

Министерство образования Украины
Запорожский государственный университет

К защите допущен
Зав. кафедрой

(подпись)

(Ф.И.О.)

(дата)

ДИПЛОМНАЯ РАБОТА

НА ТЕМУ: ФОРМАЛЬНОЕ РАСШИРЕНИЕ РЕЛЯЦИОННОЙ МОДЕЛИ
ДАНЫХ СВОЙСТВАМИ ВЗАИМОДЕЙСТВИЯ ДАНЫХ
И ПРОГРАММНОГО КОДА

Выполнил Иванюк Марина Александровна

ст. группы 8223-2

(шифр)

(подпись и дата)

М.А. Иванюк

(И.О.Ф.)

Руководитель доцент

(должность)

(подпись и дата)

В.А. Ермолаев

(И.О.Ф.)

Нормоконтролер _____

(подпись)

А.В. Бакурова

(И.О.Ф.)

Запорожье
1998

РЕФЕРАТ

Дипломная работа: 49 страниц, 3 раздела, 4 рисунка, 2 таблицы, 2 приложения, 14 источников литературы.

Объект исследования: реляционная модель данных, в описательную часть которой добавлены понятия управляющих атрибутов I и II категорий.

Цель работы: проверка допустимости расширения реляционной модели данных понятиями управляющих атрибутов, разработка алгоритмов модификации значений управляющих атрибутов.

В дипломной работе использовались основные понятия следующих разделов математики: математическая логика, теория формальных систем, алгебра, а также теория реляционных баз данных.

В данной работе были получены результаты:

- доказано, что расширение реляционной модели данных новым видом атрибутов (управляющих атрибутов) не противоречит основным положениям реляционной модели данных;
- все операции классической реляционной алгебры применимы к отношениям с управляющими атрибутами;
- на базе полученных теоретических результатов разработаны алгоритмы модификации значений управляющих атрибутов.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ, ОТНОШЕНИЕ, РЕЛЯЦИОННАЯ АЛГЕБРА, ЦЕЛОСТНОСТЬ ДАННЫХ, УПРАВЛЯЮЩИЙ АТТРИБУТ, СИСТЕМА СЛОВАРЯ ДАННЫХ.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	6
1 ПОСТАНОВКА ЗАДАЧИ.....	10
1.1 Реляционная модель данных.....	10
1.1.1 Основные определения.....	11
1.1.2 Манипулирование реляционными данными.....	13
1.1.3 Обеспечение целостности данных.....	21
1.2 Система словаря данных.....	24
1.3 Постановка задачи расширения реляционной модели данных свойствами взаимодействия данных и программного кода.....	27
2 АНАЛИЗ РЕЛЯЦИОННЫХ ОПЕРАЦИЙ ПРИ ВВЕДЕНИИ УПРАВЛЯЮЩИХ АТТРИБУТОВ	31
2.1 Требования к управляющим атрибутам.....	31
2.2 Анализ системы манипуляции данными при введении управляющих атттрибутов.....	33
2.2.1 Булевы операции над отношениями с управляющими атттрибутами.....	33
2.2.2 Специальные реляционные операторы и отношения с управляющими атттрибутами.....	33
2.3 Выводы	37
3 АЛГОРИТМЫ ТРАНЗАКЦИЙ МОДИФИКАЦИИ ЗНАЧЕНИЙ УПРАВЛЯЮЩИХ АТТТРИБУТОВ.....	38
3.1 Алгоритм модификации атттрибута I категории.....	39
3.2 Алгоритм модификации атттрибута II категории.....	42
3.3 Выводы.....	43
ВЫВОДЫ.....	45
СПИСОК ССЫЛОК.....	46
ПРИЛОЖЕНИЕ А. Блок –схема алгоритма транзакции модификации значений управляющего атттрибута I категории	48
ПРИЛОЖЕНИЕ Б. Блок-схема алгоритма транзакции модификации значений управляющего атттрибута II категории	49

ВВЕДЕНИЕ

В настоящее время существуют три основных типа моделей 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, например, Дейтом [1].

Обобщая выводы многих исследователей моделей данных (Дейт [1], Ульман [2], Цикридис, Лоховски [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 указанных моделей только реляционная имеет в своей основе теоретические предложения, предопределяющие любые правильные реализации. Реляционная база данных с математической точки зрения - это конечный набор конечных отношений различной арности между заранее определенными множествами элементарных данных. Другими словами, реляционная база данных (точнее, любое ее состояние) - это конечная модель в смысле математической логики. Над отношениями модели можно осуществлять различные алгебраические операции. Тем самым теория реляционных баз данных становится областью приложений математической логики и современной алгебры и опирается на точный математический формализм.

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.

База данных представляется на внешнем, не зависящем от структуры ЭВМ уровне как совокупность повседневно встречающихся в человеческой практике двумерных таблиц, поиск и обработка информации в которых не зависят от организации хранения данных в памяти ЭВМ. Это представление значительно упрощает взаимодействие пользователя с банком данных и значительно повышает производительность труда.

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 расширился, что повлекло за собой разработку новых методов моделирования реляционных баз данных и усовершенствования классической реляционной модели 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 усовершенствования самой реляционной теории, что позволяет использовать новые свойства и понятия для создания более полного описания предметной области.

Так, например, Кодд [5] вводит новые типы атрибутов - идентификатор, компонент, ссылка, что позволяет описывать связи между компонентами сложного объекта. Хаскин и Лори (Haskin, Lorie [6]) предлагают различные

усовершенствования классической теории. Некоторые работы посвящены способам обхода одного из основных ограничений реляционной модели данных - неделимости (атомарности) отдельных значений данных. Макинучи (Makinouchi [7]) вводит понятие реляционно-значимых атрибутов, ослабляющих значимость атомарности данных. В работе Smith, Smith [8] 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.

При разработке информационных систем обычной практикой является использование значений некоторых атрибутов в качестве переключателей режимов работы приложения. Этот подход позволяет достичь определенной гибкости приложения, но вместе с тем устанавливает весьма жесткие связи между программами приложения и значениями управляющих атрибутов. Наличие такой связи, отрицательно влияющей на целостность информационного приложения и, следовательно, на его надежность, достаточно очевидно. Целостность приложения нарушается при любой модификации значения управляющего атрибута и присваивании значения, не соответствующего установленному в программе. Известны различные попытки решения задачи установления регулируемого соответствия между программами и управляющими данными. Стандартным решением является попытка уйти от решения проблемы и запретить все операции над кортежами, содержащими значения управляющих атрибутов. Этот подход неудачный, т.к. накладывает дополнительные ограничения на объекты модели данных на всех уровнях, вызывает необходимость введения специальных режимов заполнения соответствующих таблиц, что влечет за собой снижение надежности прикладной системы. Разработчики известной СУБД Oracle, например, пошли по пути размещения обрабатывающих программ в базу данных. Система Clipper позволяет разместить в базу данных так называемые блоки кода. Такой вариант решения нашей задачи плохо согласуется с требованиями создания компилируемой программной системы и вызывает необходимость хранить программы в несколько структурно различных объектах

приложения. Как можно видеть, ни одно из выше описанных решений не является идеальным с точки зрения технологии программирования и обеспечения действительного разделения программ и данных.

В работе В.А.Ермолаева [9] предложена технология проектирования программ приложения, в которой для управления прикладной системой вместо констант, равных значениям атрибутов, предложено использовать значения введенных глобальных переменных. Там же были введены определения управляющих атрибутов I и II категорий, которые описывают связи данных и программного кода.

Целью настоящей дипломной работы является исследование возможности введения в описательную часть реляционной модели данных понятий управляющих атрибутов. Основными задачами данной работы являются определение ограничений к управляющим атрибутам, анализ операторов реляционной алгебры при введении понятий управляющих атрибутов, а также разработка алгоритмов модификации значений таких атрибутов.

Понятия управляющих атрибутов не входят в описание реляционной модели данных. Новизна данной работы состоит в проверке допустимости введения этих понятий в классическую реляционную модель данных, исследовании применимости реляционной алгебры ко множеству отношений с управляющими атрибутами.

Актуальность работы: техника управляющих атрибутов дает возможность реализации алгоритмов модификаций значений атрибутов с автоматическими изменениями в программе приложения таким образом, чтобы не нарушалась целостность приложения, что качественно повышает гибкость и надежность приложения.

1 ПОСТАНОВКА ЗАДАЧИ

Как было показано во введении, при разработке информационных систем возникает проблема, связанная с наличием жесткой связи между программами и таблицами данных. С целью решения этой проблемы, в данной работе рассматривается введение понятий управляющих атрибутов, которые описывают взаимодействия данных с программой. Для построения формального аппарата описания этих взаимодействий необходима теория, в которой можно ввести эти новые понятия. В данном случае целесообразно рассматривать теорию реляционных баз данных, преимущества которой были описаны выше (см. введение).

В данной главе будут описаны теория и техника, специфичная для реляционной модели данных. Также будет рассмотрена структура системы словаря данных, который будет использован в качестве инструмента для поддержки описания управляющих атрибутов и их связей с программами. Затем будут формально описаны понятия управляющих атрибутов в терминах теории реляционных баз данных.

1.1 Реляционная модель данных

Реляционная модель данных была предложена Коддом [4] в 1970 г. 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[12]) 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”:

- а) **сущность** представляет собой любой отличимый объект с той степенью абстракции, какая требуется в данной задаче.
- б) **связь** представляет собой ассоциирование двух или более сущностей.

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

1.1.1 Основные определения

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 свойство объекта из предметной области. В дальнейшем будет употребляться слово атрибут с тем же смыслом.

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

t– кортеж – $\langle A_i : a_i \rangle, i = \overline{1, n}$

Пользуясь терминологией Мейера [11], сформулируем следующие определения.

Определение 1.1

Схема отношения R – конечное множество имен атрибутов $\{A_1, \dots, A_n\}$.

Каждому имени атрибута A_i ставится в соответствие множество D_i – конечное или бесконечное множество допустимых значений атрибута – домен:

$$\text{dom}(A_i) = D_i = \emptyset, \quad i = \overline{1, n}$$

Определение 1.2

Отношение r со схемой R , $r(R)$ – конечное множество отображений $\{t_1, \dots, t_p\}$, $t_j : R \rightarrow D$, где $D = D_1 \cup \dots \cup D_n$ – домен схемы R , t_j – кортеж, причем каждое отображение удовлетворяет условию:

$$t_j \in r \Rightarrow t_j(A_i) \in D_i, \quad i = \overline{1, n}, \quad j = \overline{1, p};$$

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 выполнение двух требований:

Требование 1. В каждом кортеже должно быть одно и то же 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.

Определение 1.3

Ключ отношения $r(R)$ – это подмножество $K \subseteq R$, такое, что для любых различных кортежей t_1 и t_2 из r выполняется $t_1(K) \neq t_2(K)$, и ни одно собственное подмножество $K' \subset K$ не обладает этим свойством.

Множество K является *суперключом* отношения r , если K содержит ключ отношения r .

Ключи, явно перечисленные вместе с реляционной схемой, называются *выделенными ключами*. Могут быть ключи, отличные от выделенных, они называются *неявными*. Иногда один из выделенных ключей объявляется *первичным*, остальные возможные ключи - *альтернативными*.

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[12] 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 [13] 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 $r(A_1A_2\dots A_n)$ – отношение, $d_i \in D_i = \text{dom}(A_i)$, $i = \overline{1, n}$.

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

$$\text{ADD } (r; A_1=d_1, A_2=d_2, \dots, A_n=d_n)$$

Цель операции добавления ясна – добавить описанный кортеж в определенное отношение. Результат операции может быть не согласован с целями операции по следующим причинам:

1. Добавляемый кортеж не соответствует схеме определенного отношения.
2. Некоторые значения кортежа не принадлежат соответствующим доменам.
3. Описанный кортеж совпадает по ключу с кортежем, уже находящимся в отношении.

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

$$\text{DEL } (r; A_1=d_1, A_2=d_2, \dots, A_n=d_n)$$

Операция **изменения** кортежа позволяет изменить часть кортежа. Для отношения r , введенного выше, при $\{C_1, C_2, \dots, C_p\} \subseteq \{A_1, A_2, \dots, A_n\}$ операция изменения имеет вид:

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

Операцию изменения можно выразить через операции добавления и удаления. Таким образом, все возможные ошибки операций добавления и удаления присущи и операции изменения: указанный в операции кортеж не

существует, изменения имеют неправильный формат или используемые значения не принадлежат соответствующему домену, или измененный кортеж имеет тот же ключ, что и кортеж, уже принадлежащий отношению.

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 одной и той же схемой могут быть рассмотрены как подмножества всех возможных кортежей с этой схемой. К таким двум отношениям можно применить булевы операции.

Если r и s – отношение со схемой R , то $r \cup s$, $r \cap s$ и $r - s$ также являются отношениями со схемой R .

Определение 1.4

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

Определение 1.5

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

Определение 1.6

$r - s$ - 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 1.7

Дополнение отношения $r(R)$ – множество, которое можно определить как разность $\bar{r} = dom(R) - r$.

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

Определение 1.8

Если $r(A_1A_2...A_n)$ – отношение и $D_i = dom(A_i)$, $i = \overline{1, n}$, то **активным доменом** атрибута A_i относительно r называется множество

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

Пусть $adom(R, r)$ - множество всех кортежей над атрибутами из R и их активными доменами относительно r .

Определение 1.9

Активным дополнением r является множество $\tilde{r} = adom(R, r) - r$.

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

Специальные реляционные операторы:

выбор, проекция, соединение, деление

Для выбора кортежей из отношения по некоторому условию используется операция выбора.

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

Определение 1.10

Оператор выбора $\sigma_{A=a}(r) = r'(R) = \{t \in r \mid t(A) = a\}$ – 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 1.11

Оператор проекции $\pi_X(r) = r'(X) = \{t(X) \mid t \in r\}$ – 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, $R \cup S = T$.

Определение 1.12

Оператор соединения отношений $r(R)$ и $s(S)$ – бинарный оператор, результатом которого является отношение $q(T)$:

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

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

Если $(R \cap S) = \emptyset$, то $r \bowtie s$ – декартово произведение r и s – изоморфное множество RS – кортежей.

Пусть $r(R)$, $s(S)$ – отношения, $S \subseteq R$, $R' = R - S$.

Определение 1.13

Оператор деления – бинарный оператор, результатом которого является отношение r' – частное от деления r на s

$$r'(R') = r \div s = \{t \mid \forall t_s \in s \exists t_r \in r : t_r(R') = t, t_r(S) = t_s\} - \langle r, \text{разделенное на } s \rangle$$

Другой способ сформулировать определение следующий:

$r \div s$ - максимальное подмножество r' множества $\pi_{R'}(r)$, такое, что $r' \div s$ содержится в r .

При обсуждении соединения в [11] было показано, что результат операции выбора можно получить, применив соединение с постоянным отношением.

Определение 1.14

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

$\langle c_1:A_1, c_2:A_2, \dots, c_n:A_n \rangle$ есть постоянный кортеж $\langle c_1, c_2, \dots, c_n \rangle$ над схемой $A_1 A_2 \dots A_n$.

Постоянное отношение с любым числом атрибутов может быть построено из постоянных отношений с одним кортежем и одним атрибутом с помощью операторов соединения и объединения.

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

Пусть r – отношение со схемой R , $A \in R$, $B \notin (R-A)$. Положим $R' = (R-A)B$.

Определение 1.15

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

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

Требуется, чтобы A и B имели один и тот же домен (имели одну и ту же область определения).

Пусть r – отношение со схемой R , $A_1, A_2, \dots, A_k \in R$,

$$B_1, B_2, \dots, B_k \notin R - (A_1, A_2, \dots, A_k), \quad \text{dom}(A_i) = \text{dom}(B_i), \quad i = \overline{1, k}.$$

Одновременное переименование атрибутов A_1, A_2, \dots, A_k в B_1, B_2, \dots, B_k записывается так:

$$\delta_{A_1, A_2, \dots, A_k \leftarrow B_1, B_2, \dots, B_k}(r)$$

До сих пор единственным сравнением значений домена, которым пользовались, было равенство. Можно сравнивать значения доменов, используя неравенство. Часто домены упорядочены, и в этих случаях имеют смысл операции $<, \leq, >, \geq, =, \neq$.

Для общего рассмотрения таких операций вводится множество Θ символов (знаков) бинарных отношений над парами доменов. Если θ – знак сравнения, а A и B – атрибуты, то говорят, что A θ -сравним с B , если знаку θ сопоставлено бинарное отношение в $\text{dom}(A) \times \text{dom}(B)$.

Сравнения могут быть использованы для обобщения операторов выбора и соединения, в которых участвовало только равенство.

Пусть r – отношение со схемой R , $A \in R$, $a \in \text{dom}(B)$ – константа, A θ -сравним с B .

Определение 1.16

Оператор расширенного выбора создает отношение

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

Допускаются также сравнения между двумя атрибутами:

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

Пусть $r(R)$ и $s(S)$ – два отношения, причем $R \cap S = \emptyset$, и пусть $A \in R$ и $B \in S$ θ -сравнимы для $\theta \in \Theta$.

Определение 1.17

Оператор θ – соединения создает отношение $q(RS)$, такое, что

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

Допускается несколько сравнений над отношениями, например:

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

Эквисоединение – частный случай θ –соединения, когда $\Theta = \{=\}$.

Теоретико-множественные операции, операторы выбора, естественного соединения, деления, переименования и θ –соединения 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 и получить в качестве результата единственное отношение.

Пусть U – множество атрибутов, называемое *универсумом*,

D - множество доменов,

$dom: U \rightarrow D$ полная функция из U в D ,

$R = \{R_1, R_2, \dots, R_p\}$ - множество различных схем отношений, где $R_i \subseteq U$,

$d = \{r_1, r_2, \dots, r_p\}$ - множество всех таких наборов отношений, что

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

Пусть Θ – множество бинарных отношений над доменами из D , содержащее по крайней мере $=$ и \neq для любого домена.

Определение 1.18

Реляционной алгеброй над U, D, dom, R, d и Θ называется семиместный кортеж $V = (U, D, dom, R, d, \Theta, O)$, где O – множество операций объединения, пересечения, разности, активного дополнения, проекции, естественного

соединения, деления, переименования, использующего атрибуты из U и отношения из Θ .

Алгебраическим выражением над B называется любое выражение, правильно построенное из отношений, принадлежащих d , и постоянных отношений со схемами из U , использующих операторы 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 от другого, поскольку эта отличимость - одно из основных требований при описании предметной области задачи. В реляционной модели для этого используется *механизм ключей* (см. 1.1.1)

Пусть $K = \{B_1, B_2, \dots, B_m\}$ - ключ отношения r со схемой R .

Сформулируем два свойства ключа.

Первое свойство - *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_2, \dots, B_m не может быть исключен из K без нарушения свойства 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.

Таким образом, каждое отношение имеет свой 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 должно обеспечивать целостность данных.

Сформулируем правила целостности для реляционной модели данных (см. также[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 F_K , соответствующий первичному ключу P_K какого-либо объектного отношения r_1 , то каждое значение F_K в r_2 , должно либо:

а) быть равным значению P_K в некотором кортеже r_1 ,

либо

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

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

Таким образом, основными компонентами реляционной модели данных являются:

Структура данных

1. Домены, n-арные отношения (атрибуты, кортежи).
2. 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

1. Реляционная алгебра В и реляционное присваивание

1.2 Система словаря данных

Одно из главных назначений системы с базой данных - возможность создания условий для коллективного использования данных. Не менее важно предоставить пользователям достоверные данные. Наиболее удачным решением задачи обеспечения достоверности, минимальной избыточности и контроля использования данных является применение Словаря Данных . Такое решение, как правило, упрощает разработку и повышает эффективность системы.

Словарь данных - это централизованное хранилище сведений об объектах, составляющих их элементах данных, взаимосвязях между объектами, их источниках, значениях, использовании и форматах представления.

Словарь данных сам по себе является базой данных, а именно базой данных, которая содержит «данные о данных» или, другими словами, описания других объектов в системе.

Рассмотрим структуру системы словаря данных, приведенную в [9].

Система должна содержать средства для описания модели данных предметной области, включающие:

- описание объектов, атрибутов данных;
- описание отношений, использующие механизм цепочек связи между первичными и порожденными атрибутами;
- описание стратегий обработки таблиц;
- описание пользовательских представлений данных;
- описание управляющих атрибутов и связей между их значениями и программами приложения.

Система должна предоставлять полный реляционный набор программных примитивов манипуляции данными, обладающий следующими свойствами:

- инвариантностью по отношению к структурам обрабатываемых объектов данных;
- инвариантностью по отношению к набору стратегий обработки таблиц данных;
- обеспечением механизма каскадирования при обработке значений первичных и порожденных атрибутов;
- обеспечением механизмов обработки транзакций.

Система должна содержать как автономные, так и встраиваемые в приложение программные средства обеспечения целостности данных.

Система должна обеспечивать набор встраиваемых в компилируемое приложение управляющих блоков для обеспечения программных компонент информацией о модели данных, связях управляющих данных с программами, стратегиях обработки и пользовательских представлениях данных.

На рисунке 1.1 схематически изображена структура системы словаря данных.

Система словаря данных базируется на некоторых расширенных базовых понятиях теории реляционных баз данных.

Определение 1.19

Схемой R отношения r есть 4-х местный кортеж $R = \{I, A, D, T\}$,

где I - заголовок схемы R отношения,

$A = \{A_1, A_2, \dots, A_m\}$ - множество дескрипторов атрибутов $A_j = \{n_j, t_j, l_j\}$,
 $j=1, \dots, m$,

$D = \{D_1, D_2, \dots, D_m\}$ - множество доменов атрибутов $D_j = \text{dom}(a_j)$,
 $j=1, \dots, m$,

$T = \{T_1, T_2, \dots, T_m\}$ - множество типов атрибутов,

m - количество атрибутов в отношении.

Определение 1.20

Дескриптором A_j атрибута a_j отношения r называется 3-х местный кортеж

$$A_j = \{n_j, t_j, l_j\},$$

где n_j - имя атрибута,

t_j - тип данного,

l_j - спецификация длины атрибута.

Определение 1.21

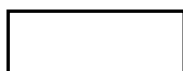
Тип атрибута a_j отношения r - это 2-х местный кортеж $T_j = \{v_j, o_j\}$,

где v_j - вид атрибута, определяющий его участие в формировании ключей отношения: 'U' - уникальный ключ, 'K' - ключ, 'O' - неключевой атрибут, ...,

o_j - правило формирования значения атрибута.

Таблица словаря

целостности



Описание информационных объектов

Подсистема контроля

данных

- Проверка наличия таблицы
- Проверка соответствия структуры таблицы модели данных
- Проверка наличия соответствующих таблице индексов
- Проверка соответствия ключевых выражений структуре таблицы
- Проверка содержимого индексного файла

- Описание индексов информационных объектов

- Атрибу Расширенное описание атрибутов и обратных связей в отношениях

- Связи Описание связей между первичными и порожденными атрибутами

- Фильтр Описание пользовательских представлений объектов, фильтры

- Связи с Описание связей управляющих атрибутов и программ

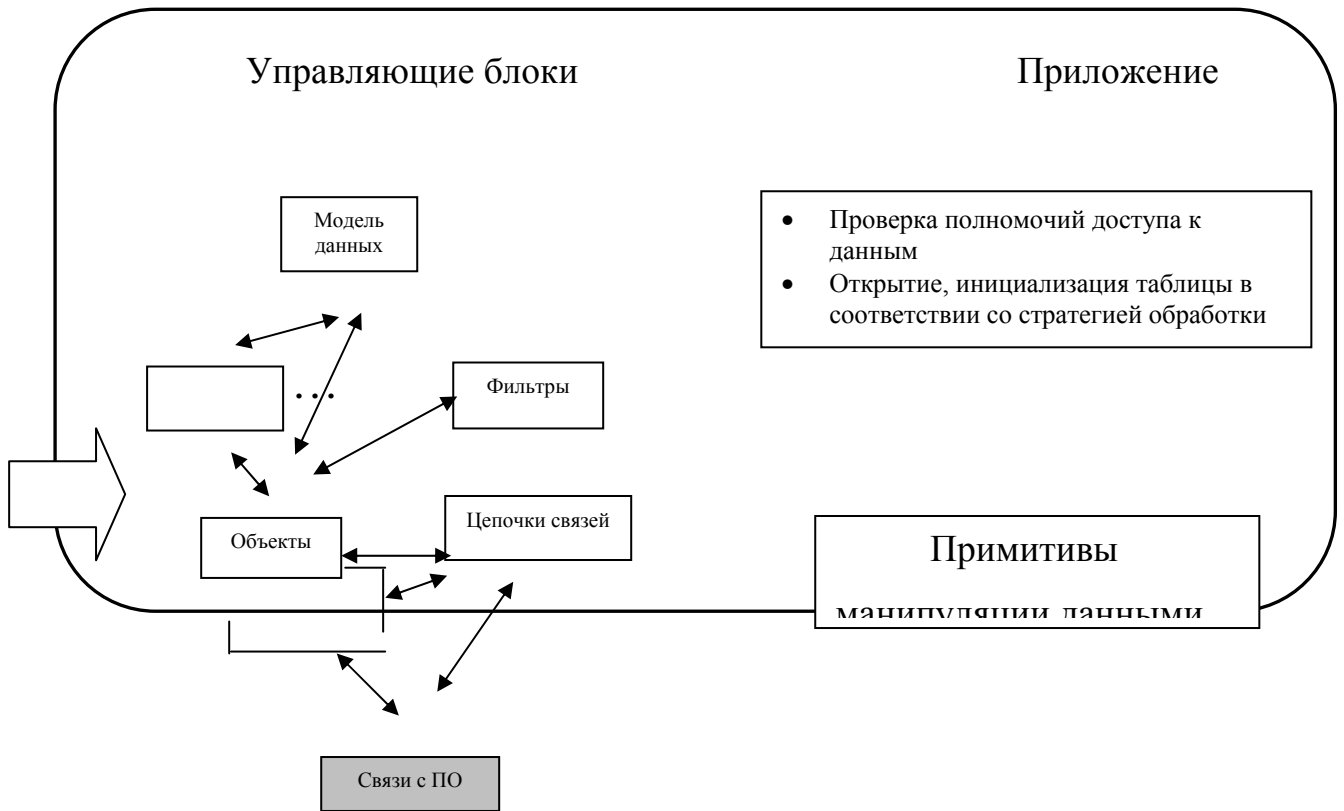


Рисунок 1.1 - Структура системы словаря данных

1.3 Постановка задачи расширения реляционной модели данных свойствами взаимодействия данных и программного кода

Как было показано в введении, один из важных интерфейсов информационной системы, оказывающей существенное влияние на гибкость приложения - это интерфейс связи программ приложения с управляющими данными. В случае существования подобной связи в традиционном приложении модификация значения управляющего атрибута вызывает непредсказуемые изменения в алгоритмах обработки. Обычно после каждого такого изменения приходится выполнять соответствующие модификации программ.

Решение задачи повышения надежности программного обеспечения влечет необходимость избавиться от жесткой привязки прикладных программ к данным. Для этого В.А. Ермолаевым в [9] предложена следующая технология проектирования программ приложения:

- каждому значению C_j^k атрибута a_j^i , связанному с программой, ставится в соответствие глобальная переменная V_k , значение которой в процессе работы прикладной системы равно соответствующему значению управляющего атрибута;
- для управления прикладной системой вместо констант, равных значениям управляющих атрибутов C_j^k , используют значения введенных глобальных переменных V_k ;
- при модификации значений управляющих атрибутов C_j^k автоматически изменяются значения соответствующих глобальных переменных V_k .

Атрибуты, связанные с программным кодом информационной системы, квалифицируются в [9] как управляющие.

Следует отметить, что предложенная технология лишь отчасти снимает ограничения, связанные с наличием жесткой функциональной связи между программами приложения и таблицами данных. Основной проблемой, которая остается нерешенной, является наличие связи вида один к одному между некоторым подмножеством значений атрибутов определенного кортежа таблицы

данных и идентификатором соответствующей глобальной переменной приложения и ее значением.

Управляющие атрибуты, значения которых формируют указанное подмножество, разделяются в [9] по виду связи с приложением на две категории:

- I - значение атрибута соответствует значению глобальной переменной приложения;
- II - значение атрибута соответствует идентификатору глобальной переменной приложения.

Управляющий атрибут I категории, таким образом, имеет значение, равное значению связанной с ним переменной в программе. Следовательно, для сохранения целостности информационной системы алгоритм транзакции модификации значения такого атрибута должен выполнять изменения значения переменной в программе.

Характерной особенностью атрибутов II категории является наличие взаимно однозначного соответствия между значением атрибута и идентификатором переменной в программе. Примером использования атрибутов II категории может служить отношение – каталог режимов обработки данных, в котором каждый кортеж соответствует некоторой подпрограмме обработки, определяемой, в свою очередь, ее идентификатором.

Для построения формального аппарата описания взаимодействия данных и программного кода в [9] приведены следующие формальные определения управляющих атрибутов I и II категорий.

Пусть V - множество переменных, связанных с некоторыми значениями некоторых управляющих атрибутов:

$$V = \{V_1, V_2, \dots, V_p\}$$

Определение 1.22

Атрибут a_j^i отношения r^i со схемой R^i будем называть **управляющим атрибутом I категории** (связанным по значению), если в отношении r^i существует кортеж C^k такой, что $(c_j^k)_{a_j^i} = V_l$,

где $(c_j^k)_{a_j^i}$ - значение атрибута a_j^i в кортеже C^k ,

$$V_l \in V, l = 1, \dots, p.$$

Определение 1.23

Атрибут a_j^i отношения r^i со схемой R^i будем называть **управляющим атрибутом II категории** (связанным по идентификатору), если в отношении r^i существует кортеж C^k и существует взаимно однозначное семантически определенное приложением отображение f такое, что $f\left(\left(c_j^k\right)_{a_j^i}\right) = \text{id}(V_l)$,

где $\text{id}(V_l)$ - идентификатор переменной V_l ,

$$V_l \in V, l = 1, \dots, p.$$

Таким образом, мы рассмотрели расширение реляционной модели данных понятиями управляющих атрибутов I и II категорий.

В связи с этим, можно сформулировать следующие задачи данной работы:

1. Определить ограничения, накладываемые на значения управляющих атрибутов, которые позволили бы корректно описывать объекты модели данных, оставаясь в рамках реляционного подхода.
2. Провести анализ операций реляционной алгебры с учетом управляющих атрибутов с целью проверки возможности введения понятий управляющих атрибутов в классическую реляционную модель данных.

3. Разработать алгоритмы модификации значений управляющих атрибутов, которые позволили бы автоматически производить изменения в программе, не нарушая целостности приложения.

2 АНАЛИЗ РЕЛЯЦИОННЫХ ОПЕРАЦИЙ ПРИ ВВЕДЕНИИ УПРАВЛЯЮЩИХ АТТРИБУТОВ

В предыдущей главе были приведены определения управляющих атрибутов I и II категорий.

В этой главе будут рассмотрены некоторые ограничения на значения управляющих атрибутов. Также будут рассмотрены реляционные операции над отношениями с управляющими атрибутами и их свойства.

2.1 Требования к управляющим атрибутам

Рассмотрим требования к управляющим атрибутам, вытекающие из правил общей целостности данных.

Требование 1.

Управляющие атрибуты I и II категорий должны иметь вид $a_j^i = U$ - уникальный ключ и не должны иметь неопределенных значений.

Это требование означает, что управляющий атрибут должен быть выделен в отдельное отношение, в котором он будет являться уникальным ключом, и в котором информация о значениях этого атрибута не будет зависеть от информации о другом атрибуте. Это вызвано процессом нормализации, при котором элементы данных группируются в таблицы, представляющие объекты и их взаимосвязи. Теория нормализации основана на том, что определенный набор отношений обладает лучшими свойствами при добавлении, изменении и удалении данных, чем все остальные наборы отношений, с помощью которых могут быть представлены те же данные. В [14] продемонстрированы примеры, в которых обосновывается выделение некоторых атрибутов в отдельное отношение.

Управляющий атрибут может входить также и в другие отношения, но в них он не будет управляющим, т.е. не будет связанным с программой и при

изменении его значений не надо будет производить соответствующую модификацию программы.

Требование 2.

При изменении значений управляющих атрибутов II категории, как показано в [10], алгоритм выполнения транзакции модификации такого значения должен учитывать следующее ограничение для нового значения атрибута:

$$\forall (c_j^k)_{a_j^i} \quad \exists f: f\left((c_j^k)_{a_j^i}\right) = \text{id}(V_l)$$

где $(c_j^k)_{a_j^i}$ - новое значение атрибута в кортеже C^k , $V_l \in V, l = 1, \dots, p$.

Для каждого отношения, имеющего управляющие атрибуты II категории, построим отображение в виде реляционного отношения, структура которого приведена на рисунке 2.1.

Идентификатор переменной	Имя атрибута	Значение
V_1	
	
V_1	a_j^i	$(c_j^k)_{a_j^i}$
	
V_p	

Рисунок 2.1- Структура реляционного отношения f

Эти требования дают возможность корректно описывать объекты модели данных, оставаясь в рамках реляционного подхода.

2.2 Анализ системы манипуляции данными при введении понятий управляющих атрибутов

2.2.1 Булевы операции над отношениями с управляющими атрибутами

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. подраздел 1.1.2).

При соблюдении условия совместимости, введение свойства атрибута, как управляющего не меняет результатов применения булевых операций (объединение, пересечение, разность, дополнение), поскольку схема R при этом не меняется.

2.2.2 Специальные реляционные операторы и отношения с управляющими атрибутами

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

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

Пусть r отношение со схемой R , $A^* \subseteq R$ - управляющий атрибут.

Оператор выбора 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 ,

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

Т.к. результатом операции выбора является отношение, которое представляет собой подмножество кортежей отношения r , то очевидно, что в нем сохраняется ключ. Следовательно, применяя оператор выбора к отношению с управляющим атрибутом A^* , мы видим, что этот атрибут сохраняет вид $A^* = U$. Кроме того, атрибут A^* в отношении $r' = \sigma_{A\theta a}(r)$ останется управляющим, т.е. связанным с программным кодом.

Пусть $\Theta = \{=\}$. Тогда, если выбор ведется по управляющему атрибуту, то результатом будет отношение, содержащее не более одного кортежа. Это следует из того, что управляющий атрибут является уникальным ключом.

Рассмотрим свойства оператора.

Операторы выбора коммутативны относительно их композиции.

Пусть $r(R)$ – отношение, A и B – атрибуты в R , и пусть $a \in \text{dom}(A)$, $b \in \text{dom}(B)$.

Тогда имеет место тождество:

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

Оператор выбора дистрибутивен относительно бинарных булевых операций:

$$\sigma_{A=a}(r \gamma s) = \sigma_{A=a} r \gamma \sigma_{A=a} s,$$

где $\gamma = \cap, \cup$ или $-$, а r и s – отношения над одной и той же схемой R .

Эти свойства верны также и для отношений с управляющими атрибутами.

Оператор проекции и его свойства

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

$$\pi_X(r) = r'(X) = \{t(X) \mid t \in r\}$$

Если $R = A^* \cup B$, A^* – управляющий атрибут, то рассмотрим разные проекции отношения $r(R)$.

Если $X = B$, то $\pi_X(r)$ не содержит управляющего атрибута и оператор проекции не теряет своего смысла.

Если $X = A^*$, то $\pi_X(r)$ будет содержать управляющий атрибут, причем останется уникальным ключом.

Проверим свойства проекции.

Проекция коммутует с выбором, когда атрибут или атрибуты для выбора находятся среди атрибутов множества, на которое осуществляется проекция. Если $A \in X$, $X \subseteq R$ и r - отношение со схемой R , то

$$\pi_X(\sigma_{A=a}(r)) = \sigma_{A=a}(\pi_X(r))$$

Установленное тождество может не выполняться, когда $A \notin X$.

Очевидно, что это свойство выполняется и для отношения с управляющим атрибутом. Причем отношение $\pi_X(\sigma_{A=a}(r))$ или $\sigma_{A=a}(\pi_X(r))$ содержит не более одного кортежа (см. выше).

Оператор соединения

Если даны два отношения $r(R)$ и $s(S)$, то оператор соединения комбинирует эти отношения по всем их общим атрибутам, а в общем случае θ -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

" $t_r(A) \theta t_s(B)$ " 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.

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

$$A \in R, B \in S, \text{dom}(A) = \text{dom}(B).$$

Частными случаями является эквисоединение ($\Theta = \{=\}$) и естественное соединение ($r \bowtie s$ - требует равенство доменов и имен атрибутов, по которым ведется соединение).

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

Если атрибут B является ключом, то управляющий атрибут в отношении $r' = r [A^* \theta B] s$ останется уникальным ключом. Если B не является ключом, то управляющий атрибут перестанет быть ключом. Это будет не столь важным, т.к. атрибут A^* уже не будет управляющим.

Оператор деления

Пусть даны два отношения $r(R)$ и $s(S)$, где $S \subseteq R$. Оператор деления $r \div s$ дает отношение $q(Q)$ ($Q = R - S$) такое, что каждый кортеж этого отношения удовлетворяет набору свойств, описанных отношением $s(S)$.

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

Можно сформулировать определение по-другому:

$r \div s$ - максимальное подмножество q множества $\pi_Q(r)$, такое, что $q \div s$ содержится в r . Соединение в этом случае является декартовым произведением.

Рассмотрим отдельно случай, когда отношение $r(A^*, X)$ содержит управляющий атрибут.

Пусть, $q = q(A^*)$ тогда $r \div q = V(X)$ - это отношение не содержит управляющего атрибута.

Если $s = S(X)$, то $r \div s = W(A^*)$ - в этом отношении управляющий атрибут остается. Поскольку результатом деления является отношение, атрибуты и кортежи которого являются подмножествами соответственно атрибутов и кортежей исходного отношения, то в нем атрибут A^* остается уникальным ключом, причем не теряет своего свойства как управляющего.

2.3 Выводы

В данной главе были описаны ограничения, накладываемые на значения управляющих атрибутов. Также был проведен анализ операций реляционной алгебры с учетом нового свойства атрибутов. Из всего этого можно сделать следующие выводы:

1. Введенные ограничения позволяют корректно описывать объекты модели данных, оставаясь в рамках реляционного подхода.
2. Введение нового свойства атрибутов не противоречит базовым понятиям реляционной модели данных.
3. Все операции классической реляционной алгебры применимы к отношения с управляющими атрибутами.
4. Все свойства этих операций также выполняются при введении понятия управляющего атрибута.

3 АЛГОРИТМЫ ТРАНЗАКЦИЙ МОДИФИКАЦИИ ЗНАЧЕНИЙ УПРАВЛЯЮЩИХ АТТРИБУТОВ

В главе 1 было рассмотрено расширение реляционной модели данных определениями управляющих атрибутов I и II категорий. Эти определения были введены для описания взаимодействий данных и программного кода. Значения управляющих атрибутов используются в качестве переключателей режимов работы приложения. При любой модификации этих значений и присваивании значения, не соответствующего установленному в программе, нарушается целостность приложения. Поэтому, для управления прикладной системой вместо констант, равных значениям управляющих атрибутов, необходимо использовать значения введенных глобальных переменных. В связи с этим, необходимо разработать алгоритмы модификации значений атрибутов, которые позволили бы автоматически изменять значения соответствующих глобальных переменных.

Для того, чтобы реализовать такие алгоритмы нам необходима некоторая надстройка над информационной системой, в которой можно бы было описать управляющие атрибуты и связи между их значениями и программами приложения. В качестве такой надстройки можно использовать систему словаря данных (см. подраздел 1.2). На рисунке 1.1 была представлена структура системы словаря данных. На нем выделен объект "Связи с ПО", который как раз и описывает связи управляющих атрибутов с программой.

Для обеспечения целостной модификации кортежей, содержащих значения управляющих атрибутов, необходимо в систему словаря данных добавить две управляющие таблицы, содержащие информацию о переменных приложения, управляющих атрибутов I и II категорий, соответствиях значений атрибутов и переменных. Вид таких таблиц приведен на таблицах 3.1 и 3.2. Для описания управляющих атрибутов I категории используется таблица 3.1, а для атрибутов II категории - таблица 3.2.

Таблица 3.1

Идентификатор переменной	Имя таблицы	Имя атрибута I категории
V_1	
.....		
V_1	r^i	a_j^i
.....		
V_p	

Таблица 3.2

Идентификатор переменной	Имя таблицы	Имя атрибута II категории	Значение атрибута
$id(V_1)$		
.....			
$id(V_1)$	r^i	a_j^i	$\binom{c_j^k}{a_j^i}$
.....			
$id(V_p)$		

В разделе 2.1 были описаны требования к управляющим атрибутам. Второе требование к управляющему атрибуту II категории означает, что этот атрибут не может получить значение, не содержащееся в таблице 3.2, а эта таблица обязана содержать все возможные значения этого атрибута.

Все описанные выше требования должны быть учтены при реализации алгоритма транзакции модификации значения управляющего атрибута.

3.1 Алгоритм модификации атрибута I категории

Пусть дано: отношение $r^i(R^i)$,

a_j^i - j -й атрибут i -го отношения, управляющий атрибут I категории,

c_j^k - значение атрибута a_j^i в кортеже C^k ,

Необходимо выполнить следующую операцию:

$$CH (r^i; a_j^i = c_j^k; c_j^k = \binom{c_j^k}{a_j^i}),$$

которая изменяет значение c_j^k на значение $\binom{c_j^k}{a_j^i}$.

Условно можно разделить выполнение алгоритма на три этапа.

Этап 1. На этом этапе проверяется выполнение требования 1 (см. подраздел 2.1). Если новое значение $(c_j^k)'$ равно какому-либо другому значению атрибута a_j^i в отношении r^i , первое требование не выполняется и необходимо перейти к этапу 2. Если же требование выполняется, то необходимо перейти к этапу 3.

Этап 2. Поскольку первое требование не выполнено, то необходимо на этом этапе выдать сообщение об ошибке. Но тогда может возникнуть проблема, т.к. возможна ситуация когда два значения одного атрибута меняются друг с другом местами. Например, необходимо выполнить следующие две операции изменения, при которых меняются местами два значения c_j^k и c_j^l управляющего атрибута a_j^i :

$$\text{СН} (r^i; a_j^i = c_j^k; c_j^k = c_j^l)$$

$$\text{СН} (r^i; a_j^i = c_j^l; c_j^l = c_j^k)$$

Тогда при выполнении первой операции изменения алгоритм прекратит работу и выдаст сообщение об ошибке. Следовательно, алгоритм модификации должен учитывать такую ситуацию. Поэтому *на этом этапе необходимо проверить, будет ли затем замена c_j^l на c_j^k* (например, запросить об этом пользователя). Если замены не будет, тогда уже просто необходимо выдать сообщение об ошибке. В обратном случае можно перейти к выполнению этапа 3.

Этап 3. На этом этапе обеспечивается автоматическая модификация значения глобальной переменной в программе, связанной со значением управляющего атрибута, и собственно модификация самого значения атрибута a_j^i .

1. Необходимо в управляющей таблице 3.1 найти имя атрибута a_j^i , значение которого изменяется.
2. Определить соответствующую этому значению переменную V_k .
3. Найти переменную V_k в программе и изменить ее значение на новое:

$$V_k = (c_j^k)'$$

4. Изменить значения всех порожденных атрибутов равные c_j^k на $(c_j^k)'$.
5. Изменить значение атрибута равное c_j^k на $(c_j^k)'$.

Выполнение этапа 3 (за исключением шага 4) схематически изображено на рисунке 3.1. Блок-схема всего алгоритма приведена в приложении А.

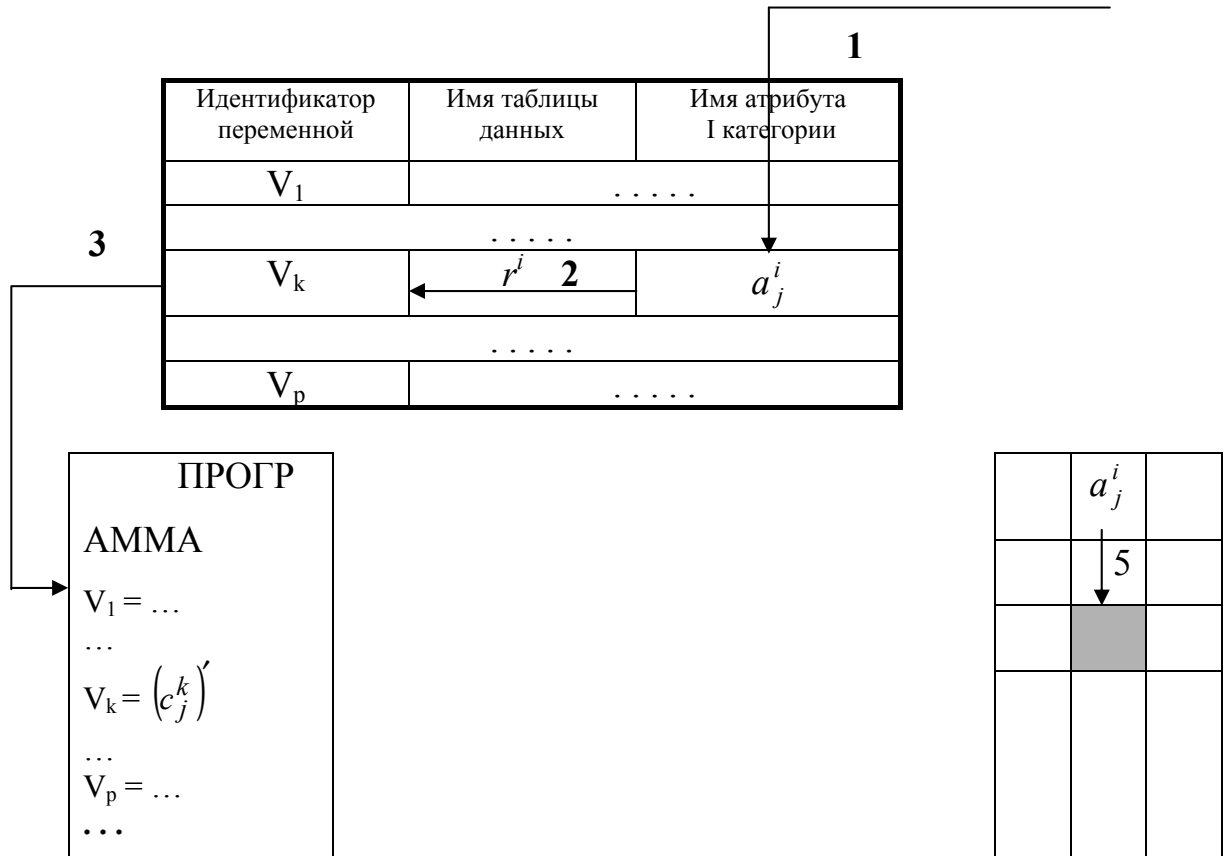


Рисунок 3.1 – Автоматическая модификация значения атрибута и значения соответствующей переменной

3.2 Алгоритм модификации атрибута II категории

Этот алгоритм будет отличаться от предыдущего, т.к. во-первых, атрибут II категории имеет другой вид связи с переменными приложения (связан не по значению, а по идентификатору), во-вторых, существует дополнительное ограничение на атрибут (см. требование 2 в подразделе 2.1).

Пусть дано: отношение $r^i(R^i)$,

a_j^i - j -й атрибут i -го отношения, управляющий атрибут II категории,

c_j^k - значение атрибута a_j^i в кортеже C^k .

Необходимо выполнить следующую операцию:

$$\text{CH} (r^i; a_j^i = c_j^k; c_j^k = (c_j^k)'),$$

которая изменяет значение c_j^k на значение $(c_j^k)'$.

Первые два этапа этого алгоритма аналогичны первым двум этапам предыдущего алгоритма, т.е.

Этап 1. Проверить выполнение требования 1. Если выполняется перейти к этапу 3, иначе к этапу 2.

Этап 2. Проверить будет ли взаимозамена двух значений одного атрибута c_j^k и c_j^l . В случае отрицательного ответа выдать сообщение об ошибке, иначе перейти к этапу 3.

Этап 3. На данном этапе модифицируются значения атрибута и автоматически производятся изменения в программе.

1. Найти в управляющей таблице 3.2 имя атрибута a_j^i , значение которого изменяется.
2. В колонке значений таблицы 3.2 найти значение c_j^k и определить соответствующий ему идентификатор переменной V_k .
3. На этом шаге проверяется выполнение требования 2, т.е. необходимо в

колонке значений таблицы 3.2 найти новое значение $(c_j^k)'$. Если оно не содержится в таблице, то второе требование не выполняется, и необходимо выдать сообщение об ошибке. В случае, если это значение находится в таблице перейти к шагу 4.

4. Определить соответствующий новому значению $(c_j^k)'$ идентификатор переменной V_1 .
5. Изменить соответствие значения управляющего атрибута переменной в программе. Для этого необходимо в программе заменить идентификатор переменной V_1 на идентификатор переменной V_k .
6. Изменить значения всех порожденных атрибутов равные c_j^k на $(c_j^k)'$.
7. Изменить значение атрибута равное c_j^k на $(c_j^k)'$.

Выполнение этапа 3 (за исключением шага 6) схематически изображено на рисунке 3.2. Блок-схема всего алгоритма представлена в приложении Б.

3.2 Выводы

Таким образом, в данной главе:

1. Для реализации алгоритма модификации были предложены структуры управляющих таблиц словаря данных, содержащие сведения об управляющих атрибутах и связях с программным кодом;
2. Были разработаны алгоритмы модификации значений управляющих атрибутов I и II категорий. Эти алгоритмы при изменении значений управляющих атрибутов позволяют автоматически производить модификации в программе, что качественно повышает гибкость и надежность приложения.

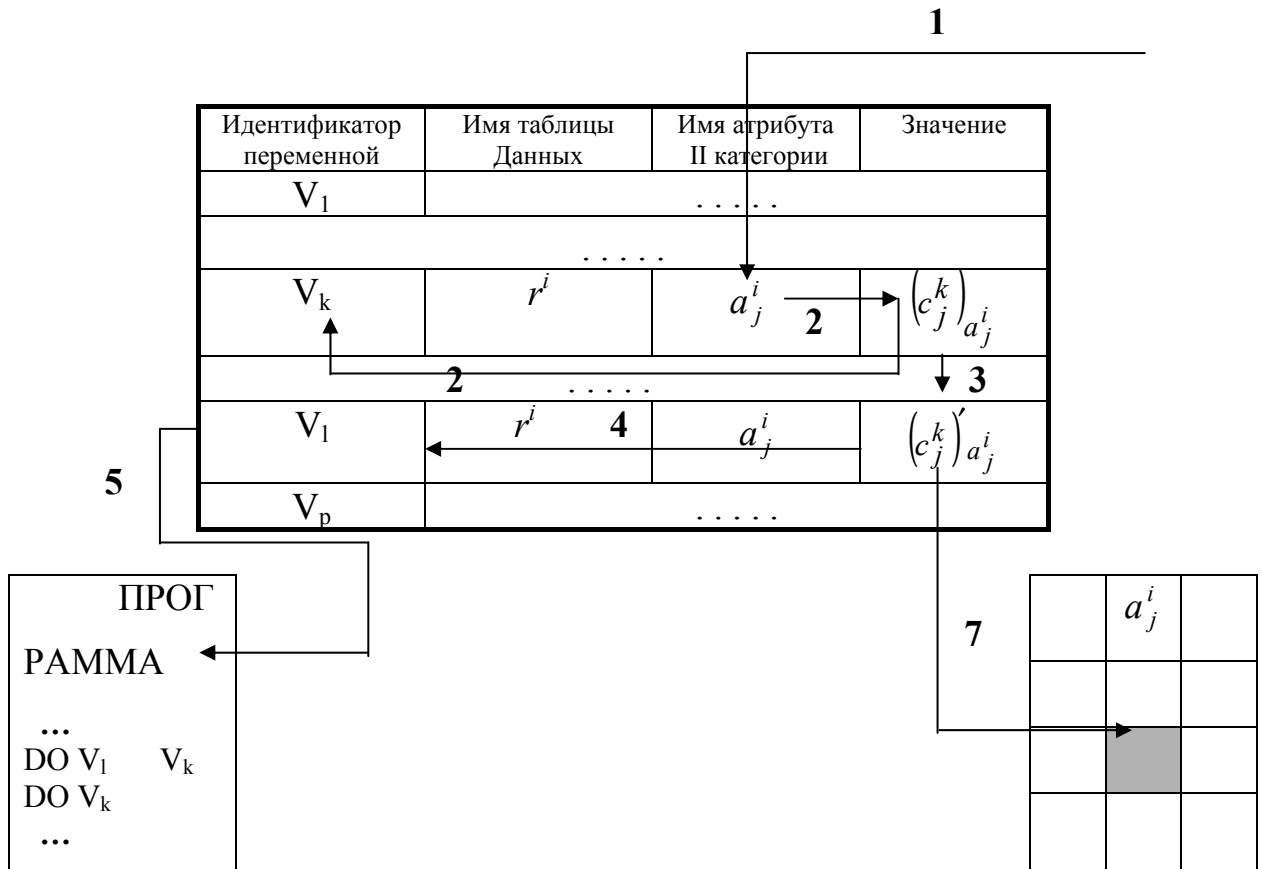


Рисунок 3.2 – Модификация значения атрибута П категории
и автоматические изменения в программе

ВЫВОДЫ

В данной дипломной работе было рассмотрено формальное расширение реляционной модели данных свойствами взаимодействия данных и программного кода, а именно было рассмотрено введение понятий управляющих атрибутов I и II категорий.

В ходе работы были описаны требования, а следовательно и ограничения, накладываемые на значения управляющих атрибутов, которые вытекают из правил общей целостности данных. Эти ограничения дают возможность корректно описывать объекты модели данных, при этом оставаясь в рамках реляционного подхода.

Следующей задачей данной работы было проведение анализа операций реляционной алгебры при введении понятий управляющих атрибутов. В результате было показано, что все операции классической реляционной алгебры применимы к отношениям с управляющими атрибутами, причем эти операции не меняют свои свойства. Также было показано, что введение нового свойства атрибута не противоречит базовым понятиям реляционной модели данных.

В следующей работе были также разработаны алгоритмы модификации значений управляющих атрибутов. Для их реализации были рассмотрены структуры управляющих таблиц, добавленные в систему словаря данных, которые описывают управляющие атрибуты и их связи с переменными приложения. Эти алгоритмы вместе с изменениями значений атрибутов, связанных с программным приложением, автоматически производят модификацию в программе таким образом, чтобы не нарушалась целостность приложения.

СПИСОК ССЫЛОК

1. Дейт К. Введение в системы баз данных. -М.: Наука,1980.-463 с.
2. Ульман Дж. Основы системы баз данных.-М.: Финансы и статистика,1983.-334 с.
3. Цикридис Д., Лоховский Ф. Модели данных. - М.: Финансы и статистка, 1985.-343с.
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. Codd E.F. Extending the Database Relational Model to Capture More Meaning. // ACM TODS,- Vol.4,-No.4,-1979.- pp.397-434.
6. Haskin R.L., Lorie R.A. On Extending the Functions of a Relational Database System. // Proc. ACM SIGMOD Conf., Orlando.- 1982.
7. Makinouchi I. A Consideration of Normal Forms of Not-Necessarily Normalized
8. Relations in the Relational Data Model. // Proc. 3rd International Conference on Very Large Data Bases, Tokyo.-1977
9. Smith J.M., Smith D.C.P. Database Abstractions: Aggregation and Generalization. // ACM TODS,- Vol.2,-No.2,-1977.- pp.105-133.
- 10.Ермолаев В.А. Система словаря данных при проектировании гибких информационных приложений. - Автореферат диссертации на соискание ученой степени кандидата физико-математических наук.- Запорожье, ЗГУ, 1994,- 20 с.
- 11.V.A.Ermolayev, N.G. Keberle. Active Data Dictionary and a Tool for Data Model Driven IS Design.-Submitted to ER-98 Intl. Conf.
- 12.Мейер Д. Теория реляционных баз данных.- М.: Мир,-1987.-608 с.
- 13.Codd E.F. Database Sublanguage Founded on the Relational Calculus.// ACM SIGFIDET Workshop on Data Description, Access and Control,- November, 1971,- pp.35-61.
- 14.Codd E.F. Relational Completeness of Data Base Sublanguages. Database Systems, Rustin R., Prentice-Hall, N.-J.,-1972.- pp.65-98.

15.Атре Ш. Структурный подход к организации баз данных.-М.: Финансы и статистика, -1983.-317 с.