

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 и полнотекстовые. 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, облачная, научная, многозначная, XML, NewSQL, онтологическая, полнотекстовая, 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, концепцій 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 постреляційних баз даних, а саме: NOSQL, ключ-значення, стовпчикові, документні, графові, NewSQL, онтологічні. 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, мультимодельна, хмарна, наукова, багатозначна, XML, NewSQL, онтологічна, повнотекстова, 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, full-text 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, multimodal, cloud, scientific, multi-valued, XML, NewSQL, ontological, full-text, Big Data.

Содержание

Введение	7
Этап 1. Становление баз данных (1960-1970).....	8
Система IDS.....	8
Система IMS.....	9
Система Total.....	10
Система Adabas	10
Этап 2. Бурное развитие (1970-1980)	11
Инфологическая и даталогическая модели	11
Архитектура баз данных ANSI/X3/SPARC.....	11
Предложения КОДАСИЛ	12
Сетевые СУБД.....	13
Концептуальное моделирование	13
Модели данных	14
Модель «объектов-ролей»	14
Модель данных, основанная на бинарных связях	14
Семантические модели	14
ER-модель	14
Этап 3. Эпоха реляционных баз данных (1970-1990+)	15
Реляционные базы данных	15
Вклад Э.Ф. Кодда в реляционные базы данных.....	16
Теория зависимостей и нормальных форм.....	17
Языки запросов реляционной модели.....	18
Языки реляционной алгебры.....	18
Языки реляционного исчисления.....	19
Графические языки.....	19
Языки, ориентированные на отображение.....	19
Экспериментальные исследования и разработки	20
IS/1 и PRTV	20
System/R и DB2.....	20
Oracle	20
Ingress	21
Postgres	21
СУБД для ПК.....	21
Оптимизация.....	22
Стандартизация	22
Управление транзакциями	23
Правила ACID	23
Сериализация.....	23
Модели транзакций.....	24
Модель страниц	24
Плоские транзакции	24
Точки сохранения	24
Модель многозвенных транзакций	24
Вложенные транзакции.....	25
Открытые вложенные транзакции	25
Распределенные транзакции.....	25
Гибкие транзакции	25
Длительные транзакции, компенсаторы и модель Saga	26
Транзакции Split/Join.....	26

Кооперативные транзакции	26
АСТА и ее производные	27
Транзакции веб-сервисов.....	27
Этап 4. Расширенные реляционные базы данных (1980+-2000+).....	28
Темпоральные базы данных.....	29
Основные понятия.....	29
Темпоральные модели данных	29
Темпоральные зависимости	30
Темпоральные языки	30
TSQL2	30
Темпоральный SQL: 2011	31
Пространственные базы данных	31
Модели пространственных данных	32
Полевая модель.....	32
Объектная модель.....	32
Операции.....	32
Пространственные типы данных	33
Отношение главного направления	33
Концептуальное моделирование	34
Пространственно-сетевые базы данных	35
Пространственно-временные сетевые БД.....	35
Пространственно-временные базы данных	35
БД перемещающихся объектов.....	36
Пространственные СУБД.....	36
Дедуктивные базы данных	36
Безопасное правило.	37
Простые правила	37
Рекурсивные правила.....	37
Правила с отрицаниями.....	38
Оптимизация.....	38
Дедуктивные системы баз данных	38
Активные базы данных	39
ЕСА-правило	39
Модели ЕСА-правил.....	39
Модель знаний ЕСА-правил.....	39
Характеристики модели знаний события.....	40
Характеристики модели знаний условия	40
Характеристики модели знаний действия.....	40
Модель исполнения ЕСА-правил.....	40
Модель исполнения событий	40
Правила проверки условий и выполнения действий	40
Запуск событием нескольких правил	41
Политика итогового эффекта.....	41
Вызов правилом другого правила.	41
Системы активных баз данных	41
Активные РБД	41
Активные ООБД.....	41
Система HiPAC.....	42
Объектные базы данных	42
Первые объектные СУБД	42
Два направления в ОБД.....	42
Манифест объектно-ориентированных систем баз данных	43
Стандарт на хранение объектов ODMG 3.0.....	43

Второй манифест.....	43
Третий манифест.....	44
Схемы реализации ОРБД.....	44
Объектно-реляционные СУБД.....	44
Распределенные базы данных.....	45
Типы РаБД.....	45
Однородные РаБД.....	45
Неоднородные РаБД.....	45
Федеративные РаБД.....	45
Посредники.....	46
Одноранговые БД.....	46
Распределение данных. Фрагментация.....	46
Распределение данных. Репликация.....	46
Тупики в РаБД.....	47
Распределенная обработка запросов.....	47
Управление параллелизмом.....	48
Блокировка.....	48
Машины баз данных.....	49
Базы данных, поддерживающие работу с массивами.....	52
Модели и языки.....	52
Хранение массивов.....	52
Архитектура реализации.....	53
Другие системы БД массивов.....	54
Статистические базы данных.....	54
Статистические модели данных.....	54
Статистические операторы.....	55
Метаданные.....	55
Системы и языки запросов статистических баз данных.....	55
Хранилища данных.....	57
Архитектура DWH.....	58
Модели DWH.....	59
Многомерная модель данных DWH. Куб данных.....	60
Операции над OLAP-кубами.....	61
Агрегирующие функции.....	61
Многомерные базы данных.....	61
Концептуальные схемы DWH.....	62
Методологии проектирования и моделирования.....	62
Инструментальные средства.....	63
Активные DWH.....	63
DWH реального времени.....	63
Эволюция DWH.....	63
Темпоральные DWH.....	63
Пространственные DWH.....	63
SQL и OLAP.....	64
Этап 5. Постреляционные базы данных (2000 – 2010+).....	65
NoSQL-базы данных.....	65
История термина NoSQL.....	65
Свойства NoSQL баз данных.....	65
Теорема CAP.....	66
Теорема PACELC.....	66
Типы NoSQL баз данных.....	67
Модель ассоциативного массива.....	67
БД ключ-значение.....	67

Модель триплетов	67
Колоночные БД.....	68
Документо-ориентированные БД.....	68
Графовые БД.....	68
Структура GLOBAL.....	68
Языки запросов.	68
Документо-ориентированные базы данных	69
Слабоструктурированные данные	69
Языки запросов	69
Системы	70
Колоночные базы данных	70
История КБД.....	70
Характерные черты и область применения.	71
Методы реализации и оптимизация	71
Графовые базы данных	73
Графовые модели данных	73
Графовые языки запросов (ГЯЗ).....	76
Графовые базы данных (ГБД).....	78
NewSQL-базы данных	79
Онтологические базы данных	80
Полнотекстовые базы данных	81
Метаданные	82
Индексация	82
Языки запросов	82
Алгоритмы точного сопоставления.....	82
Символьный подход.....	83
Подход с хешированием	83
Автоматный подход	84
Бит-параллельный подход	84
Гибридный подход	85
Алгоритмы неточного сопоставления.....	85
Расстояние Хэмминга	85
Расстояние Левенштейна.....	85
Расстояние Дамерау-Левенштейна.....	85
Расстояние Джаро-Винклера.....	85
Расстояние при упорядоченном алфавите	85
Обзор литературы	86
Электронные библиотеки.....	86
Этап 6. – Большие данные (2010 – 2020+).....	88
Характеристические свойства больших данных.....	89
Классификация больших данных	89
Принципы работы	90
Методы и технологии	90
Модель больших данных.....	90
Исследования и разработки баз данных в Советском Союзе (1970-1991)	92
Организация и инфраструктура исследований и разработок	92
Создание программного инструментария	93
Разработка приложений	94
Научные исследования в области систем баз данных.....	95

Персоналии	99
Андон Филипп Илларионович.....	99
Дрибас Виктор Прокофьевич.....	99
Замулин Александр Васильевич.....	100
Калиниченко Леонид Андреевич.....	100
Когаловский Михаил Рувимович	100
Пасичник Владимир Владимирович	101
Савинков Владимир Макарович.....	101
Стогний Анатолий Александрович	101
Столяров Геннадий Константинович.....	102
Тыугу Энн Харальдович (1935-2020).....	102
Филиппов Виктор Иванович.....	102
Цаленко Михаил Шамшонович.....	103
Заключение.....	103
Литература	103

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 немислимо 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.

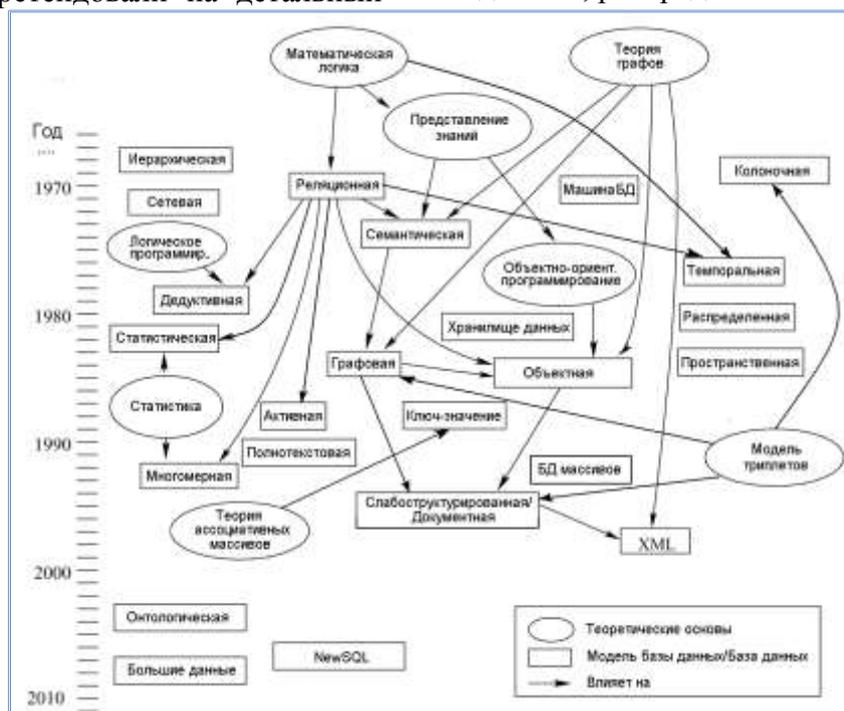
Мы не претендовали на детальный

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.



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 года был закончен прототип этой системы, а в начале 1964 года была выпущена первая промышленная версия IDS [14-16]. 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 десятилетия. 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 переименова-

на в Information Management System – IMS (система управления информацией). Согласно [1] 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 [17-19]. 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 [20]. 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 эффективно.



Томас Нис

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.

В Total поддерживалось обращение из Cobol, Fortran, PL/1 и Assembler. Язык манипулирования напоминал спецификацию Coddasy1. 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 вклады в развитие СУБД.

Система Adabas. Adabas (adaptable database system - адаптивная система баз данных) — система управления базами данных компании Software AG, Германия. Впервые выпущена для мейнфреймов IBM в 1971 го-

ду. 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.

IDS, IMS, Total, Adabas относятся к классу так называемых *навигационных баз данных*. Этот термин был введен Чарльзом Бахманом в своей статье [21], 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 (Т. William Olle) [1] этот термин впервые был введен 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 [22].

Этап 2. Бурное развитие (1970-1980)

70-е 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 востребованы 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 [23], а также формулировались требования к СУБД [24, 25].

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),



Бордже Лангефорс

которые он развивал на протяжении 15 лет [26-28]. 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.

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. Сундгрэн

Идеи Лангефорса в дальнейшем были развиты и адаптированы к технологиям баз данных шведским ученым Б. Сундгреном (Bo Sundgren) [29]

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 программой,

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 V. 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 [30].

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 выявляе-

ния этих компонент были положены следующие концептуальные положения. Во-первых, существует реальный мир, информационная модель которого должна найти свое отражение 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 мощьность механизмов описания отображений определяет степень достаточности достижения упомянутых двух видов независимостей. Результаты деятельности этой рабочей группы были представлены в отчетах [31, 32].

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 [23, 33, 34], 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 требуемой 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) [1].

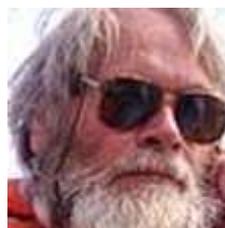


Уильям Олле

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



Дж. Грийтусен

данных с целью обеспечения основы для стандартизации в данной области. Сначала эту группу возглавил Т.Б. Стилмладший (Thomas B. Steel, Jr.), а затем Д.А. Жардин (D.A. Jardine). В результате деятельности этой группы в 1982 был выпущен отчет [35] под редакцией Дж. Грийтусена (Joost J. Van Griethuysen).

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 [13] термин "модель данных" стал использоваться в начале 70-х годов после публикации фундаментальной работы Эдгара Кодда (E. Codd) [59]. 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» (ORM - Object-Role Model) Экхарда Д. Фолкенберга (Falkenberg, Eckhard D) [36, 37], 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 объектами). 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



Экхард
Фолкенберг

Relations) стоят работы таких авторов, как Абриаль [38] (семантическая бинарная модель), Браччи [39], Дурхольц [40]. 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 термина. 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 [41, 42, 43].

Семантические модели. Отметим работы Дж. Смита и Д. Смит по моделям абстракции, агрегации и обобщения данных [44, 45], а также семантическую модель данных SDM Хаммера и МакЛеода [46]. В статье [47] 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) [2].

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 [48], а также модели Инглеса [49].



Петер Чен

Наиболее полно впервые эту модель описал П.П. Чен (Петер Пин-Шен Чен - Peter Pin-Shan Chen) [50]. 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 ре-

альный 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).

После публикации статьи Чена появилось множество статей, посвященных исследованию различных аспектов ER-моделирования предметных областей. Например, в общем случае предполагается существование n-арных связей, а Ричард Баркер (Barker Richard) предложил ER-модель только с бинарными связями [51], которая имеет определенные преимущества.

В связи с широким использованием ER-модели [3] было предложено множество различных ее расширений и обобщений [52-55], которые в конечном итоге привели к определению иерархической ER-модели (ER-модели более высокого порядка) [55].

В статье [56] ER-модель расширена включением элементов семантизации данных. Также была предложена темпорально-расширенная ER-модель [57], 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 [58]. Наконец, существует пространственная ER-модель (см. «Пространственные базы данных»).

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

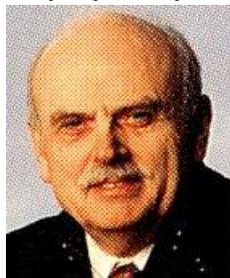
В начале 80-х годов появились первые промышленные реляционные СУБД, которые к концу 80-х гг. быстро завоевали рынок и стали господствующими практически на всех распространенных аппаратно-программных платформах и не утратили свое преимущество по настоящее время. Тем не менее, основы реляционной модели 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 транзакциями, за которые в 1998 г. Джеймс Николас Грей был награжден премией Тьюринга.

Реляционные базы данных

В 70- годах ученые, занимающиеся базами данных, были уверены, что будущее баз данных лежит в создании все более и более сложных структур данных, которые бы позволяли адекватно представлять информационную модель данных произволь-

ных 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 [59] малоизвестно-



Эдгар Франк Кодд

го на то время британского ученого Эдгара Франка Кодда (Edgar Frank Codd), работающего в компании ИВМ, в которой он предложил наиболее простую структуру данных, представляющую собой одномерную, плоскую, нормализованную таблицу.

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 [60], 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 [61], 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) [62] 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 [63]. 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 [64] 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 [64a] 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. Основоположителем этой теории стал Э.Ф. Кодд, опубликовав работы [65-67]. 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 [68] - 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) [69].



Раймонд Бойс

По справедливому замечанию Дейта, первоначально эту нормальную форму определил Ян Хит (Ian Heath) в статье [70]. 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.



Ян Хит

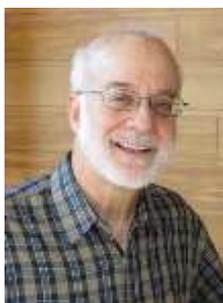
Эта теорема была названа его именем (теорема Хита). Она используется при приведении отношений в 2NF, 3NF и BCNF.

В 1974 году Вильям Армстронг (William Ward Armstrong) в статье [71] 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 [72] 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 четвертой нормальной формой (Fourth Normal Form - 4NF), а алгоритм приведения в 4NF базировался на до-

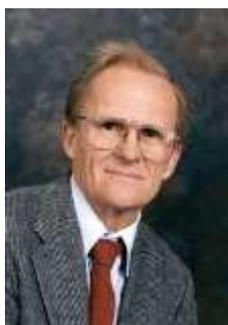


Рональд Феджин

казанной им теореме (теорема Феджина). В последующей статье [73] была предложена полная система аксиом MVD, а также две аксиомы, связывающие FD и MVD (выводимость MVD из FD и наоборот).

Отметим, что независимо от Феджина многозначную зависимость также исследовал Заниоло [74]. Кроме того, Делобел [75] 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) [76], которая явилась обобщением MVD (MVD является бинарной JD). На ее основе Феджин в статье [77] 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

Йорма Риссанен (Jorma Rissanen) определил зависимость по соединению (join dependency - JD) [76], которая явилась обобщением MVD (MVD является бинарной JD). На ее основе Феджин в статье [77] 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).



Кристофер Дейт

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 [78] 6NF равносильна доменно-ключевой нормальной форме (DK/NF) Феджина (см. далее).

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 [79]:

– улучшенная 3NF (Improved 3NF) [80];

- нормальная форма элементарного ключа (Elementary Key Normal Form – EKNF) [81];
- нормальная форма суперключа (Superkey Normal Form – SKNF) [82];
- приведенная 5NF (reduced-5NF – 5NFR) [83];
- нормальная форма без избыточности (Redundancy Free Normal Form-RFNF) [84];
- нормальная форма с существенными кортежами (Essential Tuple Normal Form – ETNF) [85];
- доменно-ключевая нормальная форма Феджина (Domain-Key Normal Form – DK/NF) [86];
- иерархическая зависимость [87] и ее связь с иерархической структурой данных [88];
- зависимость по включению (Inclusion dependency) и нормальные формы по включению (Inclusion Normal Forms) [90-92].

В заключение отметим, что мы привели только незначительное количество исследованных зависимостей. В книге [93] приведен перечень более 600 статей, посвященных теории зависимостей и нормальных форм, а в монографии [94] анализируются около 90 зависимостей.

Языки запросов реляционной модели. Реляционная модель дала существенный толчок исследованиям по созданию языков запросов.



Дональд Чемберлин

В своем обзоре [95] 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.

Языки реляционной алгебры. Были предложены и экспериментально апробированы следующие языки/системы, базирующиеся на реляционной алгебре: система MACAIMS [96], разработанная в MIT, системы IS/1 [97] и PRTV (Peterlee Relational

Test Vehicle) [98], разработанные в научном центре IBM в Питерли, Англия, система RDMS [99], 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 [101-104]. В [105] 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 [60], который непосредственно основывается на реляционном исчислении. ALPHA и позволяет пользователю, используя такие понятия, как переменные и кванторы, формулировать непроедурные запросы. Впоследствии были предложены другие языки, которые, как и ALPHA, базировались на реляционном исчислении. К ним относятся QUEL [106], созданный в рамках научно-исследовательского проекта Ingres в Калифорнийском университете в Беркли, COLARD (Calculus Oriented Language for Relational Data) [107], RIL [108].

Графические языки. В языках, относящихся к графическим, формулировка запросов производится не с использованием традиционного линейного синтаксиса, а заполнением ячеек в бланках таблиц. Язык CUPID (Casual User Pictorial Interface Design - работа с графическим интерфейсом непрофессионального пользователя) [109-112] предоставляет пользователю графический язык запросов. CUPID содержит высокоуровневый меню-образный подъ-



Моше М. Злуф

язык, который является внешним интерфейсом к системе INGRES. 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 [113], 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) [118], который явился расширением QBE для офисных приложений.

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 (Specifying QUeries As Relational Expressions - спецификация запросов в виде реляционных выражений) [119, 120]

Использование SQUARE-подобного языка для описания множественных представлений (взглядов), а также управления целостностью данных и их авторизации описано в статье [121]. 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 [120].

В 1974 г. Бойс и Чемберлин представили язык SEQUEL (Structured English QUery Language - структурированный английский язык запросов) [122], 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 [123]. Задача

В 1974 г. Бойс и Чемберлин представили язык SEQUEL (Structured English QUery Language - структурированный английский язык запросов) [122], 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 [123]. Задача

В 1974 г. Бойс и Чемберлин представили язык SEQUEL (Structured English QUery Language - структурированный английский язык запросов) [122], 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 [123]. Задача

В 1974 г. Бойс и Чемберлин представили язык SEQUEL (Structured English QUery Language - структурированный английский язык запросов) [122], 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 [123]. 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 [124, 125]) – системе, разработанной для хранения и поиска данных, представленных в виде n-арных отношений. XRM, в свою очередь, реализована на базе RM (Relational Memory) [126, 127], предоставляющей эффективный ассоциативный доступ к бинарным отношениям. Наконец, в 1976 г. был представлен язык SEQUEL2 [128], в котором уже были включены все основные средства для оперирования базами данных: определение, манипулирование и управление.



Карл Карлсон

Оригинальный подход был предложен в языке APPLE (Access Path Producing Language – язык, порождающий путь доступа) [129], 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.

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. [130]. С учетом результатов, полученных при реализации IS/1, была разработана СУБД PRTV (Peterlee Relational Test Vehicle) [131], которая позволяла оперировать большими объемами данных, имела свой собственный язык запросов ISBL

уровня реляционной алгебры и была одно-пользовательской.

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 упрощенного варианта языка [123]. Затем была реализована полнофункциональная многопользовательская версия System/R [132]. Наконец, на протяжении 1978-1979 годов System/R прошла всестороннюю практическую апробацию [133, 134], 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 [135]. Используя полученный опыт, компания IBM в 1980 г. приступила, а в 1982 г. выпустила промышленную реляционную СУБД 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



Ларри Эллисон

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 ре-

ляционную 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.



Майкл
Стоунбрейкер



Юджин Вонг

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 (INteractive Graphics and RETrieval System). Последующие два года были проведены экспериментальные исследования и разработки. Были приняты проектные решения [136, 137], разработаны структуры хранения и методы доступа [138] Был разработан оптимизационный алгоритм выполнения операций соединения отношений, получивший название алгоритма Вонга-Юсефи (Wong-Youssefi algorithm) [139], исследован механизм предоставления альтернативных взглядов (view) путем подстановки в запросы пользователей их определений взглядов [140]. Авторизация и контроль целостности обеспечивался добавлением дополнительных предикатов к запросам пользователя [141]. Реализован механизм безопасного одновременного обновления базы данных [142], а также система защиты [143]. К 1976 году была реализована экспериментальная версия INGRES [144], 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).



Лоуренс Роу

Были разработаны концептуальные проектные решения [145], предложена, объектно-реляционная модель со сложными типами данных [146], разработаны структура хранения данных [147] и система правил [148] (триггеров), которая позволяет определять дополнительные действия, иницируемые 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 [149]. 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.

В 2014 г. Майкл Стоунбрейкер стал лауреатом премии Тьюринга за фундаментальный вклад в концепции и методы, лежащие в основе современных систем баз данных [150].

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 непроектурном 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. Мы не будем останавливаться в этой статье на данной проблеме, и отсылаем читателя к прекрасному обзору С.Д. Кузнецова [151].

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) для проведения исследований по обоснованию



Майкл Броди

возможности создания стандарта по реляционным базам данных. В 1981 году эта группа выпустила отчет [152], 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



Дональд Дойч

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).

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 промышленных реляционных 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.

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 [153, 155, 158].

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 целостном 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 [153]. Вместе с тем существуют специализированные системы с ослабленными транзакционными свойствами [154].

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 [155, 158]. Кроме того, в работе [6] был определен двухфазный протокол блокирования, и исследована техника предикатного блокирования. Вопросы

области действия блокирования (гранулярности) обсуждены в статьях [155, 156].

Модели транзакций. Следует отметить, что все модели транзакций определялись, как правило, с учетом классов прикладных систем, в которых они находят применение [11]. 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 [153, 157] и Капали Эсваран (Kapali Eswaran) [158]. В это же время возникло родственное понятие - атомарное действие [159, 160]. Концепция страничной модели транзакции стала предметом интенсивных теоретических исследований в 80-х годах [161-164] и является действенной по настоящее время, хотя приобрела ряд расширений и вариаций [165]. С обзором исследований и разработок этой модели можно познакомиться в работах [11, 166]

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 [167]. Согласованность обеспечива-

ется механизмом управления целостностью. Было предложено два подхода по управлению целостности в транзакциях: включение этого механизма в СУБД [168] и поддержание целостности за счет усилий разработчиков приложений [11].

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 [132]. 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 дан-

ных, поэтому откат производится к этой точке сохранения. Отметим, в этой модели атомарность и изолированность не гарантируется для всей транзакции. Согласно [11] 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) [169], а затем в [170]. Ее концепция базировалась на понятии сфер контроля (spheres of control) [171]. 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 аварийном 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 [172] предложены следующие варианты синхронизации запуска субтранзакции триггера и по отношению к основной транзакции:

- *немедленно (immediate)* - субтранзакция запускается сразу же после наступления события триггера;
- *откладывается (deferred)* - 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) [173] 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 [173, 174]. Субтранзакции многоуровневой транзакции могут произвести фиксацию и освобождение своих ресурсов до завершения работы глобальной транзакции. Если глобальная транзакция завершается аварийно, то для поддержания атомарности следует произвести откат субтранзакции 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 [175]. В монографии [176] 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 [177] 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) [178]. Эта модель является стандартом для протокола двухфазной фиксации (2PC - Two Phase Commit).

Гибкие транзакции (Flexible Transactions) [179, 180] - были предложены для среды распределенных баз данных. В этом случае глобальная транзакция пред-

ставляет собой набор субтранзакций, каждая из которых осуществляет доступ к данным 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 [179-183]. Был предложен язык IPL [184] 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 [157], затем она была формализована в [185, 186] и, наконец, использована в модели Saga [187]. Эта модель имеет отношение к длительным транзакциям (Long-Running Transactions) [188]. 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 завершённые субтранзакции откатываются выполнением, так называемых компенсирующих субтранзакций. Saga ослабляет требование к изолированности и увеличивает межтранзакционный параллелизм. В работе [189] предложена усовершенствованная модель - вложенная Saga, которая позволяет представлять линейную структуру долговременных транзакций в виде иерархической транзакционной структуры.

Транзакции Split/Join (разделить/ со-

единить). Концепция разделения/соединения транзакций была впервые описана в [190] и затем тщательно проработана в [191] 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 равносителен передаче всей работы одному исполнителю [191]. Впоследствии операции Split и Join были включены во вложенные транзакции для создания комбинированных моделей транзакций [192]

Кооперативные транзакции были предложены в [193] 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.

АСТА и ее производные. АСТА [192, 194, 195] - это метамодель, которая облегчает спецификацию, анализ и синтез расширенных моделей транзакций. Формализм АСТА 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, АСТА предоставляет возможность специфицировать требования новых транзакционных приложений и синтезировать модели, удовлетворяющие этим требованиям. В работе [42] авторы предложили упрощенное средство разработки расширенных транзакций, названное ASSET. Оно основывается на транзакционных примитивах, позаимствованных у АСТА 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 [197, 198, 199], и в мобильных системах баз данных [200, 201].

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

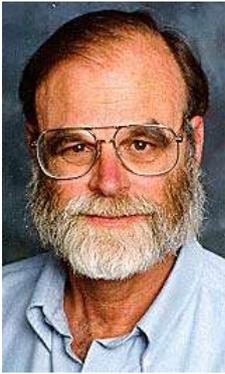
веб-сервисных приложений. В настоящее время разработано три стандарта, имеющие отношение к транзакциям веб-сервисов.

Business Transaction Protocol (BTP) [202, 203]. 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 [204].

Web Services Transactions (WS-Tx). [205, 206] 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 [207].

WS Composite Application Framework (WS-CAF) [208]. 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 [209, 211].



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.

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

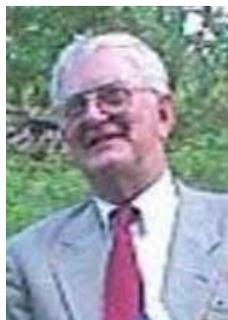
С момента возникновения реляционная модель подвергалась критике в связи с простотой ее структуры данных. В связи с этим предлагались более развитые модели, которые позволяли более адекватно представлять информационные модели различных предметных областей. Тем не менее, их характерной особенностью было 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.

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 про-



Янис Бубенко мл.

шлого столетия. Считается [212], 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) [213]. Впоследствии были предприняты активные исследо-

вания по раскрытию семантики времени на концептуальном уровне [213-218], созданию зависимых от времени моделей данных для статических реляционных баз данных [219-222] и разработке темпоральных языков запросов [223-227].

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 [227a]. 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 [228]

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, поддерживаемые 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 [229]

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) [224, 225] - битемпоральная интервальная модель;
- Джонс и Мейсон (Jones and Mason) [226] - интервальная модель действительного времени;
- Ричард Снодграс (Richard Snodgrass) [227] - битемпоральная точечная и интервальная модель;
- Лоренцос и Джонсон (Lorentzos and Johnson) [230, 231] - точечная и интервальная модель действительного времени.

Все эти модели предполагали привязку времени к кортежам. Также были предложены темпоральные модели с привязкой времени к атрибутам [232-237].

С обзором исследований по темпоральным моделям данных можно познакомиться в работах [4а, 238-240].

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) [241]. 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 (Wijssen) [242, 243]. В этом случае требуется, чтобы FD выполнялась в объединении старого и нового отношения. Вийсен также определил тренд-зависимость (trend dependency) [244], которая является обобщением определенной им же темпоральной FD.

Еще одна разновидность TFD была определена в работе [245]. В этой статье было предложено расширение теории нормализации с учетом избыточности, которая порождается TFD. В работе [246] 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 [247] были исследованы темпоральные расширения отношений специализации и обобщения.

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

Были проведены исследования и предложены темпоральные реляционные алгебры [230-232, 236, 237, 248-250] и исчисления [233, 236] для различных темпоральных моделей данных. Была также определена вложенная битемпоральная модель данных и соответствующая ей алгебра [234]

Тансел и Аркун разработали язык HQUEL [251], представляющий собой расширение языка QUEL "историческими" данными. Они же предложили язык TBE (Time-By-Example) [252], в котором воспользовались идеей графического реляционного языка QBE. Снодграс также предложил темпоральный вариант QUEL, которые был назван TQUEL [227, 253].

Были предложены различные варианты темпорального расширения SQL [235, 254-258].

Обширный перечень темпоральных реляционных и объектных языков запросов приведен в [4а], а в [258а] приводится обзор некоторых из них. Обзору темпоральных языков запросов также посвящены работы [258б, 258 в].

TSQL2. Одним из ключевых периодов в области исследований темпоральных баз данных, временем ее «официального» представления можно считать 1992–1995 гг.

Сначала Ричард Снодграсс (Richard T. Snodgrass) высказал идею о возможном темпоральном расширении стандарта SQL-92, а затем в 1993 г. был проведен семинар [259], продемонстрировавший заинтересованность научного сообщества в разработке темпорального расширения стандарта SQL-92.



Ричард Снодграсс в результате был учрежден комитет по созданию такого языка, который получил название Temporal Structured Query Language TSQL2. Ведущую роль в работе комитета сыграл Снодграсс. 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 [260], а в сентябре - учебное пособие [261]. Наконец, в 1995 г. была опубликована окончательная спецификация языка запросов TSQL2 [262].

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 [263, 264]. Окончательные предложения перехода от TSQL2 в SQL3 были сформулированы в [265].

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 про-

тиворечий. В свою 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 [266, 267].

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 [268]:

- *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 [271], 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 [273] 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 [274]. В [275] предложен расширяемый язык запросов для географических баз данных. В статье [276] впервые было предложено расширение алгебры реляционной модели введением пространственных объектов и операций. В статье [277] предложен язык SpatialSQL, в котором в язык SQL включены пространственные операции и отношения. Алгебра ROSE (**RO** bust **S**patial **E**xtension) [278] 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 [279] 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 [269]:

- *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,...);
- 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 [270, 272]. В частности OGC представил спецификацию [272] встраивания в SQL двумерных геопространственных АТД на основе объектной модели, и предложил исчерпывающий список операций. В монографии [269] 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 [269, 280 281]. 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 [276, 278, 282-286], для поверхностей и объемов в [287], для пространственных разбиений в

[288] и для пространственных сетей в [289]. Оригинальный подход для определения пространственных типов данных был предложен в [289], 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 (Cardinal direction relationships). Понятие "направление" является одной из важных характеристик пространственных систем. Алгоритмы вычисления пространственных направлений лежат в основе ПБД и ГИС.

Отношение главного направления - (ОГН) - 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 [292, 295]. 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 [298, 299]. Однако этот метод давал неправильное направление, когда объекты наложены друг на друга, переплетены или подковообразные [293]. 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 проекционного отношения направления (projection-based directional relationships) [293, 301, 302] и конусного отношения направления (cone-based directional relationships) [269]. Было также предложено моделировать ОГН тернарными отношениями [291]. Относительно перечисленных выше моделей ОГН исследовались следующие задачи:

- эффективное определение отношений, которые имеются между множествами объектов [291, 299, 300, 303],
- вычисление инверсных отношений [290, 291, 295, 299, 304],
- вычисление композиции двух или более отношений [295, 299, 301, 304],
- проверка согласованности множества отношений [295, 296, 297, 299, 302].

Оригинальное решение по моделированию направления в виде пространственного объекта было предложено в работе [305].

Концептуальное моделирование.

Общепризнанным средством концептуального моделирования является ER-язык. Было предложено много расширений этого языка для представления в нем пространственных объектов с пространственными характеристиками [306]. Одним из таких расширений является ER-схема с пиктограммами [307]. В ней пиктограммы используются для указания типов пространственных сущностей и пространственных связей. Базовыми типами геометрических фигур являются: точка, (ломаная) линия и многоугольник. Они имеют следующие пиктограммы объектов:

-  точка
-  линия
-  многоугольник

 *Мульти-фигуры* - это множества базовых фигур. Предоставляется возможность указать количество элементов в мульти-множестве 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 может быть представлен точкой или многоугольником.

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

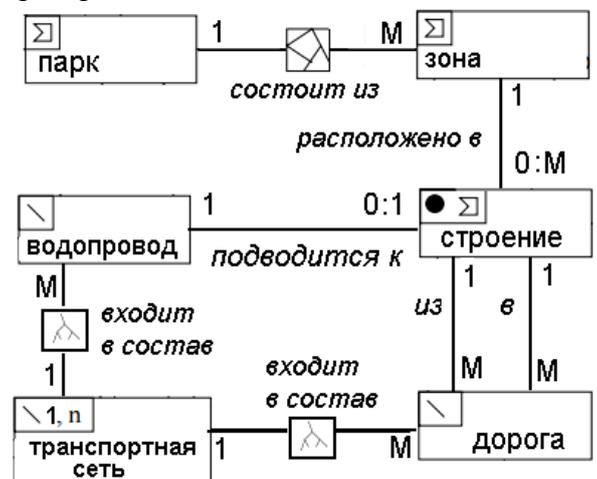
 *Определяемая пользователем фигура* - объект представляется не стандартной фигурой, а определенной пользователем.

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 [269, 308].

Были высказаны предложения по использованию геопропространственных онтологий для концептуального моделирования геоинформационных систем [309], которые давали бы возможность представлять в модели ПО более глубокую пространственную семантику. Кроме того, предлагается расширить геопропространственные онтологии темпоральными характеристиками с использованием OWL-Time [310].

Все исследования по пространственным онтологиям разделяются на две категории:

- онтологии для интеграции структур хранения пространственных данных. В этом случае решается задача интеграции различных ПБД с целью решения задачи обмена данными между ними [311];
- 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 [312], либо создается некая универсальная онтология, охватывающая максимально широкий круг понятий и характеристик геопространственных объектов [313-315].

Огромный объем геоинформационных ресурсов в интернет инициировал исследования по созданию геопространственного семантического веба [316, 317], так, например, было предложено геопространственное расширение RDF (GeoRDF) [318]

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.[307] с нагруженными необходимой информацией вершинами и ребрами. Более совершенной является транспортная модель данных UNETRANS [319].

Логическая модель данных предполагает использование модели конкретной коммерческой СУБД. Обычно использует объектно-реляционную модель. В работе [269] описываются специальные операции на графах, которые используются в ПСБД. Еще одна логическая модель системы GraphDB описана в работе [320]

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 [269]. В работе [321] предложен метод доступа ССАМ. В статьях [322, 323] 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 [324-327].

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 решения транспортных задач следует использовать пространственно-временные модели сетей. В этом направлении также проводятся исследования. Так, например, для моделирования пространственно-временных сетей в статье [328] предлагается создавать копии всей пространственной сети для каждого необходимого момента времени а в работах [329, 330] 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 точек, 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. [331]. Другой подход [332] - 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 [333].

Был предложен пространственно-временной язык запросов STQL [334] на базе SQL. В [277] также предлагается вариант пространственного SQL. В [334-336] предлагаются пространственно-временные предикаты. В работе [337] дается обзор современного состояния исследований по ПВБД.

БД перемещающихся объектов (БДПО). Это пространственно-временная база данных, предназначенная для фиксации и отслеживания местоположения движущихся объектов. Исследования по БДПО были инициированы в конце 90-х годов прошлого столетия [333, 338, 339, 340, 341]. Как правило, БДПО используют плоскую пространственно-сетевую модель данных [342, 343].

Два исследования дали жизнь этому направлению. Во-первых, была предложена модель MOST (Moving Objects Spatio-Temporal) [338, 339], 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 [340], 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) [344, 345] и разработан прототип DEDALE [341].

Различают два вида БДПО: в первом случае БДПО моделирует, представляет и дает возможность формулировать запросы к предыстории перемещения для проведения последующего пространственно-временного анализа [346, 347]. Вторая разновидность предоставляет возможность моделировать, прогнозировать и запрашивать текущее и

будущее перемещение [338, 348]. Во втором случае приходится выбирать между неточностью прогнозных результатов и затратами на обновление БД [339], что приводит к решению задачи управления неопределенностью [349].

Распространенным подходом в исследованиях по БДПО является создание специальных типов данных (перемещающиеся точки и перемещающиеся области (многоугольники), специальных операций и предикатов. Так, например, в [333] предлагаются типы данных, которые позволяют задавать зависимые от времени пространственные объекты и операции над ними. Система типов данных для движущихся объектов была строго определена в работе [350].

В заключение отметим, что в монографиях [335, 351, 352] детально освещены практически все вопросы, имеющие отношение к движущимся 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 SpatialLite для SQLite,

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

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.

В конце 70-х гг. начали формироваться подходы по использованию аппарата логики в базах данных [5, 6]. В 1982 г. Чандра и Харел [353] опубликовали статью, которую считают первой работой в области тео-

рии 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 экстенциональной базой данных (ЭБД), а вторая – интенциональной 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 А то В", где А называется "телом", 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 [354]. Затем термин "Datalog" использовался в работах [355, 356]

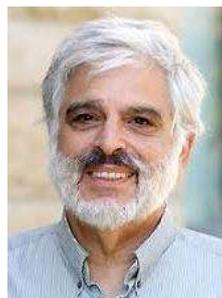


Давид Майер

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 [357]. Это термин был взят в качестве сокращения от "database dialogue" и никакого отношения к Prolog не имел.

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



Джеффри Ульман

опубликована замечательная монография Джеффри Ульмана (Jeffrey D. Ullman) [6], в которой отдельная глава посвящена детальному изложению сущности Datalog как логической модели данных. Многое из этой главы используется далее при изложении принципов Datalog.

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 [358, 359].

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, [6]). 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 разницы, являются монотонными. Поэтому в реляционной алгебре, расширенной циклами можно построить алгоритм, вычисляющий наименьшую неподвижную точку рекурсивной Datalog –программы. В [6, 360] 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 [361-368]. Отрицания также вызывают проблемы в рекурсии. В связи с этим было введено понятие стратифицированного отрицания [361, 362, 369, 370] и стратифицированных программ, которые имеют интуитивно понятную семантику [371-374]. Стратификация не гарантирует существование наименьшей неподвижной точки. Однако это ограничение дает возможность выбора среди множества минимальных неподвижных точек такой "предпочитаемой", которая будет интерпретацией смысла Datalog-программы. Были также определены более специфические классы стратифицируемых программ, например, локально стратифицированные программы [375] модульно стратифицированные программы [376].

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 исследо-

вания в этом направлении. Среди них можно отметить метод магических множеств (magic-sets) [6], а также ряд базирующихся на нем методов [377], алгоритм подсчета (counting algorithm) [378], факторинговая оптимизация (factoring optimization) [379], метод удаления избыточных правил и литералов [380], метод оптимизации экзистенциальных запросов [381], метод "конвертов" (envelopes) [382] и другие. В работе [383] 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 [384, 385]. С другой стороны, в 1999 г. в очередную версию SQL (SQL99) была введена 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 [386], а с кратким описанием практического использования - в работе [387]

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 [388, 389]. В статье [390] дается аналитический обзор объектно-дедуктивных баз данных.

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 [33, 34] (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 данных 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.

ЕСА-правило. Для поддержки перечисленных выше возможностей в активной базе данных было введено понятие ЕСА-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) [392]. Семантика правила простая: при наступлении события проверяется условие и если оно истинно, выполняется действие.

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

Действие ЕСА-правила - произвольный код, вызываемый при наступлении события и при истинности условия. Действие выполняется либо в виде составной части транзакции ЕСА-правила, либо в виде самостоятельной транзакции в зависимости от режима связывания. В пределах одного ЕСА-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 ЕСА-правил. Для описания ЕСА-правил были предложены две модели: *модель знаний* (knowledge model) и *модель исполнения* (execution model) [391, 392]. Модель знаний описывает, что собой представляют активные правила, а модель исполнения специфицирует, каким образом интерпретируются ЕСА-правила в процессе их выполнения.

Модель знаний ЕСА-правил. В работе [393] были обобщены и классифицированы характеристики модели знаний, которые ранее были представлены в литературе [394-

396]. Эти характеристики относятся ко всем составляющим ЕСА-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. Основопологающие исследования по выявлению событий были проведены при выполнении проектов HiPAC [392, 397], Snoop [398, 399], ODE [400], SAMOS [401]. В [402] дается обзор исследований по обнаружению событий, а в статье [403] - 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 [392]. Для описания составных событий в рамках разработанных систем были определены различные алгебры, например, [395, 398, 399, 401, 404]. В работе [398] для составных событий введена характеристика "политика потребления" (consumption policy), определяющая ситуацию, в которой считается, что составное событие произошло. В работе [405] предлагается общий метод и язык EPL спецификации семантики составных событий.

Характеристики модели знаний условия. К ним относятся следующие:

- *факультативность* - является ли условие обязательным в ЕСА-правиле, или нет. Считается, что в правиле должно присутствовать либо событие, либо условие.

- *контекст* - в целом в качестве контекста ЕСА-правила предлагаются следующие варианты: состояние базы при: запуске транзакции (DBT), иницировании 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, иницирование внешней среды, информирование, аварийное завершение, выполнение иного действия чем то, что инициировало событие ("do-instead" Стоунбрейкера [394])
- *контекст* - аналогично контексту условия специфицирует, какие именно данные доступны действию. Допустимыми значениями являются DBT, DBE, DBC, DBA и BindC

Модель исполнения ЕСА-правил. Модель исполнения (execution model) специфицирует, каким образом трактуются ЕСА-правила в процессе их выполнения [391]. Она касается событий, условий и действий ЕСА-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) [398, 404, 406].

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

Правила проверки условий и выполнения действий. Режимы связывания (Coupling modes). Режимы связывания определяют, как иницируется проверка условия в ответ на произошедшее событие и как планируется, диспетчируется и выполняется действие ЕСА-правила при положительном результате проверки условия.. Они впервые были исследованы в проекте HiPAC [397]. 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 [407] или выполнением специальных команд [396].
- *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 [397] 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) [408] были предложены еще два варианта режима связывания для поддержки побочных эффектов необратимых действий ЕСА-правил.

Запуск событием нескольких правил. Возможна ситуация, когда событие инициирует запуск нескольких правил. В этом случае были предложены механизмы планирования порядка выполнения правил [391, 394, 409].

Политика итогового эффекта (net effect policy). Для случая, когда в пределах одного правила выполняется несколько действий относительно одних и тех же данных предложена политика итогового эффекта [391], 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 ситу-

ации, когда выполнение внутреннего правила противоречит выполнению внешнего правила, а также возможны циклы, когда правило инициирует выполнение самого себя. Эти ситуации также рассматриваются в литературе [391].

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 [396, 410], Postgress [411], Ariel [412]. Примерами активного расширения реляционной модели данных являются работы [413-419]. Среди них особый интерес представляют те, которые исследуют взаимосвязь активных и дедуктивных баз данных [415-419].

В стандарт SQL 1999 г. были включены триггеры [420]. 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 заключается 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 [392, 395, 421], EXACT [407], NAOS [422], Chimera [423], Ode [424], SAMOS [401], Sentinel [406], REACH [408]. С кратким описанием этих систем можно познакомиться в [391], в этой же работе имеются ссылки на другие работы по активному расширению ООБД.

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 ЕСА-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

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.

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, который 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 в/из баз 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 80х 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" [425] (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 также нем 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 [434], 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,

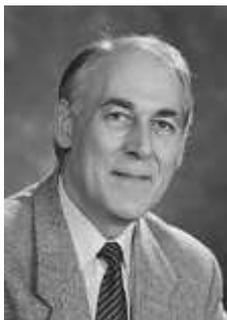
– *Формат обмена объектами (OIF - Object Interchange Format)* - язык описания загрузки и выгрузки текущего состояния СУОД в/из файлов и используются для обмена хранимыми объектами между СУОД.

– *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.

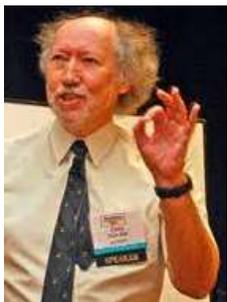
Второй манифест. Сообщество исследователей реляционных баз данных от-

ветило на активность в области OODB своим манифестом в поддержку ORBD. В 1990 г. М. Стоунбрекер и его коллеги по комитету перспективных систем БД опубликовали "Манифест систем баз данных третьего поколения" [435] (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. Стоунбрекера в университете Беркли (Калифорния, США) была разработана СУБД Postgres [436], 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 [437], также считающуюся одной из первых объектно-реляционных СУБД.



Хью Дарвен



Кристофер Дейт

Третий манифест.

В марте 1995 г. была опубликована статья Хью Дарвена (Hugh Darwen) и Кристофера Дж. Дейта (Christopher J. Date), названная авторами "Третьим манифестом" [438]. 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.

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).

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 [10].

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 [439-442], созданная в научно-исследовательском отделе корпорации Computer Corporation of America в конце 1970-х и начале 1980-х годов, система System R* [443-446], распределенная версия системы-прототипа System R, созданная в исследовательском отделе компании IBM в начале 1980-х годов, и система Distributed Ingres [447-449], распределенная версия прототипа системы Ingres, созданная также в начале 1980-х годов в Калифорнийском университете в Беркли. Также можно отметить проект POLIPHEME во Франции [450]. 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 [451-453].

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 [454, 455].

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 [456], 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 объединяются для облегчения совместного использования данных. Феде-

ративная 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. [457] и исследовалась во многих работах [458, 459]. В работе [460] приводится обзор по федеративным БД.

При существенном увеличении интегрируемых баз данных становится трудно, а иногда и невозможно определить глобальную интеграционную схему. *Мультибазы данных* [461] не предполагают существования глобальной схемы, но вместо этого язык запросов предоставляет возможность специфицировать выражения, позволяющие производить поиск по объединяемым базам данных.

Посредники. (Mediators) [462, 463] Они 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. (peer-to-peer databases - P2PDB) [464-466] Представляет собой совокупность автономных локальных репозиторий/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 [467, 469, 470, 10]. Одна из основных задач вертикальной фрагментации - определение наборов атрибутов, которые должны быть объединены в одну группу. В работах [471, 472] был предложен алгоритм энергетического связывания (bond energy algorithm) для группирования атрибутов, на основании которого производится вертикальная фрагментация. В работе [473] 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 [474]. В работах [475-477] была исследована проблема сложности задач размещения. В работах [478, 479] 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 [467, 468].

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 об-

зором исследований этого времени является статья [480].

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 [481] и предложил варианты немедленного (eager) и отсроченного (lazy) обновлений. Один из вариантов отсроченного обновления - использование первичной копии (master copy), когда основная копия обновляется оперативно, а обновление вторичных копий откладывается до удобного времени, причем синхронная репликация предполагает завершение распространения 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 [482], установлению ограничений на "устаревание" данных и обнаружению и устранению несогласованности [483].

Другое направление исследований, проводимых в контексте масштабируемой кластерной репликации, связано с разработкой методов обеспечения надежной согласованности при приемлемых затратах [484, 485]. С появлением облачных систем хранения в исследования были вовлечены внутриоблачные репликации, концептуально похожие на кластерную репликацию, а также междуоблачные и гео-репликации [486, 487]. Что касается распределенных систем, то сначала исследования были сконцентрированы на репликационных файловых системах [488, 489], затем на веб-серверных репликациях [490] и файловых репликациях одноранговых систем [491], Также были получены результаты по отказоустойчивым

репликациям объектов [492,493]. Как и в области баз данных, самые последние результаты, касающиеся репликации распределенных систем, относятся к облачной инфраструктуре, а именно, репликации в системах хранения [494], таких как HDFS и Cassandra, а также глобальная (wide-area) репликация [484, 495].

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 результате последующих действие самой транзакции, например, когда две транзакции ждут друг друга.

Имеются следующие категории алгоритмов обнаружения тупиков в РаБД [10]: централизованные, иерархические, распределенные. Централизованные алгоритмы [496, 497] используют центральный узел для обнаружения тупиков. Как отмечается в [439], централизованная двухфазная блокировка (two-phase locking - 2PL) и обнаружение тупиков являются хорошей естественной комбинацией. Централизованное определение тупиков впервые было реализовано в Distributed INGRES [498]. Иерархически алгоритмы [497, 499] для обнаружения тупиков полагаются иерархическую структуру узлов РаБД. Распределенные алгоритмы [499, 500] полагаются на кооперацию всех узлов РаБД для обнаружения тупиков. Распределенные определение тупиков впервые было реализовано в System R* [500].

В [10] представлен обзор методов управления распределенными тупиками. В обзорах [501-504] также обсуждаются различные распределенные алгоритмы обнаружения тупиков. В работе [505] приводятся сравнительный анализ дополнительных алгоритмов обнаружения тупиков: продвижение пути (path-pushing) [499], зондовый (probe-based) [506], глобального состояния (global state) [507, 508].

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 [509] (1976), Distributed INGRES [510, 511] (1977) и System R* [512, 513] (1981). Считается, что первым дистрибутивным алгоритмом оптимизации запросов является "скалолазание" (hill climbing) Вонга [514], 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 [510] 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 является динамическим.

Дистрибутивный алгоритм оптимизации запросов System R* [513] 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 [515] приведен обзор этих результатов. Было предложено несколько подходов по динамической оптимизации запросов для параллельных и распределенных баз данных [516]. Алгоритм в [517] предпо-

лагает изменение плана обработки запроса в процессе его выполнения, чтобы учитывать непредвиденные обстоятельства. В системе Mariposa [518] впервые была предложена экономическая модель оптимизации распределенного запроса. В монографиях [519, 10] детально освещаются результаты в области технологий распределенных баз данных и оптимизации распределенных запросов, полученные в 80-90-х годах. Статья [516] является более свежим обзором в этой области.

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 [520] по управлению параллелизмом в централизованных БД. Грей развил эти идеи для транзакций [521], а Спектор и Шварц [522] исследовали транзакции в распределенной среде.

Было предложено три механизма управления параллелизмом: блокировка, оптимистический протокол и упорядочение по временным отметкам (timestamp ordering)

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 [520]. Впоследствии было определено множество ее разновидностей: строгая (strict), консервативная (conservative), первичной копии (primary copy), распределенная (distributed), точная (rigorous). В [523] описан вариант 2PL, учитывающий использование старых значений. Также были предложены гибридные блокировки которые предполагают использование методов, отличных от 2PL [524-526].

Следующий тип управления параллелизмом получил название *оптимистический* в том смысле, что создаются локаль-

ные копии данных транзакции и обновляются именно они, а не сами данные. Впервые этот метод был предложен в работе [527], и с тех пор было исследовано множество его разновидностей [528, 529]

Наконец, *упорядочение по временным отметкам* 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 [530-532]. В [532, 533] также описываются многоверсионные временные отметки. Хорошим обзором по методам управления параллелизмом является статья [534].

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 [535].

В этот период были предложены решения, получившие названия *процессоров фильтров*, задача которых эффективно проверять передаваемые данные с дисков на

внешний сервер. В работах [536, 537] 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 [538] пионером исследований в области МБД был Даниэль



Даниэль Слотник

Слотник (Daniel L. Slotnick), опубликовавший в 1970 г. статью [539], 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) [540], Мински (Minsky) [541], Пархами (Parhami) [542]. На основе этой архитектуры были реализованы МБД RAP [543], CASSM [544], RARES [545].

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 [546], SURE [547].

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 [548], DIRECT [549], INFOPLEX [550], RDBM [551], DBMAC [552]

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 (header-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. ста-

тья [538] предвещала скорое исчезновение МБД.



Эсен Озкарахан

Наиболее известными монографиями по машинам баз данных первого этапа были написанные в 1986 г. Эсен Озкараханом [8], а также в 1990 г. Калиниченко Л.А. и Рывкиным В.М. [9].

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

Как отмечается в [553], 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 высокопараллельных архитектур.

В 1980-х годах исследования по машинам баз данных были сосредоточены на массивных параллельных вычислениях (massive parallel computing). Процессоры общего назначения и дисководы были со-

единены в узлы, и такие узлы затем были объединены в высокоскоростные межблочные связи [553, 554]. Некоторые из этих типов машин баз данных достигли большого успеха в промышленности.

В середине 80-х годов Стоунбрейкер предложил следующую простую классификацию параллельных мультимикропроцессорных систем [555]:

- *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 [556], DBS3 [557], Volcano [558].
- *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-архитектуры являются IBM IMS [559], Oracle Parallel Server [560], nCUBE [561], VAXclusters [562], IBM Parallel Sysplex [563].
- *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 [564], Gamma [565], Tandem [566], Bubba [567], Arbre [568] и nCUBE [569]. Примерами коммерческих систем SN-архитектуры являются NonStop SQL [570], Informix PDQ [571], NCR/Teradata DBC [572], IBM DB2 PE [573].

Вопросу анализа архитектур параллельных систем баз данных также посвящена статья Соколинского Л.Б. [574].

Со временем появились мультимикропроцессорные системы, которые сочетали характеристики SE- и SN-архитектур, поэтому Коупленд и Келлер [575] 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 [576]. Предложения Коупленда позволяют 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 [575, 577-580].

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 мультимикропроцессорную систему серии MBC-100/1000 [581], мультимикропроцессорную систему SP2 [582] компании IBM, компьютеры на основе технологии ServerNet компании Tandem [583], гибридную архитектуру CDN [584].

Согласно [585] 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 [586] 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 распределенную 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 диску. Поддерживает стандарт SQL.

Базы данных, поддерживающие работу с массивами

БД массивов дает возможность представлять и манипулировать многомерными массивами однородных данных. Считается, что предшественником БД массивов является созданная в 1982 г. PICDMS [587] - СУБД для работы с рисунками, которая предоставляет возможность оперировать двумерными массивами помощью процедурного языка.

В 1993 г. Майер и Вэнс [588] 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 БД массивов.



Питер Бауманн

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 [589, 590]. Разработанные

алгебра и язык запросов легли в основу созданной в 1996 г. под его руководством первой СУБД массивов RasDaMan [591], которая поддерживала реляционную модель данных с дополнительным типом данных "многомерный массив" и специальным языком запросов RASQL, базирующимся на SQL. Согласно данным [592] объем данных, хранящихся на всех установках RasDaMan, приближается к петабайту

Модели и языки. Было предложено множество формальных моделей и языков баз данных массивов, анализ которых можно найти в [593, 594]. Приведем некоторые из них.

Алгебра карт (Map algebra) [595, 596] - 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 [597] - эта алгебра разработана для обработки изображений и получения статистической информации

AML (Array Manipulation Language) [598] - универсальный язык манипулирования массивами, базирующийся на предложенной авторами алгебре многомерных массивов. Отличительной чертой AML является понятие битовых шаблонов и шаблонно-ориентированных функций.

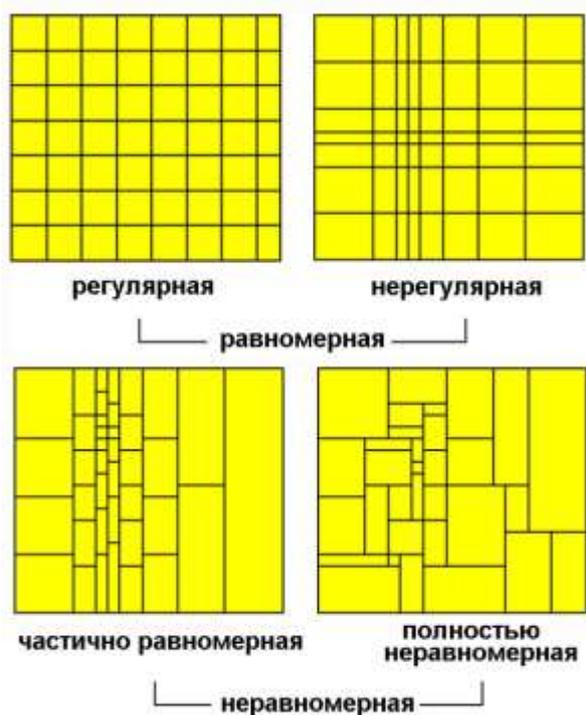
AQL (Array Query Language) [599, 600] - этот язык встраивает поддержку многомерных массивов в язык NCRA, который является расширением языка вложенного реляционного исчисления NRC.

Array Algebra [589, 590] - 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 [601, 602] - модель разработана в качестве расширения реляционной СУБД MonetDB [603].

Хранение массивов. Обычно большие многомерные массивы разбиваются на под-

массивы, которые образуют единицы доступа к ним. Такое разбиение получило название *мозаики* (tiling), а элементы мозаики - *плитки* (tile) [618]. Мозаика состоит из непересекающихся плиток, каждая плитка - многомерный подмассив исходного массива. Выделяют два основных вида 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 [618], 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 [591], SciDB [604], MonetDB/SciQL [605]);
- реализованные в виде дополнительных уровней в существующих СУБД (EXTASCID [606, 607]);
- реализованные в виде объектно-реляционных расширений (PostGIS Raster [608] Teradata Arrays [609], Oracle GeoRaster [610]),

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 [611]:

- модель данных основывается на многомерных массивах, а не на кортежах;
- модель хранения базируется на версииности, 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 [612]. Ее архитектура основана на модифицированном ядре Postgres. SciDB предназначена для хранения, обработки и анализа сверхбольших объемов многомерных распределенных массивов научных данных, масштабируемых на тысячи серверов [613]. Хранение данных организовано в виде многомерных вложенных массивов, для обработки которых раз-

работаны языки AQL (Array Query Language) и AFL (Array Functional Language).

Другие системы БД массивов.

SciQL [605]. Основанный на SQL и использующий массивы язык запросов для научных применений. Расширяет колончатую СУБД MonetDB операторами над массивами [614, 615], позволяя тем самым MonetDB эффективно функционировать как база данных массивов.

EXTASCID [606, 607]. 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 [608] (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.

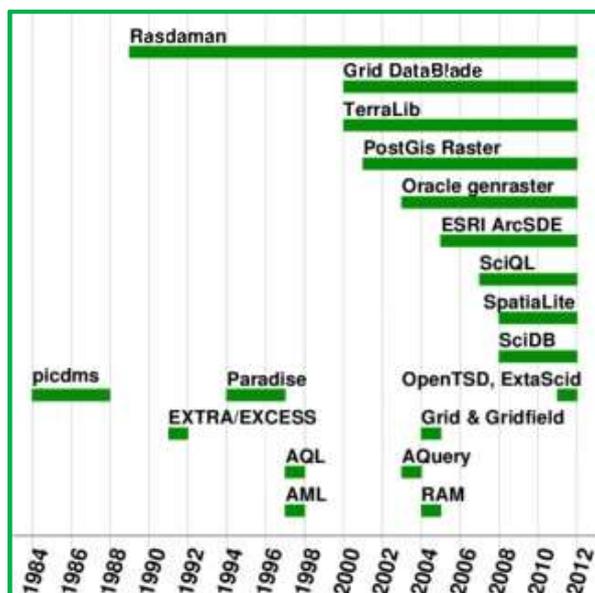
Oracle GeoRaster [610] — это встроенная в Oracle Spatial возможность по хранению, индексированию, анализу и доставке растровых изображений (например, спутниковых снимков), данных типа grid-данных, а также связанных с ними метаданных. Эти типы данных можно использовать для хранения многомерных grid-слоев и электронных изображений, которые могут быть привязаны для позиционирования на поверхности Земли или в локальной системе координат.

Teradata Arrays [609]. Недавно Teradata вела в свою СУБД массивы в виде самостоятельного типа данных.

В 1918 г. в ISO SQL была включена поддержка многомерных массивов данных [617] в виде специального типа данных.

В заключение отметим, что Альянс по Исследовательским Данным (RDA - Research Data Alliance) представил в 2021 году исчерпывающий обзор по базам данных массивов и связанными с ними технологиями [594].

В качестве иллюстрации истории развития систем БД массивов приведем рисунок, взятый из [616].



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 статистических баз данных 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 [630]. В СБД определено три типа моделей данных: *графические*, *табличные* и *многомерные*. Дадим краткое описание статистических моделей

данных, подробная информация о которых приведена в работах [619, 620]:

- SUBJECT [621] - представлена графическая модель системы SUBJECT;
- SAM (Semantic Association Model) [622] - модель была разработана для моделирования как научных статистических данных, так бизнес-ориентированных данных;
- GRASS (Graphical Approach for Statistical Summaries) [623] - является расширением SUBJECT. Для представления модели используется ориентированный, направленный, ациклический граф;
- CSM (Conceptual Statistical Model) [624] - используются две разные, но дополняющие друг друга модели данных для описания элементарных и сводных данных, а именно ER-модель Чена и переопределенную модель GRASS;
- STORM (Statistical Object Representation Model) [625] - графическая модель, в которой логическое представление отделено от физической структуры статистических таблиц;
- MEFISTO [626] - функциональная модель данных, 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 [627];
- Темпоральная статистическая модель данных [628].

Статистические операторы (алгебры). В статьях по СБД предлагается много подходов по определению операторов, которые бы соответствовали выбранной структурной модели. В работе [629] 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 [630] 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 [631], в которой используется двумерная модель представления статистических данных. В 1997 г. в работе [632] был предложен вариант расширения SQL функциональными возможностями OLAP для получения итоговых значений в многомерном пространстве. В работах [633, 634] предлагаются многомерные модели данных и операторы. Статьи [626-635] также посвящены операторам в СБД. В работе [636] представлена алгебра статистических данных.

Метаданные. Считается, что статистические данные имеют два типа атрибутов [637]: *сводные атрибуты*, представляющие собой результаты применения к исходным данным агрегирующих функций, и *дескриптивные атрибуты*, которые эти сводные данные описывают, также называемые 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 [638-642].

Системы и языки запросов статистических баз данных. В обзоре [643] дается всесторонний анализ систем и языков запросов статистических баз данных на основании таксономии, предложенной в [644]. Перечислим их, отсылая заинтересованных читателей к указанным статьям для детального ознакомления.

1) Статистические системы управления базами данных (ССУБД), построенные на основе традиционных СУБД. Большинство ССУБД данной категории построены на основе реляционных СУБД. К ним относятся:

- *STRAND* [645] базируется на ER-модели Чена, является производным от *CABLE* [646] и построен на основе реляционной СУБД *INGRES*. Запросы *STRAND* транслируются в язык *QUEL* и выполняются в *INGRES*.
- *HSDB* [647] является ССУБД, построенной на основе реляционной системы *Model 204* [648]. *HSDB* поддерживает

сводные таблицы и предоставляет ограниченный набор операций над ними. Может выполнять процедуры статистического анализа над реляционными и сводными таблицами

- *Расширенная РМД* [649]. 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* [650] Объектно-ориентированная система управления базами знаний, построенная на основе системы SQL/DS. Имеет развитые возможности по управлению метаданными и ограниченный перечень статистических функций.
- *GRAFSTAT* [651]. 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* [652], *PASTE* [653], *GPI* [654], *PEPIN-SICLA* [655] являются примерами систем, которые используют традиционную СУБД, статистический пакет и графический пакет для создания системы управления статистическими данными.

2) 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* [656] и *CAS SDB* [657], используют реляционную алгебру.
 - *ABE* [658], использует реляционное исчисление.
 - *SIR/SQL* [659], *GENISYS* [660], *CANTOR* [661] - используют SQL.

- В [662] представлен язык запросов статистической обработки неполной информации
- В системе July [663] используется универсальный реляционный интерфейс для интерпретации статистических запросов
- В статье [664] описана статистическая модель данных и ее применение в СБД.

- Системы на базе иерархической и сетевой моделей. Примерами являются: *SIR/DBMS* [665], *TPL* и *TPLDCS* [666], *BROWSE* [667].
- Формальные расширения реляционной модели: *ABE* [658], *SSDL* [668], *SSDB* [669].
- ССУБД и языки с графическим внешним интерфейсом. Системы данной категории имеют графические двумерные или табличные языки запросов. Примерами являются: *SUBJECT* [621], *GRASS* [623], *ABE* [658], *GUIDE* [670], *STBE* [671], *ALDS* [672], *GRASP* [673].
- Естественный язык интерфейса пользователя: *LIDS 86* [674].
- Языки запросов, которые вычисляют агрегированную информацию из темпоральных данных. Примерами являются *TQUEL* [675], *HQUEL* [676], *TBE* [677], *TEER* [678], расширенная реляционная алгебра Тансела [679].

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

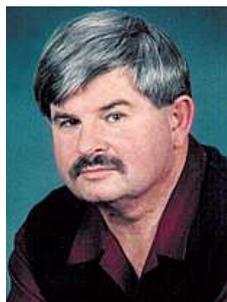
Мерал Озсойоглу специализируется по базам данных. В 2011 г. она получила звание "Действительный член ACM" (ACM Fellow) за "большой вклад в системы управления базами данных". В 2018 г. она получила премию ACM SIGMOD Contributions Award за "преданное слу-

жение 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.

Хранилища данных

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

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 [682], которая стала библией хранилищ данных и которая



Вильям Инмон

положила начало развитию индустрии хранилищ данных. Он организовал первую конференцию, впервые организовал тематический раздел в журнале.

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 стан-

дартную реляционную базу данных на основе SQL для приложений DWH и бизнес-аналитики. Его монография [683], первое издание которой было в 1997 г., является бестселлером по настоящее время.

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 [684] и многих других исследователей их творческого наследия

К пионерам исследователей по DWH, которые опубликовали свои монографии в середине 90-х годов, также относятся Брэккетт [685], Гилл и Рао [686], По [687].

Согласно определению Инмона *хранилище данных - это предметно ориентированная, интегрированная, поддерживающая хронологию и неизменяющаяся (постоянная) коллекция данных, созданная для поддержка процесса принятия решений руководством* [682] Также считается, что в широком 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 осуществ-

ляют 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 [688], 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



Найджел Пендс

Кодда, сформулировал альтернативные 5 правил принадлежности систем к категории OLAP [689], 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.

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" [690].

Архитектура DWH. Было предложено множество различных архитектурных решений DWH [691-702], каждая из которых обладает своими специфическими особенностями. В работе [703] проведен анализ 73 архитектур DWH. На базе предложений, высказанных в [691, 692], представим

обобщающую архитектуру DWH. DWH имеет трехуровневую архитектуру.

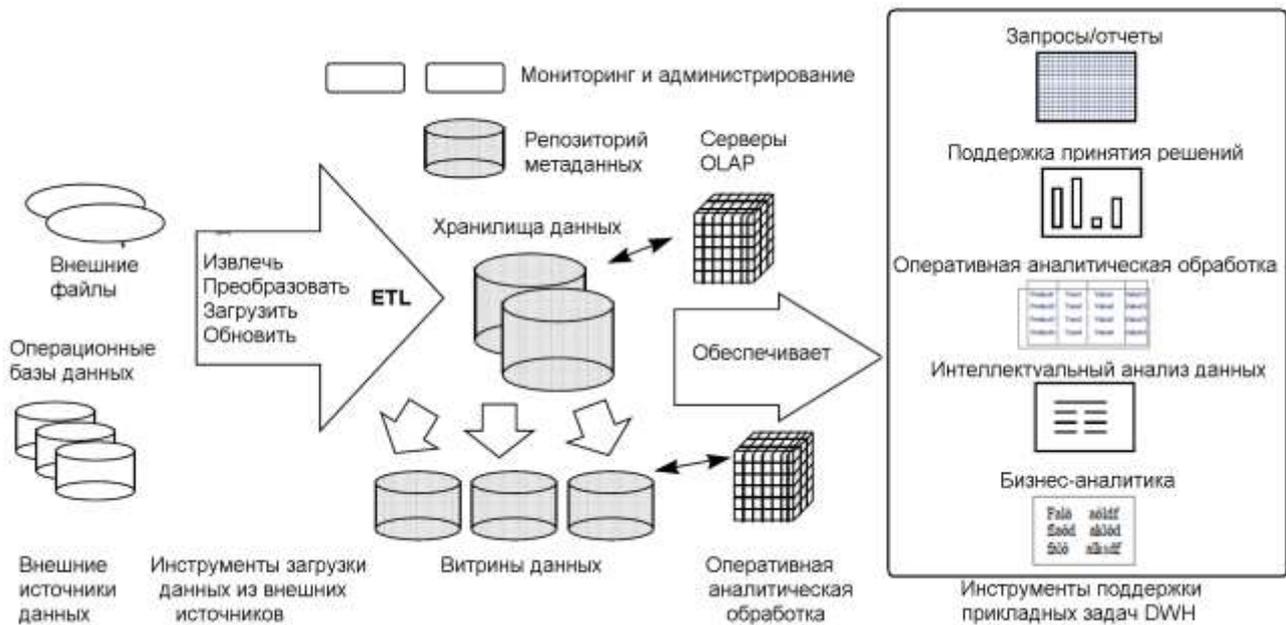
- Нижний уровень представляет собой БД DWH. Она поддерживает выбранную модель данных DWH и предоставляет средства ведения этой БД.
 - Средний уровень выполняет функции OLAP. Он обычно представлен следующими 4 типами [693, 694]:
 - *Реляционный OLAP (ROLAP)* – расширенная реляционная СУБД, которая отображает 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, Transform, Load), которая выполняет функции извлечения данных из источников, их проверки и очистки, преобразования к нужному виду, интеграции и загрузки или обновления БД DWH [704]. Концепция ETL возникла в 1970-х годах в связи с использованием централизованных репозиториях 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 [705]:

- *корпоративное хранилище*, содержащее консолидированные данные, извлеченные из нескольких операционных источников - это DWH всей корпорации [706];
- *витрина данных* - содержит подмножество корпоративных данных;
- *виртуальное хранилище* - это множество взглядов (views) операционных БД [707, 708].

Также существует точка зрения [709], 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.) [710], 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 [711]. Кратко опишем одну из них, которая является наиболее используемой, а именно, куб данных [712].

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

Согласно [713] 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.

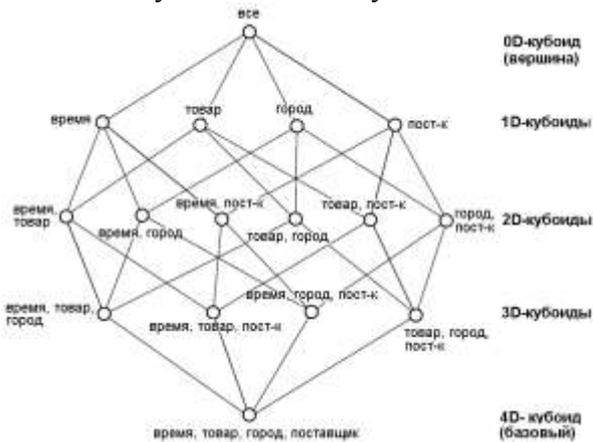
Мера (measures) – это собственно агрегированные значения Меры располагаются в ячейках куба. Выделяют три типа числовых мер. *Аддитивные меры (additive measures)* могут быть агрегированы по всем измерениям. *Полуаддитивные (semi-additive)* могут агрегироваться только по некоторым измерениям, а *неаддитивные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

Многомерная модель имеет графическое представление, которое получило название *кубоида (cuboid)* [714]. Кубоид, содержащий самый низкий уровень агрегированных данных (то есть агрегированные данные по всем измерениям), называется *базовым кубоидом*. На рис. ниже приведен трехмерный базовый кубоид.

Город	Товар							
	Монитор	Клавиатура	Память	Процессор				
Чикаго	854	882	89	623				
Нью-Йорк	1087	968	38	872				
Торонто	818	746	43	591				
Ванкувер								
Время (квартал)	Q1	605	825	14	400	682	925	698
	Q2	680	952	31	512	728	1002	789
	Q3	812	1023	30	501	784	984	870
	Q4	927	1038	38	580			

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)* [715] и называется *кубом*. На рис. ниже приводится куб (решетка кубоидов) для четырехмерного базового кубоида с измерениями: товар, время, город, по-

ставщик (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 агрегации.

Такие функции включают: функции агрегирования, статистические функции, функции ранжирования Top N, Bottom N и другие. В [716] 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 [715, 717]. Кроме того, исчерпывающий обзор по реализации ROLAP-кубов дан в [718].

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 [719, 720].

Многомерные базы данных (МБД), это разновидность БД, которая создается для хранилищ данных и оперативной аналитической обработки данных (OLAP). OLAP, работающие с МБД называются многомерными OLAP (MOLAP). Как правило, МБД используют модель многомерных кубов для представления исходных данных. В [721] 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 [714].

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. Графическое 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.
- **Аккумулярующие 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 [722].

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 более по-

лутора десятков из которых приведен в работе [722a].

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.

Активные DWH. В начале этого столетия была предложена концепция активных DWH. [723, 724] 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 [725, 726].

DWH реального времени [727]. Эта концепция предполагает, что исходные данные поступают в DWH сразу же, как только они были порождены их источником и становятся доступными для их анализа. О таких системах говорят, что они являются DWH "с нулевой задержкой". Популярность данной концепции привела к тому, что многие производители, включая IBM [728] и Oracle [729] начали производить DWH этого класса. Краткий анализ исследований по этому направлению приведен в [730].

Эволюция 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. Кимбалл, вероятно, был первым, кто обратил внимание на это в 1996 г. и предложил ряд решений [731]. 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 [732].

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 [733], концептуальное моделирование и проектирование [734], логическое моделирование и запросы [735, 736], задержка в получении измерений [737], многомерная агрегация [738], корректная агрегация при наличии изменений в данных и структуре [739], эволюция многомерных схем [740]. В работе [741] 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 [742] впервые было введено понятие пространственного OLAP (SOLAP), отражающее применение методов интеллектуального анализа к обработке пространственных данных. В работах [743, 744] было введено понятие пространственных измерений и предложена их классификация. В статье [745] 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 [742, 743, 745, 746] и совокупности точек [744].

Обычно SOLAP применяют к дискретным пространственным данным, однако многие сложные задачи ГИД-анализа предполагают использование непрерывных про-

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 [747], в которой предлагался куб данных с непрерывными измерениями. В статье [748] также предлагается многомерная модель данных с непрерывными измерениями и с набором операций, которая может использоваться для OLAP-анализа полевых данных. В работах [749–751] 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 [752]. В статье [753] 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 [754]. CUBE BY создает группирование по всем возможным комбинациям указанных в нем измерений с разными уровнями агрегации данных. Эта идея была принята 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.

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 онтологий. Одним из вариантов хранения онтологий стали онтологические базы данных. Наконец, в этот период стали бурно развиваться полнотекстовые БД и электронные библиотеки. Краткому анализу этих составляющих 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 и

использовался для описания реляционной БД, разработанной Карло Строщи (Carlo Strozzi) [755], которая не использовала SQL



Карло Строщи

в качестве языка запросов. Это первоначальное использование данного термина ничего общего не имеет с современной технологией NoSQL. Вместе с тем, в первой десятилетие 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 [756]. Главным стимулом встречи явилось появление на рынке распределенных нереляционных



Эрик Эванс

продуктов. В качестве яркой вывески семинара Эрик Эванс (Eric Evans) предложил емкий и лаконичный термин "NoSQL" [757]. 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.

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).

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 распределенных вычислений возможно 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 распределенной системе завершается корректным откликом, однако без

гарантии, что ответы всех узлов системы совпадают;

- **устойчивость к разделению (Partition tolerance)** — разделение распределенной 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. [758] и затем в 2002 г. это утверждение доказали Сет Гильберт (Seth Gilbert) и Нэнси Линч (Nancy Lynch) [759].



Эрик Брюер

Теорема впоследствии получила широкую популярность и признание в среде специалистов по распределенным вычислениям. Концепция NoSQL, в рамках которой создаются распределенные 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.

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

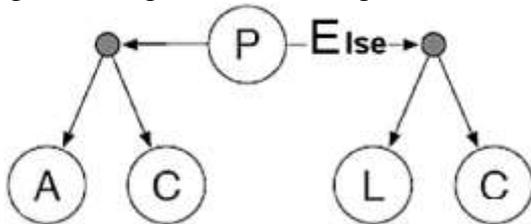
Теорема PACELC. Эта теорема, как и CAP, описывает, какие ограничения и компромиссные решения имеют распределенные системы в отношении согласованности, доступности и устойчивости к разделению. Однако, теорема PACELC дополнительно утверждает, что необходимо идти на компромисс между задержками получения ответа и согласованностью даже при отсут-

ствии 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)onsistency

Простыми словами, при условии наличия устойчивости к разделению (P) можно выбрать доступность (A) или согласованность (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).



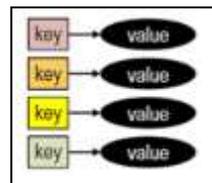
Даниэль Дж. Абади

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 [760]. В 2015 г. он был награжден премией Very Large Data Base Endowment Inc. (VLDB) за лучшую статью за предыдущие 10 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 различные типы NoSQL баз данных с указанием нескольких из них, принадлежащих каждому из типов. В работе [761] приводятся множество различных систем классификации NoSQL с обширным указанием 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 [762, 763]. Работа [764] посвящена исследованию применению этой модели в базах данных.



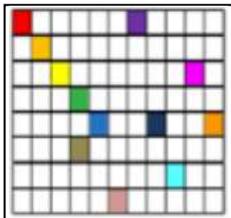
БД ключ-значение (key-value database). Ассоциативные массивы легли в основу так называемых баз данных ключ-значение.

Считается, что ключ должен быть уникальным. Согласно [765] первой системой баз данных типа "ключ-значение" была созданная в 1986 г. система GT.M (Greystone Technology M) [766], 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" [767] 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»

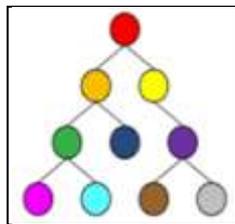
Модель триплетов (triple store model). Расширением ассоциативного массива стал модель триплетов – три элемента, связанные в выражение «субъект–предикат–объект» [768] или в классический элемент информации «объект-атрибут-значение». По существу триплет – это одна ячейка классического отношения. Благодаря указанию свой-

ства, для которого приведено значение, триплеты являются более информативными, чем пары «ключ-значение». Каждому объекту может соответствовать столько триплетов, сколько свойств имеет данный объект.

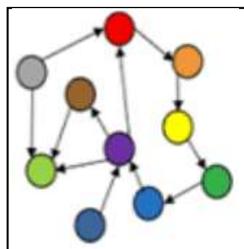
По адресу https://en.wikipedia.org/wiki/Comparison_of_triplestores приведен обширный список баз данных различных хранилищ триплетов.



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 ин-

формации, 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 [2], 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 [769] 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;
- объектные БД;
- БД сетей;
- облачные БД;
- БД XML
- многомерные БД;
- многозначные БД;
- БД источников событий (event sourcing);
- БД временных последовательностей/поточные БД (time series/streaming);
- научные и специализированные БД.

Языки запросов. Было предложено множество NoSQL языков запросов, в частности AQL - ArangoDB Query Language, CQL - Cassandra Query Language, HQL -

Hypertext 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) [770]. 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 Гиппа (Richard Hipp), создателя SQLite, и Дэмиена Каца (Damien Katz), основателя проекта CouchDB. С обзором всех этих языков можно познакомиться в [771]. Было предложено много графовых моделей баз данных, обзор которых приведен в [772].

Документо-ориентированные базы данных

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. Термин "слабоструктурированные данные" ввел Луневский и др. (Luniewski) в 1993 г. в системе Rufus [775]. В 1995 Папаконстантиноу и др. (Papakonstantinou) определил модель для слабоструктурированных данных OEM в рамках системы интеграции гетерогенных баз данных TSIMMIS [776, 777]. В 1996 Бунеман и др. (Buneman) определил модель слабоструктурированных данных [778]. В 1999 г. Дойч и др. (Deutsch) описал связь между слабоструктурированными данными и XML [780].

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 самой структурой. В 1995 г. была разработана система и язык Logel [781] - 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 [779] делается акцент на трансформациях запросов и вводится структурная рекурсия в качестве центральной парадигмы преобразования слабоструктурированных данных. В языке MSL [782], вводятся сколемовские функции для преобразования слабоструктурированных данных. Язык XML-QL [780] 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 [783] приведено краткое описание более 60 систем ДБД, среди которых по состоянию на 2021 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

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.

Колонки логически группируются в *колоночные семейства* (column family), которые также называются широкими колонками (wide-column). 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 от хранилищ семейства колонок. В первом случае предполагается, что каждая колонка хранится самостоятельно не зависимо от других колонок, а во втором - все колонки семейства запоминаются вместе.

Супер-колонка - это колонка, состоящая из других колонок. Например, супер-колонкой является ФИО, состоящая из колонок Фамилия, Имя и Отчество. С другой стороны, Фамилия, Имя и Отчество - это семейство колонок. Таким образом, семейство супер-колонок, 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 супер-колонок 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) [784], ориентированная на хранение и поиск биологических данных. К этому классу также относятся разработанная в 1975 г. медицинская система TOD [785] и система RAPID [786], 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 [787]. Одной из самых ранних систем, которая обладала чертами современных колоночных СУБД, была Cantor [788]. Были проведены исследования по организации поиска в транспонированных файлах [789]. В 1975 г. в работе [790] 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 по-атрибутную 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 [791], в которой в качестве альтернативы NSM была предложена модель декомпозированной памяти (Decom-

position Storage Model, DSM). В DSM каждая колонка таблицы запоминается отдельно, а чтобы знать, какой именно строке таблицы принадлежит значение в колонке, вме-

UID	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

сте с этим значением запоминается уникальный идентификатор строки, как правило это суррогатный первичный ключ (см рис. ниже).

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

б) Модель памяти DSM

На протяжении последующих 20 лет использовались именно термины NSM/DSM для указания строчной и колоночной моделей памяти. Последующие две статьи, посвященные исследованию проблемы распараллеливания операций работы с DSM [792], а также использованию индексов для выполнения операций соединения и проекции [793] 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 [794, 795], VectorWise (MonetDB/X 100) [796] и C-Store [798].

Первыми коммерческими КБД считаются Sybase IQ (1996) и KDB (1998). Вторая половина 2000-х годов ознаменовалась 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 гибридные 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 опти-

мизации 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 [795]. 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 [796, 800], которые существенно повышают их производительность. Блочная итерация [797] 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. [801, 802]. 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 [800, 803]. В связи с тем, что колонки содержат данные одного типа (атрибута), достигаются хорошие показатели сжатия с помощью простых алгоритмов. Было предложено множество алгоритмов сжатия для колоночных хранилищ [804].

Оперирование сжатыми данными. Во многих современных колоночных хранилищах распаковка данных откладывается до тех пор, пока это не станет абсолютно необходимым [800, 801], 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 формиро-

вать из них кортежи (производить их соединение).

Эффективная реализация операции соединения. Так как КБД предполагают по-колоночное представление данных, важным является вопрос использования различных стратегий выполнения операции соединения. В КБД используются как классические алгоритмы соединения, так и специфические [800, 805, 806].

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 [798] 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 - это принципиально новый подход в базах данных, который основывается на принципе, что создание и ведение индекса является продуктом деятельности по обработке 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 [809]. 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 [804].

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 [798, 810]. Например, в C-Store [798] сначала данные записываются в несжимаемый оптимизированный для записи буфер и затем периодически "сбрасываются" в большие сжимаемые пакеты.

Были проведены сравнительные исследования строчных и колоночных хранилищ [806-808]. В частности, результаты исследований [806] 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 [804]. Также рекомендуем статьи [811-813]. Прекрасным учебным материалом является пособие [814].

Завершим изложение колоночных СУБД следующим высказыванием, взятым из

[804]: "*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*".

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.

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 слож-

ность, 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. [825].

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) [835, 839, 840, 848]. Обычно метки именуется 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 связана несколькими дугами.

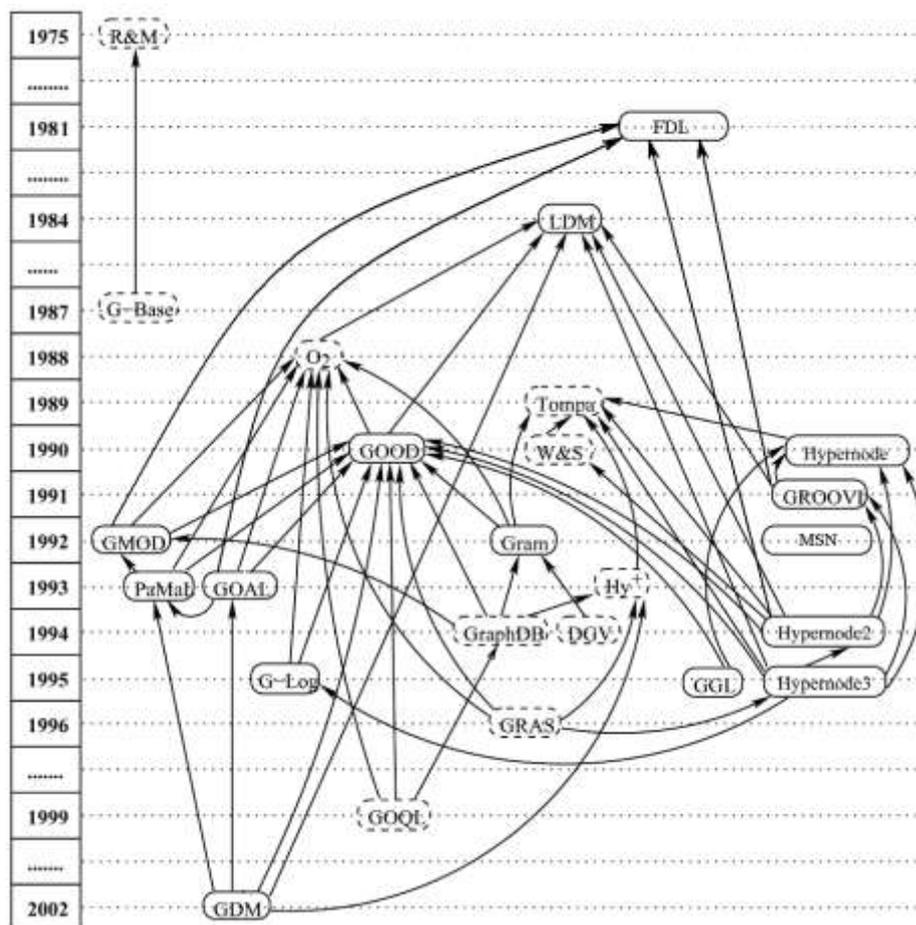
Графовая модель со свойствами (Property Graph Model). Граф со свойствами - это направленный помеченный атрибутивный мультиграф. Понятие графа со свойствами было введено в работе [827], а его формальное определение дано в [824]. Вершины/дуги могут обладать многими свойствами (атрибутами), которые позволяют указать их характеристики. Свойства задаются в виде пар ключ-значение. Языками графовых моделей со свойствами являются G-CORE [823], PROPER [854], Gremlin [885], Cypher [887], PGQL [891].

Гиперграфовая модель - сложные дуги. Гиперграф (hypergraph) - это обобщение графовой модели данных, в которой дули могут соединять любое количество вершин, как в начале дуги, так и в ее конце [828]. Такие дуги называются гипердугами (hyperedge). Гиперграфы оказываются полезными, когда данные содержат связи типа "многие-ко многим". Гиперграфовые модели представлены в работах [834, 836, 838, 844].

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 [837], а затем уточнена этими же авторами в [847]. Данной модели также посвящены статьи [838, 849, 850]. Аналогичная идея предложена в мультимасштабных сетях (multi-scaled networks) [841].

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 [819] дается детальный обзор графовых моделей данных до 2003 года. На рисунке ниже, взятом из [819], 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 [829] - введено понятие семантической сети для хранения информации о данных базы данных.
- FDL [830] - в рамках функциональной модели данных неявно определена графовая структура данных, целью которой было обеспечение "концептуально естественного" интерфейса базы данных.
- LDM [831] - в рамках логической модели данных явно определена графовая модель базы данных целью которой - обобщение реляционной, иерархической и сетевой моделей данных
- G-Base [832] - предложена графовая модель данных, названная G-Base, для представления сложных структур
- O2 [833], - определена объектно-ориентированная модель данных O2 на основе графовой структуры.
- Tompa [834] - гиперграфовая модель данных для гипертекстовых баз данных.

- GOOD [835] - граф-ориентированная объектная модель, предназначенная для систем с графоподобными средствами представления и манипулирования данными.
- W&S [836] - гиперграфовая модель для доступа к данным в базе данных
- Hypernode [837] - граф-ориентированная модель с гипервершинами, представляющими собой вложенные графы
- GROOVY [838] - объектно-ориентированная гиперграфовая модель данных
- GMOD [839] - предложен ряд концептуальных положений относительно интерфейсов пользователя в граф-ориентированной базе данных
- Gram [840] - граф-ориентированная модель данных для представления гипертекстов.
- MSN [841] - Графовая модель мультимасштабных сетей, объединяющая основы теории графов и объектно-ориентированную парадигму.

- PaMaL [842] - является расширением GOOD с явным представлением кортежей и множеств
- GOAL [843] - граф-ориентированная объектная модель с ассоциативными вершинами
- Ну+ [844] - гиперграфовая модель с языками запросов и визуализации
- GraphDB [845] - графовая модель данных и язык запросов для баз данных
- DGV [846] - графовая модель для определения и манипулирования графами различного вида, хранимыми в реляционных или объектных базах данных.
- Hypernode2 [847] - модель со вложенными графами, являющаяся развитием Hypernode
- G-Log [848] - граф-ориентированная модель и декларативный язык запросов.
- GGL [849] - теоретико-графовая модель данных баз данных карт генома
- Hypernode3 [850] графовая модель данных, являющаяся развитием Hypernode
- GRAS [851] - графовая атрибутивная модель для представления сложной информации
- GOQL [852] - объектно ориентированная графовая модель данных и графовый язык запросов
- GDM [853] - 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 [819] дается обзор ГЯЗ периода первой волны исследований и разработок в области ГБД. На ее основании в статье [855] был исследован вопрос выразительно мощности и вычислительной сложности некоторых из этих языков. В статье [821] приводится сравнительный анализ языков функционирующих на то время ГБД. В статье [856] исследуется проблема формулировки запросов к ГБД и, в частности, выразительность и сложность навигационных языков запросов. Наконец, в работе [822] дается обзор основополагающих особенностей современных

ГЯЗ, а в [825] 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 [857] исследуется вопрос эффективного выполнения операции смежности в больших динамических разреженных 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 [858].

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 [861, 862], биоинформатике [863], семантическом вебе [864]. Как показано в [859, 860] 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 количество вершин и дуг, и регулярные простые пути (regular simple paths), в которых накладываются ограничения (регулярные выражения) на вершины/дуги. В статье [861] дается обзор различных теоретико-графовых задач, связанных с путями, имеющих отношение к базам данных, включая вычисление транзитивного замыкания, выполнение рекурсивных запросов и сложность поиска путей. В статье [865] вводится понятие запро-

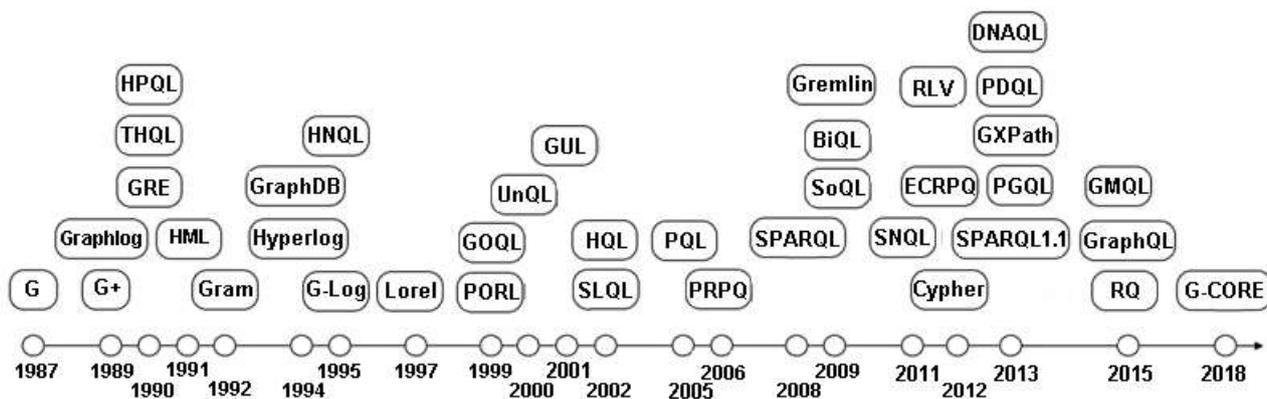
са с регулярным путем (Regular Path Query - RPQ) в качестве способа выражения запросов на достижимость, а в статье [866] анализируется проблема достижимости с учетом наличия регулярных выражений. Этот тип запросов исследовался также в работах [856, 867]. Одной из разновидностей задачи достижимости является нахождение кратчайшего пути, если их несколько, например [868].

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 [859], сложные аналитические запросы тесно связаны с алгоритмами интеллектуального анализа графа данных.

Приблизительное сопоставление и ранжирование (approximate matching and ranking). Возможны ситуации, когда поль-

зователи 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 будут 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 [869], в которой исследуется вопрос неточной отработки запроса на основе специального вида соответствия между запросом и графом. В работе [870] 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 [825], 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 [865], G+ [871], Graphlog [872], HPQL [837], THQL [836], GRE [873], HML [838] Gram [840], Hyperlog [847], GraphDB [845], G-Log [848], HNQL [850], Lorel [874], PORL [875], GOQL [852], UnQL [876], GUL [877], SLQL [878], HQL [879], PQL [880], PRPQ [881], SPARQL [882],

SoQL [883], BiQL [884], Gremlin [885], SNQL [886] Cypher [887], ECRPQ [888], RLV [889], SPARQL 1.1 [890], PGQL [891], GXPPath [892], PDQL [826], DNAQL [893], RQ [894]. GraphQL [895], GMQL [896], G-CORE [823].

Для ГБД принципиальным является эффективное выполнение запросов [897, 898]. Для этого были разработаны различ-

ные методы их индексирования [899-901] и оптимизации [902-904].

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

- <https://hostingdata.co.uk/nosql-database/>, на этом сайте, посвященном NoSQL-базам 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 гра-

фовой 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: [817, 821, 905-907]. Ниже при-

водится таблица сравнительного анализа 20 ГБД, взятая из [817]

Графовые базы данных и инструментальные средства	Модель данных				Способ хранения		Возможности запросов			Модель вычисления	
	Простой граф	Граф со свойствами	Гиперграф	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
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.

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 [908], в которой отмечается, что аппаратные предположения, 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 [909] - распределенная база данных, полностью находящаяся в памяти, и C-Store [910] - колоночная база данных. В 2010 г. Рик Кеттелл (Rick Cattell) опубликовал статью [911], 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) [912, 913]. И с тех



Мэтью Аслет

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.

Были предложены варианты классификации NewSQL-баз данных [915, 916]

Впоследствии были проведены исследования, в результате которых появилось



Гай Харрисон

множество публикаций по сравнительному анализу технологий SQL, NoSQL и NewSQL, фундаментальным трудом в этом направлении стала монография Гая Харрисона (Guy Harrison) [12].

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.

Онтологические базы данных

С появлением семантического веба появилось понятие онтологии. Было предложено несколько определений онтологий, наиболее полное и общепринятое следующее [917]:

Онтология - это явная формальная спецификация согласованной концептуализации. Детальное раскрытие этого определения дается в [918]

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 [919- 922]. С их обзором можно познакомиться в [923]. Как правило, ОнБД создаются на базе реляционных баз данных. Кроме того, в качестве языка представления онтологий нижнего уровня выбирается RDF.

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 [924, 925], 3store [922], Rstar [926], Virtuoso [927], Oracle [928]. Существенное преимущество - простота поддержки модели.

Схемная модель (schema-aware) также называемая бинарной (binary). Каждый класс и каждое свойство онтологии (RDF/С-схемы) имеет свою собственную таблицу [920, 921, 929, 930]. 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 [931].

Расширенным вариантом схемной модели является **дуальная (dual) модель**, 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 подход используется в ОнБД OntoDB [932]

Гибридная модель (hybrid) [929] 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 становятся 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 [933], OntoDB [932] и Jena2 [934]. 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 независимо 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 [935].

Вывод. 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 [936]. 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), дедуктив-

ные 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 [937] описывается язык OntoQL, который был разработан для ОнБД OntoDB. Кроме того, в этой статье определена алгебра онтологий, на базе которой построен язык OntoQL.

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 во 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 Дублинскокогго 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 дополнительные возможности, например:

- не учитывать незначимые 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 средств 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 направо, 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 пределах каждого из этих классов будем приводить их разновидности.

Алгоритмы точного сопоставления

Они предполагают точное соответствие текста образцу. Известны следующие пять вариантов: символьный, с хешированием

ем, 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 – КМП) — 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) [946] и независимо от них исследован Дональдом Кнутом (Donald Knuth) [947]. Наконец эти три ученых опубликовали совместную статью в 1977 г. [948]. Независимо от них в 1971 г. советский ученый Юрий Матиясевич предложил аналогичный алгоритм при исследовании задачи сопоставления строк в двоичном алфавите [949].

Алгоритм Бойера-Мура (Boyer-Moore – ВМ). В 1977 г. был предложен алгоритм сопоставления Бойера-Мура (ВМ) [950]. 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 ВМ 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 [16а]. Алгоритм ВМ подробно описан в [952]. Со временем появилось множество алгоритмов, которые либо модифицируют ВМ, либо объединяют его с другими подходами. К модифицирующим вариантам ВМ, которые улучшают некоторые его характеристики относятся алгоритмы Хорспула (Horspool) [186], Чжу-Такаока (Zhu-Takaoka) [954], Turbo-ВМ [955], Апостолико - Джанкарло (Apostolico and Giancarlo) [957] Смита (Smith) [958], Райта (Raita) [959], Крочемора [960], Берри-Равиндрана [961]. В свою очередь гибридные ВМ-подходы предполагают интеграцию ВМ-алгоритма с другими методами с целью повышения производительности. Они описаны в статьях [962-968].

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).[969]. 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 по-

линомиальный хеш, являющийся разновидностью кольцевого.

Хорошее описание алгоритма Рабина-Карпа приведено в статье [970]. Было предложено еще несколько алгоритмов поиска с хешированием, которые описаны в [971-973].

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 сравнении образца с подстрокой текста. Примерами являются алгоритмы, описанные в [974, 975].

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) [976], представляющий конечное множество строк, и суффиксный автомат, выполняющий функцию суффиксного индекса [977].

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 [978]. 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 сопо-

ставления. Были исследованы следующие разновидности DAWG-сопоставлений: обратный недетерминированный (backward non-deterministic) DAWG [979], двойной прямой (double-forward) DAWG [980], алгоритм сопоставления на основе обратного предсказания (Backward oracle matching - BOM) [981] и его разновидностей [982, 983].

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 [984]. Эта особенность широких окон учитывается при использовании суффиксного автомата. В последующем этот алгоритм был усовершенствован в работах [985, 986]. Данный подход также относят к классу бит-параллельных алгоритмов.

3) **Автоматный подход с пропуском ненужных попыток сопоставления.** Автоматный подход к сопоставлению темпоральных (то есть зависящих от времени) образцов с возможным пропуском ненужных попыток сопоставления был предложен в работе [987]. 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 присутствием им параллелизмом. Считается [988, 993], что битовый алгоритм точного поиска был изобретен венгерским ученым Балинтом Дёмёлки (Dömölki) [989, 990] в 1964 году и расширен Шьямасундаром (Shyamasundar) [991] в 1977 г., прежде, чем он был заново изобретен Рикардо Баеза-Йейтсом и Гастоном Гонне (Ricardo Baeza-Yates and Gaston Gonnet) в 1992 г. [992], получивший название Shift OR. Этот алгоритм в дальнейшем был расширен и усовершенствован в [994-998].

После Shift OR в 1998 г. был предложен бит-параллельный алгоритм BNDM (Backward Non Deterministic Matching) [999, 1000], 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) [1001], SBNDM (Simplified BNDM) [1001], BNDMq и SBNDMq (q-gram BNDM/SBNDM) [1002], MBNDMq (Multiple BNDMq) [1003], LBNDM (long BNDM) [1001], FBNDM (Forward BNDM) [1004].

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 [1005-1007]. Хороший обзор бит-параллельных алгоритмов приведен в работе [1008].

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 [1009-1011] сочетают в себе символьный и автоматный подходы, а [1012, 1013] - 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 расстоянии, 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 степе-

ни похожести, которые являются наиболее упоминаемыми в научной литературе.

Расстояние Хэмминга (Hamming distance). Введено Ричардом Хэммингом (Richard Hamming) [1014] 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 [1015, 1016] в 1965 г. и определяется для строк различной длины как количество относимвольных операций вставки, удаления и замены, которые переводят одну строку в другую. Вклад в изучение этого расстояния также внёс Дэн Гасфилд (Dan Gusfield) [1017]

Расстояние Дамерау-Левенштейна (Damerau-Levenshtein). Является обобщением расстояния Левенштейна, предложено Фредериком Дамерау (Frederick J. Damerau) [1018]. 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) [1019] на основе расстояния Джаро (Matthew A. Jaro.) [1020], предложенного в 1989 г.. Неформально, расстояние Джаро между двумя словами — это минимальное число относимвольных перестановок, которое необходимо для того, чтобы изменить одно слово в другое.

Наибольшая общая подпоследовательность (longest common subsequence) [1021] - 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 [1022-1028].

Расстояние при упорядоченном алфавите. Все перечисленные выше алгоритмы неточного сопоставления строк не учи-

тывают факт возможного упорядочения символов алфавита. В свою очередь упорядоченный алфавит позволяет вводить расстояние между символами алфавита, между двумя строками и на их основании - соответствующие методы сопоставления строк. Так, например, в работе [937a] определяются Δ-расстояние, Γ-расстояние и (Δ; Γ)-расстояние между строками и соответствующие им методы неточного сопоставления строк.

Обзор литературы. В заключение обзора алгоритмов сопоставления приведем еще несколько статей.

Хороший обзор точного сопоставления строк дан в работе [939]. 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) [941] дается подробное описание 35 алгоритмов точного сопоставления строк.

Хороший цикл статей представил итальянский ученый Симоне Фаро (Simone Faro) и его коллеги. В статье [942] 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 [943] 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 [944] приводится обзор алгоритмов точного сопоставления строк, разработанных после 2000 г. м экспериментальные данные оценки некоторые из этих алгоритмов. Хороший обзор точных и неточных алгоритмов сопоставления строк приведен в [945].

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 была монография Ликлайдера (Joseph Carl Robnett Licklider) [1029], написанная в 1965 г. В ней предвиделось существование всемирной сети компьютеров, содержащей оцифрованные версии всей когда-либо написанной литературы, к которой предоставляется непосредственный доступ.

До середины 70-х годов проводились исследования и разработки в области онлайн-информационных систем и сервисов, включая и ЭБ, детальный анализ которых был выполнен в монографии [1030]. Среди них первым проектом, который можно отнести к ЭБ, считается Проект «Гутенберг» [1031] - 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 (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.



Майкл Харт

Термин "электронная библиотека" (digital library) стал широко использоваться начиная с 1991 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 *IEEE Computer* [1032], 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) [1033] как обобщенное представление библиографического универсума, независимого от какого-либо варианта каталогизации или реализации.
- Концептуальная эталонная модель CIDOC CRM [1034], 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 Общеввропейский исследовательский информационный формат CERIF (Common European Research Information Format - CERIF) [1035] 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) [1036], 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 [1037], где большое внимание было уделено функциональным возможностям современных ЭБ, сначала сформулировали манифест ЭБ [1038], в котором были освещены краеугольные концептуальные принципы ЭБ, и затем разработали эталонную модели ЭБ DLRM (Digital Library Reference Model) [1039], которая стала де-факто стандартом в Европейском союзе.
- Гонсалвес (Goncalves) и др. предложили модель 5S [1040, 1041] в качестве теоретической основы ЭБ, которая способствовала появлению множества метамodelей для разных типов ЭБ.

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. [1042]. 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:

- **Bielefeld University Library** - <http://www.base-search.net/> - поисковая система □ Билефельдского университета интегрирует ресурсы более 9000 провайдеров данных;
- **OpenAire** - информационные ресурсы открытого доступа Европейского Союза (56 миллионов документов) - <https://www.openaire.eu/>;
- **Core** - самая большая в мире коллекция (более 135 миллионов) научных статей открытого доступа - <https://core.ac.uk/>.

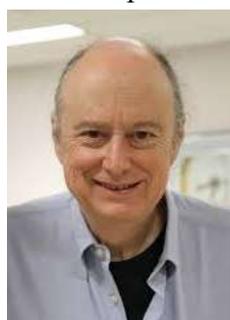
Было разработано более 30 инструментальных средств создания ЭБ (см. <http://roar.eprints.org/>). Наиболее популяр-

ными являются DSpace, EPrints, OJS, Webpress, OPUS, Fedora, Greenstone.

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 зеттабайта. 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 [1043], 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 породить 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 объемами данных?» [1044],



Джон Р. Маши

расширяться».

Следует также отметить, что впервые этот термин все же озвучил Джон Р. Маши (John R. Mashey) в 1998 г. [1045, 1046], который по поводу его употребления сказал: «мне нужна была самая простая и короткая фраза, чтобы указать, что границы вычислительной техники продолжают расширяться». Этот термин сразу же прижился в академической среде, прежде всего в отношении проблема роста и многообразия научных данных, а затем и в широко распространенном 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. В этом же году аналитическая компания Gartner отметила большие данные как тренд номер два в информационно-технологической инфраструктуре (после виртуализации). 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. [1047, 1048]. Обобщая эти и другие материалы, дадим следующее определение.

Большие данные (Big Data) – это огромные объемы неоднородной, неструктурированной или слабо структурированной, существенно распределенной и интенсивно растущей, изменяющейся и использу-

емой 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 современ-



Дуглас Лэйни

ных данных [1049], 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 рассмат-

риваться в качестве определяющих характеристик именно больших данных. Затем Зикопулос (Zikopoulos) [1050] предложил добавить еще 2 признака – достоверность и ценность (Veracity, Value), получив таким образом «5V». Со временем были предложены дополнительные определяющие характеристики Big Data [1051–1054], 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 [1055, 1056], позволяющую соотнести технологию с результатом, который ждут от обработки Big Data. Хинчклифф делит подходы к Big Data на три группы: Fast Data (быстрые данные), их объем измеряется терабайтами-петабайтными; Big Analytics (большая аналитика) – петабайтные-эксабайтные данные и Deep Insight (глубокое понимание) – эксабайты-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 нефокусированных 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 позволяют

превратить всю информацию в оперативно действующий коллективный интеллект.

Принципы работы. Исходя из определения Big Data, можно сформулировать следующие основные принципы работы с такими данными [1057]:

Распределенность. Хранить информацию в одном месте бессмысленно и практически невозможно. Поэтому техно-логия 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 продолжило 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). 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. 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 [1058] подразделения McKinsey Global Institute (MGI) международной аудиторско-консалтинговой компании McKinsey & Company приводятся методы и технологии анализа и визуализации, применимые к Big Data В нем приводятся:

- *методы анализа* (интеллектуальный анализ данных - Data Mining, краудсорсинг (crowdsourcing), машинное обучение, искусственные нейронные сети, распознавание образов, имитационное моделирование, пространственный анализ, генетические алгоритмы и др.),
- *технологии* (бизнес-аналитика (Business intelligence), облачные вычисления, хранилища данных, распределенные системы и др.),
- *средства* (Big Table, Cassandra, Dynamo, Google File System, Hadoop, MapReduce)
- *методы визуализации* (облако тегов (Tag cloud), кластерграмма (Clustergram), исторический поток (History flow), пространственный информационный поток (Spatial information flow)).

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 бес-

схемными. 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 связи с этим в больших данных применяют модели данных NoSQL. Наиболее используемой является модель ключ-значение. На этой модели определена модель



Джеффри Дин

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) [1059].



Санджай Гемават

Статья пользуется огромной популярностью. На момент подготовки данного материала она была опубликована в двух источниках и на нее было сделано более 33000 ссылок.

Эта модель вычислений 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.

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»¹, 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.², представляет собой издание объемом более 200 страниц».

¹ Когаловский М.Р. Энциклопедия технологий баз данных. М.: Финансы и статистика, 2002 г., 800 с.

² 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

полугодовые 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 Калинин, 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 ВГПИ 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

иерархического 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. Институт 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 непроизводственной 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 круп-

нейшие 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 потребно-

стей 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 обзорах³.

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 многочислен-

ные 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 бы-

³ 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

ли 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 меха-

низмов 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 обзор⁴ и отдельные статьи С.Д. Кузнецова, а также обзор В.И. Задорожного⁵ по оптимизации рекурсивных запросов в системах дедуктивных баз данных.

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.

⁴ Кузнецов С.Д. Методы оптимизации выполнения запросов в реляционных СУБД // Сб. Итоги науки и техники. Вычислительные науки. - Т. 1. - М.: ВИНТИ, 1989. - С. 76-145

⁵ 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.

Машины баз данных. Первые отечественные исследования в этой области относятся к концу 70-х — началу 80-х гг. Новый 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. Калиниченко⁶ 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, ВГПИ 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

⁶ Калиниченко Л.А., Рывкин В.М. Машины баз данных и знаний. М.: Наука; Гл. ред. физ.-мат. лит., 1990. 296 с.

архитектурная концепция системы интеграции неоднородных баз данных, основанная на некоторой интегрирующей канонической 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 моделиро-

вания 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. Тыгу, М.И. Кахро и др.].

Выбор и оценка СУБД. Проблемы выбора СУБД для конкретных приложений или для приложений в некоторой специфической 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 спомощью методов имитационного моделирования [Г.К. Столяров, О.М. Вейноров 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 ТРС 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



Андон Ф.И.

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.

Член РГБД, член программных комитетов Первой и Второй Всесоюзных конференций "Банки данных". Ученый секретарь. Комиссия 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 ГКНТ на протяжении всего времени его существования

В 70-х годах, опубликовал весьма популярные в то время препринт по реляционной модели данных⁷, а его монография по реляционной модели баз данных⁸ на протяжении многих лет пользовалась заслуженным авторитетом.

Он также занимался моделированием данных с многозначной классификацией объектов, а также рекомендациями по выбору баз данных.

⁷ Дрибас В.П., Курскова Г.Л., Столяров Г.К., и др., Введение в реляционные модели базы данных. Минск: Препринт/ инс-т математики АН БССР; № 4(20), 1977, 54 с.

⁸ Дрибас В.П. Реляционные модели баз данных. Минск : БГУ, 1982, 192 с.

Замулин Александр Васильевич (1943-2006)



Замулин А.В.

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. 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 данных⁹ и стемам программирования баз данных¹⁰.

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

⁹ Замулин А.В. Типы данных в языках программирования и базах данных. Новосибирск: Наука, Сибирское отд-ние, 1987. 152 с.

¹⁰ Замулин А.В. Системы программирования баз данных и знаний. Новосибирск: Наука, Сибирское отд-ние, 1990. 352 с.

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 обзор¹¹ и монографии по интеграции неоднородных баз данных¹² машинам баз данных¹³ не утратили полезности и цитируются до настоящего времени. Его научные исследования были воплощены в системах БАЗИС и СИЗИФ., языке СИНТЕЗ. Подготовил 10 кандидатов наук.

Член редколлегии журнала «Distributed and parallel databases»

Основатель и председатель (1992—2018) московской секции ACM SIGMOD.

Когаловский Михаил Рувимович



Когаловский М.Р.

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

¹¹ 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.

¹² Калиниченко Л.А. Методы и средства интеграции неоднородных баз данных. М.: Наука, Гл. ред. физ.-мат. лит., 1983, 424 с.

¹³ Калиниченко Л.А., Рывкин В.М. Машины баз данных и знаний. М.: Наука; Гл. ред. физ.-мат. лит., 1990, 296 с.

АСМ, ученый секретарь Московской секции ACM SIGMOD, ведущий научный сотрудник Института проблем рынка РАН, доцент МГУ.

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»¹⁴ 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 общественности¹⁵.

Пасичник Владимир Владимирович

Доктор технических наук, профессор Национального университета "Львовская политехника".



Пасичник В.В.

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 монография¹⁶, в которой

¹⁴ Когаловский М.Р. Энциклопедия технологий баз данных. М.: Финансы и статистика, 2002 г., 800 с.

¹⁵ Кузнецов С.Д. «Энциклопедия технологий баз данных» Михаила Рувимовича Когаловского. - http://www.citforum.perm.ru/book/enctbd/enctbd_vv.shtml

¹⁶ Пасичник В.В., Стогний АА. Реляционные модели баз данных. Киев : ИК АН УССР, 1983. 286 с.

исследуются вопросы реляционной модели баз данных, теории зависимостей и нормальных форм. Лауреат Государственной премии Украины в области науки и техники, отличник образования Украины.

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 толкового словаря по информатике¹⁷ и монографии по проектированию баз данных¹⁸

Стогний Анатолий Александрович (1932 — 2007)

Доктор физико-математических наук,



Стогний А.А.

профессор, член-корреспондент АН СССР и член-корреспондент НАН Украины, заместитель директора Института кибернетики НАН Украины, директор Института прикладной информатики (г. Киев).

Лауреат Государственной премии СССР в

¹⁷ Першиков В.И., Савинков В.М. Толковый словарь по информатике: Более 10000 терминов. Москва: Финансы и статистика, 1991. – 536 с.

¹⁸ Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем. Москва: Финансы и статистика, 1989 – 350 с.

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



Столяров Г.К.

Межведомственной «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.

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 Компьютер-

ного общества IEEE. Вручена за работу над программным обеспечением компьютеров «Минск», программным обеспечением информационных систем, за распространение и продвижение концепций систем управления базами данных.

Тыгу Энн Харальдович (1935-2020)



Тыгу Э.Х

Доктор технических наук, профессор, академик Эстонской Академии наук, почетный профессор Таллиннского технического университета, профессор Королевского технологического института в Стокгольме.

Лауреат государственной премии СССР в области науки и техники.

В Институте кибернетики АН ЭССР под руководством Э.Х. Тыгу в конце 70-х гг. был разработан подход по созданию единой среды языка программирования и базы данных. В монографии¹⁹ предложены методология концептуального моделирования предметной области и поддерживающий ее инструментарий, которые оказались применимы для создания СУБД

Средствами системы программирования высокого уровня ПРИЗ²⁰, которую ее идеолог Э.Х. Тыгу 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

¹⁹ Тыгу Э.Х. Концептуальное программирование. М.: Наука, Гл. ред. физ.-мат. литерат., 1984. 256 с.

²⁰ Кахро М.И. и др. Инструментальная система программирования ЕС ЭВМ (ПРИЗ) / М.И. Кахро, А.П. Калья, Э.Х. Тыгу. М.: Финансы и статистика, 1981, 158 с.

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 Шамшинович



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.

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. Савинкова²¹. Следуют также

²¹ 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

особо отметить две его монографии, в которых исследуются математические модели баз данных²² и методы моделирования семантики баз данных²³.

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, мультимедийные и изобразительные базы данных, гетерогенные базы данных, 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.

Литература

- 1) Olle T. William. The CODASYL Approach to Data Base Management. Chichester, England: Wiley-Interscience; 1978: 287p.
- 2) Tsichritzis D.C., Lochovsky F.H., Data models, Prentice-Hall, Englewood Cliffs, N.J., 1982, 381 p.
- 3) Embley D., Thalheim B., editors. Handbook of conceptual modelling: its usage and its challenges. Springer; Berlin 2011
- 4) Date C.J. An Introduction to Database Systems, 8th Edition. Addison-Wesley Longman Publishing Co., Inc. 75 Arlington Street, Suite 300 Boston, MA United States
- 4a) 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.

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

²² Цаленко М.Ш. Семантические и математические модели баз данных // Итоги науки и техники. Сер.: Информатика. Т.9, 1985. 208 с.

²³ Цаленко М.Ш. Моделирование семантики в базах данных. М.: Наука. Гл. ред. физ.-мат. лит., 1989. 288 с.

- 5) Gallaire H., Minker J., eds. Logic and Databases. New York: Plenum. 1978.
- 6) Ullman J.D. Principles of Database and Knowledge-Based Systems. Maryland: Computer Sciences Press Inc., 1989
- 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) Ozkarahan E.A. Database Machines And Database Management. Prentice Hall, 1986, 636 p.
- 9) Kalinichenko L.A., Ryvkin V.M. Database and Knowledge base Machines (Rus). Moscow, Nauka, 1990, 296 p.
- 10) Özsu M.T., Valduriez P. Principles of Distributed Database Systems, Fourth Edition, Springer, 2020
- 11) Gray J., Reuter A. Transaction Processing: Concepts and Techniques. Morgan Kaufmann, San Francisco. 1993
- 12) Harrison G. Next Generation Databases: NoSQL, NewSQL and Big Data, Apress, 2015, 235 p.
- 13) Kogalovsky M. R. Encyclopedia of databases technologies (Rus). Moscow, Finance and Statistics, 2005. — 800 c.
- 14) 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
- 15) IDS Reference Manual GE 625/635, GE Inform. Sys. Div., Pheonix, Ariz., CPB 1093B, Feb. 1968.
- 16) 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.
- 17) History of IMS: Beginnings at NASA. - <https://www.ibm.com/support/knowledge-center/zosbasics/com.ibm.imsintro.doc.intro/ip0ind0011003710.htm#ip0ind0011003710>
- 18) Long R., Harrington M., Hain R., Nicholls G. IMS Primer. - <http://www.redbooks.ibm.com/redbooks/pdfs/sg245352.pdf>
- 19) Information Management System/360, Application Description Manual H20-0524-1. IBM Corp., White plains, N.Y., July 1968.
- 20) Nies T. Cincom Systems' Total. Annals of the History of Computing, IEEE. 2009, vol. 31, No 4, pp. 55-61.
- 21) Bachman Charles W. "The programmer as navigator". Communications of the ACM, November 1973, Vol. 16 No. 11, Pages 653-658 <https://dl.acm.org/doi/10.1145/355611.362534>
- 22) Haigh T. How Data Got its Base: Information Storage Software in the 1950s and 1960s // IEEE Annals of the History of Computing (Volume: 31, Issue: 4, Oct.-Dec. 2009) pp. 6-25
- 23) CODASYL: "Data Base Task Group Report", ACM (New York 1971).
- 24) GUIDE-SHARE: "Data Base Management System Requirements", SHARE Inc. (New York 1970)
- 25) CMSAG Joint Utilities Project: "Data Management System Requirements", CMSAG (Orlando, FL 1971)
- 26) Langefors B. Theoretical Analysis of Information Systems. 402 S. m. Fig. Lund/Kopenhagen/Oslo 1966. Akademisk Forlag/Universitetsforlaget
- 27) Langefors B. Information systems theory. Inf. Syst. 2(4): 207-219 (1977)
- 28) Langefors B. Infological models and information user views. Information Systems Volume 5, Issue 1, 1980, Pages 17-32
- 29) Sungren Bo. An Infological Approach to Data Bases. National Central Bureau of Statistics, Sweden, Stockholm, 1973. 294 p.
- 30) SPARC: "Outline for Preparation of Proposals for Standardization", Document SPARC/90, CBEMA (Washington, DC 1974).
- 31) 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
- 32) 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.
- 33) CODASYL/Data Description Language Committee (DDLC), "June 73 Report". CODASYL Data Description Language Committee Journal of Development, June 1973
 - 34) "CODASYL Data Description Language Committee Journal of Development", 1978.
 - 35) Concepts and Terminology for the Conceptual Schema and the Information Base, van Griethauzen, J.J., Ed., ISO TC97/SC5/WG3, 1982, Publ. 695.
 - 36) Falkenberg E.D. Structuring and Representation of Information at the Interface Between Data Base User and Data Base Management System. Diss. Univ. Stuttgart (1975).
 - 37) Falkenberg E., Concepts of Modelling Information, Proc. of the IFIP Working Conf. on Modelling in Data Base Management Systems, Nijssen, G.M., Ed., North-Holland, 1976, p. 95-109.
 - 38) 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.
 - 39) 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.
 - 40) Durchholz R. and Richter G., "Concepts for data base management systems". In: Data Base Management, J. W. Klimbie and K. L. Koffeman, (eds.),
 - 41) Senko, M.E., Conceptual Schemas, Abstract Data Structures, Enterprise Descriptions, In: International Computing Symposium, Liege, Belgium, 1977, North-Holland Publishing Company.
 - 42) Senko M.E., Altman E.B., Astrahan MM., Fehder P.L. Data Structures and Accessing in Data-Base Systems. IBM System J., v. 12, no. 1 (1973).
 - 43) 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
 - 44) Smith J.M. and Smith D.C.P. Databases e Abstractions : Aggregation and Generalization. ACM Trans, on Database Syst, v. 2, no. 2, 1977, pp. 105133
 - 45) Smith J.M. and Smith D.C.P. Databases e Abstractions: Aggregation. Comm. of the ACM, v. 20, no. 6, 1977, pp. 405-413
 - 46) 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.
 - 47) Abiteboul, S., Hull, R., IFO: A Formal Semantic Database Model, ACM Trans. Database Syst. 12, 4 (1987), 525-565.
 - 48) Bachman, C. W., "Data Structure Diagrams", Data Base, 1969, No 1, 2, pp. 4-10.
 - 49) Engles R.W. A Tutorial on Data-base Organization, Annual review in automatic programming, Vol 7. Part I, Pergamon Press, 1972, 93 p.
 - 50) 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.
 - 51) Barker R. Case*Method: Entity Relationship Modelling Publisher: Addison-Wesley, 1990, 240 p.
 - 52) Gogolla M. An extended entity-relationship model – fundamentals and pragmatics. LNCS, vol. 767. Berlin: Springer; 1994.
 - 53) 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.
 - 54) Hohenstein U. Formale Semantik eines erweiterten Entity-Relationship-Modells. Stuttgart: Teubner; 1993.
 - 55) Thalheim B. Entity-relationship modeling – foundations of database technology. Berlin: Springer; 2000.
 - 56) 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
- 57) 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
- 58) 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
- 59) 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
- 60) 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
- 61) Codd E. F. “Relational Completeness of Data Base Sublanguages” (presented at Courant Computer Science Symposia Series 6, “Data Base Systems.” New York City, N.Y. May 24th-25th. 1971), IBM Research Report RJ987
- 62) 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
- 63) Codd E.F. “Interactive Support for Non-programmers: The Relational and Network Approaches,” Proceedings of the ACM SIGMOD Workshop on Data Description, Access, and Control, Vol. II, Ann Arbor, Michigan, May 1974.
- 64) 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
- 64a) 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.
- 65) Codd E. F. "The Second and Third Normal Forms for the Relational Model", IBM technical memo (October 6th. 1970).
- 66) 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.
- 67) 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
- 68) Date C. J. The database relational model : a retrospective review and analysisю - Addison-Wesley Educational Publishers Inc., 2000, 152 p.
- 69) Codd E.F. "Recent Investigations in Relational Database Systems," Information Processing 74, pp.1017–1021.
- 70) 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
- 71) 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
- 72) 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.
- 73) 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.
- 74) 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.
- 75) 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.
- 76) 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
- 77) 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
- 78) Date Chris J. "On DK/NF normal form". -
<https://web.archive.org/web/20120406123712/http://www.dbdebunk.com/page/page/621935.htm>
- 79) 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
- 80) 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.
- 81) 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
- 82) 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.
- 83) 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.
- 84) 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.
- 85) 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.
- 86) 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.
- 87) Delobel C. Normalization and hierarchical dependencies in the relational data model. ACM TODS 1978, 3, 3, 201-222.
- 88) 75. Pasichnik V.V., Stogniy A. A. Relational models of data bases (Rus). - M.: CNIATOMINFORM, 1983, 268 p.
- 89). 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.
- 90) 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.
- 91) 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.
- 92) Levene M. Justification for Inclusion Dependency Normal Form / M. Levene, M. W. Vincent // IEEE Transactions on Knowledge and Data Engineering, 2000. - Vol. 12, № 2. - P. 281-291.
- 93) Thalheim B. Bibliographie zur Theorie der Abhängigkeiten in relationalen Datenbanken, 1970-1984, TU Dresden 566/85, Dresden 1985.
- 94) Thalheim B. Dependencies in Relational Databases, 1991, Teubner-Texte zur Mathematik, 214 Pages
- 95) Chamberlin D.D. "Relational Data-Base Management Systems," Computing Surveys, Vol. 8, No. 1, p. 43-66, March 1976
- 96) Goldstein R.C., Strnad A.L. "The MACAIMS Data Management System,"

- Proceedings of the ACM-SIGFIDST Workshop on Data Description, Access and Control, Nov. 1970. ACM, New York, 1970, pp. 201-229.
- 97) Notley M.G. "The Peterlee IS/1 system," IBM UK Scientific Centre Report UKSC-0018, March 1972.
 - 98) Todd S.J.P. "Peterlee relational test vehicle PRTV, a technical overview," IBM Scientific Centre Report UKSC 0075, Peterlee, England, July 1975.
 - 99) Whitney V.K.M. "RDMS: A Relational Data Management System," Proceedings of the Fourth International Symposium on Computer and Information Sciences (COINS IV), Dec. 1972, Plenum Press, New York, 1972.
 - 100) 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.
 - 101) 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
 - 102) Smith J.M., Chang P. "Optimizing the performance of a relational algebra data base interface," Comm. ACM 18, 10 (Oct. 1975), pp. 568-579.
 - 103) Hall P. A. V. Optimisation of a single relational expression in a relational data base system, IBM Scientific Centre Report UKSC 0076. Peterlee, England, July 1975.
 - 104) 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
 - 105) 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
 - 106) Held G.D., Stonebraker M.R., Wong E. "INGRES: a relational data base system," Proc. AFZPS National Computer Conf., May 1975, Vol. 44, AFIPS Press, Montvale, N.J., 1975, pp 409-416.
 - 107) 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.
 - 108) Fehder P.L. The representation-independent language. Res. Rep. RJ 1121, IBM Research Laboratory, San Jose, Calif., Nov. 1972
 - 109) 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>
 - 110) McDonald N., Stonebraker M. Cupid--The friendly query language. Proc. ACM-Pacific-75, San Francisco, Calif., April 1975, pp. 127-131.
 - 111) 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.
 - 112) McDonald N., Stonebraker M. "CUPID: the friendly query language," Proc. ACM Pacific 75 Regional Conf., April 1975. ACM, New York, 1975, pp. 127-131.
 - 113) Zloof M.M. "Query by example " RC4917, IBM T. J. Watson Research Center. Yorktown Heights, N. Y., July 1974.
 - 114) Zloof M.M. "Query by Example," Proc. AFIPS National Computer Conf., May 1975, Vol. 44, AFIPS Press, Montvale, N.J., 1975, pp 431-438.
 - 115) 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.
 - 116) Zloof M.M. Query-by-Example: a data base language. IBM System J., 16:4, 1977, pp. 324-343
 - 117) 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.
 - 118) Zloof M.M. Office-by-Example: A business language that unifies data and word processing and electronic mail. IBM

- Systems Journal (Volume: 21, Issue: 3, 1982). Page(s): 272 - 304
- 119) 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.
 - 120) 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, Volume 18, No 11, p. 621-628
 - 121) 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.
 - 122) 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 Pages 249–264.
 - 123) Astrahan M.M., Chamberlin D.D. Implementation of a structured English query language. Communications of the ACM, 1975, Volume 18, No 10 pp. 580-588
 - 124) Lorie R.A. XRM-an extended (n-ary. relational memory. Tech. Report G320-2096, IBM Scientific Center, Cambridge, Mass., Jan. 1974.
 - 125) 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.
 - 126) 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.
 - 127) 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.
 - 128) 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 20(6): 560-575 (1976)
 - 129) 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, Pages 143–154
 - 130) Notley M, "Peterlee IS/1 System", UKSC Report 18, 1972
 - 131) Todd S. "The Peterlee Relational Test Vehicle - A System Overview". IBM Systems Journal. 1976, 15 (4): 285–308.
 - 132) 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
 - 133) 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
 - 134) Chamberlin D.D., et al. Support for repetitive transactions and ad-hoc queries in System R. ACM Trans. Database Syst. Vol. 6, No 1 (March 1981), 70-94.
 - 135) 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 - Volume 7, September 1981, pp. 456–464
 - 136) McDonald N., Stonebraker M., Wong E. "Preliminary design of INGRES: Part I," Electronics Research Lab. Report ERL-M435, Univ. of California, Berkeley, April 1974. <http://www2.eecs.berkeley.edu/Pubs/TechRpts/1974/ERL-m-435.pdf>
 - 137). McDonald N., Stonebraker M., Wong E. "Preliminary design of INGRES: Part II," Electronics Research Lab. Report ERL-M436, Univ. of California, Berkeley, April 1974. <http://www2.eecs.berkeley.edu/Pubs/TechRpts/1974/ERL-m-436.pdf>
 - 138). Held G., Stonebraker M. "Storage structures and access methods in the relational data base management system INGRES," Proc. ACM Pacific 75 Regional Conf., April 1975, ACM, New York, 1975, pp 26-33.

- 139) Wong E., Youssefi K. Decomposition--A strategy for query processing. *ACM Trans. on Database Systems* I, 3 (Sept. 1976), 223-241
- 140) Stonebraker M. "Implementation of integrity constraints and views by query modification," *Proc. ACM-SIGMOD Conf.* May 1975, ACM, New York, 1975, pp 65-78.
- 141) 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.
- 142) 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.
- 143) Stonebraker M., Rubinstein P. The INGRES protection system. *Proc. 1976 ACM National Conf.*, Houston, Tex., Oct. 1976
- 144) Stonebraker M., Held G., Wong E., Kreps P. "The Design and Implementation of INGRES". *ACM Transactions on Database Systems*. Vol.1, No 3. 1976 pp.189-222.
- 145) Stonebraker M., Rowe L. "The design of POSTGRES," in *Proc. 1986 ACM-SIGMOD Conf.*, Washington, DC, June 1986.
- 146) Rowe 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.
- 147) Stonebraker M. "The design of the POSTGRES storage system", in *Proc. 1987 VLDB Conf.*, Brighton, England, Sept. 1987.
- 148) Stonebraker M., Hanson E., Hong C. H. "The design of the POSTGRES rules system", *Proc. IEEE Conference on Data Engineering*, Feb. 1987.
- 149) 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
- 150) 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>
- 151) 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.-76-153. - <http://masters.donntu.org/2002/foreign/aswad/lib/mpbd.htm> или http://citforum.ru/database/articles/art_26.shtml
- 152) Brodie M.L., Schmidt J.W. Final Report of the ANSI/X3/SPARC DBS-SG Relational Database Task Group. *SIGMOD Record* 12(4): i-62 (1982).
- 153) 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,
- 154) *Advanced Transaction Models and Architectures*. Sushil Jajodia and Larry Kerschberg (eds.) Springer Science+Business Media New York. 1997
- 155) 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. Nijssen, (ed.) North Holland Publishing Company, 1976, pp.365-394
- 156) 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.
- 157) 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.
- 158) 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.
- 159) Lampson B.W. Atomic transactions. In: Lampson B.W, Paul M, Siegert H.J, edi-

- tors. Distributed systems – architecture and implementation: an advanced course, LNCS, vol. 105. Berlin: Springer; 1981. p. 246–285.
- 160 Lomet D.B. Process structuring, synchronization, recovery using atomic actions. ACM SIGPLAN Not. 1977; 12(3):128–137.
- 161) Bernstein P.A, Shipman D.W, Wong W.S. Formal aspects of serializability in database concurrency control. IEEE Trans Software Eng. 1979;SE-5(3): 203–216.
- 162) Bernstein P.A, Hadzilacos V, Goodman N. Concurrency control and recovery in database systems. Reading: Addison-Wesley; 1987.
- 163) Papadimitriou C.H. The serializability of concurrent database updates. J ACM. 1979;26(4):631–653.
- 164) Papadimitriou C.H. The theory of database concurrency control. Rockville: Computer Science; 1986.
- 165) Weikum G, Vossen G. Transactional information systems – theory, algorithms, the practice of concurrency control and recovery. San Francisco: Morgan Kaufmann; 2002.
- 166) Shasha D, Bonnet P. Database tuning – principles, experiments, and troubleshooting techniques. San Francisco: Morgan Kaufmann; 2003.
- 167) Ramamritham, K., Chrysanthis, P. K., (1997). Advances in Concurrency Control and Transaction Processing. IEEE Computer Society Press, Los Alamitos, California.
- 168) Grefen P., Apers P. (1993). Integrity Control in Relational Database Systems - An Overview. Journal of Data & Knowledge Engineering (10)2: 187-223.
- 169) 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.
- 170) 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
- 171) Davies C.T. Data processing spheres of control. IBM Syst J. 1978;17(2):179–198.
- 172) 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
- 173) 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.
- 174) Weikum G. Principles and realization strategies of multilevel transaction management. ACM Transactions on Database Systems. 1991;16(1):132–180.
- 175) Krychniak P., Rusinkiewicz M., Chichocki A., Sheth A., Thomas G. Bounding the Effects of Compensation under Relaxed Multi-Level Serializability. Distributed and Parallel Database Systems, 1996, 4(4), pp. 355-374
- 176) Lewis, P. M., Bernstein A. J., Kifer M. (2002). Databases and Transaction Processing: An Application-Oriented Approach. Addison-Wesley, United States
- 177) Breitbart Y., Garcia-Molina H., Silberschatz A. Overview of multidatabase transaction management. VLDB Journal, 1992, vol. 1, No 2, pp. 181-240.
- 178) X/Open Company Ltd., (1996). Distributed Transaction Processing: Reference Model, version 3. X/Open Company Ltd., U.K.
- 179) 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
- 180) 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
- 181) Zhang A, Nodine M, Bhargava B. Global scheduling for flexible transactions in heterogeneous distributed database sys-

- tems. *IEEE Trans Knowl Data Eng.* 2001;13(3):439–450.
- 182) 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
 - 183) 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
 - 184) 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
 - 185) Garcia-Molina H. Using Semantic Knowledge for Transaction Processing in a Distributed Database. *ACM Transactions on Database Systems*, 8(2):186-213, June 1983.
 - 186) 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
 - 187) Garcia-Molina H., Salem K. Sagas. In Proc. of ACM SIGMOD International Conference on Management of Data, 1987, pp 249-259 San Francisco, CA.
 - 188) 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
 - 189) 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
 - 190) 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.
 - 191) 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.
 - 192) Chrysanthis P.K, Ramamritham K. Synthesis of extended transaction models using ACTA. *ACM Trans. Database Syst.* 1994;19(3):450–491.
 - 193) Nodine M.H., Zdonik S.B. Cooperative transaction hierarchies: Transaction support for design applications. *VLDB Journal*, 1(1):41–80, 1992.
 - 194) 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.
 - 195) 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
 - 196) 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, pages 44-54, Minneapolis, M.N.
 - 197) Abbott R., Garsia-Molina H. Scheduling real-time transactions: a performance evaluation. *ACM Trans, on Database Syst*, 17(3), September 1992, pp. 513-560
 - 198) Agrawal D., El Abbadi A., Jeffers R. Using Delayed Commitment in Locking Protocols for Real-Time Databases. *SIGMOD Conference 1992*: 104-113
 - 199) Hong D., Johnson T., Chakravarthy S. Real-Time Transaction Scheduling: A Cost Conscious Approach. *SIGMOD Conference 1993*: 197-206.
 - 200) Alonso R., Korth H. Database System Issues in Nomadic Computing. *SIGMOD Record*, Vol. 22, No 2, 1993, pp.388-392.
 - 201) Imelinski T., Badrinath B.R. Data Management for Mobile Computing. *SIGMOD Record*, Vol. 22, No. 1, 1993
 - 202) Ceponkus A., Dalal S., Fletcher T., Furniss P., Green A., Pope B. Business transaction protocol, Version 1.1, 2002
 - 203) Business transaction protocol. - <http://www.oasis->

- open.org/committees/tc_home.php?wg_a
bbrev=businesstransaction [2004]
- 204) Stevens M., Mathew S., McGovern J., Tyagi S. Java Web Services Architecture. San Francisco: Morgan Kaufmann Publishers, 2003.
 - 205) WSTx (Web Services Transactions). - <https://searcharchitecture.techtarget.com/definition/WSTx-Web-Services-Transactions>
 - 206) IBM, BEA Systems, Microsoft, Arjuna, Hitachi, IONA, \Web Services Transactions specifications," IBM Developer Works, IBM, 2004.
 - 207) 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
 - 208) OASIS Web Services Composite Application Framework (WS-CAF), OASIS, 2006. - http://www.oasis-open.org/committees/tc_home.php?wg_aabbrev=ws-caf
 - 209) 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].
 - 210) Kratz B., Protocols For Long Running Business Transactions. Technical Report 17, Infolab Technical Report Series, 2004, 48 p.
 - 211) 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
 - 212) Snodgrass R.T., Ahn I. A taxonomy of time databases. ACM SIGMOD Record, 1985,Vo. 14, No 4, pp. 236-246
 - 213) Bubenko J.A, Jr. The temporal dimension in information modeling. Technical Report RC 6187 #26479, IBM Thomal J. Watson Research Center, Nov. 1976
 - 214) 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
 - 215) 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
 - 216) Hammer M., McLeod D. Database Description with SDM A Semantic Database Model ACM Transactions on Database Systems, 6, No 3, Sep 1981, pp 351-386
 - 217) 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)
 - 218) 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
 - 219) 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
 - 220) Sernadas A Temporal aspects of logical procedure definition. Information Systems, 1980, vol. 5, No 3, pp. 167-187
 - 221) 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
 - 222) Ariav G. A temporally oriented data model. ACM Transactions on Database Systems (TODS), 1986 vol. 11, No 4, pp. 499-527
 - 223) Ariav G., Morgan H.L., Zisman M.D. MDM: Embedding the time dimension in information systems, Technical Report 82-03-01 Department of Decision Sciences, Wharton School, University of Pennsylvania, 1982

- 224) Ben-Zvi J. "The Time Relational Model," PhD thesis, Computer Science Dept., UCLA, 1982
- 225) 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/Cummings Publishing Company, 1993). pp. 202-207
- 226) Jones S., Mason P.J. Handling the Time Dimension in a Data Base. In Proceedings of the International Conference on Data Bases, Eds. S.M. Deen and P Hammersley British Computer Society University of Aberdeen, Heyden, July 1980 pp 65-83
- 227) Snodgrass R. The temporal query language TQuel. In PODS '84: Proceedings of the 3rd ACM SIGACT-SIGMOD symposium on Principles of database systems.
- 227a) Ozsoyoglu G., Snodgrass R.T.. Temporal and Real-Time Databases: A Survey. IEEE Transactions for Knowledge and Data Engineering 7(4):513-532, 1995.
- 228) Allen J.F. "Maintaining knowledge about temporal intervals". Communications of the ACM, Nov. 1983, 26(11). pp.832-843
- 229) 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.
- 230) 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.
- 231) Lorentzos N.A., Johnson R.G. Extending relational algebra to manipulate temporal data. Inf Syst. 1988; 13(3): 289-296.
- 232) Tansel A.U. Adding time dimension to relational model and extending relational algebra. Information Systems. 1986; 11(4):343-355.
- 233) Tansel A.U. Temporal relational data model. IEEE Transactions on Knowledge and Data Engineering, 1997, 9(3): 464-479
- 234) 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
- 235) Navathe S.B., Ahmed R. A temporal relational model and a query language. Information Sciences: an International Journal. 1989;49(1-3):147-175.
- 236) Gadia S.K. A homogeneous relational model and query languages for temporal databases. ACM Trans Database Syst. 1988;13(4):418-448.
- 237) 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.
- 238) 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
- 239) Arora S. A comparative study on temporal database models: A survey, 2015 International Symposium on Advanced Computing and Communication (ISACC), 2015, pp. 161-167,
- 240) 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
- 241) Vianu V. Dynamic functional dependencies and database aging. J ACM. 1987;34(1):28-59.
- 242) Wijzen 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.
- 243) Wijzen J. Temporal FDs on complex objects. ACM Trans Database Syst. 1999;24(1):127-176.
- 244) Wijzen J. Reasoning about qualitative trends in databases. Information Systems. 1998;23(7):463-487.
- 245) 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.

- 246) Baudinet M., Chomicki J., Wolper P. Constraint generating dependencies. *J Comput Syst Sci.* 1999;59(1):94–115.
- 247) Jensen C.S., Snodgrass R.T. Temporal specialization and generalization. *IEEE Trans Knowl Data Eng.* 1994;6(6):954–974
- 248) Sarda N.L. Algebra and query language for a historical data model. *The Computer Journal.* 1990;33(1):11–18
- 249) Lorentzos N.A., Johnson R.G. Extending relational algebra to manipulate temporal data. *Information Systems.* 1988;13(3):289–296.
- 250) 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.
- 251) 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
- 252) 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.
- 253) Snodgrass S. The temporal query language TQUEL. *ACM Trans Database Syst.* 1987;12(2): 247–298.
- 254) Lorentzos N.A., Mitsopoulos Y.G. SQL extension for interval data. *IEEE Trans Knowl Data Eng.* 1997;9(3):480–99.
- 255) Sarda N.L. Extensions to SQL for historical databases. *IEEE Trans Knowl Data Eng.* 1990;2(2):220–230.
- 256) 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.
- 257) 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.
- 258) Böhlen M.H., Jensen C.S., Snodgrass R.T. Temporal statement modifiers. *ACM Trans Database Syst.* 2000;25(4):407–456.
- 258a) Chomicki J. Temporal query languages: A survey. H.J. Ohlbach and D.M. Gabbay, eds., *Proc. First Int'l Conf Temporal Logic. Lecture Notes in Artificial Intelligence 827*, Springer-Verlag, pp. 506-534, July 1994.
- 258b) 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
- 258B) 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-Wesley/ACM Press, pp. 386-408, 1995.
- 259) Snodgrass R.T. editor. In: *Proceedings of the ARPA/NSF International Workshop on an Infrastructure for Temporal Databases*, 1993.
- 260) 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.
- 261) 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. A TSQL2 tutorial. *ACM SIGMOD Rec.* 1994;23(3):27–33
- 262) Snodgrass R.T. Editor. *The TSQL2 temporal query language.* Kluwer Academic; 1995.
- 263) 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.
- 264) 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.
- 265) 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.
- 266) Kulkarni K, Michels J-E. Temporal features in SQL:2011. ACM SIGMOD Rec. 2012;41(3):34–43.
- 267) Reznichenko V.A. Temporal SQL:2011 (Rus). Software Engineering, 2013, vol. 15, No 3-4, pp. 48-65
- 268) Worboys MF, Duckham M. GIS: a computing perspective. Boca Raton: CRC press; 2004.
- 269) Shekar S, Chawla S. Spatial databases: a tour. Englewood Cliffs: Prentice-Hall; 2003
- 270) The Open Geospatial Consortium Date: 2011-12-19 OGC Reference Model. 44 p.
- 271) 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.
- 272) 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.
- 273) Tomlin C.D. A map algebra. In: Proceedings of the Harvard Computer Graphic Conference; 1983
- 274) 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 of Geographic Space, pp. 351–360. Kluwer Academic Publishers, Dordrecht, 1991
- 275) Scholl M., Voisard A. Thematic map modeling. In: Proceedings of the 1st International Symposium on Advances in Spatial Databases; 1989. p. 167–190.
- 276) Güting R.H. Geo-relational 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.
- 277) Egenhofer M.J. Spatial SQL: a query and presentation language. IEEE Trans Knowl Data Eng. 1994;6(1): 86–95.
- 278) Güting R.H, Schneider M. Realm-based spatial data types: the ROSE algebra. VLDB J. 1995;4(2):243–286.
- 279) 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.
- 280) Güting RH. An introduction to spatial database systems. VLDB J. 1994;3(4):357–99.
- 281) Rigaux P, Scholl M, Voisard A. Spatial databases - with applications to GIS. San Francisco: Morgan Kaufmann Publishers; 2002.
- 282) 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.
- 283) Schneider M. Spatial data types for database systems - finite resolution geometry for geographic information systems, vol. LNCS 1288. Berlin/New York: Springer; 1997.
- 284) Schneider M, Behr T. Topological relationships between complex spatial objects. ACM Trans Database Syst. 2006;31(1):39–81
- 285) 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.
- 286) Egenhofer M.J. & R.D. Franzosa, Point-Set Topological Spatial Relations. Int. Journal of Geographical Information Systems, 5(2), 161-174, 1991.
- 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.
- 288) Erwig M., Schneider M. Partition and conquer. In: Proceedings of the third international conference on spatial information theory; 1997. p. 389–408.
- 289) Güting R.H., Schneider M. Realms: A Foundation for Spatial Data Types in Da-

- tabase Systems. 3rd Int. Symp. on Advances in Spatial Databases, LNCS 692, 14-35, 1993.
- 290) Cicerone S., Di Felice P. Cardinal directions between spatial objects: the pairwise-consistency problem. *Inf Sci.* 2004;164(1-4):165-88.
- 291) Clementini E., Billen R. Modeling and computing ternary projective relations between regions. *IEEE Trans Knowl Data Eng.* 2006;18(6):799-814.
- 292) Freksa C. Using orientation information for qualitative spatial reasoning. In: *Proceedings of the International Conference on Spatial Information Theory*; 1992. p. 162-78.
- 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.
- 294) Hernández D. Qualitative representation of spatial knowledge. LNCS, vol. 804. Berlin: Springer; 1994.
- 295) Ligozat G. Reasoning about cardinal directions. *J Visual Lang Comput.* 1998;9(1):23-44.
- 296) Liu W., Li S. Reasoning about cardinal directions between extended objects: the NP-hardness result. *Artif Intell.* 2011;175(18): 2155-2169.
- 297) Liu W., Zhang X, Li S., Ying M. Reasoning about cardinal directions between extended objects. *Artif Intell.* 2010;174(12-13):951-983
- 298) Mukerjee A, Joe G. A qualitative model for space. In: *Proceedings of 7th National Conference on AI*; 1990. p. 721-727.
- 299) Papadias D. Relation-based representation of spatial knowledge. PhD Thesis, Department of Electrical and Computer Engineering, National Technical University of Athens; 1994.
- 300) 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.
- 301) Skiadopoulos S, Koubarakis M. Composing cardinal direction relations. *Artif Intell.* 2004;152(2):143-71
- 302) Skiadopoulos S, Koubarakis M. On the consistency of cardinal directions constraints. *Artif Intell.* 2005;163(1):91-135.
- 303) 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.
- 304) 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
- 305) 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.
- 306) Thanasis Hadzilacos, Nectaria Tryfona. An Extended Entity-Relationship Model for Geographic Applications. *ACM SIGMOD Record*, Vol. 26, No. 3, 1997, pp. 24-29
- 307) 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 Images and GIS*, 1999 pp. 77-104
- 308) 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>
- 309) 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.
- 310) Simon Jonathan David Cox, Chris Little. *Time Ontology in OWL. Technical Report* • July 2016. -

- https://www.researchgate.net/publication/305810003_Time_Ontology_in_Owl
- 311) 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.
 - 312) 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.
 - 313) Hogenboom F., Borgman B., Frasinca, 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.
 - 314) Parent C., Spaccapietra S., Zimányi E. (2006). *Conceptual modeling for traditional and spatio-temporal applications: The MADS approach*. Springer.
 - 315) Spaccapietra S., Cullot N., Parent C., Vangenot, C (2004). On spatial ontologies. *Database Laboratory, Swiss Federal Institute of Technology*. 9 p.
 - 316) 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.
 - 317) Fonseca F., Rodriguez M.A. From geopracticals to derivation ontologies: New directions for the geospatial semantic web. *Transactions in GIS, 2007*, vol. 11, No. 3, pp. 313-316.
 - 318) 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), 2006* George Mason University, Fairfax, VA, USA
 - 319) Curtin K., Noronha V., Goodchild M., Grise S. *ARCGIS transportation model (UNETRANS), UNETRANS data model reference, 2003*.
 - 320) 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
 - 321) 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.
 - 322) 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
 - 323) 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 data bases - Vol. 29, 2003*, pp. 802–813
 - 324) Miller H.J., Shaw S.L. *GIS-T data models, geographic information systems for transportation: principles and applications*. Oxford: Oxford University Press; 2001.
 - 325) Anez J., de la Barra T., Perez B. Dual graph representation of transport networks. *Transp Res.* 1996;30(3):209–216.
 - 326) Winter S. Modeling costs of turns in route planning. *GeoInformatica.* 2002; 6(4):345– 361.
 - 327) 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.
 - 328) 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
 - 329) George B., Shekhar S. Spatio-temporal network databases and routing algorithms: a summary of results. In: *Proceedings of the 10th International Symposium on Advances in Spatial and Temporal Databases; 2007*. p. 460–477.
 - 330) George B., Shekhar S. Time-aggregated graphs for modeling spatio-temporal networks - an extended abstract. In: *Pro-*

- ceedings of the 25th International Conference on Conceptual Modeling; 2006 p. 85–99.
- 331) 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.
- 332) Worboys M.F. A unified model for spatial and temporal information. *Comput J.* 1994;37(1): 25–34.
- 333) 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.
- 334) 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.
- 335) Güting R.H., Schneider M. Moving objects databases. San Francisco: Morgan Kaufmann; 2005
- 336) Erwig M., Schneider M. Spatio-temporal predicates. *IEEE Trans Knowl Data Eng.* 2002; 14(4):1–42.
- 337) 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,
- 338) 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
- 339) 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.
- 340) 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
- 341) 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
- 342) Vazirgiannis M., Wolfson O. A Spatio-temporal 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
- 343) 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
- 344) 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.
- 345) 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.
- 346) 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.
- 347) 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.
- 348) 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.
- 349) Trajcevski G., Wolfson O., Hinrichs K., Chamberlain S. Managing uncertainty in moving objects databases. *ACM Trans Database Syst.* 2004;29(3): 463–507.
- 350) 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 ob-

- jects in databases. *ACM Trans Database Syst.* 2000;25(1):1–42.
- 351) Pelekis N, Theodoridis Y. *Mobility data management and exploration*. New York: Springer; 2014.
- 352) Renso C, Spaccapietra S, Zimányi E. *Mobility data: modeling, management, and understanding*. Cambridge, UK: Cambridge University Press; 2013.
- 353) 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
- 354) Porter H.H., Oct. 1985. Optimizations to Earley deduction for DATALOG programs. Available at: <http://www.cs.pdx.edu/~harry/earley/datalog.pdf>
- 355) 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
- 356) 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.
- 357) Hafner C.D., Godden K. Portability of syntax and semantics in DATALOG. *ACM Trans. on Information Systems*, 1985, 3(2):141–164.
- 358) 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.
- 359) Ramakrishnan R., Bancilhon F., Silberschatz A. [1987]. "Safety of recursive Horn clauses with infinite relations," *Proc. Sixth ACM Symp. on Principles of Database Systems*, pp. 328-339.
- 360) Bancilhon F., Ramakrishnan R. An amateur's introduction to recursive query processing strategies. *SIGMOD Record*, v. 15, no.2, 1986, pp. 16-52
- 361) 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.
- 362) Chandra A.K., Harel D. Horn Clause Queries and Generalizations, *J. Logic Programming* 2(1):1-15 (Apr. 1985).
- 363) Gelfond M., Lifschitz V. The Stable Model Semantics for Logic Programming, in: *Proceedings of the Fifth International Conference and Symposium on Logic Programming*, 1988.
- 364) Przymusinska H., Przymusinski T.C. Weakly Perfect Model Semantics for Logic Programs, in: *Proceedings of the Fifth International Conference/Symposium on Logic Programming*, 1988.
- 365) 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.
- 366) Przymusinski T.C. Extended Stable-Semantics for Normal and Disjunctive Programs, in: *Seventh International Conference on Logic Programming*, 1990, pp. 459-477.
- 367) Ross K. Modular Stratification and Magic Sets for DATALOG Programs with Negation, in: *Proceedings of the ACM Symposium on Principles of Database Systems*, 1990, pp. 161-171.
- 368) Van Gelder A., Ross K., Schlipf J.S. The Well-Founded Semantics for General Logic Programs, *Journal of the ACM* 38(3):620- 650 (1991)
- 369) 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.
- 370) Van Gelder A. Negation as Failure Using Tight Derivations for General Logic Programs, *Journal of Logic Programming* 6(1):109-133 (1989).
- 371) 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).
- 372) Bayer R. Query Evaluation and Recursion in Deductive Database Systems,

- Technical Report 18503, Teehnisehe Universitaet Muenchen, Feb. 1985.
- 373) 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
- 374) 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.
- 375) 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.
- 376) 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.
- 377) Warren D.S. Memoing for Logic Programs, Communzations of the ACM 35 (3): 93-111 (Mar. 1992)
- 378) Sacca D., Zaniolo C. The Generalized Counting Methods for Recursive Logic Queries, in: Proceedings of the First International Conference on Database Theory, 1986.
- 379) 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.
- 380) 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.
- 381) 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.
- 382) 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.
- 383) Bancilhon F., Ramakrishnan R. An amateur's introduction to recursive query processing strategies. SIGMOD Record, Vol. 15, No.2, 1986, pp. 16-52
- 384) Gallaire H., Minker J. and Nikolas J.M. Logic and databases: a deductive approach. Computing Surveys, 16:1, 1984, pp. 154-185
- 385) Ramakrishnan R., Ullman J.D. A survey of deductive database systems. The Journal of Logic Programming, 1995, Vol. 23, No 2, pp. 125-149
- 386) 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
- 387) Reznichenko V.A. Recursive SQL (Rus). Software Engineering, 2010, vol. 4, No 4, pp. 48-65.
- 388) 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.
- 389) Liu M. Deductive Database Languages: Problems and Solutions. ACM Computing Surveys, v. 31, no. 1, 1999. pp. 27-62
- 390) Falcone Sampaio P.R., Paton N.W. (1997) Deductive object-oriented 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
- 391) Paton N.W., Díaz O. Active database systems. ACM Computing Surveys. 1999, vol. 31, No 1, pp. 63–103
- 392) 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
- 393) Paton N., Diaz O., Williams M., Campin J., Dinn A., Jaime A. Dimensions of active behaviour. In N. Paton and M. Wil-

- Williams Eds., Proc. 1st Int. Workshop on Rules in Database Systems, Springer-Verlag., 1994, pp. 40-57.
- 394) 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
- 395) 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
- 396) 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.
- 397) 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.
- 398) 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.
- 399) Chakravarthy S, Mishra D. Snoop: an expressive event specification language for active databases. Data Knowl Eng. 1994;14(1):1-26.
- 400) 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.
- 401) Gatziau 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
- 402) Mellin J., Berndtsson M. Event Detection. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 1361-1366
- 403) Mellin J., Berndtsson M. Event Specification. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 1389-1393
- 404) 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
- 405) 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
- 406) 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.
- 407) Diaz O., Jaime, A. EXACT: an EXtensible approach to ACTive object-oriented databases. VLDB Journal 1997,.6, 4, 282-295
- 408) 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
- 409) 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.
- 410) Widom J. The Starburst Rule System: Language Design, Implementation, and Applications. In: IEEE Data Engineering Bulletin, Special Issue on Active Databases, 1992, 15(4): 15-18
- 411) Stonebraker M., Kemnitz, G. The POSTGRES Next-generation Database Management System. Communications of the ACM 1991, Vol. 34, No.10, pp. 78 92
- 412) Hanson E.N. The Design and Implementation of the Ariel Active Database Rule System. IEEE Trans. Knowl. Data Eng. 1996, 8(1): 157-172

- 413) Kotz A., Dittrich K., Mülle J. Supporting semantic rules by a generalized event/trigger mechanism. In *Advance in Database Technology, EDDT, Venice* 1988, pp. 76-91.
- 414) Reddi S., Poulouvasilis 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.
- 415) Kiernan G., de Mairdreville 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
- 416) 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
- 417) 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.
- 418) 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.
- 419) 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.
- 420) 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.
- 421) Chakravarthy S. Rule management and evaluation: an active DBMS perspective. *SIGMOD RECORD* 1989, 18, 3, 20-28.
- 422) 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.
- 423) 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.
- 424) Gehani N. and Jagadish H. ODE as an Active Database: Constraints and Triggers. In R. C. G.M. Loiiiman. A. Sernadas Ed., *17th Intl. Conf. on Very Large Data Bases*, Barcelona, 1991, pp. 327-336. Morgan Kaufmann
- 425) 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.
- 426) Hull R., Tanaka K., Yoshikawa M. Behavior Analysis of Object-Oriented Databases: Method Structure, Execution Trees, and Reachability // *Lect. Notes Comput. Sci.*- 367.- 1989.- 372-388
- 427) Mozaffari M., Tanaka Y. ODM: An Object-Oriented Data Model // *New. Generat. Comp.*- 7, N 1.- 1989.- 4-35
- 428) Beeri C. A Formal Approach to Object-Oriented Databases // *Data and Knowledge Eng.*- 5.- 1990.- 353-382
- 429) Zicari R. Incomplete Information in Object-Oriented Databases // *ACM SIGMOD Record.*- 19, N 3.- 1990.- 5-16
- 430) Hong Sh., Maryanski F. Using a Meta Model to Represent Object-Oriented Data Models // *6th Int. Conf. Data Eng.*, Los Angeles, Calif., USA, Febr. 5-9, 1990.- 11-19
- 431) Cornelio, Shamkant B. Navathe, Keith L. Doty. Extending Object-Oriented Concepts to Support Engineering Applications // *6th Int. Conf. Data Eng.*, Los Angeles, Calif., USA, Febr. 5-9, 1990.- 220-227
- 432) Gunter Saake. Descriptive Specification of Database Object Behaviour // *Data and Knowledge Eng.*- 6, N 1. 1991.- 47-73
- 433) Lellani S.K., Spiratos N. Towards a Categorical Data Model Supporting Structured Objects and Inheritance // *Proc. 1st Int. East/West Database Workshop*, Kiev, Oct. 1990, *Lect. Notes Comput. Sci.*- 540.- 1991

- 434) Cattell R.G.G., Barry D.K.(eds.). The Object Data Standard: ODMG 3.0. — San Francisco, Calif.: Morgan Kaufmann, 2000.
- 435) 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
- 436) Rowe L, Stonebraker M. The Postgres data model. In: Proceedings of the 13th International Conference on Very Large Data Bases; 1987. p. 83–96.
- 437) Won Kim. UniSQL/X unified relational and object-oriented database system. SIGMOD '94: Proceedings of the 1994 ACM SIGMOD international conference on Management of data May 1994, p. 481
- 438) Darwen H., Date C.J. (March 1995). "The third manifesto". ACM SIGMOD Record. New York, NY, USA: ACM Press. 24 (1): 39–49.
- 439) 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.
- 440) 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.
- 441) 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
- 442) 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
- 443) 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.
- 444) 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.
- 445) 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
- 446) 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.
- 447) Stonebraker M.R., Neuhold E.J. A Distributed Data Base Version of INGRES // Proc. 2nd Berkley Conf. On Distributed Data Management and Computer Networks. — Lawrence Berkley Laboratory, May 1977.
- 448) 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.
- 449) Stonebraker M. The design and implementation of distributed INGRES. The INGRES Papers, Reading; 1986, p. 187–196.
- 450) Adiba M.E., Andrade J.M., Fernandez F., Gia Toan Nguyen. POLIPHEME: An experience in distributed database system design and implementation. Proc. of Int. Symposium on Distributed Data Bases. Paris, France, 1980, pp. 475-479
- 451) 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 data May 1978 Pages 169–180.
- 452) Davenport R.A. Distributed database technology — a survey. Computer Networks (1976), Volume 2, Issue 3, 1978, Pages 155-167,

- 453) 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
- 454) Sheth A.P., Larson J.A. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Comput Surv. 1990;22(3):183–236.
- 455) Stonebraker M., Aoki P.M., Pfeffer A, Sah A, Sidell J, Staelin C, Yu A. Mariposa: a wide-area distributed database system. VLDB J. 1996;5(1):48–63.
- 456) Date C.J. What is a Distributed Database System? In: Date C. J. Relational Database Writings 1985-1989. — Reading, Mass.: Addison-Wesley, 1990.
- 457) Heimbigner D., McLeod D. "A Federated Architecture for information management". ACM Transactions on Information Systems, 1985,.Volume 3, Issue 3. pp. 253–278.
- 458) 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.
- 459) Masood N., Eaglestone B. "Component and Federation Concept Models in a Federated Database System". Malaysian Journal of Computer Science, 2003, 16 (2): 47–57.
- 460) Sheth A., Larson J.A. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. ACM Computing Surveys 1990, 22(3):183--236
- 461) Litwin W., Mark L., Roussopoulos N. Interoperability of multiple autonomous databases. ACM Comput. Surv. 1990; 22(3):267–293.
- 462) Wiederhold G. Mediators in the architecture of future information systems. IEEE Comput. 1992;25(3): 38–49.
- 463) 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., Poullovassilis A. (eds) The Functional Approach to Data Management. Springer, Berlin, Heidelberg. pp. 211-238
- 464) 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
- 465) 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
- 466) 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.
- 467) Sacca D., Wiederhold G. Database partitioning in a cluster of processors. ACM Trans Database Syst. 1985;10(1):29–56.
- 468) 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.
- 469) 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.
- 470) Ceri S., Pelagatti G. Distributed databases: principles and systems. New York: McGraw-Hill; 1984.
- 471) Navathe S., Ceri S., Wiederhold G., Dou J. Vertical partitioning of algorithms for database design. ACM Trans Database Syst. 1984;9(4):680–710.
- 472) McCormick W.T., Schweitzer P.J., White T.W. Problem decomposition and data reorganization by a clustering technique. Oper Res. 1972;20(5): 993–1009
- 473) 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,
- 474) Chu W.W. Optimal file allocation in a multiple computer network. IEEE Trans Comput. 1969;- 18(10):885–889.

- 475) Apers P.M. Data allocation in distributed database systems. *ACM Trans Database Syst.* 1988;13(2): 263–304.
- 476) Bell D.A. Difficult data placement problems. *Comput J.* 1984;27(4):315–320.
- 477) 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.
- 478) 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.
- 479) 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.
- 480) Bernstein P.A., Hadzilacos V., Goodman N. *Concurrency control and recovery in database systems.* Reading: Addison Wesley; 1987.
- 481) 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.
- 482) 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.
- 483) Saito Y., Shapiro M. Optimistic replication. *ACM Comput Surv.* 2005;37(1):42–81.
- 484) 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.
- 485) Wiesmann M., Schiper A. Comparison of database replication techniques based on total order broadcast. *IEEE Trans Knowl Data Eng.* 2005;17(4):551–566
- 486) 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.
- 487) 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.
- 488) 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.
- 489) 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.
- 490) Sivasubramanian S., Szymaniak M., Pierre G., van Steen M. Replication for web hosting systems. *ACM Comput Surv.* 2004;36(3):291–334.
- 491) 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.
- 492) 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.
- 493) 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.
- 494) Almeida S., Leitão J., Rodrigues L.E.T. Chainreaction: a causal+ consistent datatore based on chain replication. In: *Proceedings of the 8th ACM*

- SIGOPS/EuroSys European Conference on Computer Systems; 2013. p. 85–98.
- 495) 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.
- 496) Gray J. Notes on data base operating systems. In: Advanced Course: Operating Systems; 1978. p. 393–481.
- 497) Ho G.S, Ramamoorthy C.V. Protocols for deadlock detection in distributed database systems. *IEEE Trans Softw Eng.* 1982;8(6):554–557.
- 498) Stonebraker M. The design and implementation of distributed ingres. In: The INGRES papers: anatomy of a relational database system; 1986. p. 187–96.
- 499) Menascé D.A., Muntz R. Locking and deadlock detection in distributed data bases. *IEEE Trans Softw Eng.* 1997;5(3):195–202.
- 500) 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.
- 501) 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.
- 502) Elmagarmid A.K. A survey of distributed deadlock algorithms. *ACM SIGMOD Rec.* 1986;15(3):37–45.
- 503) Knapp E. Deadlock detection in distributed databases. *ACM Comput Surv.* 1987;19(4): 303–328.
- 504) Singhal M. Deadlock detection in distributed systems. *Computer.* 1989;22(11):37–48.
- 505) 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.
- 506) 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.
- 507) Bracha G., Sam T. Distributed deadlock detection. *Distrib Comput.* 1985;2(3):127–138.
- 508) Chandy K.M., Lamport L. Distributed snapshots: determining global states of distributed systems. *ACM Trans Comput Syst.* 1986;3(1):63–75.
- 509) 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.
- 510) 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.
- 511) Stonebraker M. The design and implementation of distributed INGRES. In: Stonebraker M, editor. The INGRES papers. Reading: Addison-Wesley; 1986. 4)
- 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.
- 512) 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.
- 513) 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.
- 514) 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.
- 515) Yu C.T. and Chang C.C. Distributed query processing. *ACM Comput. Surv.,* 16(4):399–433, 1984.

- 516) Kossmann D. The state of the art in distributed query processing. *ACM Comput Surv.* 2000;32(4):422–469.
- 517) 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.
- 518) 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.
- 519) Ceri S., Pelagatti G. *Distributed databases principles and systems*. New York: McGraw-Hill; 1984.
- 520) 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.
- 521) Gray J.N. The transaction concept: virtues and limitations. In: *Proceedings of the 7th International Conference on Very Data Bases*; 1981. p 144–154.
- 522) Spector A.Z., Schwarz P.M. Transactions: a construct for reliable distributed computing. *ACM Operat Syst Rev.* 1983;17(2):18–35
- 523) 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
- 524) 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.
- 525) 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.
- 526) Salem K., Garcia-Molina H., Shands J. Altruistic locking. *ACM Trans Database Syst.* 1994;19(1):17–165.
- 527) 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.
- 528) 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.
- 529) Thomasian A. Distributed optimistic concurrency control methods for high-performance transaction processing *IEEE Transactions on Knowledge and Data Engineering*, 1998, 10(1):173 - 189
- 530) 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
- 531) Bernstein PA., 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.
- 532) Reed D.P. "Implementing Atomic Actions on Decentralized Data," *TOCS*, 1:1, February 1983, pp. 3–23.
- 533) Reed D.P. Naming and synchronization in a decentralized computer system. Ph. D. Thesis, MIT, Cambridge, Mass., 1977
- 534) Thomasian A. *Concurrency Control: Methods, Performance, and Analysis*, *ACM Computing Surveys*, 1998, 30(1):70-119
- 535) Ozkarahan E. *Database Machines and Database Management*. Englewood Cliffs, N.J.; Prentice-Hall, 1986 - 636 p.
- 536) 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.
- 537) 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
- 538) 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
- 539) 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. September 1976.
- 540) 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
- 541) 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
- 542) Parhami B. "A Highly Parallel Computing System for Information Retrieval" Proceedings of the Fall Joint Computer Conference, 1972. pp. 681-690
- 543) 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.
- 544) 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 Bases September 1975 Pages 456-472
- 545) 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.
- 546) Kannan K. "The Design of a Mass Memory for a Database Computer," Proc. Fifth Annual Symposium on Computer Architecture. Palo Alto, CA. April 1978, pp. 44-51
- 547) 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
- 548) 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
- 549) DeWitt D.J., "DIRECT - A Multiprocessor Organization for Supporting Relational Database Management Systems," *IEEE Transactions on Computers*. June 1979, pp. 395-406.
- 550) Madnick S.E. "The Infoplex Database Computer: Concepts and Directions," Proceedings of the IEEE Computer Conference, Feb. 1979, pp. 168-176
- 551) Hell W. "RDBM - A Relational Database Machine: Architecture and Hardware Design," Proceedings of the 6th Workshop on Computer Architecture for Non-Numeric Processing, June 1981,
- 552) 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.
- 553) DeWitt D.J., Gray J. Parallel database systems: the future of high performance database systems. *Commun ACM*. 1992;36(6):85-98.
- 554) 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.
- 555) Stonebraker M. "The Case for Shared Nothing," *Database Engineering*, Vol. 9, No. 1, 1986. pp. 4-9
- 556) 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.
- 557) 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.
- 558) Graefe G., Volcano—An Extensible and Parallel Query Evaluation System, *IEEE Trans. Knowledge Data Engineering*, 1994, vol. 6, no. 1, pp. 120-135.
- 559) 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.
- 560) 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.
- 561) Dubova N., Supercomputers nCube, Otkrytye sistemy, 1995, no. 2, pp. 42–47.
- 562) 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.
- 563) 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.
- 564) Teradata: DBC/1012 Data Base Computer Concepts & Facilities, Teradata Corp. Document No. C02-0001-00, 1983.
- 565) 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
- 566) Tandem Performance Group, "A Benchmark of Non-Stop SQL on the Debit Credit Transaction," Proceedings of the 1988 SIGMOD Conference, Chicago, IL, June 1988.
- 567) 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
- 568) 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.
- 569) Gibbs J, " Massively Parallel Systems, Rethinking Computing for Business and Science," Oracle, 1991, Vol. 6, No.1
- 570) Engler, S., Glasstone R., Hasan W., Parallelism and Its Price: A Case Study of NonStop SQL/MP, ACM SIGMOD Record, 1995, vol. 24, no. 4, pp. 61–71.
- 571) 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.
- 572) Page J., A Study of a Parallel Database Machine and Its Performance: The NCR/Teradata DBC/1012. Advanced Database Systems, Lecture Notes in Computer Science (Proc. of the 10th British Natl. Conf. on Databases. BNCOD 10, Aberdeen, 1992), Springer, 1992, vol. 618, pp. 115–137.
- 573) Baru C.K. et al. DB2 Parallel Edition, IBM System J., 1995, vol. 34, no. 2, pp. 292–322.
- 574) Sokolinsky L.B. Survey of Architectures of Parallel Database Systems (Rus). Programming and Computer Software volume 30, No 6, pages 337–346 (2004)
- 575) 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.
- 576) Graefe G., Query Evaluation Techniques for Large Databases, ACM Computing Surv., 1993, vol. 25, no. 2, pp. 73–169.
- 577) 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.
- 578) 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.
- 579) 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.
- 580) 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.
- 581) Korneev V.V. Parallel'nye vychislitel'nye sistemy (Parallel Computing Systems), Moscow: Nolidzh, 1999.
- 582) Shmidt V. IBM SP2 Systems (Rus), Otkrytye Sistemy, 1995, no. 6, pp. 53–60.
- 583) Shnitman V. Fault-Tolerant Servers ServerNet (Rus), Otkrytye Sistemy, 1996, no. 3, pp. 5–11.
- 584) Sokolinsky L.B. Organization of Parallel Query Processing in Multiprocessor Database Machines with Hierarchical Architecture, Programmirovaniye, 2001, no. 6, pp. 13–29.
- 585) 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.
- 586) 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.
- 587) 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
- 588) 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
- 589) Baumann P. Management of Multidimensional Discrete Data. VLDB Journal, Special Issue on Spatial Database Systems, 1994, Vol 4, No. 3, pp. 401–444
- 590) 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.
- 591) 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
- 592) EarthServer: The EarthServer Initiative. www.earthserver.eu
- 593) Baumann P., Holsten S. (2011) A Comparative Analysis of Array Models for Databases. In: Kim T. et al. (eds) Database Theory and Application, BioScience and Bio-Technology. BSBT 2011, DTA 2011. Communications in Computer and Information Science, vol 258. pp 80-89 Springer, Berlin, Heidelberg.
- 594) Baumann, P., Misev, D., Merticariu, V., Bang Pham Huu. Array databases: concepts, standards, implementations. Journal of Big Data, vol 8, No. 1 (2021).
- 595) Tomlin D. A Map Algebra. Harvard Graduate School of Design, 1990.
- 596) Mennis, J., Viger, R., Tomlin, C.D.: Cubic Map Algebra Functions for Spatio-Temporal Analysis. Cartography and Geographic Information Science, Vol. 32, No. 1, 2005, pp. 17-32.
- 597) Ritter G, Wilson J, Davidson J. Image Algebra: An Overview. Computer Vision, Graphics, and Image Processing. Vol. 49, No. 3, 1990, pp. 297-331.
- 598) 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.
- 599) 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
- 600) Machlin R. Index-based multidimensional array queries: safety and equivalence. In L. Libkin, editor, PODS, pp. 175–184. ACM, 2007.
- 601) van Ballegooij A.R., de Vries A.P., Kersten M. RAM: Array processing over a relational DBMS. Technical Report INS-R0301, CWI (March 2003)

- 602) 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.
- 603) Cornacchia R., Heman S., Zukowski M., de Vries A., Boncz P. Flexible and efficient IR using array databases, VLDB Journal. 7(1): 151–168.
- 604) 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
- 605) 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
- 606) 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
- 607) 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
- 608) Tollefsen, Andreas Forø (2013) PostGIS 2.0 og Raster, Kart og Plan 73(3), pp. 159–164.
- 609) Teradata. Multidimensional array options. - <https://docs.teradata.com/r/VrFCOAaniAIfJrJsA51oQJA/ZMY8sE8cSytuSPtp8QnuFA>
- 610) Oracle: GeoRaster. - http://docs.oracle.com/cd/B19306_01/appdev.102/b14254/geor_intro.htm.
- 611) Becla J., Lim K.T. Report from the first Workshop on Extremely Large Databases, Data Science Journal, 2008, Vol. 7, pp. 1-13
- 612) 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
- 613) Bauman National Library. SciDB. - <https://ru.bmstu.wiki/SciDB>
- 614) 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.
- 615) 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.
- 616) Baumann P., Stamerjohanns H. (2014) Towards a Systematic Benchmark for Array Database Systems. In: Rabl T., Poess M., Baru C., Jacobsen H.A. (eds) Specifying Big Data Benchmarks. pp 94-102.
- 617) "ISO/IEC DIS 9075-15 Information technology -- Database languages -- SQL -- Part 15: Multi-dimensional arrays (SQL/MDA)"
- 618) 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
- 619) 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>
- 620) Michalewicz Z. (ed.) Statistical and Scientific Databases. Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, 11991, 544 p.
- 621) 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
- 622) Su S. SAM: A semantic association model for corporate and scientific-statistical databases. *Journal of Information Science*, pp. 151–199, 1983.
- 623) 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,
- 624) 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
- 625) 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.
- 626) 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.
- 627) 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
- 628) 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.
- 629) Meo-Evoli L., Ricci F.L., Shoshani A., On the Semantic Completeness of Macro-Data Operators for Statistical Aggregation, *SSDBM 1992*, pp. 239-258.
- 630) 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.
- 631) Ozsoyoglu G., Ozsoyoglu Z.M., Malta F. A Language and a Physical Organization Technique for Summary Tables. *SIGMOD*, 1985: pp. 3-16.
- 632) 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
- 633) Agrawal R., Gupta A., Sarawagi S. Modeling Multidimensional Databases. *ICDE '97: Proceedings of the Thirteenth International Conference on Data Engineering* April, 1997, pp. 232–243.
- 634) 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,
- 635) 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
- 636) 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
- 637) 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
- 638) van den Berg G.M., E. de Feber. Definition and use of meta-data in statistical data processing. In *Proceedings of the 6th International Conference on Statistical and Scientific Management*, (Ascona, Switzerland), 1992, pp. 290–306
- 639) 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.
- 640) Westlake A. “A simple structure for statistical meta-data,” in *Proceedings of the 9th International Conference on scientific*

- and Statistical Databases, (Olympia, WA), pp. 186–195, IEEE Press, Aug. 1997.
- 641) Ghosh S. P. Statistical Metadata. In Kotz-Johnson Encyclopedia of Statistical Science, Vol.8, John Wiley & Sons Inc. Publ., 1988
- 642) 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
- 643) Tansel A. Query languages for statistical databases. Statistics and Computing. 1995. Vol. 5, No. 1, pp. 59-72
- 644) Ozsoyoglu G., Ozsoyoglu, Z. M. Statistical database query languages. IEEE Transactions on Software Engineering. 1985, vol 11, No. 10, pp. 1071-1080.
- 645) Johnson R. Modeling summary data. In Proceedings of the ACM SIGMOD Conference, (Ann Arbor, Michigan), pp. 93–97, 1981.
- 646) 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
- 647) 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
- 648) Computer Corporation of America. File Manager's Technical Reference Manual, Model 204 Database Management System. Computer Corporation of America, Cambridge, MA, 1979
- 649) Ghosh S.P. Statistical relational tables for statistical database management, IEEE Transactions of Software Engineering, 1986, vol. SE-12, No. 12, pp. 1106–1116.
- 650) 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
- 651) Stein D.M. A database interface to an integrated dataanalysis and plotting tool. In Proceedings of the 3rd Inter-national Workshop on Statistical and Scientific Database Management, Luxemburg, 1986, pp. 98–106
- 652) 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
- 653) 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
- 654) 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
- 655) Boufares P., Elkabbaj Y., Joiner G., Ounally H. Laversion SM90 du SGBD relationnel PEPIN. Journes SM90, Versailles, France, 1985
- 656) 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
- 657) 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.
- 658) 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
- 659) Anderson G., Snider T., Robinson B., Toporek J. An integrated research support system for inter-package 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

- 660) 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
- 661) Karasolo I., Severson 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 Management September 1983 pp. 315–324
- 662) 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
- 663) D'attri A., Ricci F.L. Interpretation of statistical queries to relational databases. In SSDBM'1988: Proceedings of the 4th international conference on Statistical and Scientific Database Management, 1988, pp. 246–258
- 664) 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
- 665) Anderson G., Snider T., Robinson B., Toporek J. An integrated research support system for inter-package communication and handling large volume output from statistical database analysis operation. In SSDBM'83: Proceedings of the Second International Workshop on Statistical Database Management, 1983, pp. 104–110.
- 666) 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
- 667) 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
- 668) 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 Management, 1983, pp.188–195
- 669) 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.
- 670) Wong H.K.T., Kuo I. GUIDE: Graphical user interface for database exploration. In Proceedings of the 8th Conference on Very Large Databases, Morgan Kaufman pubs. (Los Altos CA), McLeod and Villasenor, Mexico City, 1982, pp. 22-32
- 671) 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.
- 672) 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 Management September, 1983, pp.82–88
- 673) 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
- 674) Sato H. A data model, knowledge base and natural language processing for sharing a large statistical database. In Proceedings of the 4th International Working Conference SSDBM on Statistical and Scientific Database Management, 1988, pp. 207–225
- 675) Snodgrass R. T., The temporal query language TQuel. In Symposium on Principles of Database Systems, 1984, pp. 204–213.
- 676) 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
- 677) 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
- 678) 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
- 679) Tansel A.U. A statistical interface for historical relational databases. In *Proceedings of the Third International Conference on Data Engineering* February, 1987, pp 538–546
- 680) Reznichenko V.A. *Workig with windows in SQL (Rus)*. *Software Engineering*, 2011, vol. 7, No 3, pp. 35-48
- 681) Chandra P., Gupta M.K. Comprehensive survey on data warehousing research. *International Journal of Information Technology*. 10, pp. 217–224 (2018). <https://doi.org/10.1007/s41870-017-0067-y>
- 682) Inmon, W.H. 'Building the data warehouse', 5th Edition, John Wiley & Son. 2005
- 683) Kimball R., Ross M. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, Third Edition. John Wiley & Sons, Inc. 2013. 600 p.
- 684) 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
- 685) Brackett M.H. *The Data Warehouse Challenge: Taming Data Chaos*. John Wiley & Sons, 1996, 579 pages.
- 686) Gill S.H., Rao P.C. *The Official Client/Server Computing Guide to Data Warehouse*. QUE Corporation, 1996, 382 pages.
- 687) Poe V. *Building a Data Warehouse for Decision Support*. Prentice Hall. 1995
- 688) Codd E.F. Providing OLAP to User-Analysts: An IT Mandate // *Computerworld*. — T. 27, № 30
- 689) Pendse N. What is OLAP? - <http://dssresources.com/papers/features/pendse04072002.htm>
- 690) Ponniah P. 'Data warehousing fundamentals', John Wiley & Sons, 2001, 516 p.
- 691) 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.
- 692) 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>
- 693) 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.
- 694) Vaisman A, Zimányi E. *Data Warehouse Systems: Design and Implementation (Data-Centric Systems and Applications)* 2014th Edition. Springer; 2014.
- 695) Muhammad Arif, Ghulam Mujtaba. A Survey: Data Warehouse Architecture. *International Journal of Hybrid Information Technology* Vol.8, No. 5 (2015), pp. 349-356.
- 696) 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
- 697) 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.
- 698) 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

- 699) Data Warehouse Architecture, Concepts and Components. - <https://www.guru99.com/data-warehouse-architecture.html>
- 700) Data Warehouse Architecture: Types, Components, & Concepts. - <https://www.astera.com/type/blog/data-warehouse-architecture/>
- 701) Enterprise Data Warehouse: Concepts and Architecture. - <https://www.altexsoft.com/blog/enterprise-data-warehouse-concepts/>
- 702) 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
- 703) 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.
- 704) Kimball R., Caserta J. *The Data Warehouse ETL Toolkit*. Wiley Publ., 2004, 526 p.
- 705) Chandra P., Gupta M.K. Comprehensive survey on data warehousing research. *International Journal of Information Technology*.
- 706) 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
- 707) 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. DOI: 10.1016/j.scs.2017.08.003.
- 708) 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
- 709) 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
- 710) Demarest, “Building The Data Mart”, *DBMS Magazine*, 1994. — Vol 7, No.8. — p. 44—50.
- 711) 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.
- 712) 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.
- 713) Kimball R., Ross M. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, Third Edition. John Wiley & Sons, Inc. 2013, 600 p.
- 714) 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.
- 715) 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.
- 716) Sanjay Goil and Alok Choudhary. High performance OLAP and data mining on parallel computers. *Center of Parallel and Distributed Computing Technical Report TR9705*, 1997
- 717) Morfonios K., Ioannidis Y. Cube Implementations. In *Encyclopedia of Database Systems*, Ling Liu, M. Tamer Özsu Editors. pp. 710-716.
- 718) 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/
- 719) Pedersen TB. 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.
- 720) Vaisman A, Zimányi E. Data warehouse systems – design and smplementation. Springer; 2014.
- 721) Multidimensional DBMS. - https://tadviser.com/index.php/Article:Multidimensional_DBMS
- 722) Gosain A., Heena. 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
- 722a) Romero O., Abelló A. A Survey of Multidimensional Modeling Methodologies. *International Journal of Data Warehousing & Mining*, 5(2), 1–23, April-June 2009
- 723) 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*, Greenwich, London (UK), September 4-6, 2000. Springer LNCS, pp. 34–46
- 724) 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.
- 725) 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.
- 726) Borbst S, Rarey J. The five stages of an active data warehouse evolution. *Teradata Mag.* 2001;3(1):38– 44.
- 727) Syed Ijaz Ahmad Bukhari: Real Time Data Warehouse. CoRR abs/1310.5254 (2013)
- 728) IBM Data Warehousing. - <https://www.ibm.com/analytics/us/en/data-management/data-warehouse>.
- 729) 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>
- 730) Mohania M., Nambiar U., Tam H., Schrefl M., Vincent M. Active, Real-Time, and Intellective Data Warehousing. In *Encyclopedia of Database Systems*, Ling Liu, M. Tamer Özsu Editors. pp 41-49
- 731) Kimball R. Slowly changing dimensions. *DBMS Mag.* 1996;9(4):14.
- 732) Eder J., Koncilia C., Wiggisser K. Data Warehouse Maintenance, Evolution, and Versioning. In *Encyclopedia of Database Systems*, Ling Liu, M. Tamer Özsu Editors. pp. 884-890
- 733) 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.
- 734) Malinowski E, Zimányi E. Advanced data warehouse design: from conventional to spatial and temporal applications. Berlin/Heidelberg: Springer; 2008.
- 735) 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.
- 736) Mendelzon A, Vaisman A. Time in multidimensional databases. In: Rafanelli M, editor. *Multidimensional databases: problems and solutions*. Hershey: Idea Group; 2003. p. 166–199.
- 737) Golfarelli M, Rizzi S. Managing late measurements in data warehouses. *Int J Data Wareh Min.* 2007;3(4):51–67.
- 738) Böhlen M, Gamper J, Jensen C. Towards general temporal aggregation. In: *Proceedings of the 25th British National Conference on Databases*; 2008. p. 257–169.
- 739) Golfarelli M, Lechtenböberger J, Rizzi S, Vossen G. Schema versioning in data warehouses: enabling cross-version querying via schema augmentation. *Data Knowl Eng.* 2006;59(2):435–459.
- 740) 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.

- 741) Golfarelli M, Rizzi S. A survey on temporal data warehousing. *Int J Data Wareh Min.* 2009;5(1):1–17.
- 742) Rivest S, Bédard Y, Marchand P. Toward better support for spatial decision making: defining the characteristics of spatial on-line analytical processing (SOLAP). *Geomatica* 2001;55(4):539–555.
- 743) Bédard Y, Merrett T, Han J. Fundamentals of spatial data warehousing for geographic knowledge discovery. In: Miller H, Han J, editors, *Geographic data mining and knowledge discovery*. London: Taylor & Francis; 2001. p. 53–73.
- 744) 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.
- 745) 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.
- 746) 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.
- 747) 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.
- 748) Ahmed TO, Miquel M. Multidimensional structures dedicated to continuous spatiotemporal phenomena. In: *Proceedings of the 22nd British National Conference on Databases*; 2005. p. 29–40.
- 749) Gómez L, Gómez S, Vaisman A. Analyzing continuous fields with OLAP cubes. In: *Proceedings of the 14th ACM International Workshop on Data Warehousing and OLAP*; 2011. p. 89–94.
- 750) Gómez L, Gómez S, Vaisman A. A generic data model and query language for spatiotemporal OLAP cube analysis. In: *Proceedings of the 15th International Conference on Extending Database Technology*; 2012. p. 300–11.
- 751) Gómez L, Gómez S, Vaisman A. Modeling and querying continuous fields with OLAP cubes. *Int J Data Wareh Min.* 2013;9(3):22–45.
- 752) A.A. Vaisman, Zimányi E. Spatial Data warehousing. *Encyclopedia of Database Systems*, Ling Liu, M. Tamer Özsu Editors, Second Edition, 2018, pp. 3587-3592
- 753) Bédard, Y., T. Merrett & J. Han, 2001, *Fundamentals of Spatial Data Warehousing for Geographic Knowledge Discovery, Geographic Data Mining and Knowledge Discovery*, Taylor & Francis, Vol. Research Monographs in GIS, No. Chap. 3, p. 53-73
- 754) 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,
- 755) Strozzi C. NoSQL – A relational database management system. 2007–2010. – http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/nosql/Home%20Page
- 756) Evans E. NoSQL 2009. May 2009. – Blog post of 2009-05-12. – http://blog.sym-link.com/posts/2009/12/nosql_2009/
- 757) 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/
- 758) Fox A, Brewer E. Harvest, yield and scalable tolerant systems. In: *Proceedings of Workshop on Hot Topics in Operating Systems*; 1999. p. 174–178.
- 759) 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,
- 760) Abadi D. Consistency tradeoffs in modern distributed database system design: CAP is only part of the story. *Computer* 45(2), 37-42 (2012)

- 761) Strauch Ch. "NoSQL Databases". - <http://www.christof-strauch.de/nosqldb.pdf>
- 762) Kepner J., Chaidez J., Gadepally, Jansen H. "Associative arrays: Unified mathematics for spreadsheets, databases, matrices, and graphs," New England Database Day, 2015.
- 763) Kepner J., Chaidez J., "The Abstract Algebra of Big Data and Associative Arrays," SIAM Meeting on Discrete Math, Jun 2014, Minneapolis, MN.
- 764) Jeremy Kepner, Vijay Gadepally, Dylan Hutchison, Hayden Jananthan, Timothy Mattson, Siddharth Samsi, Albert Reuther. Associative Array Model of SQL, NoSQL, and NewSQL Databases. - <https://arxiv.org/ftp/arxiv/papers/1606/1606.05797.pdf>
- 765) A Brief History of NoSQL. - <http://blog.knuthaugen.no/2010/03/a-brief-history-of-nosql.html>
- 766) GT.M - <https://en.wikipedia.org/wiki/GT.M>
- 767) DB-Engines Ranking of Key-value Stores. - db-engines.com/en/ranking/key-value+store
- 768) Rusher J., Networks R. Triple Store. - <https://www.w3.org/2001/sw/Europe/events/20031113-storage/positions/rusher.html>
- 769) Tweed R., James G. A Universal NoSQL Engine, Using a Tried and Tested Technology. - <http://www.mgateway.com/docs/universalNoSQL.pdf>, 2010. - 25 p.
- 770) Welcome to the UnQL Specification home - <http://www.unqlspec.org/display/UnQL>
- 771) 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
- 772) Angles R., Gutierrez C. Survey of graph database models. ACM Comput. Surv. 40, 1, Article 1, 2008, 39 p.
- 773) Suciu D. Semi-structured Data Model. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 3446-3451
- 774) Suciu D. Semi-structured Query Languages. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 3457-3459
- 775) 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.
- 776) 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.
- 777) 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.
- 778) 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.
- 779) 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.
- 780) 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.
- 781) Abiteboul S., Quass D., McHugh J., Widom J., Wiener J. The Lorel query language for semistructured data. 1996. <http://www-db.stanford.edu/lore/>.
- 782) 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.
- 783) Best Document Databases. - <https://www.g2.com/categories/document-databases>
- 784) Estabrook G F., Brill R.C. The Theory of the TAXIR accessioner. Mathemati-

- cal Biosciences, 1969, Vol. 5, No 3–4, pp. 327–340.
- 785) 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.
- 786) 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
- 787) "SCSS from SPSS, Inc". *Computer-World*. September 26, 1977. p. 28.
- 788) 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 Management* September, 1983, pp.315–324
- 789) Don S. Batory. On searching transposed files. *ACM Transactions on Database Systems*, 4(4):531–544, 1979.
- 790) 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 Bases* September 1975 Pages 69–86, 1975.
- 791) 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
- 792) Khoshafian S., Valduriez P. Parallel execution strategies for declustered databases. In *Proceedings of the International Workshop on Database Machines*, pages 458–471, 1987.
- 793) 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 Engineering (ICDE)*, pp. 636–643, 1987.
- 794) Boncz P. Monet: A next-generation DBMS kernel for queryintensive applications. University of Amsterdam, PhD Thesis, 2002.
- 795) 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.
- 796) 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
- 797) 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.
- 798) 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.
- 799) Lamb A., Fuller M., Varadarajan R., Tran N., Vandiver B., Doshi L., Bear C. The Vertica analytic database: C-store 7 years later. *Proceedings of the VLDB Endowment*, Vol. 5, No 12, 2012 pp 1790–1801.
- 800) 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.
- 801) 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 Engineering (ICDE)*, pp. 466–475, 2007.
- 802) 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.
- 803) Zukowski M., Heman S., Nes N., Boncz P. Super-Scalar RAM-CPU Cache Compression. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE)*, 2006. pp. 59–71
- 804) Abadi D.J., Boncz P., Harizopoulos S., Idreos S., Madden S. (2013), "The Design and Implementation of Modern Col-

- umn-Oriented Database Systems", Foundations and Trends® in Databases: Vol. 5: No. 3, pp 197-280.
- 805) 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.
- 806) 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.
- 807) 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>
- 808) Harizopoulos S., Liang V., Abadi D.J., Madden S.R. Performance tradeoffs in read-optimized databases. In VLDB, pages 487–498, 2006.
- 809) 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
- 810) Héman S., Zukowski M., Nes N.J., Sidi-rougos L., Boncz P. Positional update handling in column stores. In Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 543–554, 2010.
- 811) Pingpeng Yuan and Hai Jin. Column Stores. In: Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors. pp. 518-523.
- 812) 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
- 813) Kanungo A. Column oriented databases. International Journal of Advanced Computational Engineering and Networking, 2017, Vol. 5, No 8, pp. 10-13
- 814) Abadi D., Boncz P., Harizopoulos S. VLDB 2009 Tutorial on Column-Stores. - <https://www.slideshare.net/abadid/vldb-2009-tutorial-on-columnstores>
- 815) Robinson I., Webber J., Eifrem E. Graph Databases, 2nd Edition. O'Reilly Media, Inc. 2015, 218 p.
- 816) Wood P.T. Graph Database. In Encyclopedia of Database Systems, Ling Liu, M. Tamer Özsu Editors, pp. 1639-1643
- 817) Angles R. Graph Databases - <http://renzoangles.net/gdm/>
- 818) Angles R., Gutierrez C. Querying RDF data from a graph database perspective European semantic web conference, 2005, pp. 346-360
- 819) Angles R., Gutierrez C. Survey of graph database models. ACM Computing Surveys, Vol. 40, No. 1, Article 1, 2008, pp. 1-39.
- 820) Angles R., Gutierrez C. The expressive power of SPARQL International Semantic Web Conference, 2008, pp.114-129
- 821) Angles R. A comparison of current graph database models. IEEE 28th International Conference on Data Engineering Workshops, 2012, 171-177
- 822) 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
- 823) 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
- 824) 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, [Online] URL: <http://ceur-ws.org/Vol-2100/paper26.pdf>
- 825) 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
- 826) 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
- 827) 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
- 828) Berge C. *Graph and Hypergraphs*. North-Holland Publishing Company, Amsterdam, 1973
- 829) 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
- 830) 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
- 831) 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
- 832) 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
- 833) 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.
- 834) Tompa F.W. A data model for flexible hypertext database systems. *ACM Transactions on Information Systems*, Vol. 7, No 1, 1989, pp. 85–100
- 835) 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
- 836) Watters C., Shepherd M.A. A transient hypergraph-based model for data access. *ACM Trans. Inform. Syst.* 8 (2), 1990, pp. 77–102
- 837) Levene M., Poulouvasilis 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
- 838) Levene M., Poulouvasilis A. An object-oriented data model formalised through hypergraphs. *Data Knowl. Eng.* 6 (3), 1991, pp. 205–224
- 839) 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
- 840) Amann B., Scholl M. Gram: A Graph Data Model and Query Language. In *European Conference on Hypertext Technology (ECHT)*. ACM, 1992, pp. 201–211
- 841) 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
- 842) 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
- 843) Hidders J., Paredaens J. GOAL, A graph-based object and association language. *Advances in Database Systems: Implementations and Applications*, CISM, 1993, pp. 247–265
- 844) Consens M., Mendelzon A. Hy+: A hypergraph-based query and visualization system. *ACM SIGMOD Record*, Vol. 22, No 2, 1993, pp. 511–516
- 845) 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
- 846) 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

- 847) Poulouvasilis 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
- 848) Paredaens J., Peelman P., Tanca L. G-Log: A graph-based query language. *IEEE Trans. Knowl. Data Eng.* 7, 3, 1995, pp. 436–453
- 849) 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
- 850) Levene M., Loizou G. A graph-based data model and its ramifications. *IEEE Trans. Knowl. Data Eng.* 7, 5, 1995, pp. 809–823
- 851) Kiesel N., Schurr A., Westfechtel B. GRAS, graph-oriented software engineering database system. *Information Systems*, Vol. 20, No 1, 1995, pp. 21-51
- 852) 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.
- 853) Hidders J. Typing graph-manipulation operations. In *Proceedings of the 9th International Conference on Database Theory (ICDT)*. Springer-Verlag, 2002. pp. 394–409
- 854) 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
- 855) Wood, P.T.: Query languages for graph databases. *ACM SIGMOD Record*, 2012, Vol. 41, No 1, pp. 50–60.
- 856) 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
- 857) Kowalik L Adjacency queries in dynamic sparse graphs. *Information Processing Letters*, 2007, vol. 102, pp. 191–195
- 858) Papadopoulos A.N., Manolopoulos Y. Nearest neighbor search - a database perspective. *Series in computer science*. Springer, Berlin, 2005, 170 p.
- 859) Aggarwal C.C., Wang H. (eds) *Managing and mining graph data. Advances in database systems*. Springer Science – Business Media, Berlin, 2005
- 860) Washio T., Motoda H. State of the Art of Graph-based Data Mining. *SIGKDD Explorer Newsletter*, 2003, vol. 5, no. 1, pp. 59–68
- 861) 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
- 862) 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
- 863) 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
- 864) Carroll J. Matching RDF Graphs. In *Proceedings of the International Semantic Web Conference (ISWC)*, 2002, pp. 5-15
- 865) 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
- 866) Fan W., Li J. Ma S., Tang N., Wu Y. Adding regular expressions to graph reachability and pattern queries. in *Proc. of the IEEE 27th International Conference on Data Engineering (ICDE)*, 2011, pp. 39–50
- 867) Mendelzon A.O., Wood P.T. Finding regular simple paths in graph databases. *SIAM J Comput*, 1995, 24(6), pp. 1235–1258
- 868) 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

- 869) 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.
- 870) Hurtado C.A., Poulouvasilis 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
- 871) 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
- 872) Consens, M.P., Mendelzon, A.O. Graph-Log: a Visual Formalism for Real Life Recursion. In: Proceedings of the 9th ACM Symposium on Principles of Database Systems. 1990, pp. 404–416.
- 873) 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.
- 874) 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
- 875) 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
- 876) 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
- 877) 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/2236754/200142116.pdf>
- 878) 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
- 879) 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
- 880) Leser U. A query language for biological networks. Bioinformatics, 2005, 21(2), pp. 33-39
- 881) 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
- 882) Prud'hommeaux, E., Seaborne, A. SPARQL Query Language for RDF. W3C Recommendation. (January 15 2008)
- 883) 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
- 884) 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
- 885) 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
- 886) 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.

- 887) Cypher - Graph Query Language - <http://neo4j.com/developer/cypher-query-language/>
- 888) 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
- 889) Santini S.: Regular Languages with Variables on Graphs. *Information and Computation*, 2012, Vol. 211, pp. 1–28
- 890) 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.
- 891) 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
- 892) Libkin L., Martens W., Vrgoc D. Querying Graph Databases with XPath. In: *Proceedings of the 16th International Conference on Database Theory (ICDT)*, 2013, pp. 129–140
- 893) 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
- 894) 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
- 895) GraphQL: A data query language. - <https://code.fb.com/core-data/graphql-a-data-query-language/>
- 896) 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
- 897) 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.
- 898) 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.
- 899) Milo T., Suciu D.. Index structures for path expressions. In: *Proceedings of the 7th International Conference on Database Theory*; 1999. pp. 277–295
- 900) 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
- 901) 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.
- 902) Calvanese D., De Giacomo G., Lenzerini M., Vardi M.Y. Reasoning on regular path queries. *SIGMOD Rec.* 2003;32(4):83–92.
- 903) 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.
- 904) 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.
- 905) Urbón P. NoSQL graph database matrix. - <http://nosql.mypopescu.com/post/619181345/nosql-graph-databasematrix>
- 906) 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
- 907) 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
- 908) Stonebraker M., Madden S.R., Abadi D.J., Harizopoulos S., Hachem N.I. 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 September 2007 Pages 1150–1160
- 909) R. Kallman, H. Kimura, J. Natkins, A. Pavlo, A. Rasin, S. Zdonik, E. Jones, S. Madden, M. Stonebraker, Y. Zhang, J. Hugg & D. Abadi, "H-store: a high-performance, distributed main memory transaction processing system," Proceedings of the VLDB Endowment, Volume 1 Issue 2, August 2008, pages 1496-1499.
- 910) M. Stonebraker, D. Abadi, A. Batkin, X. Chen, M. Cherniack, M. Ferreira, E. Lau, A. Lin, S. Madden, E. O'Neil, P. O'Neil, A. Rasin, N. Tran & S. Zdonik, "C-store: a column-oriented DBMS," Proceedings of the 31st International Conference on Very Large Data Bases (VLDB '05), 2005, pages 553 – 564.
- 911) Cattell Rick. "Scalable SQL and NoSQL data stores," ACM SIGMOD Record 39.4 (2011): 12-27.
- 912) 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>
- 913) 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/
- 914) 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>
- 915) Pavlo A., Aslett M. What's Really New with NewSQL?. SIGMOD Record, June 2016, Vol. 45, No. 2. pp. 45-55
- 916) Venkatesh, Prasanna (January 30, 2012). "NewSQL - The New Way to Handle Big Data". - <https://www.opensourceforu.com/2012/01/newsq-handle-big-data/>
- 917) Studer R., Benjamins R., Fensel D. Knowledge engineering: Principles and methods. Data & Knowledge Engineering, 25(1–2):161–198, 1998.
- 918) Guarino N., Oberle D., Staab S. What is an ontology? In Handbook on ontologies, pages 1–17. Springer, 2009.
- 919) 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
- 920) Broekstra J., Kampman A., van Harmelen F. Sesame: A generic architecture for storing and querying RDF and RDF schema. In Proc. of the First Inter. Semantic Web Conf., pp. 54–68, 2002.
- 921) 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
- 922) 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
- 923) 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
- 924) 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
- 925) 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
- 926) Ma L., Su Z., Pan Y., Zhang M, Liu M. Rstar: an rdf storage and query system for enterprise resource management. thir-

- teenth ACM international conference on Information and knowledge management, 2004:484 – 491.
- 927) Erling O., Mikhailov I.: RDF Support in the Virtuoso DBMS. In: Conference on Social Semantic Web (CSSW'07). Volume 113. (2007) 59–68
- 928) 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) 1239–1248
- 929) 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: 43-48
- 930) Abadi D.J., Marcus A., Madden S.R., Hollenbach K. Scalable Semantic Web Data Management Using Vertical Partitioning. In: Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB'07). (2007) 411–422
- 931) 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.
- 932) 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
- 933) 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) 975–980
- 934) 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.
- 935) SWAD-Europe Deliverable 10.2: Mapping Semantic Web Data with RDBM-Ses. - https://www.w3.org/2001/sw/Europe/reports/scalable_rdbms_mapping_report/
- 936) 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
- 937) 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
- 937a) Melichar B., Holub J., Polcar T. Text searching algorithms Volume I: Forward string matching. Czech Technical University in Prague, 224 p. - [/http://www.stringology.org/athens/TextSearchingAlgorithms/tsa-lectures-1.pdf](http://www.stringology.org/athens/TextSearchingAlgorithms/tsa-lectures-1.pdf)
- 938) Melichar B., Holub J., Polcar T. Text searching algorithms Volume II: Backward string matching. Czech Technical University in Prague, 61 p. - <http://www.stringology.org/athens/TextSearchingAlgorithms/tsa-lectures-2.pdf>
- 939) 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
- 941) 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>
- 942) Faro S. Exact Online String Matching Bibliography. 2016, 23 p. - <https://arxiv.org/abs/1605.05067>

- 943) Faro S., Lecroq T., Borzi, Di Mauro S., Maggio A.. The String Matching Algorithms Research Tool. *Stringology* 2016: 99-111
- 944) 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.
- 945) 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
- 946) Morris, J.H., Jr; Pratt, V. (1970). A linear pattern-matching algorithm (Technical report). University of California, Berkeley, Computation Center. TR-40.
- 947) Knuth, Donald E. (1973). "The Dangers of Computer-Science Theory". *Studies in Logic and the Foundations of Mathematics*. 74: 189–195.
- 948) Knuth D., Morris J.H., Pratt V. (1977). "Fast pattern matching in strings". *SIAM Journal on Computing*. 6 (2): 323–350.
- 949) Matiyasevich, Yuri (1973). "Real-time recognition of the inclusion relation". *Journal of Soviet Mathematics*. 1: 64–70
- 950) 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.
- 951) Baeza-Yates R., Gonnet G.H. A new approach to text searching. *Communications of the ACM*, Vol. 35, No 10, 1992 pp 74–82. - <https://doi.org/10.1145/135239.135243>
- 952) Алгоритм Бойера-Мура - https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%91%D0%BE%D0%B9%D0%B5%D1%80%D0%B0_%E2%80%94%D0%9C%D1%83%D1%80%D0%B0
- 953) Horspool R.N. Practical fast searching in strings, *Software - Practice & Experience*, 1980, 10(6) :501-506.
- 954) Zhu R.F., Takaoka T., 1987, On improving the average case of the Boyer-Moore string matching algorithm, *Journal of Information Processing* 10(3):173-177.
- 955) Turbo-BM algorithm - <http://www-igm.univ-mlv.fr/~lecroq/string/node15.html>
- 956) 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.
- 957) 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.
- 958) Smith P.D., "Experiments with a very fast substring search algorithm," *Software-Practice and Experience*, vol. 21, no. 10, pp. 1065-1074, 1991.
- 959) Raita T. Tuning the Boyer-Moore-Horspool string searching algorithm. *Software-Practice and Experience*, vol. 22, no. 10, pp. 879-884, 1992.
- 960) 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.
- 961) 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.
- 962) 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>.
- 963) Colussi L. Correctness and efficiency of pattern matching algorithms. *Information and Computation*, vol. 95, no. 2, pp. 225-251, 1991.
- 964) 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 Engineer-

- ing(ICACTE), 2010, pp. V5-310-V5-313, DOI: 10.1109/ICACTE.2010.5579620.
- 965) 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.
- 966) 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
- 967) 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.
- 968) Franek F., Jennings C.G., Smyth W.F. A simple fast hybrid pattern-matching algorithm. *J. Discrete Algorithms*, 5(4):682–695, 2007.
- 969) 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. — doi:10.1147/rd.312.0249.
- 970) Rabin–Karp algorithm. — https://en.wikipedia.org/wiki/Rabin%E2%80%93Karp_algorithm
- 971) Wu S., Manber U. "A fast algorithm for multi-pattern searching," Department of Computer Science, University of Arizona, Tucson, AZ, Report TR-94-171994.
- 972) 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.
- 973) 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.
- 974) Lecroq T. "Fast exact string matching algorithms," *Information Processing Letters*, vol. 102, no. 6, pp. 229-235, Jun 15 2007.
- 975) 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.
- 976) Daciuk J., Mihov S, Watson B., Watson R. Incremental construction of minimal acyclic finite state automata. *Computational Linguistics*, 2000, 26(1), pp.3-16.
- 977) Yang. W. Mealy machines are a better model of lexical analyzers. *Computer Languages*, Vol. 22, No 1, 1996, pp. 27-38
- 978) 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
- 979) 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
- 980) Allauzen C., Raffinot M. Simple optimal string matching algorithm, *Journal of Algorithms*, Vol. 36, No 1, 2000, pp. 102–116
- 981) 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.
- 982) 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.
- 983) 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.
- 984) He L., Fang B., Sui J. The wide window string matching algorithm. *Theoretical Computer Science*, vol. 332, no. 1-3, 2005, pp. 391-404
- 985) 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.

- 986) 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.
- 987) 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.
- 988) Bitap algorithm. - https://en.wikipedia.org/wiki/Bitap_algorithm
- 989) Bálint Dömölki, An algorithm for syntactical analysis, Computational Linguistics 3, Hungarian Academy of Science pp. 29-46, 1964.
- 990) Bálint Dömölki, A universal compiler system based on production rules, BIT Numerical Mathematics, 8(4), pp 262-275, 1968. doi:10.1007/BF01933436
- 991) Shyamasundar R.K. Precedence parsing using Dömölki's algorithm, International Journal of Computer Mathematics, 6(2)pp 105-114, 1977.
- 992) Baeza-Yates R., Gonnet G.H. A new approach to text searching. Communications of the ACM, Vol. 35, No 10, 1992 pp 74-82
- 993) Ricardo A. Baeza-Yatesm Gaston H. Gonnet. A New Approach to Text Searching. Communications of the ACM, 1992, vol. 35, No 10, pp. 74-82 - ПОВТОРЕНИЕ 5a)
- 994) 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
- 995) Salmela L., Tarhio J., Kytöjoki J. Multi pattern string matching with q-grams. Journal of Experimental Algorithms, 2006, Vol. 11, pp. 1-19
- 996) 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.
- 997) R. Baeza-Yates and G. Navarro. A faster algorithm for approximate string matching. In Dan Hirschberg and Gene Myers, editors, Combinatorial Pattern Matching (CPM'96), LNCS 1075, pages 1-23, Irvine, CA, June 1996.
- 998) G. Myers. "A fast bit-vector algorithm for approximate string matching based on dynamic programming." Journal of the ACM 46 (3), May 1999, 395-415.
- 999) 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.
- 1000) 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
- 1001) Peltola H., Tarhio J. Alternative Algorithms for Bit-Parallel String Matching. In String Processing and Information Retrieval, Spire Springer, LNCS 2857, pp. 80-93, 2003.
- 1002) 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.
- 1003) 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. - <https://doi.org/10.1109/ICGEC.2010.149>
- 1004) 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,
- 1005) Peltola H., Tarhio J. (2003) 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 2003. pp. 80-93. Lecture Notes in Computer Sci-

- ence, vol 2857. Springer, Berlin, Heidelberg.
- 1006) M. 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
- 1007) 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
- 1008) Gupta S., Rasool A. Bit Parallel String Matching Algorithms: A Survey. International Journal of Computer Applications, 2014, vol. 95, No 10, pp. 27-32
- 1009) M. Crochemore, A. Czumaj, L. Gal'gsieniec, T. Lecroq, W. Plandowski, and W. Rytter, "Fast practical multi-pattern matching," Information Processing Letters, vol. 71, no. 3-4, pp. 107-113, Aug 27 1999.
- 1010) G. Navarro, Nrgrep: A fast and flexible pattern matching tool. Software—Practice & Experience, Vol. 31, No 13, 2001, pp. 1265–1312. - <https://doi.org/10.1002/spe.411>.
- 1011) F. Franek, Jennings, C. G., and Smyth, W. F., "A simple fast hybrid pattern-matching algorithm," J. Discret. Algorithms, pp. 682-695, 2007.
- 1012) S. Deusdado and P. Carvalho, "GRASPm: an efficient algorithm for exact pattern-matching in genomic sequences," Int J Bioinform Res Appl, vol. 5, no. 4, pp. 385-401, 2009.
- 1013) P. Shivendra Kumar, H. K. Tiwari, and P. Tripathi. "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.
- 1014) Hamming R. W. "Error detecting and error correcting codes". The Bell System Technical Journal. 1950, 29 (2): 147–160
- 1015) 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.
- 1016) Levenshtein, Vladimir I. (February 1966). "Binary codes capable of correcting deletions, insertions, and reversals". Soviet Physics Doklady, 1966. 10 (8): 707–710.
- 1017) Dan Gusfield. Algorithms on stings, trees, and sequences: Computer science and computational biology ACM SIGACT News, Vol. 28, No 4, Dec. 1997, pp. 41–60. - <https://doi.org/10.1145/270563.571472>
- 1018) 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
- 1019) 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.
- 1020) 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
- 1021) Longest common subsequence problem. - https://en.wikipedia.org/wiki/Longest_common_subsequence_problem
- 1022) Hall P., Dowling G. Approximate string matching. ACM Computing Surveys, 12(4) :381-402, 1980.
- 1023) Sankoff D. , Kruskal J., editors. Time Warps. String Edits, arid Macro molecules: The Theory arid Practice of Sequence Comparison. Add is on-Wesley, 1983.
- 1024) Apostolico A., Galil Z. Combinatorial Algorithms on Words. NATO ISI Series. Springer-Verlag, 1985.
- 1025) Galil Z., Giancarlo R. Data structures and algorithms for approximate string matching. Journal of Complexity, Vol. 4, No 1, 1988, pp. 33-72
- 1026) Jokinen P, Tarhio J, Ukkonen E. A comparison of approximate string matching algorithms. Software Practice arid Experience, 26(12): 1439-1458,1996.

- 1027) Navarro G. A guided tour to approximate string matching. *ACM Computing Surveys*, Vol. 33, No. 1, 2001, pp 31–88
- 1028) 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
- 1029) Licklider J.C.R. *Libraries of the future*. Cambridge, MA: The MIT Press; 1965.
- 1030) Charles P. Bourne, Trudi Bellardo Hahn. *A History of Online Information Services, 1963-1976*. MIT Press, 2003, 496 p
- 1031) Project Gutenberg. - https://en.wikipedia.org/wiki/Project_Gutenberg
- 1032) Schatz B. (1996). Chen H. (ed.). "Building large-scale digital libraries". *IEEE Computer*. 29 (5): 22–25.
- 1033) *Functional Requirements for Bibliographic Records, Final Report / IFLA Study Group on the Functional Requirements for Bibliographic Records*. – München: K.G. Saur, 1998.
- 1034) Crofts N., Doerr M., Gill T., Stead S., Stiff M. (editors), *Definition of the CIDOC Conceptual Reference Model*, January 2008. Version 4.2.4.
- 1035) CERIF in Brief. - https://www.eurocris.org/eurocris_archive/cerifsupport.org/cerif-in-brief/index.html
- 1036) David Shotton. *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/>
- 1037) 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.
- 1038) 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)
- 1039) Candela L., Castelli D., Dobрева 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.
- 1040) 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*. 22(2), 2004, p. 270–312.
- 1041) Isah A., Serema B. C., Mutshewa A., Kenosi L. (2013). "Digital Libraries: Analysis of Delos Reference Model and 5S Theory". *Journal of Information Science Theory and Practice*. 1 (4): 38–47
- 1042) *Open Archives Initiative Protocol for Metadata Harvesting*. - https://en.wikipedia.org/wiki/Open_Archives_Initiative_Protocol_for_Metadata_Harvesting
- 1043) Cherniak L. *Big Data — new theory and practice (Rus) // Otkrytye sistemy. SUBD. — 2011. — № 10*. (Electronic resource): Access mode: <http://www.osp.ru/os/2011/10/13010990/>
- 1044) 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
- 1045) Mashey J.R. (25 April 1998). "Big Data. and the Next Wave of InfraStress". - http://static.usenix.org/event/usenix99/invited_talks/mashey.pdf
- 1046) Lohr S.(1 February 2013). "The Origins of 'Big Data': An Etymological Detective Story". *The New York Times*. Retrieved 28 September 2016.
- 1047) Dutcher J.. *What Is Big Data?* <https://datascience.berkeley.edu/what-is-big-data/>
- 1048) Zolotov O., Romanovskaya Y., Rzhannikova V. *On Definition of BigData*. - EPJ Web of Conferences 224, 04011 (2019)
- 1049) Doug L. (2001. *3D Data Management: Controlling Data Volume, Velocity, and*

- Variety. Technical Report 949, META-Group (now Gartner). [Electronic resource]: <https://blogs.gartner.com/douglaney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>
- 1050) 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 resource]: ftp://public.dhe.ibm.com/software/pdf/at/SWP10/Harness_the_Power_of_Big_Data.pdf
- 1051) The Four V's of Big Data. IBM (2011). - http://www.ibmbigdatahub.com/sites/default/files/infographic_file/4-Vs-of-big-data.jpg
- 1052) 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/>
- 1053) McNulty E. Understanding Big Data: The Seven V's. Dataconomy (22 May 2014). - <http://dataconomy.com/2014/05/seven-vs-big-data/>
- 1054) Firican G. The 10 Vs of Big Data - <https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx>
- 1055) Hinchcliffe D. Big Data, The Moving Parts: Fast Data, Big Analytics, and Deep Insight. - <https://www.flickr.com/photos/dionh/7550578346/in/photostream/>
- 1056) 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/>
- 1057) 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>
- 1058) 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/~muratkoures/cloud11f_files/MGI-full-report.pdf
- 1059) Dean J., Ghemawat S. MapReduce: simplified data processing on large clusters. Communications of the ACM, Volume 51, Issue 1. pp. 107-113

Сведение 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 коллектива «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 компьютерных наук Национального авиационного университета Украины.

Возглавляет небольшой коллектив, который создал и уже 14 лет сопровождает Научную электронную библиотеку периодических изданий НАН Украины (dspace.nbuv.gov.ua).

e-mail: reznichenko.valery47@gmail.com