

АЛГОРИТМ СИНТЕЗА РЕЛЯЦИОННОГО КАРКАСА. НЕФОРМАЛЬНОЕ ОПИСАНИЕ

Введение

Предложенный в работе [1] реляционный каркас, синтезированный на основе декартовой зависимости ключевых атрибутов, позволяет моделировать связи степенью $H : G$. Показан обобщенный алгоритм синтеза схем баз данных (БД), который моделирует предметные области (ПрО) и позволяет использовать единую процедуру автоматизации процесса создания схемы БД.

Такая процедура обеспечивает модифицируемость схемы [2], т.е. минимизирует число операций для внесения изменений в динамическом режиме — непосредственно в процессе эксплуатации приложения, а также оптимизирует объединение различных приложений, построенных в соответствии с этим алгоритмом, в единую информационную систему.

В [3] показано, что предложенный алгоритм может использоваться для автоматизированного проектирования высокоформализованных и безаномальных в смысле [4] схем БД. Однако его можно также использовать для проектирования программных систем, непосредственно не связанных с обработкой БД, как, например, в [5], а также устройств распознавания речи, устройств-переводчиков, экспертных систем, систем автоматизированного аудита приложений, поисковых систем в Интернет и приложений контент-анализа, а также иных кибернетических систем [6] и т.п.

Алгоритм синтеза схемы БД, построенный на разделении отношений-сущностей и отношений-связей, предложен также и в [7]. Однако, как отмечалось в [3], он имеет ряд недостатков.

Близким к предложенному является и алгоритм предварительного формального описания ПрО, который применен в некоторых онтологиях [8, 9]. Тем не менее в нем также имеется существенный недостаток — отсутствует единый фактор, который позволяет систематизировать семантику сущностей-объектов, т.е. сущствительных из описания ПрО.

Постановка задачи

Цель настоящей статьи — разработка алгоритма, который минимизирует число базовых категорий сущностей-объектов и позволяет проводить автоматизированное выделение сущностей-объектов из значительного числа терминов и синонимов из описания ПрО, а также относить их к той или иной категории. Будем называть такое разделение сущностей-объектов по категориям сепарацией сущностей-объектов. Алгоритм автоматизированной сепарации построен в соответствии с теорией каркасной модели данных [3]. При этом используется допущение, что для любой ПрО всегда существует ограниченная базовая совокупность сущностей-объектов, к которой относятся лишь атомарные и слабые сущности-объекты [10], а все иные сущности-объекты (которых практически всегда намного больше) синтезируются на этой совокупности благодаря каркасу связей, т.е. булеану всех подмножеств связей сущностей-объектов из базовой совокупности. Остальные сущности-объекты являются следствием функционирования этой ПрО.

Таким образом, цель автоматизированной сепарации сущностей-объектов моделируемой ПрО — получить каркасную совокупность отношений $R_l(X_i, A_i)$. Здесь X_i — конкатенированные в общем случае ключи каждого отношения, A_i — атрибуты сущностей-объектов или их связей, $l = \overline{1, L}$ — общее число отношений в схеме БД, $i = \overline{1, N}$, N — число атомарных сущностей-объектов. Каждое отношение соответствует той или иной атомарной, слабой или составной сущности-объекту (т.е. многоарной в общем случае связи). Причем, синтезировав суррогатные ключи (и их структуру) в каждом отношении, а также установив однозначное соответствие между именами сущностей-объектов и именами отношений, пользователь получает возможность в динамическом режиме пополнять каждое отношение атрибутами (т.е. свойствами) каждой сущности-объекта. При этом все необходимые документы ПрО моделируются как проекция фрагмента БД на те или иные носители — монитор, принтер, плоттер и т.п. Неопределенные сущности-объекты на момент внедрения приложения исключены.

В работе [7] также предложен алгоритм разделения сущностей-объектов моделируемой ПрО, но лишь на две категории: на сущности-объекты в статическом состоянии и на их связи. Однако такое упрощение приводит схему БД к жесткости и немодифицируемости [7].

Таким образом, задача автоматизированной сепарации сущностей-объектов не является тривиальной. И вопрос строгой формализации описанного ниже алгоритма — это вопрос дальнейших исследований. В настоящей работе приведено лишь его нестрогое описание.

Значительное число эмпирических исследований, проведенных на разных ПрО [2], позволяет высказать следующее допущение.

Гипотеза 1. Алгоритм последовательных приближений сепарации сущностей-объектов на реляционном каркасе имеет единственное решение. Такое допущение полностью согласуется с основными теоремами каркасной модели данных [3].

Отметим, что именно составные сущности-объекты в ПрО более всего замаскированы, и именно они обладают противоречивым происхождением смысла, поэтому их сепарация — это и есть основная цель каркасного проектирования схемы БД.

Уточненная классификация сущностей-объектов ПрО

Прежде всего согласно схеме реляционного каркаса введем формализацию ПрО. Пусть в описываемом алгоритме все сущности-объекты распределяются по пяти категориям. Дадим нестрогие (интуитивные) определения этим категориям, а также формализуем их.

1. *Атомарные* сущности-объекты — сущности-объекты, значения атрибутов которых не изменяются во времени, или этими изменениями в моделируемой ПрО можно пренебречь. Эти сущности-объекты не имеют также и зависимости существования [10]. В некоторых моделях данных такие сущности-объекты получили название стержневых или базовых [11].

Формально атомарная сущность-объект — семантически атомарный m -местный предикат [12] $L_j^M(A_j^m)$, где A_j^m — аргументы предиката — совокупность атрибутов j -й сущности-объекта, где $j = \overline{1, L}$ — номер сущности-объекта в ПрО, а $m = \overline{1, M_j}$ — номер места в M_j -местном предикате.

Как и в [12], семантическая атомарность — отсутствие внутренней семантической структуры предикатов, моделирующих атомарные сущности-объекты: никакие отдельные аргументы из подмножества аргументов не фигурируют ни в каких зависимостях как один 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 неконтролируемых (в дальнейшем «паразитных») зависимостей.

В [12] введено понятие ключ семантически атомарного предиката $\alpha_j(x_j)$ — такой одноместный предикат (в дальнейшем ключевой предикат), аргументом которого является минимально достаточный 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:

$$I_j^{M_j}(A_j^m) = \alpha_j(x_j) \cap F_j^{M_j}(A_j^m),$$

а вся N -совокупность P_I атомарных сущностей-объектов, так называемая статическая часть ПрО (по сути, структура системы P) — это предикат, полученный дизъюнкцией атомарных многоместных предикатов:

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

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.

2. «Слабые» сущности-объекты — сущности-объекты, которые функционально зависят от атомарных [10] и в моделях данных имеют аналогичное название, причем такая зависимость может быть как лишь на уровне идентификации слабых атрибутов, так и на уровне всего существования зависимых слабых сущностей-объектов.

Формально слабая сущность-объект — это семантически составной многоместный предикат, ключевой предикат которого получен конъюнкцией ключевых одноместных предикатов от «предковых» многоместных предикатов и ключевого одноместного предиката младшего по иерархии многоместного «предиката-наследника». При этом на любом уровне иерархии существует строгая функциональная зависимость (ФЗ) между каждым конъюнктивным «предковым» ключевым предикатом и последним в цепи ключевым «предикатом-наследником». Новый составной многоместный предикат также обладает своими собственными аргументами.

Схематически «слабый» многоместный предикат имеет вид

$$Q_i^{(k+j)}(x_1, x_2, \dots, x_k, b_1, b_2, \dots, b_j) = \alpha_1(x_1) \cap \alpha_2(x_2) \dots \alpha_k(x_k) \cap D_i^J(b_j),$$

где k — число звеньев иерархии ключа, $j=1, J_i$ — число мест для атрибутов i -го предиката. Имеется строгая ФЗ: $\alpha_k(x_k) \rightarrow (\alpha_{k-1}(x_{k-1}) \rightarrow (\dots \alpha_3(x_3) \rightarrow (\alpha_2(x_2) \rightarrow \alpha_1(x_1))))$.

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*, *КАФЕДРА*, каждая из этих сущностей-объектов не является са-

модостаточной, в ПрО функционально зависит от «предковой» сущности-объекта. Однако зависимостью ее атрибутов от времени в ПрО можно пренебречь.

Проектировщики часто делают исключение: некоторые слабые сущности-объекты принудительно назначаются как атомарные. Такой сущности-объекту искусственно назначается идентификатор, который уникально идентифицирует все атрибуты. Такие исключения — своеобразная граница ПрО, когда пользователю известно, что на протяжении значительного времени эксплуатации приложения и хранилища данных граница не будет расширяться. Однако именно такие исключения и приводят к невозможности осуществлять модификации схемы БД без изменений самого приложения как в процессе ее работы, так и после ее останова.

3. *Составные* («постсвязные») сущности-объекты в моделях данных имеют еще и название многосторонних связей [10, 13] — связь атомарных сущностей-объектов.

Формально составная сущность-объект — это семантически составной многоместный предикат, ключевой предикат которого получен конъюнкцией ключевых одноместных предикатов от связывающихся независимых атомарных сущностей-объектов (атомарных многоместных предикатов). Это означает, что наличие ФЗ в произвольном ключевом звене не является обязательным условием существования такого предиката. Составной многоместный предикат также обладает своими аргументами — аргументами связи.

Схема составного многоместного предиката имеет вид, аналогичный схеме из предыдущего пункта:

$$S_i^{(k+j)}(x_1, x_2, \dots, x_k, b_1, b_2, \dots, b_j) = \alpha_1(x_1) \cap \alpha_2(x_2) \cap \alpha_3(x_3) \dots \alpha_k(x_k) \cap T_i^J(b_j),$$

где k — число звеньев ключа, $i = \overline{1, J_i}$ — число мест для атрибутов i -го предиката. Строгая ФЗ вида $\alpha_k(x_k) \rightarrow (\alpha_{k-1}(x_{k-1}) \rightarrow (\dots \alpha_3(x_3) \rightarrow (\alpha_2(x_2) \rightarrow \alpha_1(x_1))))$ в общем случае отсутствует.

Одним из примеров 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.

В [14] описаны объекты, полученные агрегацией данных. В ней сущность-объект *ЭКЗАМЕН* так же, как и в [2], отнесена к категории агрегированных сущностей-объектов, т.е. составных: «Например, связь между сущностями *СТУДЕНТ*, *ДИСЦИПЛИНА*, *ПРЕПОДАВАТЕЛЬ*, *ОЦЕНКА* имеет смысловое описание: «студент по фамилии_получил на экзамене по дисциплине_у преподавателя по фамилии_оценку» и может быть представлена агрегированным элементом: сущностью *ЭКЗАМЕН* с атрибутами *ФИО_СТУДЕНТА*, *НАЗВАНИЕ_ДИСЦИПЛИНЫ*, *ФИО_ПРЕПОДАВАТЕЛЯ*, *КОД_ОЦЕНКИ*» [14].

Агрегированное понятие вводится пользователем в языковой обиход и делопроизводство для того, чтобы в процессе функционирования ПрО экономить ресурсы некомпьютерной обработки информационных потоков: документооборота, устных сообщений, экспертной аналитики и т.п. Очевидно, что агрегация сущностей-объектов как методология естественного языка направлена на упрощение восприятия. Однако такое упрощение приводит к избыточности языка и неоднозначности толкования терминов-заменителей. Поэтому при проектировании высокоформализованных схем БД все агрегированные термины ПрО, на которые так или иначе вынужденно ссылаются атомарные и слабые сущности-объекты, должны быть декомпозированы. Очевидно, что если таких ссылок нет, то данный агрегированный термин обозначает несвязанную с функционированием ПрО сущность-объект (тривиальный случай).

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) зависимые от атомарных, но статичные во времени. Отношения, моделирующие слабые сущности-объекты, также имеют составной ключ. На совокупности атомарных и слабых сущностей-объектов благодаря образованию разнообразных связей между ними создаются составные постсвязные сущности-объекты, характеризующие поведение ПрО. Однако в настоящей работе темпоральные атрибуты не фигурируют, а описываются лишь статические срезы ПрО. В произвольной ПрО образование слабых и составных сущностей-объектов, как правило, маскируют части речи: существительные, отглагольные существительные, термины (которые им соответствуют), категории (которые их обобщают) и т.п. Именно это делает автоматизацию описанного алгоритма актуальной.

Чаще всего подавляющее большинство составных сущностей-объектов проектировщики 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 лишь как связь его атрибутов, т.е. как тип тривиальной связи «самого с собой»:

$I_j^M(A_j^m)$ — тривиальная связь, а нетривиальные связи рассматриваются как конъюнкция ключевых атрибутов от разных семантически атомарных многоместных предикатов. Тогда схематически фрагмент булеана связей имеет вид (с несколько более усложненными индексами)

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

где b_l^j — атрибуты связи, моделируемые $S_i^{(l_j+j)}$ -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 $R_j(X_1, X_2, X_3, \dots, X_j, B_1^j, B_2^j, \dots, B_l^j)$.

Тогда схематически в общем виде модель ПрО можно представить так:

$P_I = I_1^{M_1}(A_1^{m_1}) \cup I_2^{M_2}(A_2^{m_2}) \dots \cup I_N^{M_n}(A_N^{m_n})$ — 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.

Тогда $P = P_I \cup P_S$, т.е. ПрО как многоместный предикат P есть дизъюнкция конъюнкций многоместных предикатов. С одной стороны, это и есть актуальная часть каркаса, а с другой — это классическая нормальная форма высказывания в булевой логике, широко используемая в исследованиях в смежных областях, в частности в исследованиях проблем искусственного интеллекта [15].

Описанные типы сущностей-объектов встречались и в других концептуальных моделях. Однако, как показывает практика проектирования приложений, их недостаточно. Для более точного моделирования ПрО необходимо использование еще двух типов — артефакты и неопределенные сущности-объекты.

4. *Артефакты* — это сущности-копии, данные о которых условно размещаются в БД по решению пользователя.

Формально артефакт (документ) — это предикат $E_i(A_j, B_l, C_k, D_m, \dots, Z_p)$, где каждый из аргументов $A_j, B_l, C_k, D_m, \dots, Z_p$ — это фрагмент множества аргументов «порождающих» предикатов, участвующих в формировании предиката E_i , т.е. это частичная копия многоместного предиката, моделирующего связь других сущностей-объектов, поэтому артефакт имеет некоторое сходство с составной сущностью-объектом.

К артефактам, например, можно отнести любой документ, который пользователи ПрО создают именно ради того, чтобы скопировать те или иные атрибуты тех или иных сущностей-объектов. Причем не просто скопировать атрибуты одной конкретной сущности-объекта, а еще и объединить в этой новой искусственно созданной сущности-объекте несколько атрибутов от разных сущностей-объектов.

Артефакты — это, как правило «постследственные» сущности-объекты. Поэтому, независимо регистрируя их в приложении, пользователь сталкивается со значительным дублированием данных. А это, в свою очередь, приводит к необходимости дополнительного отслеживания целостности еще и избыточных данных. Проблема заключается в том, что в ПрО вероятнее всего уже существует группа неискусственных составных сущностей-объектов, у которых имеется еще и множество атрибутов, которые не копируются в атрибуты совокупности искусственных сущностей-объектов.

Отслеживание целостности таких дублированных данных в этой совокупности естественных и искусственных сущностей-объектов очень затруднено. Каркасная модель данных позволяет исключить такие сущности-копии из схемы БД, а копии моделировать посредством проекций частей схемы БД при ответах на запросы. Однако если в ПрО отсутствуют дублирующие сущности-объекты, проектировщик может использовать составную сущность-объект под именем артефакта.

Примером артефактов могут быть *НАКЛАДНАЯ, СЧЕТ, АКТ* и т.д.

5. *Неопределенные* сущности-объекты — такие, семантика которых подлежит дальнейшему уточнению; или такие, на которые не ссылается ни одна из всей совокупности сущностей-объектов моделируемой ПрО.

Формально к неопределенным сущностям-объектам первого типа отнесем предикат $E_i(A_j, B_l, C_k, D_m, \dots, Z_p)$, совокупность $A_j, B_l, C_k, D_m, \dots, Z_p$ аргументов которого — пустое множество. К неопределенным сущностям-объектам второго типа отнесем предикат $E_i(A_j, B_l, C_k, D_m, \dots, Z_p)$, никакая часть совокупности $A_j, B_l, C_k, D_m, \dots, Z_p$ аргументов которого не встречается ни в одной связи других сущностей-объектов этой ПрО.

Неопределенные сущности-объекты возникают в описании ПрО на этапе незавершенного исследования функционирования ПрО.

Отметим, что впервые такая классификация, а также алгоритм сепарации, основанный на ней, были предложены в [16].

Предпочтения, типичные для проектировщиков

В работе [4] Р. Фейджин показал, что в реляционной модели данных (РМД) существует безаномальная схема БД, которую Д. Кренке в [17] назвал «ориентиром для проектировщика». Но в [3] доказано, что при определенных, хотя и достаточно распространенных в ПрО ограничениях на домены и ключи, каркасная совокупность схем БД удовлетворяет критериям доменно-ключевой нормальной формы (ДКНФ) [4] и тем самым является безаномальной. А поскольку, как доказано в [12], реляционный каркас — это единственная в РМД совокупность схем БД, которая обладает еще и непротиворечивой модифицируемостью, т.е. обеспечивает прило-

жению динамическую «расширяемость» [7], можно утверждать, что только каркасная совокупность решает «дилемму Кодда» [18]: позволяет получить оптимальную схему БД по этим двум важным критериям. На основании этого утверждения можно сделать вывод, что любая некаркасная совокупность реляционных схем БД будет неоптимальной, т.е. по сути аномальной, иными словами — ошибочной.

В таком контексте проанализируем типичные ошибки проектирования схем БД: ошибочную сепарацию сущностей-объектов.

1. Слабые сущности-объекты ПрО («структурные подразделения») моделируются как атомарные. Для моделирования слабых сущностей-объектов, таких как *КВАРТИРА*, *ЛАБОРАТОРИЯ*, *ОТДЕЛ*, *КАФЕДРА* и т.п. используется схема $R(ID, A_j, B_l)$, в которой ID — унарный идентификатор, A_j — собственные атрибуты, модель свойств непосредственно данной сущности-объекта, B_l — внешние атрибуты (следы внешних сущностей-объектов [1]), идентифицирующие связь данной сущности-объекта в ПрО.

Такой подход в схеме отношений может вызвать аномалии. Как указывалось выше, слабая сущность-объект не может быть идентифицирована ни названием, ни набором своих собственных атрибутов [10]. Поэтому проектировщик для ее идентификации (по сути, для работы приложения) использует взаимосвязь между сущностями-объектами в ПрО: в совокупность атрибутов отношения $R(ID, A_j)$ добавляются атрибуты «предковых» (более старших по иерархии) сущностей-объектов B_l . Но тогда в отношении $R(ID, A_j, B_l)$ появится естественный ключ. При этом ФЗ неключевых атрибутов от суррогатного ключа (унарного ID) не являются следствием ФЗ естественного ключа. Формально неключевые атрибуты становятся транзитивно зависимыми от одного из ключей. Такая комбинация ФЗ приводит схему к форме, ниже ДКНФ, а значит, к аномалиям — к потенциальным ошибкам.

Для проектирования безаномальной схемы БД необходимо в описании ПрО отслеживать наличие всех звеньев иерархии данной слабой сущности-объекта, т.е. всех «предковых» сущностей-объектов, которые позволяют однозначно идентифицировать атрибуты слабой.

2. События моделируются как атомарные или слабые сущности-объекты. Для моделирования составных сущностей-объектов, таких как *МИТИНГ*, *КОНЦЕРТ*, *ЭКЗАМЕН* и т.п., используется схема $R(ID, A_j, B_l)$, в которой ID — унарный идентификатор, A_j — собственные атрибуты — модель свойств непосредственно данной связи (составной сущности-объекта), B_l — внешние атрибуты (следы внешних сущностей-объектов [1]), идентифицирующие связь данной сущности-объекта в ПрО.

Формально проектировщик для идентификации взаимосвязи между атомарными или слабыми сущностями-объектами в ПрО (для работы приложения) вынужден в совокупность атрибутов отношения $R(ID, A_j)$ добавить атрибуты «порождающих» (участвующих в связи) сущностей-объектов B_l . Но тогда в отношении $R(ID, A_j, B_l)$ появятся естественные ключи, зависимости от которых не являются следствием зависимостей от унарного ID . Тогда в отношении появятся «паразитные» зависимости типа декартовой зависимости (ДЗ) [3] — многозначные зависимости (МЗ), зависимости проекции-соединения или транзитивные ФЗ. Это приведет схему к форме, ниже 4НФ [4], а значит, к аномалиям — к ошибкам.

Если проектировщик не декомпозирует составную (агрегированную) сущность-связь, возникнет ситуация, описанная в пункте выше: схема отношения, моделирующего составную сущность-связь как атомарную, будет аномальной.

3. Отношения, моделирующие атомарные или слабые сущности-объекты, в своей схеме имеют атрибуты иных атомарных или слабых сущностей-объектов. Формально проектировщик для идентификации взаимосвязи между атомарными или слабыми сущностями-объектами в совокупность атрибутов отношения, моделирующего любую из связываемых сущностей-объектов $R(ID, A_j)$, добавляет атрибуты другой связываемой сущности-объекта B_l . Как и в предыдущих ситуациях, в отношении $R(ID, A_j, B_l)$ появятся естественные ключи, которые не являются следствием ключа — унарного ID . В отношении появятся «паразитные» зависимости типа ДЗ [3] или транзитивные ФЗ. Это приведет схему к форме, ниже 4НФ [4], а значит, к аномалиям — к ошибкам.

4. Документы моделируются как равноправные в ПрО сущности-объекты. Формально проектировщик (аналогично предыдущим пунктам) моделирует тот или иной артефакт (документ) из ПрО отношением $R(ID, A_j, B_l, C_k, D_m, \dots, Z_p)$, где $A_j, B_l, C_k, D_m, \dots, Z_p$ — фрагменты атрибутов «порождающих», т.е. участвующих в формировании артефакта, сущностей-объектов, а сам артефакт, как правило, является результатом связи других сущностей-объектов. Тогда, если в отношении $R(ID, A_j, B_l, C_k, D_m, \dots, Z_p)$ появятся новые ключи, которые не являются следствием унарного ID , схема станет аномальной. Но даже если дополнительные ключи в схеме отношения $R(ID, A_j, B_l, C_k, D_m, \dots, Z_p)$ не возникнут, и схема будет удовлетворять единственному ограничению $ID \rightarrow (A_j, B_l, C_k, D_m, \dots, Z_p)$ (является следствием ключа — схема будет в ДКНФ), между атрибутами разных отношений схемы БД могут возникнуть «паразитные» ФЗ. Это приведет к необходимости разработки дополнительных триггеров по отслеживанию целостности, а значит, также повысит жесткость (немодифицируемость) приложения.

5. Неопределенные сущности-объекты моделируются как атрибуты других сущностей-объектов. Формально проектировщик моделирует неопределенную сущность-объект посредством ее связи с другими сущностями-объектами из ПрО. Для этого он добавляет дополнительные атрибуты в отношение $R(ID, A_j, B_l, C_k, D_m, \dots, Z_p)$. Здесь B_l , например, — атрибут (или группа атрибутов), имя которого формально совпадает с именем сущности-объекта в ПрО. Однако эта сущность-объект не исследована проектировщиком и не отнесена ни к одной из категорий. Схема такого отношения при тех или иных условиях может стать аномальной.

Из этого следует, что встречающийся в большинстве проектов схем промышленных БД прием идентификации всех сущностей-объектов и их связей посредством лишь суррогатного унарного ID , независимо от их семантики, может вызывать значительные аномалии. В работе [19] дано строгое доказательство того факта, что наличие двух разных ключей в одном отношении, к чему приводят «взаимные» [19] ФЗ, является аномалией. Это совпадает с основным выводом Фейджина [4] о том, что у безаномального отношения не должно существовать ФЗ, не являющихся следствием ключа. И хотя вывод о том, что в отношении, где существует два независимых ключа и больше, каждая ключевая ФЗ не является следствием «соседнего» ключа, в работе [4] отсутствует, нетрудно доказать, что такое отношение имеет транзитивную зависимость.

Лемма 1 (обозначения соответствуют [3, 4]). Если отношение $R(X, Y, A)$ имеет ограничения

$$DD(R) = IN(X, \forall X(X \neq Y \neq A \neq \text{const})) \cap IN(Y, \forall Y(Y \neq \text{const})) \cap IN(A, \forall A(A \neq \text{const})),$$

$$KD(R) = KEY(X) \cap KEY(Y),$$

то в нем существует транзитивная зависимость $X \rightarrow Y \rightarrow A$.

Доказательство. Из ограничений на ключи следует, что $X \rightarrow (Y + A)$ и $Y \rightarrow (X + A)$. Вследствие одной из аксиом Армстронга $X \rightarrow A$, но тогда и $X \rightarrow Y \rightarrow A$. \square

Тот же факт, что такая зависимость взаимна, т.е. $Y \rightarrow X \rightarrow A$, не исключает аномалии. Поэтому данное отношение должно быть декомпозировано на два отношения: $R_1(X, Y)$ и $R_2(X, A)$ или $R_1(X, Y)$ и $R_2(Y, A)$.

Такая реструктуризация отношения полностью совпадает с основным посылом нормализации Кодда [18]: если декомпозиция не приводит к потере значимых для ПрО зависимостей, все отношения, которые могут быть декомпозированы, должны быть декомпозированы. В противном случае могут возникать неконтролируемые приложением зависимости. Если внесенный в отношение дополнительный суррогатный ключевой атрибут не обусловлен естественной ФЗ, такая операция приводит к денормализации отношения [20]. Этот же признак совпадает с основным признаком актуальных ячеек реляционного каркаса [3], в каждой из которых существует единственный ключ.

Текстовое описание ПрО

В настоящей работе подразумевается, что описание моделируемой ПрО должно быть выражено типизированной формой на естественном языке. Единицей считывания при этом является атомарное предложение (в дальнейшем просто предложение), содержащее не больше двух сущностей-объектов, которые кодируются существительными с уникальным побуквенным написанием. А глагол с уникальным побуквенным написанием символизирует исключительно бинарную связь между парой сущностей-объектов этого же предложения.

Предполагается, что существительные, которые повторяются, обозначают одну и ту же сущность-объект. Поэтому такое повторение в пределах одного предложения означает тривиальную пару, т.е. такую, которая несет лишь информацию о существовании сущности-объекта в ПрО без связей ее с другими. Предполагается также, что и глаголы, которые повторяются в разных предложениях, означают один и тот же класс связи. Поэтому основная миссия атомарного предложения — информировать о наличии сущностей-объектов в определенной ПрО и декларировать класс связи этой пары.

Предложения, которые включают в себя больше двух сущностей-объектов, являются составными. Они подлежат автоматизированной декомпозиции. Для этого может использоваться любой известный алгоритм. Однако те составные предложения, которые невозможно автоматизировано привести к бинарной форме по технологической причине (например, из-за отсутствия четкой структуры, которая объединяет их в одно составное предложение), из описания изымаются. Этот фрагмент описания подлежит дальнейшему уточнению.

Алгоритм не предусматривает верхнего ограничения количества предложений. Тем не менее предполагается предварительный анализ наличия для каждой задекларированной сущности-объекта хотя бы одной связи с какой-либо иной сущностью-объектом.

Таким образом, проектировщик схемы БД может получить описание ПрО в виде:

- текстового файла, сформированного текстом на естественном языке;
- последовательных схем или графов;
- файла с записанным голосовым сигналом, который надиктован естественным языком;
- звукового голосового сигнала в реальном времени;
- последовательности файлов хранилищ данных, которые уже существуют и введены в эксплуатацию и т.п.

Файлы внедренных БД особенно важны для исследования возможных противоречий в схемах БД и прогнозирования затрат на модификации при дальнейшем развитии приложения. Очевидно, что каждый тип описания может использоваться не обособлено, а в сравнении один с другим.

Этимология смысла сущности-объекта

Для автоматизированного синтеза схемы БД в каждом проектируемом отношении $R_i(X_{k_i}^{m_k}, A_j)$ формируются структурированные идентификаторы, структура каждого из них произвольная, не задана пользователем и не получена каким-то иным отличным способом, а строго исчисляется.

Для автоматизированного исчисления структуры идентификатора используются математические критерии, построенные в соответствии с закономерностями, выявленными в ПрО. В основе этих критериев единый обобщенный фактор — происхождение смысла сущности-объекта, т.е. этимология смысла (в дальнейшем просто этимология).

Под этимологией сущности-объекта понимается строковый структурированный идентификатор со следующей общей схемой:

$$X_1^{m_1} + X_2^{m_2} + X_3^{m_3} + \dots + X_{k_i}^{m_k},$$

где каждое звено $X_{k_i}^{m_k}$ — отделенный идентификатор факта происхождения i -й сущности-объекта, k_i — номер звена идентификатора i -й сущности-объекта (подстрочный индекс), m_k — номер соответствующей порождающей сущности-объекта из базовой совокупности сущностей-объектов — объединенной группы атомарных и слабых сущностей-объектов (надстрочный индекс), причем каждое m_k может получить какое либо значение только из множества $\{1, 2, \dots, N_0, \dots, N\}$, где N_0 — общее количество атомарных сущностей-объектов, N — суммарное количество атомарных и слабых сущностей, i — номер произвольной сущности-объекта в ПрО. А в случае полной совокупности связей $i = \{1, 2, \dots, N_0, \dots, N, (N+1), \dots, (2^N - 1)\}$. Знак «плюс» в общем виде схемы этимологии означает строковое объединение (конкатенацию). Для атомарных сущностей этимологией есть лишь одно звено X^i , в котором $m = i$, т.е. атомарная сущность-объект порождает сама себя.

Гипотеза 2. Все факторы, характеризующие семантику любой сущности-объекта в ПрО, функционально зависимы от этимологии сущности-объекта.

В описываемом алгоритме атомарные сущности-объекты получают в общей совокупности первые номера, т.е. для них $i = 1, N_0$. Для слабых сущностей этимологией является вышеупомянутая строковая сумма звеньев, где каждому номеру k_i звено $X_{k_i}^{m_k}$ соответствует строго.

Для составных сущностей-объектов этимологией является вышеупомянутая строковая сумма звеньев, где место каждого звена $X_{k_i}^{m_k}$ нестрогое, т.е. последовательность звеньев не имеет значения.

Каждое звено этимологии сущности-объекта означает связь с иными атомарными сущностями-объектами, которые принимали участие в происхождении данной сущности-объекта, если последняя представляет собой либо слабую, либо составную, т.е. постсвязную сущность-объект.

Таким образом, каждое звено $X_{k_i}^{m_k}$ идентификатора ячейки строится в строгом соответствии с происхождением сущностей-объектов из описания ПрО. Каждая сущность-объект в ПрО может быть атомарной, а значит, иметь унарный X^i

или составной идентификатор $\sum X_{k_i}^{m_k}$, где суммирование ведется по k_i , $k_i = \overline{1, K_i}$, причем общее число звеньев K_i представляет собой арность отношения, моделирующего такую сущность-объект. При этом K_i в общем случае может равняться 2, 3, ..., 10 и т.д., а в случае атомарной сущности-объекта $K_i = 1$.

Логическая сепарация

Начальное приближение алгоритма синтеза каркаса-шаблона можно получить с помощью специализированного словаря вероятных этимологий. Анализ основан на сравнении имени и контекста (совокупности сущностей-объектов в описании ПрО) каждой сущности-объекта с задекларированными в этом словаре сущностями-отношениями.

В словаре каждому существительному и глаголу соответствует наиболее вероятная этимология. Причем те слова, которые обозначают пока что неизвестные словарю сущности-объекты и классы связей, для дальнейшего автоматизированного анализа отделяются, а если неизвестные сущности-объекты и связи в описании не выявлены, автоматизированный логический анализ завершается.

Формально пусть имеется множество сущностей-объектов в предикатной форме $E_l(A_j, B_l, C_k, D_m, \dots, Z_p)$. Для каждой l -й сущности-объекта нужно определить, к какому типу она относится с учетом всей совокупности E_l , а после этого определить, какие атрибуты действительно принадлежат E_l , какие являются внешними и символизируют связь, а какие попали в совокупность по ошибке.

На дальнейшем шаге из описания отделяются неизвестные словарю потенциальные составные сущности-объекты благодаря автоматизированному логическому сравнению каждой из неизвестных сущностей-объектов с теми, что образуются из повторяющихся существительных и повторяющихся глаголов благодаря объединению их в одну составную, т.е. многостороннюю постсвязную сущность-объект.

Такое объединение возможно при условии совпадения класса связи, т.е. совпадения глаголов между различными парами, так как именно благодаря многократной повторяемости упомянутых существительных в нескольких разных связях из одного класса, т.е. для нескольких одинаковых глаголов вероятность того, что эти сущности-объекты принадлежат именно к группе составных сущностей-объектов, значительно повышается.

Если же выяснится, что такое приближение ошибочное, это не внесет значительную некорректность. Схема будет уточняться.

На последующих шагах осуществляется последовательное или параллельное выполнение процедуры сравнения каждой сущности-объекта из каждого предложения, т.е. из каждой пары, с каждой другой сущностью-объектом.

Результат — это логическая сепарация, т.е. предоставление каждому отношению-проекту, где будут храниться данные от атрибутов каждой сущности-объекта моделируемой ПрО, соответствующих предварительных структурированных идентификаторов, и тем самым предварительная перегруппировка этих отношений по категориям.

Особенностью данного этапа является то, что синтез структуры каждого звена этимологии сущностей-объектов осуществляется автоматизированным логическим анализом существительных и глаголов, т.е. анализом содержания сущностей-объектов и содержания связей без учета множеств конкретных значений конкретных атрибутов сущностей-объектов.

В этой части алгоритма завершается автоматизированный логический анализ описания ПрО. Последнее логическое сравнение — анализ группы тех сущностей-объектов и связей, которые оказались неизвестными словарю вероятных этимологий и остались после изъятия потенциально составных сущностей-объектов.

Статистическая сепарация

На следующем шаге от предварительно отобранных групп сущностей-объектов окончательно отделяются артефакты, т.е. сущности-копии, которые могли по ошибке остаться в этой совокупности. Для этого осуществляется автоматизированное статистическое сравнение, основанное на использовании известных [21] процедур статистического анализа для выявления детерминированных ФЗ, а также корреляционных или регрессивных МЗ между значениями данных в атрибутах сущностей-объектов. Наличие или отсутствие таких зависимостей позволяет подтвердить или опровергнуть прямые совпадения групп атрибутов, а также замаскированную этимологию и семантическую структуру, полученную на предыдущих шагах.

Для отслеживания наличия, например, прямых совпадений атрибутов-копий (в случае несовпадения имен) достаточно сравнить не больше десяти групп значений, т.е. не больше десяти групп кортежей для реляционного формата хранения значений атрибутов сущностей-объектов.

Для отслеживания закономерности на этом шаге алгоритма от каждой сущности-объекта достаточно не более двух естественных атрибутов. А для отслеживания, например, МЗ, которая наблюдается лишь между атрибутами составных сущностей-объектов и отдельно атрибутами каждого из их предков, которые принимали участие в образующих связях этих составных сущностей-объектов, достаточно сравнить не более двухсот групп значений, т.е. не более двухсот групп кортежей для реляционного формата хранения значений атрибутов сущностей-объектов. Причем между каждым конкатенированным значением экземпляров общей совокупности всех отделенных атрибутов предков и значениями экземпляров какого-либо (или даже каждого) из атрибутов составных сущностей-объектов возникает уже не МЗ, а детерминированная ФЗ, если именно эти предки образовывали именно эту составную сущность-объект. Наличие такой детерминированной связи является достаточным критерием для идентификации и сепарации составных сущностей-объектов. Причем для отслеживания этой закономерности от каждой сущности-объекта также достаточно не больше двух естественных атрибутов.

Уточненная сепарация

Для корректности статистического анализа вся совокупность значений всех атрибутов от всех сущностей-объектов ПрО должна отвечать единому моменту времени жизни ПрО. Причем отрезок между соседними промежутками времени должен быть достаточным для возникновения действительно нового состояния ПрО. Если это условие не выполняется, закономерности будут некорректными.

На следующем шаге строится уточненное приближение сепарации, для этого отделяются группы значений атрибутов, которые зависят от времени, и группы значений атрибутов, которые от времени не зависят, а если и зависят, то лишь от очень значительных промежутков времени — их развитием и изменениями можно пренебречь в сравнении с другими группами значений атрибутов. Причем группа атрибутов, которая практически не зависит от времени, относится к группе сущностей-объектов, которые создают структуру ПрО. Структура какой-либо системы значительно медленнее зависит от времени, чем ее функционирование, т.е. формирование связей между сущностями-объектами. Таким образом, на этом шаге за уточненное приближение составных сущностей-объектов берется группа сущностей-объектов, которые зависят от времени, а другая группа получает статус совокупности атомарных и слабых.

На следующем шаге в группе, где отобраны атомарные и слабые сущности-объекты, повторно и более доказательно автоматизировано отделяются атомарные от слабых. Одновременно используется два критерия.

Первый критерий заключается в том, что для идентификации какого-либо значения естественного атрибута атомарной сущности-объекта достаточно лишь названия сущности-объекта и названия атрибута, что невозможно именно в случае слабой сущности. Но такое сопоставление на этом шаге осуществляется на значительно большем количестве данных.

Второй критерий заключается в том, что между атрибутами потомка и конкатенированными атрибутами всех предков наблюдается ФЗ, а поэтому обнаруживается детерминированная связь, которая дает возможность отслеживать не только сам факт слабости, а еще и конкретизировать звенья связей с более «старшими» сущностями-объектами. При этом, если связь от потомка к предку устанавливается однозначно, проверка наличия или отсутствия однозначной обратной связи от предка к множеству потомков возможна лишь благодаря интерполяции значений от атрибутов всех потомков следующего уровня, т.е. преобразования множества этих значений в математическую функцию и проверки детерминированной зависимости на отрезке в окрестности значений атрибута конкретного потомка (аналогично отслеживанию детерминированной связи, например, у периодической функции). Подтвержденная связь отображена в структуре идентификатора ячейки сущности-объекта.

Для дальнейшего уточнения не только характера и принадлежности к группе составных сущностей-объектов, а еще и окончательного восстановления структуры этимологии каждой составной сущности-объекта на базе полученной совокупности атомарных и слабых сущностей-объектов в компьютерной памяти в качестве шаблона [16] строится каркас полной совокупности связей данных.

Результирующий алгоритм последовательных приближений

В общем виде алгоритм автоматизированного каркасного проектирования схемы БД сводится к следующим шагам метода последовательных приближений, когда на каждом следующем шаге благодаря определенным логическим и математическим критериям уточняется каждая предыдущая совокупность данных.

Шаг 1. Формирование взаимно-однозначного соответствия между каждым словом из описания ПрО, обозначающим сущность-объект, и группой слов, каждое из которых обозначает атрибут каждой сущности-объекта из ПрО. Эта часть алгоритма проводится пользователем вручную. На этом же этапе пользователь исключает артефакты и неопределенные сущности-объекты. Данный шаг играет роль ручной подготовки исходных данных. Результат — список вида $E_l(A_j, B_l, \dots, Z_p)$, где E_l — совокупность многоместных предикатов, соответствующих списку слов, обозначающих сущности-объекты из ПрО, каждое из которых встречается единственный раз. A_j, B_l, \dots, Z_p — совокупность слов, обозначающих атрибуты каждой сущности-объекта, $l = \overline{1, L}$ — число всех сущностей-объектов, $i = \overline{1, I_l}$; $j = \overline{1, J_l}$; $p = \overline{1, P_l}$ — значения текущего числа атрибутов каждой сущности-объекта.

Шаг 2. Автоматическое или автоматизированное изъятие начального приближения базовой совокупности сущностей-объектов. В описании ПрО эта группа сущностей-объектов, как правило, замаскирована разнообразными агрегированными терминами, категориями, вспомогательными существительными, синонимами и т.д.

Для этого к отобранной группе атомарных сущностей-объектов присоединяется еще и подгруппа виртуально атомарных сущностей-объектов, которая получается добавлением к идентификаторам слабых сущностей-объектов отдельного унарного идентификатора. Создается начальное множество простых унарных идентификаторов.

Это действие носит сугубо технологический характер и упрощает дальнейшие шаги относительно создания комбинаций идентификаторов ячеек: назначенные виртуально атомарные сущности-объекты, которые происходят от слабых, несут в себе обе этимологии — естественную, т.е. составную, и искусственную, т.е. унарную. Это действие коренным образом отличается от процедуры автоматического назначения унарного идентификатора без учета семантики каждому объекту, что свойственно, например, объектно-ориентированной модели [22].

Шаг 3. Последовательно или параллельно сравниваются слова из описания моделируемой ПрО, обозначающие сущности-объекты и их связи, с проанализированными в других проектах. На этом шаге используется пополняемый словарь. Здесь же базовая совокупность окончательно отделяется от случайно неотобранных пользователем артефактов и неопределенных сущностей-объектов.

Шаг 4. Если словарь для анализируемых сущностей-объектов или связей пуст, осуществляется процедура автоматизированного логического сравнения сущностей-объектов между собой. Число логических процедур и критериев для сравнений ничем не ограничено — они отделены во внешнюю библиотеку, которая пополняется.

Шаг 5. Для каждого унарного идентификатора каждой сущности-объекта из базовой совокупности в хранилище отводится одинарный домен памяти для размещения элементов хранения идентификатора, структура которого строго унарная.

Таким образом, в памяти создается начальное множество простых одинарных доменов. При этом идентификаторы от слабых сущностей-объектов могут быть обозначенными дополнительно. Тем не менее способ установки подобных меток может быть произвольным, вплоть до их отсутствия.

Шаг 6. На базовой совокупности одинарных доменов осуществляется синтез каркаса-шаблона — каркасных эталонных составных сущностей-объектов — построение булеана связей по принципу «все со всеми». Этой процедурой порождается система доменов с многоарными идентификаторами.

Шаг 7. Наполнение таблиц шаблона конкретными данными из ПрО. Группы данных отбираются и вносятся таким образом, чтобы каждая группа строго соответствовала новому состоянию ПрО.

Шаг 8. Начальная сепарация составных сущностей-объектов благодаря процедурам статистического сравнения эталонных составных сущностей-объектов, полученных на каркасе-шаблоне, и тех составных сущностей-объектов, которые на завершающем этапе отделены.

Благодаря процедурам статистического анализа с использованием конкретных значений данных осуществляется проверка групп атрибутов атомарных, составных и слабых сущностей-объектов из описания ПрО, а также сформированных атомарных и составных идентификаторов на соответствие один другому.

Шаг 9. Уточнение сепарации составных сущностей-объектов посредством процедуры последовательных приближений на каркасе-шаблоне.

Уточняется соответствие путем применения повторной процедуры последовательных приближений и многократной модификации базовой совокупности, т.е. соответствующего каркаса-шаблона.

Шаг 10. Рекомендации администрации словаря о возможности пополнения его ресурсов новыми группами сущностей-объектов, если в окончательных группах никаких противоречий не выявлено.

В конечном итоге алгоритм приведет к полному совпадению этимологии всех сущностей-объектов из начального потока с этимологией искусственно синтезированных на каркасе. Очевидно, что в зависимости от специфики конкретной ПрО некоторые шаги алгоритма могут не использоваться.

Библиотека критериев сепарации

Алгоритм предусматривает возможность развития процедур логического и статистического анализа. Для этого отдельно строится внешняя библиотека, которая пополняется новыми подчиненными способами как логического, так и статистического анализа со своими новыми критериями, которые разрабатываются пользователями. Поэтому перечень подчиненных способов сравнения данных между собой, а также перечень критериев сравнения ничем не ограничивается.

Не ограничивается также и последовательность выполнения упомянутых процедур. Очевидно, что наиболее точная сепарация может быть проведена либо благодаря словарю вероятных этимологий, либо благодаря автоматизированному статистическому анализу на каркасе-шаблоне.

Первый тип сепарации еще и самый быстрый, последний самый длительный. Поэтому при отсутствии сущностей-объектов в словаре выполнение всех иных, т.е. промежуточных итераций, значительно ускоряет каркасную сепарацию и позволяет всесторонне проанализировать данные. Если словарь вероятных этимологий на начальных стадиях своего существования не является полным, постоянная эксплуатация, которая пополняет его, в конечном итоге минимизирует потребность в автоматизированном логическом и статистическом анализе описания ПрО.

Для каркасной модели доказаны теоремы о полноте и единственности реляционного каркаса, а также о его непротиворечивом росте [12]. Важным следствием этих теорем является следующая лемма.

Лемма 2. Составные сущности-объекты со схемой $R(X_i, A_j)$ между собой дальнейших связей не образуют и сущностей-объектов не порождают.

Действительно, если для какой-либо совокупности составных сущностей-объектов $R(X_i, A_j)$ искусственно назначить статус атомарных с искусственными унарным идентификатором $Y = X_i$, а затем вновь перемножить их, то образованные новые (искусственные) составные сущности-объекты (по сути связи связей) можно получить и на продолжении каркаса $R(X_i, X_i + 1, B_j)$, но при условии, что при новом перемножении продублированные идентификаторы из отношений исключаются, что соответствует реляционной модели (и здравому смыслу), отношения станут тождественными. □

Это означает, что и без переобозначения идентификаторов базовая совокупность сущностей-объектов — это еще и базовая совокупность идентификаторов. При таком ограничении синтезированные составные сущности-объекты не расширяют базовой совокупности. Тем не менее какое-либо расширение базовой совокупности сущностей-объектов приводит к появлению новых составных сущностей-объектов. Поэтому если все же возникает такая потребность, алгоритм позволяет искусственно моделировать дальнейшие связи именно расширением базовой совокупности идентификаторов, например, прибавляя к начальной совокупности еще и искусственные атомарные сущности-объекты, полученные из составных путем установки в их структуре искусственных унарных идентификаторов.

Такая ситуация может возникнуть при условии, что для некоторых ПрО характерно расширение их структуры за счет синтезированных составных сущностей-объектов. В этой ситуации важно обязательное многократное добавление идентификаторов, отвечающих за различные состояния составных сущностей-объектов или их масок, а также учет номеров отрезков времени таких модификаций в этих идентификаторах, что в рамках настоящей работы не рассматривается.

Этот механизм позволит вносить изменения в схему такой БД по полномодифицируемому принципу, а не с существенным редизайном как самой схемы БД, так и приложения.

Результаты экспериментальных исследований

В настоящей работе проведен эксперимент, основной характеристикой которого является время доступа к любому атрибуту любого отношения исследуемой БД. Как известно, самой ресурсоемкой операцией реляционной алгебры являются соединения отношений [10]. Эта операция применяется при выполнении запросов к БД низкой «нормальности» — не выше 3НФ [11]. Отличительной особенностью именно ДКНФ является то, что в БД большинство исследованных связей моделируется уже соединенными отношениями. Поэтому отсутствие операции соединения значительно экономит время доступа к БД.

Для экспериментального исследования каркасного метода проектирования схемы БД и проведения указанного численного эксперимента была выбрана ПрО: «Учет отгрузки угля на шахте». Описываемое приложение БД было разработано на шахте «Комсомолец Донбасса» с помощью CASE-оболочки SWS [23] сторонними пользователями этой инструментальной системы. В работе имеется соответствующий акт внедрения. Как показано в [24], само CASE-средство SWS проектировалось в строгом соответствии с каркасной моделью данных. При этом приложения БД, синтезируемые с помощью SWS и каркасной схемы БД, обладают высокой эффективностью.

ПрО «Учет отгрузки угля на шахте»

На основании алгоритмов нормализации [18] и каркасного синтеза ДКНФ [3] были исследованы:

- 1) семантика ПрО;
- 2) каркасная схема БД;
- 3) характеристики доступа к большим объемам данных.

Рассмотрим фрагмент ПрО. Для этого приведем список всех сущностей-объектов с указанием их классификации и схемы (курсивом выделены ключевые атрибуты):

ПОКАЗАТЕЛИ КАЧЕСТВА ПАРТИИ (*КодПоказателя*, Наименование) — атомарная сущность-объект;

ПОКАЗАТЕЛИ ПАРТИИ СКИДКИ-НАДБАВКИ (*КодПоказателя*, *КодСкидкиНадбавки*, *КодДиапазонаПоказателя*, СуммаСкидкиНадбавки) — составная сущность-объект;

ПОТРЕБИТЕЛИ (*КодПотребителя*, Наименование) — атомарная сущность-объект;

АДРЕСА ПОТРЕБИТЕЛЕЙ (*КодГос*, *КодОбл*, *КодГор*, *КодАдреснКатегор*, *НомерЗдания*, *НомерОфиса*, *КодПокупателя*, Примечание) — составная сущность-объект;

ТЕЛЕФОНЫ ПОТРЕБИТЕЛЕЙ (*КодПотребителя*, *КодТипаТелефона*, *НомерТелефона*) — слабая сущность-объект;

КОНТАКТЫ ПОТРЕБИТЕЛЕЙ (*КодПотребителя*, *КодТипаКонтакта*, *КодКонтакта*, Контакт) — составная сущность-объект;

ПОТРЕБИТЕЛИ КАК ПЛАТЕЛЬЩИКИ (*КодПотребителя*, *КодКатегорииПлательщика*, Примечание) — составная сущность-объект;

ПАРТИЯ ПРОДУКТА (*КодКатегПродукта*, *НомГода*, *КодМесяца*, *КодШахты*, *КодУчастка*, *КодПартии*, Объем партии, Примечание) — составная сущность-объект;

ЦЕНА ПРОДУКТА (*КодКатегПродукта*, *НомГода*, *КодМесяца*, *ЦенаЕдиницыПродукта*, Примечание) — составная сущность-объект;

ПАРТИЯ ПРОДУКТА ПОТРЕБИТЕЛЮ (*КодКатегПродукта*, *НомГода*, *КодМесяца*, *КодШахты*, *КодУчастка*, *КодПартии*, *КодПотребителя*, Объем продукта, Дата, Примечание) — составная сущность-объект;

УДОСТОВЕРЕНИЕ ПАРТИИ (*КодКатегПродукта*, *НомГода*, *КодМесяца*, *КодШахты*, *КодУчастка*, *КодПартии*, *НомерУдостовер*, *КодЗольности*, *КодВлажности*, *КодТеплотыСгоран*, *КодПроцСеры*, Дата, Примечание) — артефакт, заменяется одноименной составной сущностью-объектом;

ТРАНСПОРТНАЯ НАКЛАДНАЯ ПАРТИИ ПОТРЕБИТЕЛЮ (*КодКатег-Продукта, НомГода, КодМесяца, КодПартии, КодПотребителя, НомВагона, Объем партии, Дата, Примечание*) — артефакт, заменяется одноименной составной сущностью-объектом;

ЦЕНОВАЯ КОМПЕНСАЦИЯ-НАДБАВКА ПОЛУЧАТЕЛЮ ПАРТИИ (*КодПоказателя, КодСкидкиНадбавки, КодДиапазонПоказателя, КодКатегПродукта, НомГода, КодМесяца, КодШахты, КодУчастка, КодПартии, КодПотребителя, Сумма скидки-надбавки, Примечание*) — составная сущность-объект;

СЧЕТ К ОПЛАТЕ ПОТРЕБИТЕЛЮ (*КодКатегПродукта, НомГода, КодМесяца, КодШахты, КодУчастка, КодПартии, КодПоказателя, КодСкидкиНадбавки, КодДиапазонПоказателя, КодПотребителя, Объем продукта, Цена объема продукта, СуммаСкидкиНадбавки, Итого к оплате, Дата выписки, Номер счета*) — артефакт, заменяется одноименной составной сущностью-объектом;

ТЕРРИТОРИЯ (*КодОбъединения, КодТерритории, КодОбл, Наименование, Примечание*) — слабая сущность-объект;

ШАХТА (*КодОбъединения, КодТерритории, КодШахты, Наименование, Примечание*) — слабая сущность-объект;

УЧАСТОК ШАХТЫ (*КодОбъединения, КодТерритории, КодШахты, КодУчастка, Наименование, Примечание*) — слабая сущность-объект;

СМЕНА ШАХТЫ (*КодОбъединения, КодТерритории, КодШахты, КодУчастка, НомерСмены, Наименование, Примечание*) — слабая сущность-объект;

БРИГАДА ШАХТЫ (*КодОбъединения, КодТерритории, КодШахты, КодУчастка, КодБригады, Наименование, Примечание*) — слабая сущность-объект;

СМЕНА БРИГАДЫ (*КодОбъединения, КодТерритории, КодШахты, КодУчастка, КодБригады, НомерСмены, Дата, Примечание*) — составная сущность-объект;

ГОД (*НомГода, Примечание*) — атомарная сущность-объект;

МЕСЯЦ (*НомГода, НомМесяца, Наименование, Примечание*) — слабая сущность-объект;

ДОБЫЧА ШАХТЫ ДЕНЬ (*НомГода, НомМесяца, НомДня, КодОбъединения, КодТерритории, КодШахты, Объем, Примечание*) — составная сущность-объект;

ДОБЫЧА УЧАСТКА ШАХТЫ ДЕНЬ (*НомГода, НомМесяца, НомДня, КодОбъединения, КодТерритории, КодШахты, КодУчастка, Объем, Примечание*) — составная сущность-объект;

ДОБЫЧА БРИГАДЫ ШАХТЫ ДЕНЬ (*НомГода, НомМесяца, НомДня, КодОбъединения, КодТерритории, КодШахты, КодУчастка, НомБригады, Объем, Примечание*) — составная сущность-объект;

Для формирования унифицированного запроса к БД, возвращающего группу данных для анализа документов (артефактов) «Удостоверение партии» или «Счет к оплате потребителю», применяется единственная операция выборка из отношения. Причем все соединения, имеющиеся в схеме БД, сформированы не по факту выполнения запроса пользователя к БД, а по факту внесения текущих данных [24].

Запросы могут быть сформулированы на естественном языке. Например: «выбрать группу категорий продукта за все месяцы конкретного года, добытого на конкретном участке конкретной шахты, показатели зольности, влажности, теплоты сгорания и процента серы которых не превышали бы такой-то величины», а также: «выбрать все счета к оплате за текущий месяц текущего года, по которым для конкретной категории продукта, добытого на произвольных участках конкретной шахты разными бригадами и в разные смены, скидки-надбавки были бы не выше указанных».

На рис. 1 приведена каркасная схема БД указанной ПрО. Как отмечалось в [25], она существенно отличается от диаграммы Чена [12] отсутствием дуг. В каркасной диаграмме применение дуг нецелесообразно. Нецелесообразно также и отмечать на диаграмме обязательность, вхождение и степень каждой связи сущности-объекта, как предложено Р. Баркером [26].

Ценовая Компенсация- Надбавка Партии Потребителю <i>КодКатег:Продукта</i> <i>НомерГода</i> <i>КодМесяца</i> <i>КодПартии</i> <i>КодШахты</i> <i>КодУчастка</i> <i>КодПоказателя</i> <i>КодСкидкиНадбавки</i> <i>КодДиапаз:Показат</i> <i>КодПотребителя</i> Сумма Склад-Надб Примечание	Показатели Качества Партии Продукта <i>КодПоказателя</i> Наименование Цена Продукта <i>КодКатег:Продукта</i> <i>НомерГода</i> <i>КодМесяца</i> <i>ЦенаЕдиницыПрод</i> Примечание	Показатели Скидки- Надбавки к Цене Партии <i>КодПоказателя</i> <i>КодСкидкиНадбавки</i> <i>КодДиапазонаПоказат</i> Наименование Сумма Склад-Надб	Адреса Потребителей <i>КодГосуд</i> <i>КодОбл</i> <i>КодГорода</i> <i>КодАдресКатег</i> <i>НоМЗдания</i> <i>НоМОфиса</i> <i>КодПотребителя</i> Наименование Примечание	Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание	Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание	Смены на Шахте <i>КодОбъединения</i> <i>КодТерритории</i> <i>КодШахты</i> <i>КодУчастка</i> <i>КодСмены</i> Наименование Примечание	Смена Бригады <i>КодОбъединения</i> <i>КодТерритории</i> <i>КодШахты</i> <i>КодУчастка</i> <i>КодСмены</i> <i>КодБригады</i> Дата Примечание	Добыча Бригады за День <i>КодОбъединения</i> <i>КодТерритории</i> <i>КодШахты</i> <i>КодУчастка</i> <i>КодБригады</i> <i>КодСмены</i> <i>НоМГода</i> <i>КодМесяца</i> <i>НоМДня</i> Объем Примечание	Добыча за День на Участке <i>КодОбъединения</i> <i>КодТерритории</i> <i>КодШахты</i> <i>КодУчастка</i> <i>НоМГода</i> <i>КодМесяца</i> <i>НоМДня</i> Объем Примечание
Счет к оплате Потребителю <i>КодКатег:Продукта</i> <i>НомерГода</i> <i>КодМесяца</i> <i>КодПартии</i> <i>КодШахты</i> <i>КодУчастка</i> <i>КодПоказателя</i> <i>КодСкидкиНадбавки</i> <i>КодДиапаз:Показат</i> <i>КодПотребителя</i> Объем партии Цена объема Сумма Склад-Надб Итого к оплате Дата выписки Номер счета Примечание	Удостоверение Партии <i>КодКатег:Продукта</i> <i>НомерГода</i> <i>КодМесяца</i> <i>КодПартии</i> <i>КодШахты</i> <i>КодУчастка</i> <i>НомерУдостовет</i> <i>КодЗолжности</i> <i>КодТер:Сгорания</i> <i>КодСодерж:жСеры</i> Дата Объем партии Примечание	Транспортная Накладная Потребителю <i>КодКатег:Продукта</i> <i>НомерГода</i> <i>КодМесяца</i> <i>КодПартии</i> <i>КодШахты</i> <i>КодУчастка</i> <i>НомерУдостовет</i> <i>НоМВагона</i> <i>КодПотребителя</i> Дата Объем партии Примечание	Территория <i>КодОбъединения</i> <i>КодТерритории</i> <i>КодШахты</i> Наименование Примечание	Шахта <i>КодОбъединения</i> <i>КодТерритории</i> <i>КодШахты</i> <i>КодУчастка</i> <i>КодСмены</i> <i>КодБригады</i> Дата Примечание	Шахта <i>КодОбъединения</i> <i>КодТерритории</i> <i>КодШахты</i> <i>КодУчастка</i> <i>КодСмены</i> <i>КодБригады</i> Дата Примечание	Смена Бригады <i>КодОбъединения</i> <i>КодТерритории</i> <i>КодШахты</i> <i>КодУчастка</i> <i>КодСмены</i> <i>КодБригады</i> Дата Примечание	Смена Бригады <i>КодОбъединения</i> <i>КодТерритории</i> <i>КодШахты</i> <i>КодУчастка</i> <i>КодСмены</i> <i>КодБригады</i> Дата Примечание	Добыча за День на Участке <i>КодОбъединения</i> <i>КодТерритории</i> <i>КодШахты</i> <i>КодУчастка</i> <i>НоМГода</i> <i>КодМесяца</i> <i>НоМДня</i> Объем Примечание	Добыча за Смену на Участке <i>КодОбъединения</i> <i>КодТерритории</i> <i>КодШахты</i> <i>КодУчастка</i> <i>КодСмены</i> <i>НоМГода</i> <i>КодМесяца</i> <i>НоМДня</i> Объем Примечание
Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание	Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание	Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание	Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание	Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание	Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание	Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание	Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание	Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание	Потребители Плательщики <i>КодПотребителя</i> <i>КодКатег:Плател</i> Наименование Телефоны Потребителей <i>КодПотребителя</i> <i>КодКатег:Телефона</i> Номер телефона Примечание

Под обязательностью [27] понимается характеристика связи, показывающая гарантию хотя бы одного вхождения в связь экземпляра каждой сущности, а под вхождением в связь [4, 27], например, ключевого атрибута, понимается число разных значений этого атрибута при постоянном значении конкатенации всех оставшихся. Вхождение принимает одно из значений множества $\{0, 1, \dots, n\}$. Этот показатель в разных концептуальных моделях и разных работах называют по-разному. Например, в UML-моделировании [28] он назван «кратностью», а в серии работ авторов [29] исследован более обобщенный показатель «степень участия» — полная (все вхождения в связь не равны 0) и частичная (имеется хотя бы одно вхождение, равное 0).

Под степенью связи сущностей-объектов понимается традиционный дискретный показатель типа «множественности» связи (отображения) экземпляра одной сущности-объекта с экземплярами других сущностей-объектов — от 1:1 до $H:G$. Разные авторы называют его по-разному. В [29], например, это свойство названо «кардинальностью» связи.

В каркасной схеме БД указанные характеристики связей сущностей-объектов и их ключевых атрибутов не влияют на иные свойства отношений. Этот важный результат совпадает с аналогичным выводом работы [7].

В приведенной схеме БД синтезировано 25 каркасных отношений. Скорость доступа к данным по типовым запросам повысилась на несколько порядков по сравнению со схемой, разработанной в соответствии с алгоритмом нормализации Кодда [18].

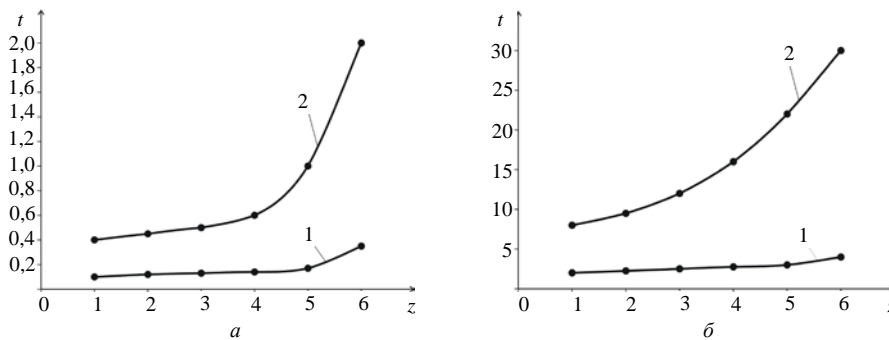


Рис. 2

На рис. 2, *a* приведены графики, иллюстрирующие рост времени доступа к данным при получении одной записи из увеличивающихся пачек записей при запросе на соединение двух ЗНФ-отношений (кривая 2) и заметно меньший прирост времени на индексную выборку этой же записи из одного бинарного отношения в ДКНФ (кривая 1) при таком же увеличении числа записей.

На рис. 2, *б* показан еще меньший прирост времени доступа к 10-арному отношению в ДКНФ (кривая 1) с увеличением числа обрабатываемых записей значительный рост времени выполнения этого же запроса при соединении десяти отношений в ЗНФ (кривая 2). При этом результирующее отношение аналогично примеру 1 из [25] обладало в среднем от 100 до 200 кортежей.

Из приведенного на рис. 1 фрагмента схемы БД видно, что чем сложнее ПрО, тем большее число многоарных связей должно присутствовать в схеме БД. Тем более интенционалы каркасных отношений напоминают высказывания — обычные словесные предложения. Это означает, что реляционные каркасные отношения обладают одновременно и безаномальностью [4], и семантической. Это не совпадает с тезисом о несемантической «нерасширенной» реляционной модели [11]. Фрагмент схемы БД показывает, что подавляющее большинство отношений моделирует связи. В [24] атомарная или слабая сущность-объект также представлена как тривиальная связь. Таким образом, вся совокупность каркасных отношений

(вся БД) есть не что иное, как совокупность связей в ПрО. Поэтому по аналогии с термином ER-модель концептуальная каркасная модель данных может быть названа R-моделью. Различия в понятиях сущности-объекты и связи между ними становятся условными.

Заключение

Описанный подход дает возможность выполнять запросы с использованием индексации идентификатора в соответствии с его структурой, что значительно увеличивает скорость получения ответа и, в свою очередь, объединяет свойства табличной и нетабличной форм хранения. Эту нетипичную форму получают благодаря нетабличному объединению совокупностей данных в атрибуты сущностей-объектов в соответствии с общими по наименованию и структурой идентификаторов. Это новое свойство также важно и для эволюционирования схемы данных при эксплуатации хранилища.

У хранилища, построенного в соответствии с описанным алгоритмом, возникает еще одно преимущество. Предоставляется возможность обособленной и параллельной обработки каждого данного независимо одно от другого, или групповой обработки нескольких объединенных групп данных как зависимо, так и независимо одна от другой. Причем нет потребности в строгом однозначном соответствии каждого данного из общего атрибута по значению, типу и размеру, как того требует, например, реляционный способ размещения, поскольку от каждого данного необходимо лишь наличие общего идентификатора со структурой, соответствующей структуре общего предиката.

Таким образом, предлагается универсальная технология размещения данных в цифровом хранилище, которая не зависит от особенностей определенной ПрО и позволяет в динамике и без переработки эксплуатирующей системы минимально достаточными операциями выполнять любые семантически целесообразные модификации схемы БД и сформировать набор единых процедур обработки данных — групповых функций — и таким образом унифицировать технологию генерации и эксплуатации хранилищ данных.

Б.С. Панченко

АЛГОРИТМ СИНТЕЗУ РЕЛЯЦІЙНОГО КАРКАСА. НЕФОРМАЛЬНИЙ ОПИС

Неформально описано алгоритм каркасної сепарації сутностей-об'єктів для проектування схем БД відповідно до нової моделі даних — реляційного каркаса. Запропоновано нову класифікацію сутностей-об'єктів, а також один з можливих алгоритмів автоматизованого синтезу каркасної схеми реляційної БД. Наведено результати чисельного експерименту доступу до даних.

B.Ye. Panchenko

ALGORITHM OF SYNTHESIS OF A RELATIONAL FRAMEWORK. INFORMAL DESCRIPTION

The paper informally describes the algorithm of framework separation of objects-entities for designing DB schemes in accordance with the new model of data – relational framework. New classification of objects-entities is offered. One of the possible algorithms of automated synthesis of a framework scheme of a relational DB is offered. Results of numerical experiment of data access are given.

1. *Панченко Б.Е.* Об алгоритме синтеза реляционного каркаса. Постановка задачи и формализация // Компьютерная математика. — 2012. — № 1. — С. 84–93.
2. *Карпуша В.Д., Панченко Б.Е.* Моделирование и проектирование реляционных баз данных : Уч. пособие. — Суми : Изд-во СумДУ, 2010. — 385 с.
3. *Панченко Б.Е.* Каркасное проектирование доменно-ключевой схемы реляционной базы данных // Кибернетика и системный анализ. — 2012. — № 3. — С. 174–187.
4. *Fagin R.* A normal form for relational databases that is based on domains and keys // ACM Transact. on Database Systems. — 1981. — 6, N 3. — P. 387–415.
5. *Панченко Б.Е.* Поведение системы некруговых отверстий в полупространстве со свободной границей от воздействия стационарных SH-волн // Международный научно-технический журнал «Проблемы управления и информатики». — 2012. — № 4. — С. 84–93.
6. *Панченко Б.Е., Печенюк Д.А.* Способ автоматизированной цифровой многопрограммной мультисигнальной коммутации. Заявка на изобретение № а 2010 11086 от 15.09.2010.
7. *Тукеев У.А., Алтайбек А.А.* Концептуальная, логическая модели и алгоритм проектирования баз данных в доменно-ключевой нормальной форме // Тр. 13-й Всерос. науч. конф. «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL'2011). — Воронеж, 2011. — С. 119–125.
8. *Соловьев В.Д., Добров Б.В., Иванов В.В., Лукашевич Н.В.* Онтологии и тезаурусы: Уч. пособие. — М. : МГУ, 2006. — 157 с.
9. *Глибовец Н.Н., Глибовец А.Н., Шабинский А.С.* Применение онтологий и методов анализа текстов при создании интеллектуальных поисковых систем // Международный научно-технический журнал «Проблемы управления и информатики». — 2011. — № 6. — С. 95–102.
10. *Ульман Д.Д., Уидом Д.* Основы реляционных баз данных. — М. : Лори, 2006. — 374 с.
11. *Codd E.F.* Extending the database relational model to capture more meaning // ACM Transact. on Database Systems. — 1979. — 4, N 4. — P. 397–434.
12. *Панченко Б.Е., Писанко И.Н.* Свойства реляционного каркаса на множестве семантически атомарных предикатов // Кибернетика и системный анализ. — 2009. — № 6. — С. 177–182.
13. *Chen P.P.* The entity-relationship model: toward a unified view of data // ACM Trans. on Database Systems. — 1976. — 1, N 1. — P. 9–36.
14. *Григорьев Ю.А., Ревунков Г.И.* Банки данных: Уч. пособие. — М. : МГТУ, 2002. — 319 с.
15. *Яловец А.Л.* Представление и обработка знаний с точки зрения математического моделирования. Проблемы и решения. — Киев : Наук. думка, 2011. — 360 с.
16. *Панченко Б.Е.* Способ предварительной каркасной сепарации данных перед их модифицируемым размещением в хранилище или процессом дальнейшей обработки. Патент Украины № 99921 от 17.02.2010 // Промислова власність. — 2012. — № 20. — С. 330.
17. *Кренке Д.М.* Теория и практика построения баз данных. — СПб. : Питер, 2003. — 800 с.
18. *Codd E.F.* The Relational model for database management. Version 2, Reading Mass. — New York : Addison-Wesley Publ. Co, 1990. — 538 p.
19. *Филиппович А.Ю.* Принципы взаимных функциональных зависимостей // Интеллектуальные технологии и системы. — 2002. — Вып. 4. — С. 222–241.
20. *Кунгурцев А.Б., Зиноватная С.Л.* Иерархическая модель объектов для исследования запросов к базе данных // Тр. Одес. политехн. ун-та. — 2008. — Вып. 2 (30). — С. 130–134.
21. *Айвазян С.А., Енюков И.С., Мещалкин Л.Д.* Прикладная статистика. Исследование зависимостей. — М. : Финансы и статистика, 1985. — 487 с.
22. *Буч Г.* Объектно-ориентированный анализ и проектирование с примерами приложений на C++. — М. : Вильямс, 2001 — 560 с.
23. *Панченко Б.Е., Гайдабрус В.Н., Церковицкий С.Л.* CASE-генератор прикладных сетевых информационных комплексов — инструментальная система SWS 1.0 // Свидетельство об официальной регистрации программы для ЭВМ № 940165. — М. : РосААП, 1994. — 2 с.
24. *Панченко Б.Е.* Исследования доменно-ключевой схемы реляционной базы данных // Кибернетика и системный анализ. — 2012. — № 6. — С. 157–172
25. *Панченко Б.Е.* К вопросу о модифицируемости и безаномальности схемы реляционной базы данных // Проблемы программирования. — 2012. — № 2-3. — С. 281–288
26. *Barker R., Longman C.* CASE-method. Function and process modeling. — Addison-Wesley Publ. Co, 1992. — 400 p.
27. *Козленко Л.А.* Проектирование информационных систем // КомпьютерПресс. — 2001. — № 9. — С. 10–11.
28. *Фаулер М., Скотт К.* UML в кратком изложении. Применение стандартного языка объектного моделирования. — М. : Мир, 1999. — 191 с.
29. *Буй Д.Б., Сильвеструк Л.М.* Модель «сутьность-зв'язок»: формалізація сутностей та зв'язків // Вісн. Київ. ун-ту. Сер. фіз.-мат. науки. — 2006. — Вип. 3. — С. 143–152.

Получено 27.08.2012

Статья представлена к публикации акад. НАН Украины А.А. Летичевским.