На правах рукописи

ЗЫКИН Владимир Сергеевич

МЕТОДЫ И АЛГОРИТМЫ ПОДДЕРЖКИ ЦЕЛОСТНОСТИ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ В ПРИЛОЖЕНИЯХ КЛАССОВ OLAP И OLTP

05.13.17 — теоретические основы информатики

Автореферат
диссертации на соискание ученой степени
кандидата физико-математических наук

Челябинск — 2019

Работа выполнена на кафедре системного программирования
ФГАОУ ВО «Южно-Уральский государственный университет
(национальный исследовательский университет)»

Научный руководитель: ЦЫМБЛЕР Михаил Леонидович
кандидат физ.-мат. наук, доцент,
доцент кафедры системного программирования, ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)» (г. Челябинск)

Официальные оппоненты: КУЗНЕЦОВ Сергей Дмитриевич
доктор техн. наук, профессор, главный научный сотрудник, ФГБУН «Институт системного программирования имени В.П. Иванникова РАН» (г. Москва)

КОСТЕНЕЦКИЙ Павел Сергеевич
кандидат физ.-мат. наук, доцент, начальник отдела Суперкомпьютерного моделирования, ФГАОУ ВО «Национальный исследовательский университет «Высшая школа экономики» (г. Москва)

Ведущая организация: ФГАОУ ВО «Казанский (Приволжский) федеральный университет» (г. Казань)

Защита состоится 25 марта 2020 г. в 11:00 часов на заседании диссертационного совета Д 212.298.18 при ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)» по адресу: 454080, г. Челябинск, пр. Ленина, 76, ауд. 1001.

С диссертацией можно ознакомиться в библиотеке Южно-Уральского государственного университета и на сайте:
https://www.susu.ru/ru/dissertation/d-21229818/zykin-vladimir-sergeevich.

Автореферат разослан «___» _____ 20__ г.

Ученый секретарь
диссертационного совета М.Л. Цымблер

Общая характеристика работы

Актуальность темы. Целостность данных является ключевым понятием в современных реляционных базах данных (БД). Целостность регламентирует соответствие данных в БД их структуре, логике и всем заданным правилам. Реляционные системы управления базами данных (СУБД) строятся в соответствии с трехуровневой архитектурой ANSI–SPARC. Внутренний (физический) уровень отвечает за физический способ организации данных. Промежуточный (концептуальный) уровень инкапсулирует реляционную схему БД. Внешний (пользовательский) уровень показывает, как выглядит БД с точки зрения конечного пользователя, и реализуется с помощью представлений. На концептуальном уровне ссылочная целостность, наряду с ограничением домена и ограничением сущности, является одним из фундаментальных классов ограничений целостности. Ссылочные ограничения целостности (referential integrity constraint) реализуются в виде зависимостей включения и позволяют сохранить структурную целостность данных, устанавливая логическое соответствие между кортежами отношений БД. В большинстве существующих реляционных СУБД поддерживаются указанные ограничения целостности.

В современных промышленных приложениях оперативной обработки транзакций (OLTP, Online Transaction Processing) и оперативного анализа данных (OLAP, Online Analitical Processing) типичной является ситуация, когда база данных насчитывает десятки и даже сотни таблиц. Для проектирования схемы корпоративных БД указанного масштаба в настоящее время широко используются средства автоматизированного проектирования схем БД (ERwin Data Modeler, ORACLE SQL Developer Data Modeler и др.). Однако, использование указанного инструментария при построении реляционной схемы БД приводит, как правило, к появлению атрибутов-синонимов и атрибутов-омонимов, принадлежащих разным отношениям схемы БД. Вследствии этого в базе данных появляются дублированные и противоречивые атрибуты, между которыми в процессе проектирования устанавливаются зависимости включения. В результате формируется некорректная схема, которая не обеспечивает целостность хранимых данных, что влечет за собой значительные финансовые и временные затраты на сопровождение БД.

С момента появления реляционной модели данных *актуальным* аспектом проблемы организации поддержки целостности данных является обработка *неопределенных значений (NULL-значений)*, которые используются для обозначения факта отсутствия информации. Для обеспечения *ссылочной целостности данных* неопределенные значения должны быть учтены при проектировании схемы БД в функциональных зависимостях и зависимостях включения, устанавливаемых между атрибутами связанных отношений. Существующие подходы к обеспечению ссылочной целостности данных основаны на использовании зависимостей включения, устанавливаемых над любыми перестановками атрибутов в отношениях. Однако в технологиях БД атрибуты идентифицируются своим именем, а не позицией в структуре логической записи, поэтому необходимо устанавливать зависимости по совпадающим множествам атрибутов.

В существующей теории зависимостей включения возникает необходимость совместной аксиоматизации функциональных зависимостей и зависимостей включения. Алгоритмы поиска полного и неизбыточного набора таких зависимостей включения характеризуются экспоненциальной сложностью. Перечисленные проблемы требуют нового формального аппарата зависимостей включения, допускающих неопределенные значения. Наличие указанного формального аппарата приведет к упрощению системы аксиом и даст возможность рассмотреть зависимости включения отдельно от функциональных зависимостей и снизить вычислительную сложность процесса построения зависимостей включения. Таким образом, *актуальной* является проблема построения новой системы аксиом (типизированных) зависимостей включения, которые устанавливаются по совпадающему множеству атрибутов и которые допускают неопределенные значения.

Важным аспектом ссылочной целостности является поддержка целостности представлений БД. *Представление (view)* реализует внешний уровень архитектуры ANSI–SPARC для определенного класса пользователей БД. В современных технологиях БД остается *актуальной* задача поддержки целостности данных при обновлении представлений. В настоящее время отсутствует общий подход к обновлению представлений, где одна запись в представлении ссылается на несколько кортежей в базовых отношениях БД. Стандарт SQL существенно ограничивает обновление представлений случаем, когда одна запись в представлении ссылается на один кортеж в хранимом отношении БД. В современных СУБД для обновления представлений программисту необходимо создавать новое приложение (триггер) для каждого отдельного представления, что влечет за собой значительные накладные расходы.

Цель и задачи исследования. *Целью* данной работы является исследование и разработка эффективных методов и алгоритмов поддержки целостности данных на внешнем и концептуальном уровнях архитектуры реляционных баз данных для приложений классов OLAP и OLTP. Для достижения этой цели были поставлены следующие задачи:

1. Разработать систему аксиом типизированных зависимостей включения, которая обеспечивает ссылочную целостность при наличии неопределенных значений.
2. Разработать алгоритм построения неизбыточного множества типизированных зависимостей включения с доказательством его корректности и оценкой его вычислительной сложности.
3. Разработать общий подход к обновлению многотабличных представлений, обеспечивающий корректную модификацию записи в представлении, которой соответствуют несколько кортежей в хранимых отношениях БД.
4. Реализовать предложенные методы и подходы в виде сопроцессора СУБД для приложений классов OLAP и OLTP.
5. Провести вычислительные эксперименты, подтверждающие эффективность предложенных подходов.

Научная новизна заключается в том, что впервые была построена полная и непротиворечивая система аксиом для типизированных зависимостей включения, допускающих наличие неопределенных значений. По сравнению с ранее известной аксиоматикой зависимостей включения рассматриваемые в данной работе типизированные зависимости включения устанавливаются только по совпадающему множеству атрибутов, что обеспечивает независимость данных от структуры БД. Разработан оригинальный алгоритм построения типизированных ациклических зависимостей включения. Разработан оригинальный общий подход к обновлению многотабличных представлений на основе коммутативных преобразований данных, который в отличие от аналогов позволяет обновлять запись в представлении, если ей соответствуют несколько кортежей в базовом отношении.

Теоретическая ценность работы заключается в том, что была сформулирована система аксиом для типизированных зависимостей включения. Получено новое доказательство полноты и непротиворечивости указанной системы аксиом, схема доказательства может быть использована при анализе других видов зависимостей в БД. Доказана корректность и получена оценка вычислительной сложности алгоритма формирования неизбыточного множества типизированных зависимостей включения. Предложен общий подход к обновлению многотабличных представлений, основанный на частичном порядке зависимостей включения, доказана корректность операций. **Практическая ценность** работы заключается в реализации сопроцессора реляционной СУБД, обеспечивающего обновление многотабличных представлений, в которых одной записи в представлении соответствует несколько кортежей в хранимых отношениях БД. Результаты, полученные в диссертационной работе, могут быть использованы для создания инструментальных средств проектирования схем баз данных и сопроцессоров обновления многотабличных представлений для различных коммерческих и свободно распространяемых реляционных СУБД.

Методы исследования. Методологической основой диссертационной работы являются методы математической логики, теория проектирования реляционных БД, теория множеств и реляционная алгебра. Для программной реализации разработанных подходов применялись методы объектно-ориентированного проектирования, язык UML и методы модульного программирования.

Степень достоверности результатов. Достоверность научных результатов, полученных в работе, подтверждается строгими математическими доказательствами. Теоретические построения подтверждаются вычислительными экспериментами, проведенными в соответствии с общепринятыми стандартами.

Апробация работы. Основные положения диссертационной работы, разработанные методы, алгоритмы и результаты вычислительных экспериментов докладывались автором на следующих международных научных конференциях:

– Международная научная конференция «Математика в современном мире» (14–19 августа 2017 г., Новосибирск);
– The 2th International Conference «Numerical Computations: Theory and Algorithms» (NUMTA2016) (19 – 25 June 2016, Pizzo Calabro, Calabria, Italy);

– The 10th IEEE International Scientific and Technical Conference on Dynamics of Systems, Mechanisms and Machines (15 – 17 November 2016, Omsk, Russia);
– VIII Международная молодежная научно-практическая конференция «Прикладная математика и фундаментальная информатика» (26 апреля – 4 мая 2018 г., Омск);
– The VI International Conference «Optimization and Applications» (OPTIMA-2015) (September 27 – October 3, 2015, Petrovac, Montenegro).

Публикации. По теме диссертации опубликовано 12 печатных работ. Работы [1–5] опубликованы в журналах, включенных ВАК в перечень изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание ученой степени доктора и кандидата наук. Работы [6–8] опубликованы в изданиях, индексируемых в Scopus и Web of Science. Работы [9,10] включены в перечень изданий, индексируемых в РИНЦ. В рамках выполнения диссертационной работы получено два свидетельства о государственной регистрации программы для ЭВМ [11,12].

Личный вклад. В работе [3] Зыкину В.С. принадлежат разделы 1–4 (обзор работ, коммутативные преобразования реляционных отношений, сопроцессор коммутативных преобразований, экспериментальное исследование, заключение, стр. 94–106), научному руководителю Цымблеру М.Л. принадлежит введение (стр. 92–94). В работе [4] Зыкину В.С. принадлежат разделы 1–4 (обзор результатов, основы формальной теории, минимальное покрытие множества зависимостей и заключение, стр. 155, 157–167), Зыкину С.В. принадлежит введение (стр. 156). В работе [5] Зыкину В.С. принадлежат разделы 1–3 (обзор работ, метод коммутативных преобразований данных, редактирование многотабличных представлений данных и заключение, стр. 332–338), Зыкину С.В. принадлежит введение (стр. 330–332). В работе [6] Зыкину В.С. принадлежат разделы 1–4 (обзор результатов, основы формальной теории, минимальное покрытие множества зависимостей и заключение, стр. 156–167), Зыкину С.В. принадлежит введение (стр. 155). В работе [7] Зыкину В.С. принадлежат разделы 2–5 (обзор работ, метод коммутативных преобразований, обновление многотабличных представлений, заключение, стр. 1–7), Зыкину С.В. принадлежит введение (стр. 1).

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения и списка литературы. Объем диссертации составляет 137 страниц. Список литературы содержит 119 наименований.

Содержание работы

Во введении приводится обоснование актуальности темы и степень ее разработанности; формулируются цели и задачи исследования; раскрываются новизна, теоретическая и практическая значимость полученных результатов; приводятся данные об апробациях и публикациях автора; дается обзор содержания диссертации.

В первой главе, «Современные методы и средства контроля целостности в реляционных базах данных», представлены понятие целостности данных и основные аспекты контроля целостности в базах данных на основе реляционной модели данных. Рассмотрен комплекс исследовательских и практических проблем, связанных с поддержкой ссылочной целостности при наличии неопределенных значений в данных и контролем целостности при обновлении многотабличных представлений. Приводится аналитический обзор публикаций, наиболее близко относящихся к теме диссертации.

Рассмотрим формальное определение зависимостей включения. Пусть $U = \{A_1, A_2, \dots, A_n\}$ – множество атрибутов, определенных в БД, $[R_i]$ – множество атрибутов, на которых определено отношение R_i, $[R_i] \subseteq U$, $1 \le i \le k$, $\mathfrak{R} = (R_1, R_2, \dots, R_k)$ – БД, $S = \{[R_1], [R_2], \dots, [R_k]\}$ – схема БД.

Определение 1. Пусть имеются $[R_i]$ и $[R_j]$ – схемы отношений (не обязательно различные), $V \subseteq [R_i]$ и $W \subseteq [R_j]$, $|V| = |W|$. *Зависимость включения* устанавливается между двумя отношениями $R_i\,[V]$ и $R_j\,[W]$, если все кортежи в R_i по атрибутам V соответствуют кортежам в R_j по атрибутам W. Зависимость включения обозначается как $R_i\,[V] \subseteq R_j\,[W]$.

В определении 1 $|V|$ – мощность множества V, $R_i\,[V] = \boldsymbol{\pi}_V\,(R_i)$ – проекция отношения R_i по атрибутам V. Определение 1 не учитывает возможное наличие неопределенных значений в отношениях БД. В рассмотренном определении зависимость включения устанавливается по различным множествам атрибутов V и W, что допускает установление зависимости по атрибутам-синонимам. Вследствие чего появляются дублированные и противоречивые атрибуты в БД.

Определение 1 не формализует способ установления зависимости включения на уровне кортежей при наличии неопределенных значений, что приводит к несогласованным определенным и неопределенным значениям.

Рональд Фэйджин и др. предложили в 1984 году следующую систему аксиом зависимостей включения:

- **IND1** (рефлексивность): $R_i[X] \subseteq R_i[X]$, если X последовательность отдельных атрибутов R_i.
- **IND2** (проецирование и перестановка): если $R_i\,[A_1, \dots, A_m] \subseteq R_j\,[B_1, \dots, B_m]$, тогда $R_i\,\left[A_{i_1}, \dots, A_{i_q}\right] \subseteq R_j\,\left[B_{i_1}, \dots, B_{i_q}\right]$ для каждой последовательности $i_1, \dots, i_q$ различных целочисленных значений из множества $\{1, \dots, m\}$.
- **IND3** (транзитивность): если $R_i\,[X] \subseteq R_j\,[Y]$ и $R_j\,[Y] \subseteq R_l\,[Z]$, тогда выполнено $R_i\,[X] \subseteq R_l\,[Z]$.

Наличие перестановок во второй аксиоме, приводит к необходимости рассматривать дополнительную зависимость включения для каждой возможной перестановки атрибутов в отношениях. В соотсветствии с этим увеличивается вычислительная сложность построения множества зависимостей включения и процесса проектирования базы данных в целом.

Во второй главе, «Ссылочная целостность отношений на основе теории зависимостей включения», предлагается новое понятие типизированной зависимости включения, строится система аксиом для типизированных зависимостей включения и доказывается ее полнота и непротиворечивость. Предлагается алгоритм построения замыкания множества типизированных зависимостей включения. Доказывается корректность предложенного алгоритма. Представлен механизм внедрения в реляционную СУБД типизированных зависимостей включения как ограничений целостности, задаваемых при создании таблицы. Представлен алгоритм поиска и удаления избыточных (выводимых) зависимостей включения.

Ссылочные ограничения являются одним из основных видов ограничений, которые позволяют сохранить структурную целостность БД. С другой стороны, обеспечение корректной обработки неопределенных значений в кортежах ссылающихся отношений стало актуальной задачей с момента создания реляционной модели данных. При проектировании схемы БД необходимо учитывать различные виды зависимостей в данных и выполнять формализацию зависимостей применительно к неопределенным значениям. Это касается и зависимостей включения, которые являются теоретической основой ссылочной целостности данных.

Далее вводится новое определение соответствующих друг другу кортежей, которое учитывает возможное наличие неопределенных значений в атрибутах этих кортежей.

Определение 2. Будем говорить, что кортеж $t_i[X]$ *соответствует* кортежу $t_j[X]$ по атрибутам X и обозначать это как $t_j[X] \preccurlyeq t_i[X]$, если для любого атрибута $A_l \in X$ выполнено одно из двух условий:

1) если $t_i[A_l] \neq NULL$, тогда $t_j[A_l] = t_i[A_l]$ или $t_j[A_l] = NULL$;
2) если $t_i[A_l] = NULL$, тогда $t_j[A_l] = NULL$.

Отношение соответствия кортежей из определения 2, очевидно, является транзитивным: если $t_j[X] \preccurlyeq t_i[X]$ и $t_i[X] \preccurlyeq t_m[X]$ тогда $t_j[X] \preccurlyeq t_m[X]$.

Определение 3. Будем говорить, что существует *типизированная зависимость включения* подчиненного отношению $R_j[X]$ от главного отношения $R_i[X]$ по атрибутам X и обозначать это как $R_j[X] \subseteq R_i[X]$, если для любого кортежа $t_j[X] \in R_j[X]$ имеется соответствующий (в смысле определения 2) кортеж $t_i[X]$ в отношении $R_i[X]$.

Все зависимости включения, которые удовлетворяют определению 1, будем называть *нетипизированными*. Типизированная зависимость включения, в отличие от нетипизированной, формализует на уровне кортежей, как соотносятся определенные и неопределенные значения. Обозначим множество всех зависимостей включения, определенных на схеме БД, через Σ, произвольную зависимость – через σ (возможно, $\sigma \in \Sigma$).

Определение 4. Зависимость σ является *логическим следствием* множества зависимостей Σ (что обозначается как $\Sigma \vDash \sigma$) при выполнении следующего

условия: если данные в БД удовлетворяют зависимостям Σ, то данные в БД удовлетворяют зависимости σ.

Заметим, что все зависимости из Σ по определению 4 являются логическим следствием Σ. Чем больше зависимостей присутствует в множестве Σ, тем с большей вероятностью они могут быть избыточны, и в дальнейшем могут быть удалены из Σ.

Для типизированных зависимостей включения предложена новая система аксиом, допускающая наличие неопределенных значений:

– **INN1** (рефлексивность): если $X \subseteq [R_i]$, тогда $R_i[X] \subseteq R_i[X]$;
– **INN2** (проекция): если $R_j[Y] \subseteq R_i[Y]$ и $X \subseteq Y$, тогда $R_j[X] \subseteq R_i[X]$;
– **INN3** (транзитивность): если $R_j[X] \subseteq R_i[X]$ и $R_i[X] \subseteq R_l[X]$, тогда выполнено $R_j[X] \subseteq R_l[X]$.

Существенно важным отличием новой системы аксиом от существующей системы аксиом **IND1–IND3** является отсутствие перестановок во второй аксиоме: сопоставление атрибутов производится по совпадению их имен, вне зависимости от их порядкового номера в заголовке отношения. При синтетическом подходе к проектированию БД атрибуты должны идентифицироваться своим именем, а не своей позицией в наборе значений.

Для отношений, удовлетворяющих аксиомам **INN1–INN3**, были получены следующие правила вывода:

$$\begin{gathered}\big(R_j[X] \cap R_i[X]\big) \subseteq (R_i[X] \cap R_l[X]),\\ \big(R_j[X] \cap R_i[X]\big) \subseteq \big(R_j[X] \cap R_l[X]\big),\\ \big(R_j[X] \cap R_l[X]\big) \subseteq (R_i[X] \cap R_l[X])\end{gathered} \qquad (1)$$

где $\cap$ – операция пересечения реляционной алгебры.

Представленные правила вывода могут быть использованы для поиска избыточных зависимостей включения.

Определение 5. Зависимость включения σ *выводима* из множества зависимостей Σ за счет системы аксиом **INN1–INN3** (что обозначается как $\Sigma \vdash \sigma$), если при применении аксиом к зависимостям Σ за конечное число шагов будет получена зависимость σ.

Для предложенной системы аксиом **INN1–INN3** доказаны следующие теоремы.

Теорема 2.1. (Непротиворечивость) Система аксиом **INN1–INN3** непротиворечива.

Теорема 2.2. (Полнота) Система аксиом **INN1–INN3** полна.

Предложен алгоритм (Алг. 1), который преобразует исходную схему БД к виду, приемлемому для установления типизированных зависимостей включения при проектировании реляционной БД. При этом необходимо избавиться от синонимов и омонимов среди атрибутов, то есть атрибутов с пересекающимися (совпадающими) доменами. Алгоритм преобразует исходное множество атрибутов отношений, исключая из него синонимы и омонимы.

В алгоритме используются следующие обозначения:

– $U = \{A_1, A_2, \dots, A_n\}$ – универсум (множество атрибутов в БД);
– $\mathrm{DOM}(A_i)$ – функция, возвращающая область значений атрибута (домен);
– $\mathrm{ATTR}(dom)$ – функция возвращающая имя для нового атрибута на основе специфицированного домена.

Алг. 1. TYPING(IN OUT U)

for each A_i **in** U
  for each A_j **in** U **where** $j > i$
    if $\left(\left(\mathrm{DOM}(A_i) \cap \mathrm{DOM}(A_j)\right) = \emptyset\right) \wedge \left(A_i = A_j\right)$ **then** //*исключение омонимов*
      $A_i := ATTR\left(DOM(A_i)\right)$
      $A_j := ATTR\left(DOM\left(A_j\right)\right)$
    if $\left(\left(\mathrm{DOM}(A_i) \cap \mathrm{DOM}(A_j)\right) \neq \emptyset\right)$ **then** //*исключение синонимов*
      if $\left(\mathrm{DOM}(A_i) \subseteq \mathrm{DOM}(A_j)\right)$ **then** $U := U \setminus A_i$
      if $\left(\mathrm{DOM}(A_j) \subseteq \mathrm{DOM}(A_i)\right)$ **then** $U := U \setminus A_j$
      if $\left(\left(\mathrm{DOM}(A_i) \not\subseteq \mathrm{DOM}(A_j)\right) \wedge \left(\mathrm{DOM}(A_j) \not\subseteq \mathrm{DOM}(A_i)\right)\right)$ **then**
        $A_i := ATTR\left(\mathrm{DOM}(A_i) - \mathrm{DOM}(A_j)\right)$
        $A_j := ATTR\left(\mathrm{DOM}(A_j) - \mathrm{DOM}(A_i)\right)$
        $U := U \cup ATTR\left(\mathrm{DOM}(A_i) \cap \mathrm{DOM}(A_j)\right)$
return U

Алгоритм 1 предполагает, что проектировщик схемы БД реализует функцию $\mathrm{ATTR}(dom)$ самостоятельно.

Определение 6. Будем называть множество отношений $R_i^+[X]$ *замыканием отношения* R_i на множестве зависимостей Σ относительно атрибутов X, если $\forall\, R_j \in R_i^+[X]$, существует зависимость включения $R_j[X] \subseteq R_i[X]$, которая выводима из Σ с помощью аксиом **INN1–INN3**.

Предлагается новый алгоритм построения замыкания для отношения R_i относительно атрибутов X (см. Алг. 2). Данный алгоритм выполняет перебор всех зависимостей включения и для каждого подчиненного отношения в Σ проверяется его выводимость. В Алг. 2 внешний цикл WHILE не имеет явного условия завершения. Однако, для выполнения вложенного цикла FOR в его теле к замыканию необходимо добавить хотя бы одно отношение. Следовательно, максимальное количество итераций в алгоритме равно $n \times k$, где n – количество зависимостей в множестве Σ и k – количество отношений в БД.

Доказана следующая **Теорема 2.3**. (Замыкание) Алг. 2 корректно формирует множество $R_i^+[X]$.

Далее сформируем неизбыточное множество зависимостей включения с использованием алгоритма поиска выводимых зависимостей включения и алгоритма 3, который производит удаление выводимых зависимостей.

Алг. 2. CLOSURE(OUT $R_i^*[X]$)

$R_i^*[X] := \emptyset$;
if $X_i^*[R_i] \neq \emptyset$ **then**
    return 0
$R_i^*[X] := R_i$
$substitution$:=**true**
while $substitution$
    $substitution$:=**false**
    for each $\{R_l[Y] \subsetneq R_m[Y]\}$ **in** Σ
        if $R_m \in R_i^*[X]$ **and** $R_l \notin R_i^*[X]$ **and** $X \subseteq Y$ **then**
            $R_i^*[X] := R_i^*[X] \cup R_l$
            $substitution$:=**true**
return $R_i^*[X]$

С учетом количества итераций в Алг. 2 результирующее количество итераций в Алг. 3 будет равно n^2k.

Алг. 3. MIN-COVER(IN OUT Σ)

for each $\{R_j[X] \subsetneq R_i[X]\}$ **in** Σ
    if $R_j \in$ CLOSURE$(R_i^*[X])$ **then**
        $\Sigma := \Sigma - \{R_j[X] \subsetneq R_i[X]\}$
return Σ

Рассмотрим случай зависимостей включения, устанавливаемых на атрибутах первичных и внешних ключей. Данный случай является типичным для классов приложений OLAP и OLTP. Для случая OLAP зависимости включения устанавливаются между таблицей фактов и таблицами размерностей. Для случая OLTP зависимости включения устанавливают ссылочные ограничения целостности.

Пусть:

– $U = \{A_1, A_2, \dots, A_n\}$ – универсум (множество атрибутов в БД);
– $[R_i]$ – схема отношения R_i (множество атрибутов, на котором определено отношение R_i), $[R_i] \subseteq U$;
– $\mathfrak{R} = (R_1, R_2, \dots, R_k)$ – БД;
– $S = \{[R_1], [R_2], \dots, [R_k]\}$ – схема БД, $1 \leq i \leq k$;
– $PK(R_i)$, либо $PK(i)$, первичный ключ отношения R_i.

Обозначим $L(i, j, X)$ – связь типа **1:1** либо типа **1:M** от главного отношения R_i к подчиненному (внешнему) отношению R_j, где X множество атрибутов, по которым установлена связь.

Пусть:

– $L_1(i, j, X)$ – связь типа **1:1**, установленная от отношения R_i к отношению R_j;
– $L_M(i, j, X)$ – связь типа **1:M** от отношения R_i к отношению R_j.

Заметим, что произвольное отношение R_i может содержать несколько потенциальных ключей, и с ним может быть установлено несколько связей, в которых R_i будет являться главным или подчиненным отношением.

Определение 7. *Связь* $L_1(i, j, X)$ может быть установлена между отношениями R_i и R_j, если они имеют совпадающие первичные ключи: $X = PK(R_i) = PK(R_j)$ и для любых реализаций отношений R_i и R_j выполнено $R_j[X] \subseteq R_i[\mathrm{X}]$.

Определение 8. *Связь* $L_M(i, j, X)$ может быть установлена между отношениями R_i и R_j, если $PK(R_i) \neq PK(R_j)$ и $PK(R_i) \subseteq [R_j]$.

В определении 8 отношение R_j может содержать кортежи с неопределенными значениями атрибутов, не являющимися компонентами первичного ключа. Заметим, что определения 7 и 8 соответствуют типизированным зависимостям включения, если соответствующим связям будет установлено свойство ссылочной целостности данных. Далее это свойство поддерживается СУБД за счет создания внешних ключей. Свойство ссылочного ограничения целостности для связи $L_M(i, j, X)$ не гарантирует выполнения соотношения $R_j[X] \subseteq R_i[\mathrm{X}]$, где $X = [R_i] \cap [R_j]$, так как атрибуты, не принадлежащие первичному ключу отношения, могут принимать неопределенные значения. Поиск связей, соответствующих определениям 7 и 8, можно произвести в автоматизированном режиме.

Определение 9. Множество всех отношений БД $\mathfrak{R} = (R_1, R_2, \ldots, R_k)$ будем называть *ациклическим*, если не существует подмножество

$$\{R_{m(1)}, R_{m(2)}, \ldots, R_{m(s)}\} \subseteq \mathfrak{R}, \tag{2}$$

для которого установлены связи

$$L(m(1), m(2), X_1), L(m(2), m(3), X_2), \ldots, L(m(s), m(1), X_s), \tag{3}$$

где $s > 1$ и $X_1 \subseteq X_2 \subseteq \cdots \subseteq X_s$. Иначе множество отношений $\mathfrak{R}$ будем называть *циклическим.*

Поскольку ребра в гиперграфе соответствуют отношениям, а узлы – атрибутам, то «связанными» оказываются отношения с одноименными атрибутами. Такой тип связи соответствует типизированным зависимостям включения. Для нетипизированных зависимостей такая интерпретация отсутствует.

Определение 10. *Гиперграфом на схеме БД* называется пара $G = (U, S)$, вершинами U являются атрибуты БД, гиперребрами S являются схемы отношений $[R_i]$, в которые объединены вершины – атрибуты, входящие на схему отношения $[R_i]$.

Гиперграф на схеме БД является *циклическим*, если для произвольного атрибута A существует последовательность $[R'_1], [R'_2], \ldots, [R'_m]$ и для $1 \leq i \leq m - 1$ выполнены следующие два условия (условия цикличности):

– $A \in [R'_1] \wedge A \in [R'_m]$;

– $[R'_i] \cap [R'_{i+1}] \neq \emptyset$,

В противном случае гиперграф на схеме БД является *ациклическим*.

Доказана следующая теорема о цикличности гиперграфов на схеме БД.

Теорема 2.4. (Цикличность). Если в $\mathfrak{R}$ присутствует подмножество $R_{m(1)}, R_{m(2)}, \ldots, R_{m(s)}$, такое, что выполнены условия цикличности: $L(m(1), m(2), X_1), L(m(2), m(3), X_2), \ldots, L(m(s), m(1), X_s)$ и $X_1 \subseteq X_2 \subseteq \cdots \subseteq X_s$ s $>$ 1, то соответствующий $\mathfrak{R}$ гиперграф на схеме БД будет циклическим.

Определение 11. Связь $L(i, j, X)$ является *избыточной*, если существует последовательность $L(l, j, Y_l)$, $l = 1, \ldots, m$, где $\bigcap_{l=1}^{m} T_l \subseteq T_i$.

Предложен алгоритм для автоматизации формирования связей на схеме БД. Алгоритм проверяет для каждой пары отношений выполнение условий, задаваемых определениями 7 и 8. Пусть L – текущее множество связей, k – количество отношений в текущей БД.

Алг. 4. FORM_REL(OUT L)

```
L := ∅
for i := 1 to k
   for each PK(R_i) in R_i
      for j := 1 to k
         if i ≠ j then
            if PK(R_i) ∈ R_j then
               L := L ∪ Ch(R_i, R_j, V)
return L
```

Алг. 4 имеет вычислительную сложность $O(k^2)$ и его корректность доказывается по индукции.

Теорема 2.5. Связь $L(i, j, X)$ на схеме БД избыточна, если существует последовательность:

$$L(i, m(1), X_0), L(m(1), m(2), X_1), \ldots, L\big(m(p), j, X_p\big) \tag{5}$$

и

$$X \subseteq PK(i) \subseteq X_s \subseteq R_{m(s)},\ s = 2, 3, \ldots, p, \tag{6}$$

где $m(i)$ – номера отношений.

Использование условий (5) и (6) для поиска избыточной связи является экспоненциальной задачей. Однако, если связи соответствуют типизированным зависимостям включения, то к поиску избыточных связей применим Алг. 3 построения замыкания.

Входными данными для Алг. 5 являются связи, установленные с помощью первичных ключей отношений. Входными данными алгоритма является множество связей L. Алгоритм проверяет выполнение условий (5) и (6) для каждой связи в L, в случае выявления избыточной связи алгоритм ее удаляет.

Определим вычислительную сложность алгоритма. Пусть $|L|$ – количество связей в L до обработки его алгоритмом. На каждой итерации цикла WHILE к p должен присоединяться как минимум номер одного отношения, тогда Алг. 5 будет иметь следующую вычислительную сложность: $O(|L|^2 k^2)$, где k – количество отношений БД. Степень при k обусловлена необходимостью проверок двух следующих условий: $v \in p[1, \ldots, q]$ и $w \notin m[1, \ldots, q]$.

Алг. 5. MIN-REL (IN OUT L)

for each $L(l,j,X)$ **in** L
  $q:=1$
  $p(q):=l$
  $changes$: =**true**
  while $changes$
    for each $L(v,w,X_q)$ **in** L **where** $L(l,j,X) \neq L(v,w,X_q)$
      del_{rel}:=**false**
      if $v \in p[1,\dots,q]$ **and** $PK(R_l) \subseteq R_w$ **then**
        if $w = j$ **then**
          $L:=L \setminus L(l,j,X)$
        else
          if $w \notin p[1,\dots,q]$ **then**
            $q:=q+1$
            $p(q):=w$
            $del_{rel}:=$ **true**
      if not del_rel **then**
        $changes:=$ **false**
return L

В третьей главе, «Целостность данных многотабличных представлений на основе коммутативных преобразований данных», предлагается новый подход к обновлению многотабличных представлений, поддерживающий ссылочную целостность данных и который основан на аппарате коммутативных преобразований БД. Приводятся формулы, реализующие операции обновления многотабличных представлений в терминах реляционной алгебры. Формулируются и доказываются теоремы о корректности преобразований, выполняемых в соответствии с предложенными формулами. Описана архитектура сопроцессора СУБД, выполняющего обновление представлений на основе коммутативных преобразований.

Модель информационной системы (*модель данных*) запишем в виде алгебраической системы:

$$\Omega = \langle M, D, O, P \rangle, \tag{7}$$

где

– M – логическая схема данных;
– D – совокупность допустимых состояний БД;
– O – набор операций для модели Ω;
– P – совокупность предикатов, ограничивающих допустимые состояния D;
– M и D в совокупности являются носителем системы.

Модель Ω будем считать *исходной*, а $\Omega' = \langle M', D', O', P' \rangle$ – *целевой* (пользовательской). Следовательно, необходимо построение преобразования:

$$\Omega \Rightarrow \Omega'. \tag{8}$$

Далее рассмотрим последовательность построения преобразования исходной модели Ω в целевую Ω', при которой гарантируется ссылочная целостность БД:

1) определяется набор состояний исходной и целевой моделей;
2) формируются ограничения целостности, накладываемые на обе модели;
3) формируется алгоритм преобразования БД, соответствующий пользовательским обновлениям в представлении.

Установление соответствия состояний целевой модели Ω' состояниям исходной модели Ω осуществляется при помощи многотабличного запроса, который в свою очередь задается в терминах стандартного языка запросов.

Определение 12. Совокупность всех значений данных в БД, неизменных в течение некоторого промежутка времени, будем называть *элементарным состоянием*.

При работе с моделью Ω' пользователь выполняет команду f'_p, которая переводит модель Ω' из состояния $d'_i \in D'$ в состояние $d'_j \in D'$. В программном обеспечении должны быть реализованы алгоритмы, выполняющие соответствующие преобразования модели Ω из состояния $d_i \in D$ в состояние $d_j \in D$.

Определение 13. Преобразование $\Omega \Rightarrow \Omega'$ будем называть *коммутативным*, если выполняются условия:

$$f'_p\big(Q(d_i)\big) = Q\big(Alg_p(d_i)\big), \tag{9}$$

где

– Q – многотабличный запрос;
– Alg_p – алгоритм преобразования исходной БД, сопоставленный команде обновления f'_p.

Другими словами, корректность выполняемых преобразований можно продемонстрировать на следующей коммутативной диаграмме:

$$\begin{array}{ccc} d'_i & \xrightarrow{f'_p} & d'_j \\ Q\uparrow & & \uparrow Q \\ d_i & \xrightarrow{Alg_p} & d_j \end{array} \tag{10}$$

Представленная диаграмма говорит о том, что из начального состояния d_i в конечное состояние d'_j можно перейти двумя различными способами: при выполнении пользовательских обновлений f'_p над представлением Q, либо при выполнении запроса Q к результату преобразований БД Alg_p. В определении 13 предполагается, что при выполнении преобразований состояние БД не изменяется другими приложениями.

Теорема 3.1 Последовательность двух коммутативных преобразований является коммутативным преобразованием.

Выполнение операции преобразования f'_p для модели Ω' должно сопровождаться преобразованием состояния модели Ω. Для этого в системе должен быть реализован алгоритм Alg_p, соответствующий операции f'_p и удовлетворяющий

условию коммутативности. Таким образом, для каждой операции f_p' должен быть реализован свой алгоритм преобразования БД: Alg_p. Из этого следует, что набор операций f_p' должен быть строго регламентирован.

При разработке алгоритмов Alg_p основное внимание должно быть уделено корректности преобразований. Однако, выполнение алгоритмов Alg_p предполагается при работе пользователя с приложением (on line), следовательно, должно быть обеспечено приемлемое время их выполнения.

Далее приводятся аналитические формулы в терминах реляционной алгебры, которые осуществляют преобразования БД в соответствии с обновлением представлений, сделанных пользователем. Формулируются и доказываются теоремы о корректности выполняемых преобразований.

Рассмотрим формальную постановку задачи. Исходная реляционная БД представлена в виде множества отношений: $R_1, R_2, \dots, R_k$. Целевая модель – представление, являющееся результатом выполнения запроса к РБД:

$$Q = \pi_{X_0}\left(\sigma_F(R_1'[X_1] \bowtie R_2'[X_2] \bowtie \cdots \bowtie R_m'[X_m])\right), \quad (11)$$

где

– π_{X_0} – операция проекции по множеству атрибутов X_0;
– σ_F – операция выборки кортежей по условию ;
– F – логическое выражение на атрибутах отношений R_1' , R_2' , … , R_m' ;
– $\bowtie$ – операция естественного соединения отношений РБД;
– $R_i'[X_i]$ – краткая запись операции проекции отношения R_i' по атрибутам X_i.

Атрибут A_j отношения R_i' принадлежит $X_i, i = 1,2 \dots m,$ если выполняется одно из следующих условий:

1) $A_j \in X_0$;
2) $\exists R_l', l \neq i$: $A_j \in \langle R_l' \rangle$;
3) $A_j \in \langle F \rangle$.

Далее рассмотрим обозначения для исходной и целевой моделей данных Ω и Ω'. Схема данных исходной модели данных есть множество схем отношений БД: $M = \{\langle R_1 \rangle, \langle R_2 \rangle, \dots, \langle R_k \rangle\}$, где

– $\langle R_i \rangle$ – все атрибуты отношения R_i;
– $M' = \cup_{i=1}^{m} \langle R_i' \rangle$ – заголовок отношения, являющегося результатом выполнения запроса Q;
– P – первичные и внешние ключи БД;
– P' – реализованные зависимости БД (образы первичных и внешних ключей);
– O и O' – операции вставки, удаления и обновления кортежей в отношениях.

Допустимые состояния D и D' определяются предикатами P и P' соответственно.

Отношения R_1', R_2',…, R_m' должны иметь упорядочение по установленным зависимостям включения: главные отношения в последовательности стоят раньше, подчиненные – позже. Таким образом, должен существовать частичный

порядок, в котором есть только одно отношение R'_m, не имеющее подчиненных отношений. Это отношение будет соответствовать семантике приложения: выполненные операции в приложении будут реализовываться только соответствующими операциями в R'_m. Далее это отношение будем называть *целевым*.

Операция удаления записи. Пусть в приложении запись u удалена из представления, соответствующего запросу Q. Далее запрос и результат его выполнения будем обозначать одним и тем же символом Q.

В соединении отношений

$$R'_1[X_1] \bowtie R'_2[X_2] \bowtie \cdots \bowtie R'_m[X_m] \tag{12}$$

множество кортежей T, удовлетворяющих логическому выражению F и принимающих значения записи u на атрибутах X_0, можно выразить формулой:

$$T = \sigma_{F \wedge (X_0 = u)}(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \cdots \bowtie R'_m[X_m]), \tag{13}$$

где выражение $(X_0 = u)$ означает равенство значений одноименных атрибутов. Тогда операция удаления записи u из представления Q сводится к удалению кортежей T из отношения R'_m

$$R''_m = R'_m \backslash \pi_{\langle R'_m \rangle}(T), \tag{14}$$

где $\mathrm{T} = \pi_{\langle \mathrm{R}'_\mathrm{m} \rangle}\left(\sigma_{\mathrm{F} \wedge (\mathrm{X}_0 = \mathrm{u})}\left(\mathrm{R}'_1[\mathrm{X}_1] \bowtie \mathrm{R}'_2[\mathrm{X}_2] \bowtie \cdots \bowtie \mathrm{R}'_\mathrm{m}[\mathrm{X}_\mathrm{m}]\right)\right)$.

Данное преобразование является коммутативным, поскольку представление

$$Q' = \pi_{X_0}\left(\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \cdots \bowtie R''_m[X_m])\right) \tag{15}$$

будет отличаться от представления Q только отсутствием одной записи u. Заметим, что выполнение операции проекции приводит к устранению кортежей-дубликатов, что позволяет поставить в соответствие удаляемой записи из представления несколько кортежей в хранимых отношениях БД.

Теорема 3.2. Операция удаления записи $u \in Q$ коммутативна.

Аналитическое выражение (14) является замещением алгоритма Alg_1 для удаления кортежа в многотабличном представлении. Для выполнения этого выражения достаточно сформировать соответствующую команду SQL с последующим ее выполнением в какой-либо СУБД.

Операция вставки записи. Введем аналитическое выражение операции вставки записи в терминах реляционной алгебры. Пусть запись u вставляется в представление Q, тогда в исходной БД должны быть выполнены следующие преобразования:

$$R''_m = R'_m \cup T, \tag{16}$$

где F' – проекция формулы F на подпространство атрибутов $\langle F' \rangle = \langle F \rangle \cap (X_1 \cup X_2 \cup \ldots \cup X_{m-1})$ и

$$T = \pi_{X_m}\left(\sigma_F(T' \bowtie u)\right), \tag{17}$$

$$T' = \pi_Y\left(\sigma_{(Z=u[Z])\&F'}(R_1'[X_1] \bowtie R_2'[X_2] \bowtie \cdots \bowtie R_{m-1}'[X_{m-1}])\right), \quad (17)$$
$$Z = X_0 \cap (X_1 \cup X_2 \cup \ldots \cup X_{m-1}),$$
$$Y = (\langle R_m' \rangle \cup \langle F \rangle \cup X_0) \cap (X_1 \cup X_2 \cup \ldots \cup X_{m-1}).$$

Если $Z = \emptyset$, то будем считать, что формула $Z = u[Z]$ принимает значение «Истина».

Операция вставки должна быть отвергнута в одном из следующих случаях:

1) может оказаться пустым множество T' – отсутствуют кортежи, с которыми может быть выполнено соединение (формула $Z = u[Z]$ принимает значение «Ложь» для всех кортежей в T');
2) может оказаться пустым множество T – отсутствуют кортежи, на которых F принимает значение «Истина».

Данное преобразование является коммутативным, если $T' \neq \emptyset$ и $T \neq \emptyset$. Это означает, что представление

$$Q' = \pi_{X_0}\left(\sigma_F(R_1'[X_1] \bowtie R_2'[X_2] \bowtie \cdots \bowtie R_{m-1}'[X_{m-1}] \bowtie R_m''[X_m])\right) \quad (18)$$

будет отличаться от представления Q только присутствием одной записи u.

Теорема 3.3. Если $T' \neq \emptyset$ и $T \neq \emptyset$, то операция вставки записи $u \in Q$ коммутативна.

Для выполнения условия коммутативности в формуле (16) к R_m' достаточно добавить только один кортеж из множества T, однако, необходима вставка всех кортежей множества T, что соединит вновь введенную информацию со всей существующей информацией в БД. Это обеспечивается выбором значений свободных атрибутов $X_1 \cup X_2 \cup X_3 \cup \ldots \cup X_m \backslash Z$ при формировании T'.

Как и в случае с операцией удаления записи, полученное аналитическое выражение (16) является замещением алгоритма Alg_2 для вставки записи в многотабличное представление. Для выполнения выражения (16) достаточно сформировать соответствующую команду SQL с последующим ее выполнением в какой-либо СУБД.

Операция обновления записи. В общем случае операция обновления записи сводится к операциям удаления старой записи u и вставки новой u'. Операции обновления соответствует следующее выражение реляционной алгебры:

$$R_m'' = R_m' \backslash \pi_{\langle R_m' \rangle}\left(\sigma_{F \wedge (X_0 = u)}(R_1'[X_1] \bowtie R_2'[X_2] \bowtie \cdots \bowtie R_m'[X_m])\right) \cup$$
$$\cup\, \pi_{X_m}\left(\sigma_F\left(\pi_Y\left(\sigma_{Z=u'[Z]\&F'}(R_1'[X_1] \bowtie R_2'[X_2] \bowtie \cdots \bowtie R_{m-1}'[X_{m-1}])\right) \bowtie u'\right)\right), \quad (19)$$

где

$$Z = X_0 \cap (X_1 \cup X_2 \cup \ldots \cup X_{m-1}),$$
$$Y = (\langle R_m' \rangle \cup \langle F \rangle \cup X_0) \cap (X_1 \cup X_2 \cup \ldots \cup X_{m-1}). \quad (20)$$

Применив эквивалентные преобразования, получим:

$$R''_m = R'_m \backslash \pi_{X_m}\left(\sigma_F(T \bowtie u)\right) \cup \pi_{X_m}\left(\sigma_F(T \bowtie u')\right), \tag{21}$$

где

$$T = \pi_Y\left(\sigma_{(Z=u[Z])\vee(Z=u'[Z])\ \&F'}(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \cdots \bowtie R'_{m-1}[X_{m-1}])\right),$$
$$Z = X_0 \cap (X_1 \cup X_2 \cup \ldots \cup X_{m-1}), \tag{22}$$
$$Y = (\langle R'_m \rangle \cup \langle F \rangle \cup X_0) \cap (X_1 \cup X_2 \cup \ldots \cup X_{m-1}).$$

В формуле (21) использованы обозначения $R''_m = R'_m \cup T$ и следующее преобразование: выражение

$$\sigma_{F\wedge(X_0=u)}(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \cdots \bowtie R'_m[X_m]) \tag{23}$$

было заменено на эквивалентное

$$\sigma_F(R'_1[X_1] \bowtie R'_2[X_2] \bowtie \cdots \bowtie u). \tag{24}$$

В T объединены списки удаляемых и вставляемых кортежей. Если какие-либо кортежи в T принадлежат обоим спискам, то при формировании $T \bowtie u$ и $T \bowtie u'$ каждый кортеж участвует только в своей операции и лишние кортежи в T игнорируются. Заметим, если $u' = \emptyset$, то формула (21) эквивалентна операции удаления записи u; если $u = \emptyset$, то (21) – вставка записи u'.

Теорема 3.4. Операция обновления записи u представления Q коммутативна.

Для того, чтобы получить возможность обновления представлений, СУБД должна корректно выполнять операции, выраженные аналитическими формулами (14), (18) и (21). Для этого на рабочей станции пользователя должен работать сопроцессор СУБД, который имеет архитектуру, представленную на рис. 1.

Сопроцессор коммутативных преобразований (СКоП) – информационная система, предназначенная для обновления многотабличных представлений данных. Основными операциями, осуществляемыми СКоП, являются операции удаления, добавления и обновления записи в представлении.

Архитектура СКоП предполагает наличие трех следующих подсистем: Коммутатор, Парсер и Локальный каталог. *Парсер* – это подсистема, которая обеспечивает получение метаданных об отношениях, прямо и косвенно вовлеченных в запрос на обновление представления, из словаря базы данных, и их сохранение в Локальном словаре. *Локальный словарь* обеспечивает хранение основных метаданных. При инициализации клиентского приложения СКоП загружает данные Локального словаря в оперативную память. Для каждого нового многотабличного представления добавляются новые записи в Локальный словарь.

Коммутатор – подсистема, которая получает запрос пользователя на обновление представления, формирует текст транзакции, выполняющей обновление набора кортежей в целевом отношении БД, и затем отправляет сформированный текст транзакции на сервер. Коммутатор возвращает клиентскому приложению отклик сервера (результат выполнения операции обновления или код ошибки).

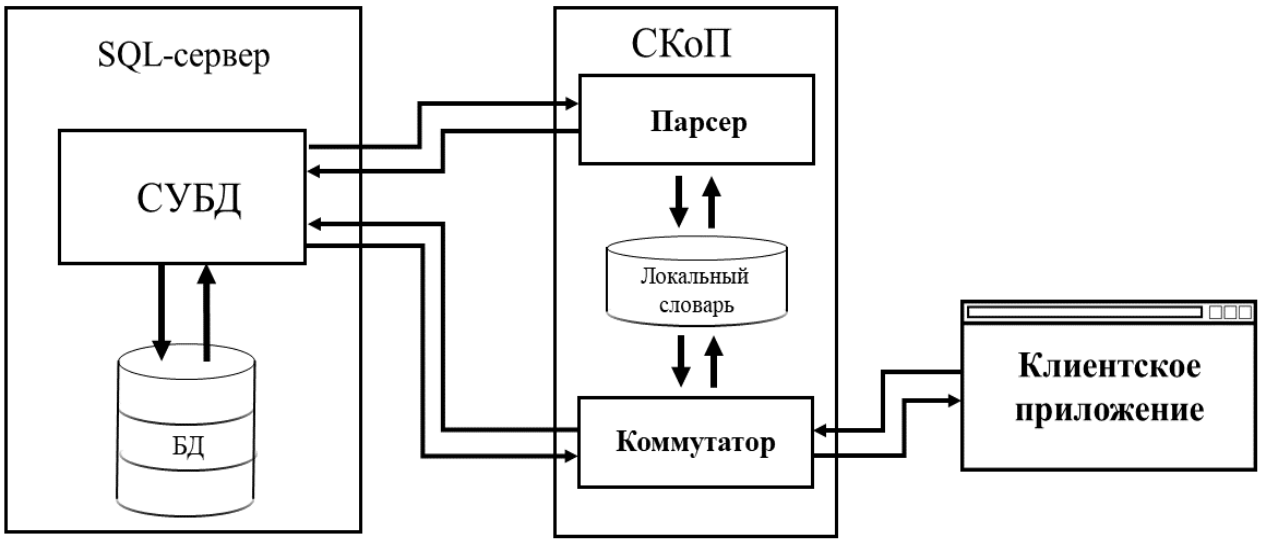

Рис. 1. Архитектура СКоП

В четвертой главе, «Программная реализация и экспериментальные исследования», описывается реализация подхода к обновлению многотабличных представлений на основе коммутативных преобразований данных применительно к свободной СУБД PostgreSQL. Приводятся результаты вычислительных экспериментов, показывающие эффективность разработанных подходов в приложениях классов OLAP и OLTP.

Предложенная архитектура Сопроцессора коммутативных преобразований была реализована для свободной СУБД PostgreSQL. Соответствующий Сопроцессор получил название pgCOPCT (PostgreSQL COProcessor of Commutative Transformations). Данное приложение было разработано на программной платформе .NET Framework на языке программирования C#. Структура pgCOPCT представлена на рис. 2. Программа является оконным приложением под управлением операционной системой Microsoft Windows, которое соединяется с СУБД PostgreSQL, установленной на сервере.

Система pgCOPCT имеет следующие основные компоненты: конструктор запросов (FormBuilder) и редактор представлений (FormResult). Конструктор выполняет формирование многотабличного представления, с помощью методов класса DBConnector, которые заполняют Локальный каталог, выполняя необходимые запросы к БД. Редактор представлений обеспечивает конечному пользователю возможность осуществить операции обновления данных в сформированном представлении. Редактор представлений отправляет текст сформированной транзакции на сервер с помощью методов класса DBEditor. Библиотека Npgsql.dll представляет собой драйвер для подключения СУБД PostgreSQL к среде Visual Studio.

Вычислительные эксперименты проводились с использованием синтетических БД, созданных в соответствии со спецификациями стандартных тестов консорциума TPC (Transaction Processing Council): TPC-H для приложений класса OLAP и TPC-E для приложений класса OLTP.

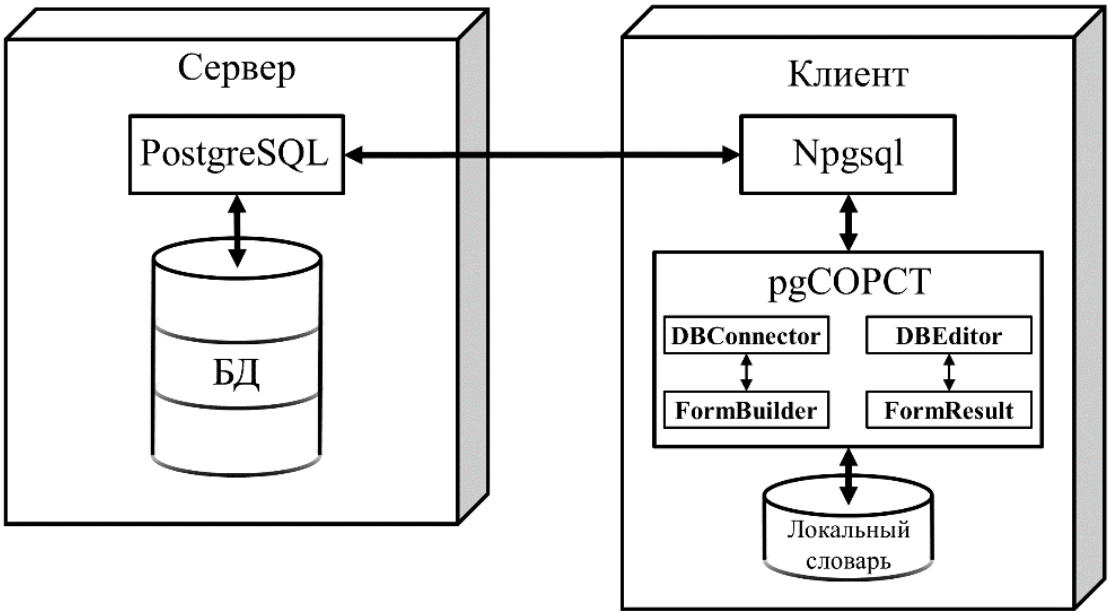

Рис. 2. Структура системы pgCOPCT

В стандартном тесте TPC-H СУБД имитирует обработку заказов. Варьируемым параметром экспериментов является количество кортежей в целевом отношении, соответствующих изменяемому. Результаты экспериментов представлены на рис. 3. Можно видеть, что СУБД MS SQL Server, PostgreSQL и Oracle выполняют операции вставки данных в многотабличное представление с помощью СКоП быстрее на 10–35 %, чем с помощью триггеров.

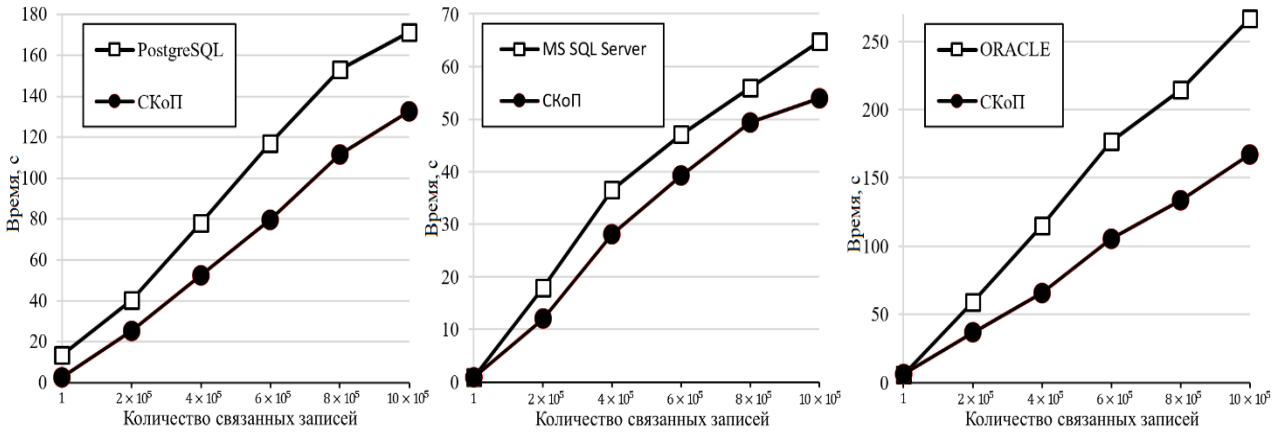

Рис. 3. Эффективность СКоП в приложениях класса OLAP

В стандартном тесте TPC-E СУБД имитирует торговлю на фондовой бирже и включает в себя следующие три сценария выполнения транзакций: обновление информации о сделке (Trade-Update), очистка информации об специфицированной сделке (Trade-Cleanup) и Market-Feed (протоколирование текущей рыночной активности), в каждом из которых выполняется модификация от четырех до шести отношений. Эксперименты проводились с использованием рассмотренного выше сопроцессора СКоП.

Для каждого из указанных сценариев теста TPC-E было сформировано многотабличное представление, обновление которого сводится к выполнению

транзакции, которая выполняет обновление отношений базы данных в соответствии со сценарием. Выполнение каждого сценария с использованием СКоП осуществлялось в режимах холодного и горячего запуска. *Холодному запуску* СКоП соответствует ситуация, когда Парсер сначала формирует Локальный словарь базы данных, а затем Коммутатор выполняет необходимые действия. *Горячий запуск* СКоП означает, что необходимость формирования Локального словаря базы данных отсутствует, и Коммутатор сразу выполняет необходимые действия.

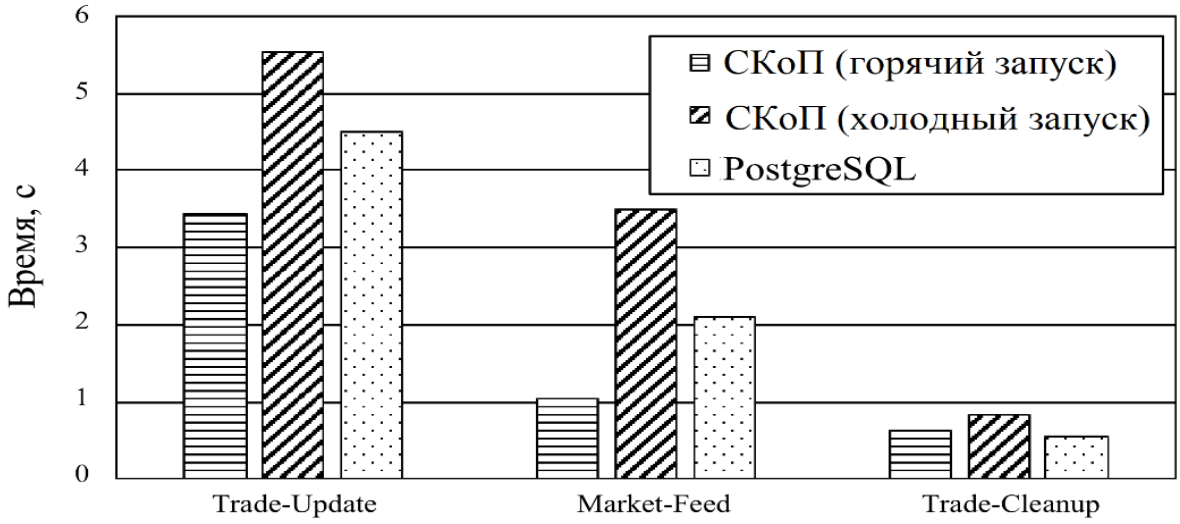

Рис. 4. Эффективность СКоП в приложениях класса OLTP

Результаты экспериментов представлены на рис. 4. Можно видеть, что для каждого из рассмотренных сценариев в режиме горячего запуска СКоП выполняет обновление многотабличного представления быстрее, чем СУБД PostgreSQL с помощью триггеров. Однако использование триггеров выгоднее в режиме холодного запуска. Таким образом, накладные расходы на поддержку Локального словаря базы данных являются необходимой платой за эффективность СКоП.

В заключении в краткой форме излагаются итоги выполненного диссертационного исследования, представляются отличия диссертационной работы от ранее выполненных родственных работ других авторов, даются рекомендации по использованию полученных результатов и рассматриваются перспективы дальнейшего развития темы.

Основные результаты диссертационной работы

На защиту выносятся следующие новые научные результаты:

1. Предложена система аксиом типизированных зависимостей включения с неопределенными значениями в реляционных БД и доказана ее полнота и непротиворечивость.
2. Разработан алгоритм построения неизбыточного множества типизированных зависимостей включения в реляционных БД, доказана его корректность и получена оценка вычислительной сложности.

3. Сформулированы и доказаны теоремы о коммутативных преобразованиях для обновления многотабличных представлений в реляционных БД. Разработана архитектура сопроцессора коммутативных преобразований СУБД и реализован сопроцессор СУБД PostgreSQL.
4. Проведены вычислительные эксперименты, подтверждающие эффективность предложенных подходов.

Публикации по теме диссертации

Статьи в журналах, включенных в Перечень ВАК

1. *Зыкин В.С., Цымблер М.Л.* Обновление многотабличных представлений на основе коммутативных преобразований базы данных // Вестник ЮУрГУ. – 2019. Серия: Вычислительная математика и информатика – Т. 8, № 2. – С. 92–106.
2. *Зыкин В.С.* Ссылочная целостность данных в корпоративных информационных системах // Информатика и ее применения. – 2015. – Т. 9, № 3. – С. 97–105.
3. *Зыкин В.С.* Инструментальная среда формирования внешних ключей на схеме реляционной базы данных // Омский научный вестник. – 2017. – № 1 (151). – С. 140–143.
4. *Зыкин В.С., Зыкин С.В*. Анализ типизированных зависимостей включения с неопределенными значениями // Моделирование и анализ информационных систем. – 2017. – Т. 24, № 2. – С. 155–167.
5. *Зыкин В.С., Зыкин С.В*. Коммутативные преобразования в базе данных при редактировании многотабличных запросов // Информационные технологии. – 2018. – Т. 24, № 5. – С. 330–338.

Статьи в издании, индексируемом в SCOPUS и Web of Science

6. *Zykin V., Zykin S.* Analysis of Typed Inclusion Dependences with Null Values // Automatic Control and Computer Sciences. – 2018. Vol. 52, Iss. 7, P. 638–646.
7. *Zykin S., Zykin V.* Updates of View in Relational Databases // Proceedings of the 12th International Scientific and Technical Conference “Dynamics of Systems, Mechanisms and Machines”, Dynamics 2018, Omsk, Russia, 13–15 November 2018. Article no. 8601495.
8. *Zykin V.* Automatization of Foreign Keys Construction // Proceedings of the 10th International Scientific and Technical Conference “Dynamics of Systems, Mechanisms and Machines”, Dynamics 2016, Omsk, Russia, 15–17 November 2016. Article no. 7819118.

Статьи в изданиях, индексируемых в РИНЦ

9. *Зыкин В.С.* Ограничения целостности для неопределенных значений кортежей // Прикладная математика и фундаментальная информатика. – 2015. – № 2. – С. 227–231.

10. *Зыкин В.С.* Сравнительный анализ различных СУБД при редактировании многотабличных представлений данных // Информационный бюллетень Омского научно-образовательного центра ОмГТУ и ИМ СО РАН в области математики и информатики. – 2017 № 1 (1). – С. 153–154.

Свидетельства о регистрации программ и баз данных

11. *Зыкин В.С.* Программа для построения неизбыточного набора связей на схеме баз данных: свидетельство о государственной регистрации программ для ЭВМ – № 2018661248 от 04.09.2018; Правообладатель: Омский государственный технический университет.

12. *Зыкин В.С.* Редактор многотабличного представления данных: свидетельство о государственной регистрации программ для ЭВМ – № 2018661249 от 04.09.2018; Правообладатель: Омский государственный технический университет

Работа выполнялась при финансовой поддержке Министерства науки и высшего образования РФ (государственное задание 2.7905.2017/8.9).

Формат 60x84 1/16. Бумага офсетная. Печать офсетная.
Усл. печ. л. 1,0. Уч.-изд. л. 1,2. Тираж 100 экз.