На правах рукописи

Ха Ван Муон

МЕТОДЫ ТРАНСЛЯЦИИ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ В ФОРМАТ NOSQL С ОБЕСПЕЧЕНИЕМ ОПТИМАЛЬНОГО ДОСТУПА К ДАННЫМ

Специальность:
05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

Санкт- Петербург – 2022

Работа выполнена на кафедре вычислительной техники федерального государственного автономного образовательного учреждения высшего образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)».

Научный руководитель: **Шичкина Юлия Александровна**
доктор технических наук, профессор кафедры вычислительной техники, ФГАОУ ВО «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)», г. Санкт-Петербург

Официальные оппоненты: **Болдырев Юрий Яковлевич**
доктор технических наук, профессор Высшей школы прикладной математики и вычислительной физики ИПММ СПбПУ, ведущий научный сотрудник научно-исследовательской лаборатории виртуально-имитационного моделирования ИПММ СПбПУ, ФГАОУ ВО «Санкт-Петербургский политехнический университет Петра Великого», г. Санкт-Петербург

Петров Олег Николаевич
кандидат технических наук, доцент кафедры вычислительной техники и информационных технологий, ФГБОУ ВО «Санкт-Петербургский государственный морской технический университет», г. Санкт-Петербург

Ведущая организация: **ФГБОУ ВО «Санкт-Петербургский государственный университет»**, г. Санкт - Петербург.

Защита состоится «25» мая 2022 г. в 15:00 часов на заседании Совета по защите докторских и кандидатских диссертаций Д 212.238.01 Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В.И. Ульянова (Ленина) по адресу: г. Санкт-Петербург, ул. Профессора Попова, д. 5.

С диссертацией можно ознакомиться в библиотеке СПбГЭТУ «ЛЭТИ» и на сайте www.etu.ru в разделе «Подготовки кадров высшей квалификации» - «Объявление о защитах»

Отзывы на автореферат в двух экземплярах, заверенные печатью, просим направлять по адресу: 197022, Санкт-Петербург, улица Профессора Попова, д. 5, лит. Ф.

Автореферат разослан «24» марта 2022 г.

Учёный секретарь
диссертационного совета Д 212.238.01
к.н.т, доцент

/ Пазников А.А.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Получение достоверной информации в современную эпоху является серьезной проблемой, с которой сталкиваются организации. Данная задача требует быстрого доступа либо к единому общему источнику информации, либо к хорошо организованной системе сбора данных из различных источников. В последнем случае проблемой является то, что каждый из источников информации обычно позволяет получить узко специализированную конкретную информацию, хранимую в нем и это, как следствие, влечет потерю представления о запрашиваемом объекте в целом или искажениям информации путем, например, некачественной синхронизации данных. Поэтому задача консолидации данных, особенно текстовых, из различных независимых источников является актуальной.

Существующие сегодня базы данных предназначены для разных типов данных (связанных, не связанных, структурированных, не структурированных и т.д.) и всегда имеют свою индивидуальную логически выстроенную архитектуру под условия конкретной задачи. Среди всего разнообразия баз данных несмотря на интенсивное развитие NoSQL и NewSQL баз данных первенство по-прежнему удерживают реляционные базы данных. Принято считать, что реляционные системы управления базами данных (СУБД) позволяют создавать довольно сложные системы хранения данных, а NoSQL СУБД не приветствуют наличие внутренних связей. Отдельно стоит отметить, что все NoSQL базы данных используют разные модели хранения данных, что усложняет процесс консолидации данных из этих баз данных.

В мире за последние 30 лет проводится много исследований в области поиска подходов к консолидации данных из различных систем хранения и, как следствие, предложено множество различных решений этой проблемы. Среди известных результатов в области консолидации баз данных можно назвать системы SemInt, LSD (Learning Source Descriptions), SKAT (Semantic Knowledge Articulation Tool), TranScm, Palopoli, ARTEMIS, MOMIS (Mediator enviroment for Multiple Information Sources). Однако все эти системы сосредоточены на консолидации реляционных баз данных и не затрагивают проблему консолидации реляционной базы данных с NoSQL или NewSQL базами данных. Таким образом до сих пор не существуют универсальных решений к проблеме консолидации реляционной базой данных с произвольной NoSQL базой данных. Задачу консолидации данных можно упростить, если предварительно провести трансляцию базы данных из собственного формата в формат будущей единой базы данных. Процесс трансляции реляционной базы данных в NoSQL можно разделить на два этапа:

1) Трансляция структуры реляционной базы данных в формат базы данных NoSQL, например «ключ значения», «ключ - документ», «семейство столбцов» или «графовую» базу данных. К наиболее известным подходам к трансляции реляционных баз данных в формат NoSQL относятся: построение правил видоизменения архитектуры базы данных и трансляция на основе методов реляционной алгебры.

2) Трансляция запросов из формата реляционной базы данных в формат NoSQL базы данных. В большинстве исследовательских работ процесс трансляции запросов сфокусирован на разработке программного обеспечения промежуточного уровня, которое выполняет SQL команды для обработки данных для NoSQL баз данных. Еще одним подходом является создание общих языков запросов, которые может использовать для реляционной и NoSQL баз данных одновременно. К таким системам выполнения запросов относятся, например UnQL (Unstructured Query Language) или Impala.

Проведенный анализ подходов к трансляции баз данных из одного формата в другой показал, что нет комплексного подхода, который бы одновременно учитывал типы связей между объектами и структуру запросов к объектам базы данных и позволял бы строить такую структуру базы данных NoSQL, чтобы запросы к ней выполнялись максимально быстро.

Еще одной проблемой, связанной с обработкой данных, является ускорение доступа к данным. Это связано с экспоненциальным ростом данных, повышением сложности моделей, которые строятся на основе этих данных, и повышением технических требований к системам, работающим в режиме реального времени. Среди работ, посвященных оптимизации доступа к данным в различных базах данных, можно выделить следующие направления:

1) Оптимизация структуры централизованной базы данных, включая реляционной базы данных и NoSQL базы данных.

2) Оптимизация структуры распределенной базы данных. Наиболее известными подходами к масштабированию распределенной базы данных являются шардинг и репликация.

3) Оптимизация запросов к базе данных по скорости обработки данных и обеспечению целостности данных. Для оптимизации запросов по скорости в некоторых работах предлагается применение методов распараллеливания и распределения вычислений.

Проведенный анализ существующих исследований в области консолидации и оптимизации баз данных показывает отсутствие:

- формализованных методов трансляции реляционных баз данных в NoSQL;
- формализованных методов оптимизации структур баз данных NoSQL;
- формализованных методов трансляции и оптимизации запросов к NoSQL базам данных с учетом их структуры.

Целью исследования является разработка совокупности формализованных методов трансляции данных из реляционной базы данных в формат базы данных NoSQL типа «ключ-документ» с оптимизацией ее схемы и запросов для ускорения обработки и хранения данных.

Задачи исследования:

1. Анализ существующих подходов к трансляции баз данных из одного формата в другой и оптимизации работы баз данных.
2. Разработка методов оптимизации структуры базы данных NoSQL для моделей «ключ-документ».
3. Разработка методики преобразования реляционной базы данных в формат базы данных NoSQL типа «ключ-документ».
4. Разработка метода оптимизации структуры распределенной базы данных NoSQL типа «ключ-документ».
5. Разработка метода трансляции запросов из формата реляционной базы данных в формат базы данных NoSQL типа «ключ-документ».
6. Создание программных модулей для тестирования разработанных методов.

Объектом исследования являются базы данных различного типа, включая реляционные и NoSQL базы данных.

Предметом исследования являются методы и алгоритмы обработки данных в реляционной и NoSQL базах данных.

Методология и методы исследования. Для решения поставленных задач в диссертационной работе использовались теория баз данных, теория множеств, теория графов, теории функциональных зависимостей, реляционная алгебра. Для разработки программного обеспечения и тестирования разработанных методов использованы языки программирования C# и Python, среда разработки Microsoft Visual Studio Community 2019 версия 16.9.3, Visual studio code версия 1.59.0, СУБД MySQL 8.0.17, СУБД MongoDB 4.4.4 Community.

Научные положения, выносимые на защиту:

1. Методы оптимизации структуры базы данных NoSQL для модели «ключ-документ» с учетом и без учета встроенных документов.
2. Методика преобразования реляционной базы данных в формат базы данных NoSQL типа «ключ-документ».
3. Метод оптимизации структуры распределенной базы данных NoSQL типа «ключ-документ».
4. Метод трансляции запросов из формата реляционной базы данных в формат базы данных NoSQL типа «ключ-документ».

Научная новизна работы заключается в следующем**:**

1. Разработаны методы определения структуры коллекций для базы данных NoSQL типа «ключ-документ» со вложенными и без вложенных документов по заданному набору свойств и запросов, которые основаны на теории множеств и позволяют формализовано представить запросы и схему базы данных и определить оптимальный состав коллекций по скорости доступа к данным и объему хранимых данных путем применения операций над множествами.

2. Разработана методика преобразования реляционной базы данных в формат базы данных NoSQL типа «ключ-документ», которая учитывает различные начальные условия, такие как существование реляционной базы данных и ее принадлежность нормальной форме. Данная методика может быть применена также для преобразования NoSQL баз данных из одного формата в другой и оптимизации схемы базы данных NoSQL типа «ключ-документ».

3. Разработан метод оптимизации структуры распределенной базы данных NoSQL типа «ключ-документ», основанный на теории графов и теории множеств, и позволяющий ускорить доступ к данным путем создания архитектуры распределенной базы данных с учетом метаинформации о запросах к ней.

4. Разработан метод трансляции запросов из формата реляционной базы данных в формат базы данных NoSQL типа «ключ-документ» с учетом ее схемы и структуры самих запросов.

Теоретическая значимость диссертационной работы заключается в:

1) выведении формул, позволяющих определять оптимальную структуру коллекций для базы данных типа «ключ-документ» по заданному набору свойств и запросов;

2) выведении формул, позволяющих определять структуру вложенных документов в БД типа «ключ-документ»;

3) разработке концептуальной схемы применения разработанных методов для получения оптимальной структуры БД типа «ключ-документ» на основе различных начальных условий трансляции или консолидации баз данных;

4) способе представления объектов распределенной базы данных и описании подходов на основе теории графов для оптимизации структуры этих баз данных;

5) формализации подхода к представлению запросов к реляционной базе данных и их трансляции в формат NoSQL с учетом структуры баз данных.

Практическая значимость диссертационной работы заключается в применимости:

1) разработанных методов для трансляции реляционной базы данных в формат NoSQL и последующей консолидации баз данных;

2) разработанных программных модулей для определения структуры коллекций базы данных NoSQL типа «ключ-документ» по заданному набору свойств и запросов, трансляции базы данных из формат реляционной базы данных в NoSQL типа «ключ-документ», трансляции запросов из формата реляционной базы данных в формат NoSQL БД типа «ключ-документ» с учетом структуры базы данных, параллельного выполнения запросов к распределённым базам данных в различных предметных областях и в научных исследованиях.

Реализация и внедрение результатов работы

1. Теоретические и практические результаты диссертационной работы были применены при решении задач в рамках международного научного проекта РФФИ «Создание информационно-технологической поддержки исследований болезни Паркинсона с учетом сбора и обработки данных большого объема в режиме реального времени» (18-57-34001 Куба_т) и программы создания и развития научного центра мирового уровня «Павловский центр «Интегративная физиология – медицине, высокотехнологичному здравоохранению и технологиям стрессоустойчивости» (соглашение от 13.11.2020 №075-15-2020-933).

2. Результаты работы используются в учебном процессе кафедры вычислительной техники ФГАОУ ВПО «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)» при преподавании курсов «Распределенные базы данных», «Не реляционные базы данных», «Оптимизация баз данных».

Апробация результатов работы. Основные положения и результаты работы представлялись и докладывались на следующих конференциях: 2021 IEEE Конференция российских молодых исследователей в области электротехники и электроники (2021 ElConRus) Санкт-Петербург, 26-29 января 2021 г.), II Международная конференция по нейронным сетям и нейротехнологиям (NeuroNT'2021)(Санкт-Петербург, 16 июня 2021 г.), The 21st International Conference on Computational Science and its Applications (Кальяри, Италия, 13-16 сентября 2021 г.).

Публикации. Полученные основные теоретические и практические результаты диссертационного исследования опубликованы в 11 трудах, в том числе в 2 научных статьях в

журналах, рекомендуемых ВАК к опубликованию основных научных результатов диссертаций на соискание ученой степени кандидата наук, 3 научных статьях, опубликованных в зарубежных журналах, входящих в базы цитирования Web of Science и Scopus, 3 публикациях в сборниках конференций, 3 свидетельствах о государственной регистрации программы для ЭВМ.

Структура и объем диссертации

Диссертация состоит из введения, 4 глав, заключения, списка литературных источников, состоящего из 147 наименований, 2 приложений. Работа изложена на 190 машинописных страницах, включая 73 рисунка и 19 таблиц.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении приведена актуальность темы диссертационной работы, освещены объект и предмет исследования, сформулированы цель и задачи исследования, основные положения, выносимые на защиту, прописана научная новизна, описана теоретическая и практическая значимость результатов.

В первой главе представлен обзор существующих подходов к трансляции баз данных из одного формата в другой и оптимизации работы с базами данных NoSQL. В результате анализа существующих типов баз данных установлено, что реляционные БД являются самыми распространенными формами хранения данных. За последние десятилетия все больше используются на практике NoSQL и NewSQL. При переходе от реляционных баз данных к NoSQL или NewSQL актуальной является проблема автоматического преобразования форматов данных и сбора данных из баз данных разного типа. Установлено, что для оптимизации структур баз данных существуют разные подходы, но отсутствуют методы оптимизации структуры NoSQL-баз данных, таких как документные или семейство столбцов.

Проанализированы методы оптимизации запросов к базам данных. Установлено, что в качестве механизмов ускорения выполнения запросов используются индексирование и кэширование, а также параллельное выполнение запросов. Для оптимизации структуры запросов используются методы, основанные на анализе плана выполнения запроса. Формализованных методов оптимизации запросов с учетом метаданных об их структуре и схеме БД не существует.

Анализ существующих подходов к формализованному представлению запросов показал, что наиболее эффективными являются методы теории графов и теории множеств.

Во второй главе описываются методы оптимизации структуры базы данных NoSQL.

Приведен метод определения структуры коллекций для баз данных типа «ключ-документ» по заданному набору свойств и запросов и описана методика применения разработанных методов для трансляции и проектирования баз данных типа «ключ-документ».

Входными данными метода являются: Совокупность свойств объектов, хранимых в базе данных. Если задача заключается в трансляции реляционной базы данных в MongoDB, то в качестве совокупности свойств объекта выступает множество полей всех таблиц.

Пусть T_r – это таблица реляционной базы данных, где r – номер таблицы, $r = 1...k$, k – число таблиц; $T_{(r,j)}$ – это поле в таблице T_r, где j – номер поля, $j = 1...r_n$, r_n – число полей в r-й таблице. Тогда множество полей одной таблицы – это множество вида:

$$T_r = \{T_{(r,j)},\ j = 1, 2, ..., r_n\} \tag{1}$$

Множество всех полей реляционной базы данных будет определяться по формуле:

$$M = \{T_{(r,j)}, r = 1, 2, ..., k \mid T_{(r,j)} \neq T_{(q,i)}, \forall r, q \leq k,\ j \leq r_n, i \leq q_n\} \tag{2}$$

Длина $|M|$ множества M – это количество его элементов.

Совокупность множеств полей, входящих в запрос:

$$S_i = \{T_{(r,j)}, r \leq k,\ j \leq r_n\} \tag{3}$$

где i – номер запроса $(i = 1, 2, ...m)$, m – число запросов к базе данных.

Выходные данные метода: Совокупность множеств коллекций документов с заданными полями:

$$V_i = \{T_{(r,j)}, r \leq k,\ j \leq r_n\}$$

удовлетворяющих условиям:

$$V_1 \cap V_2 \cap V_3 = \varnothing \tag{4}$$

$$V_1 \cup V_2 \cup V_3 = M = \bigcup_{r=1}^{k} T_r \quad (5)$$

$$(\forall S_i)(\exists V_j)(S_i \in V_j, S_i \notin V_i, i \neq j) \quad (6)$$

где i – номер коллекции $(i = 1,2,...,l)$, l – число коллекций.

Метод формирования коллекций для базы данных в формате MongoDB

Шаг 1. Создать множества полей таблиц и запросов по формулам (2) – (3). Число множеств будет равно числу таблиц.

Шаг 2. Выбрать поля, которые не участвуют в запросах. Для этого:

2.1. Для каждого поля составить множества запросов, в которых это поле участвует в любой части конструкции:

$$T'_{(r,j)} = \{S_i, i = 1...p, p \leq m \mid T_{(r,j)} \in S_i\}$$

где m – число запросов, r – номер таблицы, $r = 1...k$, k –число таблиц, j – номер поля в таблице, $j = 1...r_n$, r_n – число полей в r-й таблице.

2.2. Все поля $T_{(r,j)}$, для которых длина множества $|T'_{(r,j)}| = 0$, могут быть включены в единую коллекцию MongoDB:

$$V_1 = \{T_{(r,j)}, r \leq k, j \leq r_n \,||\, T'_{(r,j)}| = 0\}$$

Примечание 1.: коллекции должны формироваться с учетом связей между таблицами.

Шаг 3. Выбрать поля, которые участвуют только в одном запросе, т.е. $|T'_{(r,j)}| = 1$.

Если для любого $T_{(r,j)} \in S_i$ выполняется условие $|T'_{(r,j)}| = 1$, то все поля $T_{(r,j)}$ множества S_i должны войти в новую коллекцию:

$$V_p = \{T_{(r,j)}, r \leq k, j \leq r_n \mid T_{(r,j)} \in S_i \,\&\, |T_{(r,j)}| = 1\}, p > 1 \quad (7)$$

Шаг 4. Убрать из рассмотрения все поля, которые вошли в коллекции V_1 и V_p на шагах 1-3.

Шаг 5. Составить коллекции из полей, к которым обращается несколько запросов.

5.1. Составить множество рассматриваемых запросов: $I = \{S_i\}, i = 1...m$, где m – число рассматриваемых запросов.

5.2. Составить попарные пересечения множеств $S_i \in I$.

$$S'_k = S_i \cap S_j, \forall i \neq j; i, j = 1,2,...,|I|; k = 1,2,...,C^2_{|I|}$$

где $C^2_{|I|} = \dfrac{|I|!}{|(|I|-2)|!2!} = \dfrac{|I|(|I|-1)}{2}$.

5.3. Для полученных не пустых пересечений составить множество P из запросов, входящих в эти пересечения:

$$P = \{S'_i \,||\, S'_i| \neq 0, \forall i \leq k, k = 1,2,...,C^2_{|I|}\}$$

5.4. Найти разность множеств: $I - P$.

Если $I - P \neq \varnothing$, то новая коллекция будет состоять из всех полей, входящих в запросы разности множеств $I - P$.

$$V_p = \bigcup_{i=1}^{|P|} S'_i, \forall S'_i \in P$$

5.5. Положить $I = P$.

5.6. Из полученных пересечений найти новые не пустые пересечения по правилу:

$$if\,(\exists(S_i \cap S_j)\,\&\,\exists(S_j \cap S_k)),\ find\,(S_i \cap S_j \cap S_k)$$

или в общем виде:

$$S''_k = S'_i \cap S'_j, if\,\exists S_m \mid (S_m \in S'_i \,\&\, S_m \in S'_j, i = j); i, j = 1,2,...,|I|; k = 1,2,...,C^2_{|I|}$$

5.7. Повторить шаги 5.3 - 5.7 до тех пор, пока не останется одно пересечение, т.е. длина множества *I* не станет равной 1: $|I| = 1$.

Шаг 6. В новое отношение включить все поля, вошедшие в последнее единственное пересечение и не вошедшие в другие пересечения:

$$V_{p+1} = M - \bigcup_{i=1}^{p} V_i$$

Шаг 7. Конец метода.

Принцип применения предлагаемого метода для разного рода преобразования структуры базы данных приведен на рис.1.

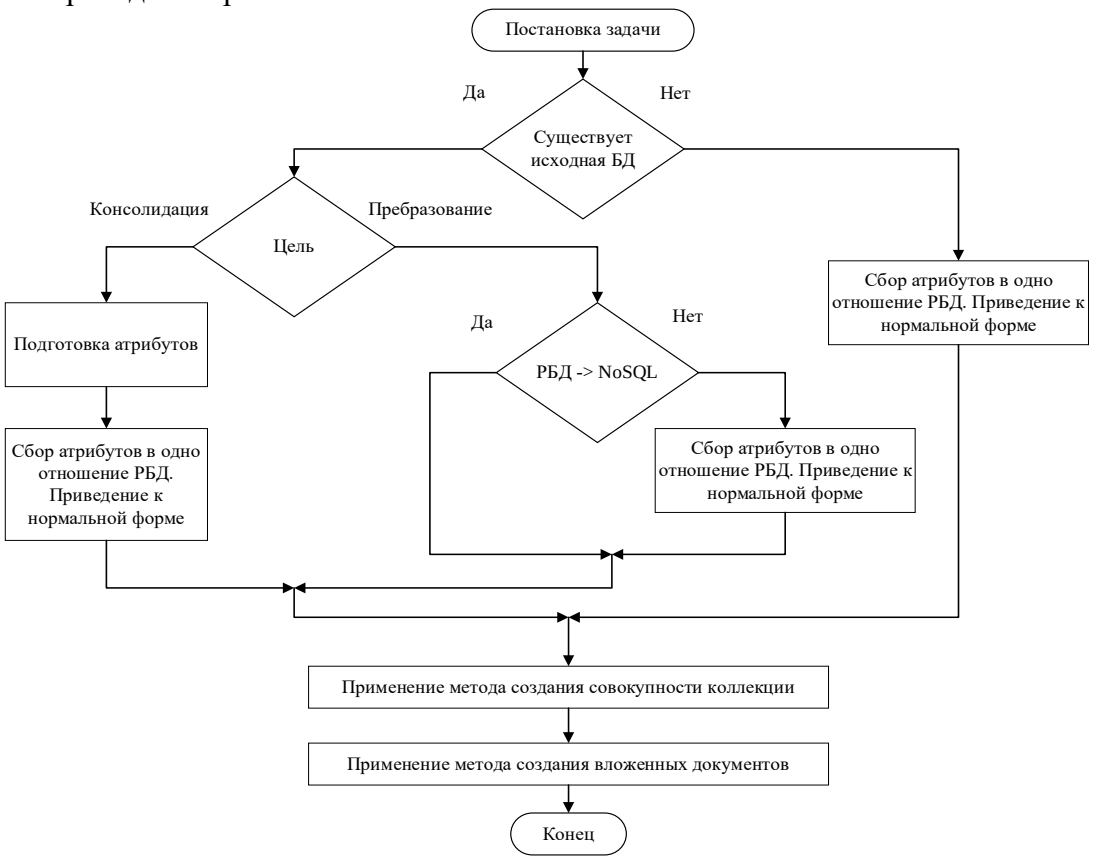

Рис. 1. Применение метода определения коллекций в БД типа ключ-документ

Разработанный метод определения коллекций может быть применен для:

1) создания эффективной структуры новой базы данных типа ключ-документ;

2) трансляции данных из реляционной базы данных в базу данных типа ключ-документ;

3) трансляции данных из произвольной базы данных в базу данных типа ключ-документ;

3) консолидации баз данных.

Есть и другие возможности для применения данного метода, такие как создание временных коллекций и т.п.

Далее в главе приведено описание **метода определения структуры вложенных документов в базах данных NoSQL типа «ключ-документ»**:

Определение вложенных документов для двух отношений

Пусть:

1. даны два отношения реляционной модели, на основе которых строится схема не реляционной БД типа «ключ-документ»: T_1 и T_2:

$$T_1\{T_{11}, T_{12}, ..., T_{1k}\}$$

$$T_2\{T_{21}, T_{21}, ..., T_{2n}\}$$

где $T_{i,j}$ – это j-е поле i-го отношения, k – число полей в отношении T_1, n – число полей в отношении T_2.

2. по методу определения коллекций в БД типа ключ-документ получена некоторая коллекция:

$$Q\{T_{11},...,T_{1m},T_{21},...,T_{2r}\}, m \le k, r \le n$$

3. отношения T_1 и T_2 имеют связи типа $1-\infty$:

4. в отношениях определены ключи, принадлежащие данной коллекции:

$$T_1 : T_{11}, T_{12},..., T_{1s1}, \text{ где } 1 \le s1 \le m$$

$$T_2 : T_{21}, T_{22},..., T_{2s2}, \text{ где } 1 \le s2 \le r$$

Выходные данные метода: новая структура коллекции Q т.е. $Q' \approx Q$

Решение: рассмотрим возможные варианты запросов к коллекции Q:

a) $S_1\{T_{11},...,T_{1s1},...,T_{1i}\}$ – запрос, который обращается только к атрибутам одного отношения, например T_1;

b) $S_2\{T_{11},...,T_{1s1},...,T_{1i},T_{21},...,T_{2s2},...,T_{2j}\}$ – запрос, который обращается к атрибутам обоих отношений T_1 и T_2.

Случай a). Так как предполагается, что схема РБД с отношениями T_1 и T_2 изначально находится в нормальной форме, как минимум Бойса-Кодда или 4NF, то аномалия избыточности в этой схеме сведена к минимуму и вложенных документов в данной коллекции быть не может.

Случай b). Связь $1-\infty$ подразумевает, что одному кортежу из отношения T_1 может соответствовать множество кортежей из отношения T_2. Отсюда следует, что для того, чтобы избежать хранения избыточных данных в коллекции Q, необходимо все атрибуты отношения T_2, участвующие в запросе S_2, выделить во вложенный документ. При этом возможны два случая:

b1): таких запросов как S_2 всего один для коллекции Q. В этом случае новая структура коллекции Q примет вид: $Q'\{T_{11},...,T_{1s1},...,T_{1m},T'_2 : \{T_{21},...,T_{2s2},...,T_{2j}\},...,T_{2r}\}$, где T'_2 – это имя нового ключа для вложенного документа.

b2): таких запросов как S_2 более одного для коллекции Q. В этом случае необходимо найти множество T'_2, которое будет объединением всех множеств атрибутов отношения T_2, входящие в запросы типа S_2. Формализовано это будет так: пусть есть совокупность запросов типа S_2 к коллекции Q:

$$S_{21}\{T_{11},...,T_{1s1},T_{21},...,T_{2s2},...,T_{2j1}\};$$

$$\dots$$

$$S_{2t}\{T_{11},...,T_{1st},...,T_{1it},T_{21},...,T_{2s2},...,T_{2jt}\}$$

тогда вложенный документ будет состоять из атрибутов множества:

$$T'_2 = \{T_{21},...,T_{2s2},...,T_{2j1}\} \cup ... \cup \{T_{21},...,T_{2s2},...,_{2jt}\} = \{T_{21},...,T_{2s2},...,T_{2j1},...,T_{2jt}\}$$

В этом случае новая структура коллекции Q примет вид:

$$Q'\{T_{11},...,T_{1s1},...,T_{1m},T'_2 : \{T_{21},...,T_{2s2},...,T_{2j1},...,T_{2jt}\},...,T_{2r}\}$$

где T'_2 – это имя нового ключа для вложенного документа.

Примечание 2. Допустим, что после отделения всех атрибутов отношения T_2, участвующих в запросах типа S_2, в новое множество T'_2, остались еще атрибуты отношения T_2, которые в запросе S_2 не участвуют. Назовем это множество $T''_2 = T_2 - T'_2$. Это может означать, что-либо к атрибутам множества T''_2 обращается отдельный запрос типа а) и опять же это означает, что коллекция построена не оптимально. Либо, если коллекция построена оптимально, существует запрос, который обращается одновременно к атрибутам T'_2 и T''_2. Последнее означает, что отделить атрибуты T'_2 в отдельный

документ нельзя, т.к. это усложнит запросы типа а), обращающиеся одновременно к атрибутам T'_2 и T''_2. Поэтому, во вложенный документ можно выделить полностью все атрибуты отношения T_2, входящие в коллекцию Q. Сформулируем унифицированное для всех случаев правило создания вложенного документа с учетом примечания 2.

Правило 1. Пусть есть совокупность запросов типа S_2 к коллекции Q:

$$S_{21}\{T_{11},...,T_{1s1},...,T_{1i1},T_{21},...,T_{2s2},...,T_{2j1}\};$$
$$...$$
$$S_{2t}\{T_{11},...,T_{1st},...,T_{1it},T_{21},...,T_{2s2},...,T_{2jt}\}$$

и совокупность запросов типа S_1 к атрибутам отношения T_2 из коллекции Q:

$$S_{11}\{...,T_{2i1},...\};$$
$$...$$
$$S_{1r}\{...,T_{2ir},...\}$$

Пусть $T'_2(S_2)$ – это множество всех атрибутов, участвующих в запросах типа S_2:

$$T'_2(S_2)=\{T_{21},...,T_{2s2},...,T_{2j1}\}\cup...\cup\{T_{21},...,T_{2s2},...,T_{2jt}\}=\{T_{21},...,T_{2s2},...,T_{2j1},...,T_{2jt}\}$$

Пусть $T'_2(S_{1i})$ – это множество всех атрибутов, участвующих в i-м запросе типа S_1. Тогда – это множество тех атрибутов, которые необходимо присоединить ко вложенному документу, т.к. они участвуют в запросе с некоторыми другими атрибутами отношения T_2, которые уже включены во вложенный документ.

Т.к. запросов типа S_1 может быть некоторое количество, то необходимо найти объединение таких атрибутов. Пусть всего запросов S_1 количество равное V_s, тогда вложенный документ будет состоять из атрибутов множества:

$$T'_2=T'_2(S_2)\cup\bigcup_{i=1}^{V_s}(S_{1i}\cap(S_{1i}-T'_2(S_2)))$$
$$T''_2=T_2-T'_2$$

В этом случае новая структура коллекции Q примет вид:

$$Q'\{T_{11},...,T_{1s1},...,T_{1m},T'_2:\{T'_2(S_2)\cup\bigcup_{i-1}^{V_s}(S_{1i}\cap(S_{1i}-T'_2(S_2)))\},T''_2\} \quad (8)$$

где T'_2 — это имя нового ключа для вложенного документа.

Примечание 3. В данном методе была рассмотрена только связь типа $1-\infty$. Если схема реляционной базы данных была изначально приведена к нормальной форме BCNF, 3NF или 4NF, то других связей между отношениями T_1 и T_2 быть не может. Но, если по каким-либо причинам, например проведенной денормализации схемы РБД, это связи существуют, то:

- для связи типа 1-1 вложенных документов быть не может, ввиду однозначности соответствия каждому кортежу из отношения T_1 единственного кортежа из отношения T_2.

- для связи типа $\infty-\infty$ вложенные документы строятся по тому же принципу, что и для связи $1-\infty$.

Определение вложенных документов для трех отношений с одним главным отношением и двумя подчиненными

Пусть:

1. даны три отношения реляционной модели, на основе которых строится схема не реляционной БД типа ключ-документ: T_1, T_2 и T_3:

$$T_1\{T_{11},T_{12},...,T_{1k}\}$$
$$T_2\{T_{21},T_{22},...,T_{2n}\}$$
$$T_3\{T_{31},T_{32},...,T_{3m}\}$$

где T_{ij} – это j-е поле i-го отношения, k – число полей в отношении T_1, n – число полей в отношении T_2, m – число полей в отношении T_3.

2. по методу определения коллекций в БД типа ключ-документ получена некоторая коллекция:

$$Q\{T_{11},...,T_{1k'},T_{21},...,T_{2n'},T_{31},...,T_{3m'}\}, k' \le k, n' \le n, m' \le m$$

3. отношения T_1, T_2 и T_3 имеют связи типа $1-\infty$: или более схематично на рис. 2.а.

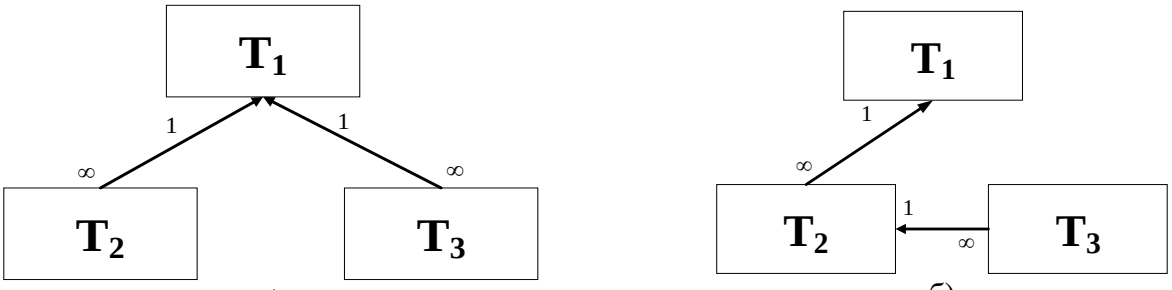

Рис. 2. Связи между отношениями T_1, T_2 и T_3

4. в отношениях определены ключи, принадлежащие данной коллекции:

$$T_1 : T_{11},T_{12},...,T_{1s1}, \text{ где } 1 \le s1 \le k'$$
$$T_2 : T_{21},T_{22},...,T_{2s2}, \text{ где } 1 \le s2 \le n'$$
$$T_3 : T_{31},T_{32},...,T_{3s3}, \text{ где } 1 \le s3 \le m'$$

Выходные данные. Новая структура коллекции Q, т.е. $Q' \approx Q$

Решение: рассмотрим возможные варианты запросов к коллекции Q:

a) $S_1\{T_{11},...,T_{1s1},...,T_{1i}\}$ - запрос, который обращается только к атрибутам одного отношения, например T_1;

b) $S_2\{T_{11},...,T_{1s1},...,T_{1i},T_{21},...,T_{2s2},...,T_{2j}\}$ - запрос, который обращается к атрибутам двух отношений T_1, T_2 или T_1, T_3.

c) $S_3\{T_{11},...,T_{1s1},...,T_{1i},T_{21},...,T_{2s2},...,T_{2j},T_{31},...,T_{3s3},...,T_{3r}\}$ - запрос, который обращается к атрибутам всех трех отношений T_1, T_2 и T_3.

d) $S_4\{T_{21},...,T_{2s2},...,T_{2j},T_{31},...,T_{3s3},...,T_{3r}\}$ - запрос, который обращается к атрибутам двух отношений T_2 и T_3.

Случай a). Этот случай рассмотрен выше при определении вложенных документов для коллекции, построенной из атрибутов двух отношений и показано, что *вложенных документов в данной коллекции быть не может*.

Случай b). Этот случай рассмотрен выше при определении вложенных документов для коллекции, построенной из атрибутов двух отношений по формуле (8) и показано, что новая структура коллекции Q примет вид:

$$Q'\left\{T_{11},...,T_{1s1},...,T_{1m},T'_2 : \left\{T'_2(S_2) \cup \bigcup_{i=1}^{V_s}(S_{1i} \cap (S_{1i} - T'_2(S_2)))\right\}, T''_2\right\}$$

где T'_2 – это имя нового ключа для вложенного документа, атрибуты $T'_2(S_2)$ – это все атрибуты отношения T_2, входящие в запросы типа S_2 к коллекции Q, V_s – количество запросов типа S_1, T''_2 – множество атрибутов, которые не вошли в T''_2 из всех атрибутов отношения T_2, входящих в коллекцию Q.

Примечание 4. Если коллекция Q была построена в соответствии с методом определения коллекция к БД типа ключ-документ, то множество $T''_2 = \varnothing$.

Случай c). Этот случай является обобщением случая b) с двух отношений со связью $1-\infty$ на две пары отношений со связью $1-\infty$. В данном случае новая структура коллекции Q примет вид:

$$Q'\begin{Bmatrix} T_{11},...,T_{1m},T'_2:\left\{T'_2(S_3)\cup\bigcup_{i=1}^{V_s}(S_{1i}\cap(S_{1i}-T'_2(S_3)))\right\},T''_2, \\ T'_3:\left\{T'_3(S_3)\cup\bigcup_{i=1}^{V_s}(S_{1i}\cap(S_{1i}-T'_3(S_3)))\right\},T''_3 \end{Bmatrix} \quad (9)$$

где T'_2, T'_3 – это имена новых ключей для вложенных документов; атрибуты $T'_2(S_3)$ – это все атрибуты отношения T_2, входящие в запросы типа S_3 к коллекции Q; атрибуты $T'_3(S_3)$ – это все атрибуты отношения T_3, входящие в запросы типа S_3 к коллекции Q, V_s – количество запросов типа S_1, T''_2 – множество атрибутов, которые не вошли в T'_2 из всех атрибутов отношения T_2, входящих в коллекцию Q, T''_3 – множество атрибутов, которые не вошли в T'_3 из всех атрибутов отношения T_3, входящих в коллекцию Q.

Случай d). Т.к. запрос к атрибутам отношений T_2 и T_3 может быть выполнен только через атрибуты отношения T_1, то этот случай в результате сводится к случаю c).

Определение вложенных документов для трех отношений с двумя главными отношениями и двумя подчиненными

Входные данные. Пусть: 1. даны три отношения реляционной модели, на основе которых строится схема не реляционной БД типа ключ-документ T_1, T_2 и T_3:

$$T_1\{T_{11},T_{12},...,T_{1k}\}$$
$$T_2\{T_{21},T_{22},...,T_{2n}\}$$
$$T_3\{T_{31},T_{32},...,T_{3m}\}$$

где T_{ij} – это j-е поле i-го отношения, k – число полей в отношении T_1, n – число полей в отношении T_2, m – число полей в отношении T_3.

2. по методу определения коллекций в БД типа ключ-документ получена некоторая коллекция:

$$Q\{T_{11},...,T_{1k'},T_{21},...,T_{2n'},T_{31},...,T_{3m'}\},k'\le k,n'\le n,m'\le m$$

3. отношения T_1, T_2 и T_3 имеют связи типа $1-\infty$: или более схематично на рис. 2.б.

4. в отношениях определены ключи, принадлежащие данной коллекции:

$$T_1:T_{11},T_{12},...,T_{1s1},\text{ где } 1\le s1\le k'$$
$$T_2:T_{21},T_{22},...,T_{2s2},\text{ где } 1\le s2\le n'$$
$$T_3:T_{31},T_{32},...,T_{3s3},\text{ где } 1\le s3\le m'$$

Выходные данные: Новая структура коллекции Q, т.е. $Q'\approx Q$.

Решение: рассмотрим возможные варианты запросов к коллекции Q:

a) $S_1\{T_{11},...,T_{1s1},...,T_{1i}\}$ - запрос, который обращается только к атрибутам одного отношения, например T_1;

b) $S_2\{T_{11},...,T_{1s1},...,T_{1i},T_{21},...,T_{2s2},...,T_{2j}\}$ - запрос, который обращается к атрибутам двух отношений T_1, T_2 или T_2, T_3.

c) $S_3\{T_{11},...,T_{1s1},...,T_{1i},T_{21},...,T_{2s2},...,T_{2j},T_{31},...,T_{3s3},...,T_{3r}\}$ - запрос, который обращается к атрибутам всех трех отношений T_1, T_2 и T_3.

d) $S_4\{T_{11},...,T_{1s1},...,T_{1i},T_{31},...,T_{3s3},...,T_{3r}\}$ - запрос, который обращается к атрибутам двух отношений T_1 и T_3.

Случай a). Этот случай рассмотрен выше при определении вложенных документов для коллекции, построенной из атрибутов двух отношений и показано, что *вложенных документов в данной коллекции быть не может*.

Случай b). Этот случай рассмотрен выше при определении вложенных документов для коллекции, построенной из атрибутов двух отношений и показано, что новая структура коллекции Q примет вид (9).

Случай c). Этот случай является обобщением случая b) с двух отношений на три отношения. В данном случае новая структура коллекции Q примет вид:

$$Q'\left\{T_{11},...,T_{1m},T'_2:\left\{\begin{array}{l}T'_2(S_3)\cup\bigcup_{i=1}^{V_s}\left(S_{1i}\cap\left(S_{1i}-T'_2(S_3)\right)\right),\\ T'_3:\left\{T'_3(S_3)\cup\bigcup_{i=1}^{V_s}\left(S_{1i}\cap\left(S_{1i}-T'_3(S_3)\right)\right)\right\}\end{array}\right\},T''_2,T''_3\right\} \quad (10)$$

где T'_2, T'_3 – это имена новых ключей для вложенных документов; атрибуты $T'_2(S_3)$ – это все атрибуты отношения T_2, входящие в запросы типа S_3 к коллекции Q; атрибуты $T'_3(S_3)$ – это все атрибуты отношения T_3, входящие в запросы типа S_3 к коллекции Q, V_s – количество запросов типа S_1, T''_2 – множество атрибутов, которые не вошли в T'_2 из всех атрибутов отношения T_2, входящих в коллекцию Q, T''_3 – множество атрибутов, которые не вошли в T'_3 из всех атрибутов отношения T_3, входящих в коллекцию Q.

Случай c). Т.к. запрос к атрибутам отношений T_1 и T_3 может быть выполнен только через атрибуты отношения T_2, то этот случай в результате сводится к случаю b).

Примечание 5. Если коллекция Q построена более чем для 3-х отношений, то вложенные документы строятся также как и для трех отношений.

Методика определения структуры вложенных документов в БД типа «ключ-документ»

Прежде чем перейти к методике применения методов определения коллекций с вложенными документами в БД типа ключ-документ, необходимо сделать несколько примечаний.

Примечание 6. Если в коллекции Q из атрибутов, относящихся к трем отношениям нет запросов, обращающихся к атрибутам всех трех отношений, то такую коллекцию лучше разбить на две коллекции.

Рассмотрим отдельно два случая.

Случай 1. Три отношения с одним главным отношением и двумя подчиненными (см. рис. 2.а).

Если к коллекции Q нет запроса, который обращается к атрибутам всех трех отношений T_1, T_2 и T_3:

$$S_3\{T_{11},...,T_{1s1},...,T_{1i},T_{21},...,T_{2s2},...,T_{2j},T_{31},...,T_{3s3},...,T_{3r}\}$$

но, есть запросы, которые обращаются к атрибутам двух отношений T_1, T_2 или T_2, T_3:

$$S_{21}\{T_{11},...,T_{1s1},...,T_{1i},T_{21},...,T_{2s2},...,T_{2j}\}$$
$$S_{22}\{T_{11},...,T_{1s1},...,T_{1i},T_{31},...,T_{3s3},...,T_{3r},...,T_{2t}\}$$

то вместо формулы (2.9) следует коллекцию Q разбить на две коллекции со вложенными документами в соответствии с формулой (2.8):

$$\begin{gathered} Q'_1\left\{T_{11},...,T_{1s1},...,T_{1m},T'_2:\left\{T'_2(S_{21})\cup\bigcup_{i=1}^{V_s}\left(S_{1i}\cap(S_{1i}-T'_2(S_{21}))\right)\right\},T''_2\right\} \\ Q'_2\left\{T_{11},...,T_{1s1},...,T_{1m},T'_2:\left\{T'_3(S_{22})\cup\bigcup_{i=1}^{V_s}\left(S_{2i}\cap(S_{1i}-T'_3(S_{22}))\right)\right\},T''_3\right\} \end{gathered} \quad (11)$$

где T'_2, T'_3 – это имена новых ключей для вложенных документов; атрибуты $T'_2(S_3)$ – это все атрибуты отношения T_2, входящие в запросы типа S_3 к коллекции Q; атрибуты $T'_3(S_3)$ – это все атрибуты отношения T_3, входящие в запросы типа S_3 к коллекции Q, V_s – количество запросов типа S_1, T''_2 – множество атрибутов, которые не вошли в T'_2 из всех атрибутов отношения T_2, входящих в коллекцию Q, T''_3 – множество атрибутов, которые не вошли в T'_3 из всех атрибутов отношения T_3, входящих в коллекцию Q.

Случай 2. Три отношения с двумя главными отношениями и двумя подчиненными (см. рис. 2.б).

Если к коллекции Q нет запроса, который обращается к атрибутам всех трех отношений T_1, T_2 и T_3:

$$S_3\{T_{11},...,T_{1s1},...,T_{1i},T_{21},...,T_{2s2},...,T_{2j},T_{31},...,T_{3s3},...,T_{3r}\}$$

но, есть запросы, которые обращаются к атрибутам двух отношений T_1, T_2 или T_2, T_3:

$$S_{21}\{T_{11},...,T_{1s1},...,T_{1i},T_{21},...,T_{2s2},...,T_{2j}\}$$
$$S_{22}\{T_{11},...,T_{1s1},...,T_{1i},T_{31},...,T_{3s3},...,T_{3r},...,T_{2t}\}$$

то вместо формулы (10) следует коллекцию Q разбить на две коллекции со вложенными документами в соответствии с формулой (2.8):

$$\begin{gathered} Q'_1\left\{T_{11},...,T_{1s1},...,T_{1m},T'_2:\left\{T'_2(S_{21})\cup\bigcup_{i=1}^{V_s}\left(S_{1i}\cap(S_{1i}-T'_2(S_{21}))\right)\right\},T''_2\right\} \\ Q'_2\left\{T_{11},...,T_{1s1},...,T_{1m},T'_2:\left\{T'_3(S_{22})\cup\bigcup_{i=1}^{V_s}\left(S_{2i}\cap(S_{1i}-T'_3(S_{22}))\right)\right\},T''_3\right\} \end{gathered} \quad (12)$$

где T'_2, T'_3 – это имена новых ключей для вложенных документов; атрибуты $T'_2(S_3)$ – это все атрибуты отношения T_2, входящие в запросы типа S_3 к коллекции Q; атрибуты $T'_3(S_3)$ – это все атрибуты отношения T_3, входящие в запросы типа S_3 к коллекции Q, V_s – количество запросов типа S_1, T''_2 – множество атрибутов, которые не вошли в T'_2 из всех атрибутов отношения T_2, входящих в коллекцию Q, T''_3 – множество атрибутов, которые не вошли в T'_3 из всех атрибутов отношения T_3, входящих в коллекцию Q.

С учетом формул (8) – (12) методика определения коллекций со вложенными документами для БД типа ключ-документ представлена на рис. 4.

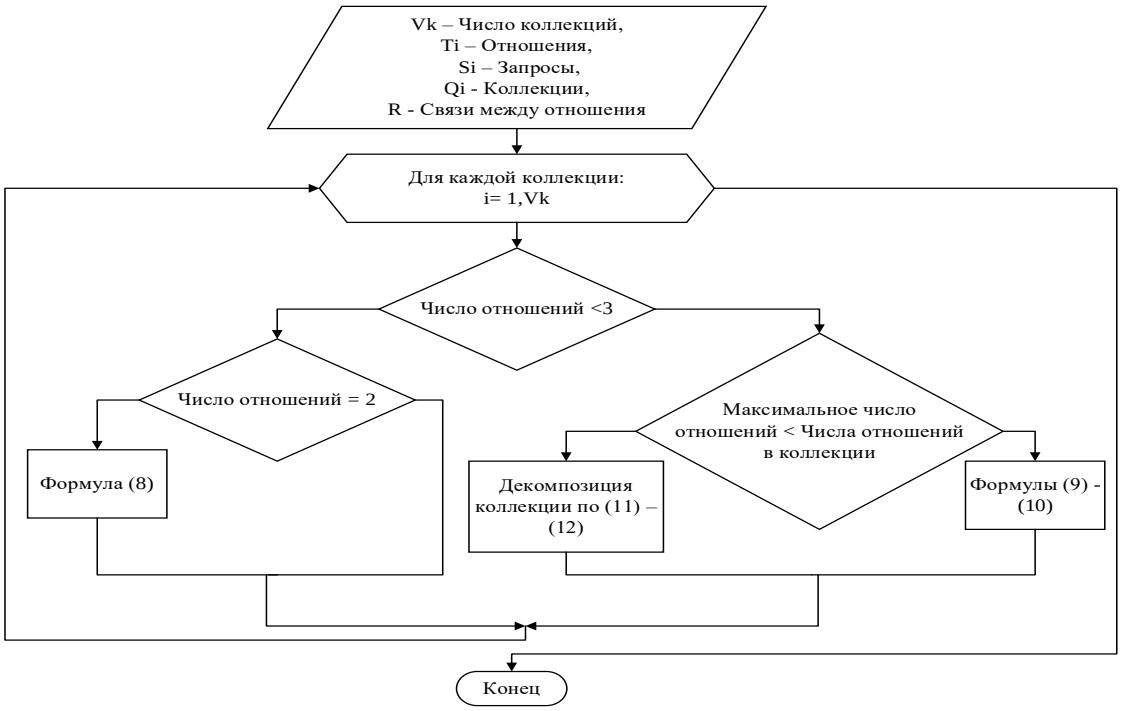

Рис. 4. Методика определения вложенных документов

В конце главы представлен разработанный метод оптимизации структуры распределенной базы данных NoSQL типа «ключ-документ». Для распределенной базы данных был применен подход, основанный на информационных графах, для исследования зависимостей подзапросов от местоположения и доступности данных в соответствии с заданной схемой распределенной БД и информацией о шардах и репликах.

Метод построения информационного графа запроса для распределенной реляционной БД:

1) Каждой таблице РБД поставить в соответствие входную вершину информационного графа.
2) Каждому элементарному подзапросу поставить в соответствие вершину информационного графа.
3) Соединить вершины информационного графа направленными ребрами в соответствии с правилом: если подзапрос A получает данные от таблицы (или подзапроса) B, то соединить вершины A и B направленным ребром от B к A.
4) Адаптация графа под горизонтальный шардинг для случая, когда таблица T распределенной БД разделена на k частей, хранимых на отдельных шардах:
   a) создать k входных вершин графа, каждая из которых будет соответствовать определенной части таблицы на определенном шарде: T_i, $i = 1...k$.
   b) если таблица T была связана ребром с запросом Q, то создать k вершин графа, каждая из которых будет соответствовать запросу Q, выполняемому для определенной части таблицы на определенном шарде: Q_i, $i = 1...k$.

   Далее возможен один из трех вариантов:

   1) Вариант 1.
      A. все вершины, которые были до этого соединены с вершиной Q, соединить с вершинами Q_i, $i = 1...k$;
      B. соединить каждую вершину, соответствующую части таблицы T, с вершиной, соответствующей запросу Q, и выполняемому для данной части таблицы T, т.е. T_i соединить с Q_i (пример этого варианта представлен на рис.5.b);
      C. если вершина Q не являлась выходной вершиной графа, то: создать вершину Q', которая будет соответствовать запросу по агрегации данных из запросов Q_i над частями таблицы T_i.

2) Вариант 2.

D. соединить каждую вершину, соответствующую части таблицы T, с вершиной, соответствующей запросу Q, и выполняемому для данной части таблицы T, т.е. T_i соединить с Q_i (пример этого варианта представлен на рис.5.c);

E. если вершина Q не являлась выходной вершиной графа, то: создать вершину Q', которая будет соответствовать запросу по агрегации данных из запросов Q_i над частями таблицы T_i;

F. если есть вершины (кроме T), которые были до этого соединены с вершиной Q, то создать узел Q'' и соединить все вершину и вершину Q' с вершиной Q'';

3) Вариант 3.

A. все вершины, которые были до этого соединены с вершиной Q, соединить с вершинами Q_i, $i = 1...k$;

B. соединить каждую вершину, соответствующую части таблицы T, с вершиной, соответствующей запросу Q, и выполняемому для данной части таблицы T, т.е. T_i соединить с Q_i (пример этого варианта представлен на рис.5.d);

C. если вершина Q не являлась выходной вершиной графа b была связана ребром с запросом Q_1, то создать k вершин графа, каждая из которых будет соответствовать запросу Q_1, выполняемому для определенной части запроса Q_i на определенном шарде: Q_{1i}, $i = 1...k$.

D. соединить каждую вершину, соответствующую части запроса Q, с вершиной, соответствующей части запроса Q_1, т.е. соединить Q_i с Q_{1i};

E. если вершина Q_1 не являлась выходной вершиной графа, то: создать вершину Q'', которая будет соответствовать запросу по агрегации данных из запросов Q_{1i}.

c) если в графе есть еще таблицы, которые также распределены по шардам, то шаг b) повторяется для каждой такой таблицы.

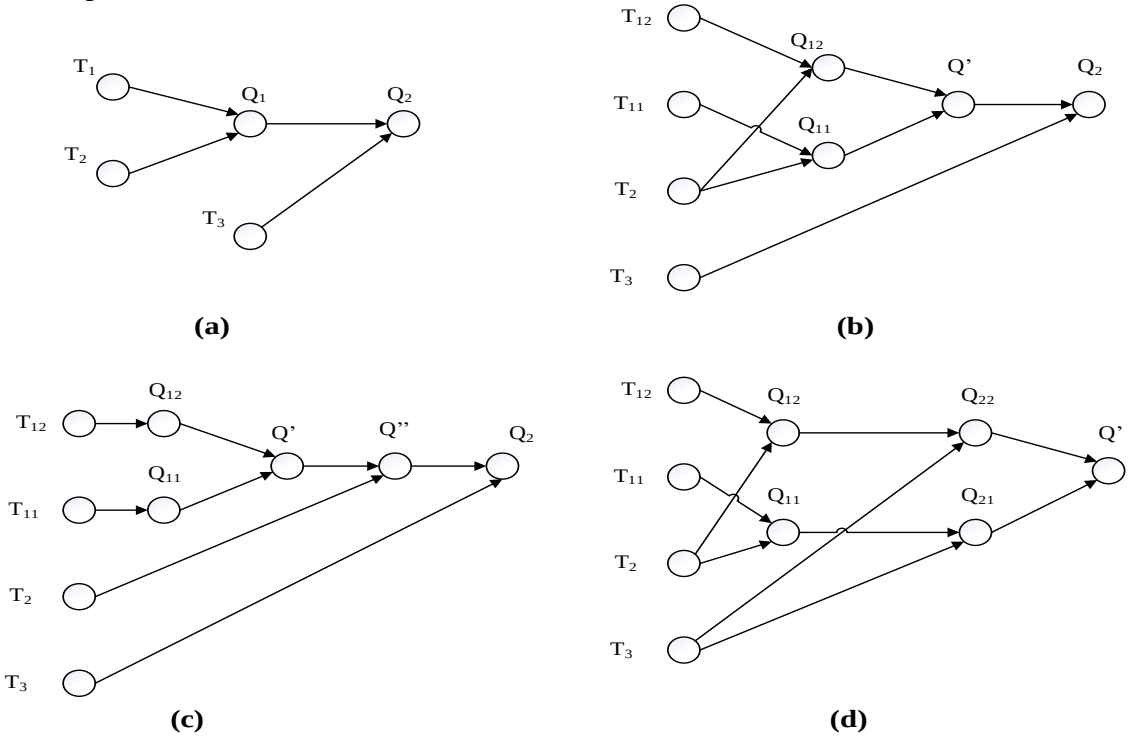

Рис. 5. Способы представление запроса к распределенной базе данных с помощью информационного графа

Метод построения схемы распределенной базы данных NoSQL типа «ключ-документ»:

1. Построить информационный граф с учетом операций шардинга и репликаций.

2. Привести информационный граф к параллельной форме, минимизирующей передачу данных с шарда на шард с помощью метода оптимизации параллельных алгоритмов по числу коммуникаций.

3. Выполнить действие 1-2 для всех запросов, структура которых должна быть учтена при оптимизации схемы документной базы данных с целью ускорения выполнения этих запросов.

4. На основе метаданных о таблицах, полях, запросах и связях с помощью методов оптимизации схемы документной базы данных с вложенными или без вложенных документов сформировать структуру коллекций документов.

5. Усовершенствовать структуру документов с учетом графов информационных зависимостей запросов, минимизировав число операций объединения данных.

В третьей главе приведено описание трансляции запросов формата SQL в формат MongoDB. Предлагаемый нами метод автоматической трансляции запросов из формата SQL в формат MongoDB с учетом структуры баз данных MySQL и MongoDB заключается в следующем:

Входными данным являются: код запроса к базе данных MySQL, метаданные о структуре баз данных MySQL и MongoDB.

Выходные данные: код запроса к базе данных MongoDB.

Шаг 1. Парсинг запроса в формате MySQL

1.1. Парсинг запроса в формате MySQL по формальной грамматике языка SQL-запросов. Результатами парсинга являются токены (слова).

1.2. Формирование списков не ключевых токенов из общего списка токенов, получаемых из парсинга запроса в формате MySQL.

Шаг 2. Формирование словаря предложений запроса MongoDB.

Шаг 3. Переформирование предложений словаря с учетом структуры базы данных MongoDB.

3.1. Определение структуры базы данных MongoDB.

3.2. Переформирование предложений словаря с учетом структуры базы данных MongoDB.

Шаг 4. Синтез выходного запроса из предложений словаря с учетом структуры базы данных MongoDB и на основе синтаксиса функции «aggregate» в MongoDB.

В четвертой главе приведено подробное описание разработанных модулей, сведение баз данных для тестирования и тестирование разработанных методов. Для тестирования разработанных метода были спроектированы разные базы данных с разными структурам и использована база данных «TPC - H». Также были сгенерированы разные объемы данных для баз данных.

Тестирование метода определения структуры коллекций для нераспределенных баз данных типа ключ-документ с учетом запросов было выполнено с базами «Telecommunication_business» и «TPC - H» с разными объёмами данных. В тестирование рассмотрены три варианты структуры базы данных: каждой таблице в реляционной базе данных поставить в соответствие отдельную коллекцию документов в MongoDB (**Вариант 1**), из всех таблиц реляционной базы данных сделать одну коллекцию документов в MongoDB(**Вариант 2**) и создать такой набор коллекций документов в MongoDB, чтобы они наиболее полно подходили под выполняемые запросы (**Вариант 3**). К базе данных «Telecommunication_business» были выполнены 9 запросы и к базе данных «TPC - H» выполнены запросы «Q1-Q5». На рис.6 приведены графики времени выполнения запросов «Q1-Q5» к базе данных «TPC - H» с разными структурами и разными объемами. Результаты тестирования показали разработанный метод является наиболее оптимальным с точки зрения времени выполнения запроса.

Для тестирования метод определения структуры вложенных документов в БД типа ключ-документ использованы проектированные базы данных «Test1 – Test5» и база данных «TPC - H». На рис.7. приведен результат тестирования с базой данных «TPC - H». Для каждой базы рассмотрены две структуры: без вложенных документов и с вложенными документами. Результаты тестирования со спроектированными базами данных и базой данных «TPC - H» показывают эффективность применения разработанного метода определения структуры вложенных документов с учетом связей между объектами в базе данных и структуры запросов, которые к ним выполняются

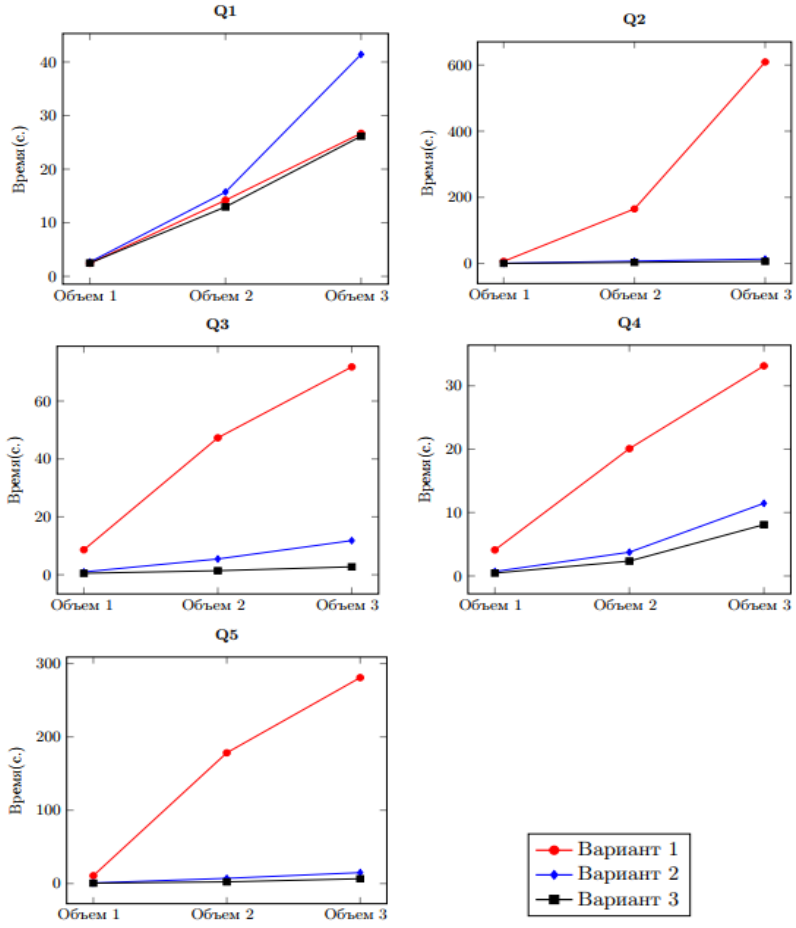

Рис. 6. - Графики времени выполнения запросов «Q1-Q5» к базе данных «TPC - H» с разными структурами и разными объемами

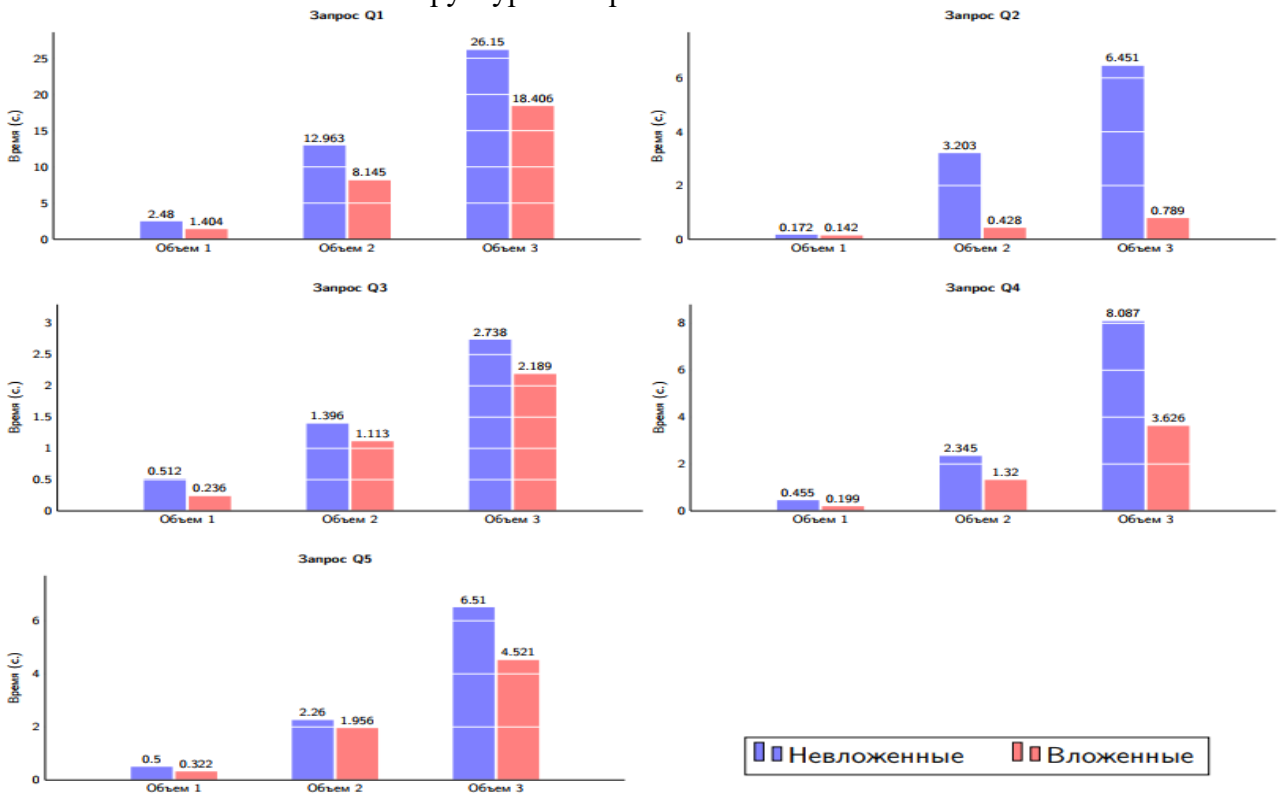

Рис. 7. - Графики времени выполнения запросов «Q1-Q5» к базе данных «TPC - H» с разными структурами и разными объемами

В тестировании методов оптимизации структуры распределенных баз данных NoSQL рассмотрены три варианты: централизованная база данных, распределенная база данных с наличием шардинга, но при отсутствии репликаций и распределенная база данных с наличием шардов и реплик. Для двух варианта распределённых баз данных, исходный запрос разделен на подзапросы и в зависимости от структур БД оптимизирован план выполнения подзапросов за счет параллельного выполнения. Тестирование выполнено на MySQL и MongoDB с разными объемами баз данных. Результат тестирования (рис. 8) показал, что что максимальное распараллеливание запроса в БД перед объединением результата программой работает быстрее всего. Кроме времени выполнения запросов, в испытании были также протестированы объем памяти при выполнении запросов для всех вариантов.

Рис. 8. - Время выполнения запроса для разных вариантов схемы БД MongoDB и объема данных

В тестировании исследовались три важных параметра: физическая память, задействованная память ОЗУ и максимальная ОЗУ при выполнении запросов с разными вариантами и разными объемами. Из результатов тестирования (рис. 9) видно, что вариант «распределенная база данных с шардированием и репликацией» использует наибольший объем памяти по сравнению с остальными вариантами. Ускоренный вариант выполнения требует большего объема памяти из-за хранения дубликатов данных.

Рис. 9. График ОЗУ при выполнении запросов для разных объемов

В заключении сформированы основные результаты диссертационного исследования.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Проанализированы существующие подходы к трансляции баз данных из одного формата в другой, а также подходы к оптимизации структуры NOSQL БД и оптимизации запросов к ним.

2. Разработаны методы оптимизации структуры базы данных NoSQL для модели «ключ-документ», основанные на теории множеств и позволяющие автоматизировать процесс построения по заданной совокупности свойств объектов и с учетом их вхождений в запросы к базе данных определить оптимальную структуру базы данных NoSQL типов «ключ-документ» и структуру вложенных документов в этой БД. Методы учитывают тип связей между таблицами реляционной модели. В зависимости от типа связи выведены правила для создания структуры коллекции со вложенными документами.

3. Разработана методика преобразования реляционной базы данных в формат NoSQL типа «ключ-документ» в зависимости от исходных данных, таких как наличие информации о

существующей реляционной базе данных, наличия необходимости трансляции исходной базы данных в новый формат, принадлежности реляционной базы данных к нормальной форме.

4. Разработан метод построения схемы распределенной базы данных NoSQL типа «ключ-документ» с учетом информационного графа запроса и структуры распределенной РБД. Метод основан на теории графов и теории множеств и позволяет построить на шардах и репликах оптимальную по скорости выполнения запросов структуру коллекций.

5. Разработан метод трансляции запросов из формата SQL в формат MongoDB с учетом структуры базы данных. Разработанный метод состоит из четырех шагов: парсинг исходного запроса SQL, формирование словаря предложений запроса MongoDB, переформирование предложений словаря с учетом структуры базы данных MongoDB и синтез выходного запроса.

6. Созданы программные модули для тестирования разработанных методов и получены результаты тестирования на сгенерированных и существующих тестовых базах данных, которые показали эффективность разработанных методов.

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

Публикации в журналах, рекомендованных ВАК

1. Ха Ван Муон. Определение совокупности коллекций для баз данных типа ключ-документ по заданному набору свойств объектов и запросов к базе данных // Ха, В. М., Шичкина, Ю. А., Костичев, С. В.// Компьютерные инструменты в образовании. 2019 г., №3. – С. 15–28.

2. Ха Ван Муон. Автоматическая трансляция запросов из формата MySQL в формат MongoDB с учетом структуры базы данных // Ха, В. М// Известия Тульского государственного университета. Технические науки. 2021 г., №4. – С. 294–301.

Публикации, входящие в перечень изданий базы Scopus и Web of Science

3. Ha Van Muon. Creating Collections with Embedded Documents for Document Databases Taking into Account the Queries/ Shichkina, Y., Ha, M.. // Computation Vol. 8. No 2- 2020. (Scopus Q2, ESCI).

4. Ha Van Muon. Translating a distributed relational database to a document database/Ha, M., Shichkina, Y. //Data Science and Engineering (2022), Springer Berlin (Scopus Q2, ESCI).

5. Ха Ван Муон. Метод создания коллекций со вложенными документами для баз данных типа ключ-документ с учетом выполняемых запросов // Шичкина, Ю. А., Ха, В. М. // Труды СПИИРАН, 2020 г., №19(4). – С. 829–854. (Scopus Q3, ВАК).

6. Ha Van Muon. The Query Translation from MySQL to MongoDB Taking into Account the Structure of the Database / M. Ha, Y. Shichkina // 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus). – 2021. – P.383-386.

7. Ha Van Muon. An Approach to Translating a Database from MySQL to Cassandra/ M. Ha, Y. Shichkina // 2021 II International Conference on Neural Networks and Neurotechnologies (NeuroNT). – 2021. – P.1-4.

8. Ha Van Muon. Translation of Query for the Distributed Document Database. In: Gervasi O. et al. (eds) Computational Science and Its Applications / Ha M., Shichkina Y.A. // In: Gervasi O. et al. (eds) Computational Science and Its Applications – ICCSA 2021. ICCSA 2021. Lecture Notes in Computer Science.Vol 12956.– 2021. – P.396-405.

Свидетельства о государственной регистрации программы для ЭВМ

9. Ха Ван Муон, Шичкина Ю.А. ПрЭВМ, Document Database Collection Builder v.1.0 (DDCB/1.0), Свидетельство № 2020613492, дата гос. рег. 17.03.2020.

10. Ха Ван Муон, Шичкина Ю.А. ПрЭВМ, Translator of queries from a relational database format to a document database format v1.0, Свидетельство № 2021664094, дата гос. рег. 31.08.2021.

11. Ха Ван Муон, Шичкина Ю.А. ПрЭВМ, Трансляция тестовой базы данных из формата реляционной базы данных в документную (Translation of a test database from a relational database format to a document database), Свидетельство № 2021680392, дата гос. 09.12.2021.