



Б.Е. ПАНЧЕНКО

УДК 004.652

**ИССЛЕДОВАНИЯ ДОМЕННО-КЛЮЧЕВОЙ СХЕМЫ
РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ**

Ключевые слова: *схема реляционной базы данных, реляционный каркас, доменно-ключевая нормальная форма, произвольная предметная область, синтез ДКНФ-схемы.*

ВВЕДЕНИЕ

Семантический анализ произвольных предметных областей (ПрО) и современное проектирование схемы реляционной базы данных (БД) осуществляется в основном с помощью модели П. Чена «сущность-связь» [1] или расширенной реляционной модели данных (РМД) Э. Кодда [2]. Обзор методов проектирования приведен в классических учебниках Д.М. Кренке [3] и С.Д. Кузнецова [4]. Однако модель «сущность-связь» обладает некоторыми недостатками — она не дает строгих формальных определений сущности и атрибута сущности, а также не учитывает функциональных требований к приложению на стадии проектирования схемы БД.

Для некоторых приложений такой подход применим, когда можно абстрагироваться от операций над данными и когда моделируются только сами данные. Но основой приложений являются функциональные требования к манипулированию данными, описанные в техническом задании (ТЗ). Стоимость проекта оценивается на основании ТЗ. И если проектировщик вынужден сопровождать схему БД, постоянно учитывая всевозможные модификации, стоимость проекта будет возрастать.

Классический подход к проектированию схемы БД имеет следующие недостатки:

- 1) сложность и трудоемкость идентификации функциональных зависимостей (ФЗ);
- 2) зависимость конечного результата проектирования от опыта и субъективного взгляда проектировщика, а не от метода проектирования;
- 3) проблема идентификации сущностей и атрибутов сущностей;
- 4) модифицируемость схемы БД;
- 5) отсутствие четких технологий получения высоконормализованных схем БД.

При значительном количестве классов сущностей и атрибутов число всевозможных ФЗ существенно возрастает. Как правило, рассматриваются не все ФЗ. И почти все отношения из-за экономии времени проходят процесс нормализации не выше 3НФ (нормальной формы), что может приводить к сбоям и ошибкам на этапе эксплуатации. Очевидно, что для их устранения потребуются значительные затраты ресурсов.

Указанные недостатки и проблемы определяют актуальность разработки метода проектирования схемы БД, устраняющего ошибки проектирования, соответствующего практическим реалиям и значительно снижающего трудозатраты. На

© Б.Е. Панченко, 2012

наш взгляд, именно модифицируемость схемы БД [5, 6] является одним из самых важных критериев качества внедряемых приложений. Однако это свойство невозможно реализовать на низконормализованных схемах. В [7] обоснован алгоритм синтеза схемы БД, находящихся в доменно-ключевой нормальной форме (ДКНФ) [8], так что одновременно удовлетворяются критерии безаномальности и высокой модифицируемости.

НЕКОТОРЫЕ КОНТРВЫВОДЫ ОТНОСИТЕЛЬНО ДКНФ

Многие исследователи сомневаются в целесообразности высокой нормализации. По мнению автора работ [9, 10], проблема заключается в отсутствии возможности доказательств факта следствия всех ограничений каждого отношения схемы и ограничений на домены и ключи этих отношений. «Неясно, как выявить все ограничения целостности. Учитывая, что ограничений много, для каждого нужно привести доказательство. Получается очень трудоемкий процесс ... в идентификации ФЗ» [10]. Однако для совокупности актуальных ячеек [7] каркаса характерны:

— обусловленность доменов и ключей подмножества бинарных, тернарных, кварternарных и других отношений-связей доменами и значениями ключей унарных (базовых) отношений, а также их ограничениями, которые строго определяются особыми ограничениями, исследованными в [6];

— отсутствие в следующих по каскаду отношениях-связях каких-либо иных ключей, кроме совокупности унарных;

— наличие единственной ФЗ шунтирующих неключевых атрибутов каждого отношения-связи только от своих составных ключевых атрибутов.

Эти свойства обеспечивают «ретрансляцию без искажений» любых ограничений, которые не противоречат особым ограничениям.

При возникновении в тех или иных актуальных ячейках каркаса дополнительных ограничений ПрО, не являющихся следствием особых ограничений, соответствующие схемы отношений переходят в ДКНФ. Локализованные таким образом и предсказанные аномалии могут отслеживаться и обрабатываться приложением отдельно.

При возникновении в ПрО дополнительных ограничений, противоречащих [8] особым ограничениям, соответствующие ячейки каркаса становятся невыполнимыми и поэтому неактуальными.

Описанные свойства схем БД, построенных на каркасе, позволяют приложению отслеживать процедуры ссылочной целостности в режиме квазиреального времени, когда ввод/редактирование данных в ДКНФ-схему на любой совокупности параллельных терминалов синхронизировано с механизмами контроля и проверки.

В работах [9, 10] утверждается, что «в СУБД используются триггеры и транзакции, не рассмотренные Р. Фейджиным. Например, при вставке записи в таблицу водительских прав должен сработать триггер, который запретит вставку, если данный пользователь числится в группе наркоманов. Перевод денег со счета на счет может выполняться исключительно в виде транзакции. Следовательно, рассуждения о ДКНФ не применимы в полной мере к современным СУБД». Однако описанные ситуации могут быть выражены ограничениями на домены ключей соответствующих связей сущностей-объектов. В схему БД вносится отношение-связь, моделирующее существующую в ПрО связь *ЗДОРОВЬЕ ПРЕТЕНДЕНТА (КодПретендента, КодКатегорииЗдоровья,...)*, которая ограничивает домен ключа для многоарного отношения-связи *ПРЕТЕНДЕНТЫ НА ПОЛУЧЕНИЕ ПРАВ (КодПретендента, КодКатегорииЗдоровья, КодСдачиЭкзаменов, КодКатегорииТранспСредства, ...)* по ключу *КодПретендента* при условии, что *КодКатегорииЗдоровья* эквивалентен значениям *НАРКОМАНЫ, ПЬЯНИЦЫ, ЦВЕТОАНОМАЛЫ* и т.п. Ссылочная целостность отслеживается не триггерами, а универсальными процедурами СУБД.

Аналогичная ситуация возникает и в случае нецелостной банковской транзакции. Проблема в том, что проектировщик схемы БД должен мыслить не лис-

тингом приложения, не триггерами СУБД и иными категориями реализации ПО, а связями в ПрО. Тогда проблем с ретрансляцией взаимных ограничений в схемах БД не возникнет.

В работе [11] отмечено, что в определении ДКНФ Фейджин допустил некорректность, заключающуюся в противоречии между нестрогим критерием зависимости отношения от совокупности доменов и ключей всей ПрО и целостностью этого локального, нормализованного по Кодду [12] отношения как носителя ФЗ, не связанных со всей ПрО. «Схема отношения у Фейджина соответствует концептуальной схеме ПрО, а ограничения, поддерживаемые СУБД, — схеме БД. Аномалии возникают из-за того, что ограничения, входящие в схему отношения, невозможно выразить через ограничения, поддерживаемые СУБД» [11].

Действительно, не все производители современных СУБД заинтересованы предоставлять проектировщику свободу манипулирования данными, что характерно для инструментальных средств, разработанных в 1980-х годах. Тем не менее при возникновении конфликтов ДКНФ дает достаточно объективный ориентир, в том числе и для принятия решения о приобретении конкретной СУБД.

В [11] приведен характерный пример — схема отношения СОТРУДНИК (*КодСотр*, *НомКом*, *НомТел*), которая рассматривается автором как «претендент» на безаномальную схему, хотя и обладает аномалиями. Задекларированы следующие ограничения: атрибут *КодСотр* (уникальный табельный номер сотрудника) — ключевой атрибут отношения, *НомКом* (номер комнаты, в которой находится рабочее место сотрудника) — уникальный неключевой атрибут, а *НомТел* (номер телефона, установленного в этой комнате) — также уникальный неключевой атрибут для исследуемого отношения; две ФЗ: $КодСотр \rightarrow НомКом$ и $НомКом \rightarrow НомТел$.

Для раскрытия этой аномалии проведем подробный анализ данного отношения. Прежде всего определим, что схема отношения согласно Кодду [12] может быть отнесена лишь к 2НФ — она обладает транзитивной ФЗ. Ключевой атрибут отношения *КодСотр* является детерминантом ФЗ, а неключевой *НомКом* — детерминантом иной ФЗ. Как известно, аномалия исключается декомпозицией отношения на два отношения по «неключевой» ФЗ. Поэтому в [11] приведенное отношение некорректно отнесено к категории «высоконормализованных». Однако целесообразно более подробно анализировать возможные зависимости в указанном отношении-связи (с некорректным наименованием, присвоенным ему автором [11]) *СОТРУДНИК* (*КодСотр*, *НомКом*, *НомТел*), где имеется три атрибута, которые в общем случае представляют не зависящие один от другого атомарные сущности-объекты из ПрО.

Таблица 1. СОТРУДНИКИ+КОМНАТЫ+ТЕЛЕФОНЫ (2НФ)

<i>КодСотр</i>	<i>НомКом</i>	<i>НомТел</i>
001	13	22
002	13	22
003	14	22
004	14	22
005	15	23
006	16	23

Из практики проектирования схем БД известно, что в подобном отношении может быть четыре различных варианта зависимостей между атрибутами, по аналогии с [8] называемые нами ограничениями ПрО. Опишем их формально и дадим неформальные комментарии.

1. Транзитивная ФЗ. $КодСотр \rightarrow НомКом$ ($F:1$), $НомКом \rightarrow НомТел$ ($F:1$). Формируется избыточная ФЗ $КодСотр \rightarrow НомТел$ ($F:1$). Эта классическая ситуация приведена в [11].

Из табл. 1 следует, что отношение находится в 2НФ. Оно должно быть декомпозировано на два бинарных отношения: *СОТРУДНИКИ+КОМНАТЫ*, *КОМНАТЫ+ТЕЛЕФОНЫ*, а отношение *СОТРУДНИКИ+ТЕЛЕФОНЫ* может быть получено запросом — потеря информации при этом не будет. При этом даже если исходное тернарное отношение — важный для ПрО отчет, которой

пользователь формирует регулярно, его аномалия не может быть шунтирована, поскольку такая «связь» не может иметь атрибутов. Действительно, формально вставить неключевой атрибут в такое отношение невозможно, потому что весь его кортеж (схема предиката связи) является ключом лишь формально, как дополнение к первичному ключу *КодСотр*. Это означает, что произвольный атрибут связи может быть лишь суррогатным, но и он будет зависеть не от всей совокупности, а лишь от первичного ключа *КодСотр*. И при этом аномалия не будет аннулирована — причина в несоответствии ограничений на ключ, так как дополнительное ограничение — «вторая» ФЗ — не является следствием ключа. Поэтому ограничения не являются следствием особых ограничений — не могут моделироваться шунтированным каркасным отношением.

Заметим, однако, что это и не нужно, поскольку в каркасной схеме БД аналогично традиционной практике, помимо трех атомарных «справочников»: *СОТРУДНИКИ* (*КодСотр*, ...), *КОМНАТЫ* (*НомКом*, ...) и *ТЕЛЕФОНЫ* (*НомТел*, ...) имелось бы два бинарных отношения-связи: *СОТРУДНИКИ+КОМНАТЫ*, *КОМНАТЫ+ТЕЛЕФОНЫ*. Отношение-связь *СОТРУДНИКИ+ТЕЛЕФОНЫ* синтезировалось бы автоматически, причем для каждой из этих схем несложно обеспечить условия соответствия ДКНФ.

2. Многозначная зависимость (МЗ) [8]. Она возникает тогда, когда одна (условно «центральная») сущность-объект (ее единственный атрибут или группа составных атрибутов) функционально зависит от каждой иной («соседней») сущности-объекта (одного или группы атрибутов), а они, в свою очередь, один от другого не зависят. Хотя обозначать это принято иначе: правая часть ФЗ многозначно определяет ее детерминант: $НомКом \twoheadrightarrow КодСотр (1:H)$ и $НомКом \twoheadrightarrow НомТел (1:H)$, а между атрибутами *КодСотр* и *НомТел* зависимость отсутствует ($F:H$).

Неформально это описывается так: к любому из телефонов, установленных в данной комнате (по несколько разных номеров, но не подключенных параллельно с телефонами в иных комнатах!), может подойти любой сотрудник этой (или иной) комнаты.

В подобной ситуации «пользователи» вносят в ПрО еще одно ограничение: внутри комнат номера телефонов закрепляются за конкретными сотрудниками. Появляется ФЗ $КодСотр \rightarrow НомТел (F:1)$, после чего МЗ в отношении пропадает, но появляется описанная транзитивная зависимость. Хотя, говоря неформально, на практике такое ограничение ПрО весьма

Таблица 2. СОТРУДНИКИ+КОМНАТЫ+ТЕЛЕФОНЫ (НЗ)

<i>КодСотр</i>	<i>НомКом</i>	<i>НомТел</i>
001	13	22
001	13	23
002	13	22
002	13	23
003	14	24
003	14	25
004	14	24
004	14	25
005	15	26
006	15	26

условно для абонента, заинтересованного не в конкретном номере телефона, а в абоненте, если закрепленный за ним номер в этот момент занят.

Из табл. 2 следует, что отношение находится в БКНФ, но не в 4НФ (и не в ДКНФ). Оно должно быть декомпозировано на два бинарных отношения: *КОМНАТЫ+СОТРУДНИКИ* и *КОМНАТЫ+ТЕЛЕФОНЫ*, потерь информации при этом не будет, или шунтировано атрибутами этой связи.

3. Зависимость проекции-соединения (ЗПС) [8]. Может возникать тогда, когда от некоторой «первой» сущности-объекта (ее атрибутов) зависит лишь одна («вторая») сущность-объект (ее атрибуты), но от нее не зависит «третья» сущность-объект (ее атрибуты), а также между собой не зависят «вторая» и «третья».

Формально: $КодСотр \rightarrow НомКом (F:1)$, но $НомКом \not\rightarrow НомТел (F:H)$ и $КодСотр \not\rightarrow НомТел (F:H)$ — это наиболее часто встречающийся тип ограничения ПрО на связи между независимыми сущностями-объектами.

Таблица 3. СОТРУДНИКИ+КОМНАТЫ
+ТЕЛЕФОНЫ (ЗПС)

<i>НомКом</i>	<i>КодСотр</i>	<i>НомТел</i>
13	001	22
14	002	23
15	003	23
14	004	24
16	005	25
16	005	23
15	006	26
17	008	27
17	009	28
18	010	27

Из табл. 3 следует, что отношение находится в 4НФ. Однако несложно убедиться, что оно имеет аномалии и удовлетворяет ЗПС, поэтому должно быть декомпозировано на три бинарных: КОМНАТЫ+СОТРУДНИКИ, КОМНАТЫ+ТЕЛЕФОНЫ и СОТРУДНИКИ+ТЕЛЕФОНЫ, потерь информации при этом не будет, или шунтировано атрибутами этой связи.

4. Декартова зависимость (ДЗ). Эта зависимость введена и обоснована в [6]. Возникает тогда, когда ни одна сущность-объект (ее атрибуты) не зависит ни от какой иной сущности-объекта.

Формально: $НомКом \twoheadrightarrow КодСотр (F:H)$, $НомКом \twoheadrightarrow НомТел (F:H)$ и $КодСотр \twoheadrightarrow НомТел (F:H)$.

При этом очевидно, что отсутствует еще и зависимость $КодСотр \twoheadrightarrow (НомКом+НомТел)$, т.е. в одной комнате может быть много телефонных номеров; это означает, что уникальному значению атрибута *КодСотр* соответствует неуникальное значение конкатенации атрибутов (*НомКом+НомТел*), и наоборот. Иными словами, означает полную независимость ТЕЛЕФОНОВ и от КОМНАТ, и от СОТРУДНИКОВ, а также СОТРУДНИКОВ от КОМНАТ. А отношение моделирует полную связь всех сущностей-объектов между собой. Таблица не приведена в связи с ее объемом и тривиальностью. Аналогичная таблица приведена для сущности-объекта «экзамен» в [5].

ПрО с такими связями встречается крайне редко, это «биржа», «библиотека», «выставочный центр», «супермаркет» и др., где распределение некоторых групп сотрудников по помещениям и телефонам осуществляется или по сменам, или вообще случайным образом. Очевидно, что в этой ПрО ситуативно может возникнуть единственная ФЗ: $КодСотр \rightarrow НомКом$, которая также будет условной. В общем случае любой сотрудник может находиться в любой комнате и быть приглашенным к любому аппарату, находящемуся в любой комнате. Иными словами, в пятой, шестой и седьмой комнатах параллельно подключено шесть телефонных номеров, а работает в них посменно то по 10 сотрудников, то по 30, то по 2. Тогда очевидно, что данное отношение является тернарной связью атомарных сущностей-объектов со степенью связи $F:H:O$.

Требование наличия в отношении единственной ФЗ обуславливает особое ограничение — ключом отношения должен быть не атрибут *КодСотр*, а конкатенация всех атрибутов ($КодСотр+НомКом+НомТел$), т.е. весь кортеж. Тогда необходимо отыскать в ПрО функционально зависящий атрибут такой связи от этого ключа, который и приведет данное отношение к 5НФ.

С формальной точки зрения для примера отношения в [11] не найдется ни одного составного многоместного предиката, соответствующего данной связи. Характерно и то, что в этом отношении будут потеряны кортежи, где для некоторых значений *КодСотр* не найдется ни одного значения атрибута *НомТел*, если данный сотрудник попросту нетелефонизирован (уборщицы, внештатные сотрудники и т.п.).

Эти аномалии обнаруживаются потому, что автор [11] в одно отношение внес два высказывания. Приведенное отношение — это и не «многоарное» высказывание-связь с темой: «сотрудники на рабочих местах с телефонами», и не «унарное» высказывание с темой: «сотрудники». Все это означает, что автор [11] не определился, связь каких сущностей-объектов моделирует данное отношение.

В общем случае отношение, моделирующее связь сущностей-объектов и не

находящееся в достаточной для соответствующих ограничений на домены и ключи совокупности отношений, может быть искусственной конструкцией, лишенной смысла. Поэтому если отношение моделирует ситуативную связь в ПрО (ситуацию в ПрО, состояние ПрО), то требуется также и учет темпоральности — номер отрезка времени, дату, час, код актуальности состояния всей совокупности отношений и т.п. В каркасной совокупности эти атрибуты могут быть шунтирующими.

Поскольку в каркасной совокупности формируется строгая иерархия доменов, описанные в [9–11] ситуации не могут возникнуть, так как доменами для ключевых атрибутов кортежей бинарных отношений выступают ключевые атрибуты кортежей унарных отношений, для тернарных — ключевые атрибуты кортежей или унарных, или бинарных, для кварternых аналогично и т.д.

ОСНОВНЫЕ СОВПАДЕНИЯ С РЕЗУЛЬТАТАМИ КЛАССИЧЕСКИХ ИССЛЕДОВАНИЙ

В статье [7] указаны работы, в которых рассматривались аналогичные вопросы, но отдельно хотелось бы перечислить более важные и перспективные совпадения.

Модель «сущность-связь». Во многих учебниках по БД [3, 4] ER-модель П. Чена [1] трактуется как концептуальная семантическая надстройка над РМД, хотя ничто не мешало «перевести» РМД на концепцию Чена и тем самым расширить ее. О том, что каждое отношение может моделировать не только «стержневую» [2] сущность-объект, но каждую ее связь, П. Чен указывает прямо и недвусмысленно, поэтому совпадение каркасного механизма анализа ПрО и методики ER-диаграмм неслучайно.

Однако Э. Кодд в 1979 г., спустя три года после выхода работы [1], в статье [2] расширил РМД, используя при этом результаты иных авторов, и тем самым отдал модель от пользователя. «Ведение в расширенную реляционную модель базы данных многих разнородных дополнительных элементов значительно усложняет описание лежащей в ее основе алгебраической структуры и построение соответствующего языка исчисления предикатов первого порядка» [13].

Основное назначение каркасной модели данных, не выходя за рамки классической РМД, и даже будучи ее подмножеством, — дать пользователю механизм наиболее простого моделирования семантики, пусть даже ценой отказа от Коддовской [12] «неизбыточности» в совокупности отношений, строго придерживаясь принципа максимально возможной нормализованности каждого отношения отдельно и совокупности отношений в целом.

ДНФ (доменная нормальная форма) как модель высказываний. В описываемом алгоритме [5] все сущности-объекты делятся на следующие категории.

Атомарные сущности-объекты — сущности-объекты, значения атрибутов которых практически не изменяются во времени в моделируемой ПрО, поэтому определяют структуру ПрО. Они также не имеют зависимости существования [14]. В некоторых моделях данных их называют стержневыми [2], базовыми и т.п.

Формально атомарная сущность-объект — семантически атомарный M_j -местный предикат [15] $I_j^{M_j}(\{A_j^m\})$, где $\{A_j^m\}$ — аргументы предиката — совокупность атрибутов j -й сущности-объекта (в дальнейшем обозначение совокупности (фигурные скобки) для простоты опускаем), где $j = 1, N$ — номер сущности-объекта в ПрО, а $m = 1, M_j$ — номер места в M_j -местном предикате.

В [15] показано, что семантическая атомарность — отсутствие внутренней семантической структуры предикатов, моделирующих атомарные сущности-объекты: никакие отдельные аргументы из подмножества аргументов не фигурируют ни в каких зависимостях — как одно с другим, так и с подмножествами аргументов других семантически атомарных предикатов. Тогда унарное отношение $R_j(X_j, A_j^m)$, эквивалентное семантически атомарному предикату, не содержит никаких «паразитных» зависимостей.

В [15] введено понятие **ключ** семантически атомарного предиката $\alpha_j(x_j)$ — такой одноместный предикат, аргументом которого является минимально доста-

точный унарный суррогатный ключевой атрибут X_j схемы отношения $R_j(X_j, A_j^m)$ такой, что $X_j \rightarrow A_j^m$. Тогда семантически атомарный предикат может быть представлен в виде конъюнкции: $I_j^{M_j}(A_j^m) = \alpha_j(x_j) \cap D_j^{M_j}(A_j^m)$, а вся N -совокупность P_I атомарных сущностей-объектов — так называемая «статическая» часть ПрО (структура системы P) — это предикат, полученный дизъюнкцией атомарных N -местных предикатов:

$$P_I = \cup I_j^{M_j}(A_j^m) = \cup (\alpha_j(x_j) \cap D_j^{M_j}(A_j^m)), \quad j = \overline{1, N}.$$

Примером атомарных сущностей-объектов может быть сущность-объект ЧЕЛОВЕК, ВСЕЛЕННАЯ, СОБАКА, КОШКА и т.п., причем принадлежность этих сущностей-объектов к определенным дальнейшим категориям — так называемая классификация атомарных сущностей-объектов — является искусственной семантической надстройкой пользователя, которая и маскирует содержание сущности-объекта.

Слабые сущности-объекты — сущности-объекты, которые функционально зависят от атомарных [14], в других моделях имеют аналогичное название, причем такая зависимость может быть как на уровне идентификации слабых атрибутов, так и на уровне всего существования зависимых слабых сущностей-объектов. В начальном приближении модели ПрО слабые сущности-объекты условно отнесены к атомарным, так как их атрибуты также не зависят от времени.

Примером слабых, но, тем не менее, условно относимых к категории атомарных, могут быть сущности-объекты ПОДРАЗДЕЛЕНИЕ, ОТДЕЛ, ЛАБОРАТОРИЯ, КВАРТИРА, каждая из этих сущностей-объектов не является самодостаточной, и в ПрО функционально зависит от «предковых» сущностей-объектов; однако зависимостью ее атрибутов от времени в ПрО можно пренебречь.

Составные (постсвязные) сущности-объекты в моделях данных имеют еще и название многосторонних связей [1, 14] — связь атомарных сущностей-объектов. Формально это семантически составной N -местный предикат, полученный конъюнкцией ключевых N -местных предикатов от связывающихся атомарных сущностей-объектов (N -местных предикатов), а также обладающий еще и аргументами связи. Тогда некоторый фрагмент множества всех подмножеств семантически составных предикатов (фрагмент булеана связей атомарных сущностей-объектов) запишем в схематическом виде:

$$\begin{aligned} S_2^{(l_2+2)}(x_1, x_2, b_1^2, b_2^2, \dots, b_{l_2}^2) &= \alpha_1(x_1) \cap \alpha_2(x_2) \cap T_2^{l_2}(b_{l_2}^2), \\ S_3^{(l_3+3)}(x_1, x_2, x_3, b_1^3, b_2^3, \dots, b_{l_3}^3) &= \alpha_1(x_1) \cap \alpha_2(x_2) \cap \alpha_3(x_3) \cap T_3^{l_3}(b_{l_3}^3), \\ &\dots \\ S_j^{(l_j+j)}(x_1, x_2, x_3, \dots, x_j, b_1^j, b_2^j, \dots, b_{l_j}^j) &= \\ &= \alpha_1(x_1) \cap \alpha_2(x_2) \cap \alpha_3(x_3) \dots \cap \alpha_j(x_j) \cap T_j^{l_j}(b_{l_j}^j), \\ &\dots \\ S_{j_3}^{(l_{j_3}+j-2)}(x_3, \dots, x_j, b_1^{j_3}, b_2^{j_3}, \dots, b_{l_{j_3}}^{j_3}) &= \alpha_3(x_3) \dots \cap \alpha_j(x_j) \cap T_{j_3}^{l_{j_3}}(b_{l_{j_3}}^{j_3}). \\ &\dots \end{aligned}$$

(Индексы будут описаны ниже.)

Примером составных сущностей-объектов являются событийные сущности-объекты — ЭКЗАМЕН, КОНЦЕРТ, ВЫСТАВКА, СОГЛАШЕНИЕ, МИТИНГ и т.п. Их содержание представляет собой «продукт» равноправного взаимодействия нескольких атомарных сущностей-объектов. Таким образом, в каркасном алгоритме сущности-объекты формируются по следующей схеме: на базе атомарных (стержневых, базовых и т.п.) порождаются слабые, т.е. функционально (иерархически) зави-

симые от атомарных, но статичные во времени. Отношения, моделирующие слабые сущности-объекты, также имеют составной ключ. А на совокупности атомарных и слабых сущностей-объектов благодаря возникновению разнообразных связей между ними создаются составные постсвязные сущности-объекты, характеризующие поведение ПрО. Ранее отмечалось, что в настоящей работе временные атрибуты не фигурируют, а описываются лишь статические срезы ПрО.

В произвольной ПрО описанный процесс образования слабых и составных сущностей-объектов, как правило, маскируют части речи — существительные, отглагольные существительные, разнообразные термины, которые им соответствуют, категории, которые их обобщают, и т.п. Этим и объясняется актуальность автоматизации описанного алгоритма.

Часто подавляющее большинство составных сущностей-объектов проектировщики ошибочно относят к категории слабых или даже атомарных, что, в свою очередь, увеличивает жесткость приложений БД и невозможность их гибкого развития без коренных переработок текстов программ.

Каркасная модель ПрО предоставляет проектировщику совокупность отношений, каждое из которых моделирует либо статическое состояние сущности-объекта (тривиальную связь), либо одну из связей совокупности сущностей-объектов.

Любой семантически атомарный многоместный предикат рассматривается как связь его атрибутов, т.е. по сути как тип тривиальной связи «самого с собой»; $I_j^{M_j}(A_j^m)$ — тривиальная связь. Нетривиальная связь — конъюнкция ключевых атрибутов от разных семантически атомарных многоместных предикатов, имеет вид

$$S_j^{(l_j+j)}(x_1, x_2, x_3, \dots, x_j, b_1^j, b_2^j, \dots, b_{l_j}^j) = \\ = \alpha_1(x_1) \cap \alpha_2(x_2) \cap \alpha_3(x_3) \cap \dots \cap \alpha_j(x_j) \cap T_j^{l_j}(b_{l_j}^j),$$

где $b_{l_j}^j$ — атрибуты связи, моделируемой $S_j^{(l_j+j)}$ -м предикатом, обладающим $(l_j + j)$ -местами, j из которых занимают ключи с первого по j -й, а с $(j+1)$ -го по $(l_j + j)$ -й занимают аргументы связи — шунтирующие атрибуты соответствующего отношения $R_j(X_1, X_2, X_3, \dots, X_j, B_1^j, B_2^j, \dots, B_{l_j}^j)$. Тогда в общем виде модель ПрО можно представить так:

- $P_I = I_1^{M_1}(A_1^m) \cup I_2^{M_2}(A_2^m) \cup \dots \cup I_N^{M_N}(A_N^m)$ — структура ПрО («бесконечно долгие» по времени связи ПрО);
- $P_s = S_2^2 \cup S_2^3 \cup S_2^4 \dots \cup S_2^{L_2} \cup S_3^2 \cup S_3^3 \cup S_3^4 \dots \cup S_3^{L_3} \dots \cup S_N^2 \dots \cup S_N^{L_n}$ — текущие связи ПрО: $P = P_I \cup P_s$.

Таким образом, ПрО как многоместный предикат P есть дизъюнкция конъюнкций атомарных предикатов.

С одной стороны, это и есть актуальная часть каркаса, с другой, это классическая нормальная форма высказывания в булевой логике, широко используемая при исследованиях в смежных областях, в частности проблемах искусственного интеллекта [16]. Но совокупность отношений ПрО, полученных классическим методом декомпозиции [12], не сводима к совокупности высказываний в ДНФ, так как лишь бинарные отношения не применяются в полноценных «многоарных» высказываниях.

Использование тернарных таблиц. Особое значение приобретает «теорема о шунтировании ДЗ (МЗ, ЗПС)» [5] в изложении Д. Кренке: «Строка любого отношения должна отражать все необходимые данные, описывающие конкретный экземпляр темы, описываемой данным отношением» [3]. Для ДЗ (МЗ, ЗПС) это означает, что если отсутствуют непустые атрибуты связи, то нет возможности определить тему. Именно в этом суть аномалии такого неактуального (нешунтированного) отношения по Д. Кренке. Именно поэтому такое отношение должно быть декомпозировано, т.е. преобразовано к нескольким актуальным.

В исследованиях некоторых авторов встречались отношения, моделирующие связи степени $F:H$, построенные на принципе шунтирования МЗ. Так, в [17] приведено отношение-связь $ЗАЧЕТЫ = СЕМЕСТР + КУРС + СТУДЕНТ$ (табл. 4).

Автор поясняет, что «с помощью отношения $ЗАЧЕТЫ$ устанавливается связь между отношениями $СЕМЕСТР$, $КУРС$ и $СТУДЕНТ$. Первичный ключ отноше-

Таблица 4. ЗАЧЕТЫ

КодСеместр	НомКурса	КодСтуд	Зачеты
01	ВТ608	967831201	4
01	ВТ608	897645123	2
01	ВТ62Э	967831201	3
01	ОФ500	967831201	4
02	ВТ601	123456789	3
02	ВТ601	897645123	2
02	ВТ605	123456789	2
02	ВТ623	967831201	3
02	ОФ500	897645123	3

ния составлен из атрибутов $КодСеместр$, $НомКурса$ и $КодСтуд$. Неключевой атрибут один — $Зачеты$. В течение одного семестра преподается несколько курсов. Каждый курс изучают несколько студентов. С другой стороны, в течение данного семестра каждый студент может изучать несколько курсов. Эта таблица позволяет реализовать взаи-

мосвязь «многие ко многим» между семестрами, курсами и студентами» [17].

В дальнейших работах автор [17] не делает вывода о том, что табл. 4 не имеет ни МЗ, ни ЗПС и поэтому не может быть подвергнута декомпозиции.

Автор работы [18] указывает на возможность в рамках РМД использовать отношения с единственной ФЗ между «многозначной» конкатенацией совокупности атрибутов-ключей и неключевыми атрибутами. Однако делается ложный вывод, что такие отношения можно отнести к категории «денормализованных». Поэтому в дальнейших исследованиях этого автора принцип шунтирования МЗ и ЗПС не развивается и соответствующая совокупность отношений не исследуется. «...Согласно теореме Р. Фейджина [8] таблицу R с тремя множествами атрибутов (A, B, C) и многозначной зависимостью $A \twoheadrightarrow B / C$ можно декомпозировать на две таблицы: (A, B) и (A, C) . Но если в таблице R есть четвертое множество атрибутов D и функциональная зависимость $(A, B, C) \rightarrow D$, то выполнить декомпозицию таблицы R нельзя, а это, в свою очередь, означает, что при использовании классического метода могут быть таблицы в БКНФ, но не в 4НФ, о которых известно, что они подвержены аномалиям модификации данных [8], но ничего с ними сделать нельзя. В качестве конкретного примера такой таблицы можно взять таблицу «РУБАШКА» с атрибутами: модель, цвет, размер, количество». Однако совпадения схем отношений подтверждает наличие потребности у проектировщика использовать многоарное отношение как модель многосторонней связи степени $F: H: \dots : Z$. Этот же тип отношений использовался и в работе [19].

CASE-генерация метаданных: системы «МикроПоиск» и SWS. Разработчики двух разных инструментальных программных систем — CASE «МикроПоиск» [19–21] и CASE-генератора SWS [22, 23] — обнаружили в РМД возможность реализации двух разных, однако взаимодополняющих подходов к проектированию приложений БД. Это механизм семантических графовых запросов к частично оптимизированной схеме БД (не выше БКНФ и даже ЗНФ) и механизм типизации большинства запросов, унификации основных алгоритмов, с ними связанных, и «упрятивания» этих алгоритмов в оптимальную схему БД (на основе ДКНФ).

Поскольку оба подхода формально обоснованы, хорошо протестированы, широко внедрены и претендуют на некоторую универсальность, они в некотором смысле конкурируют между собой. Однако максимальную эффективность пользователь получает именно при объединении этих подходов. Основной критерий применимости подходов — это коэффициент прогнозирования развития ПрО и разнообразия запросов пользователей. Под таким коэффициентом понимается отношение числа подтвержденных изменений к суммарному числу прогнозируемых и спонтанных изменений за определенный период времени, например год. Очевидно, что в такой формулировке число подтвержденных изменений не может превышать числа прогнозируемых.

Показательно, что удовлетворительной для второго подхода является ПрО с коэффициентом, равным единице, т.е. те ПрО, которые можно досконально изучать и развивать, прогнозировать, моделировать с помощью оптимизированной РМД. И приложения, моделирующие процессы ПрО, могут иметь минимальный интерфейс для низкоквалифицированного пользователя, позволяющий минимизировать (или практически исключить) использование механизма внешних запросов к БД. К таким ПрО можно отнести всевозможные бизнес-приложения компаний и корпораций, которые развиваются в соответствии с прогнозируемыми рыночными факторами, но для синтеза ДКНФ-схем необходимо исключить все замаскированные составные сущности-объекты, что, однако, не всегда реализуемо. Это означает, что в реальной схеме БД возможны участки не выше ЗНФ. Компромисс между оптимальными и низконормализованными схемами (до ЗНФ) достигается использованием на таких участках механизма семантических графовых запросов.

Что же касается ПрО с малым коэффициентом прогнозирования развития ПрО и разнообразия запросов пользователей, как, например, поисковые машины в Интернет, всевозможные социальные сети, Интернет-витрины данных, схемы БД которых зависят не от реальных причинно-следственных связей в ПрО и их владельцев, а от хаотически обращающихся пользователей, запросы эффективно моделируются гибкими языковыми конструкциями. В приложениях для таких ПрО бесспорно преимущество механизма семантических графовых запросов. Эту проблему решает CASE-система «МикроПоиск» [19–21].

Особенность «МикроПоиска» состоит в применении SQL и подобного языка запросов, ориентированного на ER-модель данных П. Чена [1]; интенционал (схема БД) представляется графом, вершинам которого соответствуют классы объектов (сущностей), а ребрам — классы бинарных связей между объектами. Элементарные атрибуты объектов соответствуют основным типам данных. Допускаются атрибуты, содержащие тексты произвольной длины (целые документы). Поименованная совокупность элементарных атрибутов может составлять повторяющуюся группу. Аппарат повторяющихся групп реализует зависимость по существованию.

В [21] сказано следующее: «Связи имеют тип: $I:I$, $I:H$, $F:H$. Обычная операция над связями — переход по связи от множества экземпляров одного класса объектов; результат — множество соответствующих экземпляров другого класса связанных объектов (переход разрешен в обоих направлениях). При наличии связи между однотипными объектами направление уточняется с помощью роли. Например, чтобы получить список подчиненных данного сотрудника, нужно перейти по связи *СУБОРДИНАЦИЯ* с указанием роли *ПОДЧИНЕННЫЙ*, а чтобы получить его начальника — по той же связи с указанием роли *НАЧАЛЬНИК*».

Иное решение предоставляет каркасная модель. Каркас является частным случаем РМД. Причем, как показано выше, одним из важных свойств реляционного каркаса является возможность минимизировать объем запросов к БД, построенных на громоздких и вычислительно сложных операциях соединения.

Как показывают исследования, проведенные на большом количестве внедренных приложений БД, схема которой построена на каркасе, моделирует до 90 % запросов без операции соединения и ее модификаций. Это позволяет значительную часть данных обрабатывать по заранее сформированным индексным таблицам, а подавляющее большинство запросов пользователей также формализовать, унифицировать и интегрировать в приложении. Такой подход дает возможность построить универсальную перенастраиваемую оболочку, управляемую группой метаданных, а массивы метаданных, отражающие ту или иную специфику ПрО, синтезировать с помощью программной оболочки по тем же высоконормализованным гибким схемам БД.

Исследования ПрО автоматизации проектирования приложений, проводимые на протяжении длительного времени, показывают, что список основных

операций реляционной алгебры, которые необходимо использовать, существенно сокращается. Сокращается также и группа операций по отслеживанию целостности БД. Это происходит, поскольку основными инструментами манипулирования данными служат индексный поиск и операция селекции данных.

Еще одной особенностью схемы БД, полученной с помощью каркасного анализа, является возможность моделирования связи любой кратности между сущностями-объектами со степенью $F:H$. Эта максимально возможная степень связи лежит в основе анализа любой совокупности связей для любой ПрО; связи с меньшей степенью моделируются как частный случай.

При проектировании инструментального средства и приложений использовалась единая схема каждого отношения — схема, построенная на конкатенации (строковой сумме) унарных ключевых атрибутов типа $A + B + C + D \dots + Z \dots$. Известно, что для такой конструкции могут быть построены элементарные индексные деревья. Именно такой механизм проектирования может быть унифицирован вплоть до стандартизации.

Таким образом, CASE-оболочка генерации метаданных SWS (Server Workstation System) [22, 23] — это не что иное как генератор отношений, в каждом из которых унифицированным способом хранятся метаданные конкретной ПрО: имена отношений, имена полей, группы индексных заголовков и другие специализированные данные.

Принципиально важным признаком каркасной группы отношений является иерархическая связь кортежа из верхнего отношения с пачкой кортежей нижнего отношения, причем эта связь осуществляется по индексированному общему ключевому атрибуту. Именно к такому атомарному сочетанию сводится вся совокупность отношений, именно к такой паре «отношение–отношение» в конечном итоге сводится любой фрагмент ПрО. Дальнейшая унификация каскадных процедур уже не является сложным процессом.

Хотя описанные инструментальные средства имеют некоторые отличия, они сходны в одном — решают задачу автоматизации не только проектирования приложений БД, но и их программирования, причем сходным способом.

Исследование закономерностей семантического ER-проектирования схем БД для разных ПрО, а также обобщение и унификация многообразия запросов пользователей позволило отказаться от подхода автоматизированного синтеза исходных текстов программ и реализовать предложенный академиком В.М. Глушковым тезис «программирование без программистов», когда готовое работающее приложение БД непосредственно формируется инструментальной оболочкой, управляемой метаданными пользователя. Такой подход крайне актуален в связи с появлением распределенных вычислительных систем сверхмалых архитектур.

ОСНОВНЫЕ СОВПАДЕНИЯ С СОВРЕМЕННЫМИ ТЕНДЕНЦИЯМИ В БД

Проблема объединения в рамках единой модели данных свойств, присущих схемам реляционной и объектно-ориентированной (ОО) БД, актуальна на протяжении нескольких десятилетий [24, 25]. Описанная теория каркасной модели данных позволяет иначе решить эту проблему. Новая модель одновременно обладает SQL- и OQL-полнотой [14], позволяет проектировать информационные системы, имеющие объединенные свойства, а теорема о модифицируемой ДКНФ-схеме БД [7] дает возможность рассматривать каркасную модель данных как единственно оптимальную платформу для такого объединения.

Возможность положительного результата такого объединения высказал Дейт в [24], где отношения реляционной модели можно рассматривать как объектные классы. Он утверждает, что «в реляционной модели не существует ничто, что может ограничить скалярные значения, существующие в доменах, ... простыми формами ... Эти скалярные значения могут быть настолько сложными, насколько мы пожелаем».

В [5] указано, что под сущностью (объектом, фактом и т.п.) в ПрО понимается расширяемая совокупность атрибутов, объединенных единым уникальным в рамках этой ПрО многоместным предикатом, а под атрибутом сущности-объекта понимается некое абстрактное множество, объединенное одноместной частью этого уникального для ПрО многоместного предиката. Кроме того, описано одно из принципиальных свойств каркасной модели — фиксированная структура единого для каждого атрибута ключа, являющегося следствием многоместного предиката. Такой подход к декомпозиции ПрО дает возможность рассматривать каждый объект ПрО как отдельный класс сущностей-объектов со своим фиксированным классом ID, каждый экземпляр которого, во-первых, имеет строго фиксированную предопределенную порождающим предикатом схему, во-вторых, уникально определяет экземпляр атрибута. При работе с SQL-запросами такая БД форматируется как реляционно-табличная — все группы атрибутов-столбцов в строгом соответствии со своими ID могут быть объединены в отношения. При выполнении OQL-запроса можно использовать механизм индексирования базы по структуре ID, что повышает скорость выполнения запроса.

В теореме 6, о модифицируемости реляционного каркаса из [6], рассмотрена классическая [26] задача о «неожиданном» дополнении сущности-объекта новым атрибутом, т.е. о модификации схемы БД [5,15]. Автор [26] для этой проблемы определил красноречивую рубрику «смерть обеим», указывая на невозможность корректного моделирования ситуации ни в рамках классической реляционной модели [12], ни известным алгоритмом ООБД [27]. Для каркасной концепции подобные модификации являются естественным и корректным процессом, формально описанным теоремой, доказанной в [15].

Как указано в [5], модифицируемость схемы БД обеспечивается генерацией ключевых атрибутов по алгоритму сочетаний «всех со всеми», аналогичному булеану (множеству всех подмножеств). Показано, что любые агрегированные сущности-объекты, замаскированные под атомарные, продуцируются конечным набором атомарных. Такой подход обобщает процедуру моделирования ООБД [27], где любому искомому сущностному (объекту) ставится в соответствие только атомарный предикат (неструктурированный ID) независимо от семантики ПрО — внутренних взаимосвязей. Поэтому каркасная модель значительно расширяет возможности ООБД. Составные сущности-объекты — это постсвязные многоместные предикаты, а слабые сущности-объекты — цепочки иерархически-зависимых многоместных предикатов. Каркас моделирует единой процедурой оба типа.

Каркасный метод анализа ПрО предлагает подход более строгого моделирования сущностей-объектов. Как известно [27], теория проектирования ООБД не дает точных рекомендаций, какие сущности-объекты или факты из описания ПрО должны быть объектами, а какие лишь маскирующими категориями (словами). Важно, что такая методика применима к проектированию любого программного обеспечения — от приложений БД до произвольных кибернетических систем.

При таком подходе к проектированию приложений БД каждое отношение автоматически становится объектом, а их совокупность — иерархической структурой программной системы, поскольку любое сечение каркаса — это иерархия, древовидный граф, в узлах которого расположены отношения. Связи по дугам графа моделируются ключами отношений, каждое отношение моделирует сущность-объект.

Каркасный анализ позволяет разработчику приложений и программных комплексов общевычислительного характера, работающему не только с БД, проектировать структуру исходного текста программ и соответствующей библиотеки объектов так, чтобы дальнейшее сопровождение и обязательные постоянные модификации не вызывали ощутимых затруднений и значительных затрат ресурсов. Использование совокупности объектов и процедур, построенных в соответствии с каркасной схемой семантики специализированной ПрО — стационарных динамических задач теории упругости [28, 29], заложенное в листинг про-

граммного комплекса на начальных этапах (1990 г.), позволило значительно расширить и обобщить приложение, а также решить вопрос его интероперабельности (для кластеров СКИТ-3 и «Инпарком-256») за очень короткий отрезок времени.

Денормализация. Из приведенной в [30] подробной классификации наиболее схожим с проектированием безаномальных схем БД на реляционном каркасе является метод нисходящей и восходящей денормализации схем БД. Пример включения в отношение-связь атрибута атомарной сущности-объекта [30], функционально зависящего только от части составного ключа, вносит не только постороннюю ФЗ и нарушает условия нормализации отношения-связи, но является чуждым, исходя из семантики высказывания, моделируемого отношением-связью. В ней нарушается «тема» отношения.

Тем не менее при такой денормализации пользователь получает дополнительное конкурентное преимущество эксплуатации БД, поскольку скорость доступа к данным значительно повышается из-за исключения операций соединения, что и отмечается в работах [30, 31].

Каркасный синтез ДКНФ позволяет избегать денормализации, однако способствует использованию нормализованных отношений-связей, которые также снижают число операций соединения.

Рассмотрим подробнее пример из [31]. Если в ПрО лечебно-профилактического управления (*ПАЦИЕНТ, ТАЛОН, ПОЛ ПАЦИЕНТА*) регулярно синтезируются отчеты по неключевому атрибуту *Пол*, то может использоваться принцип «маска для роли атомарной сущности-объекта» [5] — отдельное отношение, которое является частичной копией атомарного отношения *ПАЦИЕНТ*. При этом формируется отношение-связь *ПОЛ ПАЦИЕНТА+ТАЛОН*, которое приложением БД поддерживается в целостном и актуальном состоянии универсальными процедурами. В нем также может сохраняться необходимая статистика — формироваться неключевые шунтирующие атрибуты. И хотя такая декомпозиция отношения *ПАЦИЕНТ* на два отношения (*ПОЛ ПАЦИЕНТА* и, например, *ВОЗРАСТ ПАЦИЕНТА*) некоторыми авторами оценивается как денормализация [31], у Фейджина в [8] аналогичный прием приведен как пример нормализации схемы отношения: «...схема может быть разложена на две схемы...», кортеж, содержащий данные о поле человека, входит в первое отношение, а кортежи, содержащие данные о его профессии, входят во второе отношение. Схема *SEX_ROLE* удовлетворяет ФЗ *Person → Role*, в то же время схема *PROF_ROLE* не удовлетворяет этой зависимости. ... Преимущество разложения состоит в том, что механизм обработки ключей системы может предотвратить вставку двух ролей пола для одного человека».

При этом, как упоминалось в [31], тратится дополнительное время на поддержание ссылочной целостности. Однако эти потери, как и в [31], несоизмеримы с потерями при соединении. Но если необходимо выполнить разовый отчет, создается запрос с операцией «выбор по выбору». Осуществляется выборка из отношения-связи *ТАЛОНЫ ПАЦИЕНТОВ* (соединение, в котором условно находятся миллионы записей) только тех ключевых значений атрибута *КодПациента*, для которых отфильтрована маленькая группа кортежей («пачка записей») в отношении *ПАЦИЕНТ* по любой совокупности атрибутов (например, по значению нескольких неключевых полей «мальчики, младшие 12 лет»). «Умные» СУБД выполняют эту операцию значительно быстрее прямого соединения, поскольку операция сводится лишь к выбору в «большом» отношении-связи отфильтрованной по первичному ключу (по индексу) группы кортежей в соответствии с малой группой «родительского» отношения. Однако, к сожалению, далеко не все СУБД предоставляют такую возможность.

NoSQL — key-value БД. Современные модели БД типа «ключ–значение» (key–value — KV-модель), которые схожи с каркасной, относятся к новому, динамично развивающемуся направлению БД, объединенному красноречивым названием «не только SQL» (Not only SQL — NoSQL) [32]. Как правило, это рас-

пределенные хеш-таблицы. В этом заключается их наиболее сильная сторона, так как предоставляет высокую производительность и модифицируемость. Тем не менее эти свойства перестают быть преимуществами, когда необходимо обрабатывать списки.

В РМД нижеследующие задачи являются типовыми:

- выбрать k последних покупателей;
- выбрать самые популярные товары текущей недели;
- найти склад с максимальным оборотом за последний месяц;
- найти покупателя по любому неключевому атрибуту (по номеру мобильного телефона, по месту жительства и т.п.).

Но в KV-модели подобные задачи вызывают трудности. Например, когда необходимо осуществить выборку значений атрибутов одной сущности-объекта по вторичному ключу другой, ситуация становится схожей с подходом «нисходящей» или «восходящей» денормализации.

Таким образом, имеем тот класс задач, для которых можно и нужно использовать РМД. На практике, поскольку такие задачи взаимосвязаны, проектировщик передает БД в РМД-приложение, и, как правило, применяется гибридное решение — комбинация KV-РМД. Все данные хранятся в KV, а те свойства, по которым необходимо осуществить агрегатные выборки, дублируются в РБД (с указателем на ключ). Но сложность заключается в том, что приходится следить за синхронизацией данных в двух БД, что очень усложняет логику приложения.

Схема реляционной БД, полученная на каркасной модели данных, автоматически обладает свойствами, подходящими и под модель NoSQL. Единственный структурированный ключ каждого отношения каркасной совокупности, полученный конъюнкцией одноместных ключевых предикатов (или, что то же самое, конкатенацией унарных ключевых атрибутов), являющийся ключом отношения для SQL-запроса от приложения РМД, одновременно является и KV для соответствующих значений неключевых атрибутов при нетабличном запросе к БД от KV-приложения, а также, как указывалось выше, и ID для OQL-запроса.

ЗАКЛЮЧЕНИЕ

Особое отношение со схемой $R(X, A)$ является минимально возможным в РМД и тем самым частным случаем отношения. Но, как неоднократно отмечалось [5–7, 15], в РМД понятие частного случая имеет некоторую двойственность. Мы описали [5–7, 15], как минимум, четыре однотипных ситуации.

1. Частный случай, когда правая часть составного детерминанта является непустым множеством, позволил получить достаточно нетривиальный обобщающий результат с вполне определенным физическим смыслом — атрибутом многоарной связи.

2. Частный случай многозначности (ДЗ) позволил получить обладающее полнотой обобщающее отношение-связь, которое можно использовать как шаблон для связей максимально возможной степени и произвольной кратности.

3. Частный случай совокупности отношений со схемой $R(X_1, X_2, X_3, \dots, X_n, A_j)$ и исключительно одной ключевой ФЗ позволил получить уникальный по своей универсальности реляционный каркас, пригодный при использовании в качестве шаблона практически для любой ПрО.

4. Частный случай ограничений на домены и ключи — минимально возможная ФЗ $X \rightarrow A$ — позволила обнаружить элементарную безаномальную схему.

По-видимому, без преувеличения можно считать, что эти частные случаи взаимосвязаны и играют фундаментальную роль в РМД. Рассмотренные в [5–7, 15] важные свойства каркасной модели данных и каркасного метода проектирования схемы БД обосновываются именно этим обстоятельством.

К основным конкурентным преимуществам этого подхода относятся следующие свойства (по убыванию значимости):

- безаномальность в смысле [8];
- высокая скорость доступа к данным благодаря значительному снижению операций соединения в запросах;

- динамическая модифицируемость схемы БД, что позволяет корректно вносить изменения в эксплуатируемое приложение, а также гибко модифицировать схемы не только пользовательских, но и управляющих БД (метабаз);
- подобие схем БД для разных ПрО, несмотря на значительные отличия семантик, что обеспечивает интероперабельность и кроссплатформность приложений [21].

Для исследователей также важны следующие выводы.

- ДКНФ-схема БД — это также и схема семантики ПрО; формируя ДКНФ-схему БД, пользователь исследует семантику ПрО и тем самым имеет возможность оптимизировать саму ПрО.

- Совпадение с результатами исследований [19–21] в части «отдельных вспомогательных тернарных отношений», «ускорения доступа к данным ввиду экономии на соединениях», «схожести схем БД разных ПрО» неслучайно. Оптимальная схема БД единственна.

- Основная классическая концепция «все связи должны быть декомпонованы к бинарным, а при подготовке ответа на запрос сервер осуществит соединения» устарела. Более актуальна концепция «все актуальные связи поддерживаются в режиме онлайн».

- Каждое отношение в ДКНФ-схеме — это одно предложение, соответствующее одной «теме» [3], а их совокупность соответствует ДНФ совокупности высказываний.

- Любой текст — от одного атомарного предложения и до произвольного их количества, представляющий пользователю информацию, является фрагментом каркаса. Это означает, что каркасный подход позволяет синтезировать автоматизированный интерпретатор смыслов.

Все перечисленное дает возможность решить проблему унификации, типизации и минимизации СУБД, т.е. разработать инструментальную оболочку, которая масштабируется метаданными, и на этом основании создать малогабаритный CASE-генератор каркасных метаданных, управляющих универсальной каркасной оболочкой, а также снабдить этот гибкий инструмент библиотекой типовых функций навигации каркасной РБД, используемых в унифицированных запросах.

Такой подход минимизирует потребность в ресурсоемких операциях соединения в большинстве запросов к БД и существенно упрощает настройку приложения.

Автор выражает благодарность чл.-кор. НАН Украины, профессору О.Л. Перевозчиковой, академику НАН Украины, профессору А.А. Летичевскому, академику НАН Украины, профессору Ф.И. Андону, доктору техн. наук, профессору С.Д. Кузнецову (РАН); доктору физ.-мат. наук, профессору Д.Б. Бую.

СПИСОК ЛИТЕРАТУРЫ

1. Chen P. P. The Entity-Relationship Model: toward a unified view of data // ACM Trans. on Database systems. — 1976. — 1, № 1. — P. 9–36.
2. Codd E. F. Extending the database relational model to capture more meaning // Ibid. — 1979. — 4, N 4. — P. 397–434.
3. Кренке Д. М. Теория и практика построения баз данных. — СПб.: Питер, 2003. — 800 с.
4. Кузнецов С. Д. Базы данных: модели и языки. — М.: Бино, 2008. — 720 с.
5. Карпуша В. Д., Панченко Б. Е. Моделирование и проектирование реляционных баз данных: Уч. пособие. — Сумы: Изд. СумДУ. — 2010. — 385 с
6. Панченко Б. Е. Каркасное проектирование доменно-ключевой схемы реляционной базы данных // Кибернетика и системный анализ. — 2012. — № 3. — С. 120–129.
7. Панченко Б. Е. К вопросу о модифицируемости и безаномальности схемы реляционной базы данных // Проблемы программирования. — 2012. — № 3. — С. 281–288.
8. Fagin R. A normal form for relational databases that is based on domains and keys // ACM Trans. on Database Systems. — 1981. — 6, N 3. — P. 387–415.
9. Тарасов И. А. Целостность данных, аномалии модификации данных и нормальные формы таблиц реляционных баз данных // Проектирование телекоммуникационных и информационных средств и систем. — М.: МИЭМ, 2007. — С. 195.

10. Тарасов И. А. Недостатки теории доменно-ключевой нормальной формы (DKNF) Рональда Фагина. — <http://itsoft.ru/about/publication/dbdesign/dknf/>.
11. Голосов А. О. Аномалии в реляционных базах данных // СУБД. — 1996. — № 3. — С. 23–38.
12. Codd E. F. The relational model for database management: Ver. 2, Reading Mass. — New York: Addison-Wesley Publ. Co, 1990. — 538 p.
13. Цаленко М. Ш. Семантические и математические модели баз данных // Итоги науки и техники. Информатика. Т. 9. — М.: ВИНТИ, 1985. — 207 с.
14. Ульман Д. Д., Уидом Д. Основы реляционных баз данных. — М.: Лори, 2006. — 374 с.
15. Панченко Б. Е., Писанко И. Н. Свойства реляционного каркаса на множестве семантически атомарных предикатов // Кибернетика и системный анализ. — 2009. — № 6. — С. 120–129.
16. Яловец А. Л. Представление и обработка знаний с точки зрения математического моделирования. Проблемы и решения. — Киев: Наук. думка, 2011. — 360 с.
17. Костюк А. И. Базы данных и знаний: Курс лекций. — Таганрог: ТРТУ, 1999. — 175 с.
18. Тарасов И. А. Метод проектирования логической структуры реляционной БД для веб-приложений без нормализации таблиц: Дисс. канд. техн. наук. — М.: МИЭМ, 2009. — 112 с.
19. Гречко В. О., Темперанский В. А. Некоторые вопросы организации данных в системном анализе сложных объектов // Автоматизация проектирования информационных систем. — Киев: ИК АН УССР, 1976. — С. 78–85.
20. Grechko V., Tulchinsky V. Building standard user interface: DBMS “MicroPoisk” approach // Informatica. — 1995. — N 4. — P. 446–456.
21. Перевозчикова О. Л., Тульчинский В. Г., Коломиец А. В. и др. Высокопродуктивные методы анализа и спецификации пространств атрибутов предметной области для организации вычислений: Отчет о НИР № 0107U000800 ВФ.145.09.11. — Киев, 2011. — 378 с.
22. Панченко Б. Е., Гайдабрус В. Н., Церковицкий С. Л. Создание сетевых информационных комплексов // Компьютеры плюс программы. — 1993. — № 5 (6). — С. 46–50.
23. Панченко Б. Е., Гайдабрус В. Н., Церковицкий С. Л. Сетевые вычислительные комплексы // Там же. — 1994. Спец. вып. — С. 30–37.
24. Date C. J. Date on database: writings 2000-2006. — New York: Apress, 2006. — 539 p.
25. Чемберлин Д. Анатомия объектно-реляционных баз данных // СУБД. — 1998. — № 1–2. — С. 65–76.
26. Пржиялковский В. Что объектам здорово, то реляциям смерть, и наоборот, и еще пол-оборота. — <http://open.oracle.tu2.ru/OR.html>.
27. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. — М.: Бином, 2001. — 560 с.
28. Назаренко А. М., Панченко Б. Е. Дифракция волн сдвига на цилиндрических неоднородностях произвольного поперечного сечения // Динамика и прочность машин. — 1991. — Вып. 52. — С. 38–45.
29. Панченко Б. Е. Высокоточное кластерное решение задачи дифракции волн сдвига на системе отверстий в полубесконечной изотропной среде с защемленной границей // Проблемы программирования. — 2012. — № 1. — С. 121–131.
30. Кунгурцев А. Б., Зиноватная С. Л. Модель реструктуризации реляционной базы данных путем денормализации схемы отношений // Тр. Одес. политехн. ун-та. — 2006. — 2(26). — С. 105–111.
31. Кунгурцев А. Б., Зиноватная С. Л. Анализ целесообразности реструктуризации базы данных методом введения нисходящей денормализации // Там же. — 2006. — 1(25). — С. 104–108.
32. Stonebraker M. The NoSQL discussion has nothing to do with SQL // BLOG@CACM. — <http://cacm.acm.org/blogs/blog-cacm/50678-the-nosql-discussion-has-nothing-to-do-with-sql/fulltext>.

Поступила 27.02.2012