on Massively Parallel Systems, in Issues, Architectures, and Algorithms (Proc. of the 2nd Int. Conf. on Parallel and Distributed Information Systems (PDIS 1993), San Diego, 1993), IEEE Comput. Soc., 1993, pp. 67–68.
28) Dubova N., Supercomputers nCube, Otkrytye sistemy, 1995, no. 2, pp. 42–47.
29) Kronenberg N.P., Levy, H.M., Strecker, W.D., VAXclusters: A Closely-Coupled Distributed System, ACM Trans. Comput. Systems, 1986, vol. 4, no. 2, pp. 130–146.
30) Nick J.M., Moore B.B., Chung J.-Y., Bowen N.S., S/390 Cluster Technology: Parallel Sysplex, IBM Systems J., 1997, vol. 36, no. 2, pp. 172–201.
31) Teradata: DBC/1012 Data Base Computer Concepts & Facilities, Teradata Corp. Document No. C02-0001-00, 1983.
32) Dewitt D.J., Ghandeharizadeh S., Schneider D.A., Bricker A. Hsiao H.-I., Rasmussen R. "The Gamma Database Machine Project," IEEE Knowledge and Data Engineering, Vol. 2, No. 1, March, 1990, pp. 44-62
33) Tandem Performance Group, "A Benchmark of Non-Stop SQL on the Debit Credit Transaction," Proceedings of the 1988 SIGMOD Conference, Chicago, IL, June 1988.
34) Alexander W., Copeland G.P. Process And Dataflow Control In Distributed Data-Intensive Systems. Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data, Chicago, Illinois, June 1-3, 1988. ACM Press, 1988, p. 90-98
35) Lorie R., Daudenarde J., Hallmark G., Stamos J., Young H., "Adding Intra-Transaction Parallelism to an Existing DBMS: Early Experience", IEEE Data Engineering Newsletter, Vol. 12, No. 1, March 1989., pp. 2–8.
36) Gibbs J, " Massively Parallel Systems, Rethinking Computing for Business and Science," Oracle, 1991, Vol. 6, No.1
37) Engler, S., Glasstone R., Hasan W., Parallelism and Its Price: A Case Study of Non-Stop SQL/MP, ACM SIGMOD Record, 1995, vol. 24, no. 4, pp. 61–71.
38) Clay D. Informix Parallel Data Query (PDQ), in Issues, Architectures, and Algorithms (Proc. of the 2nd Int. Conf. on Parallel and Distributed Information Systems (PDIS 1993), San Diego, 1993), IEEE Comput. Soc., 1993, pp. 71–72.
39) Page J., A Study of a Parallel Database Machine and Its Performance: The NCR/Teradata DBC/1012. Advanced Database Systems, Lecture Notes in Comput-

er Science (Proc. of the 10th British Natl. Conf. on Databases. BNCOD 10, Aberdeen, 1992), Springer, 1992, vol. 618, pp. 115–137.
40) Baru C.K. et al. DB2 Parallel Edition, IBM System J., 1995, vol. 34, no. 2, pp. 292–322.
41) Sokolinsky L.B. Survey of Architectures of Parallel Database Systems (Rus). Programming and Computer Software volume 30, No 6, pages 337–346 (2004)
42) Copeland G.P., Keller T., A Comparison of High-Availability Media Recovery Techniques, Proc. of the 1989 ACM SIGMOD Int. Conf. on Management of Data (Portland, 1989), ACM, 1989, pp. 98–109.
43) Graefe G., Query Evaluation Techniques for Large Databases, ACM Computing Surv., 1993, vol. 25, no. 2, pp. 73–169.
44) Hua K.A., Lee C., Peir J.-K., Interconnecting Shared-Everything Systems for Efficient Parallel Query Processing, Proc. First Int. Conf. on Parallel and Distributed Information Systems (PDIS 1991) (Miami Beach, 1991), IEEE-CS, 1991, pp. 262–270.
45) Pramanik S., Tout W.R. The NUMA with Clusters of Processors for Parallel Join, IEEE Trans. Knowledge Data Eng., 1997, vol. 9, no. 4, pp. 653–666.
46) Bouganim L., Florescu D., Valduriez P. Dynamic Load Balancing in Hierarchical Parallel Database Systems, Proc. 22th Int. Conf. on Very Large Data Bases (VLDB'96) (Mumbai, India, 1996), Morgan Kaufmann, 1996, pp. 436–447.
47) Xu Y., Dandamudi S.P. Performance Evaluation of a Two-Level Hierarchical Parallel Database System, Proc. Int. Conf. Computers and Their Applications, Tempe, Arizona, 1997, pp. 242–247.
48) Korneev V.V. Parallel'nye vychislitel'nye sistemy (Parallel Computing Systems), Moscow: Nolidzh, 1999.
49) Shmidt V. IBM SP2 Systems (Rus), Otkrytye Sistemy, 1995, no. 6, pp. 53–60.
50) Shnitman V. Fault-Tolerant Servers ServerNet (Rus), Otkrytye Sistemy, 1996, no. 3, pp. 5–11.
51) Sokolinsky L.B. Organization of Parallel Query Processing in Multiprocessor Database Machines with Hierarchical Architecture, Programmirovanie, 2001, no. 6, pp. 13–29.
52) Velicanu M., Litan D., Mocanu (Virgolici) A.-M., 2010. "Some Considerations about Modern Database Machines," Informatica Economica, Academy of Economic Studies - Bucharest, Romania, vol. 14(2), pages 37-44.
53) Eric G. Oracle and storage IOs, explanations and experience at CERN," 17th International Conference on Computing in High Energy and Nuclear Physics, Prague, Czech Republic, March 2009, pp. 21 – 27.

Базы данных, поддерживающие работу с массивами

БД массивов (БДМ) дает возможность представлять и манипулировать многомерными массивами однородных данных.

Считается, что предшественником БДМ является созванная в 1982 г. PICDMS [1] - СУБД для работы с рисунками, которая предоставляла возможность оперировать двумерными массивами с помощью процедурного языка.

В 1993 г. Майер и Вэнс [2] констатировали, что технология баз данных очень редко используется в научных приложениях в связи с тем, что СУБД не поддерживают структуры с упорядоченными данными в частности, массивы. Это заявление совпало во времени с началом активного развития исследований и разработок по БДМ.

Весомый вклад в развитие теории и практики СУБД массивов внес немецкий ученый Питер Бауманн (Peter Baumann). Он был первым, кто в 1994 г. предложил декларативный язык запросов для работы с многомерными массивами, который базируется на предложенной им же алгебре многомерных массивов [3, 4]. Разработанные алгебра и язык запросов легли в основу созданной в 1996 г. под его руководством первой СУБД массивов RasDaMan [5], которая поддерживала реляционную модель данных с дополнительным типом данных "многомерный массив" и специальным языком запросов RASQL, базирующимся на SQL. Согласно данным [6] объем данных, хранящихся на всех установках RasDaMan, приближается к петабайту


Питер Бауманн

Модели и языки

Было предложено множество формальных моделей и языков БДМ, анализ которых можно найти в [7, 8]. Приведем некоторые из них.

Алгебра карт (Map algebra) [9, 10] - алгебра, базирующаяся на множествах, разработана в начале 80-х годов Даной Томлин (Dana Tomlin) для манипулирования географическими данными. Представляет двумерные и трехмерные растровые данные. В ней производится категоризация операций над массивами в зависимости от того, сколько ячеек входного массива участвуют в создании ячейки выходного массива.

AFATL Image Algebra [11] - эта алгебра разработана для обработки изображений и получения статистической информации

AML (Array Manipulation Language) [12] - универсальный язык манипулирования массивами, базирующийся на предложенной авторами алгебре многомерных массивов. Отличительной чертой AML является понятие битовых шаблонов и шаблонно-ориентированных функций.

AQL (Array Query Language) [13. 14] - этот язык встраивает поддержку многомерных массивов в язык NCRA, который является расширением языка вложенного реляционного исчисления NRC.

Array Algebra [3, 4] - предлагается алгебраическая модель массива, базирующаяся на трех ортогональных примитивах, относительно которых предоставляется набор вспомогательных функций. Этот набор обуславливается используемой моделью данных (объектной или реляционной)

RAM [15, 16] - модель разработана в качестве расширения реляционной СУБД MonetDB [17].

Хранение массивов

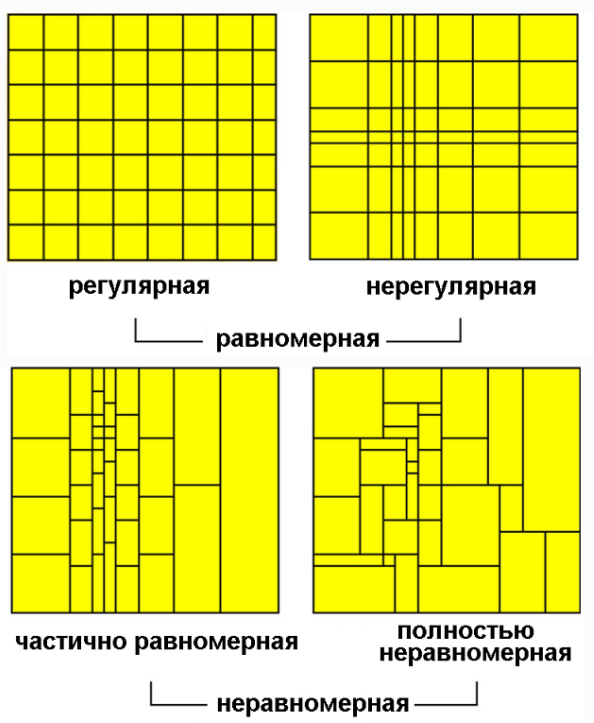
Обычно большие многомерные массивы разбиваются на подмассивы, которые образуют единицы доступа к ним. Такое разбиение получило название *мозаики* (tiling), а элементы мозаики - *плитки* (tile) [18]. Мозаика состоит из непересекающихся плиток, каждая плитка - многомерный подмассив исходного массива. Выделяют два основных вид мозаики - *равномерная* (aligned) и *неравномерная* (nonaligned). Равномерная мозаика для n-мерного массива означает, что она формируется гиперплоскостями, ортогональными осям n-мерного пространства, которые разбивают весь массив на "плитки". Если все плоскости находятся на одном расстоянии, то такая мозаика называется *регулярной равномерной*, в

противном случае - *нерегулярной*. В *неравномерной мозаике* (nonaligned tiling) некоторые плитки имеют стороны, не являющиеся продолжением сторон соседних плиток. В *частично равномерной* (partially aligned) мозаике плитки выровнены, по крайней мере, по одному из измерений, а в *полностью неравномерной* (totally nonaligned) таких измерений нет. На следующем рисунке, взятом из [18], приводится пример графической интерпретации этих четырех категорий мозаики для двумерного пространства.

Архитектура реализации

Выделяются следующие варианты архитектуры реализации систем БД массивов:

– полнофункциональные системы БД массивов, реализованные с нуля (RasDaMan [5], SciDB [19], MonetDB/SciQL [20]);
– реализованные в виде дополнительных уровней в существующих СУБД (EXTASCID [21, 22]);



– реализованные в виде объектно-реляционных расширений (PostGIS Raster [23] Teradata Arrays [24], Oracle GeoRaster [25]),

Было предложено два способа "внедрения" массивов в реляционные БД:

– добавление массивов в виде нового типа столбца (RasDaMan, Teradata, Oracle, PostGIS Raster, ISO SQL);
– представление массивов в виде таблицы (SciQL и SciDB).

В 2007 году на симпозиуме по экстремально большим базам данных (XLDB) представителями науки и промышленности был сделан вывод, что существующие СУБД не в состоянии манипулировать объемами данных, которые появятся в ближайшем будущем. Был также сделан вывод о необходимости разработки СУБД нового поколения, которые должны удовлетворять, в частности, следующим требованиям [26]:

– модель данных основывается на многомерных массивах, а не на кортежах;
– модель хранения базируется на версионности, а не на обновлении значений;
– масштабируемость до сотен петабайт и высокая отказоустойчивость;
– СУБД является свободно распространяемым программным обеспечением.

В ответ на это обращение в 2008 году, был запущен международный проект под руководством Майкла Стоунбрейкера по созданию новой СУБД, получившей название SciDB. В 2010 г. была выпущена первая публичная версия SciDB [27]. Ее архитектура основана на модифицированном ядре Postgres. SciDB предназначена для хранения, обработки и анализа сверхбольших объёмов многомерных распределенных массивов научных данных, масштабируемых на тысячи серверов [28]. Хранение данных организовано в виде многомерных вложенных массивов, для обработки которых разработаны языки AQL (Array Query Language) и AFL (Array Functional Language).

Другие системы БД массивов

SciQL [20]. Основанный на SQL и использующий массивы язык запросов для научных применений. Расширяет колончатую СУБД MonetDB операторами над массивами [29, 30], позволяя тем самым MonetDB эффективно функционировать как база данных массивов.

EXTASCID [21, 22]. Это полная и расширяемая система для обработки научных данных. Он поддерживает как массивы, так и реляционные данные. Построена на основе массивно-параллельной архитектуры GLADE для агрегирования данных.

PostGIS Raster [23] (ранее известная как WKT Raster) позволяет поддерживать растровые данные в системе PostGIS. Это обеспечивается определением нового типа данных RASTER и дополнительного набора SQL-функций, работающих с векторными и растровыми данными.

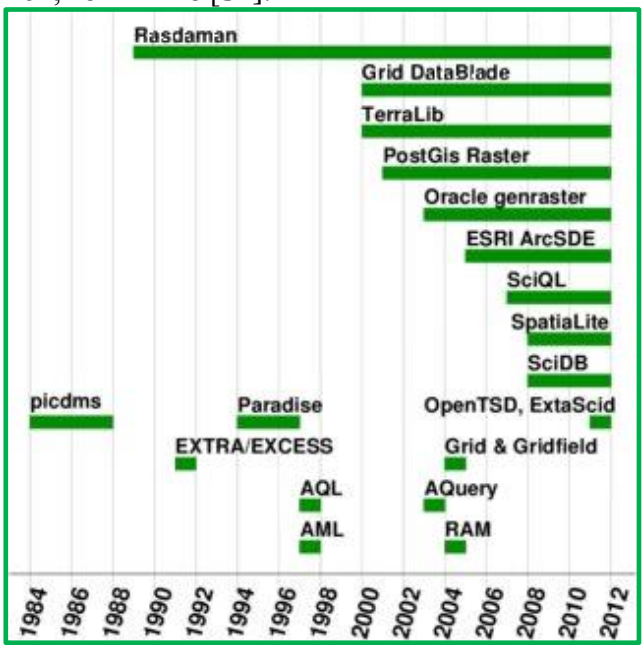
Oracle GeoRaster [25] — это встроенная в Oracle Spatial возможность по хранению, индексированию, анализу и доставке растровых изображений (например, спутниковых снимков), данных типа grid-данных, а также связанных с ними метаданных. Эти типы данных можно использовать для хранения многомерных grid-слоев и электронных изображений, которые могут быть привязаны для позиционирования на поверхности Земли или в локальной системе координат.

Teradata Arrays [24]. Недавно Teradata вела в свою СУБД массивы в виде самостоятельного типа данных.

В 1918 г. в ISO SQL была включена поддержка многомерных массивов данных [31] в виде специального типа данных.

В заключение отметим, что Альянс по Исследовательскими Данным (RDA - Research Data Alliance) представил в 2021 году исчерпывающий обзор по базам данных массивов и связанными с ними технологиями [8].

В качестве иллюстрации истории развития систем БД массивов приведем рисунок, взятый из [32].



Литераура

1) Chock M., Cardenas A., Klinger A. Database structure and manipulation capabilities of a picture database management system (PICDMS). IEEE ToPAMI, 6(4):484–492, 1984
2) Maier D., Vance B. A call to order. In PODS ’93: Proceedings of the twelfth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, 1993, pp. 1–16
3) Baumann P. Management of Multidimensional Discrete Data. VLDB Journal, Special Issue on Spatial Database Systems, 1994, Vol 4, No. 3, pp. 401–444
4) Baumann P. A Database Array Algebra for Spatio-Temporal Data and Beyond, 4th International Workshop on Next Generation Information Technologies and Systems (NGITS '99), July 5–7, 1999, Zikhron Yaakov, Israel, Lecture Notes on Computer Science 1649, Springer Verlag, pp. 76 – 93.
5) Baumann P., Dehme A., Furtado P., Ritsch R., Widmann N. The Multidimensional Database System RasDaMan. Conference: SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA. ACM SIGMOD Record, 1998, Vol. 27, No. 2, pp 575–577
6) EarthServer: The EarthServer Initiative. www.earthserver.eu
7) Baumann P., Holsten S. (2011) A Comparative Analysis of Array Models for Databases. In: Kim T. et al. (eds) Database Theory and Application, Bio-Science and Bio-Technology. BSBT 2011, DTA 2011. Communications in Computer and Information Science, vol 258. pp 80-89 Springer, Berlin, Heidelberg.
8) Baumann, P., Misev, D., Merticariu, V., Bang Pham Huu. Array databases: concepts, standards, implementations. Journal of Big Data, vol 8, No. 1 (2021).
9) Tomlin D. A Map Algebra. Harvard Graduate School of Design, 1990.
10) Mennis, J., Viger, R., Tomlin, C.D.: Cubic Map Algebra Functions for Spatio-Temporal Analysis. Cartography and Geo-

graphic Information Science, Vol. 32, No. 1, 2005, pp. 17-32.
11) Ritter G, Wilson J, Davidson J. Image Algebra: An Overview. Computer Vision, Graphics, and Image Processing. Vol. 49, No. 3, 1990, pp. 297-331.
12) Marathe A, Salem K. A language for manipulating arrays. In: Proceedings of the 23th International Conference on Very Large Data Bases; 1997. p. 46–55.
13) Libkin, L., Machlin, R., Wong, L.: A query language for multidimensional arrays: design, implementation and optimization techniques. Proc. ACM SIGMOD'96, Montreal, Canada/ ACM SIGMOD Record, 1996, vol. 25, No. 2, pp. 228–239
14) Machlin R. Indexbased multidimensional array queries: safety and equivalence. In L. Libkin, editor, PODS, pp. 175–184. ACM, 2007.
15) van Ballegooij A.R., de Vries A.P., Kersten M. RAM: Array processing over a relational DBMS. Technical Report INS-R0301, CWI (March 2003)
16) van Ballegooij A.R. RAM: A multidimensional array DBMS. In W. Lindner, M. Mesiti, C. Turker, Y. Tzitzikas, and A. Vakali, editors, EDBT Workshops, volume 3268 of Lecture Notes in Computer Science, pp. 154–165. Springer, 2004.
17) Cornacchia R., Heman S., Zukowski M., de Vries A., Boncz P. Flexible and efficient IR using array databases, VLDB Journal. 7(1): 151–168.
18) Furtado P., Baumann P. Storage of multidimensional arrays based on arbitrary tiling. In Proceedings of the 15th International Conference on Data Engineering, pp. 328–336. IEEE Computer Society, 23-26 March 1999
19) Stonebraker M., Brown P., Poliakov A., Raman S. (2011) The Architecture of SciDB. In: Bayard Cushing J., French J., Bowers S. (eds). Proceedings of the 23rd International Conference on Scientific and Statistical Database Management; 2011 pp 1-16
20) Kersten M.L, Zhang Y., Ivanova M., Nes N. SciQL, a query language for science applications. Proceedings, EDBT/ICDT 2011 Workshop on Array Databases: Uppsala, Sweden, March 25, 2011, pp
21) Cheng, Y., Rusu, F. Formal representation of the SS-DB benchmark and experimental evaluation in EXTASCID. Distrib Parallel Databases, 2015, vol. 33, No. 3, pp. 277–317
22) Cheng Y., Rusu, F. Astronomical data processing in EXTASCID. SSDBM: Proceedings of the 25th International Conference on Scientific and Statistical Database Management, 2013 Article No.: 47, pp. 1–4
23) Tollefsen, Andreas Forø (2013) PostGIS 2.0 og Raster, Kart og Plan 73(3), pp. 159–164.
24) Teradata. Multidimensional array options. - https://docs.teradata.com/r/VrFCOAaniAIfrJsA51oQJA/ZMY8sE8cSytuSPtp8QnuFA
25) Oracle: GeoRaster. - http://docs.oracle.com/cd/B19306_01/appdev.102/b14254/geor_intro.htm.
26) Becla J., Lim K.T. Report from the first Workshop on Extremely Large Databases, Data Science Journal, 2008, Vol. 7, pp. 1-13
27) Stonebraker M., Brown P., Poliakov A., Raman S. (2011) The Architecture of SciDB. In: Bayard Cushing J., French J., Bowers S. (eds). Proceedings of the 23rd International Conference on Scientific and Statistical Database Management; 2011 pp. 1-16
28) Bauman National Library. SciDB. - https://ru.bmstu.wiki/SciDB
29) Ivanova M, Kersten M.L, Manegold S. Data vaults: a symbiosis between Database technology and scientific file repositories. Proc. Intl. Conference on Scientific and Statistical Database Management (SSDBM). Athens. 2012, pp. 485-:494.
30) Zhang Y, Kersten M.L, Ivanova M, Nes N. SciQL, bridging the gap between science and relational DBMS. In: Desai B.C, Cruz I.F, Bernardino J, editors. Proceedings of the 15th Symposium on International Database Engineering and Applications; 2011. pp. 124–133.
31) "ISO/IEC DIS 9075-15 Information technology - Database languages - SQL - Part 15: Multidimensional arrays (SQL/MDA)"
32) Baumann P., Stamerjohanns H. (2014) Towards a Systematic Benchmark for Ar-

ray Database Systems. In: Rabl T., Poess M., Baru C., Jacobsen H.A. (eds) Specifying Big Data Benchmarks. pp 94-102.

Статистические базы данных

Под *статистическими базами* (СБД) данных понимают такие БД, которые предоставляют возможность получать, хранить и обрабатывать агрегированные данные, то есть данные, полученные с помощью различных способов обобщения, группирования, классификации.

Исследования СБД начались в 1970-х и получили наибольшее развитие в 1980-х еще до появления OLAP и продолжают развиваться по настоящее время.

Статистические модели данных

Статистические данные более абстрактны по сравнению с обычными, а операции имеют другую семантику. В СБД анализ производится с использованием агрегированных данных, полученных из необработанных. Сводные данные могут быть разных форм, которые не поддерживаются традиционными СУБД. Более того, реляционная модель в чистом виде также не подходит для обработки таких данных. Основная причина - многомерность статистических данных. Таким образом, для СБД нужны либо новые структуры данных и операции над ними, либо расширять реляционную модель данных, чтобы иметь возможность представлять отношения над множествами и новые операторы к ним [1]. В СБД определено три типа моделей данных: *графические, табличные* и *многомерные*. Дадим краткое описание статистических моделей данных, подробная информация о которых приведена в работах [2, 3]:

- SUBJECT [4] - представлена графическая модель системы SUBJECT;
- SAM (Semantic Association Model) [5] - модель разработана для моделирования как научных статистических данных, так и бизнес-ориентированных данных;
- GRASS (Graphical Approach for Statistical Summaries) [6] - является расширением SUBJECT. Для представления модели используется ориентированный, направленный, ациклический граф;
- CSM (Conceptual Statistical Model) [7] - используются две разные, но дополняющие друг друга модели данных для опи-

сания элементарных и сводных данных, а именно ER-модель Чена и переопределенную модель GRASS;
- STORM (Statistical Object Representation Model) [8] - графическая модель, в которой логическое представление отделено от физической структуры статистических таблиц;
- MEFISTO [9] - функциональная модель, базирующаяся на структуре, названной "статистическая сущность", и на множестве операций, составляющих алгебру манипулирования данными этой структуры;
- Расширенная реляционная модель данных с включением в реляционную алгебру дополнительных статистических операторов [10];
- Темпоральная статистическая модель данных [11].

Статистические операторы (алгебры)

В статьях по СБД предлагается много подходов по определению операторов, которые бы соответствовали выбранной структурной модели. В работе [12] вводятся статистические операторы, аналогичные реляционной алгебре, но с семантикой, характерной для многомерных объектов. Далее, вводится понятие полноты по аналогии с реляционной полнотой, и показывается полнота предложенной алгебры. В работе [1] также предлагается расширение реляционной модели данных введением отношений над множествами и операторы к ним. Еще одним примером алгебры, которая зависит от выбранной статистической модели, является работа [13], в которой используется двумерная модель представления статистических данных. В 1997 г. в [14] был предложен вариант расширения SQL функциональными возможностями OLAP для получения итоговых значений в многомерном пространстве. В [15, 16] предлагаются многомерные модели данных и операторы. Статьи [9, 17] также посвящены операторам в СБД. В работе [18] представлена алгебра статистических данных.

Метаданные

Считается, что статистические данные имеют два типа атрибутов [19]: *сводные атрибуты*, представляющие собой результаты применения к исходным данным агрегирующих функций, и *дескриптивные атрибуты*, которые эти сводные данные описывают, также называющиеся метаданными. Правильно организованные, классифицированные и описанные метаданные приносят большую пользу в понимании сути сводных данных, в связи с чем их эффективное использование метаданных очень важно в СБД. Дополнительную информацию по этому поводу можно получить в работах [20–24].

Системы и языки запросов статистических баз данных

В обзоре [25] дается всесторонний анализ систем и языков запросов статистических баз данных на основании таксономии, предложенной в [26]. Перечислим их, отсылая заинтересованных читателей к указанным статьям для детального ознакомления.

Статистические системы управления базами данных (ССУБД), построенные на основе традиционных СУБД

Большинство ССУБД данной категории построены на основе реляционных СУБД. К ним относятся:
- *STRAND* [27] базируется на ER-модели Чена, является производным от CABLE [28] и построен на основе реляционной СУБД INGRES. Запросы STRAND транслируются в язык QUEL и выполняются в INGRES.
- *HSDB* [29] является ССУБД, построенной на основе реляционной системы Model 204 [30]. HSDB поддерживает сводные таблицы и предоставляет ограниченный набор операций над ними. Может выполнять процедуры статистического анализа над реляционными и сводными таблицами
- *Расширенная РМД* [31]. Расширяется модель Кодда для представления статистических данных путем введения "статистической реляционной таблицы". Для

нее расширяются реляционные операции и вводятся новые статистические операторы. Предлагается язык запросов, имеющий сходные черты с QBE.
- *SYSTEM/K* [32] Объектно-ориентированная система управления базами знаний, построенная на основе системы SQL/DS. Имеет развитые возможности по управлению метаданными и ограниченный перечень статистических функций.
- *GRAFSTAT* [33]. Прикладная система предназначена для анализа данных с помощью функций прикладной статистики и графического представления результатов. Имеет интерфейс с DB2 и SQL/DS через SQL.
- SUBYL [34], *PASTE* [35], *GPI* [36], *PEPIN-SICLA* [37] являются примерами систем, которые используют традиционную СУБД, статистический пакет и графический пакет для создания системы управления статистическими данными.

Самостоятельно разработанные ССУБД

Они группируются на следующие шесть подкатегорий согласно используемой модели и языка запросов:
- Системы на базе реляционной модели и реляционных языков запросов. Они предлагают собственные методы физической организации данных, средства концептуального моделирования, подходящие для ССУБД, и возможности использования агрегирующих функций в языках запросов. К ним относятся:
  - RAPID [38] и CAS SDB [39], используют реляционную алгебру.
  - ABE [40], использует реляционное исчисление.
  - SIR/SQL [41], GENISYS [42], CANTOR [43] - используют SQL.
  - В [44] представлен язык запросов статистической обработки неполной информации
  - В системе July [45] используется универсальный реляционный интерфейс для интерпретации статистических запросов
  - В статье [46] описана статистическая модель данных и ее применение в СБД.
- Системы на базе иерархической и сетевой моделей. Примерами являются: SIR/DBMS [47], TPL и TPLDCS [48], BROWSE [49].
- Формальные расширения реляционной модели: ABE [40], SSDL [50], SSDB [51].
- ССУБД и языки с графическим внешним интерфейсом. Системы данной категории имеют графические двумерные или табличные языки запросов. Примерами являются: SUBJECT [4], GRASS [6], ABE [40], GUIDE [52], STBE [53], ALDS [54], GRASP [55].
- Естественно-языковый интерфейс пользователя: LIDS 86 [56].
- Языки запросов, которые вычисляют агрегированную информацию из темпоральных данных. Примерами являются TQUEL [57], HQUEL [58], TBE [59], TEER [60], расширенная реляционная алгебра Тансела [61].

В заключение отметим, что на основании анализа литературы можно сказать, что, по крайней мере, на начальном пути развития дисциплины "статистические базы данных" большой вклад внесли турецко-американские ученые Зехра Мерал Озсойоглу (Zehra Meral Ozsoyoglu) и Гюльтекин Озсойоглу (Gultekin Ozsoyoglu)

Мерал Озсойоглу специализируется по базам данных. В 2011 г. она получила звание "Действительный член ACM" (ACM Fellow) за "большой вклад в системы управления базами данных". В 2018 г. она получила премию ACM SIGMOD Contributions Award за "преданное служение сообществу баз данных». В награде упоминается ее работа в качестве главного редактора ACM Transactions on Database Systems и Proceedings of the VLDB Endowment, а также в качестве председателя программного комитета конференции VLDB и Симпозиума по принципам систем баз данных.


Мерал Озсойоглу

Литература

1) Ozsoyoglu G., Ozsoyoglu Z.M., Matos V., Extending relational algebra and relational calculus with set-valued attributes and aggregate functions," ACM Transactions on Database Systems, 1987, vol. 12, pp. 566–592.
2) Srivastava J., Ngo H.Q. Statistical Databases. Technical Report TR 99-009, 1999, Department of Computer Science and Engineering, University of Minnesota. - https://conservancy.umn.edu/bitstream/handle/11299/215365/99-009.pdf?sequence=1&isAllowed=y
3) Michalewicz Z. (ed.) Statisticai and Scientific Databases. Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, 11991, 544 p.
4) Chan P., Shoshani A. SUBJECT: A directory driven system for large statistical databases. In VLDB '81: Proceedings of the seventh international conference on Very Large Data Bases - Volume 7, 1981, pp. 553–563
5) Su S. SAM: A semantic association model for corporate and scientific-statistical databases. Journal of Information Science, pp. 151–199, 1983.
6) Rafanelli M., Ricci F.L. A visual interface for browsing and manipulating statistical entities. In Proceedings of the Fifth International Conference on Scientific and Statistical Database Management, pp. 1990, 163–182,
7) Battista G.D., Batini C. Design of statistical databases: a methodology for the conceptual step. The Journal of Information Systems, vol. 13, no. 4, pp. 407–422, 1988
8) Rafanelli M., Shoshani A. STORM: A statistical object representation model. In Proceedings of the Fifth International Conference on Scientific and Statistical Database Management, pp. 14–29, 1990.
9) Rafanelli M., F.L. Ricci, "Mefisto: A functional model for statistical entities," IEEE Transactions on Knowledge and Data Engineering, vol. 5, No. 4, pp. 670–681, Aug. 1993.
10) Ghosh S.P. (1989) Statistical relational model. In: Rafanelli M., Klensin J.C., Svensson P. (eds) Statistical and Scientific Database Management. SSDBM 1988. Lecture Notes in Computer Science, vol 339. Springer, Berlin, Heidelberg. pp. 338-355
11) Shoshani A., Kawagoe K. Temporal data management. In Proceedings of the Twenty-Second International Conference on Very Large Data Bases (VLDB), 1986 pp. 79–90.
12) Meo-Evoli L., Ricci F.L., Shoshani A., On the Semantic Completeness of Macro-Data Operators for Statistical Aggregation, SSDBM 1992, pp. 239-258.
13) Ozsoyoglu G., Ozsoyoglu Z.M., Malta F. A Language and a Physical Organization Technique for Summary Tables. SIGMOD, 1985: pp. 3-16.
14) Gray J., Bosworth A., Layman A., Pirahesh H. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. Data Mining and Knowledge Discovery, 1997, Vol. 1, No. 1, pp. 29–53
15) Agrawal R., Gupta A., Sarawagi S. Modeling Multidimensional Databases. ICDE '97: Proceedings of the Thirteenth International Conference on Data EngineeringApril, 1997, pp. 232–243.
16) Vassiliadis P. "Modeling multidimensional databases, cubes and cube operations," Proceedings. Tenth International Conference on Scientific and Statistical Database Management (Cat. No.98TB100243), 1998, pp. 53-62,
17) Gentle J.E., Bell J. Special Data Types and Operators for Statistical Data. IEEE Database Eng. Bull., 1984, Vol. 7, No. 1, pp. 34-37
18) Fortunato E., Rafanelli M., Ricci F., Sebastio A.,An algebra for statistical data. In SSDBM'86: Proceedings of the 3rd international workshop on Statistical and scientific database management, 1986, pp. 122–134
19) Bezenchek A., Rafanelli M., Tininini L. A data structure for representing aggregate data. In Proceedings: Eighth International Conference on Scientific and Statistical Database Systems, Stockholm, Sweden (P. P. Svensson and J. C. J. C. French, eds.),, IEEE Computer Society Press, 1996, pp. 22–31

20) van den Berg G.M., E. de Feber. Definition and use of metadata in statistical data processing. In Proceedings of the 6th International Conference on Statistical and Scientific Management, (Ascona, Switzerland), 1992, pp. 290–306
21) Kent J.P., Schuerhoff M. Some thoughts about a metadata management system, in Proceedings of the 9th International Conference on scientific and Statistical Databases, (Olympia, WA), pp. 155–164, IEEE Press, Aug. 1997.
22) Westlake A. "A simple structure for statistical metadata," in Proceedings of the 9th International Conference on scientific and Statistical Databases, (Olympia, WA), pp. 186–195, IEEE Press, Aug. 1997.
23) Ghosh S. P. Statistical Metadata. In Kotz-Johnson Encyclopedia of Statistical Science, Vol.8, John Wiley & Sons Inc. Publ., 1988
24) Signore M., Scanu M., Brancato G. Statistical metadata: a unified approach to management and dissemination. Journal of Official Statistics, 2015, Vol. 31, No 2, pp. 325-347
25) Tansel A. Query languages for statistical databases. Statistics and Compuing. 1995. Vol. 5, No. 1, pp. 59-72
26) Ozsoyoglu G., Ozsoyoglu, Z. M. Statistical database query languages. IEEE Transactions on Software Engineering. 1985, vol 11, No. 10, pp. 1071-1080.
27) Johnson R. Modeling summary data. In Proceedings of the ACM SIGMOD Conference, (Ann Arbor, Michigan), pp. 93–97, 1981.
28) Shoshani A. CABLE: A Chain-Based Language for the Entity-Relationship Model. Proceedings of the 1st International Conference on the Entity-Relationship Approach to Systems Analysis and Design, 1980, pp. 465–466
29) Ikeda H., Kobayashi Y. Additional facilities of a conventional DBMS to support interactive statistical analysis. In Proceedings of the 1st LBL Workshop on Statistical Database Management, Lawrence Berkeley Lab, Berkeley, CA, Dec. 1981, pp. 25–36
30) Computer Corporation of America. File Manager's Technical Reference Manual, Model 204 Database Management System. Computer Corporation of America, Cambridge, MA, 1979
31) Ghosh S.P. Statistical relational tables for statistical database management, IEEE Transactions of Software Engineering, 1986, vol. SE-12, No. 12, pp. 1106–1116.
32) Maier D., Cirilli C. SYSTEM/K: A knowledge based management system. In Proceedings of the Second Int. Workshop on Statistical Database Management, Los Altos, CA, Sept. 1983, pp. 287–294
33) Stein D.M. A database interface to an integrated dataanalysis and plotting tool. In Proceedings of the 3rd International Workshop on Statistical and Scientific Database Management, Luxemburg, 1986, pp. 98–106
34) Heiler S., Bergman R.F. SIBYL: An economist'sworkbench. In SSDBM'83: Proceedings of the 2nd International Workshop on Statistical Database Management, Los Altos, CA., 1983, pp. 73–79
35) Weiss S.E., Weeks P.L. PASTE-a tool to put application systems together easily. In SSDBM'83: Proceedings of the 2nd International Workshop on Statistical Database Management LosAltos CA, 1983, pp. 119–123
36) Hollabaugh L.A., Reinwald L.T. GPI: a statistical package/database interface. SSDBM'81: In Proceedings of the 1st International Workshop on Statistical Database Management MenloPark CA, 1981, pp. 78–87
37) Boufares P., Elkabbaj Y., Joiner G., Ounally H. Laversion SM90 du SGBD relationnel PEPIN. Journes SM90, Versailles, France, 1985
38) Turner M. T., Hammond R. Cotton P. A DBMS for large statistical databases. In VLDB '79: Proceedings of the fifth international conference on Very Large Data Bases - Volume 5, 1979, pp. 319–327
39) Johji S., Sato H. Statistical database research project in Japan and the CAS SDB project. In SSDBM'83: Proceedings of the 2nd international workshop on Statistical Database Management, 1983 pp. 325–330.
40) Klug A. ABE – a query language for constructing aggregates-by-example. In SSDBM'81: Proceedings of the 1st LBL

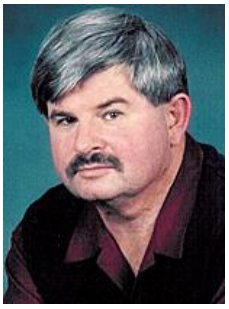
Workshop on Statistical Database Management, 1981, pp. 190–205
41) Anderson G., Snider T., Robinson B., Toporek J. An integrated research support system for interpackage communication and handling large volumne output from statistical database analysis operation. In SSDBM'83: Proceedings of the 2nd international workshop on Statistical Database Management, 1983, pp. 104–110
42) Dintelman S.M., Maness A.T. An implementation of a query language supporting path expressions. In Proceedings of the 1982 ACM SIGMOD International Conference on Management of Data, (Orlando, Florida), 1982., pp. 87–93
43) Karasolo I., Sevenson P. An overview of CANTOR – a new system for data analysis. In SSDBM'83: Proceedings of the 2nd international workshop on Proceedings of the Second International Workshop on Statistical Database ManagementSeptember 1983 pp. 315–324
44) Chan C., Michalewicz Z. A query language capable of handling incomplete information and statistics. In SSDBM'86: Proceedings of the 3rd international workshop on Statistical and scientific database management, 1986, pp. 107–115
45) D'attri A., Ricci F.L. Interpretation of statsitical queries to relational databases. In SSDBM'1988: Proceedings of the 4th international conference on Statistical and Scientific Database Management, 1988, pp. 246–258
46) Chen M., McNamee L., Melkanoff M. A model of summary data and its applications in statistical databases. In SSDBM'1988: Proceedings of the 4th international conference on Statistical and Scientific Database Management, 1988, pp 356-387
47) Anderson G., Snider T., Robinson B., Toporek J. An integrated research support system for interpackage communication and handling large volumne output from statistical database analysis operation. In SSDBM'83: Proceedings of the Second International Workshop on Statistical Database Management, 1983, pp. 104–110.
48) Weiss W., Weeks P., Byrd P. Must we navigate through databases. In SSDBM'81: Proceedings of the 1st LBL Workshop on Statistical Database Management, Lawrence Berkeley Lab, Berkeley, CA, Dec. 1981, pp. 111–122
49) Hendrix G.G., Sacerdoti E.D., Sagalowicz D., Slocum J. Developing a natural language interface to a complex system. ACM Transactions on Database Systems, 1978, Vol. 3, No. 2,, pp. 105–147
50) Brown W., Navathe S., Su S. Complex data types and a data manipulation language for scientific and statistical databases. In SSDBM'83: Proceedings of the 2nd international workshop on Statistical Database Managemen, 1983, pp.188–195
51) Ozsoyoglu G., Ozsoyoglu Z.M. Features of a system for statistical databases. In SSDBM'83: Proceedings of the 2nd international workshop on Statistical Database Management, 1983, pp. 9–18.
52) Wong H.K.T., Kuo I. GUIDE: Graphical user interface for database exploration. Iin Proceedings of the 8th Conference on Very Large Databases, Morgan Kaufman pubs. (Los Altos CA), McLeod and Villasenor, Mexico City, 1982, pp. 22-32
53) Ozsoyoglu Z. M., Ozsoyoglu G. Summary-table-by-example: A database query language for manipulating summary data. In Proceedings of the International Conference on Data Engineering, (Los Angeles, CA), 1984, pp. 193–202.
54) Thomas J., Hall D. ALDS project: Motivation, statistical database management issues, perspectives, and directions, In SSDBM'83: Proceedings of the 2nd international workshop on Statistical Database ManagementSeptember, 1983, pp.82–88
55) Catarci T., Santucci G. GRASP: A graphical system for statistical databases. In Proceedings of the Fifth International Conference on Scientific and Statistical Database Management, (Charlotte, NC), 1990, pp. 148–162
56) Sato H. A data model, knowledge base and natural language processing for sharing a large statistical database. Iin Proceedings of the 4th International Working Conference SSDBM on Statistical and Scientific Database Management, 1988, pp. 207–225

57) Snodgrass R. T., The temporal query language TQuel. In Symposium on Principles of Database Systems, 1984, pp. 204–213.
58) Tansel A.U., Arkun M.E. HQUEL, A query language for historical relational databases. In SSDBM'86: Proceedings of the 3rd international workshop on Statistical and scientific database management, 1986, pp. 135–142
59) Tansel A., Arkun M.E., Ozsoyoglu G. Time-by-example query language for historical databases. IEEE Transactions on Software Engineering (SE), 1989, vol. 15, No. 4, pp.464-478
60) Elmasri R., Kouramajian V. A temporal query language based on conceptual entities and roles. ER '92: Proceedings of the 11th International Conference on the Entity-Relationship Approach: Entity-Relationship Approach, 1992, pp. 375–388
61) Tansel A.U. A statistical interface for historical relational databases. In Proceedings of the Third International Conference on Data EngineeringFebruary, 1987, pp 538–546
62) Reznichenko V.A. Workig with windows in SQL (Rus). Software Engineering, 2011, vol. 7, No 3, pp. 35-48

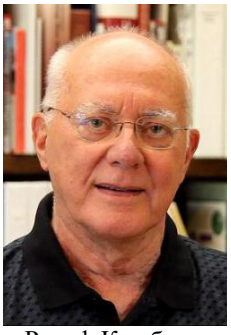
Хранилища данных

Задача сбора информации из разных источников не является новой. В конце прошлого века получила распространение концепция построения хранилищ данных (DataWarehouse).

Термин "хранилище данных" (DWH-Data WareHouse) впервые появился в начале 1980-х годов, когда исследователи IBM Пол Мерфи (Paul Murph) и Барри Девлин (Barry Devlin) разработали хранилище бизнес-данных. Однако считается, что основоположником хранилищ данных является Вильям Х. Инмон (William H. Inmon), который приступил к исследованиям в этой области в 1983 г., а собственно концепция хранилищ данных была изложена им в 1990 г. в монографии [1], которая стала библией хранилищ данных и которая положила начало развитию индустрии хранилищ данных. Он организовал первую конференцию, впервые организовал тематический раздел в журнале.


Вильям Инмон

Еще одним основоположником по праву считается Ральф Кимбелл (Ralph Kimball). Он является одним из первых архитекторов хранилищ данных, и его методология, также известная как пространственное моделирование или методология Кимбелла, стала фактическим стандартом в области поддержки принятия решений. В 1990 г. компания Red Brick Systems, основанная Кимбеллом, разработала Red Brick Warehouse - компактную стандартную реляционную базу данных на основе SQL для приложений DWH и бизнес-аналитики. Его монография [2], первое издание которой было в 1997 г., является бестселлером по настоящее время.


Ральф Кимбелл

Не смотря на то, что эти ученые порой придерживались противоположных взглядов на DWH, они основали и существенно обогатили науку DWH. Сравнение взглядов

на DWH этих двух ученых приведено в работе [3] и многих других исследователей их творческого наследия

К пионерам исследователей по DWH, которые опубликовали свои монографии в середине 90-х годов, также относятся Брэкетт [4], Гилл и Рао [5], По [6].

Согласно определению Инмона *хранилище данных - это предметно ориентированная, интегрированная, поддерживающая хронологию и неизменяющаяся (постоянная) коллекция данных, созданная для поддержка процесса принятия решений руководством* [1] Также считается, что в широко смысле хранилище данных - это совокупность технологий, которые позволяют руководству принимать решения быстрее и качественнее и в связи с этим они являются составляющими автоматизированных систем поддержки принятия решений.

DWH предполагают интеграцию гетерогенных (неоднородных) БД. Но в отличие от традиционного подхода создания гетерогенных БД, который предполагает создание оболочек и посредников, преобразующих стандартные запросы к виду, воспринимаемому каждой из интегрированных БД, в DWH информация из многих гетерогенных источников предварительно преобразуется, интегрируется и сохраняется в едином хранилище данных.

Основная задача традиционных БД, которые получили название операционных, эффективно выполнять транзакции с учетом активного обновления БД с тем, чтобы поддерживать ее целостность. Эти БД были отнесены к классу систем оперативной обработки транзакций (online transaction processing OLTP). С другой стороны, системы DWH не предполагают динамического обновления, для них не существует проблема поддержания целостности и они предназначены для пользователей, которые осуществляют анализ данных с целью принятия решений. Системы такого класса получили название систем оперативной аналитической обработки (online analytical processing - OLAP). Термин OLAP ввёл Эдгар Кодд в публикации в журнале Computerworld в 1993 году [7], в которой он определил OLAP как средство динамического анализа, синтеза и консолидации больших объемов многомерных данных, сформулировал концептуальные положения OLAP, описал архитектуру, выделил фундаментальные компоненты и предложил 12 принципов аналитической обработки, по аналогии с 12 правилами для реляционных баз данных.

В начале 1995 г. Найджел Пендс (Nigel Pendse), не удовлетворенный критериями Кодда, сформулировал альтернативные 5 правил принадлежности систем к категории OLAP [8], которые были названы тестом FASMI - аббревиатура из первых букв слов фразы "Fast Analysis of Shared Multidimensional Information" (быстрый анализ совместно используемой многомерной информации). Это определение также является весьма популярным в среде специалистов OLAP.


Найджел Пендс

Наконец, Совет OLAP (OLAP Council), созданный в 1995 г. дал следующее развернутое определение OLAP:

"*Оперативная аналитическая обработка (OLAP) - это категория программных технологий, которые позволяют аналитикам, менеджерам и руководителям получать представление о данных за счет быстрого, согласованного, интерактивного доступа до представленной в различном виде информации, преобразованной из исходных данных, с тем, чтобы они осознавали реальное положение дел на предприятии*" [9].

Архитектура DWH

Было предложено множество различных архитектурных решений DWH [10-21], каждое из которых обладает своими специфическими особенностями. В работе [22] проведен анализ 73 архитектур DWH. На базе предложений, высказанных в [10, 11], представим обобщающую архитектуру DWH. DWH имеет трехуровневую архитектуру.

– Нижний уровень представляет собой БД DWH. Она поддерживает выбранную модель данных DWH и предоставляет средства ведения этой БД.

– Средний уровень выполняет функции OLAP. Он обычно представлен следующими 4 типами [12, 13]:
  - *Реляционный OLAP* (ROLAP) – расширенная реляционная СУБД, которая отображает операции многомерной модели данных в стандартные операции реляционной алгебры.
  - *Многомерный OLAP* (MOLAP) - СУБД, который непосредственно поддерживает многомерную модель данных и ее операции;
  - *Гибридный OLAP* (HOLAP), сочетающий в себе свойства предыдущих двух видов.
  - *Специализированный SQL-сервер* - обладает развитыми возможностями языка запросов SQL для работы с DWH-схемами (звезда, снежинка, созвездие фактов) в режиме только чтения.
– Внешний уровень содержит инструментальные средства поддержки прикладных задач DWH, включая:
  - бизнес-аналитика (business intelligence),
  - оперативная аналитическая обработка (OLAP),
  - интеллектуальный анализ данных (data mining),
  - системы поддержка принятия решений (decision support systems),
  - языки запросов и создания отчетов.

Помимо этих трех уровней архитектура DWH включает:

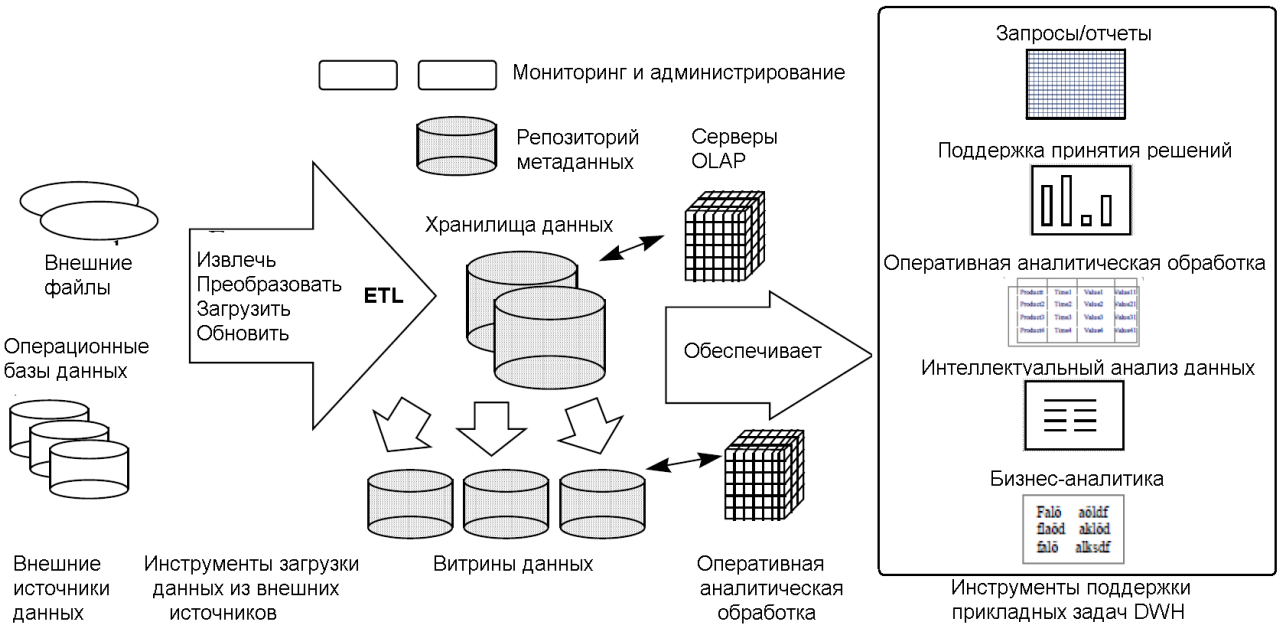
– *Репозиторий метаданных*, который содержит информацию о данных DWH.
– *Витрины данных* (data marts), содержащие подмножество корпоративных данных, представляющих интерес для определенных групп пользователей.
– *Средства управления и контроля*.
– *Инструментальные средства загрузки данных из внешних источников* (базы данных, файлы, электронные таблицы и т.д.) в БД DWH. Эта компонента получила название ETL (Extract, Ttransform, Load), которая выполняет функции извлечения данных из источников, их проверки и очистки, преобразования к нужному виду, интеграции и загрузки или обновления БД DWH [23]. Концепция ETL возникла в 1970-х годах в связи с использованием централизованных репозиториев данных. Но только в конце 1980-х и начале 1990-х годов она приобрела большую популярность в связи с появлением DWH.

Модели DWH

С архитектурной точки зрения выделяют следующие три типа моделей DWH [24]:

– *корпоративное хранилище*, содержащее консолидированные данные, извлеченные из нескольких операционных источников - это DWH всей корпорации [25];
– *витрина данных* - содержит подмножество корпоративных данных;
– *виртуальное хранилище* - это множество взглядов (views) операционных БД [26, 27].

Также существует точка зрения [28], что архитектура DWH включает: архитектуру модели данных, процессную архитектуру, информационную архитектуру, технологическую архитектуру, ресурсную архитектуру.



Архитектура DWH

Витрина данных (data mart). Концепция витрин данных была предложена Forrester Research ещё в 1991 году. Это предметно-ориентированная и, как правило, содержащая данные по одному из направлений деятельности компании база данных, ориентированная на пользователей одной рабочей группы или департамента. В витрине информация хранится оптимизировано с точки зрения решения конкретных задач.

Существует три типа витрин данных, которые различаются в зависимости от их отношения к хранилищу данных

Зависимые витрины данных - это сегменты в корпоративном хранилище данных. Этот нисходящий подход начинается с хранения всех бизнес-данных в одном центральном месте. Вновь созданные витрины данных извлекают определенное подмножество первичных данных всякий раз, когда это необходимо для анализа.

Независимые витрины данных действуют как автономная система, которая не полагается на хранилище данных. Аналитики могут извлекать данные по конкретному предмету или бизнес-процессу из внутренних или внешних источников данных, обрабатывать их, а затем сохранять в репозитории витрины данных до тех пор, пока они не понадобятся группе.

Гибридные витрины данных объединяют данные из существующих хранилищ данных и других операционных источников. Этот унифицированный подход использует скорость и удобный интерфейс нисходящего подхода, а также предлагает интеграцию независимого метода на уровне предприятия.

Идея соединить две концепции — хранилищ данных и витрин данных, по-видимому, принадлежит Марку Демаресту (Marc Demarest.) [29], который в 1994 году предложил объединить две концепции и использовать хранилище данных в качестве единого интегрированного источника данных для витрин данных.

Для взаимодействия между собой витрины данных могут объединяться в сеть, создавая тем самым виртуальное хранилище данных.

Многомерная модель данных DWH. Куб данных

Было предложено множество многомерных моделей данных, классификация, анализ и сравнение которых приведено в работе [30]. Кратко опишем одну из них, которая является наиболее используемой, а именно, куб данных [31].

Куб данных предполагает моделирование и представление данных с использованием понятия многомерного пространства. Куб данных определяется через понятия "факт" и "измерение"

Согласно [32] термины "факт" и "измерение" возникли в конце 1960-годов в результате выполнения совместного исследовательского проекта корпорации General Mills и Дартмутского университета. В 1970-х годах маркетинговые компании AC Nielsen и IRI постоянно использовали эти термины для описания своих агрегированных данных и стремились использовать пространственные модели для презентации аналитической информации.

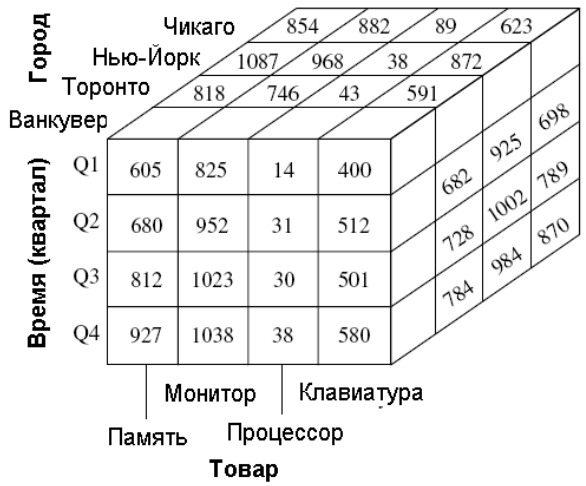
Измерение (dimension) - это характеристика, относительно которой представляются агрегируемые данные. При использовании n измерений получаем n-мерный куб. Измерение - это ось куба.

Измерение может разбиваться на подизмерения, например, измерение "страна" на подизмерения "области", а области на "города" и т.д., образуя, таким образом, иерархическую структуру измерения.

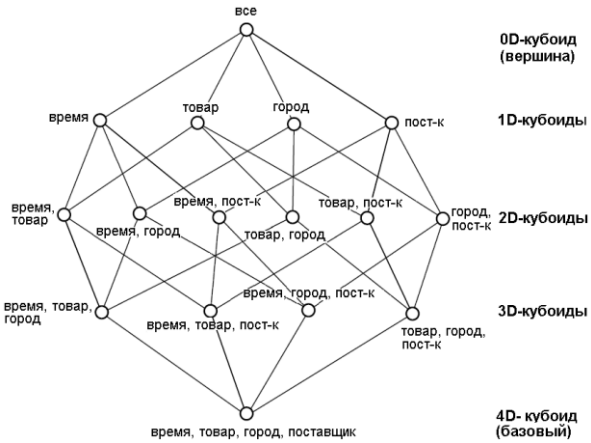
Факт - это характеристика, относительно которой представляются агрегированные данные, например, цена, количество, объем. Факт может обладать свойствами (атрибутами), например, валюта, единица измерения.

Мера (measures) – это собственно агрегированные значения Меры располагаются в ячейках куба. Выделяют три типа числовых мер. *Аддитивные меры* (additive measures) могут быть агрегированы по всем измерениям. *Полуаддитивные* (semi-additive) могут агрегироваться только по некоторым измерениям, а *неаддитивные* (non-additive) вообще не агрегируются.

Многомерная модель имеет графическое представление, которое получило название *кубоида* (cuboid) [33]. Кубоид, содержащий самый низкий уровень агрегированных данных (то есть агрегированные данные по всем измерениям), называется *базовым кубоидом*. На рис. ниже приведен трехмерный базовый кубоид.



Из этого трехмерного кубоида можно получить три двумерных кубоида путем агрегирования данных по каждому из трех измерений, из них - три одномерных и, наконец, один нульмерный кубоид (кубоид-вершина) - то есть одно агрегированное значение всех мер исходного кубоида. Такая структура является *решеткой кубоидов* (cuboids lattice) [34] и называется *кубом*. На рис. ниже приводится куб (решетка кубоидов) для четырехмерного базового кубоида с измерениями: товар, время, город, поставщик (пост-к). Иногда кубоиды также называют кубами или подкубами.



Операции над OLAP-кубами

Существует пять основных операций над кубами OLAP:

Сворачивание (roll-up), также называемая *обобщением* (drill-up). Приводит к агрегированию куба данных либо перемещением вверх по иерархической структуре измерения (переход от частного понятия к более общему), либо удалением измерения

посредством агрегирования всех мер этого измерения.

Разворачивание (roll-down), также называемая *детализацией* (drill-down). Операция, обратная к сворачиванию/обобщению - переход от обобщенных данных к более детальным посредством либо перемещения вниз по иерархической структуре измерения, либо введения новых измерений.

Срез (slice) — извлечение из куба подмножества ячеек, связанных с каким-либо одним значением одного из его измерений, то есть получается куб, у которого одно из измерений содержит одно значение. Никакая агрегация мер не производится

Фрагментация (dice) — является обобщением среза. Из куба извлекается подкуб, содержащий только те значения каждого из измерений, которые указаны в операции. Никакая агрегация мер не производится

Вращение (pivot) — позволяет менять пространственную ориентацию осей измерений куба, выбирая наиболее удобное для аналитика представление. В OLAP-технологиях куб — это прежде всего средство визуализации многомерных данных. Поэтому при его использовании необходимо решать задачу отображения информации в удобном и интерпретируемом для человека виде.

Кроме того, были предложены следующие дополнительные операции

Объединение (drill across) - позволяет объединять много кубов, которые имеют одно или более общих измерений.

Проникновение (drill through) - позволяет переходить от данных на нижнем уровне куба (базовый куб), к исходным данным, откуда куб был извлечен. Операция обычно используется для определения причины "выбросов" в кубе данных.

Агрегирующие функции

Неотъемлемая часть OLAP-модели – задание функций агрегирования. Поскольку цель OLAP – создание многоуровневой модели анализа, данные на всех уровнях, включая базовый, должны быть соответствующим образом агрегированы. По каждому измерению возможно задавать собственную (и не одну) функцию агрегирования.

Такие функции включают: функции агрегирования, статистические функции, функции ранжирования Top N, Bottom N и другие. В [35] приведена классификация агрегирующих функций с точки зрения сложности распараллеливания:

В связи со сложностью структуры куба опубликовано много статей по его эффективной реализации. Обзор исследований в этой области приведен в работах [34, 36]. Кроме того, исчерпывающий обзор по реализации ROLAP-кубов дан в [37].

В заключение отметим, что куб данных используется не только для представления многомерных данных, но и других сложных типов данных, например, пространственных, темпоральных, текстовых, мультимедийных, сетевых и графических [38, 39].

Многомерные базы данных (МнБД)

Это разновидность БД, которая создается для хранилищ данных и оперативной аналитической обработки данных (OLAP). OLAP, работающие с МнБД называются многомерными OLAP (MOLAP). Как правило, МнБД используют модель многомерных кубов для представления исходных данных. В [40] утверждается, что математический аппарат многомерных БД был разработан выдающимся американским математиком Доном Нельсоном (Don Nelson) в 60-х годах по заказу министерства обороны США.

Концептуальные схемы DWH

По аналогии с ER-схемой концептуальной модели ПО, принятой в традиционной технологии проектирования реляционных OLTP-баз данных, в технологии проектирования DWH были предложены следующие OLAP-схемы: звезда, снежинка и созвездие фактов [33].

Схема звезды. Наиболее популярная схема, которая содержит:

– одну большую центральную **таблицу фактов**, содержащую данные по всем мерам;
– множество небольших по размеру **таблиц измерений**, по одной на каждое измерение, которые содержат сведения (атрибуты) каждого из измерений. Графиче-

ское представление этой схемы напоминает звезду, в которой таблицы измерений располагаются радиально вокруг таблицы фактов.

Схема снежинки. Является обобщением схемы звезды. В данном случае, если таблица измерений содержит много "разноплановых" атрибутов (например, она содержит атрибуты не только торгового центра, но и города, региона, страны), такая таблица декомпозируется, то есть она разбивается на несколько "дополнительных" таблиц (таблиц подизмерений). Граф результирующей схемы напоминает снежинку

Созвездие фактов. Предполагает существование многих таблиц фактов, которые имеют общие таблицы измерений. Графически эта схема представляется множеством связанных схем звезд.

Разновидности таблиц фактов

Было предложено множество различных видов таблиц фактов, основные из которых являются следующие.

- **Транзакционные таблицы** (transactional table) - являются наиболее фундаментальными. Степень детализации фактов в таблице определяется принципом "одна строка таблицы на каждую транзакцию". В этом случае таблица фактов содержит наиболее подробную информацию.
- **Периодические снимки** (periodic snapshot) - фиксируется "картина мира" в выбранные моменты, например, когда собирается сводная информация работы предприятия за прошедший месяц.
- **Аккумулирующие снимки** (accumulating snapshots) - используется для представления деятельности в виде процессов, имеющих четко зафиксированные начало и конец, например, оформление заказов, с четко определенными промежуточными результатами, которые аккумулируются и окончательно фиксируются по завершению заказа.
- **Темпоральные снимки** (temporal snapshots) - фиксация ситуации не согласно моментам времени, а согласно интервалам времени.

Учитывая сложность процесса концептуального моделирования DWH, было проведено множество исследований по вопросу оценки качества этого процесса, обзор которых дается в работе [41].

Методологии проектирования и моделирования

Предлагаются следующие три методологии проектирования.

- *Проектирование снизу-вверх*. Предложена Кимбеллом и предполагает предварительное проектирование витрин данных по конкретным тематическим направлениям, которые представляют собой самостоятельные продукты, и последующего их объединения в DWH.
- *Проектирование сверху-вниз.* Предложена Инмоном и предполагает сначала создание централизованного репозитория DWH с использованием "нормализованной" модели данных ПО. Затем на основании DWH создаются витрины данных для конкретных приложений или подразделений предприятия.
- *Гибридное проектирование*, которое предполагает сочетание двух предыдущих подходов и обеспечивает всестороннее и надежное проектирование.

Как уже было отмечено, одной из первых методологий моделирования DWH была предложена Кимбеллом в 1996 г. С тех пор было предложено множество других методологий, всесторонний обзор и сравнительный анализ более полутора десятков из которых приведен в работе [42].

Инструментальные средства

Было разработано много инструментальных средств DWH. По адресу https://www.guru99.com/top-20-etl-database-warehousing-tools.html приводится краткое описание 26 наиболее популярных инструментальных систем класса DWH.

Разновидности DWH

Активные DWH

В начале этого столетия была предложена концепция активных DWH. [43, 44] с тем, чтобы DWH поддерживали автоматическое принятие решений. В активных DWH расширяется технология, лежащая в

основе активных БД, а именно, вводятся "правила анализа", которые имитируют работу аналитика во время принятия решения. В это же время появились первые коммерческие продукты DWH с ограниченными возможностями активных правил [45, 46].

DWH реального времени

Эта концепция предполагает, что исходные данные поступают в DWH сразу же, как только они были порождены их источником и становятся доступными для их анализа [47]. О таких системах говорят, что они являются DWH "с нулевой задержкой". Популярность данной концепции привела к тому, что многие производители, включая IBM [48] и Oracle [49] начали производить DWH этого класса. Краткий анализ исследований по этому направлению приведен в [50].

Эволюционные DWH

DWH предоставляют возможность сохранения и анализа данных за большой промежуток времени Так как реальный мир, отраженный в DWH, изменяется, то тоже самое должно происходить в DWH. Также могут изменяться потребности пользователей. Кимбалл, вероятно, был первым, кто обратил внимание на это в 1996 г. и предложил ряд решений [51]. Эта проблема с легкой руки Кембелла получила название "медленно изменяющиеся измерения" (Slowly Changing Dimensions - SCD). С тех пор было проведено много исследований в этом направлении, краткий обзор некоторых из них приведен в [52]. В статье [53] дается аналитический обзор 15 работ по эволюции концептуальной схемы DWH с указанием различных эволюционных операторов. В работе [54] представлена базовая модельно-ориентирования структура эволюции DWH, поддерживаюшая автоматическое распространение изменений в модели данных источника на модель данных DWH.

Темпоральные DWH

Темпоральные DWH содержат те же структурные компоненты, что и традиционные DWH, а именно, измерения, иерархии измерений, факты и меры. Основное отличие заключается в том, что в нетемпоральных DWH время может ассоциироваться только с фактами, обычно представляющее действительное время (в терминах темпоральных БД), а в темпоральных DWH предоставляется возможность отслеживать эволюцию измерений, фактов и мер. Кроме того, темпоральные DWH, как и темпоральные БД, могут быть битемпоральными. Исследования по темпоральным DWH охватывают различные аспекты, например, темпоральные типы [55], концептуальное моделирование и проектирование [56], логическое моделирование и запросы [57, 58], задержка в получении измерений [59], многомерная агрегация [60], корректная агрегация при наличии изменений в данных и структуре [61], эволюция многомерных схем [62]. В работе [63] приводится обзор темпоральных DWH.

Пространственные DWH

Пространственные DWH (Spatial DWH - SDWH) возникли в связи с бурным развитием приложений, имеющих отношение к оперированию пространственными данными и прежде всего географических информационных систем (geographic information system - GIS). SDWH это такие DWH, которые предоставляют возможность оперировать пространственными объектами для поддержки пространственно-ориентированной деловой активности и принятия решений.

В работе [64] впервые было введено понятие пространственного OLAP (SOLAP), отражающее применение методов интеллектуального анализа к обработке пространственных данных. В работах [65, 66] было введено понятие пространственных измерений и предложена их классификация. В статье [67] предложено расширение концептуальной многомерной модели пространственными изменениями, иерархиями и мерами, а также включением в модель топологических связей и операторов. Были исследованы способы представления пространственных мер для геометрических объектов с использованием системы координат [64, 65, 67, 68] и совокупности точек [66].

Обычно SOLAP применяют к дискретным пространственным данным, однако многие сложные задачи GID-анализа пред-

полагают использование непрерывных пространственных данных, обычно называемых пространственными полями. Пространственные поля, или просто поля, описывают физические явления, которые изменяются непрерывно в пространстве или во времени, например, температура и давление воздуха, возвышение земли, распространение урагана. Обычно поля представляются в виде функций, которые приписывают определенные значения каждой точке пространства. В связи с этим проводятся исследования и разработки по созданию полевых DWH. Одной из первых работ в этом направлении была статья [69], в которой предлагался куб данных с непрерывными измерениями. В статье [70] также предлагается многомерная модель данных с непрерывными измерениями и с набором операций, которая может использоваться для OLAP-анализа полевых данных. В работах [71–73] представлена модель и алгебра для работы с пространственно-временными непрерывными полями и их использование для OLAP-анализа пространственных данных.

Было проведено много других исследований по SDWH. Хорошим введением в пространственные DWH является статья [74]. В статье [63] дается аналитический обзор фундаментальных методов и концепций, лежащих в основе пространственных DWH.

SQL и OLAP

В 1995 г. группа исследователей во главе с Джеймсом Греем предложили расширение языка SQL – фразу CUBE BY, задача которой - создание OLAP-кубов [75]. CUBE BY создает группирование по всем возможным комбинациям указанных в нем измерений с разными уровнями агрегации данных. Эта идея была воспринята в SQL:1999.

В SQL:1999 появились возможности работы с OLAP-кубами. Для этого фраза GROUP BY была расширена фразами ROLLUP, CUBE и GROUPING SETS, а также добавлена функция GROUPING.

– *Фраза ROLLUP* приводит к многоуровневому иерархическому группированию по указанным в ней столбцам и создает промежуточные суммы (subtotals) в соответствии с возрастающим уровнем агрегации, от наиболее детализированных уровней представления данных к более обобщенным суммам.
– *Фраза CUBE* позволяет в одной команде вычислить все возможные комбинации промежуточных сумм. Выражаясь терминами решетки кубов, указанные в этой фразе столбцы образуют базовую таблицу и для нее строится решетка. Предложение CUBE может генерировать информацию, необходимую для перекрестных отчетов (cross-tabulation reports), в одном запросе.
– *Фраза GROUPING SETS* формирует результаты группировок по указанным в ней столбцам и объединяет их в одну таблицу, другими словами, он эквивалентен конструкции UNION ALL к указанным группам.
– *Функция GROUPING* возвращает истину, если указанное выражение является статистическим (то есть одержит итоговое значение), и ложь, если выражение нестатистическое, то есть является исходным данным.

В заключение отмнтим, что прекрасным обзором исследований в области DWH на протяжении первых двадцати лет текущего столетия является статья [76]. В ней анализируются около 40 работ по таким направлениям проблематики DWH, как архитектура, проектирование, эволюция, моделирование, аналитическая обработка, оптимизация, тестирование и безопасность.

Литература

1) Inmon W.H.‘Building the data warehouse’, 5th Edition, John Wiley & Son. 2005
2) Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition. John Wiley & Sons, Inc. 2013. 600 p.
3) Breslin M. Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models. Business Intelligence Journal. 2004, 9(1), pp. 6-20
4) Brackett M.H.The Data Warehouse Challenge: Taming Data Chaos. John Wiley & Sons, 1996, 579 pages.
5) Gill S.H., Rao P.C. The Official Client/Server Computing Guide to Data

Warehouse. QUE Corporation, 1996, 382 pages.
6) Poe V. Building a Data Warehouse for Decision Support. Prentice Hall. 1995
7) Codd E.F. Providing OLAP to User-Analysts: An IT Mandate // Computerworld. — Т. 27, № 30
8) Pendse N. What is OLAP? - http://dssresources.com/papers/features/pendse04072002.htm
9) Ponniah P. Data warehousing fundamentals. John Wiley & Sons, 2001, 516 p.
10) Han J., Kamber M.,Pei J. Data Mining: Concepts and Techniques, 3rd ed.The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 2011, 703 p.
11) Chaudhuri S., Dayal U. An Overview of Data Warehousing and OLAP Technology. ACM SIGMOD Record, Volume 26, Issue 1, March 1997 pp 65–74. - https://doi.org/10.1145/248603.248616
12) Jensen C.S., Pedersen T.B., Thomsen C. Multidimensional databases and data warehousing. Synthesis lectures on data management. San Rafael: Morgan Claypool; 2010. 111 p.
13) Vaisman A., Zimányi E. Data Warehouse Systems: Design and Implementation (Data-Centric Systems and Applications) 2014th Edition. Springer; 2014.
14) Muhammad Arif, Ghulam Mujtaba. A Survey: Data Warehouse Architecture. International Journal of Hybrid Information Technology Vol.8, No. 5 (2015), pp. 349-356.
15) Astriani W.,Trisminingsih R. Extraction, Transformation, and Loading (ETL) module for hotspot spatial data warehouse using Geokettle. Procedia, Environmental Science, Elsevier, The 2nd International Symposium on LAPAN-IPB Satellite for Food Security and Environmental Monitoring 2015, pp 626-634
16) Chaudhary S., Murala D.P., Srivastav V.K. (2011) 'A critical review of data warehouse', Global Journal of Business Management and Information Technology, Volume(1):No.(2), pp. 95-103.
17) Oliveira B., Belo O. (2015) A Domain-Specific Language for ETL Patterns Specification in Data Warehousing Systems. In: Pereira F., Machado P., Costa E., Cardoso A. (eds) Progress in Artificial Intelligence. EPIA 2015. Lecture Notes in Computer Science, vol 9273. Springer, Cham. pp 597-602. - https://doi.org/10.1007/978-3-319-23485-4_60
18) Data Warehouse Architecture, Concepts and Components. - https://www.guru99.com/data-warehouse-architecture.html
19) Data Warehouse Architecture: Types, Components, & Concepts. - https://www.astera.com/type/blog/data-warehouse-architecture/
20) Enterprise Data Warehouse: Concepts and Architecture. - https://www.altexsoft.com/blog/enterprise-data-warehouse-concepts/
21) Bhadresh Pandya, Dr. Sanjay Shah. Proposed Local Data Mart Approach for Data Warehouse Architecture. International Journal of Emerging Technology and Advanced Engineering. 2014. Vol. 4, No. 2. pp. 101-104
22) Yang Q., Ge M. Helfert M. Analysis of Data Warehouse Architectures: Modeling and Classification. In Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019), pages 604-611.
23) Kimball R., Caserta J. The Data Warehouse ETL Toolkit. Wiley Publ., 2004, 526 p.
24) Chandra P., Gupta M.K. Comprehensive survey on data warehousing research. International Journal of Information Technology.
25) Scabora L.C., Brito J.J., Ciferri R.R., Ciferri C.D.D.A. Physical data warehouse design on NoSQL databases – OLAP Query Processing over HBase. Proc. 18th Intern. Conf. SCITEPRESS. 2016, pp. 111–118. DOI: 10.5220/0005815901110118.10, pp. 217–224
26) Khan F.A., Ahmad A., Imran M., Alharbi M., Jan B. Efficient data access and performance improvement model for virtual data warehouse. Sustainable cities and society. 2017, vol. 35, pp. 232–240.
27) Gupta A., Mumick I.S. 'Maintenance of materialized views: problems, techniques, and applications', IEEE Data Engineering Bulletin, Special Issue on Materialized

Views and Data Warehousing, 1995, Vol.18, No. 2, pp. 3-18
28) Sachin Chaudhary, Devendra Prasad Murala, V.K. Srivastav. A Critical Review of Data Warehouse. Global Journal of Business Management and Information Technology. 2011, Vol. 1, No.2, pp. 95-103
29) Demarest M. Building The Data Mart. DBMS Magazine, 1994, Vol. 7, No. 8, p. 44—50.
30) Pedersen T.B., Jensen C.S. Multidimensional data modeling for complex data. In: Proceedings of the 15th International Conference on Data Engineering; 1999. p. 336–345.
31) Vassiliadis P. Modeling multidimensional databases, cubes and cube operations. In: Proceedings of the 10th International Conference on Scientific and Statistical Database Management; 1998. p. 53–62.
32) Kimball R., Ross M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition. John Wiley & Sons, Inc. 2013, 600 p.
33) Han J., Kamber M.,Pei J. Data Mining: Concepts and Techniques, 3rd ed.The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 2011, 703 p.
34) Harinarayan V., Rajaraman A., Ullman J.D. Implementing data cubes efficiently. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 1996. p. 205–216.
35) Sanjay Goil and Alok Choudhary. High performance OLAP and data mining on parallel computers. Center of Parallel and Distributed Computing Technical Report TR9705, 1997
36) Morfonios K., Ioannidis Y. Cube Implementations. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 710-716. Springer, New York, 2018
37) Morfonios K., Konakas S., Ioannidis Y., Kotsis N. ROLAP implementations of the data cube. ACM Computing Surveys, 2007. vol. 39, No. 4, Article 12, 53 pages/
38) Pedersen T.B. Managing complex multidimensional data. In: Aufaure M-A, Zimányi E, editors. Business intelligence – second European summer school, eBISS 2012. Brussels: Springer LNBIB; 2013, 15–21 July 2012, Tutorial Lectures.
39) Vaisman A., Zimányi E. Data warehouse systems – design and smplementation. Springer; 2014.
40) Multidimensional DBMS. - https://tadviser.com/index.php/Article:Multidimensional_DBMS
41) Gosain A., Madaan H. Literature Review of Data model Quality metrics of Data Warehouse. International Conference on Intelligent Computing, Communication & Convergence (ICCC-2015). Procedia Computer Science 48 (2015) 236–243
42) Romero O., Abelló A. A Survey of Multidimensional Modeling Methodologies. International Journal of Data Warehousing & Mining, 5(2), 1–23, April-June 2009
43) Schrefl M., Thalhammer T. On Making Data Warehouses Active. In M. Mohania and A. Min Tjoa, editors, DaWaK 2000: Proceedings of the Second International Conference on Data Warehousing and Knowledge Discover, Greenwhich, London (UK), September 4-6, 2000. Springer LNCS, pp. 34–46
44) Thalhammer T., Schrefl M., Mohania M. Active data warehouses: complementing OLAP with analysis rules. Data & Knowledge Engineering, 2001, Vol. 39, No. 3, pp. 241–269.
45) Brobst S. Active data warehousing: a new breed of decision support. In: Proceedings of the 13th International Workshop on Data and Expert System Applications; 2002. p. 769–772.
46) Borbst S., Rarey J. The five stages of an active data warehouse evolution. Teradata Mag. 2001;3(1):38– 44.
47) Syed Ijaz Ahmad Bukhari: Real Time Data Warehouse. CoRR abs/1310.5254 (2013)
48) IBM Data Warehousing. - https://www.ibm.com/analytics/us/en/data-management/data-warehouse.
49) Best practices for Real-time Data Warehousing. An oracle white paper. 2014. http://www.oracle.com/us/products/middleware/data-integration/realtime-datawarehousing-bp-2167237.pdf
50) Mohania M., Nambiar U., Tam H., Schrefl M., Vincent M. Active, Real-Time, and In-

tellective Data Warehousing. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 41-49. Springer, New York, 2018
51) Kimball R. Slowly changing dimensions. DBMS Mag. 1996;9(4):14.
52) Eder J., Koncilia C., Wiggisser K. Data Warehouse Maintenance, Evolution, and Versioning. In In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 884-890. Springer, New York, 2018
53) Arora M., Gosain A. Schema evolution for data warehouse: a survey. International Journal of Computer Applications, 2011, Vol. 22, No. 6, pp. 6-13
54) Taktak S., Alshomrani S., Feki J., Zurfluh G. The power of a model-driven approach to handle evolving Data Warehouse requirements. In: 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD), 2017, pp. 169-181
55) Bruckner R., Min Tjoa A. Capturing delays and valid times in data warehouses: towards timely consistent analyses. J. Intell. Inf. Syst. 2002;19(2):169–190.
56) Malinowski E., Zimányi E. Advanced data warehouse design: from conventional to spatial and temporal applications. Berlin/Heidelberg: Springer; 2008.
57) Ahmed W., Zimányi E., Wrembel R. Temporal data warehouses: logical models and querying. In: Proceedings of the Journées francophones sur les Entrepôts de Données et l'Analyse en ligne, EDA. Editions Hermann; 2015. p. 33–48.
58) Mendelzon A., Vaisman A. Time in multidimensional databases. In: Rafanelli M, editor. Multidimensional databases: problems and solutions. Hershey: Idea Group; 2003. p. 166–199.
59) Golfarelli M., Rizzi S. Managing late measurements in data warehouses. Int J Data Wareh Min. 2007;3(4):51–67.
60) Böhlen M, Gamper J, Jensen C. Towards general temporal aggregation. In: Proceedings of the 25th British National Conference on Databases; 2008. p. 257–169.
61) Golfarelli M, Lechtenbörger J, Rizzi S, Vossen G. Schema versioning in data warehouses: enabling crossversion querying via schema augmentation. Data Knowl Eng. 2006;59(2):435–459.
62) Ahmed W, Zimányi E, Wrembel R. A logical model for multiversion data warehouses. In: Proceedings of the 16th International Conference on Data Warehousing and Knowledge Discovery; 2014. p. 23–34.
63) Golfarelli M., Rizzi S. A survey on temporal data warehousing. Int J Data Wareh Min. 2009;5(1):1–17.
64) Rivest S., Bédard Y., Marchand P. Toward better support for spatial decision making: defining the characteristics of spatial online analytical processing (SOLAP). Geomatica 2001;55(4):539–555.
65) Bédard Y., Merrett T., Han J. (2001). Fundamentals of spatial data warehousing for geographic knowledge discovery. Geographic data mining and knowledge discovery, 2(2), 53-73.
66) Stefanovic N., Han J., Koperski K. Object-based selective materialization for efficient implementation of spatial data cubes. IEEE Trans Knowl Data Eng. 2000;12(6):938–958.
67) Malinowski E., Zimányi E. Representing spatiality in a conceptual multidimensional model. In: Proceedings of the 12th ACM Symposium on Advances in Geographic Information Systems; 2004. p. 12–22.
68) Bimonte S., Tchounikine A., Miquel M. Towards a spatial multidimensional model. In: Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP; 2005. p. 39–46.
69) Shanmugasundaram J., Fayyad U., Bradley P. Compressed data cubes for OLAP aggregate query approximation on continuous dimensions. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 1999. p. 223–232.
70) Ahmed T.O., Miquel M. Multidimensional structures dedicated to continuous spatiotemporal phenomena. In: Proceedings of the 22nd British National Conference on Databases; 2005. p. 29–40.
71) Gómez L., Gómez S., Vaisman A. Analyzing continuous fields with OLAP cubes. In: Proceedings of the 14th ACM Interna-

tional Workshop on Data Warehousing and OLAP; 2011. p. 89–94.
72) Gómez L., Gómez S., Vaisman A. A generic data model and query language for spatiotemporal OLAP cube analysis. In: Proceedings of the 15th Interna tional Conference on Extending Database Technology; 2012. p. 300–11.
73) Gómez L., Gómez S., Vaisman A. Modeling and querying continuous fields with OLAP cubes. Int J Data Wareh Min. 2013;9(3):22–45.
74) Vaisman A.A., Zimányi E. Spatial Datawarehousing. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, Second Edition, 2018, pp. 3587-3592
75) Gray J., Chaudhuri S., Bosworth A., Layman A., Venkatrao, M., Reichart D., Pellow F., Pirahesh H. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab and Sub-Totals. Data Mining and Knowledge Discovery, 1997, 1(1):29–54
76) Chandra P., Gupta M.K. Comprehensive survey on data warehousing research. International Journal of Information Technology. 2018, Vol. 10, No. 2, pp. 217–224 https://doi.org/10.1007/s41870-017-0067-y

Ненормализованные реляционные базы данных

С момента появления реляционной модели и реляционных баз данных многие исследователи и практики обратили внимание на несоответствие возможностей реляционных СУБД потребностям нетрадиционных приложений, например, обработка изображений, текстов, картографической информации и научных данных, автоматизированное проектирование и многие другие, которые требуют использование сложно структурированных данных. В связи с этим были инициированы исследования в следующих двух направлениях: использование сложных объектов и отказ от 1NF в реляционной модели данных.

Что касается исследований и разработок в направлении управление сложными объектами [1-5], то они интерпретировали такие объекты так, как это делается в объектно-ориентированной парадигме, то есть поддерживается структурная часть без возможностей наследования. Многие объектно-ориентированные СУБД (ООСУБД) разрабатывались на базе предыдущих работ по сложным объектам. Для ознакомления с ООСУБД см раздел "Объектные базы данных".

Что касается второго направления, то фундаментальным принципом теории реляционных баз данных является требование, чтобы отношения были как минимум в 1NF, то есть, чтобы они формулировались с использованием атомарных доменов, что не позволяло использовать сложноструктурированные данные. В связи с этим следует отметить, что даже сам Кодд, спустя год после публикации своей знаменитой статьи [6] заявил [7]: "*Для наглядности желательно иметь возможность преобразовывать нормализованное отношение в ненормализованную форму. Это можно добиться введением операции факторизации*". На это высказывание никто не обратил внимания до тех пор, пока в 1977 г. Макиночи (Makinouchi) не опубликовал статью [8], в которой было предложено использовать домены, содержащие множественные значения, и предложил новую нормальную форму, которая учитывала множественные зна-

чения при приведении отношения в 4NF. Эта статья прослужила началом нового направления в реляционных базах данных, которое получило название вложенная реляционная модель данных, ненормализованная реляционная модель, реляционная модель не в первой нормальной форма, ¬1NF модель.

Вложенная реляционная структура данных

В последующие несколько лет были проведены многочисленные исследования по определению и анализу структуры таких ¬1NF отношений. В работах [9, 10] было введено понятие частного (quotient) отношения и исследована операция горизонтальной декомпозиции. В [11] исследована вложенная реляционная структура. В [12] определено фрагментированное (partitioned) отношение. В [13] было предложено использовать ¬1NF отношения для решения проблемы избыточности в противовес процедуре декомпозиции. В [14] определен класс ¬1NF отношений, создаваемых на доменах, являющихся множествами подмножеств атомарных доменов. Авторы [15] предложили и проанализировали ¬1NF отношения, в которых все атрибуты соответствующих 1NF отношений вложены на один уровень вниз. В [16] предложено использовать ¬1NF отношения для представления иерархически структурированных данных.

Вложенная реляционная алгебра

Параллельно с определением и исследованием ¬1NF отношений стали определяться и операции над ними. Со временем они были объединены под названием вложенная реляционная алгебра (ВРА) или алгебра ¬1NF отношений. В работе [17] приведен перечень 15 статей, посвященных ¬1NF алгебре. Все они представляют собой расширение реляционной алгебры для работы с ¬1NF отношениями. Несмотря на их разнообразие, базовая ¬1NF-алгебра содержит следующие операторы:

1) Операции традиционной РА расширяются для работы с ¬1NF отношениями, это объединение, разность, селекция, проекция и декартово произведение.
2) Вводятся две операции реструктурирования: NEST и UNNEST

В 1978 г. Франсуа Бансильон (Francois Bancilhon) [19] и Ян Пареданс (Jan Paredaens) [20] независимо исследовали и определили критерий полноты языков запросов реляционной модели. Со временем этот критерий получил название ВР-полноты. Было показано, что реляционное исчисление Кодда соответствует этому критерию, что послужило строгим теоретическим обоснованием определения полноты Коддом. Этот критерий также стал использоваться в контексте других моделей баз данных. В частности Дирк Ван Гухт (Dirk Van Gucht) [21] доказал, что приведенная выше базовая ¬1NF-алгебра обладает ВР-полнотой.


Франсуа Бансильон


Дирк Ван Гухт

Кратко перечислим наиболее важные расширения данной базовой алгебры.

Операторы сохранения структуры отношения. Так как вложенные отношения могут быть структурированы так, чтобы сохранять неявно заданные в них многозначные зависимости [22], то важно иметь операторы, которые обладают этим свойством. В связи с этим были определены именно такие операторы объединения, пересечения, разности, проекции и соединения [16, 23], а также вложения (NEST) [24].

Вложенная селекция. Авторами [16] был предложен расширенный вариант операции селекции, когда условие выборка производится как на самом отношении, так и на его компонентах.

Статистические операторы. Операторы, определенные в статистической базе данных системы SSDB [25], также применимы в ¬1NF базах данных.

NULL и агрегатные функции. Были предложены расширения ¬1NF алгебры включением значений NULL и агрегатных функций [23, 26-29].

Рекурсивная алгебра. В работах [30-32] была предложена рекурсивная алгебра для вложенных отношений. В этих алгебрах рекурсия применяется в предикатах селекции и в списках атрибутов проекции. Рекурсия позволяет более естественно представлять сложные запросы, чем в нерекурсивной алгебре. Проблемы рекурсии в ¬1NF также обсуждаются в [33].

Вложенные отношения в Datalog. В статье [34] предлагается вариант дедуктивной модели Datalog для вложенных отношений.

Операторы NEST и UNNEST

Вложенные отношения - это такие отношения, которые содержат обычные атрибуты и один или более вложенных атрибутов. Атрибут является вложенным, если он содержит неатомарные значения и в общем случае его значениями могут быть отношения, которые, в свою очередь, могут содержать вложенные атрибуты и т.д.

Оператор NEST группирует кортежи традиционного реляционного отношения по равенству значений "обычных" атрибутов и формирует из каждой группы один кортеж, содержащий по одному значению обычных атрибутов и множество кортежей вложенных атрибутов, которые принадлежат этой группе. Оператор UNNEST выполняет противоположную процедуру, удаляя при этом возможные дубликаты. С формальным определением этих операторов можно познакомиться в [11, 35].

Во многих работах [11, 14, 31, 36] отмечалось, что применение UNNEST к вложенному отношению и последующее применение к нему NEST по тем же атрибутам не всегда приводит к получению исходного отношения, то есть NEST не является инверсным к UNNEST. Для преодоления этой ситуации в [37] была определена так называемая фрагментированная нормальная форма (Partitioned Normal Form - PNF): отношения в этой форме сохраняют инверсность NEST и UNNEST. Отношение находится в PNF, если все его атомарные атрибуты составляют суперключ и то же самое имеет место для всех его вложенных отношений. Отношения в PNF также исследовались в [38, 39] Однако, как было впоследствии отмечено, PNF налагает очень сильные ограничения для многих реальных предметных областей. И только спустя 20 лет в работе [35] было определено понятие декомпозируемого вложенного атрибута (Decomposable Nested Attribute - DNA) и было доказано, что такие атрибуты сохраняют инверсность NEST по отношению к UNNEST. Причем DNA налагает более слабые ограничения, чем PNF.

Вложенное реляционное исчисление

Различают две категории языков запросов - процедурные и декларативные. К процедурным относится ВРА, а к декларативным - вложенное реляционное исчисление (ВРИ). Исторически сначала была определена ВРА и только спустя несколько лет - ВРИ. Согласно [17] впервые ВРИ было определено в [40] в 1984 г. Оно было специфицировано как расширение классического кортежного реляционного исчисления для работы с вложенными отношениями. Было доказано, что это исчисление эквивалентно по выразительным возможностям определенной в этой статье алгебре.

Еще один вариант исчисления был предложен в [29] для оперирования структурой с одним уровнем вложенности и с агрегатными функциями.

Следует также отметить статью [41], в которой дается обзор полученных к тому времени результатов о выразительных возможностях процедурных декларативных языков вложенных отношений и сложных объектов.

Также представляет интерес статья [42], в которой, используя простую теорию типов и исчисление высших порядков, предложены алгебра и исчисление ненормализованных отношений, приведен алгоритм редукции и показана их применимость для прикладной иерархической структуры данных. В дальнейшем был предложен диалект SQL для работы с прикладной иерархической структурой данных SQL/H [43].

Вложенный SQL

С появлением вложенной реляционной модели данных были проведены исследования и появились предложения по модерни-

зации и развитию в этом направлении. Первым таким расширением стал язык SQL/NF [44], который поддерживает вложенные отношения и предоставляет возможность работать с ними. Впоследствии этот вариант был расширен в виде языка X-SQL/NF, в котором введено понятие ROLE, с помощью которого задается иерархия наследования (ISA-иерархия). Вопросы реализации этого языка обсуждаются в [45]. В работах [46, 47] также предлагается вариант SQL, поддерживающий вложенные отношения. В работе [33] предлагается рекурсивный SQL для поддержки вложенных отношений.

Нормальные формы

Как и в классической реляционной модели, в ¬1NF модели также были определены нормальные формы, которые способствуют проектированию схем баз данных, удовлетворяющих определенным свойствам.

- В [8] определена нормальная форма, которая учитывает множественные значения в 4NF.
- В [37] была определена так называемая фрагментированная нормальная форма (Partitioned Normal Form - PNF), которая сохраняет инверсность NEST и UNNEST (см. выше "Операторы NEST и UNNEST").
- В [48] определена NF-NR нормальная форма, которая устраняет глобальную избыточность среди множества вложенных отношений. Учитывает функциональные и многозначные зависимости и сводится к 3NF/4NF, если вложенные отношения оказываются плоскими.
- В [22] определяется так называемая вложенная нормальная форма (Nested Normal Form -NNF), которая учитывает FD и MVD.
- В [49] предлагаются нормальные формы для алгебры вложенных отношений с фиксированной точкой.
- В [50] также предлагается вложенная нормальная форма, которая учитывает функциональные и полные/вложенные многозначные зависимости.

Реализация, системы

Было проведено множество исследования по вопросам реализации баз данных с вложенными отношениями. Они были связаны со структурами хранения, методами доступа, оптимизацией, эквивалентными преобразованиями, архитектурой и др. Некоторые из них отраженны в [51-63]. В 80-90-е г. прошлого столетия было разработано ряд экспериментальных и промышленных систем управления базами данных, поддерживающих вложенные отношения. Наиболее известные среди них: VERSO [64, 65], DASDBS [66], AIM-P [47], ANDA [58], Triton [67], Kappa-II [68], Atlas [69].

В заключение отметим огромной вклад, который сделал в развитие теории баз данных и систем баз данных в целом, и вложенной реляционной модели в частности, Серж Абитебул (Serge Abiteboul). Он в 1998 г. получил премию SIGMOD за инновации, был удостоен награды Association for Computing Machinery (ACM) SIGMOD Test of Time Award в 2004 г., получил приз EADS в 2007 и награду ACM PODS Alberto O. Mendelzon Test-of-Time Award (2008). Абитебул был избран членом Французской академии наук в 2008 году, Европейской академии наук в 2011 году и стал действительным членом ACM в 2011 году.


Серж Абитебул

Литература

1) Raymond Lorie, Won Kim, Dan McNabb, Wil Plouffe. Supporting Complex Objects in a Relational System for Engineering Databases. In Query Processing in Database Systems, ed. Won Kim, David S. Reiner, Dan. S. Batory, Springer-Verlag, 1985., pp. 145-155
2) Raymond A. Lorie, Jean-Jacques P. Daudenarde. On Extending the Realm of Application of Relational Systems. In Information Processing 86, ed. H.-J. Kugler, Elsevir Science Publishers, 1986, pp. 889-894

3) Won Kim, Hong-Tai Chou, Jay Banerjee. Operations and Implementation of Complex Objects. IEEE Trans. on Software Eng. 14, No. 7, 1988, pp. 985-996
4) Mohammad A. Ketabchi, Valdis Berzins. Mathematical Model of Composite Objects and Its Application for Organizing Engineering Databases. IEEE Trans. on Software Eng.14, No. 1., 1988, pp. 71-84
5) Anant Jhingran, Michael Stonebraker. Alternatives in Complex Object Representation: A Performance Perspective. In 6th Int. Conf. Data Eng., Los Angeles, Calif., USA, Febr. 5-9, 1990, pp. 94-102
6) Codd E.F. "A Relational Model of Data for Large Shared Data Banks," Communications of the ACM, Vol. 13, No. 6 (June 1970), pp. 377-397
7) Codd E. F. Relational Completeness of Data Base Sublanguages. In Courant Computer Science Symposium 6 on Data Base Systems. R. Rustin, Ed., 1971, pp. 65-98.
8) Makinouchi A. A consideration on normal form of not-necessarily-normalized relation in the relational data model. Proceedings of 3rd International Conference on Very Large Data Bases Tokyo, 1977, pp. 447-453.
9) Furtado A.L., Kerschberg L. An algebra of quotient relations. In Proceedings of ACMSZGMOD 1977 International Conference on Management of Data (Toronto, 1977), ACM, New York, 1977, pp. 1-8.
10) [Furtado A.L. Horizontal decomposition to improve a non-BCNF scheme. ACM SZGMOD Record 12, 1 (Oct. 1981), 26-32.
11) Thomas S.J., Fischer P.C. Nested Relational Structures. In Kanellakis P.C. editor, Advances in Computing Research, Vol. 3: The Theory of Databases, pp. 269-307, 1986.
12) Orman L. Semantics of indexed data sets. Working Paper 81-05, Graduate School of Management, Cornell University, Ithaca, N.Y., Feb. 1981
13) Kambayashi Y., Tanaka K., Takeda K. Synthesis of unnormalized relations incorporating more meaning. Inf. Sci. 29 (1983), 201-247.
14) Jaeschke G., Schek H. Remarks on the algebra of non first normal form relations. In Proceedings of the ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (Los Angeles, March 1982). ACM, New York, 1982, pp. 124-138.
15) Arisawa H., Moriya K., Miura T. Operations and the properties on non-first-normal-form relational databases. In Proceedings of the Ninth International Conference on Very Large Data Bases (Florence, Oct. 1983), pp. 197-204.
16) Abiteboul S., Bidoit N. Non first normal form relations to represent hierarchically organized data. In Proceedings of the Third ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (Waterloo, Apr. 1984), ACM, New York, 1984, pp. 191-200.
17) Korth H.F., Roth M.A. (1989). Query languages for Nested Relational Databases. In: Abiteboul, S., Fischer, P.C., Schek, H.J. (eds) Nested Relations and Complex Objects in Databases. NF2 1987. Lecture Notes in Computer Science, vol 361. Springer, Berlin, Heidelberg, pp. 190-204
18) Abiteboul S., Bidoit N. (1983) Non First Normal Form Relations: An Algebra Allowing Data Restructuring. Journal of Computer and System Sciences, Vol. 33, No. 3, 1986, pp. 361–393
19) Bancilhon F. On the completeness of query languages for relational data bases. In Proceedings of the 7th Symposium on Mathematical Foundations of Computer Science. Lecture Notes in Computer Science, Vol. 64 pp. 112-123 Springer-Verlag, Berlin, 1978.
20) Paredaens J. On the expressive power of the relational algebra. Information Processing Letters, 1978, 7(2):107—111
21) Van Gucht D. On the expressive power of the extended relational algebra for the unnormalized relational model. In Proceedings of the Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, San Diego, pages 302-312, March 1987.
22) Ozsoyoğlu Z.M., Yuan L.-Y. A new normal form for nested relations. ACM Transactions on Database Systems, 12(1):111–136, March 1987.

23) Roth M.A., Korth H.F., Silberschatz A. Null Values in ¬1NF Relational Databases. Technical Report TR-85-32, Department of Computer Science, University of Texas at Austin, December 1985.
24) Van Gucht D., Fischer P.C. High-level data manipulation languages for unnormalized relational database models. In Proceedings of the XP/7.52 Workshop on Database Theory, Austin, August 1986.
25) Ozsoyoğlu G., Ozsoyoğlu Z.M. SSDB-an architecture for statistical databases. In Proceedings of the 4th Jerusalem Conference on Information Technology, Jerusalem, pages 327–341, May 1984.
26) Roth M.A., Korth H.F., Silberschatz A. (1989) Null values in nested relational databases. Acta Informatica 26, 615–642
27) Roth M.A., Korth H.F., Silberschatz A. (1991) Addendum to Null values in nested relational databases. Acta Informatica 28, 607-610
28) Deshpande, V., & Larson, P.A. (1991) An Algebra for Nested Relations with Support for Nulls and Aggregates. Technical Report CS-91-16, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada.
29) Ozsoyoglu G., Ozsoyoglu Z.M., Matos V. Extending relational algebra and relational calculus with set-valued attributes and aggregate functions. ACM Transaction on Database Systems, 12(4) Dec. 1987, pp. 566-593
30) Jaeschke G. Recursive Algebra for Relations with Relation Valued Attributes. Technical Report 84.01.003, Heidelberg Scientific Center, IBM Germany, 1984.
31) Schek H.-J., Scholl M.H. The relational model with relation-valued attributes. Information Systems, Vol.11, No. 2, pp.137–147, 1986.
32) Colby L.S. Recursive Algebra and Query Optimization for Nested Relations. ACM SIGMOD Record, Vol. 18, No. 2, 1989, pp. 273–283
33) Linnemann V. Non first normal form relations and recursive queries: An SQL-based approach. In Proceedings of the Third International Conference on Data Engineering, Los Angeles, pp. 591–598, February 1987.
34) Benczur A., Hajas C., Kovacs G. Datalog extension for nested relations. Computers & Mathematics with Applications, Vol. 30, No. 12, 1995, pp. 51-79
35) Garani G., 2008. Nest and Unnest Operators in Nested Relations. Data Science Journal, Vol. 7, pp.57–64.
36) Fischer P.C., Thomas S.J. Operators for Non-First-Normal Form Relations. Proc. of the IEEE Computer Society's 7th International Conference on Computer Software and Applications, 1983, pp. 464-475. Chicago, Illinois, USA.
37) Roth M.A., Korth H.F., Silberschatz A. Extended algebra and calculus for nested relational databases. ACM Transactions on Database Systems 13(4), Dec. 1988, pp. 389-417
38) Abiteboul S., Bidoit N. Non first normal form relations: An algebra allowing data restructuring. Journal of Computer and System Sciences, Vol.33, No. 3, 1986, pp. 361-393
39) Hulin G. On Restructuring Nested Relations in Partitioned Normal Form. Proceedings of the 16th VLDB Conference Brisbane, Australia 1990, pp. 626-637
40) Roth M.A., Korth H.F., Silberschatz A. Extended Algebra and Calculus for ¬1NF Relational Databases. Technical Report TR-84-36, Department of Computer Science, University of Texas at Austin, December 1984. revised January 1986.
41) Abiteboul S., Beeri C., Gyssens M., van Gucht D. An introduction to the completeness of languages for complex objects and nested relations. In: Abiteboul, S., Fischer, P.C., Schek, H.J. (eds) Nested Relations and Complex Objects in Databases. NF2 1987. Lecture Notes in Computer Science, vol 361. Springer, Berlin, Heidelberg. 1989, pp 117–138
42) Andon F.I., Reznichenko V.A., Yashunin A.Y. Calculus of Hierarchical Data Structure (Rus). Kibernetika, 1984, No. 6, pp. 13-17.
43) Perov A.S., Reznichenko V.A., Yashunin A.Y. High-Level Query Language for an Applied Hierarchical Data Structure (Rus). In: Organization of User Interaction with Data Banks: Collection of Scientific Pa-

pers, Kiev, IK, Academy of Sciences, UkrSSR, pp. 34-40.
44) Roth M.A., Korth H.F., Batory D.S. SQL/NF: A query language for ¬1NF relational databases. Information Systems, 12(1):99-114, 1987.
45) Ramakrishnan S. Design and Implementation of a Translator for SQL/NF with Role Joins. Master's thesis, The University of Texas at Austin, Austin, Texas, December 1986.
46) Pistor P., Anderson F. Designing a Generalized NF2 Model With An SQL-Type language Interface. Proceedings of the 12th International Conference on Very Large Data Bases, August 1986, pp. 278-285.
47) Pistor P., Dadam P. The advanced information management prototype. In: Abiteboul, S., Fischer, P.C., Schek, H.J. (eds) Nested Relations and Complex Objects in Databases. NF2 1987. Lecture Notes in Computer Science, vol 361. Springer, Berlin, Heidelberg. pp. 3–26
48) Ling T.W., Yan L.L. NF-NR: A Practical Normal Form for Nested Relations. Journal of Systems Integration 4, 1994, pp. 309-340.
49) Gyssens M., Suciu D., Van Gucht D. Equivalence and Normal Forms for the Restricted and Bounded Fixpoint in the Nested Algebra. Information and Computation, 2000, 164(1), pp. 85-117.
50) Roth M.A., Korth H.F. The design of ¬1NF relational databases into nested normal form. In Proceedings of ACM-SIGMOD 1987 Annual Conference (San Francisco, May 1987), ACM, New York, 1987, pp. 143-159.
51) Deshpande A., Van Gucht D.. A storage structure for Nested Relational Databases. In: Abiteboul, S., Fischer, P.C., Schek, H.J. (eds) Nested Relations and Complex Objects in Databases. NF2 1987. Lecture Notes in Computer Science, vol 361. Springer, Berlin, Heidelberg, 1989, pp. 69–83
52) Deppisch U., Paul H.-B., Schek H.-J. A storage system for complex objects. In Proc. Int. Workshop on Object-Oriented Database Systems, pp. 183-195, Pacific Grove, September 1986.
53) Scholl M.H. Theoretical foundation of algebraic optimization utilizing unnormalized relations. In International Conference on Database Theory, Rome (Lecture Notes in Computer Science 2431, pp. 380-396. Springer-Verlag, 1986.
54) Li Y., Kitagawa H., Ohbo N. Optimization of Join-Type Queries in Nested Relational Databases. IEICE Transactions on Information and Systems, 1994, Vol. E77-D, No.6, pp.648-659
55) Bidoit N. Efficient evaluation of relational queries using nested relations. INRIA internal report, 1985.
56) Bridges T.R., Deshpande A. An efficient implementation of nested relational databases on the massively parallel data structure machine. submitted to the International Symposium on Databases in Parallel and Distributed Systems, May 1988.
57) Dadam P., Kuspert K., Andersen F., Blanken H., Erbe Ft., Guenauer J., Lum V., Pisor P., Walch G. A DBMS prototype to support extended NF2 relations: An integrated view on flat tables and hierarchies. In Proceedings of ACM-SIGMOD ‘86 International Conference on Management of Data, Washington, D.C., pp. 356-367, 1986.
58) Deshpande A., Van Gucht D. An Implementation for Nested Relational Databases. In Bancilhon, Francois and David J. DeWitt, editors, Proceedings of the Fourteenth International Conference on Very Large Data Bases, pp. 76-87, August 1988.
59) Paredaens J., Van Gucht D. Possibilities and limitations of using flat operators in nested algebra expressions. In Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Austin, pages 29-38, 1988.
60) Hongbo HE. Implementation of Nested Relations in a Database Programming Language. Master thesis. School of Computer Science McGill University, Montreal, September 1997
61) Scholl M.H., Paul H.B., Scholl H.J. Supporting Flat Relations by a Nested Relational Kernel. Proceedings of the 13th International Conference on Very Large Data Bases, Sep. 1987, pp. 137-147

62) Hong-Chen Liu, Ramamohanarao K. Algebraic equivalences among nested relational expressions. In CIKM '94: Proceedings of the third international conference on Information and knowledge management,1994, pp. 234–243
63) Paul H.B., Schek H.J., Scholl M.H., Weikum G., Deppisch U.. Architecture and implementation of the Darmstadt database kernel system. In SIGMOD '87: Proceedings of the 1987 ACM SIGMOD international conference on Management of data, 1987, pp. 196–207
64) Bancilhon F., Fortin D., Gamerman S., Laubin J.M., Richard P., School M., Tusera D., Verroust A. VERSO: A relational backend database machine. In David K. Hsiao Ed. Advanced Database Machine Architecture, Prentice-Hall, Englewood Cliffs, N.J., 1983, pp.1-18.
65) Scholl M., Abiteboul S., Bancilhon F., Bidoit N., Garnerman S., Plateau D., Richard P., Verroust A. VERSO: A Database Machine Based on Nested Relations. In: Abiteboul, S., Fischer, P.C., Schek, H.J. (eds) Nested Relations and Complex Objects in Databases. NF2 1987. Lecture Notes in Computer Science, vol 361. Springer, Berlin, Heidelberg, pp. 27-49
66) Schek H.-J., Scholl M.H. The Two Roles of Nested Relations in the DASDBS Project. In: Abiteboul, S., Fischer, P.C., Schek, H.J. (eds) Nested Relations and Complex Objects in Databases. NF2 1987. Lecture Notes in Computer Science, vol 361. Springer, Berlin, Heidelberg, pp. 50-68
67) Harvey T.M., Schnepf C.W., Roth M.A. The Design of the Triton Nested Relational Database System. SIGMOD RECORD, Vol. 20, No. 3, 1991, pp. 62-72
68) Kawamura M., Kawamura T. Parallel Database Management System: Kappa. Proc. of FGCS '94, ICOT, Tokyo, December 1994, pp.100-105
69) Sacks-Davis R., Kent A., Ramamohanarao K., Thom J., Zobel J. Atlas: a nested relational database system for text applications. IEEE Transactions on Knowledge and Data Engineering, Vol. 7, No. 3, June 1995, pp. 454-470

Этап 5. Постреляционные базы данных (2000 – 2010+)

Проникновение интернета во все сферы нашей жизни привело к существенному росту числа источников данных, колоссальному росту их объема, неимоверно большой интенсивности их использования, что привело к проблемам хранения, обработки и сложностям оперирования неструктурированной информацией. Классические реляционные СУБД, верно служившие человечеству на протяжении 40 лет, оказались неспособными справиться с этим новым вызовом. В связи с этим появилось новое направление в БД, которое получило название NoSQL, в результате которого появились БД ключ-значение, документные, колоночные, графовые. Однако сторонники реляционных БД решили не сдаваться без боя и направили свои усилия на такую модернизацию реляционной концепции, при которой удалось бы справиться с этим вызовом начала нового века, так возникло направление NewSQL. В это же время возникла концепция семантического веба, задача которого – повысить семантику веба с целью создания механизмов более релевантного поиска. Важнейшей составляющей такого веба стало понятие онтологии. Одним из вариантов хранения онтологий стали онтологические базы данных. Еще в начале 60-гг. прошлого столетия была предложена модель векторного пространства, которая впоследствии была использована в системах информационного поиска, что в конечном итоге привело к созданию векторной модели данных и, как следствие, к векторной базе данных.Идя навстречу возникшим требованиям к разработке систем реального времени, к началу 2000-х гг. выкристализовалась концепция хранения и оперативной обработки потоковых данных, что привело к появлению потоковых БД. В этот же период стали бурно развиваться направления мультимедийных и мультимодельных БД. Краткому анализу этих разновидностей постреляционных БД посвящен данный раздел.

NoSQL-базы данных

В 2000-е гг. с появлением web-ресурсов с огромными хранилищами разнородной информации исследователи стали все больше анализировать новые структуры данных. Реляционный подход основан на четком структурировании данных и строго формализованном доступе к ним, что делает БД негибкими и замедляет скорость работы. Новый подход базировался на отказе от фиксированной схемы данных и языка SQL

NoSQL — термин, обозначающий ряд подходов, направленных на реализацию СУБД, имеющих существенные отличия от моделей, используемых в традиционных реляционных СУБД с доступом к данным средствами языка SQL.

История термина NoSQL

Термин «NoSQL» впервые появился в 1998 году – так была названа реляционная БД, разработанная Карло Строцци (Carlo Strozzi) [1], которая не использовала SQL в качестве языка запросов. Это первоначальное использование данного термина ничего общего не имеет с современной технологией NoSQL. Он также предложил, что поскольку текущие тенденции в БД направлены на отход от реляционной модели, назвать его более подходящим термином - NoREL, что соответствует «No RELational».


Карло Строцци

Вместе с тем, в первом десятилетии 21-го века появились Neo4j (2000), Google BigTable (2004), CouchDB (2005), Amazon Dynamo (2007), Hypertable (2007), Hbase (2007), Dynomite (2008), Voldemort (2009), Cassandra (2009), MongoDB (2009), которые были нереляционными СУБД.

В 2009 г. в Сан-Франциско Йохан Оскарссон (Johan Oskarsson) организовал семинар для обсуждения новых технологий по хранению и обработке данных [2]. Главным стимулом встречи явилось появление на рынке распределенных нереляционных продуктов. В качестве яркой вывески семинара Эрик Эванс (Eric Evans) предложил емкий и

лаконичный термин "NoSQL" [3]. Термин планировался лишь на одну встречу и не имел под собой глубокой смысловой нагрузки, но так получилось, что он распространился по мировой сети и стал де-факто названием целого направления в IT-индустрии. Вместе с тем термин NoSQL не обозначает какую-либо одну конкретную технологию или продукт. Он скорее характеризует вектор развития IT в сторону от реляционных баз данных.


Эрик Эванс

Свойства NoSQL баз данных

Имеется много различных типов NoSQL БД, но для большинства из них характерными свойствами являются следующие.

- **Гибкость**. Как правило, базы данных NoSQL предлагают гибкие схемы, что позволяет осуществлять разработку быстрее и обеспечивает возможность поэтапной реализации. Благодаря использованию гибких моделей данных БД NoSQL хорошо подходят для частично структурированных и неструктурированных данных.
- **Горизонтальная (эластичная) масштабируемость**. Базы данных NoSQL рассчитаны на масштабирование с использованием распределенных кластеров аппаратного обеспечения, а не путем добавления дорогих надежных серверов. Высокая доступность за счет слабой согласованности (за счет упрощенной семантики ACID)
- **Высокая производительность**. Базы данных NoSQL оптимизированы для конкретных моделей данных и механизмов доступа, что позволяет достичь более высокой производительности по сравнению с реляционными базами данных.
- **Широкие функциональные возможности**. БД NoSQL предоставляют API и типы данных с широкой функциональностью, которые специально разработаны для соответствующих моделей данных.
- **Слабоструктурированные (schemaless) данные**. Структура данных не регламентирована. Ее можно менять динамически
- **Поддержка агрегатов.** NoSQL хранилища оперируют не только атомарными, но и агрегатными объектами. В этом случае не нужны нормализованные отношения.
- **Распределенные системы**, как правило, без централизованного управления (децентрализованная)
- **Поддержка распределенных систем** без совместно используемых ресурсов (share nothing).

Классификация систем

Чтобы понять, что можно достичь при разработке и использовании баз данных, было сформулировано утверждение Брюера.

Теорема CAP, известная также как теорема Брюера (Brewer), — эвристическое утверждение о том, что в любой реализации распределённых вычислений можно обеспечить не более двух из трёх следующих свойств:

- **согласованность данных** (Consistency) — во всех вычислительных узлах в один момент времени данные не противоречат друг другу (семантика ACID);
- **доступность** (Availability) — любой запрос к распределённой системе завершается корректным откликом, однако без гарантии, что ответы всех узлов системы совпадают;
- **устойчивость к разделению** (Partition tolerance) — разделение распределённой системы на несколько изолированных секций не приводит к некорректности отклика от каждой из секций.

Принцип был предложен профессором Калифорнийского университета в Беркли Эриком Брюером (Eric Brewer) и Армандо Фоксом (Armando Fox) в 1999 г. [4] и затем в 2002 г. это утверждение доказали Сет Гильберт (Seth Gilbert) и Нэнси Линч (Nancy Lynch) [5]. Теорема впоследствии получила широкую популярность и признание в среде специалистов по рас-


Эрик Брюер

пределённым вычислениям. Концепция NoSQL, в рамках которой создаются распределённые системы управления базами данных, зачастую использует этот принцип в качестве обоснования неизбежности отказа либо от согласованности данных, либо от доступности.

На рис. ниже приводится графическое представление, на котором стороны треугольника соответствуют парам свойств теоремы CAP, возле которых приводятся примеры систем, соответствующие этим свойствам.

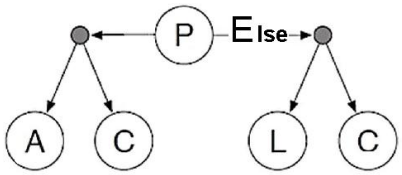
Классификация систем согласно CAP

Теорема PACELC. Эта теорема, как и CAP, описывает, какие ограничения и компромиссные решения имеют распределённые системы в отношении согласованности, доступности и устойчивости к разделению. Однако теорема PACELC дополнительно утверждает, что необходимо идти на компромисс между задержками получения ответа и согласованностью даже при отсутствии устойчивости к разделению, что обеспечивает более полное представление о возможных компромиссах для распределённых систем.

В данной теореме используется не просто треугольник CAP, а следующее условное выражение:

ЕСЛИ (**P**)artition tolerance
ТО (**A**)valability ИЛИ (**C**)onsistency
ИНАЧЕ (**L**)atency ИЛИ (**C**)onsistensy

Простыми словами, при условии наличия устойчивости к разделению (P) можно выбрать доступность (A) или согласованость (C) (это теорема CAP), иначе (E)lse, если устойчивости к разделению нет, можно выбрать между временем задержки (L) или согласованностью (C). Эта ситуация графически представлена на рис ниже.



Таким образом, эта теорема дает четыре разновидности распределенных систем:

PA/EL - высокая доступность (A) при устойчивости к разделению (P) иначе (E) высокая скорость ответа (L) (Dynamo, Cassandra, Cosmos DB, Riak);

PC/EC - в обоих случаях выбирается высокая согласованность (C) (Couchbase, VoltDB/H-Store, Megastore);

PC/EL - высокая согласованность (C) при устойчивости к разделению (P) иначе (E) высокая скорость ответа (L) (PNUTS);

PA/EC - высокая доступность (A) при устойчивости к разделению (P) иначе (E) высокая согласованность (C) (MongoDB).

Даниэль Дж. Абади

Теорема PACELC впервые была представлена в интернете Даниэлем Дж. Абади (Daniel J. Abadi) из Йельского университета в 2010 году, а затем опубликована в виде статьи в 2012 г [6]. В 2015 г. он был награжден премией Very Large Data Base Endowment Inc. (VLDB) за лучшую статью за предыдущие10 лет, а в 2020 г. получил звание "Действительный член ACM" (ACM Fellow) за "большой вклад в потоковые, распределенные, графовые и колоночные базы данных".

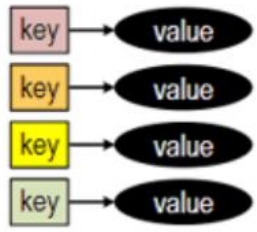
Типы NoSQL баз данных

Далее приводятся основные типы NoSQL баз данных с указанием нескольких из них, принадлежащих каждому из типов. В работе [7] приводится множество различных систем классификации NoSQL с обширным списком NoSQL баз данных, принадлежащих каждому из типов.

Модель ассоциативного массива (associative array model). Представляет собой пару «ключ-значение». Ассоциативный массив отображает ключ на значение, т.е. ассоциирует значение с ключом. В качестве значения могут быть как атомарная единица данных, так и более сложная конструкция,

например, список. Формальное описание этой модели и алгебры приведено в [8, 9]. Работа [10] посвящена исследованию применению этой модели в базах данных.

БД ключ-значение (key-value database). Ассоциативные массивы легли в основу так называемых база данных ключ-значение. Считается, что ключ должен быть уникальным. Согласно [11] первой системой баз данных типа "ключ-значение" была созданная в 1986 г. система GT.M (Greystone Technology M) [12], предназначенная для высокопроизводительной обработки транзакций. За прошедшее с тех пор время было создано множество систем баз данных этого типа. На сайте DB-engines приведено 57 систем баз данных типа "ключ-значение" [13] и по состоянию на январь 2022 г. в тройку наиболее популярных входят Redis, Amazon DynamoDB, Microsoft Azure Cosmos DB. Дополнительная информация относительно использования баз данных «ключ-значение» приводится в разделе «Большие данные»



Модель триплетов (triple store model). Расширением ассоциативного массива стал модель триплетов – три элемента, связанные в выражение «субъект–предикат–объект» [14] или в классический элемент информации «объект-атрибут-значение». По существу триплет – это одна ячейка классического отношения. Благодаря указанию свойства, для которого приведено значение, триплеты являются более информативными, чем пары «ключ-значение». Каждому объекту может соответствовать столько триплетов, сколько свойств имеет данный объект. По адресу <https://en.wikipedia.org/wiki/Comparison_of_triplestores> приведен обширный список баз данных различных хранилищ триплетов.

Колоночные БД (column-oriented databases, wide column store/column families). Базы данных, основанные на триплетах, получили название колоночных (поколоночных или баз данных, состоящих из семейства столбцов). Название объясняется тем, что собрав вместе все триплеты с одинаковым свойством (атрибутом), получаем одну колонку отношения. Представителями колоночных СУБД являются DynamoDB, Google BigTable Cassandra, Scylla, HBase, Hypertable Mulgara, PNUTS и др. Более детальное раскрытие NoSQL баз данных этого типа приводится далее в разделе "Колоночные базы данных".

Документо-ориентированные БД (document-oriented databases). Объединение триплетов, описывающих один объект, называется документом. В качестве значений могут быть строки, числа, массивы и другие вложенные триплеты. Значения могут вкладываться многократно. Базы данных, основанные на документах, получили название документо-ориентированных. Примерами СУБД этого типа являются IBM Domino, RavenDB, CouchDB, ThruDB, MongoDB, DocumentDB и др. Более детальное раскрытие NoSQL баз данных этого типа приводится далее в разделе "Документо-ориентированные базы данных".

Графовые БД (graph database). Триплету также придается семантика «объект-отношение-объект». Такой триплет предназначен для хранения информации, которая в традиционных базах данных называется связью. Представление связей между конкретными объектами позволяет описать предметную область в виде семантической сети или графа, в котором объекты образуют узлы, а отношения – дуги или ребра. Такие БД получили название графовых (graph database). Необходимо отметить, что графовые модели – не новость: описание похожих баз данных можно найти в монографии [15], изданной еще в 1985 г., где они упоминаются как бинарные базы данных. Благодаря интуитивной простоте и пригодности для описания слабо структурированной информации графовые СУБД (Neo4j, AllegroGraph, InfiniteGraph, HyperGraphDB, OrientDB и

другие) активно завоевывают рынок. Более детальное раскрытие NoSQL баз данных этого типа приводится далее в разделе "Графовые базы данных".

Структура GLOBAL. Заметим, что объекты, ассоциативные массивы, триплеты и документы – похожие понятия. В [16] рассматривается универсальная структура GLOBAL, с помощью которой можно представить любой из этих элементов. GLOBAL используется в известной СУБД Cache.

Согласно информации nosql-database.org, в настоящее время (январь 2022 года) в мире имеется более 225 систем управления NoSQL-базами данных. На сайте приведен список этих СУБД с достаточно полной их классификацией. Эта классификация помимо приведенных выше типов NoSQL-баз данных, также содержит:

- мультимодельные БД;
- мультимедийные БД;
- объектные БД;
- БД сетей;
- облачные БД;
- БД XML
- многомерные БД;
- многозначные БД;
- БД источников событий (event sourcing);
- БД временных последовательностей/потоковые БД (time series/streaming);
- научные и специализированные БД.

В работе [17] также приведено описание 11 типов NoSQL-баз данных с указанием конкретных СУБД, принадлежащих каждому из этих типов.

Языки запросов. Имеются существенные различия в возможностях запросов к NoSQL БД в зависимости от поддерживаемой модели данных. Например, БД ключ/значение часто обеспечивают поиск только по (первичному) ключу или некоторому другому идентификатору и не предоставляют возможности запрашивать любые дополнительные поля. Другие хранилища данных, такие, например, как базы данных документов, позволяют выполнять сложные запросы. Это неудивительно, так как при разработке многих БД NoSQL развитые поисковые возможности были опущены в пользу производительности и масштабируемости. В работах [6, 18] содержится анализ различных моделей организации поиска в NoSQL БД.

Было предложено множество NoSQL языков запросов, в частности AQL - ArangoDB Query Language, CQL - Cassandra Query Language, HQL - Hypertable Query Language, Cypher - язык запросов графической базы данных Neo4j. Кроме того, компания Couchbase, развивающая такие системы, как CouchDB, Memcached и Membase, анонсировала создание нового языка запросов - UnQL (Unstructured Data Query Language) [19]. Он представляет собой надмножество классического SQL, то есть во многом с ним совместим, но ориентирован на работу с неструктурированными данными. Проект выполнен совместными усилиями Ричарда Хиппа (Richard Hipp), создателя SQLite, и Дэмиена Каца (Damien Katz), основателя проекта CouchDB. С обзором всех этих языков можно познакомиться в [20]. Было предложено много графовых моделей баз данных, обзор которых приведен в [21].

Литература

1) Strozzi C. NoSQL – A relational database management system. 2007–2010. – http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/nosql/Home%20Page
2) Evans E. NoSQL 2009. May 2009. – Blog post of 2009-05-12. - http://blog.sym-link.com/posts/2009/12/nosql_2009/
3) Evans E. NoSQL: What’s in a name? October 2009. – Blog post of 2009-10-30. - http://blog.sym-link.com/posts/2009/30/nosql_whats_in_a_name/
4) Fox A, Brewer E. Harvest, yield and scalable tolerant systems. In: Proceedings of Workshop on Hot Topics in Operating Systems; 1999. p. 174–178.
5) Seth Gilbert, Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. ACM SIGACT News, Volume 33 Issue 2, June 2002, pp. 51-59,
6) Abadi D. Consistency tradeoffs in modern distributed database system design: CAP is only part of the story. Computer 45(2), 37-42 (2012)
7) Strauch Ch. "NoSQL Databases". - http://www.christof-strauch.de/nosqldbs.pdf

8) Kepner J., Chaidez J., Gadepally, Jansen H. “Associative arrays: Unified mathematics for spreadsheets, databases, matrices, and graphs,” New England Database Day, 2015.
9) Kepner J., Chaidez J., “The Abstract Algebra of Big Data and Associative Arrays,” SIAM Meeting on Discrete Math, Jun 2014, Minneapolis, MN.
10) Jeremy Kepner, Vijay Gadepally, Dylan Hutchison, Hayden Jananthan, Timothy Mattson, Siddharth Samsi, Albert Reuther. Associative Array Model of SQL, NoSQL, and NewSQL Databases. 2016 IEEE High Performance Extreme Computing Conference (HPEC), pp. 1–9. IEEE (2016)
11) A Brief History of NoSQL. - http://blog.knuthaugen.no/2010/03/a-brief-history-of-nosql.html
12) GT.M - https://en.wikipedia.org/wiki/GT.M
13) DB-Engines Ranking of Key-value Stores. - db-engines.com/en/ranking/key-value+store
14) Rusher J., Networks R. Triple Store. - https://www.w3.org/2001/sw/Europe/events/20031113-storage/positions/rusher.html
15) Tsichritzis D.C., Lochovsky F.H., Data models, Prentice-Hall, Englewood Cliffs, N.J., 1982, 381 p.
16) Tweed R., James G. A Universal NoSQL Engine, Using a Tried and Tested Technology. - http://www.mgateway.com/docs/universalNoSQL.pdf, 2010. - 25 p.
17) Acharya B., Pandey M., Rautaray S.S.. Survey On NoSQL Database Classification: New Era of Databases for Big Data. - https://www.academia.edu/26405577/Survey_On_NoSQL_Database_Classification_New_Era_of_Databases_for_Big_Data
18) Ho R. Query Processing for NOSQL DB. November 2009. – Blog post of 2009-11-28. http://horicky.blogspot.com/2009/11/query-processing-for-nosql-db.html
19) Welcome to the UnQL Specification home - http://www.unqlspec.org/display/UnQL
20) Bach M., Werner A. Standardization of NoSQL Database Languages. In: Kozielski S., Mrozek D., Kasprowski P., Małysiak-Mrozek B., Kostrzewa D. (eds) Beyond Databases, Architectures, and Structures. BDAS 2014. Communications in Computer and Information Science, vol 424. Springer, Cham. 2014, pp. 50–60
21) Angles R., Gutierrez C. Survey of graph database models. ACM Comput. Surv. 40, 1, Article 1, 2008, 39 p.

Документо-ориентированные БД

Документо-ориентированная база данных (ДОБД) (document-oriented database) - *это база данных, предназначенная для хранения и манипулирования документами. Документ - это древовидный направленный* граф с помеченными вершинами. Листьевые вершины представляют данные документа, а метки (имена) остальных вершин представляют свойства (атрибуты) соответствующих данных документа. Документы могут объединяться в коллекции в каком-то смысле однотипных документов, но при этом никакие требования на одинаковость состава атрибутов документов может и не предъявляться. Коллекции могут содержать другие коллекции.

Документная структура относится к классу так называемых бессхемных (schemaless) самоописываемых (self-describing) структур, Как правило, в документной структуре отсутствует разделение на схему и экземпляр, схемы нет, а все необходимые структурные элементы схемы присутствуют в экземпляре и в этом смысле данные являются самоописываемыми. Документо-ориентированные данные также получили название слабоструктурированных/полуструктурированных (semi-structured data).

Слабоструктурированные данные

Слабоструктурированная модель данных основана на идее представления данных без явного и отдельного определения их схемы. Вместо этого отдельные фрагменты информации перемежаются структурными/семантическими тегами, определяющими их структуру, вложенность и другие характеристики. Такое представление обеспечивает более гибкую обработку и обмен данными.

Термин "слабоструктурированные данные" ввел Луневский и др. (Luniewski) в 1993 г. в системе Rufus [1]. В 1995 Папаконстантину и др. (Papakonstantinou) определил модель для слабоструктурированных данных OEM в рамках системы интеграции гетерогенных баз данных TSIMMIS [2, 3]. В 1996 Бунеман и др. (Buneman) определил модель слабоструктурированных данных [4]. В 1999 г. Дойч и др. (Deutsch) описал связь между слабоструктурированными данными и XML [5].

Языки запросов

Для слабоструктурированных данных были определены языки запросов, которые позволяют извлекать данные из этой структуры или преобразовывать одну слабоструктурированную структуру в другую. Эти языки появились практически одновременно с само структурой. В 1995 г. был разработана система и язык Lorel [6] - один из первых языков запросов для слабоструктурированных данных, в котором введено понятие регулярного выражения пути для навигации по путям с частично известной структурой. В языке UnQL [7] делается акцент на трансформациях запросов и вводится структурная рекурсия в качестве центральной парадигмы преобразования слабоструктурированных данных. В языке MSL [8], вводятся сколемовские функции для преобразования слабоструктурированных данных. Язык XML-QL [5] стал первым языком, в котором принципы языков слабоструктурированных данных были применены к XML.

С точки зрения внутренней структуры представления документная структура является разновидностью структуры NoSQL ключ-значение, когда значение, в свою очередь, может быть парой "ключ-значение" и так далее, представляя, таким образом, многоуровневую иерархию.

Для форматирования документных данных используются стандартные языки JSON, BSON, XML, YAML и другие.

Основные операции практически всех ДОБД обозначаются аббревиатурой CRUD и означают Create, Retrieve, Update и Delete.

Предполагается, что в любой ДОБД существует механизм задания уникальных идентификаторов документов, по которым производится индексация, что существенно ускоряет поиск документов.

Системы ДОБД

Было разработано множество систем ДОБД. Так на сайте [9] приведено краткое описание более 60 систем ДОБД, среди которых по состоянию на 2022 год лучшими считаются: Amazon DynamoDB, MongoDB,

MongoDB Atlas, Couchbase Server, Google Cloud Firestore, Percona Server for MongoDB, InterSystems IRIS, ArangoDB, Database management, Azure Cosmos DB.

Литература

1) Luniewski A., Shoens K., Schwarz P., Stamos J., Thomas J. The Rufus system: information organization for semi-structured data. In: Proceedings of the 19th International Conference on Very Large Data Bases; 1993. p. 97–107.
2) Papakonstantinou Y., Garcia-Molina H., Widom J. Object exchange across heterogeneous information sources. In: Proceedings of the 11th International Conference on Data Engineering; 1995. p. 251–260.
3) Garcia-Molina H., Papakonstantinou Y., Quass D., Rajaraman A., Sagiv Y., Ullman J., Widom J. The TSIMMIS project: integration of heterogeneous information sources. J Intell Inf Syst. 1997;8(2):117–132.
4) Buneman P., Davidson S., Hillebrand G., Suciu D. A query language and optimization techniques for unstructured data. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 1996. p. 505–516.
5) Deutsch A., Fernandez M., Florescu D., Levy A., Suciu D. A query language for XML. In: Proceedings of the 8th International World Wide Web Conference; 1999. p. 77–91.
6) Abiteboul S., Quass D., McHugh J., Widom J., Wiener J.L. The Lorel query language for semistructured data. International Journal on Digital Libraries, 1997, 1(1), pp. 68–88.
7) Buneman P., Fernandez M., Suciu D. UNQL: a query language and algebra for semistructured data based on structural recursion. VLDB J. 2000;9(1): 76–110.
8) Papakonstantinou Y., Abiteboul S., Garcia-Molina H. Object fusion in mediator systems. In: Proceedings of the 22th International Conference on Very Large Data Bases; 1996. p. 413–424.
9) Best Document Databases. - https://www.g2.com/categories/document-databases

Колоночные базы данных

Колоночная NoSQL база данных (КБД) *- это такая база данных, в которой данные хранятся сгруппированными по колонкам таблицы, а не по строкам, как в реляционных базах данных.* В ней «соседними» являются не данные из двух столбцов одной и той же строки, а данные из одного и того же столбца, но из разных строк.

Отметим, что термин *колоночная БД* в общем случае имеет два значения. (1) *Колоночно-ориентированная* - это БД, не обязательно NoSQL, которая хранит данные таблицы не по строкам, а по столбцам. Такие системы обычно используются в аналитических инструментальных средствах, например, HPE Vertica. (2) *Семейство колонок* (column-family) или *широкая колонка* (wide-column) представляют тип NoSQL баз данных, которые поддерживают таблицы, имеющие различные количество, имена и форматы/типы колонок в разных строках таблицы. В данном разделе пойдет речь о колоночных БД второго вида.

Колоночные семейства могут состоять из практически неограниченного количества колонок, которые могут создаваться динамически. Чтение и запись происходит с использованием колонок, а не строк.

Колонка может быть представлена в виде множества пар ключ-значение, где ключ - имя колонки, тем самым наследуются свойства хранилищ типа ключ-значение.

В некоторых случаях различают колоночные хранилища и хранилища семейства колонок. В первом случае предполагается, что каждая колонка хранится самостоятельно не зависимо от других колонок, а во втором - все колонки семейства запоминаются вместе.

Суперколонка - это колонка, состоящая из других колонок. Например, суперколонкой является ФИО, состоящая из колонок Фамилия, Имя и Отчество. С другой стороны, Фамилия, Имя и Отчество - это семейство колонок. Таким образом, семейство суперколонок, это семейство, состоящее из семейства колонок. Такая вложенность может быть многократной. С точки зрения концепции хранилищ ключ-значение мы имеем ситуацию, когда значение в свою

очередь является парой ключ-значение. С другой стороны, прибегая к терминологии реляционных баз данных можно сказать, что семейство суперколонок это в каком-то смысле является аналогом понятия "взгляда" (view).

История КБД

Этап 1. Транспонированные файлы (1969–1985)

Считается, что история колоночных баз данных берет свое начало с конца 60-х годов с появлением так называемых транспонированных файлов (transposed files), в которых строки табличных данных переводятся в столбцы, а столбцы - в строки. Первой поддерживающей транспонированные файлы СУБД считается TAXIR (1969) [1], ориентированная на хранение и поиск биологических данных. К этому классу также относятся разработанная в 1975 г. медицинская система TOD [2] и система RAPID [3], созданная в 1976 г. компанией Statistics Canada для поиска и обработки статистических данных, которая впоследствии использовалась во многих статистических организациях до конца 90-х годов. Следует также упомянуть о созданной в 1977 г. SCSS - колоночном варианте системы SPSS (Statistical Package for the Social Sciences) - статистическом пакете для социальных наук [4]. Одной из самых ранних систем, которая обладала чертами современных колоночных СУБД, была Cantor [5]. Были проведены исследования по организации поиска в транспонированных файлах [6]. В 1975 г. в работе [7] исследован вопрос декомпозиции записей на подзаписи с последующим хранением их в отдельных файлах.

Этап 2. Модель декомпозированной памяти - DSM (1985–2000)

Следующий этап связан с появлением методов вертикального разбиения, предполагающего поатрибутную кластеризацию таблиц. К этому моменту в базах данных господствовала так называемая модель N-арной памяти NSM (N-ary Storage Model). Но в 1985 г. была опубликована статья [8], в которой в качестве альтернативы NSM была предложена модель декомпозированной памяти (Decomposition Storage Model, DSM). В DSM каждая колонка таблицы запоминается отдельно, а чтобы знать, какой именно строке таблицы принадлежит значение в колонке, вместе с этим значением запоминается уникальный идентификатор строки, как правило, это суррогатный первичный ключ (см. рис. ниже).

ID	Товар	Цена	Дата
1	ПК	500	07.10.21
2	Монитор	150	07.10.21
3	Мышка	15	09.10.21
4	Принтер	170	11.11.21

а) Модель памяти NSM

	Товар
1	ПК
2	Монитор
3	Мышка
4	Принтер

	Цена
1	500
2	150
3	15
4	170

	Дата
1	07.10.21
2	07.10.21
3	09.10.21
4	11.11.21

б) Модель памяти DSM

На протяжении последующих 20 лет использовались именно термины NSM/DSM для указания строчной и поколоночной моделей памяти. Последующие две статьи, посвященные исследованию проблемы распараллеливания операций работы с DSM [9], а также использованию индексов для выполнения операций соединения и проекции [10] показали неоспоримое преимущество DSM перед NSM при выполнении операций выборки данных из БД.

Этап 3. Бурное развитие (2000-?)

До начала 2000-х годов концепция DSM не была востребована в связи с тем, что отсутствовали классы задач, которые бы остро нуждались в КБД. И только с появлением статистических и аналитических баз данных, хранилищ данных, технологии OLAP и больших данных КБД стали весьма востребованными. Первыми исследовательскими прототипами КБД, которые были реализованы в 2000-2005 гг. и которые оказали существенное влияние на дальнейшее развитие коммерческих КБД, стали

MonetDB [11, 12], VectorWise (MonetDB/X 100) [13] и C-Store [14].

Первыми коммерческими КБД считаются Sybase IQ (1996) и KDB (1998). Вторая половина 2000-х годов ознаменовалось бурным ростом числа колоночных СУБД. В это время были реализованы Vertica (Vertica Analytic Database),, Exasol, ParAccel, Kognito, InfoBright, SAND, Ingres VectorWise, Kickfire, Paraccel. Примерами колоночных баз данных также являются Apache Cassandra, Scylla, Apache HBase, Google BigTable, Microsoft Azure Cosmos DB. Колоночно-ориентированные средства внедрены в такие реляционные СУБД, как Oracle, SQLServer, PostgreSQL, IBM BLU.

В начале 2000-х г. возникла идея гибридных хранилищ, поддерживающих как строчное, так и колоночное хранение данных. Так в 2001 было разработано хранилище PAX (Partition Attributes Across), в 2002 была предложена гибридная модель "Fractured Mirrors". Впоследствии были реализованы коммерческие гибридные хранилища SAP HANA, InfiniDB, Greenplum.

В момент написания статьи в базе данных https://dbdb.io/ (Database of Databases) была представлена 51 СУБД, имеющая отношение к колоночной модели данных.

Характерные черты и область применения

Характерными чертами КБД являются:

- высокая скорость выполнения операций поиска/доступа, особенно при выполнении запросов с агрегатными функциями;
- высокая горизонтальная масштабируемость благодаря полной свободе по распределению колонок между узлами сети;
- высокая эффективность декомпозиции и сжатия данных;
- может функционировать в бессхемном варианте.

Колоночные СУБД применяются, как правило, в аналитических системах класса business intelligence, аналитических OLAP-хранилищах данных (data warehouses) и системах класса Big Data

Методы реализации и оптимизация

Хотя современные коммерческие КБД широко используют принципы и методы, которые были предложены для транспонированных файлов и декомпозированных структур, однако они обладают возможностями, которые на начальных этапах не были предусмотрены. Особенно это относится к вопросам, которые связаны с аспектами оптимизации и повышения производительности их функционирования. Приведем их.

Виртуальный ключ. Если значения колонки фиксированной длины, то можно не хранить суррогатный ключ вместе с каждым значением колонки, а вычислять как значение ключа, так и расположение соответствующего значения колонки на основании смещения. Такой принцип был использован в MonetDB [12]. Его можно использовать при отсутствии сортировки в колонках.

Блочная организация и векторная обработка. Для КБД были предложены методы блочной организации данных и их векторной обработки [13, 15], которые существенно повышают их производительность. Блочная итерация [16] предполагает, что множество значений колонки передаются в виде одного блока от одного оператора к следующему.

Поздняя материализация (late materialization). Поздняя материализация или позднее восстановление требуемого для запроса кортежа подразумевает максимальную насколько это возможно отсрочку выполнения операций соединения колонок. [17, 18]. Поздняя материализация значительно повышает эффективность использования пропускной способности памяти.

Сжатие колонок. Сжатие колонки с использованием наиболее эффективного для него метода приводит к существенному уменьшению размеров файлов колонок [15, 19]. В связи с тем, что колонки содержат данные одного типа (атрибута), достигаются хорошие показатели сжатия с помощью простых алгоритмов. Было пред-

ложено множество алгоритмов сжатия для колоночных хранилищ [20].

Оперирование сжатыми данными. Во многих современных колоночных хранилищах распаковка данных откладывается до тех пор, пока это не станет абсолютно необходимым [16, 17], в идеальном варианте пока не появится необходимость представить результаты пользователю. В связи с этим решается задача оперирования сжатыми данными. Поздняя материализация позволяет колонкам оставаться сжатыми до тех пор не появится необходимость формировать из них кортежи (производить их соединение).

Эффективная реализация операции соединения. Так как КБД предполагают поколоночное представление данных, важным является вопрос использования различных стратегий выполнения операции соединения. В КБД используются как классические алгоритмы соединения, так и специфические [15, 21, 22].

Избыточное представление отдельных столбцов с различной сортировкой. По колонкам, отсортированным относительно конкретного атрибута, можно производить более быстрый поиск. Сохранение нескольких копий конкретной колонки, отсортированных по различным атрибутам, может существенно повысить производительность выполнения запросов. Например, система C-Store [14] физически хранит коллекции колонок, каждая из которых отсортирована по некоторому атрибуту. Группы колонок, отсортированные по некоторому атрибуту, называются проекциями, одна и та же колонка может находиться в различных проекциях. Наличие различных сортировок способствует оптимизации функционирования системы.

Крекинг и адаптивное индексирование баз данных. Крекинг (Cracking) - это принципиально новый подход в базах данных, который основывается на принципе, что создание и ведение индекса является продуктом деятельности по обработки запросов, а не создания и обновления базы данных. Каждый запрос интерпретируется не только как запрос на получение части базы данных, но и как указание на необходимость "откалывания" от нее небольших кусочков, описываемых этим запросом, По этим кусочкам строится крек-индекс с тем, чтобы увеличить скорость последующего поиска. Крек-индекс строится и поддерживается динамически по мере обработки запросов и адаптируется с учетом изменения рабочих нагрузок запросов. По отношению к КБД общая схема функционирования механизм крекинга следующая [23]. Всякий раз, когда запрос впервые формулируется по отношению к атрибуту А, механизм крекинга создает копию колонки атрибута А, которая называется крекинг-колонкой А. По мере выполнения последующих запросов распознаются те, которые обращаются к атрибуту А, и производится дальнейшая настройка крекинг-колонки А и ее индекса. Более того, крекинг-колонка А используется для увеличения скорости последующего поиска по атрибуту А. MonetDB была одной из первых КБД, поддерживающей механизм крекинга. Более подробно познакомиться с механизмами крекинга и адаптивного индексирования можно в монографии [20].

Эффективная загрузка и обновление. В связи с тем, что в КБД данные декомпозированы по колонкам и активно используется механизм сжатия данных, загрузка и обновление БД происходит намного медленнее, чем в строчных БД. Поэтому были исследованы вопросы оптимизации выполнения этих операций [14, 24]. Например, в C-Store [14] сначала данные записываются в несжимаемый оптимизированный для записи буфер и затем периодически "сбрасываются" в большие сжимаемые пакеты.

Циклотрон данных. Одной из главных задач распределенной обработки запросов является разработка самоорганизующейся архитектуры, которая оптимально использует все аппаратные ресурсы для оперативного управления базой данных с тем, чтобы минимизировать время отклика на запросы и максимизировать пропускную способность без наличия единой глобальной точки координации. Была предложена технология *Циклотрона данных* (Data Cyclotron) [25], которая решает эту проблему использованием турбулент-

ного перемещения данных через кольцо хранения, построенное из распределенной оперативной памяти с использованием функциональных возможностей, предлагаемых современными сетевыми средствами удалённого прямого доступа к памяти (Remote Direct Memory Access - RDMA) Запросы, инициированные в отдельных узлах сети, постоянно взаимодействуют с кольцом хранения, собирая фрагменты данных, которые постоянно циркулируют в кольце.

Были проведены сравнительные исследования строчных и колоночных хранилищ [22, 26, 27]. В частности, результаты исследований [22] показали, что оптимизированный вариант колоночного хранилища работает в 5 раз быстрее, чем коммерческие строчные хранилища.

По адресу https://www.predictiveanalyticstoday.com/top-wide-columnar-store-databases/ можно познакомиться со следующими 9 наиболее популярными КБД в 2021 году согласно PAT Research: MariaDB, CrateDB, ClickHouse, Greenplum Database, Apache Hbase, Apache Kudu, Apache Parquet, Hypertable, MonetDB

В свою очередь, согласно сайту https://www.g2.com/ следующие 7 КБД были наиболее популярными в 2021 г.: Amazon Redshift, Snowflake, ClickHouse, Druid, Hbase, Apache Kudu, Apache Parquet (https://www.g2.com/categories/columnar-databases)

Для более глубокого ознакомления с КБД рекомендуем монографию [20]. Также рекомендуем статьи [28-30]. Прекрасным учебным материалом является пособие [31].

Завершим изложение колоночных СУБД следующим высказыванием, взятым из [20]: "*Современные колоночные хранилища вышли за рамки простого поколоночного хранения данных, они предлагают абсолютно новую архитектуру баз данных и механизмы работы с ними, адаптированные для современных технических средств и методов аналитической обработки данных*".

Литература

1) Estabrook G F., Brill R.C. The Theory of the TAXIR accessioner. Mathematical Biosciences, 1969, Vol. 5, No 3–4, pp. 327-340.
2) Weyl S. Fries J.F. Wiederhold G., Germano F. A Modular Self-describing Clinical Databank System. Computers and Biomedical Research. 1975. 8 (3): 279–293.
3) Turner M.J., Hammond R., Cotton P. A DBMS for Large Statistical Databases. VLDB '79: Proceedings of the fifth international conference on Very Large Data Bases - Vol. 5, 1979, pp. 319–327
4) "SCSS from SPSS, Inc". ComputerWorld. September 26, 1977. p. 28.
5) Karasalo I., Svensson P. An overview of cantor: a new system for data analysis. SSDBM'83: Proceedings of the Second International Workshop on Statistical Database ManagementSeptember, 1983, pp.315–324
6) Don S. Batory. On searching transposed files. ACM Transactions on Database Systems, 4(4):531–544, 1979.
7) Hoffer J.A. , Severance D.G. The use of cluster analysis in physical data base design. In VLDB '75: Proceedings of the 1st International Conference on Very Large Data BasesSeptember 1975 Pages 69–86, 1975.
8) Copeland G.P., Khoshafian S.N. . A decomposition storage model. In Proceedings of the ACM SIGMOD Conference on Management of Data,1985, pp. 268–279
9)) 861) Khoshafian S., Valduriez P. Parallel execution strategies for declustered databases. In Proceedings of the International Workshop on Database Machines, pages 458–471, 1987.
10) Khoshafian S., Copeland G., Jagodis T., Boral H., Valduriez P. A query processing strategy for the decomposed storage model. In Proceedings of the International Conference on Data Endineering (ICDE), pp. 636–643, 1987.
11) Boncz P. Monet: A next-generation DBMS kernel for queryintensive applica-

tions. University of Amsterdam, PhD Thesis, 2002.
12) Idreos S., Groffen F., Nes N., Manegold S., Mullender S., Kersten M.L MonetDB: Two Decades of Research in Column-oriented Database Architectures. IEEE Data Eng. Bull., 35(1):40–45, 2012.
13) Boncz P., Zukowski M., Nes N. MonetDB/X100: Hyperpipelining query execution. In Proceedings of the biennial Conference on Innovative Data Systems Research (CIDR), 2005, pp. 225-.237
14)) Michael Stonebraker, Daniel J. Abadi, Adam Batkin, Xuedong Chen, Mitch Cherniack, Miguel Ferreira, Edmond Lau, Amerson Lin, Samuel R. Madden, Elizabeth J. O'Neil, Patrick E. O'Neil, Alexander Rasin, Nga Tran, and Stan B. Zdonik. C-Store: A Column-Oriented DBMS. In Proceedings of the International Conference on Very Large Data Bases (VLDB), pages 553–564, 2005.
15) Abadi D.J., Madden S.R., Ferreira M. Integrating compression and execution in column-oriented database systems. In Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 671–682, 2006.
16) Zukowski M., Boncz P.A., Nes N, Heman S. MonetDB/X100 - A DBMS In The CPU Cache. IEEE Data Engineering Bulletin, 28(2): 17–22, June 2005.
17) Abadi D.J., Myers D.S., DeWitt D.J., Madden S.R. Materialization strategies in a column-oriented DBMS. In Proceedings of the International Conference on Data Endineering (ICDE), pp. 466–475, 2007.
18) Idreos S., Kersten M.L., Manegold S. Self-organizing tuple reconstruction in column stores. In Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 297–308, 2009.
19) Zukowski M., Heman S., Nes N., Boncz P. Super-Scalar RAM-CPU Cache Compression. In Proceedings of the 22nd International Conference on Data Endineering (ICDE), 2006. pp. 59-71
20) Abadi D.J., Boncz P., Harizopoulos S., Idreos S., Madden S. (2013), "The Design and Implementation of Modern Column-Oriented Database Systems", Foundations and Trends® in Databases: Vol. 5: No. 3, pp 197-280.
21) Manegold S., Boncz P., Nes N., Kersten M.. Cache-conscious radixdecluster projections. In Proceedings of the International Conference on Very Large Data Bases (VLDB), pages 684–695, 2004.
22) Abadi D.J., Madden S.R., Hachem N. Column-stores vs. row-stores: how different are they really? In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 2008. p. 967– 980.
23) Idreos S., Kersten M., Manegold S. Database Cracking. Conference: CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, 2007, pp. 68-78
24) Héman S., Zukowski M., Nes N.J., Sidirourgos L., Boncz P. Positional update handling in column stores. In Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 543–554, 2010.
25) Goncalves R., Kersten M.L. The Data Cyclotron Query Processing Scheme. ACM Transactions on Database Systems, Vo. 36. No 4. December 2011, Article No. 27, pp. 1–35
26) Halverson A., Beckmann J.L., Naughton J.F., DeWitt D.J. A Comparison of C-Store and Row-Store in a Common Framework. Technical Report TR1570, University of Wisconsin-Madison, 2006. - https://minds.wisconsin.edu/bitstream/handle/1793/60514/TR1570.pdf?sequence=1
27) Harizopoulos S., Liang V., Abadi D.J., Madden S.R. Performance tradeoffs in read-optimized databases. In VLDB, pages 487–498, 2006.
28) Pingpeng Yuan and Hai Jin. Column Stores. In: Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors. pp. 518-523.
29) Abadi D.J., Boncz P.A. Harizopoulos S. Column-oriented database systems. Proceedings of the VLDB Endowment, Vol. 2, No. 2, 2009, pp. 1664–1665
30) Kanungo A. Column oriented databases. International Journal of Advanced Com-

putational Engineering and Networking, 2017, Vol. 5, No 8, pp. 10-13
31 Abadi D., Boncz P., Harizopoulos S. VLDB 2009 Tutorial on Column-Stores. - https://www.slideshare.net/abadid/vldb-2009-tutorial-on-columnstores

Графовые базы данных

Графовая база данных (ГБД) - *это БД, у которой модель данных имеет графовую структуру некоторого вида, включая схему и ее экземпляры, а манипулирование данными осуществляется с помощью языков, имеющих граф-ориентированные операторы.*

Взаимосвязь между графовой структурой и базами данных отслеживается с момента возникновения баз данных в 60-х годах прошлого столетия. Структура данных иерархических и сетевых баз данных, а также ER-языка, напоминают графовую структуру, но они не являются ГБД.

ГБД оказываются весьма полезными или даже незаменимыми, когда информация о взаимосвязи данных является более важной или настолько же важной, как и сами данные. В таких случаях данные и взаимосвязи между ними находятся на одном логическом уровне.

Согласно [1] выделяют две категории ГБД: *транзакционные и нетранзакционные* ГБД. Первые имеют отношение к большой совокупности небольших графов, например, лингвистические деревья. Характерными операциями для них являются поиск суперграфов, подграфов, подобных графов. Ко второй категории относятся единые большие ГБД, например, социальные сети, возможно состоящие из нескольких компонент. Характерные операции - поиск (кратчайшего) пути, поиск соседей, поиск компонент с заданными свойствами.

Можно выделить два этапа исследований и разработок в области ГБД. На первом этапе, 80-е - начало 90-х гг., активно проводились исследования в области моделей данных и языков запросов ГБД. Затем интерес к ним существенно снизился в связи с появлением геопространственных, темпоральных, слабоструктурированных и XML-баз данных. Но в начале этого столетия эти исследования и разработки возобновились в связи с появлением семантического веба, связанных данных и социальных сетей. Этот этап характеризуется, прежде всего, тем, что помимо исследований было разработано множество промышленных ГБД.

Графовые модели данных

В ГБД не существует канонической модели данных как, например, в реляционной модели. Это объясняется гибкостью графовой структуры, которая позволяет наращивать ее сложность, а также определять различные языки манипулирования и запросов. В связи с этим было предложено множество моделей и языков запросов ГБД. Тем не менее, выделяют следующие типы моделей графовых баз данных [2].

Базовая графовая модель - помеченные графы. Состоит из вершин (узлов) и направленных дуг (рёбер), которые соединяют вершины. Вершины представляют концепты (объекты), а дуги - связи между этими концептами. Графовая структура используется для задания как схемы БД, так и ее экземпляров. Вершины/дуги имеют уникальные идентификаторы и ноль или более меток (labels) [3, 4, 5, 6]. Обычно метки именуются именами соответствующих концептов, а дуги - именами соответствующих связей. Метки позволяют указать классы, которым принадлежат вершины/дуги. Более сложный вариант базовой структуры - мультиграф, в котором пара вершин может быть связна несколькими дугами.

Графовая модель со свойствами (Property Graph Model). Граф со свойствами - это направленный помеченный атрибутивный мультиграф. Понятие графа со свойствами было введено в работе [7], а его формальное определение дано в [8]. Вершины/дуги могут обладать многими свойствами (атрибутами), которые позволяют указать их характеристики. Свойства задаются в виде пар ключ-значение. Языками графовых моделей со свойствами являются G-CORE [9], PROPER [10], Gremlin [11], Cypher [12], PGQL [13].

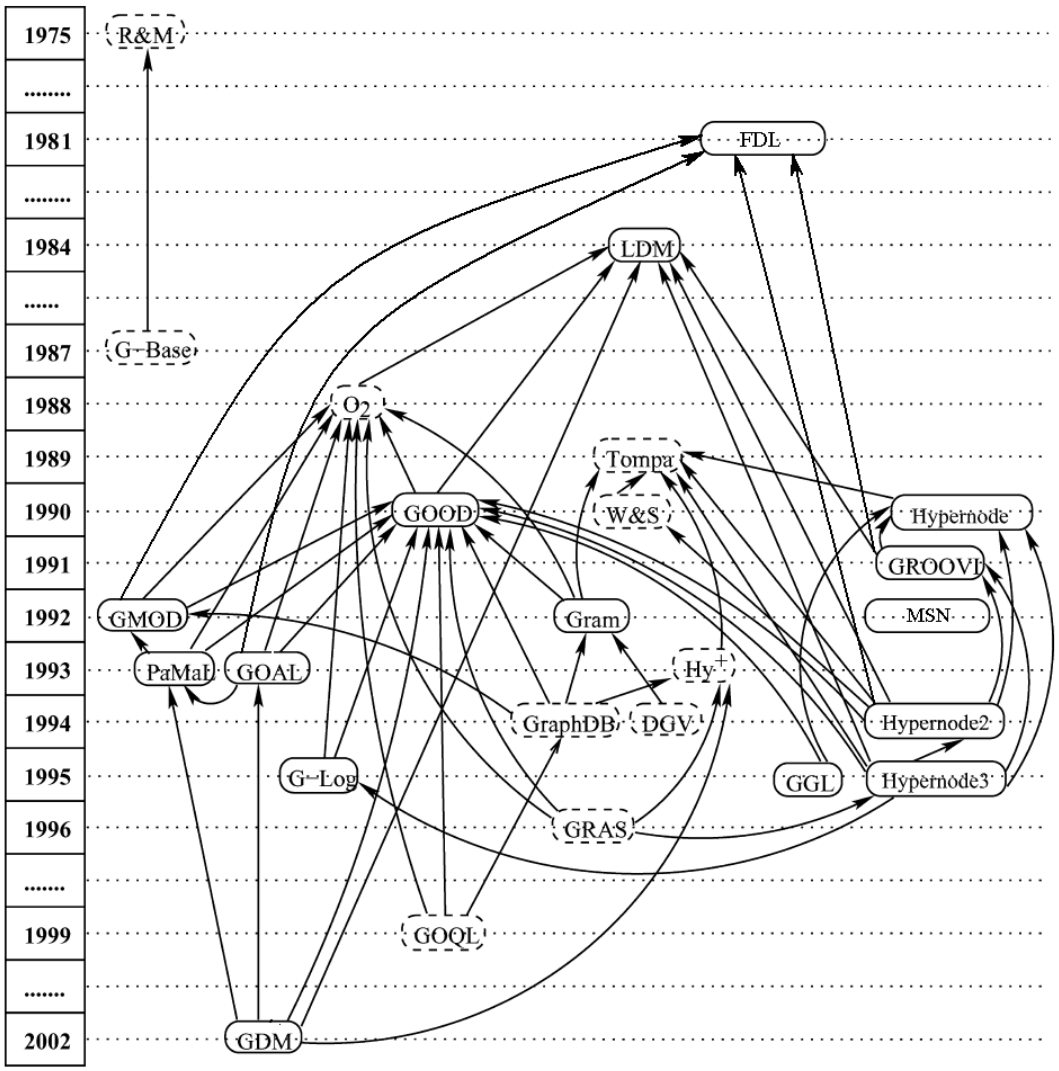
Гиперграфовая модель - сложные дуги. Гиперграф (hypergraph) - это обобщение графовой модели данных, в которой дуги могут соединять любое количество вершин, как в начале дуги, так и в ее конце [14]. Такие дуги называются гипердугами (hyperedge). Гиперграфы оказываются полезными, когда данные содержат связи типа "многие-ко-многим". Гиперграфовые модели представлены в работах [15, 16, 17, 18].

Модель гипервершин - вложенные графы. Модель гипервершин (hypernodes) - это направленный граф, в котором вершины сами могут быть графами. Такие вершины называются гипервершинами (hypernode), образуя структуру вложенных графов (nested graphs). Модель позволяет наглядно и естественно представлять объекты произвольной сложности. Модель гипервершин впервые была определена в 1990 г. в работе [19], а затем уточнена этими же авторами в [20]. Данной модели также посвящены статьи [17, 21, 22]. Аналогичная идея предложена в мультимасштабных сетях (multi-scaled networks) [23].

Граф данных веба - модель RDF. RDF - это язык представления взаимосвязанных ресурсов в вебе с использованием графической структуры данных триплетов, в которой используются направленные дуги и помеченные вершины и дуги. Используемая в RDF графическая структура является наиболее общей в том смысле, что в ней дуги также являются вершинами, Это позволяет поддерживать принцип самоописываемости (реификации), то есть формулировать утверждения относительно утверждений. В RDF-графе одновременно присутствуют схема и ее экземпляры, для отделения экземпляров от схем используются помеченные специальным именем дуги (имя - type). Модель RDF имеет собственный язык запросов SPARQL. К базам данных, поддерживающим модель данных RDF, относятся: 4Store, AllegroGraph, BigData, Jena TDB, Sesame, Stardog, OWLIM, uRiK.

В работе [24] дается детальный обзор графовых моделей данных до 2003 года. На рисунке ниже, взятом из [24], представлены наиболее известные графовые модели баз данных, упорядоченные по годам их публикации. Здесь овалы представляют модели, стрелки указывают на цитирования, а пунктирные овалы свидетельствуют о том, что соответствующие работы имеют отношение к графовой модели БД.

.



Далее приводится краткое описание каждой из моделей с указанием статей их публикации.

- R&M [25] - введено понятие семантической сети для хранения информации о данных базы данных.
- FDL [26] - в рамках функциональной модели данных неявно определена графовая структура данных, целью которой было обеспечение "концептуально естественного" интерфейса базы данных.
- LDM [27] - в рамках логической модели данных явно определена графовая модель базы данных, цель которой - обобщение реляционной, иерархической и сетевой моделей данных.
- G−Base [28] - предложена графовая модель данных, названная G−Base, для представления сложных структур.
- O2 [29] - определена объектно-ориентированная модель данных O2 на основе графовой структуры.
- Tompa [15] - гиперграфовая модель данных для гипертекстовых баз данных.
- GOOD [3] - граф-ориентированная объектная модель, предназначенная для систем с графоподобными средствами представления и манипулирования данными.
- W&S [16] - гиперграфовая модель для доступа к данным в базе данных.
- Hypernode [19] - граф-ориентированая модель с гипервершинами, представляющими собой вложенные графы.
- GROOVY [17] - объектно-ориентированная гиперграфовая модель данных.
- GMOD [4] - предложен ряд концептуальных положений относительно интерфейсов пользователя в граф-ориентированной базе данных.

- Gram [5] - граф-ориентированная модель данных для представления гипертекстов.
- MSN [23] - Графовая модель мультимасштабных сетей, объединяющая основы теории графов и объектно-ориентированную парадигму.
- PaMaL [30] - является расширением GOOD с явным представлением кортежей и множеств.
- GOAL [31] - граф-ориентированная объектная модель с ассоциативными вершинами.
- Hy+ [18] - гиперграфовая модель с языками запросов и визуализации.
- GraphDB [32] - графовая модель данных и язык запросов для баз данных.
- DGV [33] - графовая модель для определения и манипулирования графами различного вида, хранимыми в реляционных или объектных базах данных.
- Hypernode2 [20] - модель со вложенными графами, являющаяся развитием Hypernode.
- G−Log [6] - граф-ориентпированная модель и декларативный язык запросов.
- GGL [21] - теоретико-графовая модель данных баз данных карт генома.
- Hypernode3 [22] графовая модель данных, являющаяся развитием Hypernode.
- GRAS [34] - графовая атрибутивная модель для представления сложной информации.
- GOQL [35] – объектно-ориентированная графовая модель данных и графовый язык запросов.
- GDM [36] - граф-ориентированная модель с n-арными симметричными связями.

Графовые языки запросов (ГЯЗ)

Модель ГБД, помимо структуры и ограничений целостности, имеет высокоуровневый графовый язык запросов (ГЯЗ), в котором можно формулировать специфические для графов операции. В работе [24] дается обзор ГЯЗ периода первой волны исследований и разработок в области ГБД. На ее основании в статье [37] был исследован вопрос выразительно мощности и вычислительной сложности некоторых из этих языков. В статье [38] на основе анализ 9 ГБД приводится сравнительный аналитический обзор моделей ГБД, включая структуры данных, языки описания, манипулирования и запросов, ограничений целостности. В статье [39] исследуется проблема формулировки запросов к ГБД и, в частности, выразительность и сложность навигационных языков запросов. Наконец, в работе [40] дается обзор основополагающих особенностей современных ГЯЗ, а в [2] приводится аналитический обзор ГЯЗ по состоянию на 2018 год.

Как уже было сказано, ГЯЗ обладают специфическими для графов операциями. Кратко опишем их.

Смежность (adjacency) и окрестность (neighborhood). Две вершины смежные, если они соединяются дугой, две дуги смежные, если они имеют общую вершину. В статье [41] исследуется вопрос эффективного выполнения операции смежности в больших динамических разряженных графах. Более общим понятием является "окрестность". n-окрестностью заданной вершины является множество вершин, которые являются достижимыми из заданной посредством пути, содержащим не более n дуг. Исследованию проблемы смежности/окрестности в базах данных посвящена монография [42].

Сопоставление с образцом (pattern matching). Нахождение множества подграфов заданного графа базы данных, которые соответствуют заданному графу-образцу. Задача поиска по образцу характерна для многих классов прикладных задач, например, распознавание образов, идентификация сообществ в социальных сетях. Проблема сопоставления с образцом исследуется в теории баз данных [43, 44], биоинформатике [45], семантическом вебе [46]. Как показано в [47, 48] проблема сопоставления с образцом имеет отношение к проблеме интеллектуально анализа графов данных (data graph mining).

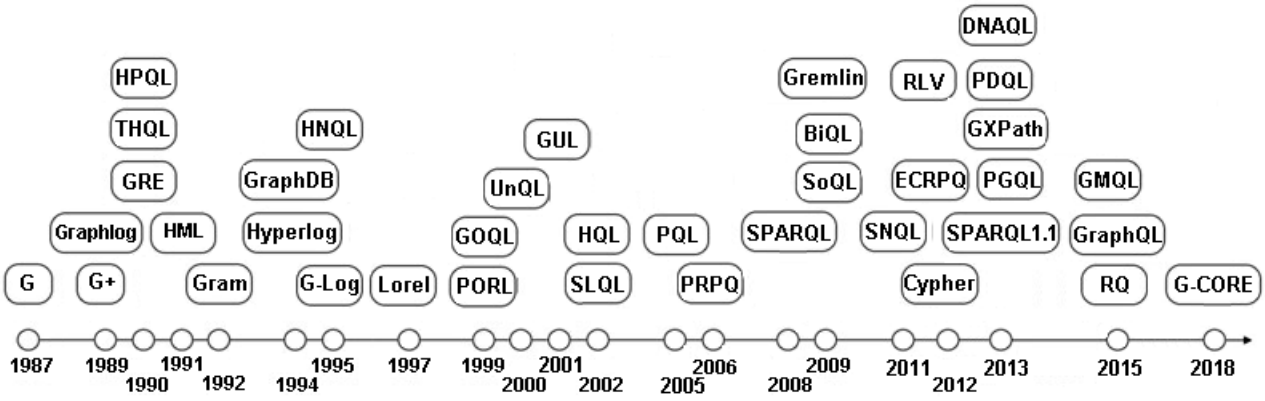
Достижимость/связность (reachability/connectivity). Проблема достижимости заключается в установлении, существует ли путь, ведущий от одной вершины к другой. В этом контексте различают два вида путей: пути фиксированной длины (fixed length paths), содержащие фиксированное количе-

ство вершин и дуг, и регулярные простые пути (regular simple paths), в которых накладываются ограничения (регулярные выражения) на вершины/дуги. В статье [43] дается обзор различных теоретико-графовых задач, связанных с путями, имеющих отношение к базам данных, включая вычисление транзитивного замыкания, выполнение рекурсивны запросов и сложность поиска путей. В статье [49] вводится понятие запроса с регулярным путем (Regular Path Query - RPQ) в качестве способа выражения запросов на достижимость, а в статье [50] анализируется проблема достижимости с учетом наличия регулярных выражений, Этот тип запросов исследовался также в работах [39, 51]. Одной из разновидностей задачи достижимости является нахождение кратчайшего пути, если их несколько, например [52].

Аналитические запросы. Агрегирование. Запросы этого вида, не работают со структурой графа, а предоставляют количественную информацию, обычно в агрегатном виде, относительно топологических свойств графа базы данных. Аналитические запросы обычно содержат специальные агрегатные операторы типа count, sum, min, max, average, которые подытоживают результаты запроса в виде количества вершин, количество соседей, длину пути, расстояние между вершинами, кратчайший путь между вершинами и т.п. Как это показано в монографии [47], сложные аналитические запросы тесно связаны с алгоритмами интеллектуального анализа графа данных.

Приблизительное сопоставление и ранжирование (approximate matching and ranking). Возможны ситуации, когда пользователи не знают структуру графа, относительно которого формулируются запросы, существующие в нем ограничения и правила. В результате чего они могут формулировать запросы, которые не будут давать результаты, или ответы не буду соответствовать ожидаемым результатам. В этом случае желательно иметь возможность получения неточных результатов и их ранжирования согласно установленным критериям. Одной из первых работ, относящихся к формулировке "гибких" запросов к слабоструктурированным текстам, была [53], в которой исследуется вопрос неточной отработки запроса на основе специального вида соответствия между запросом и графом. В работе [54] дается более обобщающее понятие приблизительного сопоставления путей графа, когда результаты поиска могут быть ранжированы согласно их "близости" к исходному запросу.

На рисунке ниже, являющимся незначительно измененным и дополненным вариантом, представленным в работе [2], приведены в хронологическом порядке так называемые "чистые" ГЯЗ, то есть те, которые предназначены для работы с графическими моделями данных.



Эти языки описываются в следующих статьях:

G [49], G+ [55], Graphlog [56], HPQL [19], THQL [16], GRE [57], HML [17] Gram [5], Hyperlog [20], GraphDB [32], G−Log [6], HNQL [22], Lorel [58], PORL [59], GOQL [35], UnQL [60], GUL [61], SLQL [62], HQL [63], PQL [64], PRPQ [65], SPARQL [66],

SoQL [67], BiQL [68], Gremlin [11], SNQL [69] Cypher [12], ECRPQ [70], RLV [71], SPARQL 1.1 [72], PGQL [13], GXPath [73], PDQL [74], DNAQL [75], RQ [76]. GraphQL [77], GMQL [78], G-CORE [9].

Для ГБД принципиальным является эффективное выполнение запросов [79, 80]. Для этого были разработаны различные методы их индексирования [81–83] и оптимизации [84–86].

Графовые базы данных

За последние 20 лет было реализовано более 60 графовых баз данных (ГБД). Их списки присутствуют на многих сайтах.

– https://hostingdata.co.uk/nosql-database/, на этом сайте, посвященном NoSQL-бзам данных, имеется раздел со списком графовых баз данных с адресами в интернете, по которым можно с ними познакомиться;
– на сайте https://dbdb.io/, представляющим собой" базу данных баз данных" и содержащим информацию о более чем 760 СУБД, приведен перечень более 40 систем баз данных, базирующихся на графовой модели данных (https://dbdb.io/browse?data-model=graph) с указанием основных сведений и адресов веб-сайтов, по которым можно с ними познакомиться;
– https://sourceforge.net/software/graph-databases/?page=1 по этому адресу приводятся сведения об около 50 графовых систем баз данных, адресов веб-сайтов, по которым можно с ними познакомиться.

Отметим, что во всех этих трех источниках даты реализации систем располагаются после 2000 года.

В таблице ниже дается обобщающий список ГБД, расположенных в хронологическом порядке.

2002	InfinityDB
2005	AllegroGraph
2006	Blazegraph**,** Sparksee
2007	DEX, Neo4J, HyperGraphDB, AnzoGraph , sones GraphDB
2008	InfoGrid
2009	VertexDB, Pregel, SylvaDB , IBM System G
2010	HyperGraphDB, InfiniteGraph, Sones, OrientDB, FlockDB, Filament , G-Store, Redis_graph, Horton, CloudGraph, Stig
2011	ArangoDB, Trinity, OrigoDB, ArangoDB, Fallen-8
2012	GraphChi-DB, GrapheneDB, SparkleDB, Sqrrl , TigerGraph, FaunaDB, JanusGraph
2013	Bitsy, imGraph, AgensGraph, Galaxybase
2014	Cayley, GraphDB, GrapheekDB, GUN
2015	Cosmos DB**,** DegDB**,** DGraph
2016	IndraDB **,** EliasDB**,** Memgraph**,** VertexDB **,** TypeDB
2017	JanusGraph, Amazon Neptune, Fluree
2018	AnzoGraph DB, Nebula Graph , Neptune
2019	TerminusDB

Имеется множество статей со сравнительным анализом различных ГБД, например: [38, 87, 88–90]. Ниже приводится таблица сравнительного анализа 20 ГБД, взятая из [87]

	Модель данных				Способ хранения		Возможности запросов			Модель вычислений	
Графовые базы данных и инструментальные средства	Простой граф	Граф со свойствами	Гипер-граф	Вложенный граф	Собственная реализация	Использование другой системы	Язык запросов	API	Алгоритмы работы с графами	Локальная	Распределенная
AllegroGraph	X				X		X	X	X	X	
ArangoDB		X			X	X	X	X		X	
Bitsy		X			X			X	X	X	
Cayley		X			X	X	X			X	X
FlockDB	X				X			X			X
GraphBase	X				X		X	X		X	
graphChi	X				X			X		X	
GraphDB	X				X		X			X	
Horton		X				X	X				X
HyperGraphDB			X		X	X	X	X		X	X
IBM System G		X			X	X		X		X	X
imGraph	X				X			X			X
InfiniteGraph		X			X		X	X	X	X	X
InfoGrid	X				X	X		X		X	X
Neo4j		X			X		X	X	X	X	
OrientDB		X				X	X	X	X		X
Sparksee/Dex		X			X			X	X	X	
Titan		X			X	X	X	X		X	X
Trinity		X	X			X	X	X	X		X
TurboGraph		X			X			X	X	X	


Ренцо Англес

В заключение отметим большой вклад в развитие теории и методологии графовых баз данных доктора кафедры компьютерных наук университета Талька (Чили) Ренцо Англеса (Renzo Angles). Данный раздел написан в основном по материалам его работ.

Литература

1) Sakr S. Pardede M. (Eds.). Graph Data Management: Techniques and Applications. IGI Global, 2011, 502 p.
2) Angles R., Gutierrez C. An Introduction to Graph Data Management: Fundamental Issues and Recent Developments. in Graph Data Management, Springer Publishing Companyt, 2018, pp.1-32
3) Gyssens M., Paredaens J., Den Bussche J.V., Gucht D.V. A graph-oriented object database model. In Proceedings of the 9th Symposium on Principles of Database Systems (PODS). ACM Press, 1990, pp. 417–424
4) Andries M., Gemis M., Paredaens J., Thyssens I., Den Bussche J.V. Concepts for graph-oriented object manipulation. In Proceedings of the 3rd International Conference on Extending Database Technology (EDBT). LNCS, vol. 580. Springer, 1992., pp. 21–38
5) Amann B., Scholl M. Gram: A Graph Data Model and Query Language. In European Conference on Hypertext Technology (ECHT). ACM, 1992, pp. 201–211
6) Paredaens J., Peelman P., Tanca L. G-Log: A graph-based query language. IEEE Trans. Knowl. Data Eng. 7, 3, 1995, pp. 436–453
7) Rodriguez M.A., Neubauer P. Constructions from dots and lines. Bulletin of the American Society for Information Science and Technology, 2010, 36,(6), pp. 35-41
8) Angles R. The property graph database model. In Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management, Cali, Colombia, CEUR Workshop Proceedings. CEUR-WS.org, 2018
9) Angles R., Arenas M., Barceló P., Boncz P., Fletcher G., Gutierrez C. G-CORE: A core for future graph query languages. Proceedings of the 2018 International Conference on Management of Data, 2018, pp. 1421-1432

10) Spyratos N., Sugibuchi T. (2016) PROPER - A Graph Data Model Based on Property Graphs. In: Grant E., Kotzinos D., Laurent D., Spyratos N., Tanaka Y. (eds) Information Search, Integration, and Personalization. ISIP 2015, pp. 23-35
11) Rodriguez M.A. The Gremlin graph traversal machine and language (invited talk). In: DBPL 2015: Proceedings of the 15th Symposium on Database Programming Languages. ACM, New York, 2015, pp 1–10
12) Cypher - Graph Query Language - http://neo4j.com/developer/cypher-query-language/
13) van Rest O., Hong S., Kim J., Meng X., Chafi H. PGQL: a property graph query language. In: Proceedings of the international workshop on graph data management experiences and systems (GRADES), 2013
14) Berge C. Graph and Hypergraphs. North-Holland Publishing Company, Amsterdam, 1973
15) Tompa F.W. A data model for flexible hypertext database systems. ACM Transactions on Information Systems, Vol. 7, No 1, 1989, pp. 85–100
16) Watters C., Shepherd M.A. A transient hypergraph-based model for data access. ACM Trans. Inform. Syst. 8 (2), 1990, pp. 77–102
17) Levene M., Poulovassilis A. An object-oriented data model formalised through hypergraphs. Data Knowl. Eng. 6 (3), 1991, pp. 205–224
18) Consens M., Mendelzon A. Hy+: A hygraph-based query and visualization system. ACM SIGMOD Record, Vol. 22, No 2, 1993, pp. 511–516
19) Levene M., Poulovassilis A. The Hypernode model and its associated query language. In Proceedings of the 5th Jerusalem Conference on Information technology. IEEE Computer Society Press, 1990, pp. 520–530
20) Poulovassilis A., Levene M. A Nested-Graph Model for the Representation and Manipulation of Complex Objects. ACM Transactions on Information Systems (TOIS) 12(1), 1994, pp. 35–68
21) Graves M., Bergeman E.R., Lawrence C.B. A graph-theoretic data model for genome mapping databases. In Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS). IEEE Computer Society, 1995, pp. 32-41
22) Levene M., Loizou G. A graph-based data model and its ramifications. IEEE Trans. Knowl. Data Eng. 7, 5, 1995, pp. 809–823
23) Mainguenaud M., Simatic X.T. A data model to deal with multi-scaled networks. Computers, Environment and Urban Systems, 1992, vol.16, No 4, pp. 281–288
24) Angles R., Gutierrez C. Survey of graph database models. ACM Computing Surveys, Vol. 40, No. 1, Article 1, 2008, pp. 1-39.
25) Roussopoulos N., Mylopoulos, J. Using semantic networks for database management. In Proceedings of the International Conference on Very Large Data Bases (VLDB). ACM, 1975, 144–172
26) Shipman D.W. The functional data model and the data language DAPLEX. ACM Transactions on Database Systems, vol. 6, No. 1, 1981, pp. 140–173
27) Kuper G.M., Vardi M.Y. A new approach to database logic. In Proceedings of the 3th Symposium on Principles of Database Systems (PODS). ACM Press, 1984, pp. 86–96
28) Kunii H.S. DBMS with graph data model for knowledge handling. In Proceedings of the 1987 Fall Joint Computer Conference on Exploring technology: Today and Tomorrow. IEEE Computer Society Press, 1987, pp. 138–142
29) Lecluse C., Richard P., Velez F. O2, an object-oriented data model. In Proceedings of the ACM SIGMOD International Conference on Management of Data. ACM Press, 1988, pp. 424–433.
30) Gemis M., Paredaens J. An object-oriented pattern matching language. In Proceedings of the First JSSST International Symposium on Object Technologies for Advanced Software. Springer-Verlag, 1993, pp. 339–355
31) Hidders J., Paredaens J. GOAL, A graph-based object and association language. Advances in Database Systems: Implemen-

tations and Applications, CISM, 1993, pp. 247–265
32) Guting R.H. GraphDB: modeling and querying graphs in databases. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB). Morgan Kaufmann, 1994, pp. 297–308
33) Gutierrez A., Pucheral P., Steffen H., Thevenin J.-M. Database graph views: A practical model to manage persistent graphs. In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB). Morgan Kaufmann, 1994. pp. 391–402
34) KIesel N., Schurr A., Westfechtel B. GRAS, graph-oriented software engineering database system.Information Systems, Vol. 20, No 1, 1995, pp. 21-51
35) Sheng L., Ozsoyoglu Z. M., Ozsoyoglu G. A graph query language and its query processing. In Proceedings of the 15th International Conference on Data Engineering (ICDE). IEEE Computer Society, 1999, pp. 572–581.
36) Hidders J. Typing graph-manipulation operations. In Proceedings of the 9th International Conference on Database Theory (ICDT). Springer-Verlag, 2002. pp. 394–409
37) Wood, P.T.: Query languages for graph databases. ACM SIGMOD Record, 2012, Vol. 41, No 1, pp. 50–60.
38) Angles R. A comparison of current graph database models. IEEE 28th International Conference on Data Engineering Workshops, 2012, 171-177
39) Barceló P. Querying graph databases. In: PODS '13: Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGAI symposium on Principles of database systems, 2013, pp. 175–188
40) Angles R., Arenas M., Barceló P., Hogan A., Reutter J., Vrgoč D. Foundations of modern query languages for graph databases. ACM Computing Surveys (CSUR), 2017, Vol. 50, No 5, Article No.: 68, pp. 1–40
41) Kowalik L Adjacency queries in dynamic sparse graphs. IInformation Processing Letters, 2007, vol. 102, pp. 191–195
42) Papadopoulos A.N., Manolopoulos Y. Nearest neighbor search - a database perspective. Series in computer science. Springer, Berlin, 2005, 170 p.
43) Yannakakis M. Graph-theoretic methods in database theory. In: Proceedings of the symposium on principles of database systems (PODS). ACM, New York, 1990, pp 230–242
44) Barcelo P., Libkin L., Reutter J. Querying graph patterns. In Proc. of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), 2011, pp. 199–210
45) Wang X. Finding patterns on protein surfaces: Algorithms and applications to protein classification. IEEE Transactions on Knowledge and Data Engineering, 2005, vol. 17, pp. 1065–1078
46) Carroll J. Matching RDF Graphs. In Proceedings of the International Semantic Web Conference (ISWC), 2002, pp. 5-15
47) Aggarwal C.C., Wang H. (eds) Managing and mining graph data. Advances in database systems. Springer Science – Business Media, Berlin, 2005
48) Washio T.,Motoda H. State of the Art of Graph-based Data Mining. SIGKDD Explorer Newsletter, 2003, vol. 5, no. 1, pp. 59–68
49) Cruz I.F., Mendelzon A.O., Wood P.T. A graphical query language supporting recursion. ACM SIGMOD Record, Vol. 16, No 3, 1987, pp 323–330
50) Fan W., Li J. Ma S., Tang N., Wu Y. Adding regular expressions to graph reachability and pattern queries. iin Proc. of the IEEE 27th International Conference on Data Engineering (ICDE), 2011, pp. 39–50
51) Mendelzon A.O., Wood P.T. Finding regular simple paths in graph databases. SIAM J Comput, 1995, 24(6), pp. 1235–1258
52) Zhu A.D., Ma H., Xiao X., Luo S,. Tang Y., Zhou S. Shortest path and distance queries on road networks: towards bridging theory and practice. In: Proceedings of the international conference on management of data (SIGMOD). ACM, New York, 2013, pp. 857–868
53) Kanza Y., Sagiv Y. Flexible queries over semistructured data. PODS '01: Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2001, pp. 40–51.

54) Hurtado C.A., Poulovassilis A., Wood P.T. Ranking approximate answers to semantic web queries. Ranking Approximate Answers to Semantic Web Queries. In: Aroyo L. et al. (eds) The Semantic Web: Research and Applications. ESWC 2009. Lecture Notes in Computer Science, vol 5554. Springer, Berlin, Heidelberg. 2009, pp. 263–277
55) Cruz, I.F., Mendelzon, A.O., Wood, P.T. G+: Recursive Queries without Recursion. In: Proceedings of the 2th International Conference on Expert Database Systems (EDS). 1989, pp. 645–666
56) Consens, M.P., Mendelzon, A.O. GraphLog: a Visual Formalism for Real Life Recursion. In: Proceedings of the 9th ACM Symposium on Principles of Database Systems. 1990, pp. 404–416.
57) Wood, P.T.: Factoring Augmented Regular Chain Programs. In: Proceedings of the 16th International Conference on Very Large Data Bases (VLDB). 1990, pp. 255–263. Morgan Kaufmann Publishers Inc.
58) Abiteboul S., Quass D., McHugh J., Widom J., Wiener J.L. The Lorel query language for semistructured data. International Journal on Digital Libraries, 1997, 1(1), pp. 68–88.
59) Flesca, S., Greco, S.: Partially Ordered Regular Languages for Graph Queries. In: Proceedings of the 26th International Colloquium on Automata, Languages and Programming (ICALP). LNCS, 1999, pp. 321–330
60) Buneman P., M. Fernandez, Suciu D. UnQL: A Query Language and Algebra for Semistructured Data Based on Structural Recursion. The VLDB Journal, 2000, 9(1), pp. 76-110
61) Hidders A.J.H. A Graph-based Update Language for Object-Oriented Data Models. Thesis (doctoral)-Technische Universiteit Eindhoven, 2001, 217 p. - https://pure.tue.nl/ws/files/2236754/20014 2116.pdf
62) Cardelli L., Gardner P., Ghelli G.: A Spatial Logic for Querying Graphs. In: Proceedings of the 29th International Colloquium on Automata, Languages, and Programming (ICALP). 2002, pp. 597–610. LNCS, Springer
63) Theodoratos D. Semantic Integration and Querying of Heterogeneous Data Sources Using a Hypergraph Data Model. In: Proceedings of the 19th British National Conference on Databases (BNCOD), Advances in Databases. 2002, pp. 166– 182. LNCS, Springer
64) Leser U. A query language for biological networks. Bioinformatics, 2005, 21(2), pp. 33-39
65) Liu Y.A., Stoller S.D. Querying complex graphs. In: Proc. of the 8th Int. Symposium on Practical Aspects of Declarative Languages. 2006, pp. 16–30
66) Prud'hommeaux, E., Seaborne, A. SPARQL Query Language for RDF. W3C Recommendation. (January 15 2008)
67) Ronen R., Shmueli O. SoQL: a language for querying and creating data in social networks. In: Proceedings of the international conference on data engineering (ICDE). IEEE Computer Society, New York, 2009, pp 1595–1602
68) Dries A, Nijssen S., De Raedt L. A query language for analyzing networks. Proceedings of the 18th ACM conference on Information and knowledge, 2009, pp. 485-494
69) San Martin M., Gutierrez C., Wood P.T. SNQL: A social networks query and transformation language. In: Barcelo, P. and Tannen, V. (eds.) Proceedings of the 5th Alberto Mendelzon International Workshop on Foundations of Data Management. CEUR Workshop Proceedings. CEUR-WS.org. 2011.
70) Barcelo P., Libkin L., Lin A.W., Wood P.T. Expressive languages for path queries over graph-structured data. ACM Transactions on Database Systems, 2012, Vol. 37, No 4, pp. 1–46
71) Santini S.: Regular Languages with Variables on Graphs. Information and Computation, 2012, Vol. 211, pp. 1–28
72) Feigenbaum L , Williams G.T., Clark K.G., Torres E. SPARQL 1.1 Protocol. W3C Recommendation. http://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/, March 21, 2013.
73) Libkin L., Martens W., Vrgoc D. Querying Graph Databases with XPath. In: Proceed-

ings of the 16th International Conference on Database Theory (ICDT), 2013, pp. 129–140
74) Angles R., Barcelo P., Rios G. A practical query language for graph DBs. In: 7th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW), 2013
75) Brijder R., Gillis J.J.M., Van den Bussche J. (2013) The DNA query language DNAQL. In: ICDT '13: Proceedings of the 16th International Conference on Database Theory, 2013, pp. 1–9
76) Reutter J.L., Romero M., Vardi M.Y.: Regular queries on graph databases. In: Proceedings of the 18th International Conference on Database Theory (ICDT). 2015, pp. 177–194
77) GraphQL: A data query language. - https://code.fb.com/core-data/graphql-a-data-query-language/
78) Masseroli M., Pinoli P., Venco F., Kaitoua A., Jalili V., Paluzzi F., Muller H., Ceri S. GenoMetric Query Language: A novel approach to large-scale genomic data management. Bioinformatics, 2015, 31(12), pp. 1881-1888
79) Giugno R., Shasha D. GraphGrep: a fast and universal method for querying graphs. In: Proceedings of the 16th International Conference on Pattern Recognition, 2002. pp. 112–115.
80) He H., K. Singh A. Graphs-at-a-time: query language and access methods for graph databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 2008. p. 405–418.
81) Milo T., Suciu D.. Index structures for path expressions. In: Proceedings of the 7th International Conference on Database Theory; 1999. pp. 277–295
82) Picalausa F., Luo Y., Fletcher G.H.L., Hidders J, Vansummeren S. A structural approach to indexing triples. In: Proceedings of the 9th Extended Semantic Web Conference; 2012. p. 406–421
83) Trißl S., Leser U. Fast and practical indexing and querying of very large graphs. In: Proceedings of the ACM SIGMOD International Conference on Management of Data; 2007. p. 845–856.
84) Calvanese D., De Giacomo G., Lenzerini M., Vardi M.Y. Reasoning on regular path queries. SIGMOD Rec. 2003;32(4):83–92.
85) Fernandez M., Suciu D. Optimizing regular path expressions using graph schemas. In: Proceedings of the 14th International Conference on Data Engineering; 1998. p. 14–23.
86) Goldman R., Widom J. DataGuides: enabling query formulation and optimization in semistructured databases. In: Proceedings of the 23rd International Conference on Very Large Data Bases; 1997. p. 436–445.
87) Angles R. Graph Databases - http://renzoangles.net/gdm/
88) Urbón P. NoSQL graph database matrix. - http://nosql.mypopescu.com/post/619181345/nosql-graph-databasematrix
89) Deepak Singh Rawat, Navneet Kumar Kashyap. Graph Database: A Complete GDBMS Survey. International Journal for Innovative Research in Science & Technology (IJIRST), 2017, Vol. 3, No 12, pp. 217-226
90) Pradeep Jadhav, Ruhi Oberoi. Comparative Analysis of Different Graph Databases, International Journal of Engineering Research & Technology (IJERT), Vol. 3, No 9, 2014, pp. 820-824

NewSQL-базы данных

В 2007 году Майкл Стоунбрейкер, разработчик систем баз данных Ingres и Postgres и будущий лауреат премии Тьюринга, возглавил исследовательскую группу, опубликовавшую основополагающую статью [1], в которой отмечается, что аппаратные предложения, лежащие в основе реляционной архитектуры, больше не применимы. Стоунбрейкер и его команда предложили ряд вариантов перспективных проектных решений относительно СУБД, два из которых стали особенно важными для дальнейшего развития направления, которое со временем получило название NewSQL. Это H-Store [2] - распределенная база данных, полностью находящаяся в памяти, и C-Store [3] - колоночная база данных. Впоследствии на базе H-Store была разработана система управления данными S-Store [4], интегрирующая семантику OLTP-транзакций с потокой обработкой данных, которую разработчики отнесли к классу NewSQL-систем.

В 2010 г. Рик Кеттелл (Rick Cattell) опубликовал статью [5], в которой он использовал термин "масштабируемый SQL", и проанализировал такие известные к тому времени масштабируемые реляционные базы данных, как MySQL Cluster, VoltDB, Clustrix, ScaleDB, ScaleBase, NimbusDB, а также сравнил подходы масштабируемого SQL и NoSQL.

Термин NewSQL был предложен в 2011 году аналитиком 451 Group Мэтью Аслетом (Matthew Aslett) [6, 7]. И с тех пор он стал употребляться для обозначения масштабируемых реляционных систем управления базами данных нового поколения с оперативной обработкой транзакций (OLTP), которые обладают способностью горизонтальной масштабируемости NoSQL и поддержкой ACID, характерной для традиционных (SQL) систем баз данных.


Мэтью Аслет

Были предложены варианты классификации NewSQL-баз данных [9, 10]

Впоследствии были проведены исследования, в результате которых появилось множество публикаций по сравнительному анализу технологий SQL, NoSQL и NewSQL, фундаментальным трудом в этом направлении стала монография Гая Харрисона (Guy Harrison) [11].


Гай Харрисон

В настоящее время NewSQL-технология нашла свою нишу на рынке баз данных и широко и используется в промышленности. К этому классу относят следующие системы: MemSQL, VoltDB, Spanner, Calvin, CockroachDB, FaunaDB, YugabyteDB.

Литература

1) Stonebraker M., Madden S.R., Abadi D.J., Harizopoulos S., Hachem N., Helland P. The End of an Architectural Era (It’s Time for a Complete Rewrite). - VLDB '07: Proceedings of the 33rd international conference on Very large data bases, 2007, p. 1150–1160
2) Kallman R., Kimura H., Natkins J., Pavlo A., Rasin A., Zdonik S., Jones E.P.C., Madden S., Stonebraker M., Zhang Y., Hugg J., Abadi D.J. H-Store: a high-performance, distributed main memory transaction processing system. In: Proceedings of the VLDB Endowment. 2008, Vol. 1, No. 2, pp. 1496-1499.
3) Stonebraker M., Abadi D., Batkin A., Chen X., Cherniack, Ferreira M.,. Lau E, Lin A., Madden S., O’Neil E., O’Neil P., Rasin A., Tran N., Zdonik S. “C-store: a column-oriented DBMS,” Proceedings of the 31st In-ternational Conference on Very Large Data Bases (VLDB ’05), 2005, pp. 553 – 564.
4) Cetintemel U., Du J., Kraska T., Madden S., Maier D., Meehan J., Pavlo A., Stonebraker M., Sutherland E., Tatbul N., Tufte K., Wang H., Zdonik S. S-Store: a streaming NewSQL system for big velocity applications. In: Proceedings of the 40th International Conference on Very Large Data Bases; 2014, Vol. 7, No. 13, pp. 1633–1636
5) Cattell R. "Scalable SQL and NoSQL data stores," ACM SIGMOD Record, 2011, Vol. 39. No. 4, pp. 12-27.

6) Matthew A. (2011). "How Will The Database Incumbents Respond To NoSQL And NewSQL?". 451 Group - https://www.cs.cmu.edu/~pavlo/courses/fall2013/static/papers/aslett-newsql.pdf
7) Matthew A. (2011)." What we talk about when we talk about NewSQL". 451 Group - https://blogs.451research.com/information_management/2011/04/06/what-we-talk-about-when-we-talk-about-newsql/
8) Stonebraker Mil. NewSQL: An Alternative to NoSQL and Old SQL for New OLTP Apps. Communications of the ACM Blog. - https://cacm.acm.org/blogs/blog-cacm/109710-new-sql-an-alternative-to-nosql-and-old-sql-for-new-oltp-apps/fulltext
9) Pavlo A., Aslett M. What's Really New with NewSQL?. SIGMOD Record, June 2016, Vol. 45, No. 2. pp. 45-55
10) Venkatesh, Prasanna (January 30, 2012). "NewSQL - The New Way to Handle Big Data". - https://www.opensourceforu.com/2012/01/newsql-handle-big-data/
11) Harrison G. Next Generation Databases: NoSQL, NewSQL and Big Data, Apress, 2015, 235 p.

Онтологические базы данных

С появлением семантического веба появилось понятие онтологии. Было предложено несколько определений онтологий, наиболее полное и общепринятое следующее [1]:

Онтология - это явная формальная спецификация согласованной концептуализации. Детальное раскрытие этого определения дается в [2].

В настоящее время онтологии широко используются в различных областях. Они стали важной составляющей семантического веба. Разработаны инструменты манипулирования онтологиями, например, Protege. Однако они не предоставляют те возможности, которые дают БД и, прежде всего, постоянное хранение, манипулирование и формулировка запросов к структуре онтологии и ее данным. В связи с этим возникло понятие *онтологической базы данных* (ОнБД) - *это база данных, которая предоставляет возможность сохранять и манипулировать онтологиями, включая как онтологическую структуру, так и данные этой структуры*. В контексте семантического веба было предложено несколько подходов по созданию ОнБД [3–6]. С их обзором можно познакомиться в [7]. Как правило, ОнБД создаются на базе реляционных баз данных. Кроме того, в качестве языка представления онтологий нижнего уровня выбирается RDF.

Модели ОнБД

Предложено несколько моделей ОнБД, наиболее популярные из которых приведены далее.

Бессхемная модель (schema-oblivious), также называемая вертикальной (vertical). В этой модели онтология хранится в единственной тернарной таблице в виде RDF-триплетов <субъект-предикат-объект>. Эта таблица содержит как структуру онтологии, так и ее данные. Модель представлена в Jena [8, 9], 3store [6], Rstar [10], Virtuoso [11], Oracle [12]. Существенное преимущество - простота поддержки модели.

Схемная модель (schema-aware) также называемая бинарной (binary). Каждый класс и каждое свойство онтологии (RDF/S-

схемы) имеет свою собственную таблицу [4, 5, 13, 14]. Классы располагаются в унарных таблицах, а свойства - в бинарных. Таблица свойства объединяет индивиды различных классов, которые обладают этим свойством. Преимущества этой модели - поддержание многозначных свойств. Модель используется в системе управления данными SOR IBM [15].

Дуальная модель. Является расширенным вариантом схемной модели, которая может содержать не только схемы классов и свойств, но и схемы (мета-схемы) структуры онтологии. Одним из вариантов являются мета-схемы IS-A включения одних классов или свойств в другие, описывая таким образом таксономическую иерархию классов и свойств. В варианте ISA-схемы явно задается таксономическая таблица для классов/свойств, экземплярами которой являются пары классов/свойств, находящихся в этом отношении. Кроме того, мета-схема может включать задание области определения (domain) и области значения (range) свойств. Такой поход используется в ОнБД OntoDB [16]

Гибридная модель (hybrid) [13], сочетающая в себе свойства двух предыдущих. Используется тернарная таблица для каждого типа области значения свойства и бинарная таблица для всех экземпляров всех классов.

Горизонтальная модель. Онтология представляется в виде реляционных таблиц, когда свойства класса стают атрибутами таблицы класса. Если же свойство является многозначным, то оно представляется бинарной таблицей. Если все свойства многозначные, то эта модель превращается в бинарную. Данная модель используется в OntoMS [17], OntoDB [16] и Jena2 [18], Существуют две разновидности этой модели:

– *Таблица кластеризации по свойствам* (clustered property table): выделяется группа свойств и строится таблица, содержащая все индивиды (экземпляры) онтологии, которые обладают этими свойствами не зависимо от их принадлежности классу, то есть таблица может содержать индивиды различных классов.
– *Таблица свойство-класс* (property-class table): таблица содержит все индивиды одного класса с заданным набором свойств. Одно и то же свойство может присутствовать в различных таблицах.

В обеих разновидностях, если существуют индивиды, которые не попадают ни в одну из этих таблиц, то они размещаются в вертикальной таблице.

Все эти модели ОнБД были реализованы в существующих RDF-хранилищах (RDFSuite, Jena, Sesame, DLDB, RStar, KAON, PARKA, 3Store, Oracle), исчерпывающий обзор которых приведен в [19].

Вывод в ОнБД

В онтологиях существует проблема вывода, когда по таксономической иерархии включения классов/свойств необходимо построить их транзитивное замыкание. В ОнБД предлагается два подхода решения этой задачи: а) либо предварительно вычислять и материализовывать их (во время компиляции), который был назван MatView, и который используется в бессхемном подходе, б) либо вычислить их тогда, когда в этом появится необходимость (во время выполнения). Второй вариант используется в схемном и гибридном подходе.

Языки

Было предложено множество языков веба и семантического веба, обзор которых приведен в статье [20]. Среди них выделяется класс языков, работающих с форматом RDF. К ним относятся: языки семейства SPARQL (SquishQL, RDQL, SPARQL, TriQL.), языки семейства RQL (RQL, SeRQL, eRQL), языки с реактивными правилами (Algae, iTQL, WQL), дедуктивные языки запросов (N3QL, R-DEVICE, TRIPLE, Xcerpt). Все они в том или ином виде могут быть применены для ОнБД, которые базируются на RDF. Вместе с тем в [21] описывается язык OntoQL, который был разработан для ОнБД OntoDB. Кроме того, в этой статье определена алгебра онтологий, на базе которой построен язык OntoQL.

Онтолого-ориентированный доступ к данным

Онтолого-ориентированный доступ к данным (Ontology Based Data Access -

OBDA) - это совокупность методов, алгоритмов и систем, имеющих отношение к интеграции неоднородных данных. В OBDA предполагается использование онтологий для обеспечения глобального концептуального представления множества локальных неоднородных наборов данных и поддержки отображений между таким глобальным описанием предметной области и локальными схемами баз данных.

С момента своего появления эта область развивалась в нескольких направлениях. Первоначально основное внимание уделялось проблеме трансляции схем данных исходных локальных источников в глобальную схему и последующую ее материализацию, включая подходы, отличные от OBDA, такие как использование технологии Extract-Transform-Load (ETL) в хранилищах данных, которая начала бурно развиваться в 70-х годах прошлого столетия. При этом ее основной задачей было обеспечение трансляции данных (data translation) между различными моделями данных.

В начале XXI столетия проблемы интеграции данных стали еще более актуальными, поскольку организации начали использовать веб-технологии для предоставления доступа к своим данным (с помощью веб-сервисов, API-интерфейсов REST, подходов семантического веба и связанных данных). Доступность и разнородность данных (как по содержанию, так и по формату) в настоящее время находится на беспрецедентном уровне. Было введено понятие «озеро данных» (data lake) для обозначения эволюции хранилищ данных, которая учитывает не только структурированные данные, но и другие типы данных слабоструктурированных и неструктурированных форматов.

В начале 2000-х гг. было предложено ряд подходов по описанию отображений между онтологическими и реляционными схемами баз данных, включая D2R-MAP [22], расширенное D2R [23], R2O [24], VisAVis [25], а также трансформации реляционной модели в онтологическую [26, 27, 28]. Были также реализованы инструментальные средства поддержки таких отображений, например, DataGenie [29], D2RQ [30], D2RMAP [31], RDB2Onto [32].

Учитывая перспективность этого направления, была создана рабочая группа W3C RDB2RDF, которая опубликовала две рекомендации по трансформации содержимого реляционных баз данных в RDF: прямое отображение [33] и R2RML [34]. В обзорной статье [35] кратко описываются 9 языков отображения реляционных баз данных в RDF и приводится их сравнительный анализ относительно 15 выделенным характеристикам.

В 2008 была опубликована статья [36], в которой была предложена идея OBDA - использования онтологий для глобального концептуального представления разнородных источников данных, которая получила всеобщее признание и активно развивается по настоящее время. При этом довольно часто предлагается подход на основе посредников-оболочек (mediators-wrappers) [37], которые используются для преодоления различий между локальными схемами и глобальным представлением. В OBDA эти отображения используются как для трансляции данных (аналогично тому, как это делается в технологии ETL), так и для трансляции запросов [38, 39], когда запросы, написанные в соответствии с глобальной схемой, преобразуются в язык запросов, поддерживаемый исходными источниками данных, а результаты преобразуются обратно в соответствии с глобальным представлением. В статье [40] приводится обзор основных теоретических результатов, методов и средств в области OBDA по состоянию на 2018 г., а также обсуждаются наиболее важные направления исследований в этой области. Теоретические основы OBDA хорошо представлены в [41].

Со временем появилась необходимость преобразовывать в онтологии не только реляционные данные. В результате появились такие языки описания отображений, как RML [42] (для отображения форматов CSV, JSON и XML), xR2RML [43] (для работы с MongoDB), KR2RML [44] (для отображения данных ненормализованной реляционной модели), D2RML [45] (для XML, JSON и REST/SPARQL). Помимо этих декларативных языков описания отображения были предложены недекларатив-

ные SPARQL-Generate [46)], Triplify [47], Helio [48], Tarql [49].

В заключение отметим статью [50], в которой приводится прекрасный аналитический обзор исследований и разработок в области использования онтологий для организации поиска и доступа к данным в базах данных на основании анализа 18 методов и инструментальных средств. А также статью [51], в которой обсуждены общие задачи проблемы онтолого-ориентированной интеграция данных в семантическом вебе, показана важность установления отображений между онтологиями и реляционными базами данных, дан краткий аналитический обзор исследований по этой теме, определена бинарная реляционная модель данных и описано отображение дескриптивной логики ALC и ее расширений в данную модель. В последующей статье [52] полученные теоретические результаты были реализованы с использованием RDF.

Литература

1) Studer R., Benjamins R., Fensel D. Knowledge engineering: Principles and methods. Data & Knowledge Engineering, 25(1–2):161–198, 1998.
2) Guarino N., Oberle D., Staab S. What is an ontology? In Handbook on ontologies, pages 1–17. Springer, 2009.
3) Alexaki S., Christophides V., Karvounarakis G., Plexousakis D., Tolle K.: The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases. In: SemWeb'01: Proceedings of the Second International Conference on Semantic Web - Volume 40 May 2001, pp. 1–13
4) Broekstra J., Kampman A., van Harmelen F. Sesame: A generic architecture for storing and querying RDFand RDF schema. In Proc. of the First Inter. Semantic Web Conf., pp. 54–68, 2002.
5) Pan Z., Heflin J.: Dldb: Extending relational databases to support semantic web queries. In: Proceedings of the 1st International Workshop on Practical and Scalable Semantic Systems (PSSS'03). 2003, pp. 109–113
6) Harris S., Gibbins N. 3store: Efficient bulk RDF storage. In Proc. of the 1st Intern. Workshop on Practical and Scalable Semantic Systems (PSSS'03), 2003. pp. 1-15
7) Theoharis Y., Christophides V., Karvounarakis G. (2005) Benchmarking Database Representations of RDF/S Stores. In: Gil Y., Motta E., Benjamins V.R., Musen M.A. (eds) The Semantic Web – ISWC 2005. ISWC 2005. Lecture Notes in Computer Science, vol 3729. Springer, Berlin, Heidelberg. pp. 685-701
8) McBride B. Jena: Implementing the RDF Model and Syntax Specification. SemWeb'01: Proceedings of the Second International Conference on Semantic Web - Volume 40, May 2001, pp, 23–28
9) Agrawal R., Somani A., Xu Y. Storage and querying of e-commerce data. In: VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc. (2001) 149–158
10) Ma L., Su Z., Pan Y., Zhang M, Liu M. Rstar: an rdf storage and query system for enterprise resource management. thirteenth ACM international conference on Information and knowledge management, 2004, pp. 484 – 491.
11) Erling O., Mikhailov I.: RDF Support in the Virtuoso DBMS. In: Conference on Social Semantic Web (CSSW'07). 2007, Volume 113, pp. 59–68
12) Wu Z., Eadon G., Das S., Chong E.I., Kolovski, V., Annamalai, M., Srinivasan, J.: Implementing an Inference Engine for RDFS/OWL Constructs and User-Defined Rules in Oracle. In: Proceedings of the 24th International Conference on Data Engineering (ICDE'08). 2008, pp. 1239–1248
13) Alexaki S., Christophides V., Karvounarakis G., Plexousakis D., Tolle K. On storing voluminous rdf descriptions: The case of web portal catalogs. In Proceedings of the Fourth International Workshop on the Web and Databases, WebDB 2001, Santa Barbara, California, USA, May 24-25, 2001, in conjunction with ACM PODS/SIGMOD 2001, pp. 43-48
14) Abadi D.J., Marcus A., Madden S.R., Hollenbach K. Scalable Semantic Web Data Management Using Vertical Partitioning. In: Proceedings of the 33rd In-

ternational Conference on Very Large Data Bases (VLDB'07). 2007, pp. 411–422

15) Jing L., Li M.,Lei Z., Jean-Sébastien B., Chen W., Yue P., Yong Y., 2007. SOR: A Practical System for Ontology Storage, Reasoning. In VLDB 2007, 33rd Very Large Data Bases Conference, pp. 1402-1405.
16) Dehainsala H., Pierra G., Bellatreche L. (2007) OntoDB: An Ontology-Based Database for Data Intensive Applications. In: Kotagiri R., Krishna P.R., Mohania M., Nantajeewarawat E. (eds) Advances in Databases: Concepts, Systems and Applications. DASFAA 2007. Lecture Notes in Computer Science, vol 4443. Springer, Berlin, Heidelberg, pp. 497-508
17) Park M.J., Lee J.H., Lee C.H., Lin J., Serres O., Chung C.W.: An Efficient and Scalable Management of Ontology. In: Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA'07). 2007, pp. 975–980
18) Wilkinson K., Sayers C., Kuno H., Reynolds D. 2003. Efficient RDF storage and Retrieval in Jena2. Proceedings of the 1st International Workshop on Semantic Web Database (SWDB'03). pp. 131–150.
19) SWAD-Europe Deliverable 10.2: Mapping Semantic Web Data with RDBMSes. - https://www.w3.org/2001/sw/Europe/reports/scalable_rdbms_mapping_report/
20) Bailey J., Bry F., Furche T., Schaffert S. (2005) Web and Semantic Web Query Languages: A Survey. In: Eisinger N., Małuszyński J. (eds) Reasoning Web. Lecture Notes in Computer Science, vol 3564. Springer, Berlin, Heidelberg, 2005, pp. 35–133
21) Jean S., Aït-Ameur Y., Pierra G. (2006) Querying Ontology Based Database Using OntoQL (An Ontology Query Language). In: Meersman R., Tari Z. (eds) On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE. OTM 2006. Lecture Notes in Computer Science, vol 4275. Springer, Berlin, Heidelberg. pp. 704-721
22) Bizer C. D2R MAP – A Database to RDF Mapping Language. In: Proceedings of the 12th International World Wide Web Conference, Budapest, Hungary, 2003.
23) Barrasa Rodríguez J., Corcho O., Gómez-Pérez A. Fund Finder: A case study of database-to-ontology mapping. In: Proceedings of the 2nd International Semantic Web Conference, ISWC 2003, Florida, USA.
24) Barrasa Rodríguez J., Corcho O., Gómez-Pérez A. R2O, an extensible and semantically based database-to-ontology mapping language. In: Proceedings of the Second Workshop on Semantic Web and Databases, SWDB 2004, 2004 . Springer-Verlag, Berlín, Alemania, pp. 1069-1070.
25) Konstantinou N., Spanos D.-E., Chalas M., Solidakis E., Mitrou N. VisAVis: An approach to an intermediate layer between ontologies and relational database contents. In: Proceedings of the CAISE'06 Third International Workshop on Web Information Systems Modeling (WISM '06), 2006, pp. 1050-1061
26) Li M., Du X., Wang S. Learning ontology from relational database. In: Proceedings of the 4th International Conference on Machine Learning and Cybernetics, 2005, pp. 3410–3415.
27) Shen G., Huang Z., Zhu X., Zhao X. Research on the rules of mapping from relational model to OWL. In: Proceedings of the OWLED*06 Workshop on OWL: Experiences and Directions, 2006.
28) Buccella A., Penabad M., Rodriguez F., Farina A., Cechich A. From relational databases to OWL ontologies. In: Proceedings of the 6th National Russian Research Conference, 2004.
29) Xu Z., Zhang S., Dong Y. Mapping between relational database schema and OWL ontology for deep annotation. In: Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on Web Intelligence, 2006, pp. 248–552.
30) Bizer C., Seaborne A. D2RQ - treating non-RDF databases as virtual RDF graphs. In: Proceedings of the 3rd International Semantic Web Conference (ISWC2004), 2004.

31) Bizer C. Database to RDF mapping language and processor, D2RMAP: http://www.wiwiss.fu-berlin.de/bizer/d2rmap/d2rmap.htm (2016).
32) Seleng M., Laclavik M., Balogh Z., Hluchy L. RDB2Onto: Approach for creating semantic metadata from relational database data. In: Proceedings of the ninth international conference on informatics. Bratislava, Slovak Society for Applied Cybernetics and Informatics, 2007, pp. 113–116.
33) Arenas M., Bertails A., Prud'hommeaux E., Sequeda J. A Direct Mapping of Relational Data to RDF, W3C Recommendation. 27 September 2012. - https://www.w3.org/TR/rdb-direct-mapping/
34) Das S., Sundara S., Cyganiak R. R2RML: RDB to RDF Mapping Language W3C Recommendation. 27 September 2012. - https://www.w3.org/TR/r2rml/
35) Hert M., Reif G., Gall H.C. A comparison of RDB-to-RDF mapping languages. In: Proceedings of the 7th International Conference on Semantic Systems, ACM, 2011, pp. 25–32.
36) Poggi A., Lembo D., Calvanese D., De Giacomo G., Lenzerini M., Rosati R. Linking Data to Ontologies. In: Spaccapietra, S. (eds) Journal on Data Semantics X. Springer, 2008, pp. 133–173.
37) Wiederhold G. Mediators in the architecture of future information systems. Computer, 1992, Vol. 25, No. 3, pp. 38–49.
38) Priyatna F., Corcho O., Sequeda J. Formalisation and experiences of R2RML-based SPARQL to SQL query translation using morph. In: Proceedings of the 23rd international conference on World wide web, ACM, 2014, pp. 479–490.
39) Rodriguez-Muro M., Rezk M. Efficient SPARQL-to-SQL with R2RML mappings. Journal of Web Semantics, 2015, Vol. 33, pp. 141–169.
40) Xiao G., Calvanese D., Kontchakov R., Lembo D., Poggi A., Rosati R., Zakharyaschev M. Ontology based data access: a survey. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence, 2018, pp. 5511-5519.
41) Kontchakov R., Rodríguez-Muro M., Zakharyaschev M. (2013). Ontology-Based Data Access with Databases: A Short Course. In: Rudolph, S., Gottlob, G., Horrocks, I., van Harmelen, F. (eds) Reasoning Web. Semantic Technologies for Intelligent Data Access. Reasoning Web 2013. Lecture Notes in Computer Science, vol 8067. Springer, Berlin, Heidelberg. 2013, pp. 194–229
42) Dimou A., Vander Sande M., Colpaert P., Verborgh R., Mannens E., Van de Walle R. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In: Proceedings of the 7th Workshop on Linked Data on the Web, 2014.
43) Michel F., Djimenou L., Faron-Zucker C., Montagnat J. Translation of relational and non-relational databases into RDF with xR2RML. In: 11th International Conference on Web Information Systems and Technologies (WEBIST'15), 2015, pp. 443–454.
44) Slepicka J., Yin C., Szekely P.A., Knoblock C.A. KR2RML: An Alternative Interpretation of R2RML for Heterogeneous Sources. In: Proceedings of the 6th International Workshop on Consuming Linked Data (COLD 2015), 2015.
45) Chortaras A., Stamou G. D2RML: Integrating heterogeneous data and web services into custom RDF graphs. In: Proceedings of the Workshop on Linked Data on the Web co-located with The Web Conference 2018 (LDOW 2018). CEUR, ceur-ws. org 2073, 2018.
46) Lefrançois M., Zimmermann A., Bakerally N. A SPARQL extension for generating RDF from heterogeneous formats. In: European Semantic Web Conference, Springer, 2017, pp. 35–50.
47) Auer S., Dietzold S., Lehmann J., Hellmann S., Aumueller D. Triplify: light-weight linked data publication from relational databases. In: Proceedings of the 18th international conference on World wide web, ACM, 2009, pp. 621–630.
48) Helio. - https://oeg-upm.github.io/helio/
49) Tarql. - https://github.com/tarql/tarql

50) Muni K., Sheraz Anjum M. The use of Ontologies for Effective Knowledge Modelling and Information Retrieval. Applied Computing and Informatics, 2017, Vol. 14, No, 2, pp. 116-126.
51) Andon P.I. Reznichenko V.A., Chystiakova I.S. Mapping of description logic to the relational data model. Cybernetics and Systems Analysis. 2017. Vol. 53, No 6, pp. 160–175.
52) Chystiakova I.S. Mapping of the description logic into RDF using binary relational data model. Problems in programming 2021. № 1. С. 56–83.

Векторные базы данных

Векторная база данных (ВБД) - это база данных хранения, индексирования и поиска данных, представленных векторной моделью данных.

Векторная модель данных

Векторная модель (vector space model) — в информационном поиске представление коллекции документов векторами из одного общего для всей коллекции векторного пространства.

Вектор - это упорядоченная последовательность чисел. Количество чисел - это размерность вектора. Если числами являются 0 и 1, то вектор бинарный. В общем случае вместо 1 могут стоять произвольные числа. Если в векторе преобладают 0, то он называется разряженным, а если числа - то он плотный.

В информационном поиске *векторное пространство* - это пространство терминов. Под терминами понимаются слова, ключевые слова, выражения. Каждый термин имеет свое измерение в этом пространстве. Вектору соответствует конкретный документ. Каждая позиция вектора соответствует конкретному термину. Количество терминов, с помощь которых описываются документы, составляет размерность пространства. В бинарной модели 0 в позиции вектора указывает, что соответствующий термин отсутствует в документе, а 1 - присутствует. Если вместо 1 используются числа, то они указывают вес/важность/рейтинг соответствующего термина в документе/запросе.

В любом случае документ представляется в виде множества взвешенных терминов без кого либо учета их упорядоченности, синтаксиса и тем более семантики. В литературе такая модель представления документов получила название "мешок слов"[1] (bag-of-words)

[1] Мешок слов — упрощенное представление текста, которое используется в обработке естественных языков и информационном поиске. В этой модели текст представляется в виде мешка (мультимножества) его слов без какого-либо учета грамматики и порядка слов, но с сохранением информации об их количестве.

Определение весов терминов

Три фактора оказывают влияние на важность термина [1]:

- частота использования термина в отдельном документе - это так называемый *локальный вес* термина;
- характер использования термина во многих/всех документов коллекции - это так называемый *глобальный* вес;
- независимость веса от длины документа - это так называемая *нормализованная длина* документа.

Кратко рассмотрим их.

Локальный вес термина

Согласно [2, 3] были предложены следующие локальные метрики:

- *бинарная* (binary - BNRY) - самая простая метрика, которая фиксирует только факт наличия термина в документе;
- *частота термина* (term frequency - TF) - количество вхождений термина в документ. Основное предположение, лежащее в основе этой метрики, заключается в том, что документ, в котором термин встречается многократно, скорее всего, будет иметь отношение к указанному термину. Эта идея впервые была высказана Гансом Петером Луном (Hans Peter Luhn) в 1953 г. [4] и исследована Солтоном в 1973 г. [5]. Простейший случай определения веса термина - это приравнять его количеству вхождений этого термина в документ. В общем случае предлагается функция веса, которая отображают количество вхождений термина в документ в некоторое числовое значение веса термина.

Ганс Петер Лун

- *логарифмическая частота термина* - задается функцией, использующей логарифм количества вхождений термина в документ.
- существуют варианты нормализованных весов (см. "Нормализация длины документа" далее).

Глобальный вес термина

Он определяет вес термина с учетом всех (или определенного множества) документов коллекции. В общем случае учитывается частота документа (document frequency) - количество документов в коллекции, которые содержат данный термин. Били предложены следующие метрики.

- *Обратная частота документа* (inverse document frequency - IDF). Вес термина обратно пропорционален частоте употребления термина во всех документах коллекции. Интуитивно понятно, что термин, упомянутый во многих документах, должен иметь меньший вес, чем тот, который встречается в нескольких документах. Учёт IDF уменьшает вес широкоупотребительных слов. Для каждого уникального слова в пределах конкретной коллекции документов существует только одно значение IDF. Основоположницей данной метрики является Карен Спарк Джонс (Karen Spärck Jones) [6], и она была сформулирована как мера специфичности или уровня детализации, на котором данное понятие представлено термином. Позже эта метрика была переформулирована и исследована в терминах глобального веса. Были также предложены такие ее разновидности, как квадратичный IDF (Squared IDF - SIDF) [7], Вероятностный IDF (Probabilistic IDF - PIDF) [8], Global Frequency IDF (GFIDF) [9]. В работах [10, 11, 12, 13] приводятся аналитические обзоры исследований в этом направлении.

Карен Спарк Джонс

- *Энтропийный* метод оценивания термина (Entropy weighting scheme) [14]. Термину приписывается вес 0, если он встречается по одному разу во всех документах, 1 если он встречается один раз в одном документе и между 0 и 1 при других вариантах.

Нормализация длины документа

Нормализация длины документа при определении веса термина используется для того, чтобы уровнять важность коротких документов по отношению к длинным, которые содержат больше терминов и в большем количестве. Суть заключается в том, чтобы уменьшать вес по мере увеличения размера документов. Приведем наиболее используемые способы нормализации.

- *Косинусная* нормализация - это наиболее используемый метод нормализации в векторной модели данных. Веса терминов в документе нормализуются длиной вектора документа.
- *Нормализованный/расширенный* вес термина (normalized/augmented term frequency - NTF) - количество вхождений термина в документ, разделенное на количество вхождений наиболее часто встречающегося термина в документе.
- *Плотность термина* (term density - TD) - количество вхождений термина в документ, разделенное на количество терминов в документе.
- *Нормализация по размеру* документа - количество вхождений термина в документ, разделенное на размер документа.

Были предложены другие методы нормализации, которые не получили широкого применения [13, 15].

Схемы взвешивания терминов

Кратко представим три основные схемы определения веса терминов. Отметим, что каждый из методов имеет множество вариантов и модификаций, которые не обсуждаются.

TF-IDF

TF-IDF (TF — term frequency, IDF — inverse document frequency) — статистическая мера, используемая для оценки важности термина в документе, являющимся частью коллекции документов. Вес некоторого термина пропорционален частоте употребления этого термина в документе и обратно пропорционален частоте употребления термина во всех документах коллекции [16]. В статье [17] приводится детальный анализ этой метрики. Впервые эта метрика была применена в системе SMART [18]. Является самой популярной и наиболее используемой схемой оценки важности термина, частично благодаря тому, что появилась она еще начале 60-х г. прошлого столетия.

BM25 (OKAPI)

Схема, иногда называемая Okapi BM25 в связи с тем, что система Okapi была первой, в которой она была реализована и описана в 1994 г. [19].

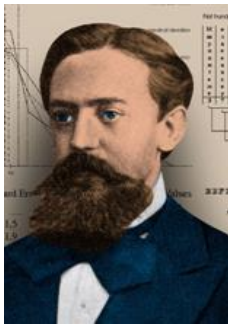
Схема основана на модели вероятностного поиска. Появление этой модели связывают с работой Марона и Кунса [20], опубликованной в 1960 г. Дальнейшее развитие этого направления было предпринято в конце 70-х- начале 80-х годов в работах Робертсона, Джонс и др. [21, 22, 23] и затем активно поддерживалось последующие 30 лет. Хорошими обзорами полученных результатов являются работы [24, 25, 26].

Основной принцип BM25 заключается в том, что конкретный документ может быть признан релевантным конкретному запросу, исходя из предположения, что термины распределяются по-разному и независимо в релевантных и нерелевантных документах. Вес данного термина рассчитывается на основе наличия или отсутствия терминов запроса в каждом документе коллекции. Терминам, которые появлялись в ранее извлеченных релевантных документах данного запроса, следует придавать больший вес, чем если бы они не появлялись в этих релевантных документах [10]. Эта схема хорошо описана в [27, 28]. Является одним из самых надежных и эффективных методов, используемых при поиске информации. Проведенные в последние годы исследования привели к созданию BM25F [29] - модифицированного варианта BM25, в котором учитываются метаданные документа, особенно те из них, которые имеют отношение к структуре документа, включая граф его связей, что привело к его успешному использованию в веб-поиске.

Моделирование языка

Моделирование языка (Language modeling - LM) является расширением вероятностного подхода к поиску. Это вероятно-

стный механизм генерации текста, впервые примененный Андреем Андреевичем Марковым в начале двадцатого века для моделирования последовательностей букв в произведениях русской литературы. В конце 1970-х годов LM успешно применялся для распознавания речи, что было его основным приложением на протяжении многих лет [30]. В 1998 году Понте и Крофт [31] первыми применили языковое моделирование для поиска информации. Их подход заключался в том, чтобы вывести языковую модель для каждого документа и оценить вероятность генерации запроса в соответствии с каждой из этих моделей, а затем ранжировать документы в соответствии с этими вероятностями. Результаты показали улучшение поиска по сравнению с традиционным TF-IDF. В настоящее время языковое моделирование с его многочисленными вариациями является наиболее популярным методом оценивания терминов при информационном поиске.


Андрей Андреевич Марков

Другие схемы и методы

Следует отметить, что были предложены многие другие методы оценивания терминов, но они не получили широкого распространения либо из-за недостаточно эффективных результатов, либо из-за сложности проведения расчетов, либо из-за того и другого. Примерами могут служить модели Term Discrimination Value (TDV) [32], Probabilistic Inverse [2], GFIDF [14] и многие другие , которые в настоящее время редко используется из-за их сложности и слабых результатов. В работах [13, 33] приводятся обзоры различных схем определения весов терминов.

Метрики расстояний

В векторной модели данных и документы и запросы представляются в виде векторов. Поиск документов соответствующих запросу осуществляется на основании сопоставления их векторов. Сопоставление производится на основании понятия сходства/подобия, которое является производным от понятия "расстояние". Два вектора подобны, если они находятся на расстоянии, не превышающем заданное. Было предложено множество метрик для определения расстояния между двумя точками многомерного пространства (евклидово расстояние, косинусная мера, манхэттенское расстояние, расстояние Чебышёва и др.), общее описание которых приведено в подразделе "Сопоставление изображений" раздела "Базы данных изображений". Краткое но содержательное изложение всех перечисленных там и ряда других расстояний приведено в [34]. В векторной модели наиболее используемой метрикой расстояния является косинусное - косинус угла между двумя векторами.

Метрики подобия/похожести

Еще одной задачей ВБД является поиск и отбор документов, подобных запросу. Для этого используются многие методы кластерного анализа, задача которого заключается в группировке набора объектов таким образом, чтобы объекты в одной группе были более похожи (в некотором смысле) друг на друга, чем на объекты в других группах. Прекрасным введением в кластерный анализ является статья в википедии [35].

Среди методов кластерного анализа наиболее используемыми в ВБД являются:

- *Метод k-ближайших соседей* (k-nearest neighbors - k-NN) - нахождение к точек (векторов), которые располагаются ближе всего (или наиболее подобны/похожи) заданной точке (вектору). Предложен был в 1951 г. [36]. Хорошо описан в статье википедии [37].
- *Приблизительный поиск ближайшего соседа.* При больших объемах исходных данных для повышения производительности с возможной потерей точности предлагаются методы приближенного поиска, суть которых заключается в том, что делается "хорошая догадка" относительно того, что собой представляет ближайший сосед [38] .
- *Метод k-средних* (k-means) — стремится минимизировать суммарное квадратичное отклонение точек кластеров от

центров этих кластеров. Был изобретён в 1956 г. математиком Гуго Штейнгаузом (Hugo Steinhaus) [39] и почти одновременно Стюартом Ллойдом (Stuart P. Lloyd) [40]. Хорошо описан в статье википедии [41].

Расширения модели векторного пространства

В связи с огромной популярностью векторной модели было сделано множество предложений по ее расширению. Приведем основные из них.

- *Расширенная векторная модель* Фокса [42]. Традиционная ВМД расширяется использованием помимо терминов понятий других типов в основном библиографического характера (авторы, название, издатель)
- *Обощенная векторная модель* Вонга и др. [43, 44]. Суть обобщения заключается во введении понятия попарной корреляции между терминами, что приводит к отказу от ортогональности векторного пространства. Хорошо описана в энциклопедической статье [45].
- *Семантическая векторная модель* Басиль и Семаан [46]. Предлагается семантическая векторная модель, в которой предоставляется возможность сохранять множество синонимов термина и определять веса терминов с учетом весов их синонимов, а предложенная схема оценивания была названа Продвинутой TF-IDF (Boosted TF-IDF).
- *Нейронная векторная модель* [47]. Модель и метод неконтроллируемого обучения латентного представления слов и документов без какой-либо информации об их релевантности.
- *Тематическая векторная модель* [48]. Еще одна модель, предполагающая отказ от ортогональности векторного пространства в связи с введением взаимосвязей между терминами, например, их подобия/схожести.

Векторное представление слов

Согласно [49] "Векторное представление слов (word embedding) — общее название для различных подходов к моделированию языка и обучению представлений в обработке естественного языка, направленных на сопоставление словам из некоторого словаря векторов небольшой размерности." В ВБД векторами обычно представляются документы и запросы, которые рассматриваются в виде совокупности терминов. В свою очередь векторное представление слов (терминов) позволяет придавать семантику словам. Широко используется для компьютерного представления и обрабоки естественных языков. Энциклопедическая статья [50] дает хорошее содержательное описание векторного представления слов.

Векторные базы данных

ВБД обеспечивают высокую производительность за счет эффективного хранения и обработки данных, развитых поисковых возможностей а также организации быстрого поиска. Они также позволяют интегрировать структурированные и неструктурированные наборы данных в единую систему, обеспечивая высокую масштабируемость для сложных проектов. Наконец, ВБД обладают гибкостью благодаря возможности хранения информационных ресурсов различного типа в единой структуре данных, обеспечивая тем самым возможность обращения к ней как с помощью SQL так и NoSQL.

В связи с этим ВБД широко используются в таких областях искусственного интеллекта (ИИ), как машинное обучение, обработка естественного языка, распознавание образов, компьютерное зрение. ВБД также находят широкое применение в геоинформационных системах, системах оперативной аналитической обработки данных, системах визуализации данных, системах хранения поиска и обработки полнотекстовых документов, изображений, видео и аудио.

Поисковая функция ВБД выглядит следующим образом:

- представление каждого документа ВБД в виде взвешенного вектора (например, TF-IDF);
- представление запроса к ВБД в виде взвешенного вектора (например, TF-IDF);

– вычисление расстояния (например, косинусного) между вектором запроса и вектором каждого из документов ВБД ;
– ранжирование документов согласно определенному критерию подобия;
– выдача пользователю N (например N=10) документов, наиболее соответствующих запросу.

На сайте https://sourceforge.net/software/vector-databases/ кратко описываются 13 наиболее популярных в 2023 г. ВБД с указанием их адресов.

Литература

1) Salton G., McGill M.J. Introduction to Modern Information Retrieval. McGraw Hill Book Co., New York, 1983.
2) SB88] Salton G., Buckley Ch. Term weighting approaches in automatic text retrieval. Information Processing and Management. 1988, Vol. 24, No.5, pp. 513-523
3) Manning C.D., Raghavan P., Schutze, H. (2009). Introduction to Information Retrieval, Cambridge University Press. 2008. In "Scoring, term weighting, and the vector space model" p. 128.
4) Luhn H.P. A new method of recording and searching information. The Journal of the Association for Information Science and Technology. 1953, Vol. 4, No. 1, pp. 14-16
5) Salton G., Yang C. On the specification of term values in automatic indexing. Journal of Documentation. 1973, Vol 29, No. 4, pp. 351-372
6) Spärck Jones K. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation. 1972, Vol. 28, No. 1, pp. 11-21
7) Larson R., Davis M. SIMS 202: Information Organization and Retrieval. UC Berkeley SIMS, Lecture 18: Vector Representation, 2002
8) Kolda T.G. Limited-Memory Matrix Methods with Applications. Applied Mathematics Program. University of Maryland at College Park, pp. 59-68, 1997.
9) Dumais S.T. Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval. Bellcore, 21236, 1992.
10) Spärck-Jones K., Walker S., Robertson S.E. A probabilistic model of information retrieval: development and comparative experiments: Part 1. Information Processing and Management, 2000, Vol. 36, No. 6, pp. 779-808
11) Spärck-Jones K., Walker S., Robertson S.E. A probabilistic model of information retrieval: development and comparative experiments: Part 2. Information Processing and Management, 2000, Vol. 36, No. 6, pp. 809-840
12) Robertson S.E. Understanding Inverse Document Frequency: On Theoretical Arguments for IDF. Journal of Documentation, 2004, Vol.60, No. 5, pp. 503–520
13) Polettini N. The Vector Space Model in Information Retrieval-Term Weighting Problem. 2004. - https://www.researchgate.net/profile/Nicola-Polettini/publication/229053068_The_Vector_Space_Model_in_Information_ Retrieval-Term_Weighting_Problem/links/6173c6900be8ec17a916ac2f/The-Vector-Space-Model-in-Information-Retrieval-Term-Weighting-Problem.pdf
14) Dumais S.T. Improving the retrieval of information from external sources. Behavior Research Methods, Instruments and Computers. 1991, Vol. 23, No. 2, pp. 229-236
15) Singhal A., Buckley Ch., Mitra M. Pivoted document length normalization. SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, 1996, pp. 21–29
16) Roelleke Th. Information retrieval models: foundations & relationships. San Rafael: Morgan & Claypool Publishers; 2013.
17) Aizawa A. An information-theoretic perspective of tf–idf measures. Information Processing and Management, 2003, Vol.39, No. 1, pp. 45–65
18) Salton G. The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall, 1971 - 556 с.
19) [Robertson S.E., Walker S., Jones S., Hancock-Beaulieu M., Gatford M. (November 1994). Okapi at TREC-3. Proceedings of the Third Text REtrieval Conference (TREC 1994). Gaithersburg, USA
20) Maron M.E., Kuhns J.L. On relevance, probabilistic indexing and information retrieval. Journal of the ACM, Vol. 7, No. 3, pp. 216–244, 1960.
21) Robertson S.E., Jones K.S. Relevance weighting of search terms. Journal of the American Society for Information Science. Vol.27, No. 3. pp. 129-146. 1976.
22) Robertson S.E. The probability ranking principle in information retrieval. Journal of Documentation, Vol. 33, pp. 294–304, 1977.
23) Robertson S.E., van Rijsbergen C.J., Porter M.F. Probabilistic models of indexing and searching. In Information Retrieval Research

(Proceedings of Research and Development in Information Retrieval, Cambridge, 1980), (R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, and P. W. Williams, eds.), pp. 35–56, London: Butterworths, 1981.
24) Crestani F., Lalmas M., van Rijsbergen C.J., Campbell I. "Is this document relevant? ... probably": A survey of probabilistic models in information retrieval. ACM Computing Surveys, Vol. 30, No. 4, pp. 528–552, 1998.
25) Jones K.S., Walker S., Robertson S.E. A probabilistic model of information retrieval: Development and comparative experiments. Part 1. In Information Processing and Management, pp. 779–808, 2000.
26) Jones K.S., Walker S., Robertson S.E. A probabilistic model of information retrieval: Development and comparative experiments. Part 2. In Information Processing and Management, pp. 809–840, 2000.
27) Amati G. BM25. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 323-326. Springer, New York, 2018.
28) Okapi BM25. - https://en.wikipedia.org/wiki/Okapi_BM25
29) Robertson M., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval, Vol. 3, No. 4, pp. 333–389, 2009.
30) Hiemstra D., de Vries A. Relating the new language models of information retrieval to the traditional retrieval models (No. TR-CTIT-00-09). Amsterdam: Centre for Telematics and Information Technology (CTIT), University of Twente; 2000.
31) Ponte J.M., Croft W.B. A language modeling approach to information retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 1998. p. 275–281.
32) Salton G., Yang C.S., Yu C.T. A theory of term importance in automatic text analysis. Journal of the American Society for Information Science. 1975, Vol. 26, No. 1, pp. 33–44.
33) Chisholm E., Kolda T.G. New Term Weighting Formulas for the Vector Space Method in Information Retrieval. Oak Ridge National Laboratory United States: N. p., 1999. Web. doi:10.2172/5698.
34) Kumar V., Chhabra J.K., Kumar M. Performance Evaluation of Distance Metrics in the Clustering Algorithms. INFOCOMP, 2014, Vol. 13, No. 1, pp. 38–51.
35) Cluster analysis. - https://en.wikipedia.org/wiki/ Cluster_analysis
36) Fix E., Hodges J.L. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. Report. USAF School of Aviation Medicine, Randolph Field, Texas, 1951, 21 p. - https://apps.dtic.mil/dtic/tr/fulltext/u2/a800276.pdf
37) k-nearest neighbors algorithm. - https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
38) Malkov Yu., Yashunin D. (2016). Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. - https://arxiv.org/abs/1603.09320
39)] Steinhaus H. Sur la division des corps materiels en parties. Bull. Acad. Polon. Sci., C1. III vol IV: 1956, pp. 801—804.
40) Lloyd S. (1957). Least square quantization in PCM. Bell Telephone Laboratories Paper. Later published in: IEEE Transactions on Information Theory, 1982, Vol. 28, No.2, pp. 129-137
41) k-means clustering. - https://en.wikipedia.org/wiki/K-means_clustering
42) [Fox 1983] Fox E.A. (1983). Extending the Boolean and Vector Space Models of Information Retrieval with P-norm Queries and Multiple Concept Types. Ph.D. Dissertation, Cornell University. - https://catalog.hathitrust.org/ Record /009232562
43) WZW85a] Wong S.K.M., Ziarko W., Wong P.C.N. Generalized vector spaces model in information retrieval. SIGIR '85: Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval, 1985, pp. 18–25
44) WZR87] Wong S.K.M., Ziarko W., Raghavan V.V., Wong P.C.N. On Modeling of Information Retrieval Concepts in Vector Spaces ACM Transactions on Database Systems, Vol. 12, No. 2, June 1987, pp. 299-321.
45) Generalized vector space model. - https://en.wikipedia.org/wiki/Generalized_vector_space_model)
46) [Bassil Y., Semaan P. Semantic-Sensitive Web Information Retrieval Model for HTML Documents. 2012. - https://arxiv.org/ftp/arxiv/papers/1204/1204.0186.pdf
47) Christophe Van Gysel, Maarten de Rijke, Evangelos Kanoulas. Neural Vector Spaces for Unsupervised Information Retrieval. ACM Transactions on Information Systems, Vol. 36, No. 4, 2018 Article No. 38, pp. 1–2
48) Becker J., Kuropka D. Topic-based vector space model. Witold Abramowicz, Gary Klein (eds.), Business Information Systems, Proceed-

ings of BIS 2003, Colorado Springs, US, 2003, pp. 7-12
49) WE] Векторное представление слов. - https://neerc.ifmo.ru/wiki/index.php?title=%D0%92%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D1%81%D0%BB%D0%BE%D0%B2
50) Word embedding. - https://en.wikipedia.org/wiki/Word_embedding

Потоковые базы данных

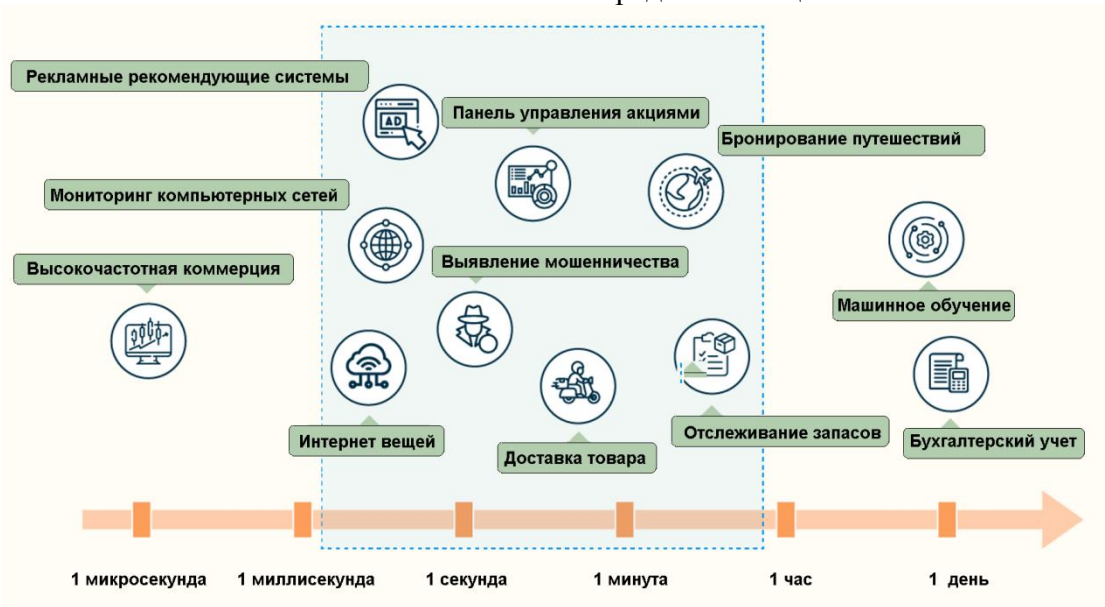
Потоковая база данных (ПтБД) - это хранилище данных, предназначенное для сбора, обработки и/или обогащения непрерывно поступающего потока данных в режиме реального времени, и обеспечения оперативного доступа к данным другим системам сразу после их поступления для дальнейшей обработки и анализа.

В отличие от традиционных БД, которые "складируют" данные с целью их возможного последующего использования, в ПтБД данные включаются в обработку сразу же после их поступления. В отличие от традиционных систем потоковой обработки, которые не запоминают данные, ПтБД сохраняют данные, предоставляя возможность их многократного использования, например, отвечая на запросы пользователей.

Области применения

ПтБД идеально подходят для приложений реального времени, которым требуются актуальные результаты с требованием свежести от долей секунды до минут, таких как аналитика в режиме реального времени, выявление и предотвращение мошенничества, мониторинг функционирования компьютерных сетей и Интернет вещей (Internet of Things - IoT), существенно упрощая их технологический процесс.

На рис. ниже, взятом в [1], приводятся примеры приложений реального времени, предполагающие использование ПтБД.



В приложениях реального времени ПтБД обычно используются вместе с другими системами работы с данными для решения двух важных задач: потоковый ввод и потоковая аналитика.

Потоковый ввод предполагает прием и "очистку" поступающих данных, объединение многих потоков, преобразование данных к нужному виду, запоминание данных в БД для их возможного последующего использования с одновременной передачей их другой системе в реальном времени.

Потоковая аналитика в таких системах фокусируется на выполнении сложных вычислений и предоставлении свежих актуальных результатов на лету. Хороший обзор методов средств потоковой аналитики дан в [2].

Потоковые таблицы, запросы и операции

Основными составляющими ПтБД являются: потоковая таблица, потоковый запрос и потоковые операции

Потоковая таблица (или просто поток) - это таблица, обладающая следующими свойствами:

- строки таблицы упорядочены согласно их поступлению в таблицу, что обеспечивается приписыванием им временных отметок их поступления;
- эта таблица доступна только для добавления строк, существующие строки никогда не могут быть обновлены или удалены

Классические таблицы и потоковые таблицы имеют разную семантику. Классические таблицы представляют данные "в покое", они отражают текущее актуальное состояние предметной области. В свою очередь потоковые таблицы предназначены для представления данных "в процессе", они описывают последовательность наступающих событий, для которых характерна упорядоченность и необратимость.

Потоковый запрос - это запрос, который на входе принимает потоковые и/или обычные таблицы и возвращает потоковую таблицу. Классические БД и ПтБД имеют различную семантику относительно роли запросов в функционировании системы. Классический взгляд предполагает следующее: "если вам надо узнать, что собой представляет ПрО в текущий момент времени, то выполните соответствующий запрос". Он получил название "*вытаскивания*" данных. А ПтБД функционирует по следующему принципу: "если что-то изменилось в ПрО, то немедленно выполняется соответствующий запрос и его результат отправляется всем заинтересованным сторонам". Он получил название "*выталкивания*" данных. В связи с этим было введено понятие потокового (непрерывно выполняющегося) запроса. Результат потокового запроса может быть направлен в потоковую таблицу, традиционную таблицу (например, созданную в виде материализованного представления) или непосредственно клиенту, который должен предварительно "подписаться" на получение соответствующих потоковых данных. В последним случае говорят, что потоковый запрос является транзитным.

ПтБД предоставляют специальные операции над потоками:

- *потоковые окна* - для потока определятся временной интервал, в пределах которого отслеживаются события, порождающие потоковые данные;
- *соединение потоков* - поток может соединяться с другим потоком/таблицей с порождением нового потока
- *объединение потоков* два или более потоков объединяются с порождением нового потока
- *агрегация потока* - множество строк исходного потока агрегируются с порождением одной итоговой строки результирующего потока.

Краткая История ПтБД

Первые попытки создания инструментов для оперативной обработки данных восходят к концу 1980-х годов, когда появились активные базы данных. Они представляли собой расширения к существующим СУБД для обработки событий при помощи триггеров и правил. См. раздел "Активные базы данных" для ознакомления с этим классом БД.

В 1990-х — начале 2000-х появились системы управления потоками данных (Data Stream Management Systems): TelegraphCQ [3], StreamBase [4], StreamSQL [5], Stream [6]. Принцип работы этих инструментов заключался в следующем: при помощи так называемых оконных операторов (window operators) потоки преобразовывались в таблицы, по отношению к которым затем можно было применять SQL-запросы. Появление таких решений было несомненным шагом вперёд, но они не могли обеспечить высокую скорость и производительность при работе с большими потоками данных.

Концепция ПтБД была впервые представлена в академических кругах в 2002 году. Группа исследователей из Брандейского университета, Брауновского университета и Массачусетского технологического института обратила внимание на потребность в управлении потоками данных внутри баз данных и создала первую ПтБД Aurora [7, 8]. Два года спустя была разработана система Borealis [9] - многопроцессорная версия Aurora. В это же время были разработаны ПтБД Gigascope [10], прежде всего для приложений мониторинга сетей, а также система обработки потоков Nile [11] на базе объектно-реляционной СУБД Predator.

Через несколько лет эта технология была принята на вооружение крупными компаниями. Три ведущих поставщика баз данных, Oracle, IBM и Microsoft, последовательно реализовали свои решения для потоковой обработки, известные как Oracle CQL [12], IBM System S [13, 14] и Microsoft SQLServer StreamInsight [15]. Вместо разработки ПтБД с нуля эти поставщики напрямую интегрировали функции потоковой обработки в свои существующие базы данных.

С конца 2000-х годов разработчики, вдохновленные MapReduce, отделили функции обработки потоков от систем баз данных и разработали крупномасштабные системы потоковой обработки (СПО), включая Apache Storm, Apache Samza, Apache Flink и Apache Spark Streaming, Apache Calcite. Эти системы были разработаны для непрерывной и оперативной обработки поступающих потоков данных и предоставления результатов последующим системам в режиме реального времени. Однако, в отличие от ПтБД, СПО не хранят данные и, следовательно, не могут обрабатывать специальные запросы, инициированные пользователями или другими системами.

ПтБД продолжают развиваться параллельно с механизмами потоковой обработки. Две потоковые базы данных, PipelineDB [16] и KsqlDB [17], были разработаны в 2010-х годах.

PipelineDB — это высокопроизводительное расширение PostgreSQL, созданное для непрерывного выполнения SQL-запросов к потоковым данным. Результаты этих непрерывных запросов сохраняется в обычных таблицах, к которым можно обращаться так же, как к любой другой таблице или представлению. Таким образом, непрерывные запросы можно рассматривать как материализованные представления с очень высокой пропускной способностью.

ksqlDB (Kafka SQL) — это ПтБД, которая предоставляет SQL интерфейс для потоковой обработки данных в Apache Kafka.

Следует также отметить разработанную в этот период систему S-Store [18], одна из целей которой - интеграция семантики транзакций с потоковой обработкой данных. Она была разработана на базе H-Store [19] - распределенной OLTP базы данных, полностью находящейся в памяти. H-Store и S-Store также относятся к классу NewSQL-баз данных (см. раздел "NewSQL-базы данных")

Краткое описание многих из приведенных выше систем дано в энциклопедической статье [20].

В начале 2020-х годов появилось несколько облачных потоковых баз данных, таких как:

- RisingWave [21] - распределенная потоковая SQL база данных с открытым исходным кодом, предназначенная для потоковой аналитической обработки данных в приложениях, управляемых событиями;
- Materialize [22] - потоковая SQL база данных для приложений реального времени;
- DeltaStream [23] - масштабируемая унифицированная платформа обработки потоков, управляемая SQL.
- HStreamDB [24] - распределенная ПтБД с открытым исходным кодом, предназна-

ченная для приема, хранения, доступа и обработки потоковых данных реального времени, поступающих из различных источников.

Эти продукты предназначены для предоставления пользователям потоковых сервисов SQL баз данных в облаке. Для достижения этой цели основное внимание уделяется разработке архитектуры, которая полностью использует ресурсы облака для достижения неограниченной горизонтальной масштабируемости и максимальной экономической эффективности

Потоковый SQL

Все потоковые системы и базы данных обладают языком для формулирования поисковых запросов к БД. Большинство потоковых языков запросов представляют собой расширения SQL. Эти расширения являются либо чисто текстовыми, либо используют GUI, с помощью которого пользователи могут построить диаграммы потоков данных, которые затем преобразуются в выражения расширенной реляционной алгебры. Наиболее важным расширением потокового языка, которого нет в традиционных БД, является введение понятия окна, с помощью которого создаются конечные структуры (то есть обычные таблицы) из бесконечных структур (то есть потоков). И различные модели и языки запросов потоковых данных обуславливаются способам создания и использования таких окон

Было реализовано ряд экспериментальных и коммерческих потоковых систем, поддерживающих расширенный вариант SQL. К потоковым SQL-языкам, разработанным в рамках экспериментальных систем, относятся:

- CQL [25] (система STREAM, разработанная в Стэнфорде),
- SQuAl [26, 27] (системы Aurora/Borealis),
- ESL [28, 29] (проект Atlas, выполненный в UCLA),
- GSQL [10] (система Gigascope, разработанная в AT&T Labs-Research).

К коммерческим языкам относятся:

- StreamSQL [30] (система Streambase),
- CCL [31] (система Coral8),
- EQL [32] (система Esper),
- StreaQuel [33] (система Truviso),
- KSQL [34] (система Apache Kafka),
- SQLStreamBuilder [35],
- SQLStreams [36],
- SamzaSQL [37],
- Storm SQL [38] (система Storm).

В статье [39] предлагается подход по классификации функциональных возможностей потоковых языков и приводится сравнительный анализ многих из перечисленных выше языков согласно этой классификации.

Литература

1) What Is a Streaming atabase?. - https://dzone.com/articles/what-is-a-streaming-database
2) Turaga D. Streaming Analytics In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 3793-3801. Springer, New York, 2018.
3) Chandrasekaran S., Cooper O., Deshpande A., Franklin M.J., Hellerstein J.M., Hong W., Krishnamurthy S., Madden S.R., Reiss F., Shah M.A. TelegraphCQ: continuous dataflow processing. In: SIGMOD '03: Proceedings of the 2003 ACM SIGMOD
4) TIBCO StreamBase. - ttps://www.tibco.com/resources/datasheet/tibco-streambase
5) StreamSQL. - https://en.wikipedia.org/wiki/StreamSQL
6) Babcock B., Babu S., Datar M., Motwani R., Widom J. Models and issues in data stream systems. In: PODS '02: Proceedings of the 21st ACM SIGACT-SIGMODSIGART Symposium on Principles of Database Systems. 2002, pp. 1-16
7) Abadi D., Carney D., Cetintemel U., Cherniack M., Convey C., Erwin C., Galvez E., M. Hatoun, Hwang J., Maskey A., Rasin A., Singer M., Stonebraker M., Tatbul N., Xing Y., Yan R., Zdonik S. Aurora: A Data Stream Management System. In ACM SIGMOD Conference, June 2003, San Diego, CA., pp. 666
8) The Aurora Project. - http://www.cs.brown.edu/research/aurora
9) Abadi D., Ahmad Y., Balazinska M., Çetintemel U., Cherniack M., Hwang J.-H., Lindner W., Maskey A.S., Rasin A., Ryvkina E., Tatbul N., Xing Y., Zdonik S.

The design of the Borealis stream processing engine. In: Proceedings of the 2nd Biennial Conference on Innovative Data Systems Research. 2005, pp. 277-289
10) Cranor C.D., Johnson T., Spatscheck O., Shkapenyuk V. Gigascope: a stream database for network applications. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. 2003. pp. 647–651.
11) Hammad M.A., Mokbel M.F., Ali M.H., Aref W.G., Catlin A.C., Elmagarmid A.K., Eltabakh M.Y., Elfeky M.G., Ghanem T.M., Gwadera R., Ilyas I.F., Marzouk M.S., Xiong X. Nile: a query processing engine for data streams. In: Proceedings of the 20th International Conference on Data Engineering; 2004. p. 851.
12) Introduction to Oracle CQL. - https://docs.oracle.com/cd/E16764_01/doc.1111/e12048/intro.htm
13) Bouillet E., Ranganathan A. Scalable, Real-Time Map-Matching Using IBM's System S. In: MDM '10: Proceedings of the 2010 Eleventh International Conference on Mobile Data Management, 2010, pp. 249–257
14) Gedik B., Andrade H., Wu K.-L., Yu P.S., Doo M. SPADE: the systems S declarative stream processing engine. In: SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 2008, pp. 1123–1134
15) Ali M. An introduction to Microsoft SQL server StreamInsight. In: COM.Geo '10: Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application, 2010, Article No.: 66, p. 1
16) PipelineDB. - https://github.com/pipelinedb/pipelinedb
17) ksqlDB. - https://dbdb.io/db/ksqldb
18) Cetintemel U., Du J., Kraska T., Madden S., Maier D., Meehan J., Pavlo A., Stonebraker M., Sutherland E., Tatbul N., Tufte K., Wang H., Zdonik S. S-Store: a streaming NewSQL system for big velocity applications. In: Proceedings of the 40th International Conference on Very Large Data Bases; 2014, Vol. 7, No. 13, pp. 1633–1636
19) Kallman R., Kimura H., Natkins J., Pavlo A., Rasin A., Zdonik S., Jones E.P.C., Madden S., Stonebraker M., Zhang Y., Hugg J., Abadi D.J. H-Store: a high-performance, distributed main memory transaction processing system. In: Proceedings of the VLDB Endowment. 2008, Vol. 1, No. 2, pp. 1496-1499.
20) Ahmad Y., Cetintemel U. Data Stream Management Architectures and Prototypes. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 854-860. Springer, New York, 2018.
21) What is RisingWave?. - https://www.risinrwave.dev
22) Materialize: The Streaming Database. - https://materialize.com/
23) DeltaStream - Unified Stream Processing. - https://www.deltastream.io/
24) HStreamDB. - https://hstream.io/
25) Arvind A., Shivnath B., JenniferW. The CQL continuous query language: semantic foundations and query execution. VLDB J. 2006, Vol. 15, No. 2, pp. 121–142
26) Abadi D., Carney D., Cetintemel U., Cherniack M., Convey C., Lee S., Stonebraker M., Tatbul N., Zdonik S. Aurora: a new model and architecture for data stream management. VLDB J. 2003, Vol 12, No. 2, pp. 120–139.
27) Cherniack M. SQuAl: The Aurora [S]tream [Qu]ery [Al]gebra, Technical Report, Brandeis University, 2003.
28) Bai Y., Thakkar H., Luo C., Wang H., Zaniolo C. A data stream language and system designed for power and extensibility. In: Proceedings of the international conference on information and knowledge management; 2006. p. 337–346.
29) Zaniolo C., Luo C., Wang H., Bai Y., Thakkar H. An introduction to the Expressive Stream Language (ESL), Technical Report, UCLA.
30) Streambase Systems. StreamSQL online documentation set. - http://streambase.com/developers/docs/latest/streamsql/index.html; 2007.
31) Coral8 Systems, Coral8 CCL Reference Version 5.1. - http://www.coral8.com/system/files/assets/pdf/current/Coral8CclReference.pdf; 2007.

32) Codehaus.org. Esper online documentation set. - http://esper.codehaus.org/tutorials/tutorials.html; 2007.
33) Chandrasekaran S., Franklin M. Streaming queries over streaming data. In: Proceedings of the 28th international conference on very large data bases; 2002. p. 203–214.
34) Introducing KSQL: Streaming SQL for Apache Kafka. - https://www.confluent.io/blog/ksql-streaming-sql-for-apache-kafka/
35) What is SQL Stream Builder?. - https://docs.cloudera.com/csa/latest/ssb-overview/topics/csa-ssb-intro.html
36) SQLStreams. https://en.wikipedia.org/wiki/Sqlstream
37) Pathirage M., Hyde J., Pan Yi, Plale B. SamzaSQL: Scalable Fast Data Management with Streaming SQL. In: 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)
38) Storm SQL integration. - https://storm.apache.org/releases/2.2.0/storm-sql.html
39) Cherniack M., Zdonik S. Stream-Oriented Query Languages and Operators. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 3802-3809. Springer, New York, 2018.

Мультимедийные базы данных

Мультимедийная база данных (ММеБД) представляет собой набор связанных мультимедийных данных. Мультимедийные данные включают в себя один или несколько основных типов мультимедийных данных. К ним относятся [1]:

- **текст**: книги, статьи, письма, газеты;
- **графика**: чертежи, эскизы, иллюстрации;
- **изображения**: цветные и черно-белые, рисунки, картины, фотографии, карты;
- **анимационные последовательности:** анимация изображений или графических объектов;
- **видео**: последовательность изображений (кадров), обычно фиксирующая реальные события и обычно воспроизводимая видеоустройством;
- **аудио**: записывается и воспроизводится с помощью специальных звуковых записывающих и воспроизводимых устройств;
- **составные мультимедиа**: комбинация двух или более из вышеперечисленных типов данных.

Мультимедийные СУБД (ММеСУБД) - это программная система, которая управляет данными различного типа, представленными в различных форматах и расположенных в различных медиа-средах и обеспечивает создание, хранение, поиск, доступ и управление данными. В определенном смысле система ММеБД является гетерогенной СУБД, так как она управляет неоднородными (гетерогенными) данными.

Выделяют три класса типов медиа [2]:

- *динамичные* (непрерывные) - медиа, которые изменяются со временем, например, аудио и видео;
- *статичные* (дискретные) - независимые от времени медиа, например, текст, изображения, графика.
- *пространственные* (3D-игры, системы автоматизированного проектирования).

Особенности мультимедийных БД

ММеСУБД должна обеспечивать поддержку традиционных для СУБД функций.

Вместе с тем специфическими характерными особенностями ММеБД являются следующие [2, 3]:

- **Отсутствие структуры**. Обычно мультимедийные данные как таковые не имеют структуры, поэтому к ним не применимы стандартные методы индексирования и контекстного поиска. Вместе с тем центральным аспектом в мультимедийных системах баз данных является понятие модели данных, поэтому ММеБД должны обладать возможностями извлечения из исходного контента (текста, изображений и т.д.) его структуры и обладать возможностями манипулирования ею.
- **Темпоральность/пространственность.** Различные мультимедийные типы данных выдвигают различные требования. Видео, аудио и анимационные последовательности обладают темпоральными требованиями, которые влияют на их хранение, обработку и представление. С другой стороны, изображения, видео и графика обладают пространственными ограничениями.
- **Большие объемы**. Обычно мультимедиа, особенно аудио и видео, большого объема, поэтому требуют специальные методы хранения, индексирования и доступа.
- **Логистика**. Нестандартные медиа могут усложнять обработку. Например, приложения мультимедийных баз данных требуют использования алгоритмов сжатия.
- **Языки мультимедийных запросов**. Эти языки должны иметь широкие возможности для выражения произвольно сложных семантических и пространственно-временных характеристик, связанных с различной мультимедийной информацией. Они должны поддерживать функции манипулирования контентом мультимедийных объектов.
- **Контекстный поиск**. Поиск в мультимедийных базах данных требует значительных вычислительных ресурсов особенно при использовании контекстного поиска, который требует создания методов и средств распознавания и представления семантики медиа-данных и использования методов неполного и нечеткого поиска. Это приводит к разработке сложных пространственно-временных многомерных иерархических структур, взаимосвязанных между собой.
- **Нелинейность презентации**. Презентация мультимедийной информации носит нелинейный характер. Можно перемещаться по различным ее частям, просматривать в прямом и обратном направлениях, воспроизводить и останавливаться на отдельных кадрах, отключать или включать синхронизированные с ней объекты.
- **Синхронизация и интеграция**. Мультимедийные системы должны обладать способностью синхронизировать различные медиа и затем интегрировать их для создания составных мультимедийных объектов, что предполагает работу с данными в реальном масштабе времени.
- **Системная поддержка.** Мультимедийные системы баз данных выдвигают дополнительные требования к компьютерным системам, включая операционные системы, сетевое обеспечение и аппаратное обеспечение.

История мультимедийных БД

Согласно [4, 5] можно выделить следующие три этапа исследований и разработок в области систем ММеБД.

Этап 1. Становление

Этот этап связан с разработкой специальных мультимедийниых систем, которые в основном служили в качестве репозиториев мультимедийных данных и использовали для этого возможности операционных систем.

В первой половине 80-х годов начали появляться первые системы, в которых внедрялись средства, имеющие отношение к медийной среде:

- было разработано ряд систем, в которых были интегрированы графические средства в СУБД для совершенствова-

ния передачи данных между компьютером и человеком [6, 7, 8];
- разработана система, позволяющая хранить и отыскивать в БД оцифрованные изображения [9];
- разработано ряд систем для хранения и поиска документов, состоящих из текстов, изображений и аудиоданных [10-14].

Во второй половине 80-х годов было инициировано ряд исследовательских проектов по созданию экспериментальных систем обработки мультимедийных данных. К ним можно отнести:

- проект по созданию MINOS [15] - объектно-ориентированной мультимедийной информационной системы, предоставляющей интегрированные средства по созданию и управлению сложными мультимедийными объектами;
- объектно-ориентированная система баз данных ORION, содержащая мультимедийный информационный менеджер MIM (Multimedia Information Manager) для обработки мультимедийных данных [16, 17];
- в токийской исследовательской лаборатории IBM были разработаны две экспериментальные системы баз данных MODES1 и MODES2 [18], поддерживающие, в частности, мультимедийные объекты. В лаборатории Hitachi в Кавасаки разработана экспериментальная система мультимедийной базы данных MANDRILL [19]. В работе [20] дан краткий обзор исследований и разработок в Японии в этот период по данной теме;
- в Европе в рамках проекта ESPRIT была разработана мультимедийная архивная система (сервер документов) MULTOS [21].

В первой половине 90-х годов были инициированы исследования по разработке моделей данных и методов поиска в мультимедийных системах, в результате которых были предложены такие модели, как MORE [22], VODAK [23], CORE [24], AIR [25], ADMIRE [26]. В этот же период появились первые коммерческие, реализованные "с нуля" и полноценные системы. К ним относятся MediaDB, которая со временем была переименована в MediaWay [27], JASMINE [28] и ITASCA, которая стала коммерческим преемником системы ORION [16].Они были способны оперировать различными данными, включая мультимедийные, и предоставляли механизмы их вставки, обновления, поиска и представления.

Этап 2. Развитие

Он берет свое начало с середины 90-х годов. Было предложено множество моделей данных для ММеБД, среди которых можно выделить следующие:

- видео-алгебра [29] - это одна из первых ММ-моделей, которая оказала влияние на многие последующие;
- видео-модель данных LHVDM (Logical Hypervideo Data Model) [30], позволяет определять многоуровневые видео-абстракции и семантические ассоциации между ними.
- VisualMOQL [31] - модель данных и язык для работы с изображениями и пространственными объектами;
- обобщенная индексная модель VIDEX [32], реализованная в мультимедийной информационной системе SMOOTH [33], в которой описываются семантические классы и семантические связи (включая пространственные и темпоральные) между ними и мультимедийными объектами;
- компонентная мультимедийная модель [34], включающая физическую, концептуальную, сенсорную и контентную компоненты.

Обзор семантических моделей для ММеБД этого периода представлен в монографии [35].

В 1996 г. в фундаментальной работе [36], была сделана первая успешная попытка сформулировать теоретические основы мультимедийных баз данных.

В этот период создаются полноценные коммерческие системы, которые оперируют мультимедийным контентом, предоставляя объекты сложных типов для различных видов мультимедийных данных. Использование объектно-ориентированного подхода дало возможность определять новые типы данных и операции над ними, поддержи-

вающие такие мультимедийные объекты, как изображения, аудио и видео. В связи с этим широко используемые мультимедийные системы базировались на объектно-реляционных СУБД, которые начали успешно использоваться в 1996-1998 гг. Наиболее развитые решения по созданию мультимедийных систем были реализованы в Oracle 10g, IBM DB2 (IBM DB2 Multimedia Extenders) и IBM Informix (IBM Informix DataBlades). Помимо этого было инициировано ряд исследовательских проектов, в результате которых реализованы полноценные мультимедийные системы баз данных. К ним относятся MIRROR [37], DISIMA [38, 39], SEMCOG [40].

Этап 3. Влияние стандарта MPEG

Третий этап в основном обусловлен появлением в начале 2000-х годов MPEG-стандартов (Moving Picture Experts Group) MPEG-7 (стандарт по описанию мультимедийного контента) и MPEG-21 (стандарт по структуре мультимедиа) [41]. Оба стандарта оказали существенное влияние на создание последующих продуктов, как с точки зрения проектных решений, так и реализации. К ним можно отнести систему MARS (Multimedia Analysis and Retrieval System) [42, 43], MPEG-7 Multimedia Data Cartridge (MDC) - системное расширение СУБД Oracle 9i [44], система MPEG-7 MMDB [45], PTDOM [46], SMOOTH [33],

Мультимедийный SQL

Рассматривая историю ММеБД, отдельно отметим исследования и разработки, посвященные мультимедийному SQL.

В начале 1991 г. Нил Шапиро (Neil R. Shapiro) предложил язык SFQL [47] (Structured Full-text Query Language), целью которого было описание расширения языка SQL для работы с полнотекстовыми документами. Это было, по сути, первой попыткой "облагораживания" SQL мультимедийными данными. В конце 1991 г. SFQL был опубликован в качестве возможного стандарта Ассоциации воздушного транспорта [48], а в 1992 г. разработчики систем текстового поиска, действуя под протекцией организации IEEE, разработали спецификацию этого языка.

В 1992 г. корпорация Oracle выпустила SQL*TextRetrieval Version 2 [49] - программный продукт полнотекстового поиска с использованием СУБД Oracle. В этом же году компания IDI представила систему BASISplus [50], которая поддерживала полнотекстовый поиск в среде реляционной СУБД, а австралийский институт Information Technology Research опубликовал отчет [51] о создании реляционной системы баз данных ATLAS, поддерживающей вложенную структуру данных и расширенный SQL для работы с текстами.

В середине 90-х годов были начаты работы по созданию стандарта мультимедийного SQL (SQL/MM) [52]. В результате было предложено расширение SQL для работы с мультимедийными данными [53] с включением полных текстов, пространственных данных и изображений. Оно было принято для рассмотрения рабочей группой подкомитета SC32 ISO и в 1999-2001 г. было опубликовано 5 документов по стандарту SQL/MM [54-58]. Тем не менее, следует отметить, что до сих пор нет стандарта мультимедийного SQL, принятого ISO/ANSI.

В разделах "Полнотекстовые базы данных", "Базы данных изображений", "Базы данных видео" дается описание соответствующих типов ММеБД.

Гипермедийные базы данных

Часто считают, что термины мультимедиа и гипермедиа являются взаимно заменяемыми. Но это не так. ММеСУБД управляют ММеБД, а гипермедийные СУБД (ГМСУБД) не только управляют ММеБД, но и обеспечивают навигацию между ММеБД с помощью существующих между ними связей. То есть ГМСУБД содержит в себе ММеСУБД. Прекрасным примером ГМБД является Веб.


Билл Аткинсон

Считается, что одной из первых успешных гипермедийных систем является гипертекстовая система программирования HyperCard (https://en.wikipedia.org/wiki/ HyperCard), которая была создана в 1987 г. Биллом Аткинсоном (Bill Atkinson). Она оказала существенное влияние на создание Веба благодаря тому, что ее хорошо знал и высоко оценивал Роберт Кайо (Robert Cailliau), работавший в то время с Тимом Бернерс-Ли по созданию первого веб-браузера. Со временем было создано еще ряд аналогичных систем, среди которых HyperNext, HyperStudio, LiveCode, SuperCard.

Литература

1) Adjeroh D.A., Nwosu K.C. Multimedia database management: Requirements and issues. IEEE MultiMedia, Vol. 4, No. 3, 1997, pp. 24–33
2) Kalipsiz O. Multimedia databases. Proceedings of IEEE International Conference, 2000, pp. 111-115.
3) Ghafoor A. Multimedia database management systems. ACM Computing Surveys, Vol. 27, No 4, 1995, pp. 593–598
4) Kosch H. “Distributed Multimedia Database Technologies supported by MPEG-7 and MPEG-21,” CRC Press. 280 pages. November 2003.
5) Kosch H., Döller M. Multimedia Database Systems: Where are we now? Institute of Information Technology, University Klagenfurt Universitätsstr. Dept. of Comp. Sci.,65/67, A -9020 Klagenfurt, Austria, 2006. https://citeseerx.ist.psu.edu/viewdoc/dowload?doi=10.1.1.90.9044&rep=rep1&type=pdf
6) Fogg D. “Lessons from a ‘Living in a Database’ Graphical Query Interface”, ACM SIGMOD, 1984, pp. 100-106
7) Wilson G., Herot C . “Semantics vs Graphics - To Show or Not To Show,” Proc. VLDB, 1980, pp. 183-197
8) Wong H., Kuo I. “GUIDE Graphical User Interface for Database Exploration”, Proc. VLDB, 1982, pp. 22-32
9) Sloan K.R., Llppman A. “Data Bases of / about / with Images”, IEEE Conf. on Pattern Recognition and Image Processmg, June 1982, pp. 441-446
10) Forsdick H.C., Thomas R.H., Robertson G.G., Travers V.H. “Initilal Experience with Multimedia Documents in Diamond,” IEEE Database Engineering Quarterly Bulletin, Vol. 7, No 3, Sept. 1984
11) Christodoulakis S., Vanderbroek J., Li J., Wan S., Wang Y., Papa M., Bertino E. Development of a Multimedia Information System for an Office Environment,” Proc. VLDB, 1984, pp. 261-271
12) Poggio A., Garcia Luna Aceves J.J., Craighill E.J., Moran D., Aguilar L., Worthington D., Hlght J. “CCWS A Computer-Based Multimedia Information System,” IEEE Computer, Vol. 18, No 10, Oct. 1985, pp. 92-103
13) Yankelovich N., Meyrowitz N., van Dam A. Readmg and Writing the Electronic Book. IEEE Computer, Vol. 18, No 10, Oct. 1985, pp. 15-30
14) Sakata S., Ueda T. A Dlstrlbuted Interoffice Mall System. IEEE Computer, Vol. 18, No 10, Oct 1985, pp. 106-116
15) Christodoulakis S., Theodoridou M., Ho F., Papa,M., Pathria A., "Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and a System," ACM Transactions on Office Information Systems, Vol. 4, No. 4, Oct. 1986, pp. 345-383
16) Woelk D., Kim W., Luther W. An object-oriented approach to multimedia databases. ACM SIGMOD Record, Vol. 15, No 2, 1986 pp 311–325
17) Woelk D., Kim,W., "Multimedia Information Management in an Object-Oriented Database System". VLDB '87: Proceedings of the 13th International Conference on Very Large Data Bases, Sept. 1987, pp. 319–329
18) Kosaka K., Kajitani K., Satoh M., "An Experimental Mixed-Object Database System," in Proc. IEEE CS Office Automation Symposium (Gaithersburg, MD, April

1987), IEEE CS Press, order no. 770, Washington 1987, pp. 57-66.
19) Yamarnoto, Y., Namioka, M., Moki, K., and Sata, K., "An Experimental Multimedia Database System: MANDRILL — its Architecture and Language," Int. Symp. on Database Sys. for Adv. App., Seoul, Korea, Apr.1989.
20) G. Wiederhold, D. Beech, T. Minoura. Multimedia Database Development in Japan. Data Engineering. 1991, Vol. 14, No. 3, pp. 36-45
21) Bertino E., Gibbs S., Rabitti F., Thanos C., Tsichritzis D., "Architecture of a Multimedia Document Server," in Proc. 2nd ESPRIT Technical Week, Brussels, Sept. 1985, pp. 167–168
22) Tsuda K., Yamamoto K., Hirakawa M., Tanaka M., Ichikawa T. MORE: An object-oriented data model with a facility for changing object structures. IEEE Transactions on Knowledge and Data Engineering, 1991, Vol. 3, No 4, p. 444-460
23) Gu J., Neuhold E.J. A data model for multimedia information retrieval. Proc. Multimedia Modeling, Singapore, 1993, p. 113-127
24) Wu J.K., Narasimhalu A.D., Mehtre B.M., Lam A.P., Gao Y.J. CORE: a content-based retrieval engine for multimedia information systems. In Multimedia Systems, Vol. 3, No. 1, 1995, p. 25-41
25) Gudivada V.N., Raghavan V.V., Vanapipat K. A Unified Approach to data Modelling and Retrieval for a Class of Image database applications. In Multimedia Database Systems, Springer, 1996, p. 37-78
26) Velthausz, D.D., C.M.R. Bal, and E.H. Eertink, A Multimedia Information Object Model for Information Disclosure. MMM'96 proceedings of the Third International Conference on MultiMedia Modelling, Toulouse, France, 12-15 November, 1996, p/ 289-304
27) Phillips B. Mediaway presses access to multimedia database. PC Week, 13(7), 1996, 39-40.
28) Khoshafian S., Dasananda S., Minaasian N., The Jasmine Object Database: Multimedia Applications for the Web (Morgan Kaufmann Publishers,1998).
29) Weiss R., Duda A., Gifford D.K. Composition and search with a video algebra. IEEE MultiMedia, 2(1): 12-25, 1995.
30) Jiang H., Montesi D., Elmagarmid A.K. Integrated video and text for content-based access to video databases. Multimedia Tools and Applications, 9(3): 227- 249, 1999
31) Oria V., Ozsu M.T., Xu B., Cheng L.I., Iglinski P. VisualMOQL: The DISIMA visual query language. In IEEE International Conference on Multimedia Computing and Systems, Vol. 1, pp. 536-542, Florence, Italy, June 1999.
32) Tusch R., Kosch H., Boszormenyi L. VideX: An integrated generic video indexing approach. In Proceedings of the ACM Multimedia Conference, pages 448-451, Los Angeles, USA, Oct.-Nov. 2000.
33) Kosch H., Tusch R., Boszormenyi L., Bachlechner A., Dorflinger B., Hofbauer C., Riedler C., Lang M., Hanin C. SMOOTH - A distributed multimedia database system. In Proceedings of the International VLDB Conference, pages 713-714, Rome, Italy, September 2001.
34) Jaimes A. A component-based multimedia data model. MHC '05: Proceedings of the ACM workshop on Multimedia for human communication: from capture to convey, 2005, pp. 7–10
35) Chen S.-C., Kashyap R.L., Ghafoor A. Semantic Models for Multimedia Database Searching and Browsing. Kluwer Academic Publishers, 2000, 148 p.
36) Marcus S., Subrahmanian V.S. Foundations of multimedia database systems Journal of the ACM, Vol. 43, No 3, 1996, pp. 474–523
37) Arjen P. de Vries, Mark G. L. M. van Doorn, Henk M. Blanken, Peter M. G. Apers, The MIRROR MMDBMS architecture. Proc. of the International Conference on Very Large Databases, Edinburgh, Scotland, 1999, 758-761
38) Oria V., Özsu M.T., Li X., Liu L., Li J., Niu Y., Iglinski P.J. Modeling images for content-based queries: The DISIMA approach. In Proceedings of Visual'97, pp. 339-346, San Diego, California, December 1997.

39) Oria V., Özsu M.T., Iglinski P., Lin S., Ya B. DISIMA: A distributed and interoperable image database system, Proc. of the ACM SIGMOD International Conference of Management of Data, Dallas, Texas, USA, 2000, p. 600.
40) Li W.-S., Candan K.S., Hirata K., Hara Y. SEMCOG: an object-based image retrieval system and its visual query language. In Proceedings of ACM SIGMOD International Conference on Management of Data, pages 521—524, Tucson, Arizona, May 1997.
41) Kosch M. Distributed Multimedia Database Technologies supported by MPEG-7 and MPEG-21. CRC Press. Nov. 2003., 280 p.
42) Porkaew K., Ortega M., Mehrotra S. Query reformulation for content based multimedia retrieval in MARS. In: IEEE International Conference on Multimedia Computing and Systems, vol. 2, Florence, Italy, 1999, p. 747-751.
43) Chakrabarti K., Ortega-Binderberger M., Mehrotra S., Porkaew K. Evaluating refined queries in top-k retrieval systems. IEEE Transactions on Knowledge and Data Engineering (TKDE), Vol. 16, No 2, 2004, pp. 256-270
44) Döller M., Kosch H. An MPEG-7 Multimedia Data Cartridge, Proc. of the SPIE conference on Multimedia Computing and Networking 2003 (MMCN 2003), Santa Clara, California, USA, 2003, 126-137
45) Döller M., Kosch H. The MPEG-7 Multimedia Database System (MPEG-7 MMDB). Journal of Systems and Software, Vol. 81, No 9, 2008, pp. 1559-1580
46) Westermann, Utz, Klas, Wolfgang, 2006. PTDOM: a schema-aware XML database system for MPEG-7 media descriptions. Software: Practice and Experience 36 (8), 785–834.
47) Shapiro N.R., Diamantopoulos E., Cotton P. CD-ROM Disc Interchangeability Standards: Beyond ISO 9660 with Structured Full-Text Query Language (SFQL), ATA/ AIA 89-9C Monograph, April 1991.
48) ATA 89-9C SFQL Committee, "Advanced Retrieval Standard —SFQL: Structured Full-text Query Language," ATA specification 100, Rev 30, Version 2.2, Prerelease C, Air Transport Association, ATA 89-9C.SFQL V2.2/PR-C (October 1991) 84 p.
49) Oracle Corporation, SQL *TextRetrieval Version 2 Technical Overview, Oracle Corporation, 1992. 45 pp.
50) Seyboid Publications, "IDI Pursues Document Management," Report on Publishing Systems, Vo1. 21, No. 16, May 1992.
51) Sacks-Davis R., Kent A., Ramamohanarao K., Thorn J., Zobel J., "Atlas: a nested relational database system for text applications", Technical Report CITRI/TR-92-52, Collaborative Information Technology Research, Victoria, Australia, July 1992.
52) International Organization for Standardization, ISO/IEC JTC1/SC21 Information Retrieval, Transfer and Management for OSI WG3 Database: ISO/IEC SC21/WG3 N1679 SQL/MM SOU-004 ISO Working Draft SQL Multimedia and Application Packages (SQL/MM) - Part 2: Full-Text, March 1994.
53) Melton J., Eisenberg A. SQL multimedia and application packages (SQL/MM). ACM SIGMOD Record, Vol. 30, No 4, 2001, p. 97–102
54) ISO/IEC 13249-1:2000, Information technology — Database languages — SQL Multimedia and Application Packages — Part 1: Framework, International Organization for Standardization, 2000.
55) ISO/IEC 13249-2:2000, Information technology — Database languages — SQL Multimedia and Application Packages — Part 2: Full-Text, International Organization For Standardization, 2000.
56) ISO/IEC 13249-3:1999, Information technology — Database languages — SQL Multimedia and Application Packages — Part 3: Spatial, International Organization For Standardization, 2000.
57) ISO/IEC 13249-5:2001, Information technology — Database languages — SQL Multimedia and Application Packages — Part 5: Still Image, International Organization For Standardization, 2001.
58) (ISO/IEC) FCD 13249-6, Information technology — Database languages — SQL Multimedia and Application Packages — Part 6: Data Mining. [FCD = Final Committee Draft for ballot]

Полнотекстовые базы данных

Полнотекстовая база данных (ПТБД) *- это база данных, которая содержит полнотекстовые документы и предоставляет возможность их отыскивать. Документы могут содержать обычный текст, структурированный, слабо структурированный или неструктурированный, может иметь специальные описательные элементы, так называемые метаданные, а также иметь мультимедийные компоненты.*

Исторически вопросом хранения текстовых документов в компьютерах начали заниматься практически в то же время, как и хранение числовых данных. Первые документальные системы предоставляли возможность редактировать и форматировать тексты и являлись частью издательских систем. Электронный документооборот стал неотъемлемой частью автоматизации делопроизводства, которая начала активно развиваться в 1970-е и 1980-е годы. Хранение полнотекстовых документов в компьютерах привело к появлению гипертекстовой и гипермедийной технологии, которая в настоящее время составляет ядро всемирной паутины.

Одной из первых систем управления полнотекстовыми документами была STAIRS (STorage And Information Retrieval System - система хранения и поиска информации), разработанная в IBM в 1969 г. для своих внутренних потребностей, а в 1973 г. была представлена как коммерческий продукт. Однако ПТБД стали широко использоваться только в начале 1990-х годов, когда компьютерные технологии хранения сделали их экономически выгодными и технологически возможными. Существует два основных класса ПТБД: расширение классических библиографических баз данных до ПТБД и ПТБД на основе Интернет (на основе поисковых систем или XML).

Метаданные

В ПТБД полнотекстовые документы могут содержать описания, например, авторы, название, аннотации, ключевые слова и УДК для научных статей. Такие описания получили название метаданных. Различные типы информационных ресурсов могут иметь различные наборы метаданных, которые получили название схем метаданных. Например, разработаны схемы метаданных для описания персон и организаций (vCard и FOAF), библиографических ресурсов (MARC, UNIMARC, DC), музейных и исторических ценностей (CDWA), архивов и электронных ресурсов (GILS, EAD) и многие другие. В электронных библиотеках наиболее используемой является схема Дублинскокго Ядра (Dublin Core)

Индексация

При небольшом количестве документов незначительного размера при поиске производится последовательный просмотр всего документа. Однако при больших размерах последовательный просмотр не эффективен, поэтому в этом случае предварительно производится индексирование - построение списка поисковых терминов с указанием, в каких местах они встречаются. В общем случае качестве поисковых терминов выступают все слова текста документа. Кроме текста самого документа индексации смогут подвергаться метаданные. Для обеспечения более изысканных поисковых возможностей процесс индексации может включать дополнительны возможности, например:

- не учитывать незначащие слова, такие как местоимения, предлоги, союзы, междометия и т.д.;
- помимо места расположения слова запоминать его порядковое место в тексте (поиск по близости слов в фразе);
- вместе со словом запоминать его контекст, то есть фрагмент текста, в котором оно расположено;
- все однокоренные слова приводятся в индексе один раз в виде одного нормированного слова.

Языки запросов

Языки запросов современных поисковых систем в ПТБД предоставляют развитые средства более точной формулировки того, что надо найти, которые включают:

- спецификацию пространства поиска, то есть подмножества ПТБД, в котором следует производить поиска;

– использование логических выражений;
– спецификацию как отдельных поисковых терминов, так и фраз;
– спецификацию расстояния между словами, на котором они находятся в документе;
– использование регулярных выражений;
– спецификацию неточного поиска, то есть нахождение документов, которые содержат слова в каком-то смысле близкие к тем, которые указаны в поисковом выражении.

Все эти и многие другие возможности реализуются как с помощью развитых средствах индексирования, о которых говорилось выше, так и разработки современных механизмов поиска, обзор которых приводится далее.

Далее мы будем использовать следующие термины, которые являются устоявшимися в публикациях по алгоритмам поиска:
– **образец** (sample) - термин или фраза, который отыскивается
– **текст** (text) - документ, в котором производится поиск;
– **сопоставление** (matching) - процедура нахождения образца в тексте.

Существуют различные способы классификации алгоритмов сопоставления, например:
– точное/неточное сопоставление;
– сопоставление слева неправо, справа налево, в указанном порядке, в произвольном порядке;
– с учетом или без учета стоп-слов;
– нахождение в тексте единственного или всех образцов;

и многие другие. Мы за основу берем первый вариант, а в пределе каждого из этих классов будем приводить их разновидности.

Алгоритмы точного сопоставления

Они предполагают точное соответствие текста образцу. Известны следующие пять вариантов: символьный, с хешированием, автоматный, бит-параллельный, гибридный.

Символьный подход

Символьный подход является классическим подходом сопоставления строк, который предполагает посимвольное сравнение образца с текстом.

Самым простым вариантом является ***полный перебор*** (brute-force). Сравнение образца начинается с самого начала текста. Порядок посимвольного сравнения не определяется. В случае неудачного сравнения образец сдвигается на одну символьную позицию вправо и процесс сравнения повторяется. С момента его появления были предприняты большие усилия по разработке более эффективных алгоритмов, основные из которых кратко описаны ниже.

Алгоритм Кнута–Морриса–Пратта (Knuth–Morris–Pratt – KMP) — эффективный алгоритм, осуществляющий поиск подстроки в строке. Время работы алгоритма линейно зависит от объёма входных данных. Первоначально алгоритм был предложен в 1970 г. Джеймсом Моррисом (James H. Morris) и Воганом Праттом (Vaughan Pratt) [1] и независимо от них исследован Дональдом Кнутом (Donald Knuth) [2]. Наконец эти три ученых опубликовали совместную статью в 1977 г. [3]. Независимо от них в 1971 г. советский ученый Юрий Матиясевич предложил аналогичный алгоритм при исследовании задачи сопоставления строк в двоичном алфавите [4].

Алгоритм Бойера-Мура (Boyer-Moore – BM). В 1977 г. был предложен алгоритм сопоставления Бойера-Мура (BM) [5]. Разработан Робертом Бойером (Robert S. Boyer) и Джеем Муром (J Strother Moore), является алгоритмом общего назначения, и стал стандартным и эталонным алгоритмом этого класса. Алгоритм предполагает две фазы:
– Фаза предобработки. Согласно определенному правилу строится таблица, определяющая величины, на которые следует производить сдвиги образца вправо в случае его неудачного сравнения с текстом.
– Фаза сравнения. Образец сравнивается с текстом справа налево. В процессе сравнения вычисляется значение сдвига по умолчанию.

Алгоритм BM является наиболее эффективным алгоритмом поиска для многих приложений, например, текстовых редакторов. Он хорошо работает для алфавитов умеренного размера и длинных шаблонов, однако размеры алфавита и образцов существенно влияют на время предобработки [6]. Алгоритм BM подробно описан в [7]. Со временем появилось множество алгоритмов, которые либо модифицируют BM, либо объединяют его с другими подходами. К модифицирующим вариантам BM, которые улучшают некоторые его характеристики, относятся алгоритмы Хорспула (Horspool) [8], Чжу-Такаока (Zhu-Takaoka) [9], Turbo-BM [10], Апостолико - Джанкарло (Apostolico and Giancarlo) [11] Смита (Smith) [12], Райта (Raita) [13], Крочемора [14], Берри-Равиндрана [15]. В свою очередь гибридные BM-подходы предполагают интеграцию BM-алгоритма с другими методами с целью повышения производительности. Они описаны в статьях [16–22].

Символьный подход лучше всего подходит для приложений, в которых отыскиваются большие текстовые образцы.

Подход с хешированием

Хеш-подход предполагает предварительно перед сравнением применять хеш-функцию к образцу и сравниваемой строке с тем, чтобы сравнивать не символы, а числа, что делается намного быстрее. Он был разработан в 1987 году Майклом Рабином (Michael O. Rabin) и Ричардом Карпом (Richard M. Karp).[23]. Предполагает наличие двух фаз:

- предобработка - это однократное вычисление хеш-функции образца и многократное вычисление хеш-функций сравниваемых подстрок текста
- сравнение хеш-функций образца и подстроки. В случае их равенства производится дополнительное посимвольное сравнение образца с подстрокой, так как хеш-функция может выдавать одинаковые числа для разных строк.

Проблема данного подхода заключается в пересчитывании хеша для каждой подстроки. Ее решение состоит в использовании для подсчёта следующего хеш-значения текущего хеша, что достигается использованием так называемого кольцевого хеша. Рабин и Карп предложили использовать полиномиальный хеш, являющийся разновидностью кольцевого.

Хорошее описание алгоритма Рабина-Карпа приведено в статье [24]. Было предложено еще несколько алгоритмов поиска с хешированием, которые описаны в [25-27].

Оригинальной разновидностью хеш-подхода является так называемый q-gram-подход, который предполагает разбиение образца на части, вычисление хеш-функций каждой части и последующее их использование при сравнение образца с подстрокой текста. Примерами являются алгоритмы, описанные в [28, 29].

Этот подход лучше всего используется там, где нужна повышенная скорость сопоставления строк.

Автоматный подход

Автоматный подход в решении задачи сопоставления строк предполагает использования концепции автомата. Одним из них является так называемый подход с использованием суффиксного автомата - он использует два взаимосвязанных, однако различных автомата: детерминированный ациклический конечный автомат (deterministic acyclic finite state automaton) [30], представляющий конечное множество строк, и суффиксный автомат, выполняющий функцию суффиксного индекса [31].

Суффиксный автомат - это наименьший частичный детерминированный конечный автомат, который распознает набор суффиксов данной строки. Граф состояний суффиксного автомата называется «ориентированным ациклическим графом слов» (directed acyclic word graph - DAWG). Суффиксный автомат был впервые описан группой учёных из Денверского и Колорадского университетов в 1983 году [32]. Имеется три разновидности автоматного подхода, которые кратко представлены далее.

1) ***Ориентированный ациклический граф слов (DAWG).*** DAWG - это структура данных, которая обеспечивает быстрый поиск слов. В DAWG вершина представляет символ. Специальная вершина представляет начальный символ. Часть вершин являются конечными. Вершины имеют направленные

связи. Перемещение от начальной вершины к другим вдоль связей решает задачу сопоставления. Были исследованы следующие разновидности DAWG-сопоставлений: обратный недетерминированный (backward non-deterministic) DAWG [33], двойной прямой (double-forward) DAWG [34], алгоритм сопоставления на основе обратного предсказания (Backward oracle matching - BOM) [35] и его разновидностей [36, 37].

2) ***Широкое окно***. Широкое окно - это окно, размер которого превышает размер образца, в связи с чем в результате перемещения образца в процессе поиска окна перекрываются, что дает определенные преимущества. Впервые идея широких окон была предложена в 2005 г. в работе [38]. Эта особенность широких окон учитывается при использовании суффиксного автомата. В последующим этот алгоритм был усовершенствован в работах [39, 40]. Данный подход также относят к классу бит-параллельных алгоритмов.

3) ***Автоматный подход с пропуском ненужных попыток сопоставления***. Автоматный подход к сопоставлению темпоральных (то есть зависимых от времени) образцов с возможным пропуском ненужных попыток сопоставления был предложен в работе [41]. Темпоральный подход к проблеме сопоставления характерен для систем реального времени и веб-приложений.

Бит-параллельный подход

Побитовые операции над компьютерными словами типа NOT, OR, AND, XOR обладают присущим им параллелизмом. Считается [42, 43], что битовый алгоритм точного поиска был изобретен венгерским ученым Балинтом Дёмёлки (Dömölki) [44, 45] в 1964 году и расширен Шьямасундаром (Shyamasundar) [46] в 1977 г., прежде, чем он был заново изобретен Рикардо Баеза-Йейтсом и Гастоном Гонне (Ricardo Baeza-Yates and Gaston Gonnet) в 1992 г. [6], получивший название Shift OR. Этот алгоритм в дальнейшем был расширен и усовершенствован в [47–51].

После Shift OR в 1998 г. был предложен бит-параллельный алгоритм BNDM (Backward Non Deterministic Matching) [52, 53], который относится к классу Shift AND алгоритмов, то есть используются операции Shift и AND для параллельного поиска. Впоследствии были предложены усовершенствованные варианты BNDM: TNDM (Two way Non Deterministic Matching) [54], SBNDM (Simplified BNDM) [54], BNDMq и SBNDMq (q-gram BNDM/SBNDM) [55], MBNDMq (Multiple BNDMq) [56], LBNDM (long BNDM) [54], FBNDM (Forward BNDM) [57].

Одной из разновидностей бит-параллельного поиска является аппаратный подход, предполагающий использование компьютерной SIMD-архитектуры. SIMD-алгоритмы разработаны для ускорения выполнения сопоставления строк с использованием аппаратного подхода, а именно, функциональных возможностей SIMD-команд. К ним относятся алгоритмы, описанные в работах [54, 59, 60]. Хороший обзор бит-параллельных алгоритмов приведен в работе [61].

Этот подход особенно быстрый, когда образец не превышает размер компьютерного слова.

Гибридный подход

Он хорошо подходит для решения трудных задач, так как он сочетает в себе преимущества различных алгоритмов. Многие алгоритмы используют гибридную концепцию. Например, алгоритмы, описанные в [62–64] сочетают в себе символьный и автоматный подходы, а [65, 66] - символьный и хешированный.

Алгоритмы неточного сопоставления

Были также предложены и исследованы алгоритмы неточного (приблизительного) сопоставления.

Неточные алгоритмы предполагают введение понятия расстояния между строками и нахождения тех, которые располагаются на настоянии, не превышающем заданное относительно исходной строки.

В связи с этим было введено понятие расстояния редактирования как способ количественной оценки того, насколько две строки (не) похожи друг на друга, путем подсчета минимального количества операций, необходимых для преобразования одной строки в другую.

В статье "String metric" в википедии (https://en.wikipedia.org/wiki/String_metric) представлены 20 видов метрик на строках, Приведем те из методов определения степени похожести, которые являются наиболее упоминаемыми в научной литературе.

Расстояние Хэмминга (Hamming distance). Введено Ричардом Хэммингом (Richard Hamming) [67] в 1950 г. и определяет число позиций, в которых соответствующие символы двух слов одинаковой длины различны. В более общем случае расстояние Хэмминга применяется для строк одинаковой длины и служит метрикой различия объектов одинаковой размерности.

Расстояние Левенштейна. Введено советским ученым Владимиром Левенштейном [68, 69] в 1965 г. и определяется для строк различной длины как количество односимвольных операций вставки, удаления и замены, которые переводят одну строку в другую. Вклад в изучение этого расстояния также внёс Дэн Гасфилд (Dan Gusfield) [70]

Расстояние Дамерау-Левенштейна (Damerau–Levenshtein). Является обобщением расстояния Левенштейна, предложено Фредериком Дамерау (Frederick J. Damerau) [71]. Это мера разницы двух строк символов, определяемая как минимальное количество операций вставки, удаления, замены и транспозиции (перестановки двух соседних символов), необходимых для перевода одной строки в другую.

Расстояние Джаро-Винклера (Jaro–Winkler distance). Было предложено в 1990 году Уильямом Э. Винклером (William E. Winkler) [72] на основе расстояния Джаро (Matthew A. Jaro) [73], предложенного в 1989 г.. Неформально, расстояние Джаро между двумя словами — это минимальное число односимвольных перестановок, которое необходимо для того, чтобы изменить одно слово в другое.

Наибольшая общая подпоследовательность (longest common subsequence) [74] - определяет расстояние с использованием операций вставки и удаления (без замены).

На основе перечисленных выше расстояний было предложено множество алгоритмов неточного сопоставления строк, с которыми можно познакомиться в обзорах [75–81].

Расстояние при упорядоченном алфавите. Все перечисленные выше алгоритмы неточного сопоставления строк не учитывают факт возможного упорядочения символов алфавита. В свою очередь упорядоченный алфавит позволяет вводить расстояние межу символами алфавита, между двумя строками и на их основании - соответствующие методы сопоставления строк. Так, например, в работе [82] определяются Δ-расточние, Г-расстояние и (Δ; Г)-расстояние между строками и соответствующие им методы неточного сопоставления строк.

Математические модели информационного поиска

Было предложено ряд математических моделей информационного поиска, которые нашли широкое практическое применение. Приведем наиболее используемые из них.

Модель векторного пространства (vector space model - VSM). Алгебраическая модель представления текстовых документов в виде векторов из одного общего для всей коллекции векторного пространства, например, ключевых слов, которые используются в качестве. измерений. Она используется для фильтрации информации, поиска информации, индексации и ранжирования по релевантности. Впервые была предложена Джерардом Солтоном в 1963 г. [83] и затем применена для индексирования и кластеризации [84]. Также впервые была использована в системе информационного поиска SMART [85], краткое описание которой приведено в [86]. Кроме того, используется в таких известных пакетах, как Apache Lucene, Gensim, Weka. В работе [87] была предложена обощенная модель векторного пространства.

Вероятностная модель. Впервые была предложена Мароном и Кунсом (Maron and Kuhns) в 1960 г. [88] и в дальнейшем развита в работах [89, 90]. В этой модели документы и запросы также представляются в виде векторов, Однако их сравнение производится с использованием функции вероятностного сопоставления. Веротностная модель основывается на оценке вероятности

того, что документ релевантен информационным потребностям пользователя, выраженным в запросе.

Согласно [91] существует три самостоятельных направления вероятностного поиска: *вероятностная схема релевантности* (Probabilistic Relevance Framework - PRF) [92], *вероятностная модель языка* (Probabilistic Language Model - PLM) [93] и *отклонение от случайности* (Divergence From Randomness - DFR) [94].

В работе [95] дается аналитический обзор исследований в этом направлении.

Модель логического вывода. В 1964 г. Марон (Maron) [96] обратил внимание на существенную разницу в информационном поиске между процедурой точного сопоставления с использование булевой логики и механизмом логического вывода. Во втором случае возможен вывод больше релевантной информации, чем было задано в запросе и явно хранится в БД. Со временем эта идея была воплощена в процесс информационного поиска в виде модели логического вывода. Считается, что основателем моделей информационного поиска этого класса является Рейсберген (Rijsbergen) [97].

Механизм логического вывода также воплощен в дедуктивных базах данных (см. раздел «Дедуктивные базы данных»).

Латентно-семантический анализ (Latent semantic analysis - LSA). Это метод обработки информации на естественном языке, анализирующий взаимосвязь между библиотекой документов и терминами, в них встречающимися, и выявляющий характерные факторы, присущие всем документам и терминам. В области информационного поиска данный подход называют латентно-семантическим индексированием (Latent semantic indexing - LSI). Теоретической основой LSA являются: сингулярное разложение матриц (Singular Value Decomposition - SVD) - стандартный метод в линейной алгебре и статистике, факторный анализ [98], латентный анализ классов [99], VSM. LSA был предложен в 1988 г. [100], а LSI - в 1990 [101, 102]. Хорошим введением в LSA является статья [103], а детальный математический анализ выполнен в [104]. В [105] анализируются области практического использования LSA.

Вероятностный латентно-семантический анализ (Probabilistic Latent Semantic Analysis - PLSA). Также известный как вероятностное латентно-семантическое индексирование (PLSI) в области информационного поиска. Это статистический метод анализа корреляции двух типов данных. Является дальнейшим развитием LSA. Применяется в таких областях как информационный поиск, обработка естественного языка, машинное обучение и смежных областях. Впервые был опубликован в 1999 году [106].

Латентное размещение Дирихле (Latent Dirichlet allocation - LDA). Позволяет определять тематическое моделирование коллекции документов. Документ содержит множество слов и с каждой темой ассоциируется набор определенных слов. Задача LDA - определить темы, которым принадлежит документ. Считается, что документы с одинаковыми темами используют одинаковые наборы слов. Применительно к генетике (population genetics) модель LDA была предложена в 2000 г. [107], а в 2003 г. она впервые была представлена применительно к машинному обучению [108]. Также используется во многих приложениях по обработке естественного языка и информационного поиска.

Метод главных компонент (principal component analysis, PCA). Один из основных способов уменьшить размерность данных с потерей минимального количества информации. Изобретён Карлом Пирсоном [109] в 1901 году. Основной задачей PCA является замена исходных данных на некие агрегированные значения в новом пространстве, решая при этом две задачи: 1) объединение наиболее важных значений в меньшее количество параметров, но более информативных, 2) уменьшение шума в данных. Применяется во многих областях, таких как эконометрика, биоинформатика, распознавание образов, обработка изображений, компьютерное зрение, сжатие данных, в общественных науках. Исчерпывающе описан в монографии [110].

Формальный анализ концептов (Formal concept analysis - FCA) [111] - это принципиальный способ получения иерархии концептов (понятий) или формальной онтологии из набора объектов и их свойств.

Каждый концепт в иерархии представляет объекты, обладающих некоторым набором свойств, а каждый подконцепт в иерархии представляет собой подмножество объектов (а также надмножество свойств) в концептах над ним. Термин был введен Рудольфом Вилле в 1981 [112] году и основан на математической теории решеток. Метод FCA хорошо описан в википедии [113].

Обзор литературы

В заключение обзора алгоритмов сопоставления приведем еще несколько статей.

Хороший обзор точного сопоставления строк дан в работе [43]. В статье дается обширная классификация методов сопоставления, описание более 50 алгоритмов и их сравнительный анализ, обсуждаются новые направления, возможные проблемы, текущие тенденции в области алгоритмов сопоставления строк с основным упором на алгоритмы точного сопоставления.

В монографии французских ученых Кристиан Чаррас (Christian Charras) и Тьерри Лекрок (Тьерри Лекрок) [114] дается подробное описание 35 алгоритмов точного сопоставления строк.

Хороший цикл статей представил итальянский ученый Симоне Фаро (Simone Faro) и его коллеги. В статье [115] приводится классификация алгоритмов точного сопоставления строк и 124 алгоритма с краткой аннотацией, которые были предложены за 1970 - 2016 годы, с указанием их библиографии. В работе [116] описывается эффективное и гибкое инструментальное средство SMART (String Matching Algorithms Research Tool), предназначенное для разработки, тестирования, сравнения и оценки алгоритмов сопоставления строк. Также приводится список 124 алгоритмов сопоставления, которые были разработаны за период 1970-2016 гг. и которые представлены в SMART. В [117] приводится обзор алгоритмов точного сопоставления строк, разработанных после 2000 г., и экспериментальные данные оценки некоторые из этих алгоритмов. Хороший обзор точных и неточных алгоритмов сопоставления строк приведен в [118].

Электронные библиотеки

Электронная библиотека – это распределенная документальная информационно-поисковая система, созданная на базе ПТБД, предоставляющая разнородные коллекции электронных документов в глобальной сети компьютеров в удобном для пользователей виде.


Джозеф Ликлайдер

Одной из самых ранних работ по электронным библиотекам считается монография Ликлайдера (Joseph Carl Robnett Licklider) [119], написанная в 1965 г. В ней предвиделось существование всемирной сети компьютеров, содержащей оцифрованные версии всей когда-либо написанной литературы, к которой предоставляется непосредственный доступ.


Виктор Михайлович Глушков

Следует также отметить монографию [120] выдающегося советского кибернетика Виктора Михайловича Глушкова, в которой он предвидел и представил научно-технические направления автоматизированной обработки и хранения информации в безбумажном (компьютерном) представлении.

До середины 70-х годов проводились исследования и разработки в области онлайновых информационных систем и сервисов, включая и ЭБ, детальный анализ которых был выполнен в монографии [121]. Среди них первым проектом, который можно отнести к ЭБ, считается Проект «Гутенберг» [122] - общественная некоммерческая инициатива, направленная на создание и распространение цифровой коллекции находящихся в общественном достоянии произведений. Проект был создан 4 июля 1971 года, когда студент Иллинойского университета Майкл Харт


Майкл Харт

(Michael Stern Hart) вручную перепечатал текст Декларации независимости США и отправил его другим пользователям своей сети, На 2021 год в коллекции Проекта было более 60 000 книг.

Термин "электронная библиотека" (digital library) стал широко использоваться начиная с1991 г. в связи с проведением серии семинаров, финансируемых Национальным научным фондом США (US National Science Foundation NSF). В этом же году была создана система “e-print archive”, которая затем была переименована в arXiv.

В середине 90-х годов была осознана важность создания ЭБ на государственных уровнях. В 1994 г. в США стартовал национальный проект "Инициатива электронных библиотек" (Digital Library Initiative - DLI), который стал серьезным толчком в развитии как ЭБ, так и национальных программ, проектов и организационных структур в других странах: Великобритании, Канаде, Австралии, Японии, ЕС и других. В 1996 г в специальном выпуске журнала EEE Computer [123], посвященном построению больших ЭБ, были подведены итоги выполнения проектов вплоть по 1996 г.

В связи с активным ростом количества ЭБ к концу прошлого столетия научная общественность пришла к пониманию необходимости построения концептуальных положений и моделей ЭБ в целом и для различных предметных областей в частности. В связи с этим в последующее десятилетие было предложено много таких моделей.

- В 1991–1997 годах Международной федерацией библиотечных ассоциаций и учреждений была разработана ER-модель "Функциональные требования к библиографическим записям" (Functional Requirements for Bibliographic Records, FRBR) [124] как обобщенное представление библиографического универсума, независимого от какого-либо варианта каталогизации или реализации.
- Концептуальная эталонная модель CIDOC CRM [125], Международного комитета по документации Международного совета музеев, предназначена для интеграции, посредничества и обмена информацией в области мирового культурного наследия и связанных областей.
- В 1991 г. был представлен Общеевропейский исследовательский информационный формат CERIF (Common European Research Information Format - CERIF) [126] в качестве всеобъемлющей информационной модели предметной области научных исследований. Модель CERIF является стандартом в ЕС.
- Уже много лет ведутся исследования в области анализа семантики связей между научными материалами. Системным обобщением этих результатов стало появление комплекса онтологий SPAR (Semantic Publishing and Referencing) [127], обеспечивающего достаточно детальную категоризацию отношений, которые могут возникать между научными материалами в электронном виде, и воплощающих их связей.
- Группа специалистов DELOS Европейского исследовательского консорциума по информатике и математике ERCIM в 2006–2007 гг., основываясь на анализе имеющихся библиотечных систем [128], где большое внимание было уделено функциональным возможностям современных ЭБ, сначала сформулировали манифест ЭБ [129], в котором были освещены краеугольные концептуальные принципы ЭБ, и затем разработали эталонную модели ЭБ DLRM (Digital Library Reference Model) [130], которая стала де факто стандартом в Европейском союзе.
- Гонсалвес (Goncalves) и др. предложили модель 5S [131, 132] в качестве теоретической основы ЭБ, которая способствовала появлению множества метамоделей для разных типов ЭБ.

Стандарты OAI-PMH/OAI-ORE

Появление тысяч ЭБ в Интернете привело к необходимости их интеграции. Было ряд предложений в этом направлении и наиболее эффективным и общепринятым стал протокол OAI PMH - протокол межбиблиотечного обмена метаданными, созданный в 2001 г. [133]. Его создание привело к появлению сайтов-интеграторов (харвесторов), объединяющих большое количество ЭБ и предоставляющих единую точку поиска и доступа к ним. Примерами таких

харвесторов являются следующие (данные приводятся по состоянию на январь 2023 г.):

- **BASE (Bielefeld Academic Search Engine)** - поисковая система Билефельдского университета содержит более 315 миллиона документов от около 11 тыс. провайдеров данных (http://www.base-search.net/);
- **OpenAire** - информационные ресурсы открытого доступа Европейского Союза. Содержит более 149 миллионов публикаций из 124 тыс. источников данных. (https://www.openaire.eu/);
- **CORE** - самая большая в мире коллекция (более 254 миллионов) научных статей открытого доступа - https://core.ac.uk/.

Протокол OAI-PMH предоставляет обмен метаданными, предполагая, что сами информационные ресурсы остаются у провайдеров данных. Восполняя этот пробел, OAI в 2008 г. выпустила новый стандарт - OAI-ORE (Object Reuse and Exchange), который предназначен для описания, повторного использования и обмена агрегациями информационных ресурсов веба. Эти агрегации, также называемые составными информационными объектами, могут объединять объекты многих типов, включая тексты, данные, изображения, видео- и аудиообъекты и другие. Целью OAI-ORE является предоставление приложениям богатого контента для поддержания создания, хранения, обмена, визуализации, повторного использования и сохранения агрегированных данных. Современные программные продукты, предназначенные для создания ЭБ, поддерживают этот стандарт.

Инструментальные средства ЭБ

Было разработано более 30 инструментальных средств создания ЭБ (см. http://roar.eprints.org/). Наиболее популярными являются DSpace, EPrints, OJS, Bepress, OPUS, Fedora, Greenstone.

Литература

1) Morris J.H. Jr, Pratt V. A linear pattern-matching algorithm (Technical report). 1970, University of California, Berkeley, Computation Center. TR-40.
2) Knuth D.E. The Dangers of Computer-Science Theory. Studies in Logic and the Foundations of Mathematics. 1973, Vol. 74, pp. 189–195.
3) Knuth D., Morris J.H., Pratt V. Fast pattern matching in strings. SIAM Journal on Computing. 1977, Vol, 6, No. 2. pp. 323–350.
4) Matiyasevich Yu. Real-time recognition of the inclusion relation. Journal of Soviet Mathematics. 1973, Vol. 1, No. 1, pp. 64–70
5) Boyer R.S., Moore J.S. A fast string searching algorithm. Communications of the ACM. 1977, vol. 20, No 10. pp. 762—772. — doi:10.1145/359842.359859.
6) Baeza-Yates R., Gonnet G.H. A new approach to text searching. Communications of the ACM, 1092, Vol. 35, No 10, pp 74–82
7) Boyer–Moore string-search algorithm. - https://en.wikipedia.org/wiki/Boyer%E2%80%93Moore_string-search_algorithm
8) Horspool R.N. Practical fast searching in strings, Software - Practice & Experience, 1980, 10(6) :501-506.
9) Zhu R.F., Takaoka T. On improving the average case of the Boyer-Moore string matching algorithm, Journal of Information Processingб 1987, Vol. 10, No. 3, pp. 173-177
10) Turbo-BM algorithm - http://www-igm.univ-mlv.fr/~lecroq/string/node15.html
11) Apostolico A., Giancarlo R. "The Boyer-Moore-Galil String Searching Strategies Revisited," (in English), SIAM Journal on Computing, vol. 15, No. 1, pp. 98-105, Feb 1986.
12) Smith P.D., "Experiments with a very fast substring search algorithm," Software-Practice and Experience, vol. 21, no. 10, pp. 1065-1074, 1991.
13) Raita T. Tuning the Boyer-Moore-Horspool string searching algorithm. Software-Practice and Experience, vol. 22, no. 10,pp. 879-884, 1992.
14) Crochemore M., Czumaj A., Gasieniec L., Jarominek S., Lecroq T., Plandowski W. Rytter W. "Speeding up two string-matching algorithms," Algorithmica 12(4-5):247-267, 1994.

15) Berry T., Ravindran, S. (2001) A Fast String Matching Algorithm and Experimental Results. Proceedings of the Prague Stringology Club Workshop ’99, Collaborative Report DC-99-05, Czech Technical University, Prague, 16-26.
16) Sunday D.M. " A very fast substring search algorithm," Communications of the ACM, Vol. 33, No 8, 1990 pp 132–142. - https://doi.org/10.1145/79173.79184.
17) Colussi L. Correctness and efficiency of pattern matching algorithms. Information and Computation, vol. 95, no. 2, pp. 225-251, 1991.
18) Xian-feng H., Yu-bao Y., Xia L. "Hybrid pattern-matching algorithm based on BM-KMP algorithm." (ICACTE) 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), 2010, pp. V5-310-V5-313.
19) Cao Z., Zhenzhen Y., Lihua L. "A fast string matching algorithm based on low-light characters in the pattern." In Advanced Computational Intelligence (ICACI), 2015 Seventh International Conference on, pp. 179-182. IEEE, 2015.
20) Hakak S., Kamsin A., Shivakumara P., Idna Idris M.Y., Gilkar G.A. "A new split based searching for exact pattern matching for natural texts." PloS ONE 13, no. 7 (2018): e0200912. Skid
21) Hakak S., Amirrudin K., Shivakumara P., Idna Idris M.Y. "Partition-Based Pattern Matching Approach for Efficient Retrieval Of Arabic Text." Malaysian Journal of Computer Science 31, no. 3 (2018): 200-209.
22) Franek F., Jennings C.G., Smyth W.F. A simple fast hybrid pattern-matching algorithm. J. Discrete Algorithms, 5(4):682–695, 2007.
23) Rabin M.O., Karp R.M. Efficient randomized pattern-matching algorithms. IBM Journal of Research and Development. 1987, vol. 31, No 2, pp. 249–260.
24) Rabin–Karp algorithm. - https://en.wikipedia.org/wiki/Rabin%E2%80%93Karp_algorithm
25) Wu S., Manber U. "A fast algorithm for multi-pattern searching," Department of Computer Science, University of Arizona, Tucson, AZ, Report TR-94-171994.
26) Kim S., Kim Y., "A fast multiple string pattern matching algorithm," in Proceedings of 17th AoM/IAoM Conference on Computer Science, 1999, pp. 44-49.
27) Simone F. "A very fast string matching algorithm based on condensed alphabets." In International Conference on Algorithmic Applications in Management, pp. 65-76. Springer, Cham, 2016.
28) Lecroq T. "Fast exact string matching algorithms,"Information Processing Letters, vol. 102, no. 6, pp. 229-235, Jun 15 2007.
29) Kalsi P., Peltola H., Tarhio J. "Comparison of exact string matching algorithms for biological sequences," in Proceedings of the Second International Conference on Bioinformatics Research and Development, BIRD, 2008. pp. 417-426.
30) Daciuk J., Mihov S, Watson B., Watson R. Incremental construction of minimal acyclic finite state automata. Computational Linguistics, 2000, 26(1), pp. 3-16.
31) Yang. W. Mealy machines are a better model of lexical analyzers. Computer Languages, Vol. 22, No 1, 1996, pp. 27-38
32) Blumer A., Blumer J., Ehrenfeucht A., Haussler D., McConnel R. Linear size finite automata for the set of all subwords of a word: an outline of results. Bull. European Assoc. Theoret. Comput. Sci., 21:12-20, 1983
33) Commentz-Walter B. A string matching algorithm fast on the average. Proceedings of the 6th Colloquium, on Automata, Languages and Programming, 1979, pp. 118–132
34) Allauzen C., Raffinot M. Simple optimal string matching algorithm, Journal of Algorithms, Vol. 36, No 1, 2000, pp. 102–116
35) Allauzen C., Crochemore M., Raffinot M. Factor oracle: A new structure for pattern matching. In 26th Seminar on Current Trends in Theory and Practice of Informatics (SOFSEM’99), Nov 1999, Milovy, Czech Republic, Czech Republic. pp.291-306.
36) Faro S., Lecroq T. Efficient variants of the Backward-Oracle-Matching algorithm. In Proceedings of the Prague Stringology Conference, Czech Republic, 2008, pp. 146-160: Czech Technical University.

37) Fan H., Yao N., Ma H. Fast variants of the backward-oraclemarching algorithm. In Fourth International Conference on Internet Computing for Science and Engineering, 2009, pp. 56-59.
38) He L., Fang B., Sui J. The wide window string matching algorithm. Theoretical Computer Science, vol. 332, no. 1-3, 2005, pp. 391-404
39) Liu C., Wang Y., Liu D., Li D. Two improved single pattern matching algorithms. In ICAT Workshops, Hangzhou, China,, 2006, pp. 419-422: IEEE Computer Society.
40) Hongbo F., Shupeng S., Jing Z., Li D. Suffix Type String Matching Algorithms Based on Multi-windows and Integer Comparison. In International Conference on Information and Communications Security, pp. 414-420. Springer, Cham, 2015.
41) Masaki Waga, Ichiro Hasuo, Kohei Suenaga. "Efficient online timed pattern matching by automata-based skipping." In International Conference on Formal Modeling and Analysis of Timed Systems, pp. 224-243. Springer, Cham, 2017.
42) Bitap algorithm. - https://en.wikipedia.org/wiki/Bitap_algorithm
43) Hakak S., Kamsin A., Shivakumara P., Gilkar G., Khan W.Z., Imran M. Exact String Matching Algorithms: Survey, Issues, and Future Research Directions. IEEE Access, 2019, Vol. 7, pp 69614-69637
44) Bálint Dömölki, An algorithm for syntactical analysis, Computational Linguistics 3, Hungarian Academy of Science pp. 29–46, 1964.
45) Bálint Dömölki, A universal compiler system based on production rules, BIT Numerical Mathematics, 8(4), pp 262–275, 1968. doi:10.1007/BF01933436
46) Shyamasundar R.K. Precedence parsing using Dömölki's algorithm, International Journal of Computer Mathematics, 6(2)pp 105–114, 1977.
47) Fredriksson K., Grabowski S. Practical and optimal string matching. In SPIRE'05: Proceedings of the 12th international conference on String Processing and Information Retrieval, 2005, pp. 376–387. - https://doi.org/10.1007/11575832_42
48) Salmela L., Tarhio J., Kytojoki J. Multi pattern string matching with q-grams. Journal of Experimental Algorithms, 2006, Vol. 11, pp. 1-19
49) Udi Manber, Sun Wu. "Fast text search allowing errors." Communications of the ACM, 35(10): pp. 83–91, October 1992, doi:10.1145/135239.135244.
50) Baeza-Yates V., Navarro G. A faster algorithm for approximate string matching. In Dan Hirchsberg and Gene Myers, editors, Combinatorial Pattern Matching (CPM'96), LNCS 1075, pages 1–23, Irvine, CA, June 1996.
51) Myers G. "A fast bit-vector algorithm for approximate string matching based on dynamic programming." Journal of the ACM 46 (3), May 1999, 395–415.
52) Navarro G., Raffinot M. A Bit-parallel Approach to Suffix Automata: Fast Extended String Matching. In Proc CPM'98, Lecture Notes in Computer Science 1448: 14-33, 1998.
53) Navarro G., Raffinot M. Fast and flexible string matching by combining bit-parallelism and suffix automata. ACM Journal. Experimental Algorithmics,2000, 5(4): 1-36
54) Peltola H., Tarhio J. Alternative Algorithms for Bit-Parallel String Matching. In Nascimento M.A., de Moura E.S., Oliveira A.L. (eds) String Processing and Information Retrieval, Spire Springer, LNCS 2857, pp. 80-93, 2003.
55) Branislav Durian, Jan Holub, Hannu Peltola and Jarma Tarhio,"Tuning BNDM with q-grams", In the proc. Of workshop on algorithm engineering and experiments, SIAM USA, pp. 29-37, 2009.
56) Miao C., Chang G., Wang X. Filtering Based Multiple String Matching Algorithm Combining q-Grams and BNDM. In ICGEC '10: Proceedings of the 2010 Fourth International Conference on Genetic and Evolutionary Computing, 2010, pp. 82–585.
57) Faro S., Lecroq T. Efficient variants of the backward-oracle-matching algorithm. International Journal of Foundations of Computer Science, vol. 20, no. 6, pp. 967–984, Dec. 2009,
2857. Springer, Berlin, Heidelberg.

58) Crochemore M., Czumaj A., Gasieniec L., Jarominek s., Lecroq T., Plandowski W., Rytter W. 1992, Deux méthodes pour accélérer l'algorithme de Boyer-Moore, in Théorie des Automates et Applications, Actes des 2e Journées Franco-Belges, D. Krob ed., Rouen, France, 1991, pp 45-63, PUR 176, Rouen, France.
59) Oguzhan Külekci, "Filter based fast matching of long patterns by using SIMD instructions," in Proceedings of the Prague Stringology Conference, Prague, Czech Republic, 2009. pp. 118--128
60) M. Oguzhan Külekci, A method to overcome computer word size limitation in bit-parallel pattern matching. In Proceedings of the 19th International Symposium on Algorithms and Computation, ISAAC, 2008. pp. 496--506
61) Gupta S., Rasool A. Bit Parallel String Matching Algorithms: A Survey. International Journal of Computer Applications, 2014, vol. 95, No 10, pp. 27-32
62) Crochemore M., Czumaj A., GaÌˇgsieniec L., Lecroq T., Plandowski W., Rytter W. Fast practical multi-pattern matching. Information Processing Letters, vol. 71, no. 3-4, pp. 107-113, Aug 27 1999.
63) Navarro G. Nrgrep: A fast and flexible pattern matching tool. Software—Practice & Experience, Vol. 31, No 13, 2001, pp. 1265–1312.
64) Franek F., Jennings C.G., Smyth, W.F. A simple fast hybrid pattern-matching algorithm. J. Discret. Algorithms, pp. 682-695, 2007.
65) Deusdado S., Carvalho P. GRASPm: an efficient algorithm for exact pattern-matching in genomic sequences. Int J Bioinform Res Appl, vol. 5, no. 4, pp. 385-401, 2009.
66) P. Shivendra Kumar, Tiwari H.K., Tripathi P. Hybrid approach to reduce time complexity of string matching algorithm using hashing with chaining. In Proceedings of International Conference on ICT for Sustainable Development, pp. 185-193. Springer, Singapore, 2016.
67) Hamming R.W. "Error detecting and error correcting codes". The Bell System Technical Journal. 1950, 29 (2): 147–160
68) Levenshtein V.I. Binary codes with correction of dropouts, insertions and substitutions of symbols (Rus). Reports of the Academy of Sciences of the USSR, 1965. 163.4: 845-848.
69) Levenshtein V.I. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 1966. 10 (8): 707–710.
70) Gusfield D. Algorithms on stings, trees, and sequences: Computer science and computational biology ACM SIGACT News, Vol. 28, No 4, Dec. 1997, pp. 41–60.
71) Damerau F.J. A technique for computer detection and correction of spelling errors. Communications of the ACM, 1964, vol. 7, No 3, pp 171–176
72) Winkler W.E. (1990). "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage" (PDF). Proceedings of the Section on Survey Research Methods. American Statistical Association: 354–359.
73) Jaro M.A. Advances in record linkage methodology as applied to the 1985 census of Tampa Florida Journal of the American Statistical Association. 1989, Vol. 84, No. 406, pp. 414-420
74) Longest common subsequence problem. - https://en.wikipedia.org/wiki/Longest_common_subsequence_problem
75) Hall P., Dowling G. Approximate string matching. ACM Computing Surveys, 12(4) :381-402, 1980.
76) Sankoff D., Kruskal J., editors. Time Warps. String Edits, arid Macro molecules: The Theory arid Practice of Sequence Comparison. Addison-Wesley, 1983.
77) Apostolico A., Galil Z. Combinatorial Algorithms on Words. NATO ISI Series. Springer-Verlag, 1985.
78) Galil Z., Giancarlo R. Data structures and algorithms for approximate string matching. Journal of Complexity, Vol. 4, No 1, 1988, pp. 33-72
79) Jokinen P, Tarhio J, Ukkonen E. A comparison of approximate string matching algorithms. Software Practice arid Experience, 26(12): 1439-1458,1996.

80) Navarro G. A guided tour to approximate string matching. ACM Computing Surveys, Vol. 33, No , 1, 2001, pp 31–88
81) Syeda Shabnam Hasan, F. Ahmed, Rosina Surovi Khan. Approximate String Matching Algorithms: A Brief Survey and Comparison. International Journal of Computer Applications, 2015, Vol. 120, No. 8, pp. 26-31
82) Melichar B., Holub J., Polcar T. Text search-ing algorithms Volume I: Forward string matching. Czech Technical University in Prague, 224 p. - /http://www.stringology.org/athens/TextSearchingAlgorithms/tsa-lectures-1.pdf
83) Salton G. Associative document retrieval techniques using bibliographic information. J ACM. 1963, Vol. 10, No.4, pp. 440–457.
84) Salton G., Wong A., Yang C.S. A Vector Space Model for Automatic Indexing. Communications of the ACM, 1975, Vol. 18, No 11, pp. 613— 620.
85) Salton G., Lesk M.E. The SMART automatic document retrieval systems—an illustration. Communications of the ACM. 1965, Vol. 8, No 6, pp. 391–398
86) SMART Information Retrieval System. - https://en.wikipedia.org/wiki/SMART_Information_Retrieval_System
87) Wong S.K.M., Ziarko W., Wong P.C.N. Generalized vector spaces model in information retrieval. Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '85, SIGIR ACM, 1985, pp. 18–25,
88) Maron M.E., Kuhns J.L. On relevance, probabilistic indexing and information retrieval. Journal of the ACM, Vol. 7, No. 3, pp. 216–244, 1960.
89) Robertson S.E., Jones K.S. Relevance weighting of search terms. Journal of the American Society for Information Science. Vol.27, No. 3. pp. 129-146. 1976.
90) Robertson S.E., van Rijsbergen C.J., Porter M.F. Probabilistic models of indexing and searching. In Information Retrieval Research (Proceedings of Research and Development in Information Retrieval, Cambridge, 1980), (R. N. Oddy, S.E. Robertson, C. J. van Rijsbergen, and P. W. Williams, eds.), pp. 35–56, London: Butterworths, 1981.
91) Robertson M., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval, Vol. 3, No. 4, pp. 333–389, 2009.
92) Robertson S.E., van Rijsbergen C.J., Porter M.F. Probabilistic models of indexing and searching. In Information Retrieval Research (Proceedings of Research and Development in Information Retrieval, Cambridge, 1980), (R. N. Oddy, S. E. Robertson, C. J. van Rijsbergen, and P. W. Williams, eds.), pp. 35–56, London: Butterworths, 1981.
93) Lavrenko V., Croft W.B. Relevance based language models. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 120–127, ACM, 2001.
94) Amati G., Van Rijsbergen C.J. Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Systems, 2002, Vol. 20, No. 4, pp 357–389
95) Crestani F., Lalmas M., van Rijsbergen C.J., Campbell I. "Is this document relevant? ... probably": A survey of probabilistic models in information retrieval. ACM Computing Surveys, Vol. 30, No. 4, pp. 528–552, 1998.
96) Maron M.E. Mechanized documentation: the logic behind a probabilistic interpretation. Statistical Association Methods For Mechanized Documentation. National Bureau of Standards Miscellaneous Publications 269. (M. E. Stevens, V. E. Guiliano and L. B. Heilprin. eds). pp 9-13. 1964.
97) van Rijsbergen C.J. A non-classical logic for information retrieval. The Computer Journal. Vol. 29, No. 6, pp. 48 -485. 1986.
98) 1160) Borko H., Bernick M. Automatic Document Classification. Journal of the ACM, 1963, Vol. 10, No 2, pp 151–162
99) Baker F.B. Information Retrieval Based upon Latent Class Analysis. Journal of the ACM , 1963, Vol. 9, No 4, 512-521.
100) Dumais S.T., Furnas G.W., Landauer T.K., Deerwester S. (1988), "Using latent

semantic analysis to improve information retrieval." In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, pp. 281-285.
101) Deerwester S., Dumais S.T., Landauer T.K., Furnas G.W., Harshman, R.A. "Indexing by latent semantic analysis." Journal of the Society for Information Science, 1990, Vol. 41, No. 6, pp. 391-407.
102) Foltz, P. W. (1990) "Using Latent Semantic Indexing for Information Filtering". In R. B. Allen (Ed.) Proceedings of the Conference on Office Information Systems, Cambridge, MA, 40-47.
103) Landauer T.K., Foltz P.W., Laham, D. (1998). An Introduction to Latent Semantic Analysis.Discourse Processes, 25, pp. 259-284.
104) Berry M.W., Dumais S.T., O'Brien G.W. Using Linear Algebra for Intelligent Information Retrieval. SIAM Review, 1995, 37:573-595.
105) Landauer T.K. Applications of Latent Semantic Analysis. 24th Annual Meeting of the Cognitive Science Society. 2002.
106) Hofmann T. Probabilistic Latent Semantic Indexing. SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, pp. 50–57
107) Pritchard, J. K.; Stephens, M.; Donnelly, P. (June 2000). "Inference of population structure using multilocus genotype data". Genetics. 155 (2): pp. 945–959.
108) Blei D.M., Ng A.Y., Jordan M.I. Latent Dirichlet allocation. Journal of Machine Learning Research. 2003, Vol. 3, No 4-5), pp. 993—1022.
109) Pearson K., On lines and planes of closest fit to systems of points in space, Philosophical Magazine, (1901) 2, 559—572
110) 1172)Jolliffe I.T. Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY, 2002, XXIX, 487 p.
111) Ganter B., Wille R. Formal Concept Analysis: Mathematical Foundations. Berlin, Springer, 1999.
112) Wille R. (1982). "Restructuring lattice theory: An approach based on hierarchies of concepts". In Rival, Ivan (ed.). Ordered Sets. Proceedings of the NATO Advanced Study Institute held at Banff, Canada, August 28 to September 12, 1981. Nato Science Series C. Vol. 83. Springer. pp. 445–470.
113) Formal concept analysis. - https://en.wikipedia.org/wiki/Formal_concept_analysis
114) Christian Charras, Thierry Lecroq. Handbook of exact string matching algorithms. College Publications (February 27, 2004), 256 p. - http://www-igm.univ-mlv.fr/~lecroq/string/string.pdf
115) 1Faro S. Exact Online String Matching Bibliography. 2016, 23 p. - https://arxiv.org/abs/1605.05067
116) Faro S., Lecroq T., Borzi, Di Mauro S., Maggio A.. The String Matching Algorithms Research Tool. Stringology 2016: 99-111
117) Faro S., Lecroq T. The exact online string matching problem: A review of the most recent results. ACM Comput. Survey, Article 13, 42 pages, February 2013.
118) Koloud Al-Khamaiseh, Shadi AL Shagarin. A Survey of String Matching Algorithms. Int. Journal of Engineering Research and Applications, 2014, vol. 4, No 7, pp.144-156
119) Licklider J.C.R. Libraries of the future. Cambridge, MA: The MIT Press; 1965.
120) Glushkov V.M. Fundamentals of paperless informatics (Rus). - M.: Nauka, Chief editor of physic.-math. lit., 1987. - 552 p.
121) Charles P. Bourne, Trudi Bellardo Hahn. A History of Online Information Services, 1963-1976. MIT Press, 2003, 496 p
122) Project Gutenberg. - https://en.wikipedia.org/wiki/Project_Gutenberg
123) Schatz B., Chen H. Building large-scale digital libraries. IEEE Computer. 1996, Vol. 29, No. 5, pp. 22–26
124) Functional Requirements for Bibliographic Records, Final Report / IFLA Study Group on the Functional Requirements for Bibliographic Records. – München: K.G. Saur, 1998.
125) Crofts N., Doerr M., Gill T., Stead S.,, Stiff M. (editors), Definition of the CIDOC Conceptual Reference Model, January 2008. Version 4.2.4.

126) CERIF in Brief. - https://www.eurocris.org/eurocris_archive/cerifsupport.org/cerif-in-brief/index.html
127) Shotton D. Introduction the Semantic Publishing and Referencing (SPAR) Ontologies. October 14, 2010. http://opencitations.wordpress.com/2010/10/14/introducing-the-semantic-publishing-and-referencing-spar-ontologies/
128) Candela L., Castelli D., Fuhr N., Ioannidis Y., Klas C.-P., Pagano P., Ross S., Saidis C., Schek H.-J., Schuldt H., Springmann M. Current Digital Library Systems: User Requirements vs Provided Functionality. IST-2002- 2.3.1.12. Technology-enhanced Learning and Access to Cultural Heritage. March 2006.
129) Candela L., Castelli D., Ioannidis Y., Koutrika G., Pagano P., Ross S., Schek H.J., Schuldt H. Setting the foundations of digital libraries: the DELOS manifesto. D-Lib Mag. 2007;13(3/4)
130) Candela L., Castelli D., Dobreva M., Ferro N., Ioannidis Y., Katifori H., Koutrika G., Meghini C., Pagano P., Ross S., Agosti M., Schuldt H., Soergel D. The DELOS Digital Library Reference Model Foundations for Digital Libraries. IST-2002-2.3.1.12. Technology-enhanced Learning and Access to Cultural Heritage. Version 0.98, December 2007.
131) Goncalves M.A., Fox E.A.., Watson L.T. and Kipp N.A. Streams, structures, spaces, scenarios, societies (5S): A formal model for digital libraries. ACM Transactions on Information Systems. 2004, Vol, 22, No. 2, pp. 270–312.
132) Isah A., Serema B. C., Mutshewa A., Kenosi L. Digital Libraries: Analysis of Delos Reference Model and 5S Theory. Journal of Information Science Theory and Practice. 2013, Vol. 1, No. 4, pp. 38–47
133) Open Archives Initiative Protocol for Metadata Harvesting. - https://en.wikipedia.org/wiki/Open_Archives_Initiative_Protocol_for_Metadata_Harvesting

Базы данных изображений

База данных изображений (БДИ) - *это база данных, которая предоставляет эффективные и развитые средства и технологии для поддержания процессов моделирования, хранения, индексирования, поиска и манипулирования изображениями и их метаданными* [1].

Текстовый поиск изображений

Деятельность по использованию изображений в базах данных берет свое начало в середине 70х годов прошлого столетия в связи с появлением развитых промышленных СУБД. Основные работы этого периода были сосредоточены на текстовом подходе к поиску изображений (text based image retrieval - TBIR). Суть его заключалась в аннотировании и поиске изображений на основе текстовой информации. Изображения описывались набором ключевых слов или текстовыми дескрипторами и с помощью заложенных в СУБД средств поиска по тексту отыскивались требуемые изображения. В 1979 г. была проведена международная конференция по использованию технологий баз данных в графических приложениях, на которой были подведены итоги в области БДИ. В статьях [2, 3] даются основательные обзоры работ по текстовому описанию и поиску изображений в БД по состоянию на 1984 и 1992 г.

Со временем в связи со сложностью и многообразием описательных элементов изображений исследователи пришли к осознанию необходимости создания универсальных управляемых словарей, классификационных схем и других подходов по терминологическому упорядочению описания изображений. Впервые этот вопрос подняла Сара Шатфорд (Sara Shatford) в работе [4], которая косвенно привела к появлению со временем различных тезаурусов, имеющих отношение к изображениям.


Сара Шатфорд

Контентный поиск изображений

В начале 90-х годов появились работы по поиску изображение по их содержимому.

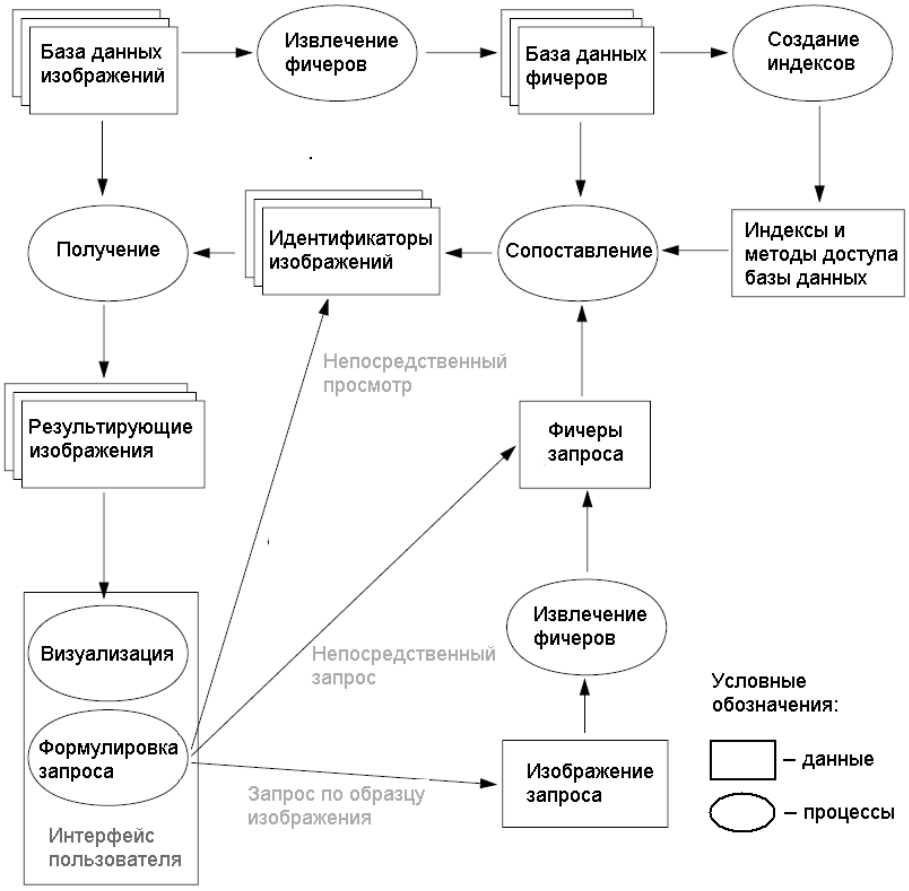

Тошиказу Като

Это направление получило название контентного поиска изображений (content-based image retrieval - CBIR). Считается [5], что термин CBIR впервые был использован японским инженером Тошиказу Като (Toshikazu Kato) в 1992 г. для описания процедуры автоматического поиска изображений в базе данных на основании задаваемых цветов и фигур [6]. Со временем этот термин стал использоваться для описания технологий извлечения, индексирования, сравнения и поиска изображений с использованием их низкоуровневых характеристик - фичеров (features), таких, например, как цвет, текстура, фигура, место расположения и др., которые можно извлекать из изображений автоматически.

Отметим, что термин фичер (feature) широко используется в машинном обучении и распознавании образов и означает индивидуальное измеряемое свойство или характеристику явления/объекта, а фичерный вектор - это n-мерный вектор числовых фичеров, которые представляют некоторый объект. Именно в этом контексте мы используем этот термин в БДИ.

С тех пор было исследовано и предложено множество методов и технологий в этом направлении и реализовано множество исследовательских и коммерческих систем поиска изображений. При этом используются методы, средства и алгоритмы из таких научных областей, как статистика, распознавание образов, обработка сигналов и компьютерное зрение [7]. Этой тематике было посвящено множество специальных выпусков ведущих журналов, а также опубликовано много монографий [8–13].

В работе [14] на основании анализа большого количества систем CBIR была предложена общая концептуальная схема функционирования CBIR, которая приведена на рис. ниже.



Опишем полученные результаты в области CBIR, опираясь на эту схему

Обработка изображений – сегментация

Сегментация изображения является важным шагом в решении задач распознавания, сжатия, визуализации и поиска изображений [15].

Начальным этапом анализа (распознавания содержимого) изображения является его сегментация. Суть сегментации - разбиение изображения на множество областей/объектов с целью упрощения и/или изменения представления изображения, чтобы его было проще и легче анализировать [16]. Обычно используется для определения объектов и границ (линий, кривых и т.д.) в изображении.

Исследования в области сегментации изображений были инициированы в начале 90-х г. практически одновременно с появлением направления CBIR.

Существует острая необходимость определить, какими фичерами обладает изображение. Обычно это делается разбиением изображения на множество однородных (относительно некоторого свойства) прямоугольных областей, каждая из которых называется сегментом, а сам процесс - сегментация. Существует множество методов сегментации, обзор и анализ которых приведен в [16-19]. Самым "свежим" на момент написания статьи является обзор [20], в котором анализируется 11 методов 4-х категорий и их применение в 10 предметных областях.

Извлечение фичеров

Из изображений извлекаются низкоуровневые характеристики (фичеры) и запоминаются в БД в виде фичерных векторов. Извлечение фичеров составляют основу CBIR. К базовым фичерам изображений относятся цвет, текстура, фигура, место расположения. Кратко обсудим существующие методы извлечения и представления этих фичеров.

Цвет. Цвет является самым используемым фичером при поиске изображений. Цвета определяются согласно выбранному цветовому пространству. Имеется множество цветовых пространств и используются они в различных приложениях. Описание цветовых пространств можно найти в [21]. Стандарт MPEG-7 [22] в состав дескрипторов цвета включил: основной цвет, прямой цвет, цветовое пространство, квантование цвета, цветовая раскладка, масштабируемый цвет, структура цвета. Наиболее используемым способом представления является цветовая гистограмма [23]. Обычно это совместное распределение интенсивности трех цветовых каналов (RGB). Учитывая, что большинство гистограмм являются разряженными и поэтому чувствительными к шуму, была предложена кумулятивная гистограмма [24], которая продемонстрировала ее преимущества по сравнению с обычной цветной гистограммой. В работе [25] также предложен метод квантования цветового пространства, что повышает эффективность поиска. Были также предложены другие способы представления цвета для целей поиска, например, ковариационная матрица цвета (color covariance matrix), моменты цветов (color moments) [24], наборы цветов (color sets) [26, 27], векторы когерентности цвета (color coherence vectors) [28].

Текстура. Текстура — изображение, воспроизводящее визуальные свойства каких-либо поверхностей или объектов. Она содержит важную информацию о структурном упорядочении поверхностей и их взаимосвязь с окружающей средой. Она оказалась весьма полезной в свое время в решении задач распознавания образов и компьютерного зрения, а также плодотворно используется в CBIR.

В начале 70-х г. был предложен метод представления текстурных фичеров в виде матрицы совпадения (co-occurrence matrix) [29]. Он в дальнейшем был развит в работе [30]. В работе [31] был предложен вариант представления текстурных фичеров на основании психологических исследований визуального восприятия человеком изображений. Этот метод оказался весьма привлекательным в CBIR, так как способствовал созданию боле удобного для пользователя интерфейса. Усовершенствованный вариант был применен в системах QBIC [32] и MARS [33].

В начале 90-х годов многие ученые стали использовать вейвлет-преобразования в изучении способов представления тексту-

ры. В работе [34] предложено использовать извлеченную из поддиапазонов вейвлета статистику в качестве представления текстуры. С помощью этого подхода была достигнута 90% точность на 112 текстурах Бродаца. В работе [35] для дальнейшего улучшения классификационной точности была использована древовидная структура вейвлет-преобразования. Для повышения производительности были предложены подходы, сочетающие вейвлет-преобразования с другими методами, например, расширение Карунена–Лоэва (Karhunen–Loève expansion) и карты Кохонена (Kohonen maps) [36], матрица совпадений [37, 38]. Были также предложены и исследованы представления текстуры марковскими случайными полями [39], многоканальной фильтрацией [40], фильтрацией Габора [41] и фрактальное [42].

На протяжении многих лет публиковались обзорные и сравнительные статьи [43–45]. Одной из последних статей этого типа является [46]. В ней анализируются 22 метода/модели представления текстур, принадлежащие 4 классам и 18 подклассам.

Фигура. Существует два способа представления фигур - в виде контура и в виде плоскости [47]. Наиболее используемыми методами этих двух представлений являются дескрипторы Фурье и моментные инварианты (moment invariants).

Основная идея использования дескрипторов Фурье заключается в представлении фичера фигуры в виде контурного преобразования Фурье. Первые работы в этом направлении относятся к началу 70-х годов [48, 49]. В работе [48] предложен модифицированный дескриптор Фурье для устранении помех при оцифровке изображений. Основная идея использования моментных инвариантов заключается в представлении фичера фигуры с помощью поверхностных моментов, которые инвариантны к преобразованиям. В 1962 г. в работе [50] было выделено 7 таких моментов. В [51] предложен быстрый метод вычисления моментов в бинарных изображениях. Для представления фигур также были предложены: метод конечных элементов (finite element method - FEM) [52], функция поворота (turning function) [53], гистограмма направления граней (Edge Directions Histogram) [54], вейвлет-дескриптор [55]. Были также опубликованы обзорные статьи [56, 57]. Проводились исследования по представлению объемных фигур [58–61].

Цветовое распределение (color layout). В середине 90-х годов в связи с ростом размеров изображений пришли к осознанию того факта, что хорошим решением для представления и поиска изображений является использование не точного попиксельного представления изображений, а так называемого цветового распределения (как с точки зрения фичеров цвета так и пространственных взаимосвязей). Идея заключалась в разделении изображения на подблоки и извлечении цветовых фичеров из каждого из них [61, 62]. Концепция цветового распределения была исследована и усовершенствована в последующих работах [28, 63–67], а также применена к текстуре и другим фичерам изображений.

Пространственное расположение. Также является важным фичером изображения и применяется обычно к пространственным объектам. Обычно оно определяется в виде понятий *вверху*, *внизу*, *слева*, *справа* т.п. согласно расположению конкретного объекта [68]. В [41] для представления пространственной информации используется центр тяжести объекта и минимальный ограничивающий его прямоугольник. В других работах предлагается использовать только центр объекта [69].

Взаимно расположение объектов является более важным, чем их абсолютные координаты. Для указания взаимного расположения наиболее часто используется строки символов (слева/справа, сверху/снизу) [70]. В работе [71] для поддержки семантического поиска изображений представлен алгоритм моделирования пространственного контента. Оригинальный метод пространственного взаимного расположения объектов предложен в [72].

Многомерное индексирование

Чтобы CBIR был действительно масштабируемым для коллекций изображений большого размера, необходимы эффективные методы многомерного индексирования. В этом отношении существенную роль играют следующие два фактора:

– Большая размерность. Размерность фичерных векторов обычно порядка 10^2.
– Неевклидова мера подобия. Поскольку евклидова мера не может эффективно имитировать человеческое восприятие определенного визуального контента, следует использовать другие меры подобия.

Поэтому для решения этих проблем сначала следует уменьшить размерность векторов, а затем использовать соответствующие методы многомерного индексирования, которые способны поддерживать неевклидовы меры подобия.

Что касается уменьшения размерности, то было предложено два основных подхода: преобразование Карунена – Лоэва (Karhunen–Loeve transform - KLT) и кластеризация по столбцам (column-wise clustering). Например, метод KLT исследуется и используется в работах [77-80]. Метод кластеризации по столбцам был предложен в 1983 г. [81].

Существующие популярные методы многомерного индексирования включают блочный (bucketing) алгоритм, k-d-деревья, приоритетные k-d-деревья, квадродеревья, K-D-B-, hB-, R-, R+-, R*-деревья. История методов многомерного индексирования восходит к середине 1970-х годов, когда впервые были разработаны клеточные (cell) методы, квадродеревья (quad-tree) и k-d-деревья (k-d-tree). Однако их производительность была далеко от удовлетворительной. Учитывая потребности индексации пространственных данных, то есть многомерной информации, возникающие в системах ГИС и САПР, в 1984 г. в работе [82] Антонин Гуттман (Antonin Guttman) впервые предложил структуру индексации R-дерева. На основе этой работы было предложено множество других вариантов R-дерева: R+-дерево [83], усовершенствованное R-дерево [84], R*-дерево [85]. Однако было показано, что они перестают масштабироваться при размерах векторов выше 20. В статьях [78, 86] представлены обзоры сравнительного анализа различных методов индексирования по состоянию на 1996 г.

Как мы уже отметили, вторым аспектом в проблеме поиска изображений является неевклидова мера подобия. Были предложены два подхода в этом направлении: кластеризация и нейронные сети. В работе [87] был предложен метод инкрементной кластеризации для динамического поиска информации. Этот метод предоставлял возможность обрабатывать многомерные данные и использовать неевклидовы меры подобия. В дальнейшем он был развит в работе [88].

В работе [89] было предложено использовать нейронные сети карт самоорганизации (self-organization map - SOM) как инструмент построения индексов древовидной структуры при поиске изображений.

В работах [73, 74, 75] дается обширный обзор методов индексирования и доступа. Более детальная информация по этому поводу приведена в разделе "Пространственные базы данных".

Сопоставление изображений

Исследования по подобию изображений берут свое начало во второй половине 70-х годов, когда Амос Тверски (Amos Tversky) опубликовал статью [90], в которой он предложил парадигму "контрастных фичеров" (Feature Contrast), характеризующую подобие двух объектов в терминах контрастности соответствующих фичеров.


Амос Тверски

Были предложены и изучены два основных подхода по поиску изображений по принципу подобия: метрический подход и трансформационный подход

Метрический подход предполагает, что определяется понятие расстояния меду двумя изображениями и чем ближе друг к другу они, тем более похожи они. Этот подход широко используется в БД.

Трансформационный подход является более общим по сравнению с метрическим. Он предполагает, что уровень подобия зависит от количества (и стоимости) выполнения операций преобразования одного изображения в другое. Такие операции включают поворот, масштабирование, растяжение/сжатие, перемещение, вырезание, добавление и многие другие, которым могут

приписываться весовые/стоимостные характеристики.

В работе [91] приводится детальный обзор проблемы подобия по состоянию на1996 г.

Метрический подход

Для сравнения изображения запроса с изображениями базы данных следует определить метрики их подобия. Это делается введением понятия расстояния между изображениями, чем меньше расстояние между ними, тем более близки они друг к другу.

В монографии [92], объемом более 750 страниц, приводится исчерпывающий перечень метрик расстояний, которые используются в различных науках. В частности, там описано 150 метрик для измерения расстояний в компьютерных науках, из них 35 метрик, применяемых в изображениях.

Приведем ряд из них, которые наиболее часто используются в научных статьях и системах для определения подобия изображений и текстов.

Евклидово расстояние — наиболее часто используемая метрика для определения степени близости изображений, определяется как расстояние между двумя точками евклидова пространства, вычисляемое по теореме Пифагора.

Расстояние городских кварталов (cityblock distance) — метрика, введённая Германом Минковским. Согласно этой метрике, расстояние между двумя точками равно сумме модулей разностей их координат, то есть это расстояние между двумя точками с нанесенной прямоугольной сеткой, когда перемещаться можно только по сторонам сетки. Также называется метрикой Манхэттена, прямоугольной метрикой, метрикой сетки.

Расстояние шахматной доски (chessboard distance) Предполагает, что имеется сетка и можно перемещаться по сторонам сетки и по диагонали, то есть как король в шахматах. Также называется расстоянием хода короля и расстоянием Чебышева.

Расстояние Махаланобиса (Mahalanobis distance) — мера расстояния между векторами случайных величин, обобщающая понятие евклидова расстояния. Предложено индийским статистиком Махаланобисом в 1936 году [93]. С помощью расстояния Махаланобиса можно определять сходство неизвестной и известной выборки. Оно отличается от расстояния Евклида тем, что учитывает корреляции между переменными и инвариантно к масштабу.

Косинусное расстояние (cosine distance), также известное как косинусное сходство, является мерой величины разности между двумя фичерными точками, используя косинусное значение угла между двумя векторами в векторном пространстве.

Расстояние Минковского (Minkowski distance) — параметрическая метрика на евклидовом пространстве, которую можно рассматривать как обобщение евклидова расстояния и расстояния городских кварталов. Названа в честь немецкого математика Германа Минковского, впервые систематически изучившего данное семейство функций расстояния.

Корреляционное расстояние (correlation distance) — базируется на коэффициенте корреляции Пирсона, который был предложен еще в 1895 г. [94]. Популярный способ измерения расстояния в виде корреляции двух фичерных точек с конечными дисперсиями.

Индекс Жаккара (Jaccard index). Предложен французским ученым Полом Жаккаром в 1901 г. [95] как мера подобия двух множеств, выражающаяся делением пересечений двух множеств на их объединение.

Краткое но содержательное изложение всех перечисленных выше и ряда других расстояний приведено в [96].

Системы CBIR

В типичной системе CBIR (см рис выше) исходные изображения хранятся в БД с использованием соответствующих методов. Из этих изображений извлекаются визуальные фичеры, они представляются многомерными фичерными векторами, запоминаются в БД фичеров и индексируются для более быстрого поиска с использованием соответствующих методов доступа. Для нахождения соответствующего изображения пользователь указывает его в запросе. Система извлекает из этого изображения его фичеры и представляет их в виде фичерных

векторов. Затем с помощью разработанных процедур подобия производится сопоставление фичерных векторов изображения запроса и изображений базы данных с использованием индексов и методов доступа. Удовлетворяющие сопоставлению изображения базы данных передаются пользователю.

Первая коммерческая система CBIR была разработана в 1995 г. в IBM и называлась QBIC (Query By Image Content) [32]. С тех пор было разработано множество коммерческих и экспериментальных систем CBIR, например, MARS [33], Photobook [52], Virage [97], VisualSEEK [98], Netra [99], SIMPLIcity [100]. В отчете [14] дается анализ 58 систем и приложений CBIR, которые были разработаны к 2002 г., с указанием их первоисточников. Кроме того, приведена итоговая таблица с указанием используемых в них фичерах. Статья [101] содержит обзор более 200 публикаций по CBIR по состоянию на 2000 год. Прекрасные обзоры по методам и принципам поиска информации в CBIR-системах опубликованы в статьях [102, 103, 104].

Семантический поиск изображений

Недостаток CBIR заключается в отсутствии семантики. С помощью низкоуровневых фичеров невозможно описывать высокоуровневые понятия, воспринимаемые человеком. То есть существовал "семантический разрыв" между низкоуровневыми фичерами изображения и используемыми человеком высокоуровневыми понятиями предметной области. В статье [101] семантический разрыв определяется как "*несовпадение информации, которую можно извлечь из визуальных данных, и интерпретации этих же данных пользователем в конкретной ситуации*". В связи с этим в начале 2000-х годов появились исследования, а затем и разработки по семантическому поиску изображений (Semantic-Based Image Retrieval - SBIR).

В отчете [105] выделяются три уровня языков.

Уровень 1. Поиск по низкоуровневым фичерам, таким как цвет, текстура, фигура, пространственное расположение. Типичным является запрос "найти изображения, похожие на заданное".

Уровень 2. Поиск объектов заданного типа, идентифицируемых указанными фичерами, с возможным применением логических правил вывода. Пример: "найти изображения с автомобилями".

Уровень 3. Поиск по абстрактным характеристикам, включающим высокоуровневые рассуждения о целях, способах, методах представления изображенных предметов или сцен. Может включать поиск названных событий или изображения с эмоциональным или религиозным значением, и т. п. Например «найти изображения ликующей толпы».

Уровни 2 и 3 относятся к классу SBIR, а различия между уровнями 1 и 2 характеризуют "семантический разрыв".

В настоящее время предложены следующие 5 методов уменьшения семантического разрыва:

- *использование онтологий* для определения высокоуровневых концептов/понятий;
- *методы машинного обучения* для установления взаимосвязей между низкоуровневыми фичерами и высокоуровневыми концептами запроса;
- *кластеризация данных*;
- *обратная связь по релевантности* (relevance feedback - RF) в поисковый цикл для непрерывного изучения намерений пользователей;
- *семантические шаблоны* (semantic template ST) для поддержки высокоуровневого поиска изображений;

Кратко опишем эти методы, подробнее можно познакомиться с ними в [106–109].

Онтологии объектов

Онтологический подход предполагает построение таксономической онтологической структуры понятий относительно фичеров изображений. В таких системах сначала определяются различные интервалы для низкоуровневых фичеров. Эти интервалы определяют дескрипторы изображений промежуточного уровня, например, "светло-зеленый, зеленый, темно-зеленый". Они также могут обобщаться с построением в конечном варианте онтологии понятий и ко-

торые могут использоваться для определения высокоуровневых понятий запросов, например, "облако" может быть определено как "произвольная выпуклая" (фигура), "светло-голубого" (цвета), "однотонная" (текстура), "вверху" (пространственное расположение). На онтологиях можно определять специальные правила вывода. Простейшими из них являются таксономические правила. Примером такой онтологической системы является [69].

Разбиение фичеров на интервалы требует использование единых правил именования вершин создаваемой онтологии. Так, например, в [110] предложена система именования цветов. В работе [111] предлагается система из 12 основных цветов, пяти уровней яркости и трех уровней насыщенности, всего 180 вариантов цвета. Для поиска картин определяются понятия: теплый цвет, холодный цвет, контрастность (светло-темный, тепло-холодный, дополняющий), например, найти картины, написанные в светло-темных тонах. По аналогии с цветом возникает необходимость создания системы именования текстуры, которая бы стандартизовала описание и представление текстур [112]. Как оказалось именование текстур довольно сложная задача и по настоящее время нет единой системы именования.

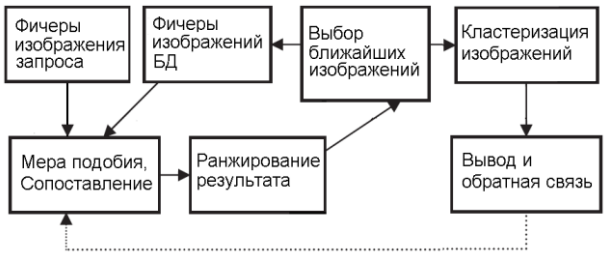
Машинное обучение

Во многих случаях для получения высокоуровневых семантических фичеров следует использовать такие формальные средства, как обучение с учителем и без него.

В SBIR используются такие методы обучения с учителем, как метод опорных векторов (support vector machine - SVM), байесовский классификатор, искусственная нейронная сеть, бутстрэппинг, дерево решений. Например, в [113] SVM используется для аннотирования изображений, в [114] с помощью бинарного байесовского классификатора высокоуровневые понятия выводятся из низкоуровневых фичеров изображений. В [115] искусственная нейронная сеть также применяется для установления взаимосвязи между фичерами изображений и предварительно выбранными высокоуровневыми понятиями. В [116] для аннотирования большой коллекции изображений был применен метод бутстрэппинга. В [117] сначала строится дерево решений на основе набора изображений, релевантных запросу, и затем оно используется в качестве модели для классификации изображений базы данных согласно двух классов: релевантные и нерелевантные.

Кластеризация изображений

Это типичный метод обучения без учителя для целей поиска изображений. Цель кластеризации - разбиение многомерного неструктурированного множества данных на ряд подмножеств данных со схожими характеристиками [118, 119]. Он предполагает группировку наборов данных изображений таким образом, чтобы максимизировать сходство изображений внутри кластеров и минимизировать сходство между разными кластерами. Для кластеризации изображений наиболее популярными являются методы k-средних (k-means) [106] и его варианты, Ncut [120], нечёткой кластеризации c-средних (fuzzy c-means) [121]. Показательным примером использования кластеризации является автоматическое уточнение меры подобия для сопоставление изображений по схеме, приведенной на рис. ниже и предложенной в [122].



Она функционирует так. На основании изображения запроса и выбранной меры подобия отыскиваются похожие изображения, они ранжируются и выбираются ближайшие из них. Затем, на основе гипотезы, что изображения с одинаковой семантикой имеют тенденцию группироваться, производится кластеризация для распределения результирующих изображений по различным семантическим классам. Затем система выводит кластеры изображений и уточняет меру подобия согласно обратной связи пользователя.

Для поиска изображений также применяются методы распознавания объектов.

Обратная связь по релевантности

В отличие от предыдущих подходов, обратная связь по релевантности (relevance feedback - RF) RF предполагает онлайновую обработку, который обеспечивает оперативную реакцию на намерения пользователя. Традиционно RF использовался в текстовых информационно-поисковых системах, а с середины 90-х г. стал применяться в CBIR для включения пользователя в поисковый цикл с целью уменьшения "семантического разрыва" между тем, что формулируется в запросе и что намеревается найти пользователь. Как показали исследования, применение RF существенно повышает производительность CBIR-систем [123]. Тремя наиболее используемыми стратегиями RF являются изменение веса фичеров (re-weighting) [124], перемещение точки запроса (query point movement) [125] и байесовский метод [126]. В статье [127] описываются приведенные выше три метода и дается краткий обзор еще 21 метода RF в CBIR.

Дополнительную информацию по использованию RF в БД можно получить в подразделе "Обратная связь по релевантности" раздела "Базы данных видео".

Семантические шаблоны

Семантический шаблон (ST) - это отображение между высокоуровневыми понятиями и низкоуровневыми визуальными фичерами Обычно ST определяется как «репрезентативный» фичер понятия/концепта, вычисляемый из коллекции образцов изображений. Использованию ST в SBIR посвящены работы [128–131].

Литература

1) Döller M., Kosch H. Image Database. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 1761-1766
2) Tamura H, Yokoya N. Image database systems: a survey. Pattern Recognition, 1984,, Vol. 17, No 1, pp 29-43
3) Chang S.-K., Hsu A. Image information systems: Where do we go from here? IEEE Trans. on Knowledge and Data Engineering Vol. 4 No. 5, 1992 pp. 431–442
4) Shatford S. Analyzing the subject of a picture: a theoretical approach. Cataloging & Classification Quarterly. 1986, Vol. 6, No 3, pp. 39–62.
5) Content-based image retrieval. - https://en.wikipedia.org/wiki/Content-based_image_retrieval
6) Kato T. (April 1992). "Database architecture for content-based image retrieval". SPIE/IS&T 1992 Symposium on Electronic Imaging: Science and Technology, International Society for Optics and Photonics, 1992, pp. 112–123.
7) Lew M.S., Sebe N., Djeraba Ch., Jain R. Content-based multimedia information retrieval: State of the art and challenges. ACM Transactions on Multimedia Computing, Communications, and Applications, 2006, Vol. 2, No. 1, pp. 1–19
8) Jain R., Guest E., Special Issue on Visual Information Management, Communications of ACM, 40(12), 30-32, Dec. 1997.
9) Gudivada V.N., Raghavan J.V. Special issue on content-based image retrieval systems, IEEE Computer Magzine, Vol. 28, No. 9, September 1995.
10) Narasimhalu A.D. Special section on content-based retrieval. Multimedia Systems, 1995, 3 (1): 1-2.
11) Pentland A., Picard R., Special issue on digital libraries, EEE Transactions on Pattern Analysis and Machine Intelligence, 1996. No 8,
12) Schatz B., Chen H., Building large-scale digital libraries, Computer, 1996, Vol. 26, No. 5, pp. 22-26
13) Linda G. Shapiro, George C. Stockman. Computer Vision. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001,608 p.
14) Veltkamp R.C, Tanase M. Content-Based Image Retrieval Systems: A Survey, Dept. Computing Science, Utrecht University, Utrecht, The Netherlands, Tech. Rep. UU-CS-2000-34, 2002
15) Frank Y. Shih. Image Segmentation. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 1795-1803
16) Linda G. Shapiro and George C. Stockman (2001): “Computer Vision”, New Jersey, Prentice-Hall, 609 p.
17) Image segmentation. - https://en.wikipedia.org/wiki/Image_segmentation

18) Pal N.R., Pal S.K. A review on image segmentation techniques. Pattern Recognition. 1993, vol. 26, No. 9, pp. 1277–1294.
19) Manpreet Kaur, Lal Chand. Review of image segmentation and its techniques. Journal of Emerging Technologies and Innovative Research (JETIR), 2018, Vo. 5, No. 7, pp. 974-981
20) Salwa Abdulateef, Mohanad Salman. A Comprehensive Review of Image Segmentation Techniques. Iraqi Journal for Electrical And Electronic Engineering, 2021, vol. 17, No. 2, pp. 166-175
21) Plataniotis K.N., Venetsanopoulos A.N. Color Image Processing and Applications. Springer, Berlin, 2000.
22) Manjunath B.S., et al. Introduction to MPEG-7. Wiley, New York, 2002.
23) Zhao Q., Yang J., Liu H. Stone Images Retrieval Based on Color Histogram. In IEEE International Conference on Image Analysis and Signal Processing, 2009, pp. 157-161.
24) Stricker M., Orengo M. Similarity of color images. In Proc. SPIE Storage and Retrieval for Image and Video Databases, 1995.
25) Wan X,, Kuo K. Color distribution analysis and quantization for image retrieval. In SPIE Storage and Retrieval for Image and Video Databases IV, vol.SPIE 2670, 1996, pp. 9- 16
26) Smith J.R., Chang S.-F. Single color extraction and image query, in Proc. IEEE Int. Conf. on Image Proc., 1995.
27) Smith J.R., Chang S.-F. Tools and techniques for color image retrieval, in IS & T/SPIE Proceedings, Vol. 2670, Storage & Retrieval for Image and Video Databases IV, 1995.
28) Pass G., Zabih R., Miller J. Comparing images using color coherence vectors. In MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia, 1997, pp. 65–73
29) Haralick R.M., Shanmugam K., DinsteinI. Texture features for image classification, IEEE Transactions on Systems, Man, and Cybernetics, 1973, Vol. SMC-3, No. 6, pp.: 610-621
30) Gotlieb C.C., Kreyszig H.E. Texture descriptors based on cooccurrence matrices, Computer Vision, Graphics, and Image Processing, Vol. 51, No. 1, 1990, pp. 70–86
31) Tamura H., Mori S., Yamawaki T. Texture features corresponding to visual perception. EEE Transactions on Systems, Man, and Cybernetics, 1978, Vol.8, No. 6, pp. 460-473
32) Flickner M., Sawhney H., Niblack W., Ashley J., Qian Huang, Dom B., Gorkani M., Hafner J., Lee D., Petkovic D., Steele D., Yanker P. "Query by image and video content: the QBIC system". Computer. 1995, Vol. 28, No.9, pp. 23–32.
33) Huang T.S., Mehrotra S., Ramchandran K. Multimedia Analysis and Retrieval System (MARS) project. In P.B. Heidorn. B. Sandore (eds) Proceedings of the 33rd Annual Clinic on Library Application of Data Processing: Digital Image Access and Retrieval, Urbana, IL, March 1996, pp. 100-117. University of Illinois, 1997.
34) Smith J.R., Chang S.-F. Transform features for texture classification and discrimination in large image databases. In Proceedings of 1st International Conference on Image Processing, 1994, pp. 407-411.
35) Chang, Kuo C.-C.J. Texture analysis and classification with tree-structured wavelet transform. IEEE Transactions on Image Processing, 1993, Vol. 2, No.4, pp. 429–441
36) Gross M.H., Koch R., Lippert L., Dreger A. Multiscale image texture analysis in wavelet spaces, In Proceedings of 1st International Conference on Image Processing, 1994. pp. 412–416
37) Kundu A. Chen J.-L. Texture classification using QMF bank-based subband decomposition. CVGIP: Graphical Models and Image Processing 54(5), 1992, pp. 369–384.
38) Thyagarajan K.S., Nguyen T., Persons C. A maximum likelihood aproach to texture classification using wavelet transform. In Proceedings of 1st International Conference on Image Processing, 1994, vol. 2, pp. 640–644
39) Cross, Jain A.K. Markov random field texture models. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. PAMI-5, No.1, 1983, pp.25–39

40) Plataniotis K.N., Androutsos D., Venetsanopoulos A.N. Multichannel filters for image processing. Signal Processing: Image Communication, Vol 9, No. 2, 1997, pp.143-158
41) Ma W.Y., Manjunath B. Netra: a toolbox for navigating large image databases. Proceedings of the IEEE International Conference on Image Processing, 1997, pp. 568–571.
42) Pentland A.P. Fractal-based description of natural scenes, EEE Transactions on Pattern Analysis and Machine Intelligence. Vol. PAMI-6, No. 6, 1984, pp. 661–674.
43) Weszka J., Dyer C., Rosenfeld A., A comparative study of texture measures for terrain classification. EEE Transactions on Systems, Man, and Cybernetics. Vol. SMC-6, No. 4, 1976, pp. 269-285
44) Ohanian P.P., Dubes R.C. Performance evaluation for four classes of texture features. Pattern Recognition, vol. 25, no. 8, pp. 819–833, 1992.
45) Ma W.Y. Manjunath B.S. A comparison of wavelet transform features for texture image annotation. In Proceedings Second International Conference on Image Processing (ICIP'95), 1995, vol. 2, pp. 256-259,
46) Armi L., Fekri-Ershad S. Texture image analysis and texture classification methods - a review. International Online Journal of Image Processing and Pattern Recognition, Vol. 2, No.1, pp. 1-29, 2019
47) Rui Y., She A.C., Huang T.S. Modified fourier descriptors for shape representation—a practical approach, in Proc. of First International Workshop on Image Databases and Multi Media Search, 1996, pp. 22-23
48) Zahn C.T., RoskiesR.Z. Fourier descriptors for plane closed curves, IEEE Transactions on Computers, 1972, Vol. C-21, No. 3, pp. 269-281
49) Persoon E., Fu K.S. Shape Discrimination Using Fourier Descriptors. IEEE Transactions on Systems, Man, and Cybernetics, 1977, Vol. 7, No. 3, pp. 170-179
50) Hu M. Visual Pattern Recognition by Moment Invariants. IRE Transactions on Information Theory, IT-08, 1962, Vol. 8, No.2, pp. 179-187.
51) Yang M., Algregtsen F. Fast computation of invariant geometric moments: A new method giving correct results, in Proceedings of 12th International Conference on Pattern Recognition, 1994, pp. 201-204.
52) Pentland A., Picard R.W., Sclaroff S. Photobook: Content-based manipulation of image databases. International Journal of Computer Vision, 1996, 18 (3), pp. 233–254.
53) Arkin E.M., Chew L., Huttenlocher D., Kedem K., Mitchell J. An efficiently computable metric for comparing polygonal shapes, IEEE Trans. Patt. Recog. Mach. Intell. 13(3), 1991.
54) Lin H.-C., Chiu C.-Y., Yang S.-N. Finding textures by textual descriptions, visual examples, and relevance feedbacks. Pattern Recognition Letters, 2003, vol. 24, No. 12, pp. 2255-2267
55) Chuang G.C.-H., Kuo C.-C.J. Wavelet descriptor of planar curves: Theory and applications, IEEE Trans. Image Proc. 5(1), 56–70, 1996.
56) Li B., Ma S.D. On the relation between region and contour representation. Proceedings of 12th International Conference on Pattern Recognition, 1995.
57) Mehtre B.M., Kankanhalli M., Lee W.F. Shape measures for content based image retrieval: A comparison, Information Processing & Management 33(3), 1997.
58) Taubin G. Recognition and positioning of rigid objects using algebraic moment invariants, in SPIE Vol. 1570, Geometric Methods in Computer Vision, 1991.
59) Wallace I., Mitchell O. Three-dimensional shape analysis using local shape descriptors, IEEE Trans. Patt. Recog. and Mach. Intell., PAMI-3(3), May 1981.
60) Wallace I., Wintz P. An efficient three-dimensional aircraft recognition algorithm using normalized Fourier descriptors, Computer Graphics and Image Processing 13, 1980.
61) Faloutsos C., Flickner M., Niblack W., Petkovic D., Equitz W., Barber R. Efficient and Effective Querying by Image Content, Journal of Intelligent Information Systems, Vol. 3, No. 3-4, 1994, pp. 231–262.

62) Chua T.S., Tan K.-L., Ooi B.C. Fast signiture-based color-spatial image retrieval. In ICMCS '97: Proceedings of the 1997 International Conference on Multimedia Computing and Systems, 1997 .
63) Lu H., Ooi B., Tan K., Efficient image retrieval by color contents. In Proc. of the 1994 International Conference on Applications of Databases, 1994, pp 95–108
64) Smith J.R. Chang S.-F. Single color extraction and image query. In Proc. IEEE International Conference on Image Processing, 1995.
65) Rickman R.M., Stonham T.J. Content-based image retrieval using colour tuple histograms,.In Proc. SPIE Storage and Retrieval for Image and Video Databases IV, 1996.
66) Stricker M., Dimai A. Color indexing with weak spatial constraints. In Proc. SPIE Storage and Retrieval for Image and Video Databases IV, 1996.
67) Huang J., Kumar S., Mitra M., Zhu W.-J., Zabih R. Image indexing using color correlogram. In Proc.of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '97), 1997.
68) Mojsilovic A., Rogowitz B. ISee: perceptual features for image library navigation, Proceedings of the SPIE, Human Vision and Electronic Imaging, vol. 4662, 2002, pp. 266–277.
69) Mezaris V., Kompatsiaris I., Strintzis M.G. An ontology approach to object-based image retrieval. Proceedings of the ICIP, vol. II, 2003, pp. 511–514.
70) Chang S.K., Shi Q.Y., Yan C.W. Iconic indexing by 2D string. IEEE Trans. Pattern Anal. Mach. Intell. 9 (3) (1987) 413–428.
71) Ren W., Singh M., Singh C., Image retrieval using spatial context, Ninth International Workshop on Systems, Signals and Image Processing (IWSSIP'02), Manchester, November, 2002.
72) Smith J.R., Li C.-S. Decoding image semantics using composite region templates, IEEEWorkshop on Content-Based Access of Image and Video Libraries (CBAIVL-98), June 1998, pp. 9–13.
73) Gaede V., Günther O. Multidimensional access methods. ACM Comput Surv. 1998;30(2):170–231.
74) Ahn H.K., Mamoulis N., Wong H.M. A survey on multidimensional access methods. Technical report, Utrecht University (2001)
75) Venkateswaran J. A Survey of Recent Multidimensional Access Methods. Technical Report, University of Missouri-Rolla. -2004. 162 p.
76) Samet H. Foundations of Multidimensional and Metric Data Structures, Morgan Kaufman Series in Dala Management. Morgan Kaufman Publishers. San Francisco, CA. USA, 2006, 993 p.
77) Faloutsos C., Lin K.-I. Fastmap: A fast alogorithm for indexing, datamining and visualization of traditional and multimedia datasets. ACM SIGMOD Record, Vol. 24, No 2, 1995, pp. 163–174
78) Ng R.T., Sedighian A. Evaluating multidimensional indexing structures for images transformed by principal component analysis, in Proc. SPIE Storage and Retrieval for Image and Video Databases, 1996.
79) White D., Jain R. Similarity indexing: Algorithms and performance, in Proc. SPIE Storage and Retrieval for Image and Video Databases, 1996.
80) Chandrasekaran V., Manjunath B.S.,Wang Y.F., Winkeler J., Zhang H. An eigenspace update algorithm for image analysis. CVGIP: Graphical Models and Image Processing Journal, Vol. 59, No. 5,1997, pp. 321-332
81) Salton G., McGill M.J. Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.
82) Guttman A. R-tree: A dynamic index structure for spatial searching. ACM SIGMOD Record, Vol. 14, No. 2, 1984, pp. 47–57.
83) Sellis T., Roussopoulos N., Faloutsos C. The R+-tree: A dynamic index for multidimensional objects. Proceedings of the 13th VLDB Conference, Brighton 1987, pp. 507-518.
84) Greene D. An implementation and performance analysis of spatial data access methods. Proceedings of the Fifth International Conference on Data Engineering, 1989, pp. 606–615.
85) Beckmann N., Kriegel H.-P., Schneider R., Seeger B. The R*-tree: An efficient and

robust access method for points and rectangles. SIGMOD '90: Proceedings of the 1990 ACM SIGMOD international conference on Management of data, 1990, pp. 322–331.
86) White D. A., Jain R. Similarity indexing: Algorithms and performance. Proc. SPIE 2670, Storage and Retrieval for Still Image and Video Databases IV, 1996.
87) Charikar M., Chekur C., Feder T., Motwani R. Incremental clustering and dynamic information retrieval. Proc. of the 29th Annual ACM Symposium on Theory of Computing, 1997, pp. 626–635.
88) Rui Y., Chakrabarti K., Mehrotra S., Zhao Y., Huang T.S. Dynamic clustering for optimal retrieval in high dimensional multimedia databases. University of Illinois, Department of Computer Science Technical Report MARS-97-10. Urbana, IL: Department of Computer Science, 1997.
89) Zhang H.J., D. Zhong. A scheme for visual feature based image retrieval. Proc. SPIE 2420, Storage and Retrieval for Image and Video Databases III, 1995
90) Tversky A. Features of Similarity. Psychological Review, 1977, 84(4): 327-352.
91) Santini S., Jain R. "Similarity Matching." IEEE Transactions on Pattern Analysis and Machine Intelligence. 1996.
92) Deza M.M., Deza E. Encyclopedia of Distances. Springer; Softcover reprint of the original 3rd ed. 2014. 753 p.
93) Mahalanobis P.C. On the generalised distance in statistics. Proceedings of the National Institute of Sciences of India, 1936, vol. 2, No. 1, pp. 49—55.
94) Pearson K. Notes on the History of Correlation. Biometrika, 1920, Vol. 13, No. 1, pp. 25-45.
95) Jaccard P. (1901). Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. Bulletin de la Société Vaudoise des Sciences Naturelles 37, 241-272.
96) Kumar V., Chhabra J.K., Kumar M. Performance Evaluation of Distance Metrics in the Clustering Algorithms. INFOCOMP, 2014, Vol. 13, No. 1, pp. 38–51.
97) Gupta A., Jain R. Visual information retrieval, Commun. ACM 40 (5) (1997) 70–79.
98) Smith J.R., Chang S.F. VisualSeek: a fully automatic contentbased query system, Proceedings of the Fourth ACM International Conference on Multimedia, 1996, pp. 87–98.
99) Ma W.Y., Manjunath B. Netra: a toolbox for navigating large image databases, Proceedings of the IEEE International Conference on Image Processing, 1997, pp. 568–571.
100) Wang J.Z., Li J., Wiederhold G. SIMPLIcity: semantics-sensitive integrated matching for picture libraries, IEEE Trans. Pattern Anal. Mach. Intell. 23 (9) (2001) 947–963.
101) Smeulders A.W.M, Worring M., Santini S., Gupta A., Jain R. Content-based image retrieval at the end of the early years. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000, Vol. 22, No 12, pp.1349–1380.
102) Rui Y., Huang T.S., Chang S.-F. Image Retrieval: Current Techniques, Promising Directions, and Open Issues. Journal of Visual Communication and Image Representation. 1999, Vol. 10, No. 1, pp. 39-62
103) Long F., Zhang H.J., Feng D.D. Fundamentals of content-based image retrieval. In: Feng, D.D., Siu, WC., Zhang, HJ. (Eds.), Multimedia Information Retrieval and Management, Springer, Berlin, 2003.
104) Patil R.S., Agrawal A.J. Content-based Image Retrieval Systems: A Survey. Advances in Computational Sciences and Technology, 2017, Vol. 10, No. 9, pp. 2773-2788.
105) Eakins J., Graham M. Content-based image retrieval, Technical Report, University of Northumbria at Newcastle, 1999, 59 p.
106) Ying Liua, Dengsheng Zhanga, Guojun Lua, Wei-Ying Mab. A survey of content-based image retrieval with high-level semantics. Pattern Recognition, Vol. 40, No. 1, 2007, pp. 262–282
107) Mussarat Yasmin, Sajjad Mohsin, Muhammad Sharif. Intelligent Image Retrieval Techniques: A Survey. Journal of Applied Research and Technology, 2014, Vol. 12, No. 1, pp. 87-103
108) Kherfi M.L., Ziou D., Bernardi A. Image Retrieval from the World Wide Web: Issues, Techniques, and Systems, ACM

Computing Surveys, 2004, Vol. 36, No. 1, pp. 35–67
109) Popescu A., Grefenstette G. A Conceptual Approach to Web Image Retrieval. Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 2008, pp. 297-304
110) Berk T., Brownston L., Kaufman A. A new color-naming system for graphics language. IEEE Comput. Graphics Appl. 2 (3), 1982, pp. 37–44.
111) Stanchev P.L., Green Jr. D., Dimitrov B. High level color similarity retrieval, Int. J. Inf. Theories Appl. 10 (3) (2003) 363–369.
112) Rao A.R., Lohse M. Towards a texture naming system: identifying relevant dimensions of texture, IEEE Proceedings of the Fourth Conference on Visualization, 1993, pp. 220–227.
113) Shi R., Feng H., Chua T.-S., Lee C.-H. An adaptive image content representation and segmentation approach to automatic image annotation. International Conference on Image and Video Retrieval (CIVR), 2004, pp. 545–554.
114) Vailaya A., Figueiredo M.A.T., Jain H.J., Zhang A.K. Image classification for content-based indexing. IEEE Transactions on Image Processing, 2001, Vol. 10, No.1, pp. 117–130.
115) Town C.P., Sinclair D. Content-based image retrieval using semantic visual categories. Society for Manufacturing Engineers, Technical Report MV01-211, 2001.
116) Feng H., Chua T.-S. A boostrapping approach to annotating large image collection. Workshop on Multimedia Information Retrieval in ACM Multimedia, November 2003, pp. 55–62.
117) MacArthur S.D., Brodley C.E., Shyu C.-R. Relevance feedback decision trees in content-based image retrieval. Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL'00), June 2000, pp. 68–72.
118) Jain A., Dubes R. Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice-Hall, 1988.
119) Baraldi A., Alpaydin M Constructive feedforward ART clustering networks—Part I and II. IEEE Trans. Neural Netw., Vol. 13, No. 3, pp. 645–677, May 2002.
120) Shi J., Malik J. Normalized Cuts and Image Segmentation. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, Vol. 22, pp. 888-905
121) Dunn J.C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics, 1973, Vol. 3, No. 3, pp. 32–57
122) Chen Y., Wang J.Z., Krovetz R. An unsupervised learning approach to content-based image retrieval. IEEE Proceedings of the International Symposium on Signal Processing and its Applications, July 2003, pp. 197–200.
123) Rui Y., Huang T.S., Ortega M., Mehrotra S. Relevance feedback: a power tool for interactive content-based image retrieval, IEEE Trans. Circuits Video Technol. 8 (5) (1998) 644–655.
124) Wu Y., Zhang A. A feature reweighting approach for relevance feedback in image retrieval. Proceedings of the 9th International Conference on Image Processing (ICIP'02), pp. 581-584, 2002.
125) Hua K., Liu D. Query Point Movement Techniques for Content-Based Image Retrieval. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 2282–2288. Springer, New York, 2018
126) Giacinto G., Roli F. Bayesian relevance feedback for content-based image retrieval. Pattern Recognition, Vol. 37, No. 7, pp. 1499-1508, 2004.
127) Dongping Tian. A Review on Relevance Feedback for Content-based Image Retrieval. Journal of Information Hiding and Multimedia Signal Processing. 2018, Vol 9, No 1, pp. 108-119
128) Smith J.R., Li C.-S. Decoding image semantics using composite region templates. IEEEWorkshop on Content-Based Access of Image and Video Libraries (CBAIVL-98), June 1998, pp. 9–13.
129) Zhuang Y., Liu X., Pan Y. Apply semantic template to support content-based image retrieval, Proceedings of the SPIE, Storage and Retrieval for Media Databases, vol. 3972, December 1999, pp. 442–449.
130) Chang S.-F., Chen W., Sundaram H. Semantic visual templates: linking visual fea-

tures to semantics, International Conference on Image Processing (ICIP), Workshop on Content Based Video Search and Retrieval, vol. 3, October 1998, pp. 531–534.
131) Zhuang Y, Liu X., Pan Y. Apply semantic template to support content-based image retrieval, Proceedings of the SPIE, Storage and Retrieval for Media Databases, vol. 3972, December 1999, pp. 442–449.

Базы данных видео

База данных видео (БДВ*) - это база данных, которая предоставляет эффективные и развитые средства и технологии для поддержки процессов моделирования, хранения, структурирования, анализа, индексирования, поиска и манипулирования видео и их метаданными.*

Моделирование видеоконтента

Создание модели видеоконтента является решающим в использовании базы данных того или иного типа для хранения, индексирования, поиска и анализа видеоданных. Были предложены следующие подходы по моделированию [1, 2]:

- на основе сегментации;
- на основе аннотирования;
- на основе существенных объектов;
- алгебраический;
- статистический.

Моделирование на основе сегментации [3]. Моделирование видео этого класса предполагает разбиение видео на элементарные составляющие, с последующим их объединением в более сложные конструкции. Это по сути выявление структуры видео, о чем идет речь в подразделе "Структура видеоконтента".

Моделирование на основе аннотирования. В простейшем варианте данное моделирование предполагает связывание видео с аннотациями (ключевые слова или просто текст), которые описывают семантику видео. Были предложены две разновидности аннотирования: простыми значениями и парами атрибут-значение. В первом случае значения ассоциируются с логическими [4, 5] или физическими [6] последовательностями кадров. Во втором случае атрибуты используются для указания типа, диапазона, семантики или каких-либо других характеристик значений [7]. Подробнее об аннотировании видео идет речь в разделе "Аннотирование видео".

Моделирование на основе существенных объектов. В этом случае контент видео представляется в виде различных взаимосвязей важных для видео объектов.

Было предложено две разновидности этого моделирования:

- Сегментация на основе перемещения. Извлекаются существенные объекты видео вместе с данными об их перемещении (направление, траектория, скорость, вращение). Предлагаются графовый [8-10] и векторный [11] способы представления перемещения.
- Пространственно-временные взаимосвязи объектов. Является более информативным по сравнению с предыдущим подходом. Были предложены: пространственно-временной логический метод [12], графовый подход [13], метод с использованием иерархических сетей Петри [14], подходы с использованием скрытой марковской модели [15], динамической байесовской сети [16], факторного графа [17].

Алгебраическое моделирование видеоданных. Видеопоток определяется рекурсивным применением множества алгебраических операций к исходным видеоданным. В работе [18] впервые была определена алгебра для формулировки запросов к видео базе данных и на ее основе реализована экспериментальная система AVE (Algebraic Video Environment).

Статистическое моделирование. Используются практически на всех этапах технологии работы с видео: определение границ фрагментов, способов создания сцен, извлечения фичеров, анализа, классификации и аннотирования видео. В работе [19] анализируются различные статистические методы выявления структуры видео.

На приведенном ниже рис., взятым с небольшими дополнениями из [20], приведена общая схема технологии работы с БДВ, включающая анализ структуры видео, выявление фичеров, интеллектуальный анализ, индексирование и поиск видеоконтента. В этой технологической среде объединены многие из моделей и методов, описанных выше.

Используя данную схему опишем этот процесс.

Структура видеоконтента

Выявление структуры видеоконтента является важным шагом в решении задачи анализа видеоконтента (семантического анализа видео). Непосредственный анализ видео без его индексирования является довольно сложной задачей в связи с большими размерами и неструктурированостью его представления. Выявление структуры видеоконтента рассматривается как процесс иерархической декомпозиции видео на его компоненты и выявления взаимосвязей между ними. Базовой структурной компонентой видео является кадр, однако в связи с

огромным их количеством практически невозможно эффективно решать задачу поиска и доступа к видео. В связи с этом были предложены более сложные структурные компоненты.

В 1993 г. в работе [21] впервые было предложено разбивать видео на *фрагменты* (shots) - непрерывную последовательность кадров, заснятую одной камерой. Первый кадр каждого фрагмента устанавливался в качестве ключевого для индексирования. Затем в статье [22] был предложен метод разбиения фрагментов на *подфрагменты* с учетом поведения камеры - статичная/перемещается, изменяется/фиксируется направление взгляда. Также была предоставлена возможность ассоциировать с каждым фрагментом/подфрагментом множество ключевых кадров. Считается, что ключевой кадр - это такой кадр, который наилучшим образом представляет контент фрагмента/подфрагмента.

В [23] было предложено группировать фрагменты в *сцены* (scenes) - совокупность семантически связанных соседних во времени фрагментов, которые отражают события, происходящие в одном месте (в одной локации). Сцены обладают более высокой семантикой по сравнению с фрагментами

История (story) - совокупность семантически связанных соседних во времени сцен и отражающих законченную последовательность событий в видео. В настоящее время большинство методов идентификации историй проработаны только для видеоновостей.

Таким образом, в настоящее время иерархическая структура видео представлена следующим образом: кадр - ключевой кадр - подфрагмент - фрагмент - сцена - история - видео.

Метода анализа структуы видео

С учетом этой структуры были предложены и исследованы следующие пять типов методов анализа структуры видеоконтента: выявление видео-фрагментов, создание сцен, идентификация историй, сегментация на подфрагменты и извлечение ключевых кадров. Кратко рассмотрим их.

Выявление видео-фрагментов

Сущность выявления фрагментов (Video Shot Detection - VSD) заключается в идентификации границ между парами соседних фрагментов, которые бы позволяли группировать кадры в фрагменты. В зависимости от стиля перехода от одного фрагмента к другому выделяют два типа границ [24]:

- с резким переходом (abrupt/cut transition),
- с плавным переходом (gradual/soft transition).

В свою очередь переход с плавными границами имеет следующие три разновидности:

- **растворение** (dissolve) - значения пикселов текущего фрагмента плавно переходят в значения пикселов следующего фрагмент;
- **затемнение/появление** (fade in/out) - текущий фрагмент плавно переходит в кадр с одинаковым значением всех его пикселей и затем либо плавно появляется первый кадр следующего фрагмента (fade in), либо он появляется "мгновенно" (fade out);
- **вытеснение** (wipe) - пикселы текущего фрагмента последовательно заменяются пикселами следующего фрагмента согласно некоторому пространственному шаблону, например, построчно или по столбцам.

Было предложено множество методов выявления границ фрагментов, сущность большинства из которых заключается в задании критерия и меры различия между двумя соседними кадрами и установления предела, когда эта мера становится существенной. Наиболее известными являются следующие методы [24-28]:

- на основе пикселов (Pixel-Based),
- на основе гистограмм (Histogram-Based),
- на основе пороговых значений (Threshold-Based),
- статистический (Statistical-Based),
- трансформационный (Transform-Based),
- блочный (Block-based),
- на основе граней (Edge-Based),

– на основе перемещений (Motion-Based),
– на основе машинного обучения (Machine Learning-Based).

Хорошим введением в проблему выявления видео-фрагментов является энциклопедическая статья [28]. Содержательный обзор этой проблемы дан в статье [30]. Наконец, самыми последними всесторонними обзорами выявления фрагментов и ключевых кадров являются статьи [24, 31, 32]. Особый интерес представляет обзор [28]. На основании анализа около 100 источников описывается 71 метод выявления видео-фрагментов, которые сгруппированы в 7 классов. По каждому из методов указывается его вычислительная сложность.

Создание сцен

Были предложены и изучены три категории подходов по созданию сцен [20]:

– **На основе ключевых кадров** [34]. Видео-фрагменты представляются ключевыми кадрами, из них извлекаются фичеры и последовательные фрагменты с похожими фичерами группируются в сцены.
– **На основе взаимосвязи аудио и видео информации** [35]. Граница фрагмента, на которой одновременно изменяется контент аудио и видео, устанавливается границей сцены.
– **На основе фона** [36]. Предполагается, что фрагменты, имеющие похожие фоны, принадлежат одной сцене.

Что касается способов создания сцен, то выделяют 4 подхода: слияние, разделение, статистическое моделирование и на основе классификации границ фрагментов [20, 37].

Метод слияния относится к классу задач, решаемых снизу-вверх, то есть формирование сцен объединением подходящих фрагментов. Обычно слияние выполняется в два этапа [38, 39]: объединение фрагментов в группы/кластеры на основании различных критериев, например, визуальное подобие и темпоральная непрерывность, и затем объединение подходящих групп в сцены.

В свою очередь **метод разделения** поддерживает технологию сверху-вниз, когда видео разделяется на набор последовательных сцен. Например, в работах [40, 41] предложен граф-ориентированный метод разделения, когда вершины графа представляют фрагменты, а ребра определяют их подобие и темпоральную упорядоченность. Согласно выбранным критериям граф разбивается на подграфы, которым соответствуют сцены.

Модельный подход предполагает использование статистических методов для группирования фрагментов в сцены. Так в [42] для группирования используется модель гауссовой смеси (Gaussian Mixture Model), а в [37] предложено использовать так называемую схему минимизации энергии (energy minimization scheme). Хорошие обзоры модельной кластеризации приведены в статьях [43, 44].

Метод классификации границ фрагментов предполагает излечение фичеров границ фрагментов и последующую их разделение на таковые, которые принадлежат/не принадлежат границам сцен.

Идентификация историй

Идентификация историй требует более глубокого понимания семантики видеоконтента и обычно применяется к структурированным жанрам видео, например, к новостным. Выделяют два основных класса методов идентификации - на основе правил и на основе обучения. Базирующиеся на правилах методы обычно используют предварительные знания о предметной области. Так в [45] проанализирована структура новостных видео и идентификация истории производится на основе обнаружения фрагментов определенного типа, например, выступление диктора. Методы, основанные на обучении применимы к более широкому классу видео. Обычно в этих методах сначала определяется множество кандидатов на начало истории (границы фрагментов, наличие аудио-паузы), а затем определяется, является ли кандидат действительно началом новой истории, на основе методов, полученных в результате предварительного обучения. Дополнительную информацию об этих методах можно получить в [45, 46].

Сегментация на подфрагменты

Подфрагмент - это сегмент фрагмента. Подфрагменты формируются на основании анализа перемещения камеры. Были проведены исследования по анализу способов перемещения камеры (см. например, [47]), в результате которых были выделены следующие 6 видов перемещения, послужившие критериями создания подфрагментов:

- *панорамирование* (pan) - изменение направления взгляда камеры,
- *наклон* (tilt),
- *перемещение камеры* вперед/назад вдоль направления взгляда (увеличение/уменьшение изображения) (zoom),
- *поворот* (roll),
- *следование за объектом* (object motion),
- *статичное положение.*

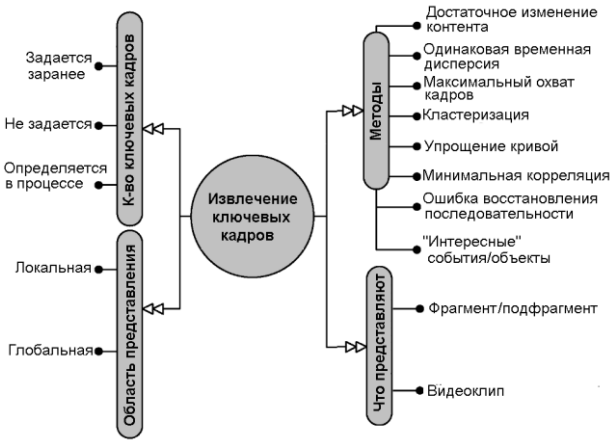
В статье [48] приводится обзор методов видео-сегментации.

Извлечение ключевых кадров

Ключевые кадры - это такие кадры, которые наилучшим образом представляют контент фрагмента/подфрагмента или всего видеоклипа. Они используются для организации поиска и доступа к видео. Было предложено множество методов извлечения ключевых кадров. На рис. ниже, взятом из [49] приводятся составляющие процесса извлечения ключевых кадров. Разработанные методы по разному ведут себя относительно количества создаваемых ключевых кадров. В одном случае это количество является входным параметром метода, в другом - никакого ограничения нет, в третьем - определяется в процессе работы метода.

Что касается области представления, то при локальной области каждый ключевой кадр представляет содержимое видеопоследовательности от его временной позиции до следующего ключевого кадра. А при глобальной области представления, ключевой кадр представляет визуальное содержимое всех кадров в своем кластере.

Наконец, они могут представлять фрагмент/подфрагмент или даже весь видеоклип.



Согласно [24, 25, 49] были предложены следующие методы извлечения ключевых кадров:

- последовательное сравнение кадров,
- на основе выборки (Sampling-Based),
- на основе фрагментов и подфрагментов (Shot-Based),
- кластеризация (Clustering-Based),
- собственные значения (Eigen Values),
- достаточное изменение контента (Sufficient Content Change),
- равных временных дисперсий (Equal Temporal Variance),
- максимальный охват кадров (Maximum Frame Coverage),
- минимальная корреляция между ключевыми кадрами (Minimum Correlation Among Keyframes),
- ошибка восстановления последовательности (Sequence Reconstruction Error),
- упрощение кривой (Curve Simplification),
- "интересные" события/объекты ("Interesting" Events/Objects),
- ссылочный кадр (Reference Frame).

Детально познакомиться с ними можно в же упомянутых обзорных статьях [24, 25, 49].

Извлечение фичеров

Кратко представим существующие подходы по извлечению визуальных и аудио-фичеров. Извлечение текстовых фичеров в этом подразделе не рассматривается.

Статичные фичеры ключевых кадров

Ключевые кадры отражают некоторые характеристические свойства видео. Тради-

ционно к ключевым кадрам применяются методы анализа изображений. В индексации и поиске ключевых кадров используются такие фичеры изображений, как цвет, текстура, фигуры, цветовое распределение, пространственное расположение, которые описаны в разделе "Базы данных изображений"

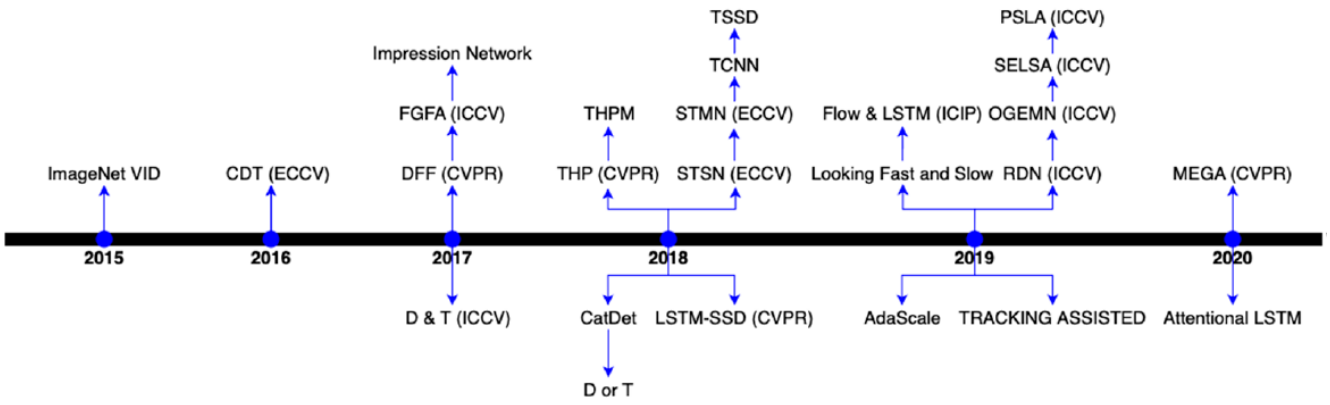
Фичеры объектов

Для обнаружения видеообъектов предлагаются различные методы анализа пространственно-временных взаимосвязей.

Согласно [50] были предложены приводимые далее методы обнаружения объектов в видео.

- **Потоковые методы** (Flow-Based). Эти методы либо используют идею распространения фичеров ключевого кадра на неключевые [51], либо агрегируют пространственно-временную информацию соседних кадров для уточнения фичеров каждого из них [52], либо предлагают комбинированные подходы.
- **Долгая краткосрочная память** (Long short-term memory, LSTM). Разновидность архитектуры рекуррентных нейронных сетей, предложенная в 1997 г. [53]. Чтобы в полной мере использовать пространственно-временную информацию, сверточная LSTM использовалась для обработки последовательных данных и выбора важной информации в течение длительного времени.
- **Основанные на внимании** (Attention-Related). Известно, что для обнаружения видеообъектов важным является использование темпоральных контекстных взаимосвязей, которые устанавливаются анализом продолжительных видео, что требует больших ресурсов. Для их уменьшения был предложен механизм внимания, впервые использованный для машинного перевода, а затем примененный к обнаружению видеообъектов [54].
- **Основанные на отслеживании** (Tracking-Based). Трекинг - это процесс отслеживания (построения траектории) движения объекта в кадрах. Были разработаны методы [55-58] для обнаружения объектов на кадрах с фиксированным интервалом и отслеживания их в кадрах между ними.
- **Методы глубокого обучения.** В статье [59] на основании анализа 230 источников дает обширный обзор использования методов глубокого обучения в решении задач обнаружения видеообъектов.
- **Комбинированные методы**. Были также разработаны методы, в которых объединяются приведенные выше методы.

В обзоре [50] приводится детальный анализ перечисленных выше методов и их разновидностей, дается их сравнительный анализ с указанием более 120 библиографических ссылок. На рис. ниже приводится упорядоченный по времени список 25 методов обнаружения видеообъектов, охваченных в этом обзоре.



Фичеры движения

Движение является важной характеристикой, которая отличает динамические видео от статичных изображений. Информация о движении характеризует визуальный контент во временном контексте. Фичеры движения обладают большей семантичностью, чем фичеры статичных ключевых кадров и объектов. В видео рассматриваются два вида движенний: перемещение фона, обусловленное перемещением камеры и перемещение переднего плана, обусловленное перемещением объектов. Разновидности перемещения камеры приведены в подразделе "Сегментация на подфрагменты". В свою очередь фичеры перемещения объектов классифицируются следующим образом [20, 27]:

- **Статистические**. Извлекаются статистические характеристики перемещения точек в кадрах для моделирования распределения локальной или глобальной динамики видео.
- **Траекторные.** Фичеры этого класса характеризуют траектории движения объектов.
- **Взаимосвязи объектов**. Эти фичеры описывают пространственные взаимосвязи между объектами

Аудио-фичеры

При индексации и поиске видео также используются следующие аудио-фичеры [60].

- **Кратковременная энергия** (Short Time Energy). В основном используется для разделения речевых и неречевых сегментов в аудио-записи. Оказывается весьма полезным при наличии шумов, так как они имеют низкую среднюю кратковременную энергию по сравнению с обычной речью
- **Частота основного тона** (Fundamental Frequency – F0). Очень важный фичер аудиоанализа, особенно для обнаружения выразительной человеческой речи. Он представляет собой ведущую частоту сложного аудиосигнала. В речи высота тона приобретает более высокие значения в результате возбуждения говорящего.
- **Мел-частотные кепстральные коэффициенты** (Mel-Frequency Cepstral Coefficients - MFCC) . Является одним из самых известных и популярных фичеров. Метод основан на изменении человеческого голоса с критической пропускной способностью с использованием частотных треугольных фильтров
- **Частота пауз** (Pause Rate). Этот фичер предназначен для определения количества речи в аудиосигнале. Может использоваться как указание на акцентированную человеческую речь. Вычисляется подсчетом количества бесшумных аудиокадров в аудиоклипе.
- **Обнаружения начала** (Onset detection). Предназначена для локализации переходных процессов в звуковом сигнале. Как правило, отыскиваются временные координаты значащих звуковых событий в аудиофрагменте.
- **Хромаграмма** (chromagram) или профиль класса высоты тона (pitch class profile) связана с двенадцатью различными классами высоты тона. Является мощным инструментом для анализа музыки, высота звука которой может быть осмысленно разделена на 12 категорий.
- **Латентное восприятие** (Latent Perceptual). Весь аудиоклип представляется как единый вектор в пространстве латентного восприятия. Это делает управляемой решение задачи определения меры сходства на основе аудиосигналов, требующую больших вычислительных ресурсов.

Интеллектуальный анализ видеоданных

Интеллектуальный анализ, классификация и аннотирование базируются в основном на структуре видео и извлеченных из него фичеров. Между этими составляющими технологии работы с видео нет принципиальных различий, в частности очень близкими являются понятия классификации и аннотирования.

Сущность интеллектуального анализа видеоданных (video data mining) заключает-

ся в том, чтобы на основании извлеченных из видео фичеров находить структурные закономерности видеоконтента, модели поведения движущихся объектов, содержательные характеристики сцен, закономерности событий и их ассоциаций, а также другие характеризующие семантику видео знания с тем, чтобы обеспечить функционирование интеллектуальных видео-приложений. В связи с мультимедийностью видео (изображения, звуки и текст) анализ видео включает анализ визуального, звукового и текстового контентов. Кроме того, также выделяют в качестве самостоятельного вида анализ динамики (перемещения). Далее, говоря об анализе видео-контентента, будем иметь в виду анализ визуального контента.

Анализ видеоконтента во многом базируется на анализе изображений, который описан в разделе "Базы данных изображений". Однако в случае видео мы имеем дело с большой совокупностью взаимосвязанных упорядоченных изображений (кадров), что привносит динамику, что является самостоятельным предметом анализа. Согласно [61] первые попытки семантического анализа видео и изображений были предприняты в 60-х - 70-х гг. в приложениях, связанных со сжатием видео и с видеофонами. Но уже в середине 90-х годов стали активно развиваться цифровые видео и появились первые исследовательские системы, которые выполняли анализ контента видео, продолжительностью в несколько минут.

Основные направления исследований

Выделяют три основных направления исследований в этой области:

- *построение детекторов* концептов (Concept Detectors);
- *анализ содержимого* изображений/кадров с точки зрения пространственных характеристик;
- *анализ динамических взаимосвязей* между кадрами в темпоральном аспекте.

Детекторы концептов - это процесс такой спецификации/выявления некоторых семантических концептов из определенной предметной области, чтобы их можно было бы распознавать в изображениях/видео. В связи с существованием огромного количества концептов в реальной жизни и очень сложным процессом их машинной спецификации, возникает вопрос: какое количество концептов реально требуется, какие именно детекторы концептов нужны. Так в работе [62] утверждается, что системы поиска видео, использующие несколько тысяч детекторов концептов, будут работать достаточно хорошо, даже если отдельные детекторы будут обладать низким качеством обнаружения. В свою очередь в работе [63] даются рекомендации по выбору наборов таких детекторов. В работе [64] описываются совместные усилия исследователей в области мультимедиа, ученых-библиотекарей и конечных пользователей по созданию большой стандартизированной таксономии, получившей название Large-Scale Concept Ontology for Multimedia (LSCOM). Результаты их усилий включают:

- lite-версию LSCOM, включающую 449 концептов;
- корпус из 61901 ключевого кадра видео, взятых из набора данных 2006 TRECVID, и аннотированных с помощью lite-LSCOM;
- полную таксономию 2638 концептов, построенную полуавтоматически, с отображением 884 концептов, вручную выбранных авторами, в базу знаний Cyc[2].

Наконец, работа [65] посвящена оценке качества детекторов концептов.

Анализ содержимого изображений/кадров с точки зрения пространственных характеристик. Научные исследования этого направления в основном концентрируются на извлечении визуальных фичеров синтаксического и семантического уровней. На начальном этапе исследовались методы извлечения фичеров низкого уровня: люминисцентность, цвет, текстура, ребра, фигуры, движение [66]. Но их было совершенно недостаточно, чтобы распознавать семантику видео. Поэтому в дальнейшем исследования были сконцентрированы

[2] Cyc - долгосрочный проект из области искусственного интеллекта по созданию всеобъемлющей онтологической базы знаний, охватывающей основные понятия и правила о том, как устроен мир. (https://en.wikipedia.org/wiki/Cyc).

на высокоуровневых фичерах, которые должны были сократить семантический разрыв. К ним относятся объекты видео (человек, машина, деревья) и семантические понятия (восход солнца, спортивное состязание, покупка товаров). Двумя популярными подходами достижения этих целей являются модель "мешок визуальных слов" (bag-of-visual-words) [67, 68] и машинное обучение [69].

Анализ динамических взаимосвязей между кадрами в темпоральном аспекте. Также проводятся исследования по распознаванию синтаксиса и семантики видео посредством анализа их темпоральной структуры. В основе этого направления лежит распознавание в видео таких его структурных компонент, как фрагментов, подфрагментов и их ключевых кадров (см. подраздел "Структура видеоконтента" выше). На основе синтаксической структуры фрагментов проводится высокоуровневый анализ видео с формированием сцен и историй. Наконец, еще одним шагом является процесс обнаружения видео-событий, которые в работе [70] представляются как стохастические темпоральные процессы в пространстве семантических концептов.

Стратегии интеллектуального анализа

Выбор стратегии интеллектуального анализа видеоданных зависит от используемого приложения. К настоящему времени были предложены и исследованы следующие стратегии [20].

- **Интеллектуальный анализ объектов**. Заключается в распознавании и группировании различных экземпляров объекта, появляющихся в различных структурных частях видео.
- **Обнаружение специальных ситуаций** (Special Pattern Detection). Применяются для обнаружения действий или событий, для которых заранее известны модели поведения, например, человеческая деятельность, спортивные события, движение транспорта, чрезвычайные ситуации.
- **Обнаружение закономерностей** (Pattern Discovery). Автоматическое распознавание в видео неизвестных ситуаций с использованием методов машинного обучения без учителя. Используются для выявления в видео новых данных или для порождения моделей для дальнейшего анализа. Неизвестные ситуации обычно выявляются посредством кластеризации различных фичерных векторов. В статье [71] дается обзор использования методов распознавания образов (pattern recognition) в задачах контентного поиска изображений и видео.
- **Анализ и выявление ассоциаций в видео** (Video Association Mining). Используется для выявления взаимосвязей между различными событиями или объектами, например, одновременное присутствие двух объектов, частота переключения фрагментов. Также включает механизмы вывода взаимосвязей между семантическими понятиями в одном фрагменте, а также вывод наличия семантических понятий в текущем фрагменте на основании анализа соседних фрагментов.
- **Анализ тенденций** (Tendency Mining). Это обнаружение и анализ тенденций определенных событий путем отслеживания текущих событий.
- **Анализ предпочтений** (Preference Mining). На основании интеллектуального анализа информационных потребностей пользователя строится модель его предпочтений, например, для построения персонализированного портала мультимедийных новостей.

Классификация видео

Задача классификации видео заключается в том, чтобы на основании извлеченных фичеров выявить определенные знания о видео, на основании которых можно было бы отнести видео к той или иной предварительно определенной категории. Классификация является важной составляющей повышения эффективности поиска видео. В связи с наличием семантического разрыва между извлеченными фичерами и содержательной интерпретацией видео человеком, делает задачу классификации контента видео довольно трудной. Согласно [72] классификация контента видео выполняется на

следующих уровнях:

- **Классификация по жанрам**. Были предложены следующие методы: статистические, основанные на правилах, на основе машинного обучения.
- **Классификация по событиям**. Эти методы тесно переплетаются с аналогичными методами интеллектуального анализа видео-данных. Было опубликовано множество работ по классификации событий, многие из которых кратко проанализированы в [20].
- **Классификация по объектам**. Тесно переплетаются с аналогичными методами интеллектуального анализа объектов. Предполагает предварительное извлечение фичеров объектов и последующую их классификацию. Во многих случаях используются предварительные знания об объектах определенного вида, например, модель его внешнего вида.

Аннотирование видео

Аннотирование - это процесс ассоциирования фрагментов и/или их сегментов с предварительно определенными семантическими понятиями, например, человек, транспортное средство, облако, дерево. Аннотирование сильно коррелирует с классификацией за исключением двух отличий: у них, как правило, разная онтология понятий, и во-вторых, классификация относится ко всему видео, а аннотирование - к фрагментам и сегментам, в связи с чем видео в целом может иметь множество аннотаций.

В 2006 г. в работе [73] были предложены следующие две стратегии аннотирования:

- **Аннотирование отдельными концептами** (Individual/Isolated Concept Annotation). Концепты выявляются индивидуально и независимо без учета какой-либо корреляции между ними.
- **Контекстно-зависимое аннотирование** (context-based annotation). Присутствие контекста, в котором используются концепты способствует повышению качества и эффективности выявления концептов. Задача контекстного аннотирования заключается в том, чтобы либо улучшить результаты выявления индивидуальных концептов либо выводить концепты более высокого уровня из выявленных низкоуровневых индивидуальных с использованием, так называемой стратегии контекстно-зависимого слияния концептов (Context Based Concept Fusion - CBCF).

Год спустя в работе [74] была предложена еще одна стратегия - **интегрированная многозначная** (Integrated Multi-label). Она предполагает одновременное выявление как индивидуальных концептов, так и взаимосвязей между ними. Кроме того, она позволяет производить многозначное аннотирование, то есть приписывать видео множество концептов.

В дальнейшем по каждой из этих стратегий было разработано множество методов, краткое описание которых приведено в [20]. Кроме того, в работе [75] приводится сравнительный анализ современных методов аннотирования видео, извлечения фичеров и семантического поиска.

Индексирование видео

Были предложены две разновидности индексации видео: синтаксическая и семантическая.

Синтаксическая индексация предполагает использование синтаксических низкоуровневых фичеров. Семантическая индексация использует высокоуровневые семантические концепты, полученные в результате интеллектуального анализа классификации и аннотирования. В статье [76] приводится краткий обзор методов индексирования изображений/видео.

Поиск видео

Построенная база данных индексов является основой для выполнения поиска, который включает формулировку запроса, собственно поиск с использованием различных мер подобия и возможную обратную связь для уточнения поиска.

Языки запросов

Языки запросов к БДВ можно специфицировать используя подходящие расширения SQL для видеоданных, например,

SQL2, STL (Spatial Temporal Logic), VideoSQL. Однако, в результате многолетних исследований были предложены специальные языки запросов различного вида, основные из которых приведены далее.

Запрос по образцу (Query by Example). Запрос содержит образец видно или изображения, из которого извлекаются низкоуровневые фичеры и затем с помощью соответствующих мер подобия производится сопоставление и выбор похожих видео. В этом случае, как правило, используются статические фичеры ключевых кадров.

Запрос по эскизу (Query by Sketch). Пользователю предоставляется возможность нарисовать эскиз отыскиваемого видео. Из него извлекаются фичеры, которые сопоставляются с фичерами запомненных видео. Например, в [77] предлагается язык, с помощью которого рисуется траектория движения, которая сопоставляется с траекториями, извлеченными из видео.

Запрос по объектам (Query by Objects). В запросе указывается изображение объекта и система отыскивает все его вхождения в видео [78]. Отличие запросов данного вида от двух предыдущих заключается в том, что в результате отыскиваются все места расположения объекта.

Запрос по ключевым словам (Query by Keywords). Самый простой вид запроса. Ключевые слова могут отражать семантику видео, Они могут относиться к метаданным видео, визуальным концептам и любой другой текстовой информации, которая приписывается к видео.

Запрос на естественном языке (Query by Natural Language). Это наиболее естественный и удобный способ формулировки запросов. Так, например, в [79] используется способ определения семантического подобия слов для нахождения и ранжирования наиболее релевантных видео. Для естественноязыковых запросов наиболее трудным является синтаксический разбор языка и выявление наиболее точной семантики.

Комбинированные запросы (Combination-Based Query). Сочетают в себе возможности запросов различного типа, например, текстовых и по образцу. Они хорошо подходят для мультимедийного поиска.

Для языков запросов к видео важным является вопрос создания удобного интерфейса. Согласно [20] по состоянию на 2011 год наиболее известными являлись Informedia [80] и MediaMill [81].

Сопоставление видео

Сопоставление - это основной механизм контекстного поиска, когда видео из БД сравниваются с тем, что указано в запросе. Производится неточное сравнение, а с учетом используемого понятия подобия, которое основывается на используемых в системе мер подобия. Существуют следующие разновидности сопоставления видео:

- сопоставление фичеров,
- сопоставление текстов,
- онтологическое сопоставление,
- комбинированное сопоставление.

Сопоставление фичеров. Наиболее простой мерой подобия является расстояние между двумя наборами фичеров. Например, в запросах по образцу сопоставляются низкоуровневые фичеры изображения запроса и кадров видео. В общем случае могут использоваться статические фичеры ключевых кадров, фичеры объектов и фичеры движения. В разделе "Базы данных изображений" описываются методы сопоставления изображений.

Сопоставление текстов. В самом простом варианте предполагает сопоставление терминов запроса с названиями концептов видео. Общие методы сопоставления текстов описаны в разделе "Полнотекстовые базы данных"

Онтологическое сопоставление. Для сопоставления используются онтологии концептов или семантические взаимосвязи между ключевыми словами. В этом случае сформулированный запрос обогащается или уточняется терминами и понятиями из соответствующих онтологий.

Комбинированное сопоставление. Одновременно используются методы из приведенных выше вариантов сопоставления. Успешно используется в мультимедийных системах.

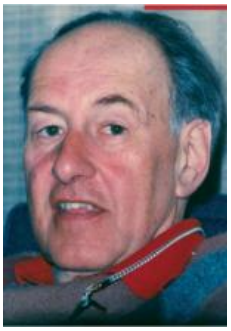
Обратная связь по релевантности

При работе с мультимедийными базами, как правило, невозможно точно выразить в запросе информационные потребнос-

ти. Это происходит по многим причинам и прежде всего в связи с существованием "семантического разрыва" между низкоуровневыми фичерами и используемыми человеком высокоуровневыми понятиями предметной области. Это привело к созданию механизма "обратной связи по релевантности" (relevance feedback - RF), суть которого заключается в том, что формулируется и выполняется поисковый запрос и на основании полученного результата производится уточнение/расширение/модификация запроса и повторное его выполнение. Эта процедура итеративно повторяется до тех пор, пока не будет получен удовлетворительный ответ.

Исследования и разработки в области автоматизированных информационно-поисковых систем, берут свое начало с проекта SMART (System for the Mechanical Analysis and Retrieval of Text), который был открыт в 1961 г. в Гарвардском университете [82], а в 1965 г. исследования и разработки по проекту переместились в Корнельский университет. В рамках исследований по проекту SMART возникло много концепций информационного поиска, наиболее важные из которых: модель векторного пространства [83], обратная связь по релевантности, алгоритм Роккио, частотная модель TF-IDF оценки термина в документе, а реализованная в 1965 г. система SMART [84] стала образцом на протяжении многих последующих лет для создания других поисковых систем. В 1971 г. была опубликована фундаментальная коллективная монография по SMART [85].

Профессор Корнельского университета Джерард А. Солтон (Gerard A. Salton) был руководителем проекта SMART, считается основателем компьютерного информационного поиска за что получил имя "отец информационного поиска" [86]. Солтон был главным редактором журналов Communications of the ACM и Journal of the ACM, а также редактором ACM Transactions on Information Systems. Возглавлял Специальную группу по поиску информации (SIGIR). Получил награду за заслуги Американского Общества Информационных Наук (1989 г.) и был первым лауреатом премии ACM/SIGIR за выдающийся вклад в поиск информации (1983) - теперь называется Премия Джерарда Солтона. Стал действительным членом ACM в 1995 г.,


Джерард Солтон

Впервые идея RF была "озвучена" в 1965 г. в статье Роккио (Rocchio J.J.) [87]. В ней помимо содержательного описания даются формальные математические основы RF. Со временем результаты, полученные Роккио, были систематизированы им в работе [88] и были названы "алгоритмом Роккио". Это классический алгоритм для реализации метода RF. Он инкорпорирует модель RF в модель векторного пространства. Он был реализован в SMART и стал широко известен благодаря этой системе. В последующем было предложено множество других методов и алгоритмов RF, в основном основанных на алгоритме Роккио [89-93].

Существует три типа RF:

– явная обратная связь по релевантности;
– обратная связь по псевдорелевантности;
– неявная обратная связь по релевантности.

Кратко рассмотрим их.

Явная обратная связь по релевантности (ERF)

В ERF пользователь самостоятельно анализирует результаты запроса, принимает решение об их релевантности и на основании этого модифицирует запрос для повторного выполнения. Предлагается два способа указания релевантности [94]:

– *бинарный* - документ указывается как релевантный или нерелевантный;
– *оценочный* - для каждого из документов результата указывается степень его релевантности.

На основании принятого решения о релевантности результатов поиска происходит модификация запроса. Предлагается три способа:

– *изменение веса терминов*. Указывается степень важности ранее использованных терминов для последующего поиска;
– *расширение запроса*. Добавляются новые термины с соответствующими весами;
– *комбинированный* - используются два предыдущих метода.

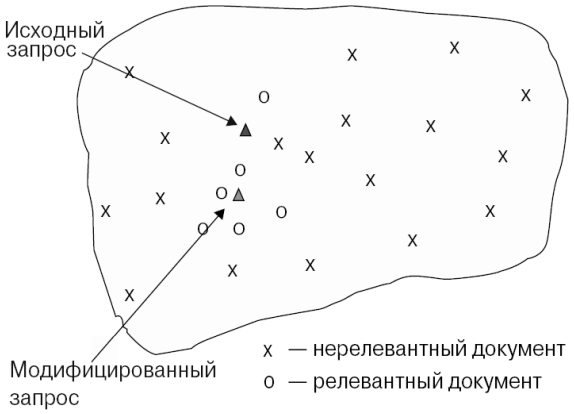
Во многих работах предлагается учитывать информацию о том, какие результаты запроса являются нерелевантными, например, указанием в запросе терминов, которые не должны присутствовать в последующих результатах.

Начиная с 2005 г. публикуются результаты исследований по метрикам производительности для измерения полезности алгоритмов ранжирования на основе ERF (см. https://en.wikipedia.org/wiki/Relevance_feedback)

Обратная связь по псевдорелевантности

Обратная связь по псевдорелевантности (pseudo relevance feedback - PRF), или слепая обратная связь по релевантности (blind relevance feedback), — это метод автоматизации той части RF, которая выполняется вручную, так что пользователь повышает качество поиска, не вступая в дополнительное взаимодействие с системой. В рамках этого метода сначала выполняется поиск и находится исходная совокупность наиболее релевантных документов, в которой первые k документов, имеющие наибольшие ранги, предполагаются релевантными, а затем к ним применяется метод RF с учетом этого предположения.

Считается [95], что метод PRF восходит к алгоритму Роккио [88], согласно которому происходило обновление запроса с использованием линейной комбинации разреженных векторов, представляющих как сам запрос, так и документы обратной связи с наивысшим рейтингом. Воспользуемся рисунком, взятым из [96], для иллюстрации PRF по алгоритму Роккио.



Со временем этот классический алгоритм PRF был уточнен и расширен другими моделями, например, Ide dec-hi [97], RSV (Robertson Selection Value) [89], CHI-2 (Chi-Squared) [91], Bo1 [92], KLD (Kullback-Leibler Divergence) [91] и RM3 [98], HI-Rocchio [99], которые продемонстрировали свою эффективность. Как правило, эти модели идентифицируют и взвешивают термины, которые часто встречаются в документах, принимаемых в расчет при обратной связи, и редко встречаются в остальных документах.

В последние годы в проблематике PRF стал развиваться подход по применению методов глубокого обучения с использованием нейронных трансформеров[3] с целью обогащения статистической информации о терминах документов [100-102].

Публикация в 2013 г. статьи [103] и появление программного пакета word2vec ознаменовали новое направление в информационном поиске, которое получило название вложения слов (word embeddings) или векторное представление слов. То есть было предложено вместо векторного представления документов/запросов представлять векторами слова, давая тем самым возможность представлять семантику слов. Таким образом, вместо того, чтобы идентифицировать термины в документах PRF с помощью статистических методов, методы уточнения запросов на основе вложений [104-108] расширяют запрос терминами, наиболее близкими к терминам запроса из пространства вложенных слов.

Были предложены и изучены два метода вложения слов:

– *Статический или контекстно-независимый,* когда слово имеет одно и то же представление не зависимо от того, где оно встречается. К нему относятся модели: Skip-Gram, реализованная в Word2Vec [109], GloVe [110], fastText [111].

[3] Трансформер (Transformer) - модель машинного обучения на основе архитектуры глубоких нейронных сетей. Предназначены для обработки текстов на естественном языке, решения задач машинного перевода, автоматического реферирования. См. https://en.wikipedia.org/wiki/Transformer_(machine_learning_model)

– Динамический или контекстно-зависимый, когда вложение слова меняется в зависимости от окружения, в котором оно встречается. К нему относятся модели ELMO [112], FLAIR Embeddings [113], BERT [114]. В настоящее время наиболее известными моделями этого класса для задач PRF являются как сама модель BERT[4], так и базирующиеся на ней модели Neural PRF [115], BERT-QE [116], CEQE [117], ColBERT-PRF [95]

Неявная обратная связь по релевантности

(Implicit relevance feedback IRF)

В этом случае в качестве базы для обратной связи используются косвенные свидетельыства вместо явных оценок релевантности. Система самостоятельно делает заключение об информационных потребностях человека на основании наблюдения, сбора и анализа его поведения при взаимодействии с поисковой системой.

Согласно [118, 119] существуют следующие методы IRF:

– *Данные по кликам*. Наблюдения за тем, на каких результатах поиска пользователь производит щелчки [118, 121, 122]. Это, возможно, наиболее широко используемая форма IRF. Основная идея заключается в том, что пользователь, вероятно, склонен производить щелчки на более релевантных результатах.
– *История запросов пользователя*. Наблюдения за историей запросов пользователя [121, 122]. Сюда входит переформулировка или переписывание ранее введенного запросов, что может свидетельствовать о неудовлетворенности человека возвращенными результатами. Изучение запросов, которые непосредственно предшествовали запросу, также может указывать на интерес пользователя, что можно использовать для устранения неоднозначности. Каноническим примером такого запроса является "Java", который может относиться к кофе, индонезийскому острову или языку программирования; зная, что один из предыдущих запросов был "C++", можно точно определить смысл текущего запроса.
– *Время просмотра*. Наблюдение за количеством времени, которое пользователь тратит на ознакомление с каждым результатом [123, 124, 125]. В работе [44] утверждается , что пользователь тратит больше времени на более релевантные результаты. В работе [123] приводится более десятка статей, посвященных этому вопросу.
– *Отслеживание взгляда*. Наблюдение за такими характеристиками человека, как фиксация направления взгляда и расширение зрачка, когда он наблюдает за результатами [126, 127] . Гипотеза состоит в том, что такие характеристики, как продолжительность фиксации и диаметр зрачка, различаются между релевантными и нерелевантными результатами. например, больший диаметр зрачка может указывать на релевантность.
– *Анализ, сохранение, использование, аннотирование*. В работе [119] были предложены и проанализированы такие виды человеческой активности, как анализ, сохранение, использование (examination, retention, reference), которые могут послужить основанием для выяснения его информационных потребностей. В [120] этот список расширяется аннотированием.
– *Вся история пользователя*. Наблюдения за всей информацией, созданной, скопированной или просмотренной человеком. Это может включать в себя все: просмотренные веб-страницы, электронные письма, участие в чатах, присутствие в социальных сетях, документы в файловой системе пользователя. В работе [128] предложены различные способы исполь-

[4] BERT - языковая модель, представляющая собой нейронную сеть, основу архитектуры которой составляет композиция кодировщиков трансформера. BERT является автокодировщиком. Предназначена для предобучения языковых представлений с целью их последующего применения в широком спектре задач обработки естественного языка. Начиная с 2019 г. Google использует BERT для распознавания смысла поисковых запросов. См. https://en.wikipedia.org/wiki/BERT_(language_model).

зования всей этой информации для вывода суждений о релевантности. Все это привело к появлению и использованию концепции персональных профилей информационных предпочтений (информационная модель пользователя) [120]. Краткий анализ этого направления приведен в [109].

Отметим еще 2 статьи, которые имеют отношение к IRF. В статье [129] приводится детальный анализ эффективности использования IRF с точки зрения процесса совершенствования задачи поиска и повышения уровня поисковой квалификации пользователей системы. В статье [130] дается классификация видов деятельности пользователей, которые следует учитывать в IRF, приводится перечень 34 работ согласно этой классификации и краткое описание результатов семи ключевых статей из этого списка.

В заключение данного подраздела отметим, что за все время исследований и разработок по теме RF было опубликовано множество научных отчетов, статей и монографий. Так, например, в базе данных DBLP (https://dblp.org/) имеется около 1400 статей, в названии которых присутствует фраза "relevance feedback". Тем не менее, приведем незначительный список монографий [131-134] и обзорных статей [135-139], которые помогут вам глубже понять суть проблематики RF.

Литература

1) Chen L. Video Content Modeling. In: Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 4389-4393
2) Marques O., Furht B. Introduction to Video Databases. In The Handbook of Video Databases: Design and Applications, B. Furht and O. Marqure, ed., CRC. Press, 2003, pp. 1-22
3) Cooper M. Video segmentation combining similarity analysis and classification. In: Proceedings of the 12th ACM International Conference on Multimedia; 2004. p. 252–255.
4) Smith T.G.A., Davenport G. The stratification system: a design environment for random access video. In: Proceedings of the International Workshop on Networking and Operating System Support for Digital Audio and Video; 1992. p. 250–261.
5) Weiss R., Duda A., Gifford D.K. Composition and search with a video algebra. IEEE Multimed. 1994;1(2):12–25.
6) Hjelsvold R., Midtstraum R. Modelling and querying video data. In: Proceedings of the 20th International Conference on Very large Data Bases; 1994. p. 686–694.
7) Oomoto E., Tanaka K. Ovid: design and implementation of a video-object database system. IEEE Trans. Knowl. Data Eng. 1993;4(5):629–643.
8) Courtney J.D. Automatic video indexing via ojbect motion analysis. Pattern Recognit. 1999;30(4): 607–625.
9) Li J., Özsu M.T., Szafron D. Modeling of moving objects in a video databas. In: Proceedings of the International Conference on Multimedia Computing and Systems; 1997. p. 336–343.
10) Nabil M., Ngu A.H.H., Shepherd J. Modeling moving objects in multimedia database. In: Proceedings of the 8th International Conference Database and Expert Systems Applications; 1997. p. 67–76.
11) Chang S.F., Chen W., Meng H.J., Urama H., Zhong D. A fully automated content-based video search engine supporting spatiotemporal queries. IEEE Trans. Circ. Syst. Video Technol. 1998; 8(5): 602–615.
12) Bimbo A.D., Vicario E., Zingoni D. Symbolic description and visual querying of image sequences using spatio-temporal logic. IEEE Trans. Knowl. Data Eng. 1995;7(4):609–22.
13) Day Y.F., Dagtas S., Iino M., Khokhar A., Ghafoor A. Object-oriented conceptual modeling of video data. In: Proceedings of the 11th International Conference on Data Engineering; 1995. p. 401–408.
14) Al-Khatib W., Ghafoor A. An approach for video meta-data modeling and query processing. In: Proceedings of the 7th ACMInternational Conference on Multimedia; 1999. p. 215–224.
15) Xie L., Chang S.F., Divakaran A., Sun H. (2002) Structure Analysis of Soccer Video with Hidden Markov Models. IEEE International Conference on Acoustics, Speech, and Signal Processing, Orlando, 13-17 May 2002, Vol. 4, 4096-4099.

16) Huang C.-L., Shih H.-C., Chao, C.-Y. (2006) Semantic Analysis of Soccer Video Using Dynamic Bayesian Network. IEEE Transactions on Multimedia , 8, 749-760.
17) Naphade M., Kozintsev I., Huang T., Ramchandran K. (2000) A Factor Graph Framework for Semantic Indexing and Retrieval in Video. Proceedings Workshop on Content-Based Access of Image and Video Libraries , Hilton Head Island, 12 June 2000, 35-39.
18) Picariello A., Sapino M.L., Subrahmanian V.S. A Video Database Algebra. In The Handbook of Video Databases: Design and Applications, B. Furht and O. Marqure, ed., CRC. Press, 2003, pp. 457-482
19) Vasconcelos N. Statistical Models of Video Structure and Semantics. In The Handbook of Video Databases: Design and Applications, B. Furht and O. Marqure, ed., CRC. Press, 2003, pp. 45-68
20) Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng, Stephen Maybank. A Survey on Visual Content-Based Video Indexing and Retrieval. IEEE Transactions on Systems, Man, And Cybernetics—Part C: Applications And Reviews, 2011, Vol. 41, No. 6, pp. 797- 819
21) Zhang H.-J., Kankanhalli A., Smoliar S.W. Automatic paritioning of full-motion video. Multimed Syst. 1993;1(1):10–28
22) Kim J-G, Chang H.S, Kim J, Kim H.M. Efficient camera motion characterization for MPEG video indexing. In: Proceedings of the IEEE International Conference on Multimedia and Expo. ICME2000. 2000, vol. 2, pp. 1171–1174
23) Rui Y., Huang T.S., Mehrotra S. Constructing table-of-content for video. Multimed Syst. 1999;7(5):359–368.
24) Bashir Olaniyi Sadiq, Bilyamin Muhammad, Muhammad Nasir Abdullahi, Gabriel Onuh, Ali Abdulhakeem Muhammed, Adeogun Emmanuel Babatunde. Keyframe Extraction Techniques: A Review. ELEKTRIKA- Journal of Electrical Engineering, 2020, 19(3):54-60
25) Wu K. Simple Implementations of Video Segmentation, Key Frame Extraction and Browsing, 2011. - https://imkaywu.github.io/assets/files/eece570_shot_boundary_detection.pdf
26) Borkar S.V., Katariya S.S. Survey on content Based Video Retrieval. International Journal Of Advance Research And Innovative Ideas In Education, 2016, Vol. 2, No 6, pp. 571-577
27) Fegade A., Dalal V. A Survey on Content Based Video Retrieval. International Journal Of Engineering And Computer Science, 2014, Vol. 3, No.7, pp. 7271-7279
28) Sébastien Lefèvre, Jérôme Holler, Nicole Vincent. A Review of Real-Time Segmentation of Uncompressed Video Sequences for Content-Based Search and Retrieval. Real Time Imaging, Elsevier, 2003, 9 (1), pp.73-98.
29) Chong-Wah Ngo. Video Shot Detection. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, 4434-4438
30) Yuan J., Wang H., Xiao L., Zheng W., Li J., Lin F., Zhang B. A formal study of shot boundary detection. IEEE Trans Circuit Syst Video Tech. 2007. Vol. 7, No. 2, pp.168–186.
31) Abdulhussain S.H., Ramli A.R., Saripan M.I., Mahmmod B.M., Al-Haddad S.A.R., Jassim W.A. Methods and challenges in shot boundary detection: a review. Entropy, Vol. 20, No. 4, pp. 214, 2018.
32) Smeaton A.F., Over P., Doherty A.R. Video shot boundary detection: Seven years of TRECVid activity. Comput. Vis. Image Understanding, 2010, Vol. 114, No. 4, pp. 411–418.
33) Shaimaa Toriah Mohamed Toriah, Atef Zaki Ghalwash, Aliaa A. A. Youssif. Semantic-Based Video Retrieval Survey. Journal of Computer and Communications, 2018, Vol.6, No.8, pp. 28-44
34) Truong B.T., Venkatesh S., Dorai C. Scene extraction in motion pictures. IEEE Trans. Circuits Syst. Video Technol., 2003, Vol. 13, No. 1, pp. 5–15
35) Sundaram H., Chang S.-F. Video scene segmentation using video and audio features. in Proc. IEEE Int. Conf. Multimedia Expo., New York, 2000, pp. 1145–1148.
36) Chen L.-H., Lai Y.-C., Liao H.-Y. M. Movie scene segmentation using background information. Pattern Recognit., 2008, Vol. 41, No. 3, pp. 1056–1065
37) Gu Z., Mei T., Hua X.-S., Wu X., Li S. EMS: energy minimization based video

scene segmentation. In: Proceedings of the IEEE International Conference on Multimedia and Expo; 2007, pp. 520-523
38) Rasheed Z., Shah M. Scene detection in Holleywood movies and TV shows. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2003, Vol. 2, pp. 343–350.
39) Rui Y., Huang T.S., Mehrotra S. Constructing table-of-content for video. Multimed Syst. 1999. Vol 7, No. 5, pp. 359–368.
40) Rasheed Z., Shah M. Detection and representation of scenes in videos. IEEE Trans Multimed. 2005, Vol. 7, No. 6, pp. 1097–1105
41) Yeung M., Yeo B., Liu B. Segmentation of videos by clustering and graph analysis. Computer Vision and Image Understanding. 1998, Vol. 71, No. 1, pp. 94–109.
42) Tang Y.-P., Lu H. Model-based clustering and analysis of video scenes. In: Proceedings of the International Conference on Image Processing, Volume 1, 2002, pp.617-620.
43) Melnykov V. Challenges in model-based clustering. WIREs Computational Statistics, Vol. 5, No 2, 2013, pp. 135–148
44) Grün B. Model-based Clustering, 2018. - https://arxiv.org/pdf/1807.01987.pdf
45) Zhang H.-J., Tan S.Y., Smoliar S.W. Automatic parsing and indexing of news video. Multimed Syst. 1995, Vol. 2, No. 6, pp. 256–265.
46) Chua T.-S., Chang S.-F., Chaisorn L., Hsu W. Story boundary detection in large broad-cast news video archives – techniques, experiences and trends. In: MULTIMEDIA '04: Proceedings of the 12th ACM International Conference on Multimedia, 2004, pp. 656-659
47) Kim J.-G., Chang H.S., Kim J., Kim H.M. Efficient camera motion characterization for MPEG video indexing. In: Proceedings of the IEEE International Conference on Multimedia and Expo; Vol.2, 2000. pp. 1171–1174
48) Dattatraya A. Jadhav, Parul S. Arora, Yogesh Kumar Sharma. Review of video segmentation approach. Journal of Emerging Technologies and Innovative Research (JETIR), 2019, Vol. 6, No 3, pp. 579-586
49) Truong B T, Venkatesh S. Video abstraction: A systematic review and classification. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), 2007, Vol. 3, No 1, pp. 1-37.
50) Haidi Zhu, Haoran Wei, Baoqing Li, Xiaobing Yuan, Nasser Kehtarnavaz. A Review of Video Object Detection: Datasets, Metrics and Methods. Applied Sciences, 2020, Vol. 10, No 21, 7834
51) Zhu X., Xiong Y., Dai J., Yuan L., Wei Y. Deep Feature Flow for Video Recognition. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Venice, Italy, 22–29 October 2017; pp. 4141–4150.
52) Zhu X., Wang Y., Dai J., Yuan L., Wei Y. Flow-Guided Feature Aggregation for Video Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 408–417.
53) Hochreiter S., Schmidhuber J. Long short-term memory. Neural Computation. 1997, Vol. 9, No 8, pp. 1735–1780.
54) Guo C., Fan B., Gu J., Zhang Q., Xiang S., Prinet V., Pan C. Progressive Sparse Local Attention for Video object detection. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019.
55) Mao H., Kong T., Dally W.J. CaTDet: Cascaded Tracked Detector for Effcient Object Detection from Video. arXiv 2018, arXiv:1810.00434.
56) Kim H.U., Kim C.S. CDT: Cooperative Detection and Tracking for Tracing Multiple Objects in Video Sequences. In Computer Vision—Eccv 2016; Part VI; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerland, 2016; pp. 851–867.
57) Luo H., XieW., Wang X., ZengW. Detect or Track: Towards Cost-Effective Video Object Detection/Tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
58) Feichtenhofer C., Pinz A., Zisserman A. Detect to Track and Track to Detect. In Proceedings of the 2017 IEEE Internation-

al Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3057–3065.
59) Zhong-Qiu Zhao, Peng Zheng, Shou-Tao Xu, Xindong Wu. Object Detection with Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(11), pp. 3212-3232
60) Patel B.V., Meshram B.B. Content based video retrieval systems. International Journal of UbiComp (IJU), 2012, Vol.3, No.2, pp. 13-30
61) Hauptmann A. Video Content Analysis. In: Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 4381-4388
62) Hauptmann A., Yan R., Lin W.-H., Christel M., Wactlar H. (2007) Can High Level Concepts Fill the Semantic Gap in Video Retrieval? A Case Study with Broadcast News. IEEE Transactions on Multimedia, 9, 958-966.
63) Lin W.-H., Hauptmann, A. (2006) Which Thousand Words Are Worth a Picture? Experiments on Video Retrieval Using a Thousand Concepts. 2006 IEEE International Conference on Multimedia and Expo, Toronto, 9-12 July 2006, 41-44.
64) Naphade M., Smith J.R., Tesic J., Chang S.-F., Hsu W., Kennedy L., Hauptmann A., Curtis J. Large-Scale Concept Ontology for Multimedia," IEEE MultiMedia, vol. 13, no. 3, pp. 86-91, July-September 2006.
65) Aly R., Hiemstra D. Concept Detectors: How Good is Good Enough? MM'09: Proceedings of the 17th ACM international conference on Multimedia, October 2009, pp. 233–242.
66) Li Y., Kuo C.-C. Introduction to content-based image retrieval – overview of key techniques, Chapter 10. In: Image Databases. New York: Wiley; 2002.
67) Li F., Perona P. A Bayesian hierarchical model for learning natural scene categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2005. p. 524–531.
68) Sivic J. Efficient visual search of videos cast as text retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, Vol. 31, No. 4. IEEE. pp. 591–605.
69) Boureau Y.-L., Bach F., LeCun Y., Ponce J. Learning mid-level features for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2010. p. 2559–2566.
70) Shahram Ebadollahi, Lexing Xie, Fu Chang, John Smith. Visual Event Detection using Multi-Dimensional Concept Dynamics. Conference: Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, ICME 2006, July 9-12 2006, Toronto, Ontario, Canada, pp. 881-884
71) Antani S., Kasturi R., Jain. A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. Pattern Recognition, 2002, Vol. 35, No 4, pp. 945-965
72) Yuan Y. Research on video classification and retrieval. Ph.D. dissertation, School Electron. Inf. Eng., Xi'an Jiaotong Univ., Xi'an, China, pp. 5–27, 2003..
73) Jiang W., Chang S.-F., Loui A. Active concept-based concept fusion with partial user labels. In Proceedings of IEEE International Conference on Image Processing, 2006.
74) Qi G.-J., Hua X.-S., Rui Y., Tang J.H., Mei T., Zhang H.-J. Correlative multi-label video annotation. In MM '07: Proceedings of the 15th ACM international conference on Multimedia, Augsburg, Germany, 2007, pp. 17–26.
75) Tamil Priya D, Divya Udayan J. A Comprehensive Survey On Various Semantic Based Video/Image Retrieval Techniques. International Journal of Innovative Technology and Exploring Engineering (IJITEE). 2019, Vol.8, No 6, pp. 186-196
76) Geetha P., Vasumathi Narayanan. A Survey of Content-Based Video Retrieval. Journal of Computer Science 4 (6): 474-486, 2008
77) Hu W.M., Xie D., Fu Z.Y., Zeng W.R. Maybank S.Semanticbased surveillance video retrieval. IEEE Trans. Image Process., Apr. 2007, Vol. 16, No. 4, pp. 1168–1181
78) Sivic J. Zisserman A. Video Google: Efficient visual search of videos. In Toward Category-Level Object Recognition.. Ber-

lin, Germany: Springer, 2006, pp. 127–144.
79) Aytar Y., Shah M., Luo J.B. Utilizing semantic word similarity measures for video retrieval. In Proc. IEEE Conf. Comput. Vis. Pattern Recog., Jun. 2008, pp. 1–8.
80) Christel M., Huang C., Moraveji N., Papernick N. Exploiting multiple modalities for interactive video retrieval. In Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Montreal, QC, Canada, 2004, vol. 3, pp. 1032–1035.
81) Worring M., Snoek C., de Rooij O., Nguyen G.P., Smeulders A. The mediamill semantic video search engine. In Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Apr. 2007, vol. 4, pp. IV.1213–IV.1216.
82) Salton G. The Smart document retrieval project. In: Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval. 1991. p. 356-358.
83) Salton G. A Vector Space Model for Information Retrieval. Communications of the ACM, 1975, Vol. 18, No.11, pp. 613-620
84) Salton G, Lesk M.E. "The SMART automatic document retrieval systems—an illustration". Communications of the ACM. 1965, Vol. 8. No.6, pp. 391–398.
85) Salton G. The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall, 1971 - 556 c.
86) The father of Information Retrieval. - http://www.cs.cornell.edu/gries/40brochure/pg24_25.pdf
87) Rocchio J.J. Jr. Relevance Feedback in Information Retrieval, Scientific Report ISR-9, Section 23, Harvard Computation Laboratory, Cambridge MA, August 1965.
88) Rocchio J.J. Relevance Feedback in Information Retrieval. In The SMART Retrieval System: Experiments in Automatic Document Processing, pp. 313-323, Englewood Cliffs, NJ, 1971. Prentice Hall, Inc.
89) Robertson S.E. On term selection for query expansion. Journal of Documentation 46(4), pp. 359–364 (1990)
90) Robertson S.E., Walker S., Hancock-Beaulieu M., Gatford M., Payne A. Okapi at TREC-4. In: The Fourth Text REtrieval Conference (TREC-4) , pp. 73–97 (1995)
91) Carpineto C., de Mori R., Romano G., Bigi B. An information-theoretic approach to automatic query expansion. ACM Transactions on Information Systems, 2001, Vol. 19, No. 1, pp. 1–27
92) Amati G., Van Rijsbergen C.J. Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Systems, 2002, Vol. 20, No. 4, pp 357–389
93) Ye Z., He B., Huang X., Lin H. Revisiting Rocchio's Relevance Feedback Algorithm for Probabilistic Models. In: Cheng P.-J., Kan M.-Y., Lam W., Nakov P. (eds.) Information Retrieval Technology - 6th Asia Information Retrieval Societies Conference, AIRS 2010. LNCS, vol. 6458, pp. 151–161
94) Gay, G., Haiduc, S., Marcus, A., Menzies, T. On the use of relevance feedback in IR-based concept location. In 2009 IEEE International Conference on Software Maintenance, 2009, pp. 351–360.
95) Wang X., Macdonald C., Tonellotto N., Ounis I. Pseudo-Relevance Feedback for Multiple Representation Dense Retrieval ICTIR '21: Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, 2021, pp. 297–306
96) Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008
97) Ide E. New experiments in relevance feedback. In The SMART Retrieval System: Experiments in Automatic Document Processing, pp. 337-354, Englewood Cliffs, NJ, 1971. Prentice Hall, Inc..
98) Jaleel N.A., Allan J., Croft B.W., Diaz F., Larkey L.S., Li X., Smucker M., Wade C. UMass at TREC 2004: Novelty and HARD. Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004.
99) Zeng A., Huang Y. A Text Classification Algorithm Based on Rocchio and Hierarchical Clustering. In ICIC'11: Proceedings

of the 7th international conference on Advanced Intelligent Computing, 2011, pp. 432–439
100) Dai Z., Callan J. Context-Aware Document Term Weighting for Ad-Hoc Search. In WWW '20: Proceedings of The Web Conference 2020, 2020, pp. 1897–1907
101) Nogueira R., Yang W., Lin J., Cho K. 2019. Document expansion by query prediction. arXiv preprint arXiv:1904.08375 (2019).
102) Nogueira R., Lin J. 2019. From doc2query to docTTTTTquery. Online preprint (2019).
103) Mikolov T., Chen K., Corrado G., Dean J. (2013) Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. arXiv:1301.3781v1
104) Diaz F., Mitra B., Craswell N. Query Expansion with Locally-Trained Word Embeddings. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 367–377, Berlin, Germany
105) Kuzi S., Shtok A., Kurland O. Query expansion using word embeddings. In CIKM '16: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management,2016, pp. 1929–1932.
106) Roy D., Paul D., Mitra M., Garain U. 2016. Using word embeddings for automatic query expansion. In Proceedings of SIGIR Workshop on Neural Information Retrieval. arXiv:1606.07608.
107) Zamani H., Croft B.W., Embedding-based query language models. In ICTIR '16: Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval, 2016, pp. 147–156.
108) Roy D., Ganguly D., Bhatia S., Bedathur S., Mitra M. Using word embeddings for information retrieval: How collection and term normalization choices affect performance. In CIKM '16: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2018, pp. 1835–1838.
109) McCormick C. (2016, April 19). Word2Vec Tutorial - The Skip-Gram Model. - http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
110) Pennington J., Socher R., Manning C. GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543,
111) Bojanowski P., Grave E., Joulin A., Mikolov T. Transactions of the Asation for Computational Linguistics, 2017, Vol. 5, pp. 135–146.
112) Peters M.E., Neumann M., Iyyer M., Gardner M., Clark Ch., Lee K., Zettlemoyer L.. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018, pp. 2227–2237,
113) Akbik A., Blythe D., Vollgraf R. Contextual String Embeddings for Sequence Labeling. In Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1638–1649
114) Devlin J., Chang M.-W., Lee K., Toutanova K. (11 October 2018). "BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding". arXiv:1810.04805v2
115) Li C., Sun Y., He B., Wang L., Hui K., Yates A. Sun L., Xu J. 2018. NPRF: A Neural Pseudo Relevance Feedback Framework for Ad-hoc Information Retrieval. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4482–4491
116) Zheng Z., Hui K., He B., Han X., Sun L., Yates A. 2020. BERT-QE: Contextualized Query Expansion for Document Reranking. In Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 4718–4728.
117) Naseri S., Dalton J., Yates A., Allan J. (2021). CEQE: Contextualized Embeddings for Query Expansion. In: Hiemstra, D., Moens, MF., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds) Advances in Information Retrieval. ECIR 2021, pp. 467–482

118) Lee L. Lecture 15: Implicit Relevance Feed-back & Clickthrough Data. - https://www.cs.cornell.edu/courses/cs6740/2010sp/guides/lec15.pdf
119) Oard D.W., Kim J. (1998) Implicit Feedback for Recommender System. In AAAI Workshop on Recommender Systems, Madison, WI: 81-83
120) Kim J., Oard D. W., Romanik K. User Modeling for Information Access Based on Implicit Feedback. In: Proceedings of ISKO-France July 5–6 2001, Nanterre: Universite de Paris X, 2001, pp. 1–11
121) Shen X., Tan B., Zhai C. Context-sensitive information retrieval using implicit feedback. SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, 2005, pp. 43–50
122) Radlinski F., Joachims Th. Query Chains: Learning to Rank from Implicit Feedback KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, 2005 pp. 239–248
123) Kelly D., Belkin, N.J. (2004). Display time as implicit feedback: Understanding task effects. In Proceedings of the 27th Annual ACM International Conference on Research and Development in Information Retrieval (SIGIR '04), Sheffield, UK, 377-384.
124) Morita M., Shinoda Y. Information filtering based on user behavior analysis and best match text retrieval. In SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 272-281, New York, NY, USA, 1994. Springer-Verlag New York, Inc.
125) Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. (1997) GroupLens: Applying collaborative filtering to Usenet News. Communication of the ACM, March, 40(3), 77-87.
126) 46) Salojärvi J., Puolamäki K., Kaski S. (2005). Implicit Relevance Feedback from Eye Movements. In: Duch, W., Kacprzyk, J., Oja, E., Zadrożny, S. (eds) Artificial Neural Networks: Biological Inspirations – ICANN 2005. pp. 513–518
127) Loboda T.D., Brusilovsky P., Brunstein J. Inferring word relevance from eye-movements of readers. IUI '11: Proceedings of the 16th international conference on Intelligent user interfaces, 2011, pp. 175–184
128) Teevan J., Dumais S.T., Horvitz E. Personalizing search via automated analysis of interests and activities. In SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 449-456, New York, NY, USA, 2005. ACM.
129) White R.W., Ruthven I., Jose J.M. A study of factors affecting the utility of implicit relevance feedback. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2005. pp. 35–42.
130) Kelly D., Teevan J. "Implicit feedback for inferring user preference: a bibliography." ACM SIGIR Forum. Vol. 37. No. 2. pp. 18–28, ACM, 2003.
131) Harman, D. K. 1992. Relevance feedback and other query modification techniques. In Information Retrieval – Data Structures and Algorithms, W. B. Frakes and R. Baeza-Yates Eds., Prentice Hall, Englewood Cliffs, N. J., 241–263.
132) Baeza-Yates R., Ribeiro-Neto B. 1999. Modern Information Retrieval. Addison Wesley.
133) Manning C.D., Raghavan P., Schutze H. 2008. Introduction to Information Retrieval. Cambridge University Press.
134) Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.
135) Ruthven I, Lalmas M. 2003. A survey on the use of relevance feedback for information access systems. The Knowledge Engineering Review, Vol. 18, No. 2, 2003, pp. 95–145
136) Zhou X.S, Huang T.S. Relevance feedback in image retrieval: A comprehensive review. Multimedia Systems, Vol. 8, No. 6, 2003, pp. 536-544
137) Carpineto C., Romano G. A Survey of Automatic Query Expansion in Infor-

mation Retrieval. ACM Computing Surveys, 2012, Vol. 44, No. 1, Article No. 1, pp. 1–50
138) Mohanan A., Raju S. A Survey on Different Relevance Feedback Techniques in Content Based Image Retrieval. International Research Journal of Engineering and Technology, Vol. 4, No. 2, 2017, pp. 582-585
139) Azad H.K., Deepak A. Query expansion techniques for information retrieval: A survey. Information Processing & Management, Vol. 56, No. 5, 2019, pp. 1698-1735

Мультимодельные базы данных

Концепция мультибаз данных, то есть систем, предоставляющих возможность работать со многими БД, зародилась в конце 70-х годов в связи с широким внедрением БД на производстве и созданием компьютерных сетей. Эта концепция постоянно развивалась на протяжении последующих 40 лет. Можно выделить следующие этипы ее развития:

- интеграция неоднородных баз данных;
- федеративные базы данны;
- многовариантное хранение;
- мультимодельные базы данных;
- поли-базы данных.

Далее подробно остановимся на мультимодельных БД, кратко описав все остальные.

Интеграция неоднородных баз данных

Идея интеграции баз данных с различными моделями данных зародилась в начале 80- годов, когда господствовали три модели данных - иерархическая, сетевая и реляционная, с целью обеспечения одновременного и совместного использования прикладной программой нескольких баз данных, организованных в рамках различных СУБД. Интеграции была направлена на преодоление программной и информационной несовместимости баз данных. Существенные результаты в этом направлении были получены советским ученым Л.А. Калиниченко. В 1983 г. он опубликовал монографию [1], в которой представлены методы решения проблемы интеграции баз данных, базирующиеся на создании общей модели данных высокого уровня и преобразовании произвольных моделей данных в общую модель.


Калиниченко Л.А.

Федеративные базы данных

Федеративная база данных (ФБД) - *это виртуальная база данных, представляющая собой прозрачную интеграцию многих автономных, возможно неоднородных и*

распределенных БД в логически единую БД для совместного использования и обмена данными.

Впервые идея федеративности была высказана в 1979 г. в отчете [2] и затем детально проработана Хеймбигнером и Маклеодом (Heimbigner , McLeod) в 1985 г. в работе [3]. ФБД предоставляет единый внешний интерфейс, позволяющий запоминать и отыскивать данные в автономных БД с использованием единого языка запросов. Для этого ФБД декомпозирует запрос на подзапросы для их отработки составляющими БД и затем объединяет полученные результаты с использованием так называемых «оболочек» (wrappers).

Очень важным аспектом ФБД является *автономность* их БД-компонент, то есть степень их самостоятельности. В работах [4, 5] была предложена и исследована следующая классификация автономности:

- *проектная* – способность самостоятельно принимать проектные решения любого плана;
- *коммуникационная* - способность принимать решения относительно того, следует ли взаимодействовать с другими БД-компонентами и каким образом;
- *исполнительная* – способность выполнять собственные локальные операции, инициируемые локальными пользователями или событиями, без какого-либо взаимодействия с внешним окружением федерации;
- *ассоциативная* – способность принимать решения относительно того, следует ли «делиться» своими функциональными возможностями и ресурсами с другими участниками федерации, и если да, то в какой степени, вплоть до самостоятельно выхода из федерации или входа в нее.

В работах [6-9] были также предложены и исследованы другие виды автономности.

Отличительной характеристикой ФБД является их *гетерогенность* (неоднородность), которая относится к моделям данных, семантике данных, ограничениям целостности и языкам запросов.

Важной особенностью ФБД является их способность поддерживать правила отображения/сопоставления схем баз данных федерации. Общепринятым решением является использование глобальной схемы, которая содержит релевантные составляющие схем-членов федерации и описания отображений в виде взглядов (views). При этом предлагается следующие два принципиальных решения в зависимости от направления отображения [10]:

- Global as View (GaV): глобальная схема определяется в терминах локальных схем;
- Local as View (LaV): локальные схемы определяются в терминах глобальной схемы.

Например, в известной федеративной системе Multibase [11] поддерживается глобальная схема и единый интерфейс для формулировки запросов. Запросы, сформулированные относительно глобальной схемы, декомпозируются согласно существующих подсхем и обрабатываются локальными базами данных.

Выделяют три категории ФБД [12]: *слабосвязанные* (loosely coupled), *сильносвязанные* (tightly coupled) и *гибридные* в зависимости от того, кто управляет федерацией и каким образом интегрируются компоненты. В слабосвязанной ФБД именно администратор БД ответственен за включение БД в федерацию и глубину ее интегрции с другими БД федерации. БД-компоненты такой федерации не находятся под управлением администратора ФБД. В некоторых источниках слабосвязанные ФБД называются *интероперабельными БД* (interoperable database).В сильносвязанной ФБД на ее администраторе лежит ответственность за создание и управление федерацией и активном контроле за доступом к компонентам-БД. В гибридном подходе делается попытка объединить преимущества предыдущих двух подходов, например, возможность прямого доступа ко многим другим хранилищам данных и использование глобальной схемы дл получения информации о локальных схемах..

Система является *однофедеративной* (single federation), если в ней можно представить неболее одной федеративной схемы. В противном случае систнема является *многофедеративной* (multiple federation).

Были проведены исследования по ар-

хитектуре ФБД, всесторонний анализ которых дан в обзоре [4],

В 80-х годах было разработано ряд слабосвязанных [13–15], сильносвязанных однофедеративных [16, 17] и сильносвязанных многофедеративных ФБД [18, 19][5].

Во второй половине 80-х г. была опубликована целая серия аналитических обзоров по данной тематике [4, 25-29], что свидетельствует о ее большой популярности в то время.

Многовариантное хранение

Возрождение идеи мультимодельности баз данных пришлось на начало 2000-х годов. В 2006 г. Нил Форд (Neal Ford) высказал идею многоязычного программирования (Polyglot Programming) [30]. Суть его заключается в следующем. Каждый язык программирования лучше всего подходит для решения задач определенного класса. В связи с этим при создании крупных систем желательно использовать не один язык, а несколько, программируя каждую функциональную задачу тем языком, который наиболее эффективно ее реализует.


Скотт Лебернайт


Лука Гарулли

На основании этой идеи в 2008 г. Скотт Лебернайт (Scott Leberknight) ввел понятие *многовариантного хранения* (polyglot persistence) [31] для баз данных. Имеется в виду возможность предоставлять подходящие способы представления, хранения и манипулирования данными для различных классов задач разрабатываемой системы с использованием множества одномодельных разнотипных баз данных и программы-посредника для интеграции этих баз данных. Было разработано ряд исследовательских прототипов, поддерживающих концепцию многовариантного хранения [32–34]. Так в системе Spark SQL [34] предоставляется API, в котором с помощью языков DataFrames и SQL можно работать с такими хранилищами данных, как JSON, JDBC, Hive, ORC и Parquet.

Мультимодельные БД

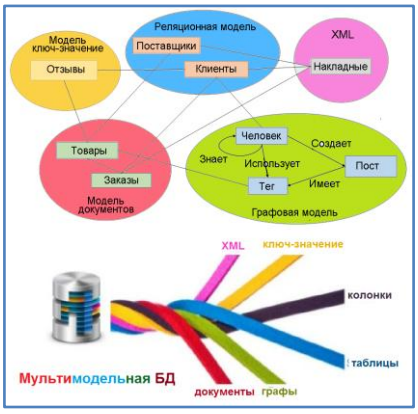
В 2009 г. Лука Гарулли (Luca Garulli) впервые разработал и выпустил на рынок мультимодельную БД OrientDB. Затем он впервые применил термин *многомодельность* по отношению к базам данных в 2012 г. на конференции "Nosql Matters 2012" в Кёльне, Германия, и предвосхитил развитие систем класса NoSQL в новые с дополнительной функциональностью, включая поддержку различных моделей данных [35)]. Он предложил создавать единые интегрированные многофункциональные NoSQL-продукты вместо того, чтобы собирать различные, отдельные системы NoSQL вместе для обеспечения аналогичного результата. И с тех пор именно в понимании Гарулли стали использовать термин "*мультимодельная база данных*", то есть единая система баз данных, поддерживающая множество моделей данных. Отметим, что Гарулли также разработал СУБД ArcadeDB, и Arcade Trader.

Мультимодельная БД (ММБД) - *это такая БД, которая поддерживает множество моделей данных в пределах одной интегрированной СУБД и обеспечивает стандарты данных и стандарты языков запросов каждой из моделей.*

Теория категорий. Было проведено ряд исследований [36-38], в которых обосновывается применение теории категорий для формального описания отображений между моделями данных в ММБД.

На рис. ниже приведена графическая интепретация мультимодельной БД.

[5] Как было отмечено в [4], в 80-е г. термин мультибазы данных широко использовался исследователями в различном контексте. Например, в работах [13, 14] под ним подразумеваются слабосвязанные ФБД, в работах [20, 21] имеются в виду сильносвязанные многофедеративные БД , а в работах [22-24] – сильносвязанные однофедеративные.

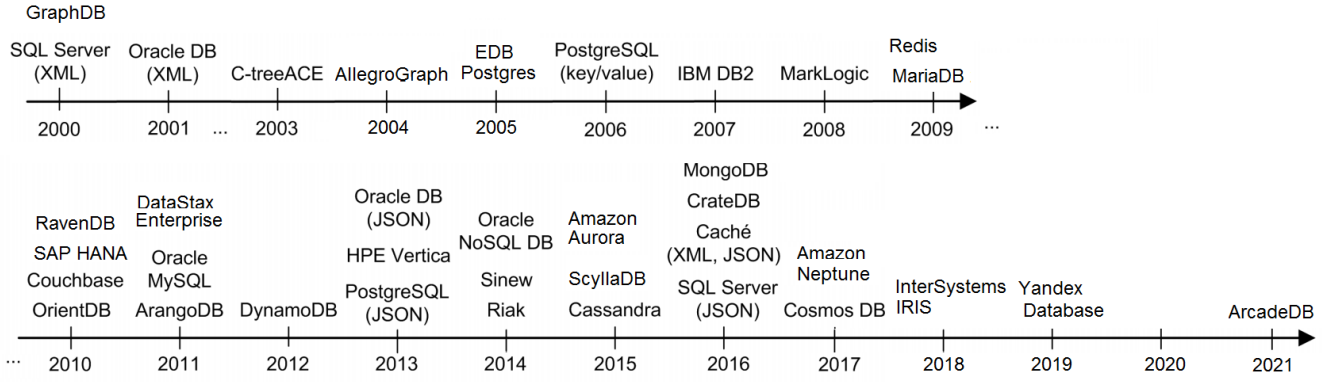


ММБД обладают всеми характерными для БД свойствами:

– хранение данных, их резервное копирование и восстановление;
– развитые языки запрооов и механизмы индексирования;
– проддержку ACID транзакций;
– "бесшовную" интеграцию различных моделей;
– развитые механизмы защиты и ограничений прав доступа.

Системы ММБД

На рис. ниже приведенна временная ось с укзанием годов появления мультимодельх систем либо благодаря расширению исходного формата новым, либо изначальной реализацией в качестве мультимодельной СУБД. Эволюция систем естественным образом соответствовала росту популярности соответствующих моделей. Так, например, первая волна появления ММБД пршлась на начало 21-го столетия в связи с появлением XML. Реляционные СУБД стали включать XML, используя стандарт SQL/XML или его диалекты. Вторая волна связана со вторым десятилетием в связи с наступлением эпохи NoSQL и больших данных. Уже в 2015 г. ведущие аналитики Gartner заявили: "*Будущее СУБД, их архитектур и способов использования — мультимодельность*" [39]. Начиная с 2017 г. все лидирующие производители СУБД предлагают мультимодельные решения, реляционные и NoSQL, на основе единой платформы.



Стратегии поддержки мультимодельности

Существуют три стратегии поддержки мультимодельности [40]:

– разработка принципиально новой стратегии хранения для поддержки многих моделей данных;
– расширение исходной стратегии хранения для поддержки дополнительных моделей данных;
– создание нового внешнего интерфейса на основе исходной стратегии хранения.

В следующей таблице приводятся ММБД согласно этой классификацми.

Стратегия поддержки мультимодельности	Тип исходной модели	СУБД
Новая стратегия хранения	реляционная	PostgreSQL
		SQL server
		IBM DB2
		Oracle DB
	колоночная	Cassandra
		CrateDB
		DynamoDB
	ключ/значение	Riak key
	документная	Cosmos DB
Расширение исходной страте-	реляционная	MySQL
	колоночная	HPE Vertica

гии хранения	документная	ArangoDB
		MongoDB
	графовая	OrientDB
	объектная	Cache
Новый внешний интерфейс на основе исходной стратегии хранения	реляционная	Sinew
	ключ/значение	C-treeACE
		Oracle NoSQL Database
	документная	Couchbase
		MarkLogic

Далее приводится таблица с указанием моделей/структур данных, поддерживаемых соответствующими ММБД. В списке дополнительных моделей присутствует столбец, который включает объектную модель, определяемые пользователем типы и вложенные структуры данных. Популярность СУБД взята из сайта DB-Engines Ranking (https://db-engines:com/en/ranking) по состоянию на март 2022 г.

Сравнительный анализ ММБД

По мере увеличения количества платформ мультимодельных баз данных стали проводиться исследования по их сравнитльному анализу. Например, в работах [41–43] приводится анализ существующих ММБД и даются сравнительные оценки ММБД с другими SQL и NoSQL БД.

Обзоры по ММБД

За последние годы было написано несколько аналитических обзоров и монографий по ММБД [4, 12, 40, 44–46]. В частности, данный раздел в основном написан на материале обзоров [4, 45].

Полихранилища

Поли-БД (poly-database) – это мульти-база данных, которая интегрирует множество гетерогенных баз данных и предоставляет множество интерфейсов для формулировки запросов [44]. Поли-БД сочетает в себе свойства мультимодельных и многовариантных БД. Как и мультимодельная, она поддерживает множество гетерогенных моделей данных, и вместе с тем, как и многовариантная, предоставляет множество внешних интерфейсов под каждую из поддерживаемых моделей данных. Считается, что первой системой полихранилищ была BigDAWG, представленная в 2015 г. [33]. Кроме того, к этому классу также относятся CloudMdsQL [47], Myria [48], Apache Drill [49], QoX [50], Musketeer [51], Rheem [52], AWESOME [53].

Исходная модель	СУБД	Дополнительные модели/структуры								Популярность (2022)	Адрес
		Реляционная/SQL	Колоночная	Ключ/значение	JSON	XML	Графовая	RDF	Вложенная/ОПТ/объектная		
Реляционная	PostgreSQL	✓		✓	✓	✓			✓	*****	https://wiki.postgresql.org/wiki/Main_Page
	SQL Server	✓			✓	✓	✓		✓	*****	https://ru.wikipedia.org/wiki/Microsoft_SQL_Server
	IBM DB2	✓				✓	✓	✓	✓	*****	https://en.wikipedia.org/wiki/IBM_Db2
	Oracle DB	✓		✓	✓	✓	✓	✓	✓	*****	https://en.wikipedia.org/wiki/Oracle_Database
	Oracle MySQL	✓		✓					✓	*****	https://en.wikipedia.org/wiki/MySQL
	Sinew	✓		✓						*	
	SAP HANA	✓			✓		✓			*****	https://en.wikipedia.org/wiki/SAP_HANA
Колоночная	Cassandra		✓				✓		✓	*****	https://ru.wikipedia.org/wiki/Apache_Cassandra
	CrateDB	✓	✓		✓		✓			***	https://en.wikipedia.org/wiki/CrateDB
	DynamoDB		✓	✓	✓		✓		✓	*****	https://en.wikipedia.org/wiki/Amazon_DynamoDB
	HPE Vertica		✓		✓		✓			***	https://en.wikipedia.org/wiki/Vertica
Ключ/ значение	Riak			✓	✓	✓	✓			*****	https://en.wikipedia.org/wiki/Riak
	c-treeACE	✓		✓			✓			*	https://en.everybodywiki.com/C-treeACE
	Oracle NoSQL DB	✓		✓			✓	✓		****	https://en.wikipedia.org/wiki/Oracle_NoSQL_Database
	Datastax	✓		✓			✓			*****	https://en.wikipedia.org/wiki/DataStax
	Redis			✓	✓		✓			*****	https://uk.wikipedia.org/wiki/Redis
Документная	ArangoDB			✓	✓		✓			****	https://en.wikipedia.org/wiki/ArangoDB
	Couchbase			✓	✓					*****	https://en.wikipedia.org/wiki/Couchbase_Server
	MongoDB			✓	✓				✓	*****	https://en.wikipedia.org/wiki/MongoDB
	Cosmos DB	✓	✓	✓	✓		✓		✓	*****	https://en.wikipedia.org/wiki/Cosmos_DB
	MarkLogic	✓			✓	✓	✓	✓	✓	*****	https://en.wikipedia.org/wiki/MarkLogic
	AllegroGraph				✓		✓			***	https://en.wikipedia.org/wiki/AllegroGraph
	ArcadeDB	✓		✓	✓		✓			*	https://en.wikipedia.org/wiki/ArcadeDB
	EnterpriseDB (EDB Postgres)	✓		✓	✓	✓				****	https://en.wikipedia.org/wiki/EnterpriseDB
Графовая	OrientDB	✓		✓	✓		✓			****	https://en.wikipedia.org/wiki/OrientDB
	GraphDB						✓	✓		****	https://db-engines.com/en/system/GraphDB
	Amazon Neptune						✓	✓		****	https://en.wikipedia.org/wiki/Amazon_Neptune
Объектная	InterSystems Caché	✓			✓	✓			✓	****	https://en.wikipedia.org/wiki/InterSystems_Cach%C3%A9

Литература

1) Kalinichenko L.A. Methods and Tools for Integration of Heterogeneous Databases (Rus). Moscow, Nauka, 1983, 424 p.
2) Hammer M., McLeod D. On database management system architecture. Tech. Rep. MIT/LCS/TM-141, 1979. Massachusetts Institute of Technology, Cambridge, Mass.
3) Heimbigner D., McLeod, D. A Federated architecture for information management. ACM Transactions on Information Systems, 1985, Vol. 3, Np. 3. pp. 253–278
4) Sheth A.P., Larson J.A. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Survey, 1990, Vol. 22, No. 3, pp. 183–236
5) Veijalainen J., Popescu-Zeletin R. Multidatabase systems in ISO/OSI environment. In Standards in Information Technology and Industrial Control, Malagardis N., and Williams T., Eds. North-Holland. The Netherlands. DD. 1988, 83-97.
6) Alonso R., Barbara D. Negotiating data access in federated database systems. In Proceedings of the 5th International Conference on Data Engineering, 1989, pp. 56-65.
7) Heimbigner D., McLeod D. A federated architecture for information management. ACM Transactions on Information Systems. 1985, Vol. 3, No. 3, pp. 253-278.
8) Du W., Elmagarmid A., Kim W. Effects of local autonomy on heterogeneous distributed database systems. MCC Tech. Rep. ACT-OODS-EI-059-90, 1990. Microelectronics and Computer Technology Corp., Austin Tex.
9) Garcia-Molina H., Kogan B. Node autonomy in distributed systems. In Proceedings of the International Symposium on Databases in Parallel and Distributed Systems (Austin, Tex., Dec.), 1988, pp. 158-166.
10) Lenzerini M. Data integration: a theoretical perspective. Proceedings of the 21-st ACM SIGCAT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2002). NewYork: ACM Press, 2002 ,pp. 233–246. .
11) Huang J.-W. MultiBase: a heterogeneous multidatabase management system. In Proceedings Eighteenth Annual International Computer Software and Applications Conference (COMPSAC 94), 1994, pp. 332–339.
12) Bondiombouy C., Valduriez P. Query processing in multistore systems: an overview. International Journal of Cloud Computing 5.4 (2016): 309-346
13) Litwin W. An overview of the multidatabase system MRDSM. In Proceedings of the ACM National Conference, 1985, pp. 495-504.
14) Rusinkiewicz M., Elmasri R., Czejdo B., Georakopoulous D., Karabatis G., Jamoussi A., Loa L., Ll Y. 1989. OMNIBASE: Design and implementation of aultidatabase system. In Proceedings of the 1st Annual Symposium in Parallel and Distributed Processing (Dallas, Tex., May), pp. 162-169.
15) Jacobson G., Piatetsky-Shapiro G., Lafond C., Rajinikanth M., Hernandez J. CALIDA: A knowledge-based system for integrating multiple heterogeneous databases. In Proceedings of the 3rd International Conference on Data and Knowledge Bases (Jerusalem, Israel, June), 1988, pp. 3-18.
16) Litwin W., Boudenant J., Esculier C., Ferrier A., Glorieux A., La Chimia J., Kabbaj K., Moulinoux C., Rolin P., Stangret C. SIRIUS Systems for Distributed Data Management. In Distributed Data Bases, H.-J. Schneider, Ed. North-Holland, The Netherlands, 1982, pp. 311-366.
17) Dwyer P., Larson J. Some experiences with a distributed database testbed system. In Proc. IEEE, 1987, Vol 75, No. 5, pp. 633-647.
18) Templeton M., Brill D., Chen A., Dao S., Lund E., McGregor R., Ward P. 1987. Mermaid: A front-end to distributed heterogeneous databases. In Proc. IEEE, 1987, Vol 75, No. 5, pp. 695-708.
19) Landers T., Rosenberg R. An overview of Multibase. In Distributed Databases, H.-J. Schneider, Ed., North-Holland, The Netherlands, 1982, pp. 153-184.
20) Ellinghaus D., Hallmann M., Holtkamp B.,Kreplin K. 1988. A multidatabase sys-

tem for transaction autonomy. In Proceedings of the International Conference on Extending Database Technology (Venice, Italy, Mar.). In Computer Science, Vol. 303, Springer-Verlag, New York, pp. 600-605.
21) Veijalainen J., Popescu-Zeletin R. 1988. Multidatabase systems in ISO/OSI environment. In Standards in Information Technology and Industrial Control, Malagardis, N., and Williams, T., Eds. North-Holland. The Netherlands. DD.-83-97.
22) Dayal U., Hwang H. 1984. View definition and generalization for database integration in a multidatabase system. IEEE Trans. Soft. Eng. SE-IO, 6 (Nov.), 628-644.
23) Belcastro, V., et al. 1988. An overview of the distributed query system D&S. In Proceedings of the International Conference on Extending Data Base Technolozy (Venice, Italv. Mar.). In Computer Science. Vol. 303, Spriger-Verlag, NewYork, pp. 170-189.
24) Breitbart Y., Silberschatz A. 1988. Multidatabase update issues. In Proceedings of the ACM SZGMOD Conference (June), 135-142.
25) Barker K., Ozsu T. 1988. A survey of issues in distributed heterogeneous database systems. Tech. Rep. TR 88-9, Univ. of Alberta Edmonton, Canada.
26) Litwin W., Zeroual A. 1988. Advances in multidatabase systems. In Research into Networks and Distributed Applications (Proceedings of the EUTECO‘88). Sneth. R.. Ed. Elsevier Science Publishers’ B.V., North-Holland, pp. 1137-1151.
27) Ram S., Chastain C. 1989. Architecture of distributed data base systems.Journal of Systems and Software, Vol. 10, No. 2, pp. 77-95.
28) Siegel M. 1987. A survey on heterogeneous database systems. Tech. Note 87-174.1, GTE Laboratories, Waltham, Mass.
29) Batini C., Lenzerini M., Navathe S. 1986. A comparative analysis of methodologies for database schema integration. ACM Computing Surveys, Vol. 18, No. 4, pp. 323-364.
30) Ford N. Polyglot Programming. - http://memeagora.blogspot.com/200/12/polyglot-programming.html, December 05, 2006
31) Leberknight S. Polyglot Persistence. - http://www.sleberknight.com/blog/sleberkn/entry/polyglot_persistence, October 15, 2008
32) Harold Lim, Yuzhang Han, and Shivnath Babu. 2013. How to Fit when No One Size Fits. In CIDR. www.cidrdb.org.
33) Duggan J., Elmore A.J., Stonebraker M., Balzinska M., Howe B., Kepner J., Madden S., Maier D., Mattson T.G., Zdoni S.B. 2015. The BigDAWG Polystore System. SIGMOD Record 44, 2 (2015), 11–16.
34) Michael Armbrust, Reynold S. Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K. Bradley, Xiangrui Meng, Tomer Kaftan, Michael J. Franklin, Ali Ghodsi, and Matei Zaharia. 2015. Spark SQL: Relational Data Processing in Spark. In SIGMOD '15: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, May 2015, pp. 1383–1394.
35) Garulli L. NoSQL adoption: what's the next step. - https://www.slideshare.net/lvca/no-sql-matters2012keynote/47-MultiModel_storage_12_one_product
36) Zhen Hua Liu, Jiaheng Lu, Dieter Gawlick, Heli Helskyaho, Gregory Pogossiant, Zhe Wu. Multi-Model Database Management Systems - a Look Forward. In: Heterogeneous Data Management, Polystores, and Analytics for Healthcare, 2018, pp.16-29
37) Valter Uotila, Jiaheng Lu. A Formal Category Theoretical Framework for Multi-Model Data Transformations. Rezig E.K. et al. (eds) Heterogeneous Data Management, Polystores, and Analytics for Healthcare. DMAH 2021, Poly 2021. Lecture Notes in Computer Science, vol 12921. pp. 14-28
38) Henrik Forssell, Håkon Robbestad Gylterud, David I. Spivak. Type theoretical databases. Journal of Logic and Computation, Vol. 30, No 1, January 2020, pp. 217–238
39) Zaidi E., Heudecker N., Adrian M. Market Guide for NoSQL DBMSs. - https://www.gartner.com/en/documents/3105622

40) Jiaheng Lu, Irena Holubová, and Bogdan Cautis. Multi-model Databases and Tightly Integrated Polystores: Current Practices, Comparisons, and Open Challenges. In CIKM '18: Proceedings of the 27th ACM International Conference on Information and Knowledge ManagementOctober 2018. pp. 2301–2302.
41) Pluciennik-Psota E., Zgorzalek K. The Multimodel Databases - A Review. BDAS 2017, pp. 141–152.
42) Fábio Roberto Oliveira, Luis del Val Cura. "Performance Evaluation of NoSQL Multi-Model Data Stores in Polyglot Persistence Applications". IDEAS '16: Proceedings of the 20th International Database Engineering & Applications Symposium, July 2016, pp. 230–235
43) Zhang C., Lu J., Xu P., Chen Y. (2019) UniBench: A Benchmark for Multi-model Database Management Systems. In: Nambiar R., Poess M. (eds) Performance Evaluation and Benchmarking for the Era of Artificial Intelligence. TPCTC 2018. Lecture Notes in Computer Science, vol 11135. Springer, pp 7-23
44) Ran Tan, Rada Chirkova, Vijay Gadepally, and Timothy G. Mattson. 2017. Enabling query processing across heterogeneous data models: A survey. 2017 IEEE International Conference on Big Data (Big Data). 3211–3220
45) Jiaheng Lu, Irena Holubová. Multi-model Databases: A New Journey to Handle the Variety of Data. ACM Computing Surveys, Vol. 52, No 3, 2020, Article No.: 55, pp 1–38
46) Aven P., Burley D. Building on Multi-Model Databases. O'Reilly Media, Inc., 2017, 96 p.
47) B. Kolev, C. Bondiombouy, P. Valduriez, R. Jimenez-Peris, ´ R. Pau, and J. Pereira, "The CloudMdsQL multistore system," in Proc. ACM International Conference on Management of Data (SIGMOD'16), 2016, pp. 2113–2116.
48) J. Wang, T. Baker, M. Balazinska, D. Halperin, B. Haynes, B. Howe, D. Hutchison, S. Jain, R. Maas, P. Mehta, D. Moritz, B. Myers, J. Ortiz, D. Suciu, A. Whitaker, S. Xu. The Myria big data management and analytics system and cloud service. The 8th Biennial Conference on Innovative Data Systems Research (CIDR '17), 2017
49) Hausenblas M., Nadeau J. Apache Drill: Interactive adhoc analysis at scale. Big Data, vol. 1, no. 2, pp. 100–104, 2013.
50) Simitsis A., Wilkinson K., Castellanos M., Dayal U. Optimizing analytic data flows for multiple execution engines, In Proc. ACM International Conference on Management of Data (SIGMOD'12), 2012, pp. 829–840.
51) Gog I., Schwarzkopf M., Crooks N., Grosvenor M.P., Clement A., Hand S. Musketeer: all for one, one for all in data processing systems. In EuroSys '15: Proceedings of the Tenth European Conference on Computer Systems, April 2015 Article No.: 2, pp. 1–16
52) Agrawal D., Ba L., Berti-Equille L., Chawla S., Elmagarmid A., Hammady H., Idris Y., Kaoudi Z., Khayyat Z., Kruse S., Ouzzani M., Papotti P., Quiane-Ruiz J.-A., Tang N., Zaki M.J. Rheem: Enabling multi-platform task execution. In Proc. ACM International Conference on Management of Data (SIGMOD'16), 2016, pp. 2069–2072.
53) Dasgupta S., Coakley K., Gupta A. Analytics-driven data ingestion and derivation in the AWESOME polystore. In Proc. IEEE International Conference on Big Data (ICBD'16), 2016, pp. 2555–2564.

Этап 6. – Большие данные (2010 – 2020+)

Мировой объем оцифрованной информации растет по экспоненте. Начиная с 1980-х годов цифровая информация удваивается каждые 40 месяцев. По данным компании IBS, к 2003 году мир накопил 5 эксабайтов данных (1 ЭБ = 1 млрд гигабайтов), а теперь это количество порождается каждые два дня. К 2008 году этот объем вырос до 0,18 зеттабайта (1 ЗБ = 1024 эксабайта), к 2011 году – до 1,76 зеттабайта, к 2013 году – до 4,4 зетта-байта. В мае 2015 года глобальное количество данных превысило 6,5 зеттабайта. 2020 году, по прогнозам, человечество сформирует 40–44 зеттабайтов информации, а к 2025 г. – 163 зеттабайт.

Приведем цитату из [1449], которая раскрывает суть проблемы больших данных: «Данных становится все больше и больше, но при всем этом упускается из виду то обстоятельство, что проблема отнюдь не внешняя, она вызвана не столько обрушившимися в невероятном количестве данными, сколько неспособностью старыми методами справиться с новыми объемами. Наблюдается дисбаланс – способность порождать данные оказалась сильнее, чем способность их перерабатывать... Под именем Big Data скрывается намечающийся качественный переход в компьютерных технологиях, способный повлечь за собой серьезные изменения. Не случайно этот переход называют новой технической революцией».


Клиффорд Линч

Широкое использование термина «большие данные» связывают с Клиффордом Линчем (Clifford Lynch), редактором журнала Nature, подготовившим к 3 сентября 2008 года специальный выпуск номера старейшего британского научного журнала, посвященный поиску ответа на вопрос «*Как могут повлиять на будущее науки технологии, открывающие возможности работы с большими объёмами данных*?» [2]. В этом номере были собраны материалы о феномене взрывного роста объёмов и многообразия обрабатываемых данных и технологических перспективах в парадигме вероятного скачка «от количества к качеству»; термин был предложен по аналогии с расхожими в деловой англоязычной среде метафорами «большая нефть», «большая руда», отражающими не столько количество чего-то, сколько переход количества в качество. Этот специальный номер подытоживает предшествующие дискуссии о роли данных в науке вообще и в электронной науке (e-science) в частности.

Следует также отметить, что впервые этот термин все же озвучил Джон Р. Маши (John R. Mashey) в 1998 г. [3, 4], который по поводу его употребления сказал: «*Мне нужна была самая простая и короткая фраза, чтобы указать, что границы вычислительной техники продолжают расширяться*».


Джон Р. Маши

Некоторые вехи в истории развития Big Data

Этот термин был сначала введен в академической среде и прежде всего обсуждалась проблема роста и многообразия научных данных, но начиная с 2009 года термин широко распространился в деловом мире. В 2010 году появляются первые продукты и технологии, относящиеся непосредственно к проблеме обработки больших данных. К 2011 году большинство крупнейших поставщиков информационных технологий в своих деловых стратегиях начинают использовать понятие «большие данные», это, в частности, относится к IBM, Oracle, Microsoft, Hewlett-Packard, EMC, а основные аналитики рынка информационных технологий посвящают концепции специальные исследования.

Большой шум вокруг темы больших данных возник после того, как в июне 2011 года консалтинговая компания McKinsey выпустила доклад «Большие данные: следующий рубеж в инновациях, конкуренции и производительности», в котором оценила

потенциальный рынок больших данных в миллиарды долларов.

В этом же году аналитическая компания Gartner отметила большие данные как тренд номер два в информационно-технологической инфраструктуре (после виртуализации). В это же время прогнозировалось, что технология больших данных окажет наибольшее влияние на информационные технологии, в производстве, здравоохранении, торговле, государственном управлении.

В 2012 году администрация президента США выделила 200 миллионов долларов для того, чтобы различные американские ведомства организовывали конкурсы по внедрению технологий больших данных в жизнь. Если в 2009 году американские венчурные фонды вложили в отрасль всего 1,1 миллиарда долларов, то в 2012 – уже 4,5 миллиарда долларов.

С 2013 года большие данные как академический предмет начинают изучать в появившихся вузовских программах по науке о данных и вычислительным наукам и инженерии. В 2015 году Gartner отметила, что технология больших данных перешла от этапа шумихи к практическому применению.

Имеется множество определений больших данных. [5, 6]. Обобщая эти и другие материалы, дадим следующие определение.

Большие данные (Big Data) – это огромные объемы неоднородной, неструктурированной или слабо структурированной, существенно распределенной и интенсивно растущей, изменяющейся и используемой цифровой информации, которую невозможно обработать традиционными средствами. А также методы, технологии и средства их сбора, хранения и обработки и анализа с целью получения воспринимаемых человеком результатов.

Характеристические свойства больших данных

В 2001 г. Дуглас Лэйни (Douglas Laney), аналитик Gartner Inc., сформулировал определяющие характеристики современных данных [7], которые получили название "Три V": Volume, Velocity, Variety (объем, скорость, разнообразие). Хотя он не говорил о больших, данных, а просто о данных, однако в научном мире эти три свойства стали рассматриваться в качестве определяющих характеристик именно больших данных.


Дуглас Лэйни

Volume (объем). Считается, что Big Data начинаются с объемов в петабайты. Big Data появляются тогда, когда сотни миллионов людей объединяются в сообщества и выкладывают свои информационные ресурсы, либо объединенные центры научных исследований предоставляют данные результатов своих исследований, например в 2017 году дата-центр CERN превысил размер 200 петабайт и ежегодно этот объем увеличивается на 15 петабайт.

Velocity (скорость). Является одной из наиболее важных характеристик Big Data с точки зрения их практического использования. Под скоростью подразумевается как скорость прироста (поступления, накопления) данных, так и скорость их обработки с целью получения конечных результатов. Кроме того, в эту категорию включаются характеристики интенсивности и объемов информационных потоков. Для этого технология обработки таких данных должна допускать возможность их анализа уже в момент их порождения (иногда называемой «оперативной аналитикой » - in- memory analytics), то есть до того, как они попадут в хранилище данных. Несколько цифр, характеризующих эту категорию, которые взяты из [8] и некоторых других источников.

YouTube: Имеет более 1 миллиарда зарегистрированных пользователей и ежемесячно сайт посещают 1,9 миллиарда пользователей. Ежеминутно закачивается новых фильмов на 100 часов и скачивается фильмов на 700 тысяч часов. Для просмотра фильмов, выгруженных в YouTube в течении дня, потребуется 15 лет.

Facebook: Имеет 1,4 миллиарда пользователей. Ежедневно на сайт выгружается 100 терабайт данных и ежеминутно ставятся более 34 тысячи лайков. Каждую минуту загружается 200 000 фотографий. Каждый ме-

сяц выкладывается в открытый доступ 30 млрд новых источников информации.

Twitter: Сайт имеет более 645 миллиона пользователей. Каждый день генерируется 175 миллион твитов.

Google: Каждую минуту отрабатывается 2,4 миллиона поисковых запросов (40 000 запросов в секунду). Каждый день обрабатывается 25 петабайт данных.

По мнению специалистов, к категории Big Data относится большинство потоков данных свыше 100 Гб в день.

Variety (разноообразие). Возможность воспринимать, хранить и обрабатывать различные данные. Говоря о многообразии, подразумевается:

- наличие различных источников получения данных;
- существование различных способов представления данных;
- семантическое разнообразие;
- различная степень структурированности данных (структурированные, слабо структурированные, неструктурированные).

Технология Big Data позволяет объединять и обрабатывать данные, обладающие приведенному выше многообразию.

Затем Зикопулос (Zikopoulos) [9] предложил добавить еще 2 признака – достоверность и ценность (Veracity, Value), получив таким образом «5V».

Veracity (достоверность). Свойство, которое характеризует надежность данных. Технология создания и использования традиционных БД предполагает, что в БД поступают тщательно отобранные и проверенные данные. В Big Data дело обстоит иначе. Исходные данные могут быть «сырыми» (неполными, неточными, нечеткими, расплывчатыми, искаженными), то есть поступают без какой-либо предварительной обработки, они могут быть субъективными, случайными и содержать много «шума». Еще один критерий этой характеристики – степень доверия к поступающим данным. Хотя Big Data предоставляют прекрасные возможности для анализа и принятия решений, однако их ценность во многом зависит от качества исходных данных. Технология Big Data учитывает эту характеристику и позволяет надежно работать с такими данными.

Value (ценность). Когда мы говорим о ценности данных, то подразумеваем их значимость с точки зрения прикладных задач. По расчетам IBS, только 1,5 % накопленных массивов данных имеет информационную значимость. Большое количество данных – это хорошо, но если они не представляют никакого интереса, то они бесполезны.

Со временем были предложены дополнительные определяющие характеристики Big Data [10–13], которые получили название «7V» и “10V”.

В научном мире принято считать, что большие данные начинаются с объемов в петабайты и с информационными потоками в 100 ГБ в сутки.

Классификация больших данных

Редактор журнала Web 2.0 Journal Дайон Хинчклифф (Dion Hinchcliffe) дал классификацию Big Data [14, 15], позволяющую соотнести технологию с результатом, который ждут от обработки Big Data. Хинчклиф делит подходы к Big Data на три группы: Fast Data (быстрые данные), их объем измеряется терабайтами-петабайтными; Big Analytics (большая аналитика) – петабайтные-экзабайтные данные и Deep Insight (глубокое понимание) – экзабайты-зеттабайты. Группы различаются между собой не только оперируемыми объемами данных, но и качеством решения задач по их обработке.

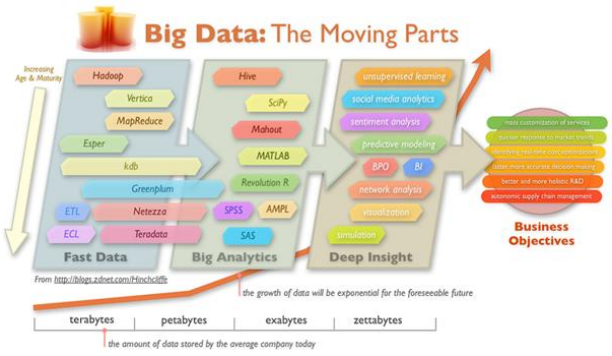
*Быстрые данные***.** Осознавая, что традиционные методы хранения, перемещения, обработки и выборки данных недостаточны, индустрия больших данных создала совершенно новый набор методов и адаптировала некоторые из существующих, которые позволили обрабатывать всю совокупность информации за приемлемое время. Обработка для Fast Data не предполагает получения новых знаний, ее результаты соотносятся с априорными знаниями и позволяют судить о том, как протекают те или иные процессы, она позволяет лучше и детальнее увидеть происходящее, подтвердить или отвергнуть какие-то гипотезы. Только небольшая часть из существующих сейчас технологий подходит для решения задач Fast Data, в этот список попадают некоторые технологии работы с хранилищами

(продукты Hadoop, MapReduce, Greenplum, Netezza, Oracle Exadata, Teradata, СУБД типа Verica и kdb). Скорость работы этих технологий должна возрастать синхронно с ростом объемов данных.

*Большая аналитика***.** Задачи, решаемые средствами Big Analytics, заметно отличаются, причем не только количественно, но и качественно, а соответствующие технологии должны помогать в получении новых знаний — они служат для преобразования зафиксированной в данных информации в новое знание. Однако на этом среднем уровне не предполагается наличие искусственного интеллекта при выборе решений или каких-либо автономных действий аналитической системы — она строится по принципу «обучения с учителем». Иначе говоря, весь ее аналитический потенциал закладывается в нее в процессе обучения. Классическими представителями такой аналитики являются продукты MATLAB, SAS, Revolution R, Apache Hive, SciPy Apache и Mahout.

Глубокое понимание. Мощных, но несфокусированных инструментов Big Analytics недостаточно, чтобы пожинать плоды больших данных. Deep Insight предполагает целенаправленное обучение без учителя (unsupervised learning) и использование современных методов аналитики, применяемых в конкретных областях, а также различные способы визуализации. На этом уровне возможно обнаружение знаний и закономерностей, априорно неизвестных. Методы глубокого проникновения позволят превратить всю информацию в оперативно действующий коллективный интеллект.

На рис. ниже приведена визуальная схема Дайона Хинчклиффа составляющих Big Data



Принципы работы

Исходя из определения Big Data, можно сформулировать следующие основные принципы работы с такими данными [16]:

*Распределенность***.** Хранить информацию в одном месте бессмысленно и практически невозможно. Поэтому технология работы с Big Data должна использовать распределенное хранение, управление, обработку и анализ данных, хранящихся в разнообразных хранилищах данных во всем мире.

*Горизонтальная масштабируемость***.** Поскольку данных может быть сколь угодно много – любая система, которая подразумевает обработку больших данных, должна быть расширяемой. В 2 раза вырос объём данных – в 2 раза увеличили кластер и всё продолжает работать с такой же производительностью.

*Отказоустойчивость***.** Принцип горизонтальной масштабируемости подразумевает, что машин в кластере может быть много. Например, Hadoop-кластер Yahoo имеет более 42000 машин. Это означает, что часть этих машин будет гарантированно выходить из строя. Методы работы с большими данными должны учитывать возможность таких сбоев и переживать их без каких-либо значимых последствий.

Локальность данных. В больших распределённых системах данные распределены по большому количеству машин. Если данные физически находятся на одном сервере, а обрабатываются на другом, то расходы на передачу данных могут превысить расходы на саму обработку. Поэтому одним из важнейших принципов проектирования BigData-решений является принцип локальности данных – по возможности обрабатываем данные на той же машине, на которой они хранятся.

Интерпретация данных в процессе их обработки (schema-on-read). Традиционные базы данных работают по приципу *schema-on-write* - сначала следует определить схему и только после этого можно вводить данные и работать с ними. В Big Data работа с данными может производиться по принципу *schema-on-read* - данные поступают в хранилище такими, как есть, без какого-либо

их предварительного описания, без указания их структуры и семантики. И только в процессе их выборки для обработки происходит их «осмысливание».

Все современные средства работы с большими данными так или иначе следуют этим пяти принципам.

Методы и технологии анализа и визуализации, применимые к Big Data

К настоящему времени создано и адаптировано множество методов и технологий для сбора, агрегирования, манипулирования, анализа и визуализации больших данных. Эти методы и технологии заимствованы из различных областей, включая статистику, информатику, прикладную математику и экономику. Это означает, что для извлечения выгоды из больших данных, следует использовать гибкий междисциплинарный подход. Некоторые методы и технологии были разработаны для оперирования значительно меньшими объемами и разнообразием данных, но были успешно адаптированы для Big Data. Другие были разработаны в последнее время, в частности, для сбора и анализа больших данных. В отчете [17] подразделения McKinsey Global Institute (MGI) международной аудиторско-консалтинговой компании McKinsey & Company приводятся методы и технологии анализа и визуализации, применимые к Big Data. Приведем их краткое описание.

Методы анализа Big Data

Методы класса Data Mining:

- *обучение ассоциативным правилам* (association rule learning) – это метод, базирующийся на правилах, используется для обучения машин способам обнаружения зависимостей между данными в больших базах данных;
- *классификация* – методы категоризации новых данных на основе принципов, ранее применённых к уже наличествующим данным;
- *кластерный анализ* – статистический метод классификации объектов, который приводит к разделению разнообразных групп на более мелкие группы подобных (сходных) объектов, для которых критерий подобия заранее не известен;
- *регрессионный анализ*.

Краудсорсинг (crowdsourcing) – метод сбора, категоризация и обогащение данных силами широкого круга лиц, привлечённых на основании публичной оферты, без вступления в трудовые отношения, обычно посредством использования сетевых медиа.

Слияние и интеграция данных (data fusion and integration) – набор методов, позволяющих интегрировать и анализировать разнородные данные из разнообразных источников для глубинного анализа более точно и эффективно, чем из единственного источника данных. В качестве примеров методов этого класса является цифровая обработка сигналов и обработка естественного языка.

Обучение ассоциативным правилам (association rule learning). Совокупность методов для анализа необходимых взаимосвязей, то есть «ассоциативных правил», среди переменных в больших базах данных.

Машинное обучение (machine learning). Класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач. Включает обучение с учителем (supervised learning) и без учителя (unsupervised learning), а также Ensemble learning – использование моделей, построенных на базе статистического анализа или машинного обучения для получения комплексных прогнозов на основе базовых моделей (constituent models).

Обработка естественного языка (Natural language processing – NLP). Общее направление искусственного интеллекта и математической лингвистики. Оно изучает проблемы компьютерного анализа и синтеза естественных языков. Применительно к искусственному интеллекту анализ означает понимание языка, а синтез – генерацию грамотного текста. Многие NLP-методы являются методами машинного обучения.

Искусственные нейронные сети (artificial neural networks). Математическая модель, построенная по принципу организации и функционирования биологических ней-

ронных сетей – сетей нервных клеток живого организма.

Сетевой анализ (network analysis). Набор методов, используемых для описания и анализа отношений между дискретными узлами в графе или сети. В анализе социальной сети анализируются связимежду людьми в сообществе или организации, например, как перемещается информация или кто имеет наибольшее влияние на кого.

Распознавание образов (pattern recognition). Набор методов машинного обучения, развивающих основы и методы классификации и идентификации предметов, явлений, процессов, сигналов, ситуаций и т. п. объектов, которые характеризуются конечным набором некоторых свойств и признаков.

Прогнозная аналитика (predictive analytics). Класс методов анализа данных, концентрирующийся на прогнозировании будущего поведения объектов и субъектов с целью принятия оптимальных решений.

Анализ тональности текста (sentiment analysis). Класс методов контент-анализа в компьютерной лингвистике, предназначенный для автоматизированного выявления в текстах эмоционально окрашенной лексики и эмоциональной оценки авторов (мнений) по отношению к объектам, речь о которых идёт в тексте.

Имитационное моделирование. (simulation modeling) – метод исследования, при котором изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему (построенная модель описывает процессы так, как они проходили бы в действительности), с которой проводятся эксперименты, с целью получения информации об этой системе.

Пространственный анализ (Spatial analysis) – набор методов, которые анализируют топологические, геометрические или географические свойства, представленные в наборе данных. Часто данные для пространственного анализа поступают из географических информационных систем (ГИС).

Статистический анализ, примеры: A/B-тестирование (контрольная группа элементов сравнивается с набором тестовых групп, в которых один или несколько показателей были изменены, для того, чтобы выяснить, какие из изменений улучшают целевой показатель) и анализ временных рядов.

Анализ временных рядов (timeseries analysis) – совокупность математико-статистических методов анализа, предназначенных для выявления структуры временных рядов и для их прогнозирования. Сюда относятся, в частности, методы регрессионного анализа. Выявление структуры временного ряда необходимо для того, чтобы построить математическую модель того явления, которое является источником анализируемого временного ряда.

Технологии и средства работы с Big Data

Существует множество технологий для агрегации, манипулирования, управления и анализа больших данных. Далее приводится список наиболее известных и используемых технологий и средств. Они приводятся в алфавитном порядке.

Big Table. Запатентованная распределенная система баз данных, построенная на основе Google File System.

Business intelligence (BI) (бизнес-аналитика). Совокупность методологий, процессов, архитектур и технологий, которые преобразуют большие объемы «сырых» данных в осмысленную и полезную информацию, пригодную для бизнес-анализа и для поддержки принятия оптимальных тактических и стратегических решений.

Cassandra. Свободно распространяемая система управления базами данных, предназначенная для манипулирования данными огромного объема в распределенных системах.

Cloud computing (облачные вычисления). Вычислительная парадигма, в которой высокомасштабируемые вычислительные ресурсы, обычно сконфигурированные в виде распределенных систем, предоставляются в сетях качестве сервисов.

Data Warehouse (хранилище данных). Предметно-ориентированная информационная база данных, специально разработанная и предназначенная для подготовки отчётов и анализа данных с целью поддержки принятия решений в организации и является одной из основных компонент бизнес-

анализа. Выступает центральным репозиторием данных, поступающих из различных источников. Хранит текущие и исторические данные. Строится на базе систем управления базами данных и систем поддержки принятия решений.

Distributed system (распределенная система). Множество компьютеров, взаимодействующих по сети и объединенных для решения общей вычислительной задачи.

Dynamo. Запатентованная распределенная система хранения данных, разработанная в Amazon.

Extract, transform, and load (ETL) (извлечь, преобразовать, загрузить). Технология, используемая для извлечения данных из внешних источников, преобразования их для удовлетворения операционных потребностей, и загрузки их в базу данных или хранилище данных.

Google File System. Запатентованная распределенная файловая система. На ее основе построен Hadoop.

Hadoop. Проект фонда Apache Software Foundation, свободно распространяемый набор утилит, библиотек ифрейм-ворк для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов. Используется для реализации поисковых и контекстных механизмов многих высоконагруженных веб-сайтов, в том числе, для Yahoo! и Facebook. Базируется на MapReduce и Google File System.

HBase. Свободно распространяемая распределенная нереляционная база данных, созданныя на основе Big Table Google.

MapReduce. Модель распределённых вычислений, представленная компанией Google, используемая для параллельных вычислений над очень большими, вплоть до нескольких петабайт, наборами данных в компьютерных кластерах. Эта модель реализована в Hadoop.

Mashup. Веб-приложение, объединяющее данные из нескольких источниковв один интегрированный, например, при объединении картографических данных Google Maps с данными о недвижимости с Craigslist получается новый уникальный веб-сервис, изначально не предлагаемый ни одним из источников данных.

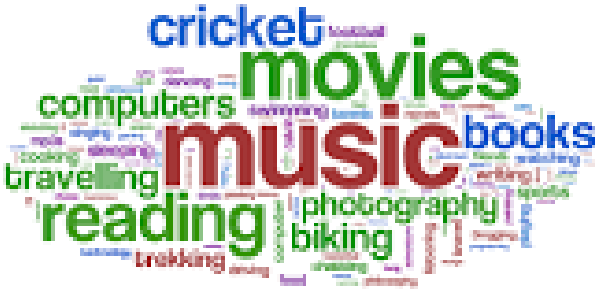
R. Свободно распространяемый язык программирования среда программирования для статистических и графических вычислений.

Stream processing. Технология, предназначенная для обработки больших потоков данных в реальном масштабе времени.

Визуализация Big Data

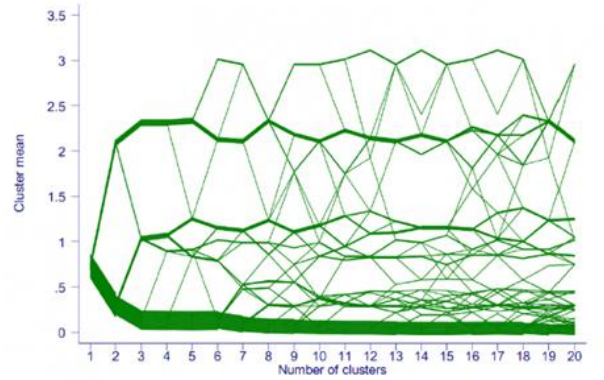
Наглядное представление результатов анализа больших данных таким образом, чтобы их можно было легко воспринимать, является ключевой проблемой анализа данных, имеет принципиальное значение для их интерпретации. Восприятие человека ограничено, и ученые продолжают вести исследования в области совершенствования современных методов представления данных в виде изображений, диаграмм или анимаций. В качестве иллюстрации приводим несколько прогрессивных методов визуализации, относительно недавно получивших распространение.

Облако тегов (Tag cloud). Каждому элементу в облаке тегов присваивается определенный весовой коэффициент, который коррелируется размером шрифта. В случае анализа текста величина весового коэффициента напрямую зависит от частоты употребления (цитирования) определенного слова или словосочетания. Позволяет читателю в сжатые сроки получить представление о ключевых моментах сколько угодно большого текста или набора текстов.

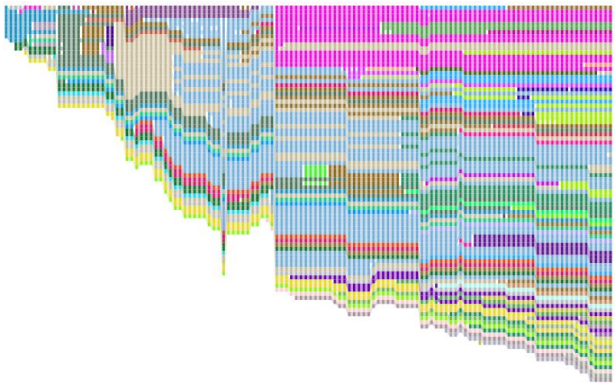


Кластерграмма (Clustergram). Метод визуализации, использующийся при кластерном анализе. Показывает, как отдельные элементы множества данных соотносятся с кластерами по мере изменения их количества. Выбор оптимального количества кластеров – важная составляющая кластерного анализа. Этот способ визуализации позволяет аналитику лучше понять, как ре-

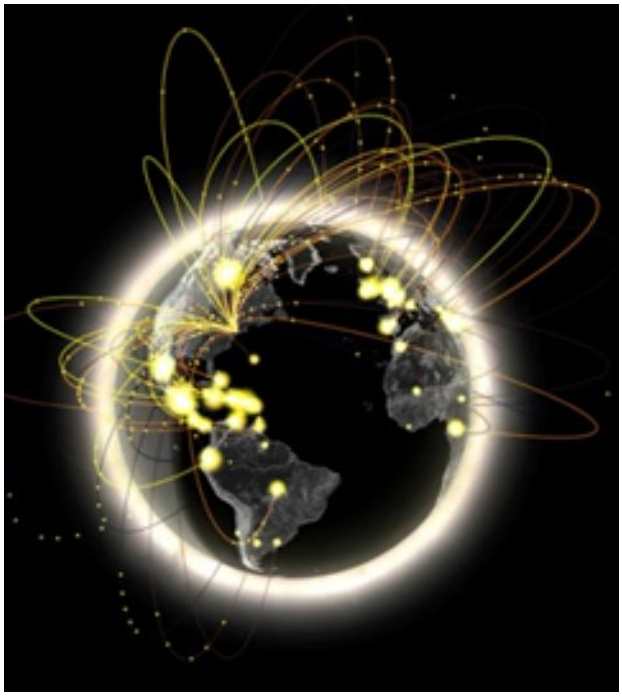
зультаты кластеризации изменяются по мере изменения количества кластеров.



Исторический поток (history flow). Помогает следить за эволюцией документа, над созданием которого работает одновременно большое количество авторов. В частности, это типичная ситуация для сервисов wiki. По горизонтальной оси откладывается время, по вертикальной – вклад каждого из соавторов, т. е. объем введенного текста. Каждому уникальному автору присваивается определенный цвет на диаграмме. Приведенная ниже диаграмма – результат анализа эволюции статьи «Afrotheria» в Википедии. Хорошо видно, как возрастала активность авторов с течением времени.



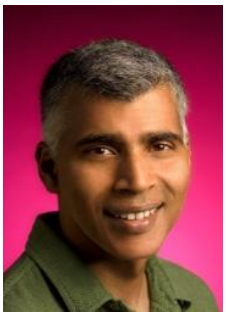
Пространственный поток (spatial information flow). Диаграмма этого вида позволяет отслеживать пространственное распределение информации. Приведенная ниже в качестве примера диаграмма построена с помощью сервиса New York Talk Exchange. Она визуализирует интенсивность обмена IP-трафиком между Нью-Йорком и другими городами мира. Чем ярче линия – тем больше данных передается за единицу времени. Таким образом, не составляет труда выделить регионы, наиболее близкие к Нью-Йорку в контексте информационного обмена.



Модель больших данных

Реляционная модель данных (РМД) не применима для больших данных. Ее структура строго формализована, в свою очередь большие данные могут быть слабоструктурированными или вообще не иметь структуры. РМД предполагает обязательное существование схемы, а большие данные могут быть бессхемными. Реляционная алгебра для больших данных абсолютно не применима. Проблема независимости данных вообще не ставится перед большими данными, поэтому классические архитектурные решения РМД в лице архитектуры ANSI/X3/SPARC не применимы. Концепция концептуальной информационной модели в больших данных отсутствует. Гордость РМД - теория зависимостей и нормальных форм абсолютно не применима, так как порождаемая декомпозиция становится губительной для больших данных и для них больше подходит концепция существования единого универсального отношения. Еще одна гордость РМД - принцип ACID для транзакций является дорогим, неэффективным и ненужным удовольствием в больших данных.

В связи с этим в больших данных применяют модели данных NoSQL. Наиболее используемой является модель ключ-значение. На этой модели определена модель вычислений MapReduce - модель распределенной обработки данных, предложенная компанией Google для обработки больших объёмов "сырых" данных на компьютерных кластерах (большого количества компьютерных узлов). MapReduce была разработана сотрудниками Google Джеффри Дин (Jeffrey Dean) и Санджай Гемават (Sanjay Ghemawat) [18]. Эта статья объемом в 6 страниц пользуется огромной популярностью. По состоянию на январь 2023 г. она была опубликована в двух источниках и на нее было сделано более 35500 ссылок.


Санджай Гемават

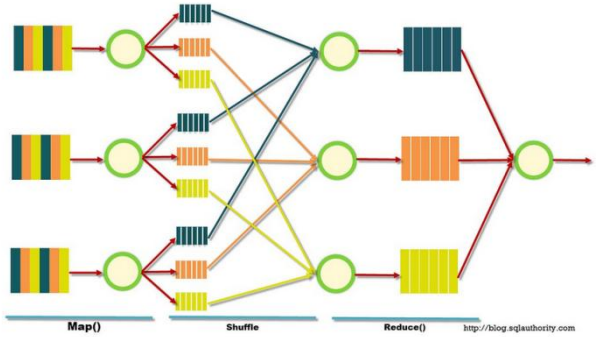

Джеффри Дин

Модель вычислений MapReduce предполагает выполнение трех этапов.

Этап Map - предварительная обработка и фильтрация входных данных в виде большого списка значений. При этом главный узел кластера получает этот список, делит его на части и передает рабочим узлам. Каждый рабочий узел применяет функцию Map к локальным данным и в результате выдаётся множество пар "ключ-значение". Что будет находиться в ключе и в значении – решать пользователю.

Этап Shuffle. Проходит незаметно для пользователя. На этой стадии на каждом рабочем узле на основе ключей, созданных функцией Map, «разбирается по корзинам» (сортируется) – каждая корзина соответствует одному ключу вывода стадии Map. Эти корзины послужат входом для Reduce.

Этап Reduce. Каждая «корзина» со значениями, сформированными на этапе Shuffle, попадает на вход функции Reduce. Функция Reduce задаётся пользователем и вычисляет финальный результат для отдельной «корзины». Множество всех значений, возвращённых функцией Reduce, является финальным результатом MapReduce-задачи.



Не смотря на простоту MapReduce, ее заслуга в том, что это архитектура, которая обеспечивает:

– автоматическое распараллеливание данных из огромного массива по множеству узлов обработки, выполняющих процедуры MapReduce;
– эффективную балансировку загрузки этих вычислительных узлов, не дающую им простаивать или быть перегруженными сверх меры;
– технологию отказоустойчивой работы, предусматривающую тот факт, что при выполнении общего задания часть узлов обработки может выйти из строя или по какой-либо другой причине перестать обрабатывать данные.

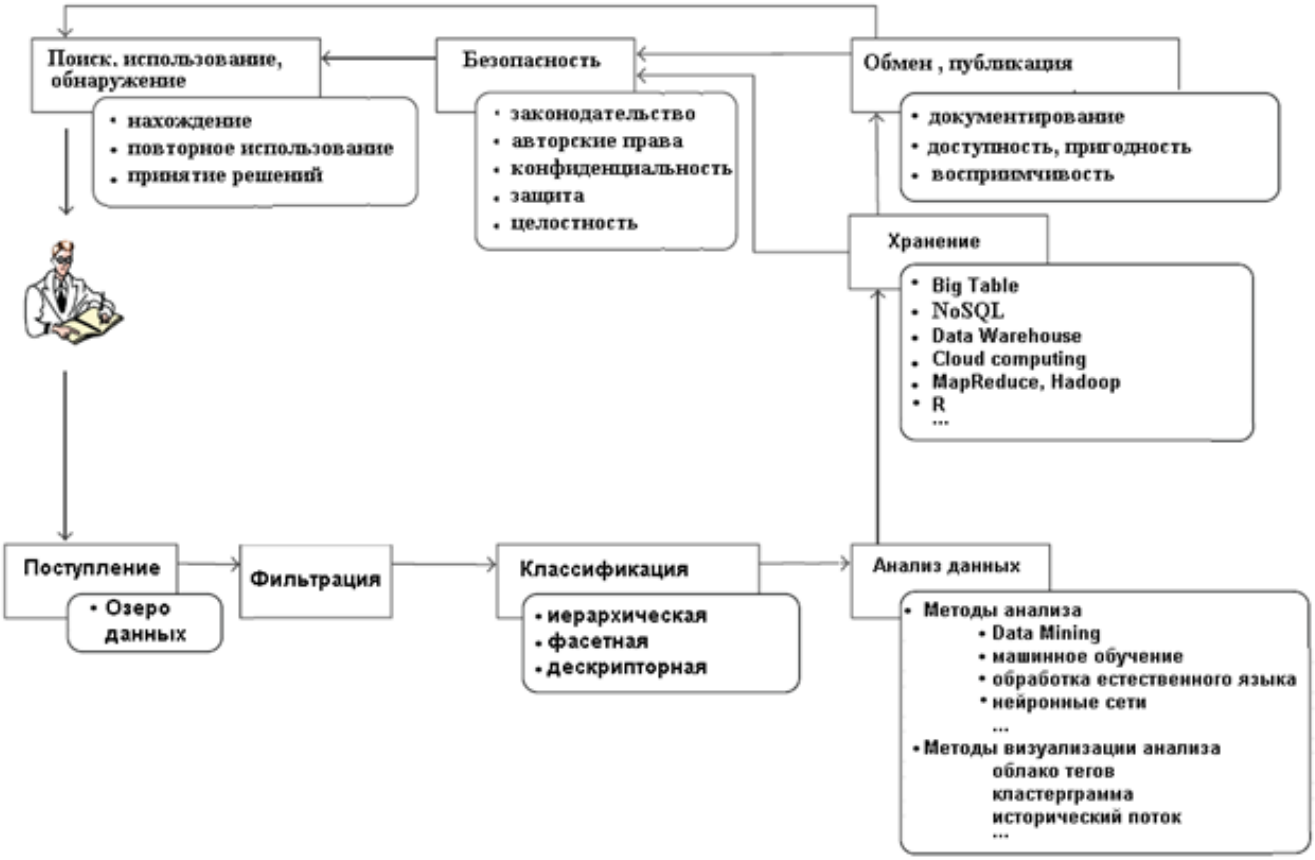
Жизненный цикл управления данными с использованием технологии Big Data

Опишем в общих чертах жизненный цикл управления данных, который использует технологию Big Data. Идея этого цикла взята из работы [19] Предлагаемый жизненный цикл данных состоит из следующих этапов: сбор, фильтрация и классификация, анализ данных, хранение, обмен и публикация, а также поиск и обнаружение данных. Далее кратко описывается каждый этап согласно приведенному ниже рисунку жизненного цикла.

Поступление данных

Поступление (сбор) данных – это первый этап жизненного цикла данных. Большое количество данных поступает из различных источников. Такими источниками могут быть: файлы журналов, которые ведутся на серверах, датчики различного вида,

мобильные устройства, данные, поступающие со спутников, результаты научных исследований, данные вычислительных экспериментов, результаты выполнения поисковых запросов, данные, порождаемые в социальных сетях, и многие другие. При сборе данных используются разнообразные методы получения исходных сырых данных из различных источников. Рассмотрим несколько методов сбора данных и используемые ими технологии



Файлы журналов (log-файлы). Этот метод используется для автоматической регистрации данных, связанных с различными событиями, происходящими в автоматизированных системах. Log-файлы используются практически во всех компьютерных системах, например, веб-сервера фиксирует все транзакции, выполняемые сервером. При наличии очень больших файлов журнала их информация запоминается в базах данных, а не в виде тестовых файлов.

Сенсорные данные (Sensor data.). Часто датчики используются длясъема физических характеристик, которые затем преобразуются в воспринимаемые цифровые сигналы для их сохранения и обработки. К сенсорным данным можно отнести, например, данные, которые поступают в виде звуковых, вибрационных, голосовых волн, результатов физических, химических, биологических, метеорологических или других видов исследований, результатов съема характеристик (показателей) производственных процессов.

Мобильные устройства. С помощью различные технологий, которые встраиваются в мобильные устройства, можно получать и передавать информацию географическом местоположении, воспринимать аудио- и видеоинформацию, делать фотографии, с помощью сенсорных экранов и гравитационных датчиков получать информацию о состоянии здоровья человека.

В результате сбора таких данных образуется так называемое озеро данных (Data lake). Это централизованное хранилище больших данных в сыром, необработанном виде. В нем хранят данные из разных источников, разных форматов, структурированные, слабо структурированные, неструк-

турированные и бинарные данные (изображения, аудио видео-данные). Они хранятся как правило, в несистематизированном виде такими, как есть, без какой либо предварительной обработки. Это обходится значительно дешевле традиционных хранилищ, в которые помещаются только структурированные данные. Data lake позволяют анализировать большие данные в исходном виде.

Фильтрация данных

В исходных данных может быть много шума.Так, например, при некачественной аудио-записи фоновый шум может быть настолько сильным, что не позволяет выделить полезную аудио-информацию с использование современных средств распознавания, или камера видео-наблюдения произвела съемку в темное время и изображение абсолютно черным. Фильтрация позволяет ибавиться от такой информации.

Классификация данных

Любые поступающие данные всегда обладают какой-то минимальной информацией. Например, известно, где именно установлена видео-камера, куда она направлена и к какому времени суток привязаны те или иные кадры, или что собой представляют поступающие научные данные, результатами какого эксперимента они являются, при каких условиях эксперимент проводился, и так далее. Таким образом, любые поступающие данные обладают так называемыми метаданными, которые можно использовать для проведения первоначальной классификации, которая являетсяпервоначальным шагом выявления семантики данных. Эта семантика служит хорошей основой для проведения последующего анализа данных.

Методы классификации – это совокупность приемов разделения множества-объектов на подмножества. В науке известны три метода классификации объектов: иерархический, фасетный, дескрипторный. Эти методы различаются разной стратегией применения классификационных признаков.

Иерархический метод. Это метод, при котором заданное множество последовательно делится на подчиненные подмножества, постепенно конкретизируя объект классификации. При этом основанием деления служит некоторый выбранный признак. Совокупность получившихся группировок при этом образует иерархическую древовидную структуру.

Фасетный метод. Подразумевает параллельное разделение множества объектов на независимые классификационные группы. При этом не предполагается жёсткой классификационной структуры и заранее построенных конечных групп. Классификационные группировки образуются путём комбинации значений, взятых из соответствующих фасетов.

Дескрипторный метод. Суть этого метода заключается в следующем: отбирается совокупность ключевых слов или словосочетаний, описывающих определенную предметную область или совокупность однородных объектов, они подвергаются нормализации, на основании этого создается словарь дескрипторов, который служит основой для проведения классификации.

Анализ данных

Анализ данных позволяет воспринять и обработать огромные объемы Big Data. Анализ данных является сложной задачей и во многом зависит от тех задач, которые надо решать с использованием этих данных, выдвигаемых требований к точности и скорости решения, наличия технических средств и, наконец, состояний исходных данных. Анализ данных включает решения следующих двух основных задач:

- на первом этапе должна быть решена задача раскрытия синтаксиса данных, то есть выявление структуры данных, например, какие объекты предоставляемые данные представляют, какими свойствами они обладают, что собой представляют значения этих свойств, каким образом взаимосвязаны объекты, какова природа и каковы характеристики этих связей;
- второй этап связан с раскрытием семантики данных. Это так называемый этап интеллектуального анализа данных (data mining). В разделе «Методы анализа Big Data» приводится краткое описание используемых методов.

Для гибкой организации анализа данных в работе [1465a] были предложены следующие три принципа:

- во-первых, для достижения поставленных целей следует использовать не единственный, а множество релевантных методов анализа;
- во-вторых, для хранения данных следует использовать различные методы и устройства хранения, которые могут быть распределены по компьютерам сети;
- в-третьих, следует предоставлять высокоэффективные методы и средства доступа и обработки данных.

Анализ данных производится с учетом следующих факторов: гетерогенность, точность и сложность данных, возможность их масштабирования.

Хранение, совместное использование, публикация

После сбора,очистки и анализа полученные данныезапоминаются в соответствующих хранилищах, к ним предоставляется доступ и/или они публикуются для ознакомления с ними широкого круга заинтересованных лиц. Большие по объему и интенсивно использующиеся наборы данных. Big Data должны храниться и управляться с большой степенью надежности, доступности и простоте использования. Инфраструктура хранения должна обладать достаточной степенью гибкости. Система хранения должна быть распределенной. Такая распределенная система хранения должно обеспечить поддержку целостности, обеспечение доступности, устойчивости к отказам различного вида.

Безопасность

Безопасность данных – это защита данных от несанкционированного (случайного или намеренного) доступа, изменения или разрушения. Сфера применения Big Data в современном мире практически не имеет границ. Раскрытие, изменение или разрушение данных в Big Data может иметь катастрофические последствия. При этом следует отметить, что все среды для работы с большими данными подвержены рискам. В связи с этим необходимо обеспечивать надежную защиту Big Data при их хранении, передаче и обработке за счет внедрения и использования процедур и технологических решений в области защиты информации.

Поиск, повторное использование, обнаружение

Поиск данных обеспечивает (гарантирует) качество данных, увеличение их значимости и сохранности посредством механизма повторного использования и сохранения с целью выявления новой более осмысленной информации. Сфера этой деятельности включает поиск, обнаружение, управление, аутентификацию, архивирование, сохранение и представление данных .После публикации данных другие исследователи должны иметь возможность аутентифицировать и регенерировать их в соответствии со своими интересами для проведения своих исследований. Возможность повторного использования опубликованных данных также должна быть гарантирована в научных сообществах. При многократном использовании определение семантики опубликованных данных является обычной ситуацией. Обычно эта процедура выполняется вручную. В Европейском Союзе активно поддерживается концепция открытой науки, например, инициированием Европейского облака открытой науки для обеспечения открытого доступа к результатам научных исследований из финансируемых государством проектов.

Литература

1) Cherniak L. Big Data — new theory and practice (Rus) // Otkrytye sistemy. SUBD. — 2011. — № 10. (Electronic resource): http://www.osp.ru/os/2011/10/13010990/
2) Clifford A. Lynch, "Big data: How do your data grow?" Nature, vol. 455, no. 7209 (September 3, 2008). October 2008 Nature 455(7209), p.28-29
3) Mashey J.R. (25 April 1998). "Big Data. and the Next Wave of InfraStress". - http://static.usenix.org/event/usenix99/invited_talks/mashey.pdf
4) Lohr S.(1 February 2013). "The Origins of 'Big Data': An Etymological Detective Story". The New York Times. Retrieved 28 September 2016.

5) Dutcher J.. What Is Big Data? https://datascience.berkeley.edu/what-is-big-data/
6) Zolotov O., Romanovskaya Y., Rzhannikova V. On Definition of BigData. - EPJ Web of Conferences 224, 04011 (2019)
7) Doug L. (2001. 3D Data Management: Controlling Data Volume, Velocity, and Variety. Technical Report 949, METAGroup (now Gartner). [Electronic resourse]: https://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf
8) Nawsher Khan, Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Zakira Inayat, Waleed Kamaleldin Mahmoud Ali, Muhammad Alam, Muhammad Shiraz, and Abdullah Gani1. Big Data: Survey, Technologies, Opportunities, and Challenges // Hindawi Publishing Corporation The Scientific World Journal Volume 2014, Article ID 712826, 18 pages.
9) Zikopoulos P., Parasuraman K., Deutsch T., Giles J., Corrigan D. (2013. Harness the power of big data The IBM big data platform. McGraw Hill Professional, New York, NY. - [Electronic resourse]: ftp://public.dhe.ibm.com/software/pdf/at/SWP10/Harness_the_Power_of_Big_Data.pdf
10) The Four V´s of Big Data. IBM (2011). - http://www.ibmbigdatahub.com/sites/default/files/infographic_file/4-Vs-of-big-data.jpg
11) Biehn N. The Missing V's in Big Data: Viability and Value. Wired (1 May 2013). - https://www.wired.com/insights/2013/05/the-missing-vs-in-big-data-viability-and-value/
12) McNulty E. Understanding Big Data: The Seven V's. Dataconomy (22 May 2014). - http://dataconomy.com/2014/05/seven-vs-big-data/
13) Firican G. The 10 Vs of Big Data. - https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx
14) 1461) Hinchcliffe D. Big Data, The Moving Parts: Fast Data, Big Analytics, and Deep Insight. - https://www.flickr.com/photos/dionh/7550578346/in/photostream/
15) Hinchcliffe D. The enterprise opportunity of Big Data: Closing the "clue gap". - https://www.zdnet.com/article/the-enterprise-opportunity-of-big-data-closing-the-clue-gap/
16) Big Data from A to Z. Part 1: Principles of working with Big Data, paradigm MapReduce. - https://medium.com/@DCA/big-data-from-a-to-z-part-1-principles-of-work-with-big-data-mapreduce-paradigm-84b47079d70e
17) James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Angela Hung Byers. Big data: The next frontier for innovation, competition, and productivity. McKinsey Global Institute, 2011. - https://personal.utdallas.edu/~muratk/courses/cloud11f_files/MGI-full-report.pdf
18) Dean J., Ghemawat S. MapReduce: simplified data processing on large clusters. Communications of the ACM, 2008, Vol. 51, No. 1. pp. 107-113
19) Begoli E., Horey J. Design principles for effective knowledge discovery from big data. In Proceedings of the 10th Working IEEE/IFIP Conference on Software Architecture (ECSA '12), 2012, p. 215–218

Исследования и разработки баз данных в Советском Союзе (1970-1991)

Невозможно описать исследования и разработки в области баз данных без охвата результатов, полученных в этой области в Советском Союзе. В начале этого столетия известный ученый-энциклопедист в области баз данных М.Р. Когаловский опубликовал фантастическую монографию «Энциклопедия технологий баз данных»[6], в которой имеется раздел «Отечественные исследования и разработки» в главе «Краткий очерк эволюции технологий баз данных», содержащий обширный материал по истории баз данных в Союзе. Весь последующий текст представляет собой существенно сокращенный и незначительно переработанный указанный выше раздел. Пришлось опустить огромную библиографию и многочисленные ссылки на нее и на другие разделы энциклопедии. По возможности ссылки заменены фамилиями авторов. Единственное, что добавлено самостоятельно, так это последний раздел, связанный с персоналиями.

Чтобы представить себе объем проведенных к тому времени исследований приведем фрагмент из монографии М.Р. Когаловского, которая содержит библиографию в 380 наименований: «... мы вынуждены в большинстве случаев обходиться здесь минимальным количеством библиографических ссылок, так как достаточно представительная библиография заняла бы сотни страниц. Даже весьма фрагментарная библиография, составленная по инициативе члена-корреспондента Академии наук СССР А.А. Стогния, которая была опубликована в 1984 г.[7], представляет собой издание объемом более 200 страниц».

[6] Когаловский М.Р. Энциклопедия технологий баз данных. М.: Финансы и статистика, 2002 г., 800 с.

[7] Банки данных и информационно-поисковые системы. Библиографический указатель. Киев: АН УССР. Институт кибернетики им. В.М. Глушкова, 1984. 231 с.

Организация и инфраструктура исследований и разработок

Несмотря на то что потребности и предпосылки развития исследований и прикладных разработок, связанных с технологиями баз данных, существовали в стране и ранее, активная деятельность в этой области развернулась лишь в начале 70-х гг. Именно в этот период в стране началось массовое производство вычислительных систем третьего поколения ЕС ЭВМ, обладающих дисковыми устройствами внешней памяти прямого доступа, без которой невозможно создание систем баз данных.

Первым крупным форумом заинтересованных в рассматриваемой области специалистов стала состоявшаяся осенью 1973 г. в Ташкенте Всесоюзная конференция по автоматизированным системам управления, которая проводилась в Институте кибернетики Узбекской Академии наук. На конференции работала специализированная секция «Банки данных». Конференция привлекла внимание к проблематике баз данных в стране. К этому времени в нескольких организациях уже велись разработки инструментального программного обеспечения.

Конференция выявила острую необходимость создания постоянно действующего научно-общественного организационного ядра отечественного сообщества специалистов в области баз данных. Эта функция была возложена на учрежденную Государственным комитетом по науке и технике в 1974 г. Рабочую группу по программному обеспечению банков данных (РГБД), впоследствии (в 1984 г.) реорганизованную в Научно-техническую комиссию ГКНТ по банкам данных. Эта комиссия функционировала до 1987 г.

Председателем РГБД в течение всего периода ее функционирования являлся Г.К. Столяров (Институт математики АН БССР), заместителями председателя — Л.А. Калиниченко (ИНЭУМ), В.М. Савинков (ВНИИПОУ) и А.А. Стогний (Институт кибернетики АН УССР), ученым секретарем — В.П. Дрибас (Институт математики АН БССР).

РГБД проводила Всесоюзные конференции по банкам данных и Всесоюзные

полугодичные семинары; создавала временные целевые и экспертные подгруппы; издавала «Отчёты РГБД» и методические материалы, руководила редакционным советом основного периодического сборника по СУБД и БД в СССР «Прикладная информатика» (главный редактор – Савинков В.М. (Москва), среди заместителей гл. редактора – Столяров Г.К. (Минск)); сотрудничала с международными профильными редакциями и рабочими группами USA CODASYL DBTG и British Computer Society DBAWG.

Позднее, в 1978 г. для развертывания и координации работ в рассматриваемой области в рамках Академии наук СССР была учреждена Комиссия по банкам данных и информационно-поисковым системам при Президиуме Академии наук, которая функционировала до 1991 г.

Председателем комиссии в течение всего этого периода являлся член-корреспондент Академии наук СССР А.А. Стогний, зам. председателя — В.И. Филиппов (ВЦ АН СССР), а ученым секретарем — Ф.И. Андон (СКБ программного обеспечения ИК АН Украины)

Совместными усилиями РГБД и Комиссии было организовано пять Всесоюзные конференции по банкам данных (1-я – в Тбилиси, 1980 г.; 2-я – в Ташкенте, 1983 г.; 3-я – в Таллине, 1985 г.; 4-я – в Калинине, 1989 г., 5-я – во Львове, 1991 г.), осуществлялось формирование государственных планов научных исследований по профилю этих организаций, разрабатывались методические материалы, проводилась экспертиза разработок крупных систем программного обеспечения.

Наконец, необходимо отметить здесь важную роль ряда отечественных издательств. Пионером в издании монографической литературы отечественных и зарубежных авторов по проблематике систем баз данных является издательство "Финансы и статистика". Значителен также вклад издательств "Наука " и "Мир".

Создание программного инструментария

Одним из необходимых условий практического использования технологий баз данных является оснащение организаций-разработчиков и пользователей приложений необходимым программным инструментарием, прежде всего системами управления базами данных. Как уже отмечалось, на начальном этапе развития технологий баз данных, в стране не существовало таких программных средств и возможности их приобретения за рубежом. Поэтому необходимо было осуществить самостоятельные разработки СУБД, несмотря на отсутствие опыта создания таких сложных программных систем.

Первые шаги в решении этой задачи относятся к началу 70-х гг., когда началось производство вычислительных машин семейства ЕС ЭВМ. Работы проводились в двух направлениях. Прежде всего были предприняты попытки создания собственных оригинальных отечественных СУБД. Вместе с тем, в ускоренном режиме разрабатывались аналоги некоторых широко распространенных за рубежом СУБД, способных функционировать на отечественных аппаратно-программных платформах. Подобный подход был использован также при создании СУБД для аппаратных платформ, серийное производство которых началось в стране позднее появления платформы ЕС ЭВМ, — для СМ ЭВМ, АСВТ, IBM-совместимых персональных компьютеров и др.

Вероятно, первым проектом в стране, направленным на создание оригинальной отечественной СУБД, соответствующей передовым достижениям международного уровня, была СУБД типа CODASYL НАБОБ для платформы ЕС ЭВМ, разработка которой началась в указанный период в ВГПТИ ЦСУ СССР.

Наряду с СУБД НАБОБ впоследствии были разработаны также и другие оригинальные отечественные системы. СУБД типа CODASYL ПАРМА для платформы ЕС ЭВМ с операционной системой ОС ЕС была создана НИИУМС (г. Пермь). В Вычислительном центре Академии наук была разработана СУБД типа CODASYL КОМПАС для платформы БЭСМ-6 с операционной системой ДИСПАК.

В Институте проблем управления были начаты работы по реализации СУБД ие-

рархического типа для платформы ЕС ЭВМ, которые были продолжены в ВНИИСИ и завершились созданием системы ИНЕС. Эта система имела большое число пользовательских установок. В Институте кибернетики АН УССР было создано семейство совместимых реляционных СУБД ПАЛЬМА для платформ ЕС ЭВМ, СМ ЭВМ и IBM-совместимых персональных компьютеров. В Министерстве легкой промышленности Латвии была разработана развитая реляционная СУБД ВЕРА. Институт системного программирования РАН создал мобильный SQL-сервер на платформе UNIX и в качестве средства свободно распространяемого программного обеспечения передал его вместе с исходными текстами в консорциум Free Software Foundation. Институтом системного анализа РАН разработана мультимедийная СУБД НИКА для персональных компьютеров. Следует упомянуть также, созданное в Воронежском СКТБ «Системпрограмм», семейство совместимых реляционных СУБД ИНТЕРЕАЛ для различных программно-аппаратных платформ и,

Среди СУБД и других средств программного обеспечения систем баз данных, созданных в стране и имеющих зарубежные аналоги, наиболее широкое распространение получили: СУБД СИНБАД (МНИПИ АСУ ГХ); СУБД ОКА и телемонитор КАМА Института кибернетики АН УССР; СУБД ДИСОД, разработанная НИИ «Восход»; система БАНК Пермского НИИУМС; созданная Советско-болгарским институтом ИНТЕРПРОГРАММА в Софии система СЕДАН; система РЕБУС Всесоюзного научно-исследовательского института непромышленной сферы и ряд других разработок. Нужно заметить, что некоторые из перечисленных систем, например, ОКА и ДИСОД, имели чрезвычайно развитое окружение, функционально значительно более богатое, чем у систем-прототипов.

Разработка приложений

Главной сферой применения технологий баз данных в нашей стране в период 70-х — 80-х гг. являлись автоматизированные системы управления различного уровня в экономике. Разрабатывались такие крупнейшие уникальные системы макроуровня, как Автоматизированная система плановых расчетов (АСПР) Госплана страны и плановых органов республик и Автоматизированная система Государственной статистики (АСГС). Несколько позднее СУБД стали неотъемлемым компонентом программного обеспечения многочисленных отраслевых систем управления.

Однако наиболее массовой сферой применения были автоматизированные системы управления предприятиями. Типовую архитектуру таких систем и комплекс типовых приложений разработал институт «Центрпрограммсистем» (г. Калинин). Этот инструментарий использовался на практике многочисленными промышленными предприятиями страны.

Активную поддержку деятельности в указанном направлении оказывал международный Советско-болгарский институт «Интерпрограмма» (София), которым было создано разнообразное типовое программное обеспечение, получившее широкое распространение в обеих странах.

В 80-е гг. на основе технологий баз данных был создан ряд информационных систем центральных организаций различных ведомств — патентной службы, Госстандарта, Высшей аттестационной комиссии, Всесоюзного научно-технического информационного центра и др. СУБД начали использоваться для создания информационных систем на транспорте и в строительстве, в крупнейших государственных библиотеках, в системах управления сложными техническими системами и во многих других областях. Однако все эти разработки были доступны лишь крупным организациям, способным содержать в своей структуре центры обработки данных.

Радикальное изменение ситуации произошло во второй половине 80-х гг., когда в стране стали появляться персональные компьютеры. Даже весьма скромные по своим функциональным возможностям и чрезвычайно простые в использовании СУБД, созданные для этой быстро прогрессирующей аппаратной платформы, дали возможность применять простейшие технологии баз данных в системах обработки данных для удовлетворения информационных потребностей

практически в любой области жизнедеятельности.

Некоторые дополнительные сведения о развитии приложений технологий баз данных в стране в 70-е — 80-е гг. можно найти в обзорах[8].

Научные исследования в области систем баз данных

Исследования, связанные с разработками новых СУБД. Фактор абсолютной новизны проблемы для отечественных специалистов требовал проведения исследований на многих этапах реализации первой отечественной СУБД типа CODASYL — НАБОБ. При создании системы ИНЭС, ставшей прототипом СУБД ИНЕС, исследовались методы доступа и хеширования, разработана древовидная структура индекса с расщепляющимися блоками. В ИНЕС впервые среди иерархических СУБД применена идея самоописываемости баз данных, предложенная ранее для реляционных систем. На основе опыта реализации системы КОМПАС ее авторами была предложена интегрированная реляционно-сетевая модель данных.

При создании SQL-сервера Института системного программирования был обобщен и эффективно использован опыт реализации пионерских исследовательских прототипов реляционных СУБД — System R и Ingres, воплощены концепции открытых систем и мобильности программного обеспечения, использованы некоторые принципы объектного подхода. Создатели системы ПАЛЬМА использовали в своем проекте принципы многоуровневой архитектуры СУБД и технику отображения моделей данных.

Развитие теории реляционных баз данных. Проблемы математической теории реляционных баз данных вызвали в стране столь же значительный интерес, как и за рубежом. Им были посвящены многочисленные исследования, выполненные в основном в 70-е — 80-е гг.

Наибольшее число работ этого направления было связано с исследованиями в области теории зависимостей и теории нормализации, с оценкой возможностей реляционных языков, с вопросами эквивалентности реляционных баз данных, с алгебраическими аспектами реляционной модели данных. Исследовались также аксиоматические подходы в области реляционной модели, формальные методы синтеза схем и логического проектирования реляционных баз данных, взаимосвязь логики и реляционной модели, вопросы вычислимости реляционных запросов. Большое внимание привлекали проблемы неполноты информации в реляционных базах данных.

Моделирование данных. Отечественные работы в этой области начались еще в 70-е гг. К этому направлению относятся, в частности, исследования, связанные с созданием канонической модели данных для систем интеграции неоднородных баз данных [Л.А. Калиниченко и др.] и моделей данных концептуального уровня архитектуры мультимодельной многоуровневой СУБД [М.Р. Когаловский и др.]. Некоторыми авторами вводятся различного рода расширения реляционной модели.

В середине 70-х гг. в языках программирования сформировалась концепция абстрактного типа данных, которая оказала влияние на дальнейшее развитие подходов в области моделирования данных [А.В. Замулин и др.].

Более мощные модели потребовались в системах интеграции неоднородных информационных ресурсов. Одна из моделей такого рода определяется языком Синтез [Л.А. Калиниченко].

Исследованию логико-математических основ моделирования данных посвящены работы В.И. Филиппова, В А. Крахта, М Ш. Цаленко.

Отображение моделей данных. В связи с разработками распределенных систем баз данных, систем интеграции неоднородных баз данных и СУБД с многоуровневой архитектурой, в том числе мультимодельных систем, возникли проблемы отображения моделей данных. Их решению

[8] Перевозчикова О.Л., Ющенко Е.Л. Тенденции развития систем обработки данных // Программирование. 1977. № 5. С. 70-90.

Dale A.G. Database Management Systems Development in the USSR Computing Surveys, Vol. 11, No. 3, 1979, pp. 213-226

были посвящены исследования отечественных авторов, направленные на создание методов преобразования моделей данных и конструирования коммутативных отображений [Л.А. Калиниченко], разработку архитектурных аспектов отображения моделей данных [М.Р. Когаловский] и спецификаций определения отображений для конкретных моделей данных. [Р.П. Крамаренко, А.Л. Виллемс]

СУБД с мультимодельным внешним уровнем. В отечественных исследовательских проектах, связанных с разработками мультимодельных СУБД, использовалось два подхода. В первом из них [М.Р. Когаловский. М.М. Виноградов и др.] роль концептуальной модели данных играет функционально достаточно развитая модель, обеспечивающая возможности отображения широко распространенных моделей.

Второй подход ориентировался на новые достижения в языках программирования. При этом концептуальная модель, строго говоря, не фиксируется в системе. В системе программирования баз данных АТЛАНТ [А.В. Замулин] предусматривается возможность ее спецификации как некоторой системы типов данных, определяемых средствами инструментальной системы пользователем. Аналогичный подход фактически применяется в [Х.-М.Х Хаав], где инструментальной системой служит система программирования ПРИЗ, на основе которой реализована СУБД DABU.

Управление конкурентным доступом. Среди ранних отечественных публикаций в области управления конкурентным доступом в системах баз данных можно назвать прежде всего работу [Оленин М.В. и др.], в которой предложена и исследована модель параллельных транзакций для распределенной объектной среды. Заслуживает также упоминания осуществленная в рамках проекта свободно распространяемого мобильного SQL-сервера реализация метода сериализации транзакций, основанного на двухфазном протоколе предикатных блокировок [С.Д. Кузнецов и др.].

Г.Г. Домбровской исследовалась техника поддержки вложенных транзакций и транзакций других типов на уровне механизмов управления буферизацией в среде хранения базы данных Она же показала, что включение некоторой дополнительной информации в дерево активных транзакций позволяет существенно расширить область применения рассматриваемой техники управления транзакциями.

Оптимизация запросов в системах баз данных. Следует отметить фундаментальный аналитический обзор[9] и отдельные статьи С.Д. Кузнецова, а также обзор В.И. Задорожного[10] по оптимизации рекурсивных запросов в системах дедуктивных БД.

Системы программирования баз данных и знаний. Идеи создания языков программирования, которые обеспечивали бы единую эффективную среду как для разработки приложений, так и для управления данными, были впервые высказаны А.В. Замулиным. Система программирования с входным языком Бояз была реализована на платформе БЭСМ-6 и использовалась в некоторых организациях. Группой А.В. Замулина был выполнен большой комплекс исследований в области языков программирования баз данных, разработан и реализован более современный (по сравнению с Бояз) язык Атлант.

Аналогичный весьма интересный подход в области создания единой среды языка программирования и базы данных был предложен позднее, в конце 70-х гг. в Институте кибернетики АН ЭССР. Средствами системы программирования высокого уровня ПРИЗ, которую ее идеолог Э.Х. Тыугу квалифицирует как инструмент концептуального программирования, можно не только программировать приложения, но и описывать, а также поддерживать на стадии исполнения нужную модель данных для этого приложения. Таким образом авторами была реализована, например, СУБД DABU.

Машины баз данных. Первые отечественные исследования в этой области относятся к концу 70-х — началу 80-х гг. Но-

[9] Кузнецов С.Д. Методы оптимизации выполнения запросов в реляционных СУБД // Сб Итоги науки и техники. Вычислительные науки.- Т. 1. - М.: ВИНИТИ, 1989. - С. 76-145

[10] Задорожный В.И. Методы вычисления и оптимизации рекурсивных запросов в дедуктивных базах данных. Препринт докл. // V Всесоюз. конф. "Системы баз данных и знаний". Львов, 1991. 47 с.

вый всплеск разработок в этой области был связан с учреждением в стране в середине 80-х гг. программы НИР по созданию средств вычислительной техники нового поколения

В монографии Л.А. Калиниченко[11] представлены результаты проведенного в Институте проблем информатики РАН комплексного и следования представления данных и знаний в машинах баз данных, их архитектуры и методов эффективной аппаратной реализации.

Объектные базы данных. В ранней работе С.Д. Кузнецова анализируются важнейшие принципы объектного подхода и концепции объектных СУБД. При этом уделяется особое внимание аспектам моделирования данных, языкам запросов в таких системах и оптимизации объектных запросов. Принципы отображения развитых объектных моделей исследовались в рамках проекта СИНТЕЗ Института проблем информатики РАН.

Дедуктивные базы данных. К числу ранних исследований, выполненных в этом направлении, можно отнести разработки Института прикладной математики Академии наук, в результате которых была создана действующая система «Вопрос-ответ» [Э.З. Любимский и др.]. Большой цикл теоретических исследований в области дедуктивных баз данных выполнен совместно М.И. Дехтярем А.Я. Диковским. Основательный анализ и классификация известных методов вычисления и оптимизации рекурсивных запросов в системах дедуктивных баз данных содержится в работах В.И. Задорожного. Ему принадлежат также другие результаты в области языков запросов и оптимизации в дедуктивных базах данных.

Распределенные базы данных. В 70-е гг. в стране активизировались работы по созданию вычислительных сетей. При этом в стиле, вполне адекватном централизованному характеру управления экономикой и другими сферами жизнедеятельности советского государства, была поставлена масштабная задача создания Государственной сети вычислительных центров (ГСВЦ). В проекте такой сети предусматривалось и создание функционирующих в ее среде распределенных баз данных. Основные исследования в этом направлении развернулись в научных учреждениях Москвы (ИНЭУМ, ИПМ, ВГПТИ ЦСУ СССР), Киева (ИК АН УССР), Риги (ИЭВТ АН Латвийской ССР).

Публичные обсуждения научно-технических проблем распределенных баз данных начались в 1975 г., когда в Институте кибернетики АН УССР состоялся семинар "Принципы построения РАБД государственной сети ВЦ". Через год более широкий семинар по этим проблемам организовала в Паневежисе РГБД совместно с Институтом физики и математики АН Литовской ССР.

Одно из направлений исследований было посвящено разработке математических моделей, позволяющих оптимизировать организацию и функционирование систем распределенных баз данных [Е.М. Бениаминов и др.].

Важная проблема — организация неоднородных распределенных баз данных с возможностями интеграции данных — рассматривалась в соответствии с концепциями исследовательского проекта СИЗИФ, который выполнялся в этот период в ИНЭУМ.

Уже на раннем этапе исследований разрабатывались конкретные инструментальные средства для создания распределенных баз данных. В качестве примера можно сослаться на гибридную СУБД БАЗИС (ИНЭУМ), поддерживающую интегрированные базы данных с фактографическими и текстовыми данными.

Интеграция информационных ресурсов. Исследования в области интеграции информационных ресурсов начались в нашей стране в середине 70-х гг. в рамках работ по созданию распределенных баз данных.

Наиболее ярким представителем отечественных разработок этого периода, без сомнения, является пионерский проект СИЗИФ Института электронных управляющих машин. В проекте была разработана архитектурная концепция системы интеграции неоднородных баз данных, основанная на некоторой интегрирующей канонической

[11] Калиниченко Л.А., Рывкин В.М. Машины баз данных и знаний. М.: Наука; Гл. ред. физ.-мат. лит., 1990. 296 с.

модели, обеспечивающей единое представление данных для всех включаемых в систему баз данных. Это единое представление - схема виртуальной базы данных — описывается с помощью специального языка. Был предложен также основанный на логике предикатов язык манипулирования данными, представленными в терминах этой схемы.

Авторами проекта был разработан, кроме того, метод построения коммутативных отображений моделей данных, обеспечивающий поддержку соответствия между данными интегрируемых баз данных и данными виртуальной базы данных. Применение этого метода было продемонстрировано на примере отображения сетевой модели данных CODASYL в реляционную модель.

В ряде публикаций по материалам проекта были показаны возможности использования языка Синтез для однородного описания информационных ресурсов, представленных средствами разнообразных моделей структурированных и слабоструктурированных данных.

Проектирование баз данных и разработка приложений. Едва ли не самая популярная сфера исследований и разработок в области технологий баз данных связана с проблемами проектирования систем баз данных, решение которых имеет весьма важное значение для обеспечения эффективного практического использования этих технологий.

Пик активности отечественных исследований в этой области пришелся на 80-е гг. Не случайно на Второй Всесоюзной конференции «Банки данных» для обсуждения проблем проектирования баз данных была организована специальная секция. В этот период в различных научных центрах страны над указанной проблематикой успешно работало несколько групп исследователей. Конечно же, весьма привлекательным было направление, связанное с формальными методами синтеза схем реляционных баз данных. Однако разрабатывались и иные подходы, главной целью которых являлось создание средств инфологического моделирования предметной области системы базы данных и отображения его результатов в среду конкретных СУБД.

Одно из направлений этих работ было связано с созданием «инженерной» методики проектирования концептуальной схемы базы данных в терминах, близких к ER-модели, и преобразования ее в схему базы данных избранной проектировщиком СУБД [В.В. Бойко и др.].

Более формализованный подход с использованием специально разработанных развитых средств инфологического моделирования был предложен группой исследователей из ВНИПИ АСУ Газпрома и ВНИИПОУ ГКНТ. По замыслу авторов это исследование должно было стать теоретическим базисом автоматизированной системы проектирования баз данных. Был разработан прототип такой системы — Омега-1.

Другой, также нацеленный на автоматизированную технологию проектирования подход, был предложен в ИК УССР. Разработанная модель для описания предметной области поддерживает иерархию абстракций различного рода. На ее основе создан язык описания концептуальных схем. Реализован прототип системы ПРОБАД, базирующийся на предложенном подходе.

В работах Г.И. Фурсина и др. главные цели заключались в создании концептуальной модели данных высокого уровня, основанной на исчислении предикатов, технологии ее использования, а также инструментария для поддержки процесса моделирования предметной области системы базы данных ее средствами.

Представляет интерес подход В.М. Ветошкина и др., в котором источником информации для формализованного процесса синтеза схемы реляционной базы данных служит вербальное описание предметной области. Разработан также метод синтеза схемы базы данных, оптимальной относительно введенного автором критерия сложности.

Наряду с указанными подходами развивалось направление, связанное с моделированием семантики предметной области средствами, используемыми в системах представления знаний [М.Ш. Цаленко, Э.Х. Тыугу, М.И. Кахро и др.].

Выбор и оценка СУБД. Проблемы выбора СУБД для конкретных приложений или для приложений в некоторой специфи-

ческой предметной области, а также для оценки характеристик их функционирования злободневны на всех стадиях развития технологий, когда речь идет о разработках крупных систем и систем с критическими требованиями к производительности, ресурсам памяти, надежности.

В отечественных разработках систем баз данных делались попытки определения совокупности факторов, которые могут стать основой выбора и оценки СУБД для конкретного приложения. Проводился сравнительный анализ характеристик различных СУБД, предлагались методики оценки и выбора СУБД для конкретных приложений.

Предпринимались также попытки оценки характеристик функционирования СУБД с помощью методов имитационного моделирования [Г.К. Столяров, О.М. Вейнеров и др.]. Применение техники имитационного моделирования не получило, однако, дальнейшего развития. Вероятно, одна из причин состоит в том, что получаемые с помощью дорогостоящих имитационных моделей оценки оказываются весьма грубыми. Более эффективными оказались подходы, основанные на использовании средств сбора статистики функционирования, которыми оснащены современные СУБД. Для оценки производительности СУБД в среде некоторых типовых приложений консорциумом TPC разработаны эталонные тесты. В период, когда этот консорциум еще не был учрежден, близкий подход использовался в исследованиях Центрпрограммсистем, связанных с получением сравнительных оценок производительности промышленно-сопровождаемых СУБД.

Персоналии

Приводится небольшой список лиц, принимавших активное участие в развитии баз данных в Советском Союзе, с которыми автор статьи был знаком по совместной работе в РГБД, либо зачитывался их монографиями и статьями. Заранее приношу извинения многим из тех, кто внес существенный вклад в развитие баз данных, но не приведен в этом разделе. Персональные сведения приведены в алфавитном порядке.

Андон Филипп Илларионович

Академик НАН Украины, доктор физико-математических наук, заслуженный деятель науки и техники Украины. Лауреат государственных премий в области науки и техники УССР и Украины, премий Совета Министров СССР, премий НАН Украины им. В.М. Глушкова и имени С.А. Лебедева.


Андон Ф.И.

Член РГБД, член программных комитетов Первой и Второй Всесоюзных конференций "Банки данных". Ученый секретарь Комиссии по банкам данных и информационно-поисковым системам Координационного комитета Академии наук СССР по вычислительной технике.

Подготовил 11 кандидатов и 5 докторов наук. Опубликовал более 200 научных работ, в том числе 5 монографий.

Под его руководством разработано ряд систем общегосударственного и отраслевого уровня. Он был главным конструктором систем ИНФОР и ЮПИТЕР, а также научным руководителем систем ПАЛЬМА, ОКА, КАМА.

Дрибас Виктор Прокофьевич

Сотрудник Института математики АН БССР. Секретарь РГБД ГКНТ на протяжении всего времени его существования

В 70-х годах, опубликовал весьма популярные в то время препринт по реляционной модели данных[12], а его монография по

[12] Дрибас В.П., Курскова Г.Л., Столяров Г.К., и др., Введение в реляционные модели базы данных.

реляционной модели баз данных[13] на притяжении многих лет пользовалась заслуженным авторитетом.

Он также занимался моделированием данных с многозначной классификацией объектов, а также рекомендациями по выбору баз данных.

Замулин Александр Васильевич (1943-2006)

Ученик А.П. Ершова, доктор физико-математических наук, профессор, главный научный сотрудник Института систем информатики имени А.П. Ершова СО РАН, зав. кафедрой Новосибирского государственного университета.

А.В. Замулин активно работал в области информационно-поисковых систем и систем управления базами данных. Он возглавлял создание информационно-поисковой системы общего назначения Вега для ЭВМ БЭСМ-6, одной из лучших на то время ИПС в нашей стране.


Замулин А,В.

Он по праву считается одним из основателей в стране нового научного направления - создание систем программирования баз данных. Под его руководством был разработан первый в мире язык программирования баз данных БОЯЗ (1976) и основанная на нем система программирования баз данных БОЯЗ-6 (1979); язык программирования баз данных Атлант (1986) и основанная на нем система программирования баз данных (1989); язык спецификаций баз данных Руслан (1994), который нашел признание за рубежом.

А.В. Замулин опубликовал более 100 работ, в том числе 2 монографии, посвященные типам данных в языках программирования и базах данных[14] и стемам программирования баз данных[15].

Являлся сопредседателем РГБД и членом Комиссии по банкам данных Координационного комитета АН СССР по вычислительной технике.

А.В. Замулин был членом редколлегии отечественного журнала "Программирование" и международных журналов "Information systems", "Universal Computer Science", "The Computer Journal", членом редколлегии периодического сборника статей "Системная информатика".

Калиниченко Леонид Андреевич (1937-2018)

Доктор физико-математических наук, зав. лаб. Института проблем информатики РАН, профессор ВМК МГУ, лауреат Государственной премии СССР в области науки и техники, заместитель председателя РГБД.

Область научных интересов — методы интеграции неоднородных баз данных, и управления распределенными базами данных.


Калиниченко Л.А

Весьма содержательный обзор[16] и монографии по интеграции неоднородных баз данных[17] и машинам баз данных[18] не утратили полезности и цитируются до настоящего времени. Его научные исследования были воплощены в системах БАЗИС и СИЗИФ., языке СИНТЕЗ

Подготовил 10 кандидатов наук.

Минск: Препринт/ инс-т математики АН БССР; № 4(20), 1977, 54 с.

[13] Дрибас В.П. Реляционные модели баз данных. Минск : БГУ, 1982, 192 с.

[14] Замулин А.В. Типы данных в языках программирования и базах данных. Новосибирск: Наука, Сибирское отд-ние, 1987. 152 с.

[15] Замулин А.В. Системы программирования баз данных и знаний. Новосибирск: Наука, Сибирское отд-ние, 1990. 352 с.

[16] Калиниченко Л А. и др. Архитектура и алгоритмы систем управления распределенными базам и данных / Л.А. Калиниченко, О.Е. Костромина, О.Н. Хитрова. М.: ИНЭУМ, 1982. 140 с.

[17] Калиниченко Л А. Методы и средства интеграции неоднородных баз данных. М.: Наука, Гл. ред. физ.-мат. лит., 1983, 424 с.

[18] Калиниченко Л А., Рывкин В.М. Машины баз данных и знаний. М.: Наука; Гл. ред. физ.-мат. лит., 1990, 296 с.

Член редколлегии журнала «Distributed and parallel databases»

Основатель и председатель (1992—2018) московской секции ACM SIGMOD.

Когаловский Михаил Рувимович

Учёный в области баз данных и информационных систем, кандидат технических наук, старший научный сотрудник, доцент, член редколлегий журналов «Программирование», «Информационное общество», «Электронные библиотеки», профессиональный член ACM, ученый секретарь Московской секции ACM SIGMOD, ведущий научный сотрудник Института проблем рынка РАН, доцент МГУ.


Когаловский М.Р.

Научный редактор и переводчик русских изданий монографий по базам данных Джеффри Ульмана, Кристофера Дейта, Алана Саймона, спецификаций языка определения данных CODASYL, а также знаменитого отчета ANSI/X3/SPARC.

Член рабочей группы по программному обеспечению банков данных (РГБД) при Госкомитете по науке и технике всё время её существования (1974—1987). Член и сопредседатель Программных комитетов ряда крупных международных и отечественных научных конференций, имеет более 200 печатных работ, в том числе 6 монографий. Его монография «Энциклопедия технологий баз данных»[19] оценивается специалистами как «фантастически тяжелый труд, который реально закрывает дыру в литературе, посвященной базам данных», а издание книги оценивается как исключительно полезное как для отечественных специалистов, так и для мировой общественности[20].

[19] Когаловский М.Р. Энциклопедия технологий баз данных. М.: Финансы и статистика, 2002 г., 800 с.

[20] Кузнецов С.Д. «Энциклопедия технологий баз данных» Михаила Рувимовича Когаловского. - http://www.citforum.perm.ru/book/enctbd/enctbd_vv.shtml

Пасичник Владимир Владимирович

Доктор технических наук, профессор Национального университета "Львовская политехника".

Воспитанник научной школы Института кибернетики имени В. М. Глушкова. Участник и руководитель многих международных научных проектов и перспективных научно-исследовательских разработок.

Автор 14 монографий и учебных пособий, среди которых особо выделяется монография[21], в которой исследуются вопросы реляционной модели баз данных, теории зависимостей и нормальных форм. Лауреат Государственной премии Украины в области науки и техники, отличник образования Украины.


Пасичник В.В.

Работал ведущим экспертом по технологиям баз данных и знаний ГКНТ СССР и стран - членов Совета экономической взаимопомощи.

Подготовил более двух десятков кандидатов и докторов наук в области баз данных и знаний, информационного анализа и современных информационных технологий.

Савинков Владимир Макарович

Зам. директора по научной работе ВГПТИ ЦСУ СССР, Москва. Заместитель председателя РГБД. Член организационных и программных комитетов Всесоюзных конференций «Банки данных».

Ответственный редактор сборников "Алгоритмы и организация решения экономических задач" и "Прикладная информатика". В то время они были наиболее авторитетными периодическими изданиями в стране, публиковавшими работы по тематике систем баз данных и информационных систем. Соавтор учебника по


Савинков В.М.

[21] Пасичник В.В., Стогний АА. Реляционные модели баз данных. Киев : ИК АН УССР, 1983. 286 с.

Алголу[22], толкового словаря по информатике[23] и монографии по проектированию баз данных[24]. Редактор перевода с английского широко известной монографии Чарльза Мидоу[25].

Стогний Анатолий Александрович (1932-2007)

Доктор физико-математических наук, профессор, член-корреспондент АН СССР и член-корреспондент НАН Украины, заместитель директора Института кибернетики НАН Украины, директор Института прикладной информатики (г. Киев).


Стогний А.А.

Лауреат Государственной премии СССР в области науки и техники 1968 года в составе коллектива разработчиков ЭВМ МИР-1. Лауреат премии им. Н. Островского, премии им. В. М. Глушкова.

Область научных интересов: системы обработки данных, информационные системы и базы данных.

Заместитель председателя РГБД, Председатель Комиссия по банкам данных и информационно-поисковым системам Координационного комитета Академии наук СССР по вычислительной технике. Член организационного и программного комитетов Всесоюзных конференций "Банки данных".

Столяров Геннадий Константинович

Инициатор и бессменный председатель созданной в стране национальной Междуведомственной «Рабочей группы по программному обеспечению банков данных (РГБД)» (1973–87гг.) объединившей ведущих разработчиков банков данных СССР. Наблюдатель от Академии наук СССР в рабочих группах по базам данных США и Великобритании.


Столяров Г.К.

Председатель Комиссии Президиума Академии наук БССР по автоматизации (информатизации) научных исследований. Руководитель Рабочей группы Комиссии Академий наук соцстран по вычислительной технике.

Был заместителем главного конструктора ЭВМ «Минск-1», «Минск-2», «Минск-23». Руководил разработкой программного обеспечения для ЭВМ «Минск». Лауреат Государственной премии СССР в области науки и техники (1970) и Государственной премии Белорусской ССР в области науки и техники (1982)

Инициатор, научный руководитель и участник разработки и внедрения семейства совместимых документально-фактографических информационных систем для больших, мини- и персональных компьютеров, баз данных и конвертеров.

Награжден медалью «Пионер компьютерной техники» (Computer Pioneer Award) — самой престижной наградой Компьютерного общества IEEE. Вручена за работу над программным обеспечением компьютеров «Минск», программным обеспечением информационных систем, за распространение и продвижение концепций систем управления базами данных.

---

[22] В.М. Савинков, В.Д. Цальп. Программирование на АЛГОЛе (Учеб. пособие для втузов). М. : Высшая школа, 1975. - 215 с.

[23] Першиков В.И., Савинков В.М. Толковый словарь по информатике: Более 10000 терминов. Москва: Финансы и статистика, 1991. – 536 с.

[24] Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем. Москва: Финансы и статистика, 1989 – 350 с..

[25] Мидоу, Чарльз Т.. Анализ информационных систем: сокр. пер. с англ. / Ч. Мидоу ; под общ. ред. и с послесл. канд. техн. наук В. М. Савинкова ; [пер. Б. В. Ананьев и др.]. - М. : Прогресс, 1977. - 400 с.

Тыугу Энн Харальдович (1935-2020)

Доктор технических наук, профессор, академик Эстонской Академии наук, почетный профессор Таллиннского технического университета, профессор Королевского технологического института в Стокгольме.


Тыугу Э.Х

Лауреат государственной премии СССР в области науки и техники.

В Институте кибернетики АН ЭССР под руководством Э.Х. Тыугу в конце 70-х гг. был разработан подход по созданию единой среды языка программирования и базы данных. В монографии[26] предложены методология концептуального моделирования предметной области и поддерживающий ее инструментарий, которые оказались применимы для создания СУБД.

Средствами системы программирования высокого уровня ПРИЗ[27], которую ее идеолог Э.Х. Тыугу квалифицирует как инструмент концептуального программирования, можно поддерживать нужную модель данных. Таким образом авторами была реализована СУБД DABU.

Филиппов Виктор Иванович

Сотрудник ВЦ АН СССР. Коллектив во главе с Виктором Ивановичем Филипповым разработал методы реализации систем управления базами данных. Им была разработана интерпретирующая система ДИАЛОГ для ЭВМ БЭСМ-6, с которой пользователи общались в интерактивном режиме сначала через телетайпы, а потом и через дисплеи. Также была разработана СУБД типа CODASYL КОМПАС для платформы БЭСМ-6 с операционной системой ДИСПАК. Под руководством В.И. Филиппова была реализована первая в стране СУБД СУРНА и интерактивная реляционная СУБД ДИСУР. Им была предложена модель, интегрирующая функциональные возможности реляционной модели и сетевой модели CODASYL, а также теоретико-множественный подход к моделям данных.

В.И. Филиппов был заместителем председателя Комиссия по банкам данных и информационно-поисковым системам Координационного комитета Академии наук СССР по вычислительной технике.

Цаленко Михаил Шамшонович

Кандидат физико-математических наук, доктор технических наук, профессор, заведовал кафедрой математики Российского государственного гуманитарного университета, заведовал научно-исследовательскими лабораториями, преподавал в Военной инженерной академии им. Ф.Э. Дзержинского, в Московском государственном университете и в Педагогическом институте им. В.И. Ленина.


Цаленко М.Ш.

Автор монографий по современной алгебре и теории баз данных, десятков статей по алгебре, информатике и лингвистике.

В начале 70-х годов он выпустил первый в стране препринт по реляционной модели данных, написанный по работам Кодда, который стал настольной книгой практически всех исследователей, занимающихся базами данных. Позже он в переработанном виде был напечатан в двух номерах сборника "Алгоритмы и организация решения экономических задач" выходящем под редакцией В.М. Савинкова[28]. Следут также особо отметить две его монографии, в которых исследуются математические модели

[26] Тыугу Э Х. Концептуальное программирование. М.: Наука, Гл. ред. физ.-мат. литерат., 1984. 256 с.

[27] Кахро М.И. и др. Инструментальная система п рограммирования ЕС ЭВМ (ПРИЗ) / М.И. Кахро, А.П. Калья, Э.Х. Тыугу. М.: Финансы и статистика, 1981, 158 с.

[28] Цаленко М.Ш. Реляционные модели баз данных (обзор) // Алгоритмы и организация решения экономических задач / Под ред. В.М. Савинкова. Вып. 9. М.: Статистика, 1977. С. 18-36

Цаленко М.Ш. Реляционные модели баз данных (обзор) // Алгоритмы и организация решения экономических задач / Под ред. В.М. Савинкова. Вып. 10. М.: Статистика, 1977. С. 16-29

баз данных[29] и методы моделирования семантики баз данных[30].

Заключение

К сожалению, многие вопросы, связанные с историей баз данных, остались не раскрытыми в этом обзоре. К ним относятся базы данных в интернете, базы данных XML, структуры хранения, методы доступа и вопросы оптимизации, языки и системы программирования баз данных, словари/справочники, работа конференций и симпозиумов, издательская деятельность. Надеемся, что эти вопросы все же будут освещены в будущем.

[29] Цаленко М.Ш. Семантические и математические модели баз данных // Итоги науки и техники. Сер.: Информатика. Т.9, 1985. 208 с.

[30] Цаленко М Ш. Моделирование семантики в базах данных. М.: Наука. Гл. ред. физ.-мат. лит., 1989. 288 с.

Сведения об авторе

Резниченко Валерий Анатольевич, ведущий научный сотрудник отдела автоматизированных информационных систем Института программных систем НАН Украины, канд. физ.-мат. наук, с.н.с.

Лауреат Премия Совета Министров СССР 1990 г. в составе авторского коллектива «За разработку и внедрение программного обеспечения информационных систем и банков данных», лауреат государственной премии Украины в области науки и техники 2009 г. в составе авторского колектива «За комплекс учебников «Информатика» в семи книгах».

Область научных интересов: информационные системы, базы данных и знаний, дескриптивные логики, онтологии семантического веба, электронные библиотеки. Является соавтором двух монографий и трех учебников.

20 лет преподавал дисциплину «Базы данных» на кафедре инженерии программного обеспечения факультета комьютерных наук Национального авиационного университета Украины.

Возглавляет небольшой коллектив, который создал и уже 15 лет сопровождает Научную электронную библиотеку периодических изданий НАН Украины (dspace.nbuv.gov.ua).

e-mail: reznichenko.valery47@gmail.com