Министерство образования Украины

Запорожский государственный университет

Факультет **математики и экономической кибернетики**

Кафедра **ММИТ**

ДИПЛОМНАЯ РАБОТА

на тему: АНАЛИЗ НЕПРОТИВОРЕЧИВОСТИ ФОРМАЛЬНОГО РАСШИРЕНИЯ ОПИСАТЕЛЬНОЙ ЧАСТИ РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ ПРИ ВВЕДЕНИИ СВОЙСТВА ВЫЧИСЛЯЕМОСТИ РЕЛЯЦИОННЫХ АТРИБУТОВ

Выполнил Кеберле Наталья Геннадьевна

ст. группы 8222-1
/шифр/ /подпись и дата/ /ф., и., о./

Руководитель доцент Ермолаев В.А..
/должность/ /подпись и дата/ /ф., и., о./

Нормоконтролер Бакурова А. В.
/подпись/ /ф., и., о./

Запорожье
1997

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 6
1 МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ 10
1.1 Реляционная модель данных 10
1.1.1 Реляционная структура данных 10
1.1.2 Манипулирование реляционными данными 12
1.1.3 Обеспечение целостности данных 18
1.2 Функциональные зависимости 20
1.3 Постановка задачи анализа непротиворечивости реляционной модели данных при введении свойства вычисляемости атрибутов 24
2 АНАЛИЗ НЕПРОТИВОРЕЧИВОСТИ ФОРМАЛЬНОГО РАСШИРЕНИЯ РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ ПРИ ВВЕДЕНИИ ВЫЧИСЛЯЕМЫХ АТРИБУТОВ 27
2.1 Булевы операции над отношениями с вычисляемыми атрибутами 27
2.2 Специальные реляционные операторы и отношения с вычисляемыми атрибутами 28
2.2.1 Выбор и его свойства 28
2.2.2 Проекция и ее свойства 29
2.2.3 Соединение отношений 30
2.2.4 Деление отношений 31
2.3 Алгебра B’ как система запросов 33
2.4 Выводы 34
3 ВЫЧИСЛЯЕМЫЕ ФУНКЦИОНАЛЬНЫЕ ЗАВИСИМОСТИ 35
3.1 Аксиомы вывода и вычисляемые F-зависимости. 35
3.2 Нормализация отношений 38
3.3 Оптимизация запросов и отношения с вычисляемыми атрибутами 46
3.4 Выводы 47
4 ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ ВЫЧИСЛЯЕМЫХ АТРИБУТОВ 49
4.1 Пример использования вычисляемых атрибутов 51
4.2 Выводы 57
ЗАКЛЮЧЕНИЕ 58
ПЕРЕЧЕНЬ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 59

ВВЕДЕНИЕ

В настоящее время существуют три основных типа систем баз данных, каждый из которых имеет свою сферу применения. Это иерархические, сетевые и реляционные подходы. Каждый из подходов имеет свои преимущества и недостатки. Сравнительный анализ этих подходов выполнен Дейтом [1].

Обобщая выводы многих исследователей моделей данных (Дейт [1], Ульман [2], Цикридзис,Лоховский [3]), выделим основные преимущества реляционного представления модели данных:

1.Простота использования, т.е. время на разработку программ и время формулирования запросов оптимально, что ведет к широкой применяемости и высокой производительности реляционных баз данных, особенно малых и средних размеров.

2. Из трех моделей указанных моделей только реляционная имеет в своей основе теоретические предложения, предопределяющие любые правильные реализации.

3. Небольшое число понятий в описательной части реляционной модели данных, причем различные понятия четко отделяются и не переплетаются сложными взаимосвязями.

Все данные рассматриваются как хранимые в таблицах, в которых каждая строка имеет один и тот же формат. Каждая строка в таблице представляет некоторый объект реального мира или соотношение между объектами.

4. Физический и логический уровни проектирования разделены, поэтому проблемы, связанные с разными уровнями, решаются по отдельности.

5. Эффективность реализации, т.е. простота перевода спецификаций концептуальной схемы в реализацию, эффективную с точки зрения необходимого производства и обработки запросов.

Реляционный подход к представлению модели данных, разработанный Коддом (Codd [4]) в 1970 г. получил огромную известность благодаря простоте основных идей и строгому формальному теоретическому фундаменту. Реляционная модель описывается в понятиях общей алгебры, математической логики, теории формальных систем и теории графов. Данное обстоятельство обеспечивает развитие формальных методов проектирования баз данных.

Со времени появления первых работ Кодда по реляционной модели данных круг задач по созданию баз данных расширялся, что повлекло за собой разработку новых методов моделирования реляционных баз данных и усовершенствования классической реляционной модели данных.

В 1975 г. в работе Schmid, Swenson [5] были изложены начала теории семантики данных в применении к реляционной модели данных. Предлагалось в рамках предметной области задачи рассматривать объекты, их связи и характеристики. Chen [6] в 1976 г. сформулировал метод концептуального моделирования баз данных в виде модели данных “Сущность - Связь”(Entity-Relationship Model, ER - модель), в которой предметная область задачи представляется двумя типами отношений - объектными и связными. Эти работы послужили теоретической базой для создания реляционных баз данных.

В настоящее время существует несколько основных направлений исследования реляционного подхода к проектированию баз данных.

Одно из направлений исследования реляционного подхода состоит в расширении сферы его применения путем усовершенствований самой реляционной теории, что позволяет использовать новые свойства и понятия для создания более полного описания предметной области.

Так, например, Кодд [7] вводит новые типы атрибутов - идентификатор, компонент, ссылка, что позволяет описывать связи между компонентами сложного объекта. Хаскин и Лори (Haskin,Lorie [8]) предлагают различные усовершенствования классической теории. Некоторые работы посвящены способам обхода одного из основных ограничений реляционной модели данных - неделимости (атомарности) отдельных значений данных. Макинучи (Makinouchi [9]) вводит понятие реляционно-значимых атрибутов, ослабляющих значимость атомарности данных. В работе Smith, Smith [10] предлагается расширение реляционной модели данных при допущении, что область значений (домен) атрибута может быть множеством имен других отношений. Такое расширение способствует обобщению представления объектов предметной области.

Существуют работы, посвященные исследованию представления неполной и неоднородной информации в базе данных, поскольку эта проблема стоит перед всякой моделью данных.

Концепция “Сущность-Связь” для реляционной модели данных является очень популярной. В связи с этим возникают вопросы объединения лучших качеств этой концепции с другими. Так, например, в работе Parent,Spaccapietra[11] предлагается расширение реляционной модели данных для описания объектов неограниченной степени сложности, объединяющее достоинства ER - концепции и принципов объектно-ориентированного моделирования (ERC+ -концепция, модель данных и алгебра).

Другое направление исследований связано со спецификой реляционной модели данных, а именно: сложный объект описывается в рамках реляционного подхода не одной таблицей, а набором таблиц. В этом случае возникают проблемы избыточности данных и

аномалий обновления информации. Разрабатываются пути приведения реляционной базы данных к оптимальному неизбыточному виду. В этой связи следует упомянуть работу Codd[12] , в которой были сформулированы понятия функциональной зависимости между данными, нормальной формы базы данных и предложены пути достижения этих нормальных форм. В работе Casey,Delobel [13] выясняется связь между функциональными зависимостями и булевскими функциями.

В работах многих авторов изучаются свойства функциональных и многозначных зависимостей и способы использования их при проектировании реляционных баз данных. (Armstrong [14,15], Beeri, Bernstein [16], Bernstein[17],)

Еще одно направление исследований посвящено практической реализации реляционного подхода к проектированию баз данных, поскольку на сегодняшний день не существует системы управления базами данных, полностью отвечающей требованиям реляционного подхода. На основании математического аппарата манипулирования отношениями в реляционной модели созданы мощные системы запросов, такие как SQL (Structured Query Language), QBE (Query By Example), основанные на реляционном исчислении, ISBL (Information System Base Language), основанный на реляционной алгебре, и др. Изучаются возможности оптимизации запросов, механизмы обеспечения целостности данных, построения и поддержки распределенных баз данных, наиболее экономичного представления баз данных.

В этой связи следует отметить работу Maier, Warren [18]. Авторы предложили для практических целей использовать отношения, названные вычисляемыми. Эти отношения создаются по мере надобности на основе базовых отношений (так называемых табличных, т.е. содержащих данные изначально). Такие вычисляемые отношения очень часто используются в системах запросов при создании сложных вложенных запросов. Отношение *q(Q)* называется вычисляемым, если *q(Q) =f(r,s,w,...),* где

r,s,w,... - базовые отношения

R,S,W,...- их схемы,

f - отображение, $f : dom(R) \times ... \times dom(W) \to dom(Q)$

Однако, поскольку любое отношение характеризуется набором атрибутов, то можно спуститься на более низкий уровень концептуального моделирования и рассматривать вычисляемые атрибуты. Значение такого атрибута является функцией от значений некоторого набора других атрибутов той же схемы.

Строгое определение вычисляемого атрибута было дано в работе Ермолаева [19].

Основное отличие вычисляемых атрибутов от вычисляемых отношений (кроме того, что любое отношение состоит из атрибутов, а атрибут - понятие атомарное) состоит в том, что вычисляемые атрибуты как свойства объектов предметной области закладываются в модель базы данных на этапе концептуального моделирования и к моменту начала функционирования базы данных их значения являются определенными.

В настоящей дипломной работе исследуется непротиворечивость реляционной модели данных при введении в ее описательную часть свойства вычисляемости реляционных атрибутов.

Понятие вычисляемого атрибута не входит в описание реляционной модели данных. Новизна данной работы состоит в проверке допустимости введения понятия вычисляемого атрибута в классическую реляционную модель данных, исследовании применимости реляционной алгебры ко множеству отношений с вычисляемыми атрибутами

Проводится анализ операций реляционной алгебры над множеством отношений с вычисляемыми атрибутами и предлагаются пути использования аппарата расширенной реляционной модели данных на практике.

Актуальность работы: техника вычисляемых атрибутов дает возможность реализации механизма косвенных множественных связей между множествами атрибутов различных отношений, позволяет автоматизировать процесс каскадного обновления отношений базы данных таким образом, чтобы не нарушалась целостность данных. Это, в свою очередь, дает возможность корректно описывать сложные объекты логической модели данных, оставаясь в рамках реляционного подхода и используя все его преимущества.

1 МАТЕМАТИЧЕСКАЯ ПОСТАНОВКА ЗАДАЧИ

1.1 Реляционная модель данных

Реляционная модель данных была предложена Коддом [4] в 1970г. Он показал, что набор “таблиц” (relations) может быть использован для моделирования взаимосвязей между объектами реального мира и для хранения данных об этих объектах. Такой подход к представлению данных в виде таблиц приобрел огромную популярность благодаря простоте основных идей.

В 1976 г. Чен (Chen[6]) предложил улучшение реляционной модели данных, которую он назвал “Entity - Relationship Model” (модель “Сущность” - “Связь”). В этой работе предлагалось представлять предметную область задачи двумя типами отношений:

Первый тип описывает некую сущность (entity) - объектное отношение.

Второй тип описывает связи между сущностями предметной области (relationships) - отношения связей.

Дадим неформальное определение терминов “сущность” и “связь”:

- **сущность** представляет собой любой отличимый объект с той степенью абстракции, какая требуется в данной задаче.

- **связь** представляет собой ассоциирование двух или более сущностей.

В общем случае связи бывают двух типов: “многие ко многим” и “многие к одной”. Связи типа “многие к одной “ называются **характеристическими**, связи типа “многие ко многим” называются **ассоциативными.**

Реляционная модель данных (РМД) - особый способ видения данных, т.е. предписание, определяющее, каким образом следует представлять данные и как ими манипулировать. Рассмотрим основные пункты, отличающие эту модель от других.

1.1.1 Реляционная структура данных

Наименьшая единица данных в РМД - отдельное значение данных. Такое значение рассматривается как **атомарное**, т.е. неразложимое на другие, когда дело касается данной задачи.

Домен - множество значений данных одного типа. Значения одного домена могут сравниваться между собой, значения из разных доменов - в общем случае - нет.

Атрибут - именованный домен, представляющий семантически значимый объект.

Имя атрибута - символическое имя данных, характеризующих отдельное свойства объекта из предметной области. В дальнейшем будет употреблятся слово **атрибут** с тем же смыслом.

Кортеж - множество значений , взятых по одному для каждого значения атрибута из схемы отношения.

t - кортеж - $< A_i : a_i >$, $i = \overline{1,n}$

Пользуясь терминологией Мейера [22], сформулируем следующие определения:

Определение 1.1

Схема отношения R - конечное множество имен атрибутов $\left\{A_1,...,A_n\right\}$

Каждому имени атрибута A_i ставится в соответствие множество D_i - конечное или бесконечное множество допустимых значений атрибута - домен: $dom(A_i) = D_i = \varnothing$, $i = \overline{1,n}$

Определение 1.2

Отношение *r* со схемой *R, r(R)* - конечное множество отображений, $\left\{t_1...t_p\right\}$, $t_j : R \to D$, где $D = D_1 \bigcup ... \bigcup D_n$ - домен схемы R, t_j - кортеж, $t_j \in r \Rightarrow t_j(A_i) \in D_i, i = \overline{1,n}, j = \overline{1,p}$;

p - кардинальное число отношения r, n - степень отношения r.

Т.о., отношение определяется математически как множество кортежей, и это множество является подмножеством декартова произведения фиксированного числа доменов. В терминах концептуального моделирования отношение- некоторый тип сущностей. Для того, чтобы использовать в качестве типов сущностей математические отношения, необходимо выполнений двух требований:

Требование 1. В каждом кортеже должно быть одно и тоже число атрибутов, значения каждого атрибута выбираются только из соответствующего атрибуту домена.

Требование 2. Поскольку множество не может иметь совпадающих элементов, а кортежи можно различать только по значениям их компонент, то никакие два различных кортежа не могут иметь полностью совпадающих значений атрибутов.

1.1.2 Манипулирование реляционными данными

Поскольку отношение было определено как множество кортежей, ко множествам можно применять аппарат исчисления предикатов, с тем, чтобы получить очень мощные языки запросов.

Кодд [20] предложил разновидность языка исчисления предикатов для работы с реляционными отношениями, которую назвал реляционным исчислением. В реляционном исчислении каждая формула определяет отношение, обладающее некоторым набором свойств и удовлетворяющая набору требований.

Для реализации этого исчисления он в работе [21] дал определения операций над реляционными отношениями и объединил их в реляционную алгебру. Выражения реляционной алгебры также определяют отношения, но эти отношения получают путем применения определенной последовательности операций.

Рассмотрим основные операции над отношениями в реляционной модели данных.

Прежде всего, любое отношение хранит информацию, изменяющуюся во времени. Для адекватного отражения состояния предметной области задачи в любой момент времени требуются операции обновления отношений.

Обновление отношений

Простейшие, в логическом смысле, операции над отношениями - операции над кортежами одного отношения. Эти операции позволяют отразить изменение отношения во времени. К ним относятся операции добавления, удаления и изменения кортежей.

Пусть $r(A_1 A_2 \ldots A_n)$ - отношение, $d_i \in D_i = dom(A_i), 1 \le i \le n$.

Операцию **добавления кортежа** можно записать в общем виде как:

$ADD\ (r;\ A_1=d_1,\ A_2=d_2,\ldots,\ A_n=d_n)$

Результат операции может быть отрицательным, если:

1. Добавляемый кортеж не соответствует схеме определенного отношения;
2. Некоторые d_i не соответствуют D_i.
3. Добавляемый кортеж совпадает с кортежем, уже находящемся в отношении.

Операцию **удаления кортежа** можно записать в виде:

$DEL\ (r;\ A_1=d_1,\ A_2=d_2,...,\ A_n=d_n)$

Операция **изменения кортежа** позволяет изменить часть кортежа.

Для отношения $r(A_1\ A_2...\ A_n)$, подмножества $\{C_1...C_p\}$ атрибутов r

$$CH(r;\ A_1=d_1,\ A_2=d_2,...,\ A_n=d_n\ ;\ C_1=e_1, C_2=e_2,...,C_p=e_p)$$

Операцию изменения можно выразить через операции добавления и удаления.

Если есть последовательность операций обновления, которые нужно применить к отношению r, то изменение порядка следования этих операций, очевидно, не всегда приводит к одинаковым результатам. Например, если последовательность состоит из операций удаления и добавления, или из операций добавления и изменения, то результат применения разных последовательностей этих операций будет разным, по причине невозможности удаления или изменения кортежа, еще не добавленного.

Традиционные теоретико-множественные операции: объединение, пересечение, разность, дополнение

Необходимым условием применения бинарных булевых операций есть *условие совместимости по объединению*, т.е. отношения должны быть одной и той же степени, и их i-е атрибуты должны быть связаны с одним и тем же доменом. Имена атрибутов могут и не совпадать.

Два отношения с одинаковыми схемами могут быть рассмотрены как подмножества всех возможных кортежей с этой схемой. К таким двум отношениям можно применить булевы операции.

Если отношения r и s имеют схему R, то существуют отношения $r \cup s$, $r \cap s$, $r - s$.

Определение 1.3

$r \cup s$ - множество всех кортежей, принадлежащих или r, или s - **объединение отношений.**

Определение 1.4

$r \cap s$ - множество всех кортежей, принадлежащих и r, и s - **пересечение отношений**.

Определение 1.5

$r - s$ - множество всех кортежей, принадлежащих r, но не принадлежащих s **разность отношений**.

Пусть *dom(R)*- множество всех возможных кортежей над атрибутами схемы R и их доменами.

Определение 1.6

Дополнение *r(R)* - множество, равное *dom(R) - r.*

Однако, если какой-либо атрибут $A \in R$ имеет бесконечный домен, то дополнение будет бесконечным и не является отношением в строгом понимании. Модифицированная версия дополнения, называемая **активным дополнением** , всегда дает отношение.

Определение 1.7

Если $r(A_1, A_2, \ldots, A_n)$- отношение, $d_i \in D_i = dom(A_i)$, то **активным доменом атрибута** A_i относительно r называется множество

$$adom(A_i, r) = \{d \in D_i : \exists\, t \in r : t(A_i) = d\}$$

Определение 1.8

Активное дополнение - множество $\tilde{r} = adom(A_i, r) - r$.

Множество отношений с данной схемой замкнуто относительно объединения, пересечения, разности и активного дополнения.

Специальные реляционные операторы:
выбор, проекция, тета-соединения, деление

Для выбора кортежей из отношения по некоторому условию используется операция **выбора**. В простейшем виде она записывается так:

Пусть дано отношение *r(R),* атрибут $A \in R$ и $a \in dom(A)$.

Определение 1.9

Оператор выбора $\sigma_{A=a}(r) = r'(R) = \{t \in r:\ t(A) = a\}$ - унарный оператор, результатом применения которого является отношение, состоящее из таких кортежей r , в которых атрибут A равен a.

Если требуется получить часть отношения только с некоторыми столбцами, используется оператор **проекции**:

Пусть r отношение со схемой R, $X \subseteq R$.

Определение 1.10

Оператор проекции $\pi_X(r) = r'(X) = \{t(X):\ t \in r\}$ - унарный оператор, результатом применения которого является отношение, полученное из r вычеркиванием столбцов R-X и исключением из оставшихся столбцов повторяющихся строк.

Оператор **соединения** комбинирует два отношения по всем их общим атрибутам.

Пусть $r(R)$, $s(S)$ - отношения, $R \bigcup S = T$.

Определение 1.11

Оператор соединения отношений $r(R)$ и $s(S)$ - бинарный оператор, результат которого - отношение $q(T)$:

$$r \rhd\lhd s = q(T) = \{t(T): \exists\, t_r \in r, \exists\, t_s \in s:\ t_r = t(R), t_s = t(S)\}$$

Таким образом, каждый кортеж в q является комбинацией кортежа из r и кортежа из s с равными ($R \cap S$)-значениями.

Если $R \cap S = \varnothing$, то $r \rhd\lhd s$ - декартово произведение r и s - изоморфное множество RS-кортежей.

Пусть $r(R)$, $s(S)$ - отношения, $S \subset R$.

$R' = R$-S.

Определение 1.12

Оператор деления - бинарный оператор, результатом которого является

r' - **частное** от деления r на s

$r' = r \div s = (R') = \{t: \exists\, t_s,\ t_r \subset r:\ t_r(R') = t$ и $t_r(D) = t_s\}$ -" r, разделенное на s "

Другой способ сформулировать определение:

$r \div s$ - максимальное подмножество множества, такое, что $r \rhd\lhd s$ содержится в r.

При обсуждении операции соединения было показано, что результат операции выбора можно получить, применив соединение с **постоянным отношением.**

Определение 1.13

Если A_1, A_2, ..., A_n - различные атрибуты, а c_i = *const*, $c_i \in dom(A_i), i = \overline{1,n}$, то запись

$<c_1{:}A_1, c_2{:}A_2, ..., c_n{:}A_n>$ есть **постоянный кортеж** $<c_1\ c_2 ...\ c_n>$ над схемой $A_1, A_2, ..., A_n$.

Постоянное отношение с любым числом атрибутов может быть построено из постоянных отношений с одним кортежем и одним атрибутом с помощью операции □̃

Иногда требуется сделать соединение отношения с самим собой. Для того, чтобы избежать дублирования столбцов в таком соединении, используется **оператор переименования атрибутов .**

$r(R)$, $A \subseteq R$, $B \subseteq (R-A)$.

Пусть $R' = (R-A)B$.

Определение 1.14

Оператор переименования - унарный оператор, результатом применения которого является отношение r с атрибутом A, переименованным в B, обозначается как $\delta_{A \leftarrow B}(r)$

$$r'(R') = \{t': \exists\, t \in r: t'(R-A) = t(R-A),\ t'(B) = t(A)\}$$

Требуется, чтобы $dom(A) = dom(B)$.

Пусть $r(R)$ - отношение, $A_1, A_2, ..., A_k \subseteq R$. $B_1, B_2, ..., B_k \subseteq R-(A_1 ... A_k)$, $dom(A_i) = dom(B_i)$,

Одновременное переименование A_i в B_i записывается так:

$$\delta_{A_1,...,A_n \leftarrow B_1,...,B_n}(r)$$

В определении оператора соединения, отношения могут комбинироваться только по одноименным столбцам и должны комбинироваться по всем таким столбцам. Выше было показано, как произвести соединение по подмножеству таких столбцов (с помощью переименования атрибутов)

Часто домены упорядочены, и имеют смысл операции $<, \leq, >, \geq, =, \neq$.

Множество Θ̣символов бинарных отношений над парами доменов вводится для возможных сравнений.

Если θ-знак сравнения, а A и B - атрибуты, то говорят, что A θ- сравним с B, если знаку сопоставлено бинарное отношение в $dom(A)\ x\ dom(B)$.

Сравнения будут использоваться для обобщения операторов выбора и соединения, в которых участвовало только равенство.

Теперь, с учетом множества Θможно сформулировать **оператор расширенного выбора.**

Определение 1.15

Пусть $r(R)$ - отношение, $A \in R$, $a \in dom(B)$ - константа, A θ- сравним с B.

Тогда **оператор расширенного выбора** создает отношение

$$\sigma_{A\theta a}(r) = r'(R) = \{t \in r:\ t(A)\theta a\}$$

$$t(A)\theta a = \theta(t(A), a)$$

Допускается сравнение не с константой, а с другими атрибутом.

$$\sigma_{A\theta B}(r) = r'(R) = \{t \in r:\ t(A)\theta t(B)\}$$

Определение 1.16

Пусть $r(R)$, $s(S)$: $R \cap S = \varnothing$, A и B θ- сравнимы.

Оператор тета - соединения создает отношение $q(RS)$, такое что

$$r[A\theta B] = q(RS) = \{\exists\, t_r \in r\ \exists t_s \in s:\ t_r(A)\theta t_s(B)\ t_r = t(R)\ t_s = t(S)\}$$

Допускается несколько сравнений над отношениями:

$$r\ [A_1 < B_1,\ A_2 = B_2,\ A_3 > B_3]\ s$$

Эквисоединение - частный случай тета-соединения, когда Θ = {=}.

Теоретико-множественные операции, операции обобщенного выбора, проекции, тета-соединения, деления, а также оператор переименования вместе с постоянными и регулярными отношениями относятся к реляционной алгебре.

Любое выражение, правильно построенное с помощью этих операторов и отношений, называется **алгебраическим выражением**.

Для данного алгебраического выражения E и данных значений всех отношений в E можно вычислить E и получить в качестве результата единственное отношение.

Пусть **U** - множество атрибутов - универсум

D - множество доменов

dom: **U**→D - полная функция из **U** в D

R = $\{R_1,..., R_p\}$ - множество различных схем отношений, где $R_i \subseteq \mathbf{U}$,

d = $\{r_1... r_p\}$ - множество всех таких наборов отношений, что

$r_i=r_i(R_i),\ i = \overline{1,n}$

Пусть Θ - множество бинарных отношений над доменами из D, содержащее по крайней мере = и ≠ для любого домена.

Определение 1.17

Реляционной алгеброй над **U**, D, dom, ***R***, d, и Θ называется семиместный кортеж

B = (U, D, dom, *R*, d, Θ, O), где O - множество операций

$\cup,\cap,-,\sim,\sigma,\pi,\div,\delta$,тета-соединения, использующих атрибуты из **U** и отношения из Θ .

Алгебраическим выражением над B называется любое выражение, правильно построенное из отношений из d, **U** и операторов из O .

1.1.3 Обеспечение целостности данных

Каждая сущность предметной области представляет собой описание целого класса объектов одного типа. Очевидно, что должны существовать правила, позволяющие отличить объекты одного типа один от другого, поскольку эта отличимость - одно из основных требований при описании предметной области задачи.

В реляционной модели для этого используется *механизм ключей.*

Пусть $R[A_1,...,A_n]$ - схема отношения *r*.

Говорят, что ключ отношения *r* со схемой *R* - подмножество *K:*

$$K = \{B_1, B_2,..., B_m\} \subseteq R:\ \forall t_1, t_2 \in r :\ t_1(K) \neq t_2(K)$$

и ни одно собственное подмножество $K' \subset K$ не обладает этим свойством.

Сформулируем 2 свойства ключа.

Первое свойство ключа - *уникальность*:

“Не существует двух кортежей, имеющих одно и то же значение ключа на всех атрибутах из K”

Второе свойство - *минимальность:*

“Ни один из атрибутов $B_1,...,B_m$ не может быть исключен из K без нарушения условия уникальности”

Поскольку любое отношение в реляционной модели данных должно удовлетворять требованию - не иметь одинаковых кортежей, то любое отношение обладает по крайней мере одним ключом - комбинацией всех его атрибутов.

Один произвольно выбранный ключ для данного отношения принимается за его **первичный ключ**. Остальные возможные ключи называются альтернативными.

Таким образом каждое отношение имеет свой первичный ключ. Если отношение описывает сущность, то этот ключ служит для идентификации конкретного объекта данной сущности. Отношение связи - связывает ключи двух и более сущностей.

Внешний ключ - это атрибут или комбинация атрибутов одного отношения, значение которого обязательно должно совпадать со значением первичного ключа некоторого другого отношения. Т.е. любое отношение связи содержит один или более внешних ключей некоторых сущностей.

Для того, чтобы набор сущностей и связей адекватно описывал предметную область задачи, необходимо, чтобы данные в разных отношениях модели данных были согласованы между собой.

Поскольку база данных должна описывать изменяющийся мир, то состояние отношений и связей между объектами в каждый момент времени может изменяться. Значит, эти изменения должны осуществляться так, чтобы в каждый момент времени данные были непротиворечивы. Требование непротиворечивости должна обеспечивать целостность данных.

Сформулируем правила целостности для реляционной модели данных (также см. [1])

1 . **Целостность по сущностям**.

Не допускается, чтобы какой-либо атрибут, участвующий в первичном ключе объектного отношения, принимал неопределенные значения.

Если бы первичный ключ принимал неопределенные значения, это говорило бы о том, что есть сущность, которая не обладает индивидуальностью.

2. **Целостность по ссылкам.**

Если объектное отношение r_2 включает некоторый внешний ключ *FK* , соответствующий первичному ключу *PK* какого-либо объектного отношения r_1, то каждое значение *FK* в r_2 должно либо:

а) быть равным значению *PK* в некотором кортеже r_1,

либо

б) быть полностью неопределенным, т.е. каждое значение атрибута, участвующего в этом значении *FK*, должно быть неопределенным.

(При этом r_1, r_2 - не обязательно различные объектные отношения)

Таким образом, основными компонентами реляционной модели данных являются:

Структура данных

Домены, n-арные отношения (атрибуты , кортежи)

Ключи (возможные, первичные, альтернативные, внешние)

Целостность данных

1.Значения первичных ключей не должны быть неопределенными.

2.Значения внешних ключей должны соответствовать значениям первичных ключей (или быть неопределенными)

Манипулирование данными

Реляционная алгебра В и реляционное присваивание.

1.2 Функциональные зависимости

Реляционное представление предметной области задачи накладывает ряд определенных ограничений на описание структуры и свойств объектов.

Так, например, при описании сложного объекта, с большим количеством ассоциаций и связей приходится представлять этот объект в виде набора объектных и связных отношений. Это позволяет понизить избыточность данных и повысить надежность всей системы в целом.

Для реализации такого представления объектов используют *технику нормализации*, состоящую в следующем:

1. Атрибуты одного отношения должны зависеть от ключа, от всего ключа целиком и ни от чего другого, кроме ключа. Если это не так, следует разбить исходное отношение на отдельные отношения, удовлетворяющие этому требованию.

2. Если можно разбить отношение на 2 или более отношений с меньшим числом атрибутов так, что (реляционное) соединение новых отношений воспроизводит исходное отношение, то это сделать нужно.

В результате нормализации, в идеальном случае, должно получиться следующее:

Каждое отношение состоит из:

а) первичного ключа, представляющего уникальный идентификатор некоторого типа сущностей, а также,

б) нуля или более дополнительных полей, представляющих дополнительные свойства этого типа сущностей.

Очевидно, что при нормализации должна использоваться некая дополнительная информация об объекте, для того, чтобы правильно представить этот объект в виде набора других, более простых.

Эта информация есть в самой схеме отношения, описывающей данных объект.

Каждое отношение описывается схемой - набором атрибутов. Между атрибутами одного отношения могут существовать некоторые однозначные зависимости, называемые функциональными.

Говорят, что атрибут A **функционально зависит** от атрибута B отношения r со схемой R, если и только если для каждого различного значения атрибута B существует единственное значение атрибута A.

Пишут: $B \rightarrow A$

Пусть r отношение со схемой $R, X \subseteq R, Y \subseteq R$

Определение 1.18

Отношение *r(R)* **удовлетворяет функциональной зависимости** $X \rightarrow Y$

если $\pi_{X=x}\left(\sigma_{Y=y}\left(r\right)\right)$ имеет не более чем один кортеж для каждого X-значения x

Говорят, что $X \rightarrow Y$ является **F-зависимостью** над схемой R.

Для отношения *r(R)* в любой момент времени существует семейство

F-зависимостей. Множество F-зависимостей, применимых к *r(R)*, конечно, т.к. существует только конечное число подмножеств множества R. Для того, чтобы получить все F-зависимости над схемой R, к исходному множеству F применяют так называемые аксиомы вывода.

Аксиома вывода - правило, устанавливающее, что если отношение удовлетворяет определенным F-зависимостям, то оно должно удовлетворять и некоторым другим F-зависимостям.

Армстронг [14] сформулировал эти аксиомы и доказал их полноту.

Аксиома F1: (Рефлексивность)

$$X \to X$$

Аксиома F2: (Пополнение)

$$X \to Y \quad \Rightarrow XZ \to Y$$

Аксиома F3: (Аддитивность)

$$X \to Y, X \to Z \quad \Rightarrow \quad X \to YZ$$

Аксиома F4: (Проективность)

$$X \to YZ \Rightarrow X \to Y$$

Аксиома F5: (Транзитивность)

$$X \to Y, Y \to Z \quad \Rightarrow \quad X \to Z$$

Аксиома F6: (Псевдотранзитивность)

$$X \to Y, YZ \to W \quad \Rightarrow \quad XZ \to W$$

Аксиомы F1,F2,F6 являются независимыми и называются аксиомами Армстронга.

Пусть F - множество F-зависимостей для $r(R)$.

Определение 1.19.

Замыкание *F*, F^+, - наименьшее содержащее F множество, такое, что при применении к нему аксиом Армстронга нельзя получить ни одной F-зависимостей, не принадлежащей F

Определение 1.20

Множество F-зависимостей G называется **покрытием** множества F-зависимостей F, если замыкания обоих множеств равны ($F^+ = G^+$).

Тогда множества F и G называются **эквивалентными** ($F \equiv G$)

Использование информации обо всех функциональных зависимостях над данной схемой отношения оказывается необходимым при создании непротиворечивых баз данных.

Дадим определение базы данных.

Определение 1.21

Схема базы данных *R* над **U -** множеством атрибутов - совокупность схем

$\{R_1,R_2,\ldots,\}$, где $R_i=(S_i, K_i)$, $i=\overline{1,p}$

$$\bigcup_{i=1}^{p} S_i = \mathbf{U} \text{ и } S_i \neq S_j, \quad i \neq j$$

Пусть $\boldsymbol{R}=\{R_1,R_2,\ldots,\}$ - схема базы данных.

Определение 1.22

База данных d со схемой *R* - совокупность схем отношений $\{r_1,\ldots,r_p\}$ таких, что для каждой схемы $R=\{S, K\}$ из ***R*** существует отношение r в d, являющееся отношением со схемой S и удовлетворяющим каждому ключу из K.

Пусть F - множество F-зависимостей, определенных в результате анализа предметной области.

Определение1.23

Пусть R - схема отношения.

F-зависимость $X \to Y$ **применима** к схеме R, если $X \subseteq R, Y \subseteq R$

Определение 1.24

База данных d **удовлетворяет множеству *F***, если любая F-зависимость $X \to Y$ из F, применимая к некоторой схеме R_i из ***R***, выполняется в отношении r_i.

Пусть G - подмножество F^+, применимых к некоторой схеме R_i из ***R***

Определение 1.25

Любая F-зависимость из G^+ называется **навязанной** схеме ***R*** .

Любая F-зависимость из F^+ - G^+ называется **ненавязанной** схеме ***R.***

Множество F **навязано**, если любая F-зависимость из F навязана схеме ***R***.

Чтобы показать, что F навязано схеме базы данных ***R***, достаточно найти его покрытие F', такое, что база данных удовлетворяет множеству F' и F' навязано схеме этой базы ***R***.

Говорят, что база данных d **подчиняется** множеству F-зависимостей F, если F навязано схеме $\boldsymbol{R}$ и d удовлетворяет F^+.

При всех преобразованиях схемы $\boldsymbol{R}$ базы данных d необходимо, чтобы d подчинялась множеству F-зависимостей F.

1.3 Постановка задачи анализа непротиворечивости реляционной модели данных при введении свойства вычисляемости атрибутов

В предыдущем разделе было введено понятие функциональной зависимости между наборами атрибутов в рамках одного отношения.

Факт существования F-зависимостей между атрибутами отношения говорит лишь о том, что есть некоторое однозначное соответствие одного подмножества атрибутов другому.

Однако, в некоторых случаях это однозначное соответствие может быть описано явно, с помощью некоторого функционала, т.е. существует формальное выражение, описывающее, *как именно* одно подмножество атрибутов зависит от других.

Определение 1.26

Функциональная зависимость $X \to A*$ называется **вычисляемой**, если существует $f: dom(X) \to dom(A*)$ - некоторый взаимно однозначный функционал.

Тогда пишут: $X \xrightarrow{f} A*$

Из вычисляемой функциональной зависимости $X \xrightarrow{f} A*$ всегда следует обычная функциональная зависимость $X \to A*$ (поскольку первая есть частный случай второй).

Атрибут схемы отношения, значение которого вычисляется на наборе значений других атрибутов схемы , называется вычисляемым (см. также [19]).

Определение 1.27

Атрибут $A*$ схемы $R = X \bigcup Y \bigcup A*, R[X_1,...,X_n,Y_1,...,Y_m,A*]$ называется **вычисляемым** (сокращенно $A* = f(X)$), если в отношении $r(R)$ значение кортежа t на атрибуте $A*$ есть функционал от значения кортежа на X, т.е.

$t(A^*)=f(t(X))$.

Тогда отношение, содержащее вычисляемый атрибут, называется отношением с вычисляемым атрибутом.

Определение 1.28

Отношение r со схемой $R = X \cup Y \cup A^*, R[X_1,...,X_n, Y_1,...,Y_m, A^*]$, где $A^* = f(X)$, есть множество $\{t \in r: \forall t_{XY} \in r(XY)\ \exists! t^* \in r(A^*): t(XY) = t_{XY},\ t(A^*) = t^*\}$ - называемое **отношением с вычисляемым атрибутом.**

Предлагается к рассмотрению реляционная алгебра над отношениями с вычисляемыми атрибутами.

Пусть **U** - множество атрибутов - универсум

D - множество доменов

dom: **U**$\rightarrow$D - полная функция из **U** в D

F - множество функционалов, определяющих вычисляемые атрибуты.

R = $\{R_1,..., R_p\}$ - множество различных схем отношений, где $R_i \subseteq$ **U**, ,

d = $\{r_1... r_p\}$ - множество всех таких наборов отношений, что

$r_i=r_i(R_i),\ i = \overline{1,n}$

Пусть Θ - множество бинарных отношений над доменами из D, содержащее по крайней мере = и ≠ для любого домена.

Определение 1.29

Реляционной алгеброй над **U**, D, dom, **F, R**, d, и Θ называется восьмиместный кортеж

B’ = (U, D, dom, **F**, R, d, Θ, O), где O - множество операций

$\cup, \cap, -, \sim, \sigma, \pi, \div, \delta$,тета-соединения, использующих атрибуты из **U** и отношения из Θ.

Сравним классическую реляционную алгебру с алгеброй B’. Согласно математического определения [23], любая алгебра есть:

1. Базовое множество M.
2. Набор операций над элементами базового множества.

Наборы операций (см. Опр.1.17) реляционных алгебр B и B’ одинаковы. Отличаются базовые множества.

В связи с этим, подлежат обсуждению такие вопросы:

- Исследование свойств операторов реляционной алгебры B’;
- Анализ применимости реляционного исчисления кортежей как системы запросов к отношениям из алгебры B’ ;
- Использование свойств операций реляционной алгебры B’ на практике.

2 АНАЛИЗ НЕПРОТИВОРЕЧИВОСТИ ФОРМАЛЬНОГО РАСШИРЕНИЯ РЕЛЯЦИОННОЙ МОДЕЛИ ДАННЫХ ПРИ ВВЕДЕНИИ ВЫЧИСЛЯЕМЫХ АТРИБУТОВ

В предыдущей главе было приведено определение вычисляемого атрибута, даны определения вычисляемой функциональной зависимости и отношения с вычисляемым атрибутом. На основании этих определений была предложена реляционная алгебра, поддерживающая такие отношения.

В этой главе будут рассмотрены реляционные операции над отношениями с вычисляемыми атрибутами и их свойства. Также исследуется применимость реляционного исчисления кортежей к таким отношениям.

Сразу необходимо отметить, что вычисляемые атрибуты не нарушают требования атомарности данных. Кроме того, проясним понятие функционала. Поскольку домены атрибутов могут быть любыми (числовыми, литерными и др.), то функционал в данном случае есть отображение, ставящее в соответствие набору атрибутов (со своими доменами) единственный атрибут. Подразумевается, что для каждого функционала, допустимого в реляционной алгебре B', существует алгоритм (программа), его вычисляющий, для допустимых исходных данных за конечное время.

Пусть реляционная алгебра представлена такими операциями над отношениями - объединение, пересечение, разность, декартово произведение, дополнение, проекция, выбор, соединение и деление.

2.1 Булевы операции над отношениями с вычисляемыми атрибутами

Вначале рассмотрим теоретико-множественные операции, поскольку любое отношение в реляционной модели данных есть множество кортежей.

Все эти операции над множествами применяются к отношениям с одинаковыми схемами R (эти отношения рассматриваются как подмножества множества всех кортежей со схемой R на домене D).

Предполагаем, что отношения, принимающие участие в булевых операциях, совместимы по объединению (см. раздел 1.1.2).

При соблюдении условия совместимости, введение свойства вычисляемости некоторых атрибутов не меняет результатов применения операций объединения, пересечения и разности, поскольку схема R при этом не меняется.

Операция декартова произведения комбинирует два отношения r_1 и r_2 (не обязательно с одинаковыми схемами) по всем атрибутам обоих отношений. Очевидно, что в этом случае непринципиально знать о вычисляемости некоторого атрибута одного из отношений.

Также рассмотрим операцию дополнения, которая определяется так:

$\bar{r} = dom(R) - r$ - из всех возможных значений кортежей на схеме R и домене D убрать те кортежи, которые есть в r.

Ее эквивалент в случае бесконечности одного из доменов атрибутов - операция активного дополнения,

$\tilde{r} = adom(R,r) - r$, $adom(R, r)$ - все возможные комбинации значений атрибутов схемы R, встречающихся в отношении r.

Т.е. булевы операции не меняют своего смысла при введении вычислимости некоторых атрибутов.

2.2 Специальные реляционные операторы и отношения с вычисляемыми атрибутами

Рассмотрим теперь специальные реляционные операции в применении к отношениям с вычисляемыми атрибутами.

Пусть $\Theta = \{=, <, >, \neq, \geq, \leq\}$ - множество операций сравнения на множествах.

2.2.1 Выбор и его свойства

Оператор выбора $\sigma_{A\,\theta\,a}(r)$ - выбирает из отношения r все кортежи t, такие что $t(A)$ θ-сравним с a

$$\sigma_{A\,\theta\,a}(r) = \{t \in r : t(A)\ \theta\ a\}$$

Если выбор ведется по вычисляемому атрибуту A^*, то

$$\sigma_{A^* \,\theta\, a^*}(r) = \{t \in r :\ t(A^*)\ \theta\ a^*\} = \{t \in r:\ f(t(X))\ \theta\ a^*\} \qquad (2.1)$$

$$a^* \in dom\,(A^*)$$

Пусть $\Theta = \{=\}$. Тогда выбор коммутативен:

если X - подмножество R, $A^* = f(X)$ и B - атрибуты схемы R, отношение r имеет схему R, то

$$\sigma_{A^* = a^*}(\sigma_{B = b}(r)) = \sigma_{A^* = a^*}(\{t \in r:\ t(B) = b\}) = \{t \in r:\ t(B) = b\ \ u\ \ t(A^*) = a^*\} =$$

$$= \sigma_{B = b}(\sigma_{A^* = a^*}(r)) = \sigma_{A^* = a^*,\, B = b}(r)$$

независимо от того, принадлежит B множеству X или нет, и дистрибутивен относительно бинарных булевых операций - объединения, пересечения, разности:

$$\sigma_{A=a}(r\ \gamma\ s) = \sigma_{A=a}(r)\ \gamma\ \sigma_{A=a}(s),\ \ \gamma = \{\cup, \cap, -\}, r(R), s(R)$$

Можно дать следующую интерпретацию формулы (2.1) при $\Theta = \{=\}$:

"Выбор кортежей с $A^* = a^*$ есть выбор по таким кортежам $<x_1, x_2, \ldots, x_n>$, что

$f(<x_1, x_2, \ldots, x_n>) = a^*$ "

$$\sigma_{A^* = a^*}(r) = \{t \in r :\ t(A^*) = a^*\} = \{t \in r: f(t(X)) = a^* = f(\langle x_1, x_2, \ldots, x_n \rangle)\}$$

Следовательно, если $A^* = f(X)$ и f - взаимно однозначная функция, то

$$\sigma_{X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n}(r) = \sigma_{A^* = a^*}(r) \qquad (2.2)$$

В общем случае, если $\Theta = \{=, <, >, \neq, \geq, \leq\}$, выбор некоммутативен.

2.2.2 Проекция и ее свойства

Оператор проекции $\pi_X(r)$ выбирает подмножество X столбцов отношения r и исключает из них повторяющиеся строки.

Если $R = X \cup Y \cup A^*$, $A^* = f(X)$, то рассмотрим разные проекции отношения $r(R)$.

Если множество $T = X$, или $T = Y$, или $T = X \cup Y$, то оператор проекции никак не меняет смысла, а при $T = X \cup A^*$ атрибут A^* в $\pi_T(r)$ - остался вычисляемым.

Если множество $T=A^*$, или $T=A^* \cup A$, $A \subset X$, или $T=A^* \cup B$, $B \subseteq Y$, то проекция $\pi_T(r)$ дает новое отношение, в котором атрибут A^* уже не обладает свойством вычисляемости.

Проверим свойства проекции:

Проекция коммутирует с выбором (при $\Theta = \{=\}$), если атрибуты для выбора находятся среди атрибутов множества, на которое осуществляется проекция.

Если $A \in T$, $T \subseteq R$, r - отношение со схемой R, то

$$\pi_T(\sigma_{A=a}(r)) = \sigma_{A=a}(\pi_T(r)) \qquad (2.3)$$

Для проекции на множество атрибутов, среди которых есть вычисляемый:

$$\pi_{A^* \cup X}(\sigma_{A^*=a^*}(r)) = \sigma_{A^*=a^*}(\pi_{A^* \cup X}(r)) = \sigma_{X_1=x_1,\ldots,X_n=x_n}(\pi_{A^* \cup X}(r))$$

в силу формулы (2.2).

Тождество (2.3) может не выполняться, если $A \notin T$.

В случае вычисляемости одного из атрибутов:

если $R = X \cup Y \cup A^*$, $A^* = f(X)$, $r(R)$, то

$\pi_T(\sigma_{A^*=a^*}(r)) = \sigma_{A^*=a^*}(\pi_T(r))$, но если $T=A \cup B$, $A \subset X$, $B \subseteq Y$, то

$\pi_T(\sigma_{A^*=a^*}(r)) \neq \sigma_{A^*=a^*}(\pi_T(r))$

2.2.3 Соединение отношений

Если даны два отношения $r(R)$ и $s(S)$, то оператор соединения комбинирует эти отношения по всем их общим атрибутам, а в общем случае тета-соединение отношения r по атрибуту A с отношением s по атрибуту B - множество всех кортежей, таких, что t является объединением некоторого кортежа t_r из r и t_s из s, и предикат "$t_r(A)$ θ $t_s(B)$" принимает значение "истина".

При этом атрибуты A и B должны быть определены на одном и том же домене, и для этого домена операция θ имела бы смысл.

$$r[A \theta B]s = \{t: \exists t_r \in r, \exists t_s \in s: t(R) = t_r \; u \; t(S) = t_s \; u \; t(A) \; \theta \; t(B)\},$$
$$A \in R, B \in R, \; dom(A) = dom(B) \qquad (2.4)$$

Частными случаями являются эквисоединение ($\Theta = \{=\}$), и естественное соединение ($r \rhd\lhd s$ - требуется равенство доменов и имен атрибутов, по которым ведется соединение).

Очевидно, если A^* - вычисляемый, то $\theta-$соединение отношения r по атрибуту A^* с отношением s по атрибуту B возможно только если $dom(A^*) = dom(B)$

(B - тоже может быть вычисляемым)

Возможна ситуация, что существует атрибут A: $A \in R$ и $A \in S$. Желательно, чтобы для каждого вхождения атрибута A был свой столбец. Для этого можно потребовать в определении, чтобы $R \cap S = \varnothing$, а иначе следует переименовать атрибут A.

Если домены $dom(A)$ и $dom(B)$ в формуле (2.4) не равны, то:

либо $dom(A) \cap dom(B) = \varnothing$, либо $dom(A) \cap dom(B) \neq \varnothing$, но $dom(A) \neq dom(B)$

В первом случае при тета-соединении получаем декартово произведение $dom(A) \times dom(B)$

Во втором случае тета-соединение будет проводиться по всем значениям того домена, который меньше.

2.2.4 Деление отношений

Пусть даны два отношения, $r(R)$ и $s(S)$, отношение $s(S)$ представляет некоторый набор кортежей, описывающий свойства и отношение $r(R)$ представляет сущности с данными свойствами ($S \subset R$). Оператор деления $r \div s$ дает отношение $q(Q)$ ($Q = R-S$) такое, что каждый кортеж этого отношения удовлетворяет набору свойств, описанных отношением $s(S)$

($Q \cup S = R$)

$$r \div s = q(Q) = \{t: \forall t_s \in s \; \exists t_r \in r: t_r(Q) = t \text{ и } t_r(S) = t_s\}$$

Можно сказать, что $r \div s$ - максимальное подмножество множества $\pi_Q(r)$, такое что $q \rhd\lhd s$ содержится в r..

При делении r на s нужно найти такие кортежи из r, чтобы их (R-S) - значения были одинаковы для всех S-значений из s .

Рассмотрим отдельно случай, когда известно, что $A^*=f(X)$ в отношении r со схемой $R[X_1, \ldots, X_n, Y_1, \ldots, Y_m, A^*]$.

Пусть $s=s(X_1,\ldots,X_n)$. Разделим r на s.

$$r \div s = w(A^* Y_1, \ldots, Y_m)$$

Пусть также $q=q(A^*)$.

$$r \div q = v(X_1, \ldots, X_n, Y_1, \ldots, Y_m)$$

По определению оператора деления, $s \rhd\lhd w \subseteq r$, $q \rhd\lhd v \subseteq r$. Проверим, равны ли между собой эти два соединения . При построении частного от деления используется оператор выбора $\sigma_{X_1 = x_1^i, X_2 = x_2^i, \ldots, X_n = x_n^i}(r)$, и результат выбора запоминается. Если для любого кортежа из s, $\langle x_1^i, \ldots, x_n^i \rangle$, $i = \overline{1, p}$, (p- кардинальное число отношения s), все кортежи - результаты выборов совпадают, эти кортежи и составляют частное от деления.

$w = \bigcap_{i=1}^{p} \pi_{A^*Y}(\sigma_{X_1 = x_1^i, \ldots, X_n = x_n^i}(r)) \neq \varnothing$ - частное от деления $r(XA^*Y)$ на $s(X)$

$v = \bigcap_{i=1}^{p} \pi_{XY}(\sigma_{A^* = a_i^*}(r)) \neq \varnothing$ - частное от деления $r(XA^*Y)$ на $q(A^*)$.

Из тождества (2.2) следует, что выбор можно производить и по вычисляемому атрибуту, используя те его значения, которые соответствуют значениям из s.

Значит, $s \rhd\lhd w = q \rhd\lhd v$

(2.5)

Кроме того, поскольку $X \xrightarrow{f} A^*$ - взаимно однозначное соответствие, то добиться того, чтобы результат деления r на отношение s(X) или q(A*) был непуст, можно только в случае деления r на постоянное отношение, состоящее только из одного кортежа. Это очевидно, т.к. каждому $t(S) \in s$ соответствует единственный кортеж t(A*), а значит, единственный кортеж t(A*Y).

2.3 Алгебра B’ как система запросов

Реляционная алгебра является системой манипулирования реляционными данными. Операция над отношениями, результатом которой также является отношение называется *запросом*. Т.о. реляционная алгебра есть система запросов. Система запросов образует базовую структуру *языка запросов*, т.е. специализированных языков программирования, которые используются в системах баз данных для формулирования команд.

При сравнении разных систем манипулирования данными основной характеристикой является выразительная сила (полнота) языка запросов. Традиционно считается, что классическая реляционная алгебра представляет минимальный уровень возможностей языка запросов к реляционной базе данных, поэтому любая система запросов к таким базам должна хотя бы поддерживать этот уровень. Все расширения чаще всего связаны с реализациями этого языка. Ульман [2] предложил способ проверки выразительной силы языка запросов и доказал, что реляционное исчисление кортежей и доменов, как системы запросов, и реляционная алгебра являются равными по выразительной силе.

В настоящей работе используются результаты Ульмана для доказательства того, что выразительная сила алгебры B’ не меньше с классической реляционной алгеброй.

Утверждение 1.

Алгебра B’ не менее выразительна, чем классическая реляционная алгебра, т.к. для любого допустимого выражения над B существует эквивалентное выражение над B’ для любого состояния базы данных.

Доказательство этого факта следует из определения алгебры B’, поскольку все операции над отношениями остались прежними, а значит всякому выражению из B соответствует эквивалентное (по определению конкретной операции) выражение из B’, а отношения .

Утверждение 2.

Поскольку реляционная алгебра B’ полна и замкнута относительно множества своих операций, то на основании утверждения 1 и результатов Ульмана, реляционная алгебра B’ является не менее выразительной, чем реляционное исчисление кортежей и доменов.

Каждой формуле исчисления кортежей соответствует эквивалентное выражение алгебры B’.

2.4 Выводы

Проведя изучение свойств операций реляционной алгебры B’, можно сделать следующие выводы:

1. Введение нового свойства атрибутов не противоречит базовым понятиям реляционной модели данных.
2. Все операции классической реляционной алгебры применимы к отношениям с вычисляемыми атрибутами. Замкнутость алгебры при этом не нарушается.
3. Все свойства этих операций также выполняются и в алгебре B’.
4. Реляционные операции в алгебре B’ обладают рядом дополнительных свойств, не присущих классической алгебре B.
5. При сравнении классической реляционной алгебры и алгебры B’ как систем запросов выяснилось, что выразительная сила алгебры B’ не меньше, чем выразительная сила B. Реляционное исчисление кортежей позволяет описание формул, эквивалентных выражениям реляционной алгебры B’. Это означает, что любые отношения с вычисляемыми атрибутами являются допустимыми при применении языков запросов, основанных на исчислении, в отличие от вычисляемых отношений из работы [18].

3 ВЫЧИСЛЯЕМЫЕ ФУНКЦИОНАЛЬНЫЕ ЗАВИСИМОСТИ

В этой главе будут рассмотрены вычисляемые функциональные зависимости и их использование при проектировании баз данных.

Прежде всего, в разделе 3.1 будет рассмотрен вопрос о нахождении множества всех F-зависимостей, F^+, на заданной схеме баз данных и исследована система аксиом Армстронга в применении к множеству F-зависимостей, среди которых есть вычисляемые, поскольку информация об F^+ используется при нормализации схемы базы.

Затем рассматривается процесс нормализации отношений с вычисляемыми атрибутами и выясняются его закономерности (раздел 3.2).

При создании системы манипулирования базой данных важное место занимает выбор способа описания действий над данными на высоком уровне и оптимизация такого описания для достижения выигрыша в ресурсах и времени. В разделе 3.3 рассматриваются пути оптимизации системы запросов, основанной на классической реляционной алгебре, использующие свойства операторов алгебры B'.

3.1 Аксиомы вывода и вычисляемые F-зависимости.

Как было описано в гл.1.2., используя полную систему аксиом Армстронга, можно получить для заданного набора F-зависимостей замыкание F^+. Информация о множестве всех возможных F-зависимостей над схемой базы данных используется для получения различных покрытий множества F-зависимостей. Получив некоторое покрытие G множества F, мы имеем возможность построить альтернативные наборы нормализованных отношений.

Необходимо проверить, можно ли получить все вычисляемые F-зависимости из замыкания множества исходных F-зависимостей, используя аксиомы Армстронга.

Применим аксиомы F1-F6 к вычисляемым F-зависимостям.

Пусть задано отношение r со схемой R, $X \subseteq R, A^* \in R, A^* = f(X)$

F1. Рефлексивность

$$A* \xrightarrow{f} A*$$

F2. Пополнение

$$X \xrightarrow{f} A* \;\Rightarrow\; XZ \xrightarrow{f_1} A*$$

Если $t(A*) = f(t(X)),\quad f : dom(X) \to dom(A*)$, то пусть

$f_1 : dom(XZ) \to dom(A*)$ такая, что $f_1(t(XZ)) \equiv f(t(X)),$ тогда

$$t(A*) = f_1(t(XZ)) = f(t(X))$$

Данная аксиома выполняется для вычисляемых функциональных зависимостей.

F3.Аддитивность

$$X \xrightarrow{f_1} A*, X \xrightarrow{f_2} B* \;\Rightarrow\; X \to A*B*$$

По определению вычисляемой F-зависимости, одновременно значения двух атрибутов по одной и той же функциональной зависимости на одном и том же значении множества атрибутов X вычислить нельзя.

Аксиома аддитивности не выполняется.

F4: Проективность

Поскольку в определении вычисляемой F-зависимости область значений функционала - домен одного атрибута, то данная аксиома неприменима к таким зависимостям.

F5: Транзитивность

$$X \xrightarrow{f_1} A*, A* \xrightarrow{f_2} B* \;\Rightarrow\; X \xrightarrow{f_3} B*$$

Пусть $f_1 : dom(X) \to dom(A*),\quad f_2 : dom(A*) \to dom(B*)$. Тогда

пусть $f_3 : X \to B*,\quad f_3(t(X)) \equiv f_2(f_1(t(X)))$.

Т.е. аксиома транзитивности выполняется для вычисляемых F-зависимостей.

F6: Псевдотранзитивность

$$X \xrightarrow{f_1} A^*,\ A^* B \xrightarrow{f_2} C^* \ \Rightarrow\ XB \xrightarrow{f_3} C^*$$

Для проверки аксиомы F6 определим вспомогательные функции.

Пусть $g_1 : dom(R) \to dom(A^*)$,

пусть $f_1 : dom(X) \to dom(A^*)$ - сужение функции g_1 на подмножестве атрибутов X.

$f_1(t(X)) \equiv g_1(\langle x_1,...,x_n;0,...,0\rangle)$, где 0 - неопределенное значение.

Пусть $g_2 : dom(R) \to dom(C^*)$,

пусть $f_2 : dom(A^*) \times dom(B) \to dom(C^*)$ - сужение функции g_2 на подмножестве атрибутов A^*B.

$$f_2(t(A^* B)) \equiv g_2(\langle a^*, b_1,...,b_m;0,...,0\rangle)$$

Тогда определим $f_3 : dom(X) \times dom(B) \to dom(C^*)$ - сужение функции g_2 на подмножестве атрибутов $X \cup B$.

$$f_3(t(XB)) \equiv g_2(\langle x_1,...,x_n, b_1,...,b_m;0,...,0\rangle)$$

Поскольку $f_2(t(A^* B)) = g_2(\langle f_1(t(X)), b_1,...,b_m;0,...,0\rangle)$, то верно следующее:

$$t(C^*) = f_3(t(XB)) = f_2(t(A^* B)) = g_2(\langle f_1(t(X)), b_1,...,b_m;0,...,0\rangle)$$

что подтверждает применимость аксиомы псевдотранзитивности к вычисляемым F-зависимостям.

Т.о. видно, что не все аксиомы F1-F6 применимы к вычисляемым F-зависимостям. Аксиомы F1,F2,F5,F6 - выполняются и не выводят за пределы множества вычисляемых функциональных зависимостей. Аксиомы F3,F4 неприменимы по определению вычисляемой F-зависимости.

На основании проведенного анализа можно выделить следующие факты:

1. Из функциональных зависимостей вывести вычисляемые F-зависимости нельзя. Наоборот - можно (например, применяя аксиому вывода F3).

2. Для данного множества F-зависимостей над схемой $R[A_1,\ldots,A_n;A^*;B_1,\ldots,B_m]$, среди которых есть и вычисляемые, с помощью аксиом F1-F6 выводятся все возможные F-зависимости над схемой R.

3. Т.к. аксиомы F1, F2, F6 являются независимыми (т.е. ни одна из них не следует из других) и применимыми к вычисляемым F-зависимостям, использование этих аксиом дает возможность получить **все** вычисляемые F-зависимости над схемой R.

3.2 Нормализация отношений

Часто отношения, описывающие объекты и связи между объектами предметной области, содержат избыточную информацию или данные скомпонованы таким образом, что при изменении данных в одном отношении нарушается общая целостность данных. Такие ситуации возникают в тех случаях, когда при построении базы данных не учитывается или не полностью учитывается информация о функциональных зависимостях внутри отношений базы данных.

Проиллюстрируем эти проблемы на примерах:

- Избыточность данных

Одна из главных целей при создании базы данных - уменьшить избыточность данных, т.е. создать такие условия, чтобы выполнялось требование “каждый факт - в одном месте”. Хранение одной информации в нескольких местах ведет к бесполезной трате места хранения и увеличивает общие размеры базы данных. Обновление информации в такой базе данных имеет потенциальную возможность привести к противоречиям.

Пример:

В отношении $r(\underline{AB}CDE)$, первичный ключ AB, хранится несколько кортежей.

Видно, что в отношении несколько раз повторяются подкортежи $<a_1 c_1 d_1>, <a_3 c_3 d_3>$.

В этом случае причина избыточности в том, что при формировании схемы R не учтено существование F-зависимостей $A \to C, A \to D$ внутри этой схемы.

r	*A*	*B*	*C*	*D*	*E*
	a_1	b_1	c_1	d_1	e_1
	a_2	b_2	c_2	d_2	e_2
	a_1	b_3	c_3	d_3	e_3
	a_3	b_4	c_4	d_4	e_4
	a_3	b_5	c_5	d_5	e_5
	a_4	b_6	c_6	d_6	e_6

Рисунок 3.1 - Избыточность данных

- Аномалии обновления

Дублирование информации ведет к аномалиям обновления: если операция обновления сделана, но не все кортежи изменены.

Пример:

В отношении *r(<u>AB</u>CD),* первичный ключ *AB,* есть F-зависимость $A \to D$

r	*A*	*B*	*C*	*D*
	a_1	b_1	c_1	d_1
	a_1	b_2	c_2	d_1
	a_2	b_3	c_1	d_2

Рисунок 3.2 - Аномалии обновления

При выполнении операции обновления $CH(r;A=a_1,B=b_1;C=c_1,D=d_3)$ новое отношение перестает удовлетворять этой F-зависимости. Для того, чтобы избежать такой ситуации, нужно после каждого обновления просмотреть полученное отношение и исправить все вхождения d_1 на d_3 . Хотя требовалось изменить всего один кортеж.

- Аномалии удаления

Если к отношению на рисунке 3.1 применить операцию удаления DEL $(r;$ $A=a_1,B=b_1)$, то в случае, если это единственное отношение, показывающее связь между B и E, информация о значении e_1 атрибута E в базе данных будет утеряна. Чтобы избежать этого, необходимо учесть F-зависимость $B \to E$.

Пути решения этих проблем существуют, а именно - использование функциональных зависимостей при построении базы данных .

Пусть задана база данных d со схемой R над U.

Рассмотрим отношение r со схемой R из базы данных d. Пусть K - множество выделенных ключей. В идеальном случае никаких других F-зависимостей, кроме $K \to F$ ($K \in K$, F - некоторый **непервичный** атрибут) быть не должно.

Почти всегда это не так, поэтому цель нормализации - избавиться от всех других F-зависимостей, которые не имеют нужного вида.

Следует рассмотреть, по существу, два случая:

1. Пусть отношение r со схемой R имеет составной первичный ключ, например (K_1,K_2) и включает также поле F, которое функционально зависит от K_2 (или K_1).

Определение 3.1

***F* частично зависит** от $X=\{K_1,K_2\}$, если существует $X' \subset X$, такое, что $X' \to Y$ - также F-зависимость, которой удовлетворяет отношение r.

Иначе - атрибут F **полностью зависит** от X.

Выход: (см. рисунок 3.3).

r (<u>K1,K2</u>,F) $\quad\quad$ r_1 (<u>K2</u>,F)

$\boldsymbol{K}$={K1 K2} $\Rightarrow$

$K2 \to F$ $\quad\quad$ r_2 (<u>K1,K2</u>)

Рисунок 3.3 - Удаление частичной зависимости

Пусть отношение $r(R\ )$ имеет первичный ключ K, атрибуты F_1, F_2, такие, что $F_1 \to F_2$

Определение 3.2

Для данной схемы R отношения r, $X \subseteq R, A \in R$ и множества F-зависимостей атрибут A называется **транзитивно зависимым** от X в R, если

$$\exists Y \subseteq R\text{: } X \to Y,\ \neg\left(Y \to X\right),\ Y \to A,\ A \notin XY$$

Выход: (см. рисунок 3.4)

r (K,F1,F2) $\quad\quad$ $r1$ (K,F1)

K={K} $\quad\quad$ $\Rightarrow$

$F1 \to F2$ $\quad\quad$ $r2$ (F1,F2)

Рисунок 3.4 - Удаление транзитивной зависимости

Пусть R - схема отношения в схеме базы данных $\boldsymbol{R}$ над $\boldsymbol{U}$, F - множество F-зависимостей .

Пусть $X \subseteq R, A \in R$.

Определение 3.3

Атрибут A **внешне зависит** от X в F, если в $\boldsymbol{U}$ существует подмножество Y, такое, что

Y не является подмножеством R, и

$$X \to Y, \neg(Y \to X), Y \to A, \ A \notin XY$$

С понятием нормализации тесно связано понятие **нормальной формы**, введенной Коддом [11].

Определение 3.4

Схема R отношения r находится в **первой нормальной форме (1НФ)**, если значения в $dom(A)$ являются атомарными для любого A из R.

Это требование исходит из самого определения реляционной модели данных, а именно, предполагается, что все атрибуты любого отношения являются атомарными, а не последовательностью значений.

Атрибут A называется **первичным**, если он принадлежит какому-либо ключу схемы R.

Определение 3.5

Схема R отношения r находится во **второй нормальной форме (2НФ)** относительно множества F-зависимостей F, если она находится в 1НФ и каждый непервичный атрибут полностью зависит от каждого ключа для R.

Определение 3.6

Схема R отношения r находится в **третьей нормальной форме (3НФ)** относительно множества F-зависимостей F, если она находится в 1НФ и ни один из непервичных атрибутов в R не является транзитивно зависимым от ключа для R.

Справедливо, что любая схема отношения, находящегося в 3НФ, находится в 2НФ(см.[20])

Существует два основных подхода к нормализации схем баз данных.

Первый подход состоит в следующем: задана схема базы данных, известно множество функциональных зависимостей , которому подчиняется эта схема. Найти альтернативную схему базы данных, находящуюся в третьей нормальной форме. Такой подход получил называние **декомпозиции**.

Второй подход называется **синтезом**. Суть его в том, что по заданному множеству функциональных зависимостей F нужно построить схему базы данных, находящуюся в 3НФ, такую, что множество F навязано этой схеме.

Эффективный алгоритм синтеза был предложен Бернштейном [17]. Этот алгоритм основан на построении замыкания множества функциональных зависимостей. Поскольку для вычисляемых функциональных зависимостей была показана применимость аксиом вывода (в разделе 3.1), то алгоритм синтеза применяется к множеству функциональных зависимостей, среди которых есть вычисляемые. В рамках данной работы этот алгоритм подробно не рассматривается, поскольку его изучение не входило в цели работы.

Остановимся на декомпозиции отношений, поскольку этот метод нормализации проще и наглядно показывает особенности нормализации схемы базы данных с вычисляемыми атрибутами.

Декомпозиция реляционной схемы $R = (A_1, A_2, \ldots, A_n)$ - замена этой схемы набором реляционных схем $\{R_1, \ldots, R_m\}$, таких, что $R_i \subseteq R,\ i = \overline{1,m}$, и $\bigcup_{i=1}^{m} R_i = R$.

Алгоритм декомпозиции в общем виде таков:

"Каждую схему отношения, не находящуюся в 3НФ относительно F разложить в схему базы данных в 3НФ. Разложение - разбиение исходной схемы R на пару схем так, чтобы любое $r(R)$ разлагалось без потерь (см.[20]) на R_1 и R_2."

Рассмотрим варианты декомпозиции на примере схемы $R=(A_1\ A_2\ A_3\ A^*\ B_1\ B_2)$.

Пусть $A^*=f(A_1,A_2)$.

На рисунке 3.5 показано отношение с частичной зависимостью вычисляемого атрибута от выделенного ключа и его декомпозиция. На рисунке 3.6 отношение имеет транзитивную зависимость вычисляемого атрибута от ключа.

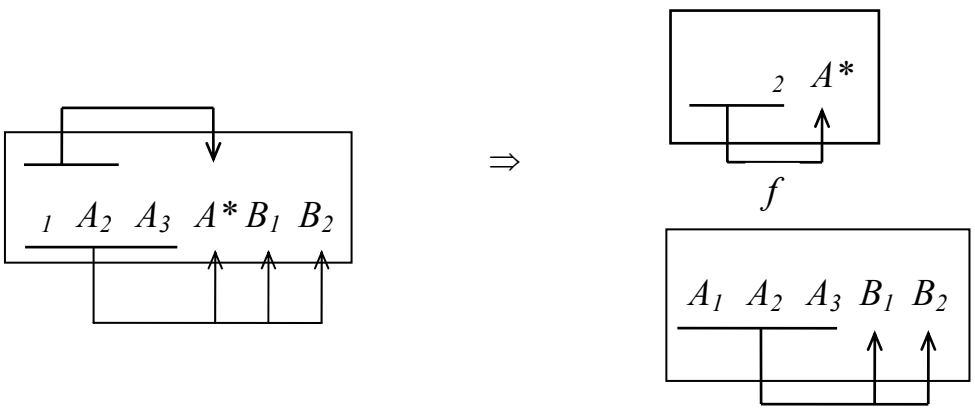

Рисунок 3.5 - Частичная зависимость A* от ключа

В общем случае, если A^* вычисляется на подмножестве атрибутов ключа, то отношение распадается на два других. Одно отношение отвечает только вычисляемой F-зависимости, другое - содержит все атрибуты исходного, кроме вычисляемого.

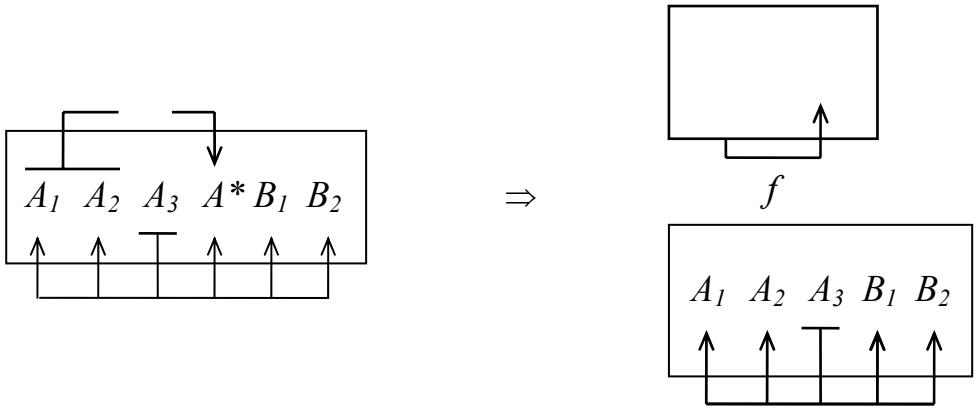

Рисунок 3.6 - Транзитивная зависимость A^* от ключа

На следующем рисунке показан обобщенный вид зависимости A^* от ключа.

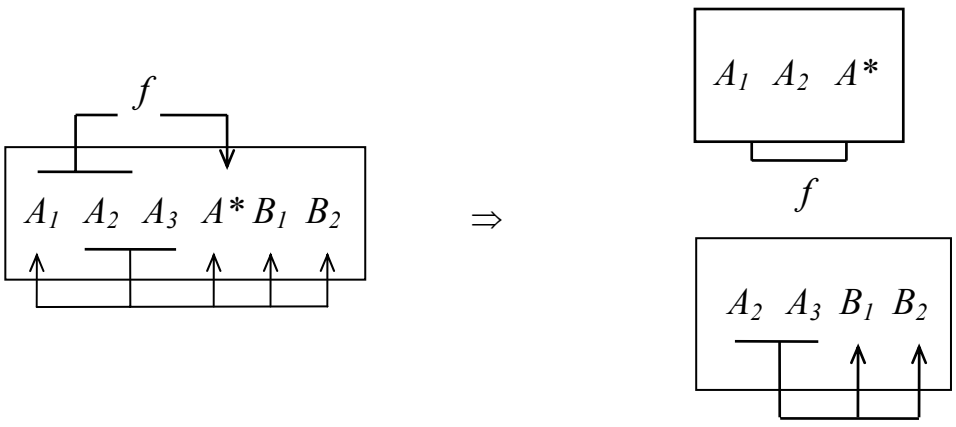

Рисунок 3.7 - Частичная и транзитивная зависимость A^* от ключа

Проанализируем эти примеры. Легко видеть, что при наличии среди F-зависимостей транзитивных и частичных зависимостей, а также их комбинаций, применение правил декомпозиции ведет к разбиению исходного отношения на два или более, среди которых обязательно будет отношение, удовлетворяющее только вычисляемой F-зависимости.

Другой интересный факт состоит в том, что при декомпозиции возможна замена набора атрибутов X на один атрибут A^*, даже если атрибуты X являются первичными, т.е. принадлежат какому-либо ключу этой схемы.

Более того, поскольку основное требование к функционалу, определяющему вычисляемую F-зависимость, есть его биективность (см. гл.2), то из $A_1A_2 \xrightarrow{f} A^*$ также следует $A^* \to A_1A_2$.

Пусть теперь в схеме R (A_1 A_2 A_3 A^* B_1 B_2) есть такие функциональные зависимости:

$$F = \{A_1A_2A_3 \to A^*B_1B_2,\ A_1A_2 \to A^*,\ A^* \to B_1\}$$

т.е. атрибут B_1 транзитивно зависит от ключа $A_1A_2A_3$..

При обычных функциональных зависимостях при удалении транзитивной зависимости иногда возникает такая ситуация, что множество F-зависимостей перестает быть навязанным этой схеме, а именно, при декомпозиции получаются три отношения,

R_1 ,R_2 и R_3, со схемами:

$R_1[A^*B_1]$, $K_1 = \{A^*\}$

$R_2[A_1A_2A^*]$, $K_2 = \{A_1A_2\}$

$R_3[A_1A_2A_3B_2]$, $K_3 = \{A_1A_2A_3\}$

но такая F-зависимость как $A_1A_2 \to B_1$ уже не выполняется, хотя и принадлежит F^+.

Получается, что B_1 внешне зависит от $A_1A_2A_3$.. Появление внешних зависимостей относится к недостаткам декомпозиции.

Однако, если атрибут A^* является вычисляемым, т.е. $A^*=f$ (A_1, A_2), то при декомпозиции внешней зависимости B_1 от $A_1A_2A_3$ не будет, т.к. существует обратная F-зависимость,

$A^* \to A_1A_2$.

Таким образом, декомпозиция может эффективно применяться для нормализации отношений с вычисляемыми атрибутами. При этом устраняется один из недостатков декомпозиции - появление внешних зависимостей.

3.3 Оптимизация запросов и отношения с вычисляемыми атрибутами

Реляционная алгебра является низкоуровневой системой запросов, т.к. любое алгебраическое выражение показывает, что нужно сделать с отношениями, чтобы получить искомый результат. Реляционное исчисление, напротив, является высокоуровневой системой запросов, т.е. допустимое выражение реляционного исчисления показывает, что нужно получить, а не как это сделать. Поэтому в высокоуровневых языках запросов при обработке запроса происходит его трансляция в алгебраическое выражение. Правомочность такого преобразования была доказана (см. ссылки к гл.1.2)

Оптимизация запросов имеет своей целью получение выражения, эквивалентного данному, но требующего для его вычисления меньше времени и памяти.

Значительные успехи в оптимизации запросов часто могут быть достигнуты просто путем изменения порядка выполнения операций.

Были разработаны алгоритмы преобразования запросов, сводящиеся к следующему (см.[2]):

1. Переместить селекции, чтобы они выполнялись как можно раньше.
2. Вынести проекции наружу, чтобы они выполнялись в последнюю очередь.
3. Собрать селекции и проекции, применяемые к одному и тому же отношению.
4. Скомбинировать селекции и проекции с теоретико-множественным объединением и разностью, чтобы использовать однопроходный метод там, где это возможно.Сделать также и обычные упрощения.
5. Внести проекции внутрь и оценить “стоимость” (обычно это стоимость сортировки) перемещения проекция на различную глубину, что позволит облегчить выполнение соединений за счет обработки меньшего количества более коротких отсортированных кортежей.
6. Снова проверить, стоило ли комбинировать селекции .
7. Найти общие выражения и организовать их однократное вычисление, запомнив полученные значения.

Какие же свойства реляционных операций могут быть использованы в оптимизации запросов к отношениям с вычисляемыми атрибутами ?

1. Перестановка операций селекции и соединения

Поскольку операции типа соединения и декартова произведения работают как генераторы, создавая большое количество кортежей, то выполнив сначала селекции, мы уменьшаем количество кортежей, участвующих в соединении.

2. Перестановка проекций и соединений, где это возможно (см. раздел 2.2.2).

При выполнении перестановок проекции и селекции следует помнить, что эта операция не всегда коммутативна.

3. Модификация выражений с несколькими применениями селекций.

Существуют “быстрые” и “медленные” операции селекций. “Быстрые” операции селекции можно осуществить с использованием прямого доступа к нужным элементам, например, с помощью индексирования. Для этих селекций время выполнения зависит в основном от количества отобранных кортежей, а не от всего отношения целиком. Поэтому быстрыми селекциями могут быть лишь селекции по равенству значений в некоторых столбцах отношения. Используя свойство операции выбора (см. формулу (2.2)), отметим следующее:

Если ведется выборка по некоторому предикату P, действующему на подмножестве атрибутов схемы отношения, таких, что вычисляемый атрибут определяется в точности на этом подмножестве, то этот предикат можно заменить на предикат P’ вида “$A^*=a^*$”, и в дальнейшем использовать эквивалентное выражение с предикатом P’.

Таким образом, возможно “ускорение” “быстрых” селекций за счет смены условия поиска. Однако, есть и ограничение: такая селекция очень эффективна, если выбор ведется по всему множеству атрибутов, определяющих вычисляемый атрибут, и по отношению равенства.

Модифицированный с учетом вычисляемых атрибутов вариант алгоритма оптимизации запросов имеет следующую особенность:

Пункт 1: ”Селекции переносятся вглубь выражения, при этом любая селекция по множеству атрибутов, определяющих некоторый вычисляемый, заменяется равной селекций по этому вычисляемому атрибуту”.

3.4 Выводы

Использование вычисляемых F-зависимостей при проектировании схем баз данных:

- допустимо, т.к. такие F-зависимости есть частный случай обычных функциональных;

- полезно при нормализации схем баз данных, т.к позволяет упростить этот процесс;

Свойства операторов реляционной алгебры над отношениями с вычисляемыми атрибутами могут использоваться при оптимизации запросов.

4 ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ ВЫЧИСЛЯЕМЫХ АТРИБУТОВ

Вычисляемые атрибуты могут использоваться на практике для решения самых разных задач. Выделение вычисляемых атрибутов как отдельных свойств объектов или связей может быть оправдано по таким причинам:

- частота использования атрибута достаточно велика;

- вычисление значения атрибута ресурсоемко (по времени, по памяти);

- необходим контроль за изменениями атрибута;

Специфика вычисляемых атрибутов состоит в том, что значение такого атрибута несет обобщенную информацию об объекте или связи между объектами. Поясним причины использования вычисляемых атрибутов на практике.

Первый и самый простой путь использования таких атрибутов следует из анализа предметной области задачи. Часто такие атрибуты являются неотъемлемыми свойствами объектов модели данных.

Например, рассмотрим отношение *"*СПРОС*"*, отражающее спрос на лекарства разных производителей в каждый момент времени (см. рисунок 4.1):

"СПРОС" [*id поставщика, id лекарства, Всего, Продано, Дата_поставки, Уровень_спроса*]

Атрибут "У*ровень_спроса*" вычисляется на атрибутах "*Всего*","*Продано*","*Дата_поставки*". Значение этого атрибута есть такая функция, которая ставит в соответствие арифметическому выражению признак уровня спроса (например, строку вида "Высокий", "Средний" и т.п.).

Рисунок 4.1 - Отношение "СПРОС"

Атрибут "уровень_спроса" является необходимым свойством связного отношения "СПРОС".

Следующий пример использования вычисляемых атрибутов - автоматически вычисляемые оценки наборов тех или иных параметров, например, результаты психологических тестов, химических и биологических анализов, вычисления оценок в задачах распознавания образов. В этом случае использование именно вычисляемых атрибутов оправдано сложностью самого процесса вычисления таких оценок, т.е. при обработке запроса к базе данных вычисление каких-то функций существенно увеличивает время работы системы. Использование вычисляемых атрибутов удобно для пользователя: вместо того, чтобы хранить в некотором отношении посторонние ключи нескольких таблиц, хранится один посторонний ключ, которому соответствует некоторый вычисляемый на этих ключах первичный атрибут. При использовании таких атрибутов уменьшаются размеры отношений и упрощаются внешние модели представления базы данных (представления пользователя).

Вычисление и обновление атрибута происходит автоматически при изменении атрибутов из набора, его определяющего. Вычисляемые атрибуты изменяются только в том случае, если меняется значение одного или нескольких атрибутов, определяющих данный вычисляемый.

Механизм обновления вычисляемых атрибутов такой же, как и при обновлении первичных ключей:

"В каждом отношении, содержащем вычисляемый атрибут как посторонний ключ, обновить его значение, если значение первичного ключа уже изменилось."

Автоматизация такого процесса каскадного обновления подразумевает использование словаря данных. Реляционные системы управления базами данных, такие как Informix, Oracle, и др., имеют средства создания и поддержки таких словарей.

Техника вычисляемых атрибутов применяется при создании вычисляемых первичных ключей.

В самом деле, вычисляемая функциональная зависимость есть обобщение понятия ключа. Хотя в качестве выражений ключа используют чаще всего такие:

$K=A_1 + A_2 + ... + A_n$, где

знак “+” означает конкатенацию значений атрибутов $A_1 , A_2 ,..., A_n$,

никаких ограничений на вид ключевого выражения нет. Поэтому вычисляемые атрибуты могут играть роль первичных ключей. Преимущество таких первичных ключей в сравнении с обычными составными ключами заключается в атомарности вычисляемых

ключей, а также в том, что такие вычисляемые атрибуты становятся уникальными идентификаторами отношений, а значит - несут смысловую нагрузку. Кроме того, известные алгоритмы поиска информации в базе данных (сравнение ключей, цифровой поиск с помощью B-деревьев, хэширование - см. в Кнут [24]) обеспечивают скорость поиска лучшую для ключа, состоящего из одного атрибута, чем для составного ключа.

4.1 Пример использования вычисляемых атрибутов

Проанализируем использование техники вычисляемых атрибутов на примере создания концептуальной модели базы данных фирмы - поставщика лекарств.

Построим концептуальную схему этой базы данных, используя модель ”Сущность-Связь” Чена, а затем преобразуем ее к реляционной.

Следуя ER-методике, вначале выделим независимые объекты предметной области:

“ЛЕКАРСТВО”,“СКЛАД”,“ПОСТАВЩИК”- основные сущности предметной области

и связи между сущностями : “СПРОС”, “ПРЕДЛОЖЕНИЕ”, “НАЛИЧИЕ”.

Поскольку объекты сложны, компоненты их внутренней структуры могут пересекаться.

Сущность “Лекарство” полностью описывает характеристики лекарства (см. рисунок 4.2):

“Русское Название” и **“Латинское Название”** определяют название, **“Состав”** - строка, описывающая составные части и их количество, **”Форма выпуска”** - в каком виде лекарство может поставляться, **“Тип Лекарства”** - к какой группе лекарств принадлежит (например, антибиотик, витамин, ...), **“Применение”** - способ употребления (например, внутреннее, внутримышечное, ...).

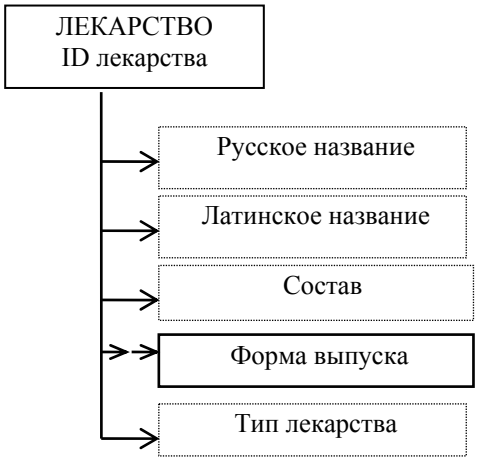

Рисунок 4.2 - ER-диаграмма для сущности “ЛЕКАРСТВО”

Как видно, каждое лекарство может иметь несколько форм выпуска, единственный состав, тип лекарства и уникальное латинское название. Для отображения этого факта создаем объект “ФОРМА ВЫПУСКА”. Структура этого объекта показана на рисунке 4.3.

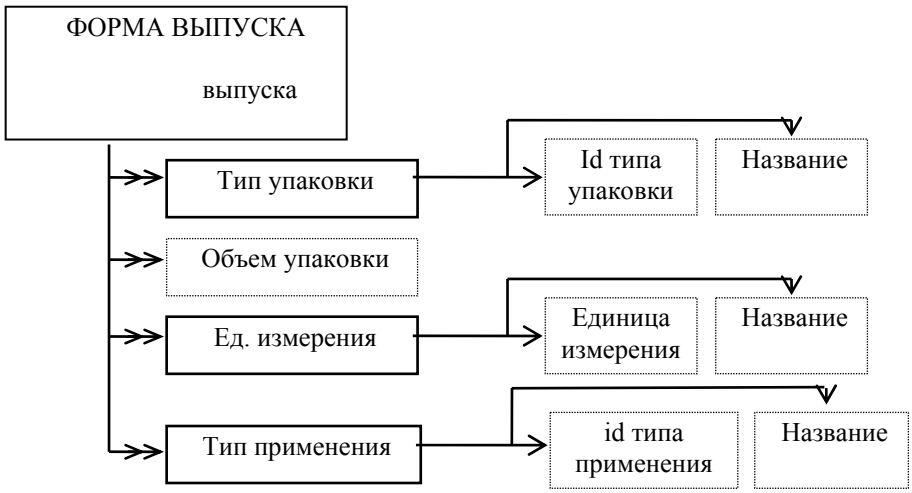

Рисунок 4.3 - ER- диаграммы сущности “ФОРМА ВЫПУСКА”

На рисунке 4.4 показана схема взаимодействия объектов в рамках сложного объекта “ЛЕКАРСТВО”

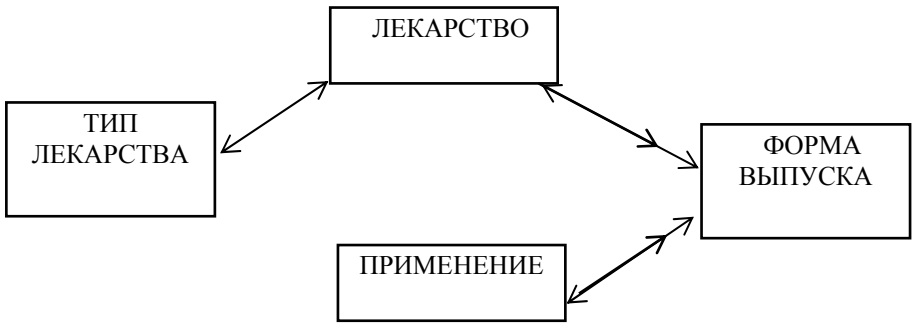

Рисунок 4.4 - Взаимодействие сущностей в рамках объекта “ЛЕКАРСТВО”

На рисунке 4.5 изображены остальные сущности предметной области.

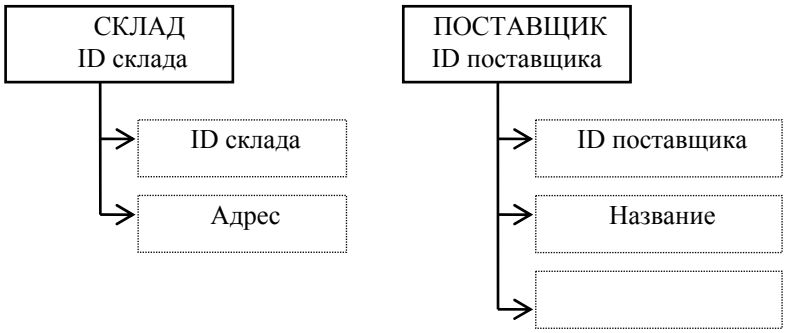

Рисунок 4.5. - ER - диаграммы сущностей “СКЛАД”, “ПОСТАВЩИК”

Изобразим ER-диаграммы связей между отношениями (см. рисунки 4.6, 4.7, 4.8).

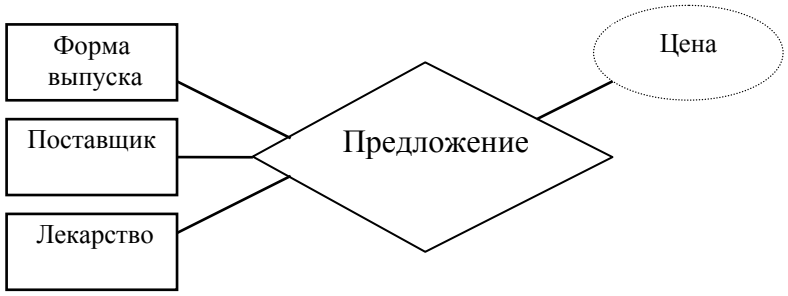

Рисунок 4.6 - ER-диаграмма связи “ПРЕДЛОЖЕНИЕ”

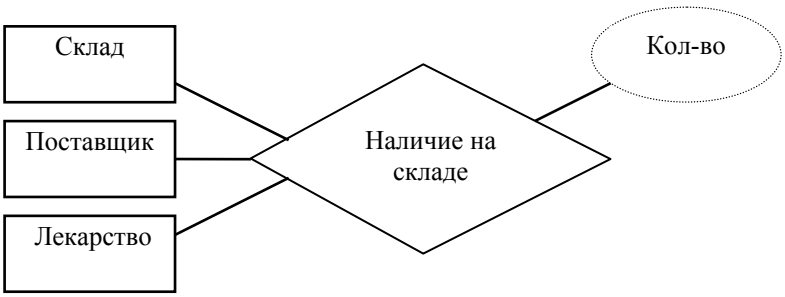

Рисунок 4.7 - ER-диаграмма связи “НАЛИЧИЕ НА СКЛАДЕ”

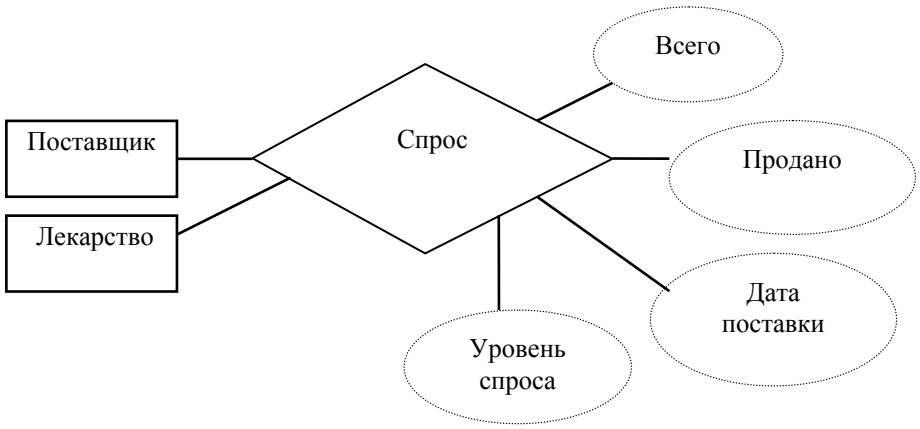

Рисунок 4.8 - ER - диаграмма связи “СПРОС”

Преобразуем ER - диаграммы в реляционные отношения. Для объектов необходимы уникальные идентификаторы. При выборе идентификаторов будем использовать вычисляемые атрибуты, где это необходимо. Пример схемы базы данных представлен на рисунке 4.9.

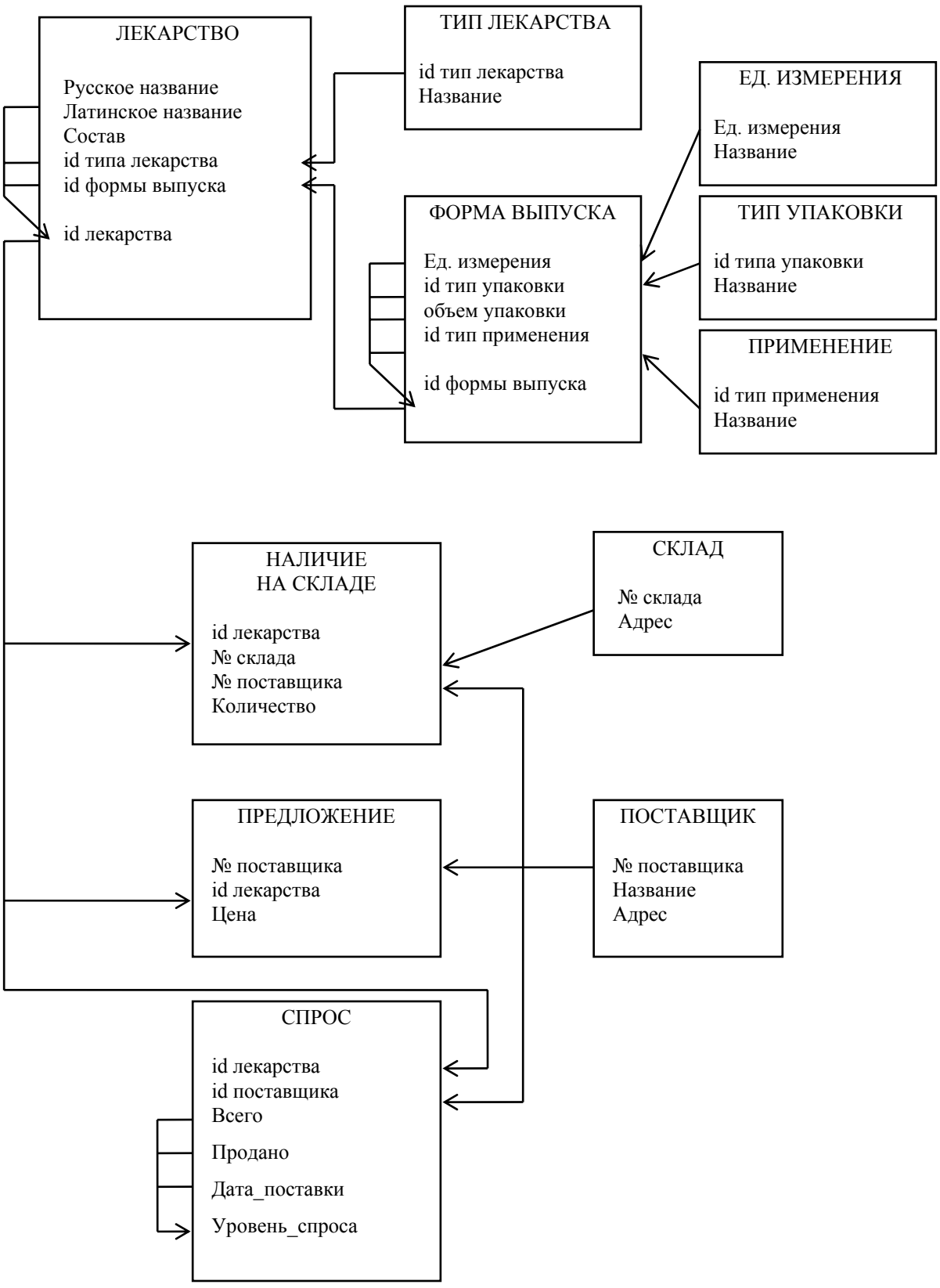

Рисунок 4.9 - Часть схемы базы данных фирмы - поставщика лекарств

Как видно, первичные ключи объектных отношений “ЛЕКАРСТВО” и “ФОРМА ВЫПУСКА” являются вычисляемыми атрибутами и хранятся вместе с отношениями:

“id лекарства” = f (“Латинское название”, “id типа лекарства”, “id формы выпуска”)

“id формы выпуска” = g (“Ед.измерения”, “id типа упаковки”, “id типа применения”)

“уровень_спроса” = h (“Всего”, “Продано”, “Дата_поставки”)

Функции f и g в данном случае представляют конкатенацию преобразованных к строковому виду значений этих атрибутов.

Функция h может иметь, например, такой вид:

h(*“Всего”,”Продано”,”Дата_поставки”*) =

=(*“Всего”-“Продано”*) / (*“Дата_поставки - Текущая_дата*)

Рассмотрим отношения, входящие в схему базы данных на рисунке 4.9. Все объектные отношения имеют первичные ключи. В сложных отношениях эти ключи являются вычисляемыми, они перечислены вместе с соответствующими схемами и являются атрибутами. Выделенные ключи не всегда описываются как атрибуты отношения (например, связыне отношения, такие как “НАЛИЧИЕ_НА_СКЛАДЕ”, имеет первичный ключ

{“№ поставщика”, “*id лекарства*”, “*количество*”}, но отдельного атрибута для хранения этого значения нет).

Перечисление первичного ключа как атрибута отношения, как уже говорилось выше, дает значительное удобство представления многозначных зависимостей между компонентами сложных объектов. Вместо того, чтобы в отношении “ЛЕКАРСТВО” перечислять все атрибуты, касающиеся отношения “ФОРМА ВЫПУСКА”, упоминается лишь его идентификатор (он же - первичный ключ в данном случае).Если предположить, что такое перечисление все же произошло, т.е. отношения “ЛЕКАРСТВО” имело вид:

“id лекарства”,”Русское название”, “Латинское название”, “Состав”, “id типа применения”, “Ед. измерения”, “id тип упаковки”, “Объем упаковки”,

то очевидна избыточность такого отношения: например, для каждого лекарства существует несколько разных упаковок, для каждой упаковки весь кортеж, кроме одного значения, повторяется. Как обсуждалось в главе 3.2 , избыточность данных - прямой путь к проблемам при их хранении и обновлении.

4.2 Выводы

Таким образом, применение вычисляемых атрибутов является оправданным по следующим причинам:

1. Это позволяет четко структурировать объекты и связи между ними. Можно использовать, например, такое правило: “Если сложный объект состоит из нескольких подобъектов, то имеет смысл сохранить идентификаторы этих подобъектов в вычисляемых атрибутах и использовать их для связи со сложным объектом”. Результат - неизбыточность данных.
2. При исполнении запросов к базе данных ведется сортировка и поиск нужных значений по ключу. Отдельно создаются индексные файлы для поиска по первичному ключу и для поиска по другим ключам отношения. Если ключ отношения хранится как вычисляемый атрибут, поиск значений по нему ведется цифровыми методами, например, с помощью сбалансированных B-деревьев, тогда как поиск по вторичным ключам использует более медленные алгоритмы (инвертированные списки и их модификации).
3. Процесс обновления отношений с вычисляемыми атрибутами может быть автоматизирован. Для этого необходимо предусмотреть способ хранения вычисляемых функциональных зависимостей, например, словарь базы данных. Тогда при любом обновлении базы процесс обновления вычисляемых атрибутов будет происходить без участия пользователя.

ЗАКЛЮЧЕНИЕ

В дипломной работе рассмотрено расширение описательной части реляционной модели данных, а именно, введения свойства вычисляемости атрибутов.

Исследование непротиворечивости такого расширения реляционной модели данных было основной целью дипломной работы. Была предложена к рассмотрению реляционная алгебра, основанная на классической реляционной алгебре, но позволяющая использование отношений с вычисляемыми атрибутами. Показана замкнутость этой алгебры, исследованы операции реляционной алгебры и выполнение их свойств над отношениями с вычисляемыми атрибутами, были найдены новые свойства этих операций. Выяснено, что расширенная алгебра обладает не меньшей выразительной силой, чем реляционная алгебра и реляционное исчисление. Обсуждалось применение реляционного исчисления кортежей к отношениям с вычисляемыми атрибутами.

Следующей целью исследования было изучение свойств отношений с вычисляемыми атрибутами. Результаты, касающиеся операций расширенной реляционной алгебры, позволили рассмотреть принципы оптимизации запросов к отношениям базы данных. При декомпозиции отношений введение понятия вычисляемой функциональной зависимости позволяет избежать некоторых недостатков метода декомпозиции, а именно, “внешних” зависимостей.

Третьей целью дипломной работы были предложения по использованию техники вычисляемых атрибутов на практике. Оказалось, что такие атрибуты могут успешно использоваться в реальных базах данных для описания и связывания сложных объектов, благодаря тому, что по своей природе они несут обобщенную характеристику того или иного объекта или связи. Пример в разделе 4.1 является характерной иллюстрацией методов использования вычисляемых атрибутов. Показаны задачи, в которых применение вычисляемых атрибутов вместе с обычными является удобным и простым решением: это базы данных со сложными зависимостями объектов и подобъектов, базы данных, в которых решения должны приниматься по большому количеству критериев в масштабе реального времени.

ПЕРЕЧЕНЬ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дейт К. Введение в системы баз данных. -М.: Наука, 1980.-463 с.
2. Ульман Дж. Основы систем баз данных. - М.: Финансы и статистика, 1983.-334 с.
3. Цикридзис Д., Лоховский Ф. Модели данных.- М.: Финансы и статистика, 1985.-343 с.
4. Codd E.F. A Relational Model of Data for Large Shared Data Banks. //Comm.ACM, -Vol.13,- No.6.-1970.-pp.377-387.
5. Schmid H., Swenson J. Relations And Entities.// Proc.IFIP TC-2 Working Conference on Modelling in DBMS, North-Holland.,-1975
6. Chen P.S. The ER-model: Towards a Unified View of Data.// ACM TODS,- Vol.1, 1976.-pp.9-36.
7. Codd E.F. Extending the Database Relational Model to Capture More Meaning. // ACM TODS,-Vol.4,-No.4,-1979.- pp.397-434.
8. Haskin R.L., Lorie R.A. On Extending the Functions of a Relational Database System. // Proc. ACM SIGMOD Conf.,Orlando.-1982.
9. Makinouchi I. A Consideration of Normal Forms of Not-Necessarily Normalized Relations in the Relational Data Model. // Proc. 3rd International Conference on Very Large Data Bases, Tokyo.-1977.
10. Smith J.M., Smith D.C.P. Database Abstractions: Aggregation and Generalization.//ACM TODS,-Vol.2,-No.2,-1977.-pp.105-133.
11. Parent C., Spaccapietra S. About Entities, Complex Objects and Object - Oriented Data Models.// Dept. d'Informatique, LBD, EPFL, Lausanne, -1996.
12. Codd E.F. Further Normalization of the Relational Data Model.// Database Systems, Courant Computer Science Symposia Series,-Vol.6.-1972.- pp. 33-64.
13. Casey R., Delobel C. Decomposition of a Data Base and the Theory of Boolean Switching Functions.// IBM Journal, R&D, - Vol. 17.- No 5.,-1973.
14. Armstrong W.W. Dependency Structures of Data Base Relationships.// IFIP Cong., Geneva, Switzerland.,-1979.- pp.580-583.
15. Armstrong W.W., Delobel C. Decompositions and Functional Dependencies in Relations. // ACM TODS.-Vol.5.-No.4.-1980.-pp.404-430.
16. Beery C., Bernstein P.A. Computational Problems Related to the Design of Normal Form Relational Schemas.// ACM TODS.-Vol.4.- No.1.-1979.-pp.30-59.
17. Bernstein P.A. Synthesizing Third Normal Form Relations from Functional Dependencies. // ACM TODS.- Vol.1.-No.4.-1976.-pp.277-298.

18. Maier D., Warren D. Incorporated Computed Relations in Relational Databases. // ACM SIGMOD Conf. -1981.-pp.176-187.

19. Ермолаев В.А. О расширении реляционной модели данных путем введения вычисляемых атрибутов.// Тезисы докладов научной конференции студентов и преподавателей ЗГУ. - Том 5.- Ч.1.-1995.-стр.69.

20. Codd E.F. A Database Sublanguage Founded on the Relational Calculus.//ACM SIGFIDET Workshop on Data Description,Access and Control,-November, 1971,- pp.35-61.

21. Codd E.F. Relational Completeness of Data Base Sublanguages. Database Systems, Rustin R., Prentice-Hall, N.-J.,-1972.- pp.65-98.

22. Мейер Д. Теория реляционных баз данных. -М.: Мир,-1987.-608 с.

23. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера.-М: Энергоатомиздат,-1988. - 480с.

24. Кнут Д. Искусство программирования, в 10 томах. Том 3. Сортировка и поиск. -М.: Мир,-1978.-850 с.