

Владимирский государственный университет

БАЗЫ ДАННЫХ И ЭКСПЕРТНЫЕ СИСТЕМЫ

**Методические указания
к практическим работам**

Владимир 2004

Министерство образования Российской Федерации
Владимирский государственный университет
Кафедра Физики и Прикладной математики

БАЗЫ ДАННЫХ И ЭКСПЕРТНЫЕ СИСТЕМЫ

Методические указания
к практическим работам

Составитель:
В.Н. ГОРЛОВ

Владимир 2004

УДК 681.3(076)

Рецензент

Доктор технических наук, профессор кафедры «Управление и информатика в технических и экономических системах» Владимирского государственного университета

С.И. МАЛАФЕЕВ

Настоящие методические указания подготовлены по материалам лекционного курса, посвященного основам теории баз данных, языку запросов SQL и разработке экспертных систем, который автор читает студентам третьего курса по специальности «Прикладная математика и информатика».

В методических указаниях обсуждаются вопросы современной реляционной модели данных, которая является основой практически для всех коммерческих систем управления базами данных и наиболее распространена в настоящий момент. Рассмотрен современный стандарт языка работы с базами данных SQL.

1. РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

1.1. Основные определения

Реляционная модель данных некоторой предметной области представляет собой набор отношений, изменяющихся во времени. Отношение является основной структурой данных в модели.

N -арным отношением R называют подмножество декартова произведения $D_1 \times D_2 \times \dots \times D_n$ ($n \geq 1$), необязательно различных. Исходные множества D_1, D_2, \dots, D_n называют в модели доменами.

$$R \in D_1 \times D_2 \times \dots \times D_n,$$

где $D_1 \times D_2 \times \dots \times D_n$ – полное декартово произведение.

Таблица 1.1. Элементы реляционной модели

Элемент реляционной модели	Форма представления
Отношение	Таблица
Схема отношения	Строка заголовков столбцов таблицы (заголовок таблицы)
Кортеж	Строка таблицы
Сущность	Описание свойств объекта
Атрибут	Заголовок столбца таблицы
Домен	Множество допустимых значений атрибута
Значение атрибута	Значение поля записи
Первичный ключ	Один или несколько атрибутов
Тип данных	Тип значений элементов таблицы

Полное декартово произведение – это набор всевозможных сочетаний из n элементов, где каждый элемент берется из своего домена. Пусть, например, заданы три домена: D_1 содержит три фамилии, D_2 – набор из двух учебных дисциплин, D_3 представляет собой набор из трех оценок.

Допустим, содержание доменов следующее:

- ° D_1 – {Иванов, Крылов, Степанов};
- ° D_2 – {Теория автоматов, Базы данных};
- ° D_3 – {3,4,5}.

Для этих значений полное декартово произведение содержит набор из 18 троек, где первый элемент – одна из фамилий, второй – название одной из учебных дисциплин, третий – одна из оценок.

<Иванов,Теория автоматов,3>; <Иванов,Теория автоматов,4>; <Иванов,Теория автоматов,5>; <Крылов,Теория автоматов,3>; <Крылов,Теория автоматов,4>; <Крылов,Теория автоматов,5>; <Степанов,Теория автоматов,3>; <Степанов,Теория автоматов,4>; <Степанов,Теория автоматов,5>; <Иванов,Базы данных,3>; <Иванов,Базы данных,4>; <Иванов,Базы данных,5>; <Крылов,Базы данных,3>; <Крылов,Базы данных,4>; <Крылов,Базы данных,5>; <Степанов,Базы данных,3>; <Степанов,Базы данных,4>; <Степанов,Базы данных,5>.

Отношение R моделирует реальную ситуацию и оно может содержать, например, только 5 строк, которые соответствуют результатам сессии (Крылов экзамен по «Бадам данных» еще не сдавал):

$\langle \text{Иванов, Теория автоматов, 4} \rangle$; $\langle \text{Крылов, Теория автоматов, 5} \rangle$; $\langle \text{Степанов, Теория автоматов, 5} \rangle$; $\langle \text{Иванов, Базы данных, 3} \rangle$; $\langle \text{Степанов, Базы данных, 4} \rangle$.

Отношение имеет простую графическую интерпретацию и может быть представлено в виде таблицы, столбцы которой соответствуют входениям доменов в отношение, строки – наборам из n значений, взятых из исходных доменов, которые расположены в строго определенном порядке в соответствии с заголовком.

R		
Фамилия	Дисциплина	Оценка
Иванов	Теория автоматов	4
Иванов	Базы данных	3
Крылов	Теория автоматов	5
Степанов	Теория автоматов	5
Степанов	Базы данных	4

Данная таблица обладает рядом специфических свойств:

1. В таблице нет двух одинаковых строк.
2. Таблица имеет столбцы, соответствующие атрибутам отношения.
3. Каждый атрибут в отношении имеет уникальное имя.
4. Порядок строк в таблице произвольный.

Вхождение домена в отношение принято называть *атрибутом*. Строки отношения называются *кортежами*.

Количество атрибутов в отношении называется степенью, или рангом, отношения.

Следует заметить, что в отношении не может быть одинаковых кортежей, это следует из математической модели: отношение – это подмножество декартова произведения, а в декартовом произведении все элементы различны.

В соответствии со свойствами отношений два отношения, отличающиеся только порядком строк или порядком столбцов, будут интерпретироваться в рамках реляционной модели как одинаковые, то есть отношение R и отношение R_1 , представленное далее, одинаковы с точки зрения реляционной модели данных.

Любое отношение является динамической моделью некоторого реального объекта внешнего мира. Поэтому вводится понятие экземпляра отношения, которое отражает состояние данного объекта в текущий момент вре-

R1		
Дисциплина	Фамилия	Оценка
Теория автоматов	Иванов	4
Базы данных	Иванов	3
Теория автоматов	Крылов	5
Теория автоматов	Степанов	5
Базы данных	Степанов	4

мени, и понятие схемы отношения, которая определяет структуру отношения.

Схемой отношения R называется перечень имен атрибутов данного отношения с указанием домена, к которому они относятся:

$$S_R = (A_1, A_2, \dots, A_n), A_i \subseteq D_i.$$

Если атрибуты принимают значения из одного и того же домена, то они называются *θ -сравнимыми*, где θ - множество допустимых операций сравнения, заданных для данного домена. Например, если домен содержит числовые данные, то для него допустимы все операции сравнения, тогда $\theta = \{=, <, >, \geq, \leq, <=, >= \}$. Однако и для доменов, содержащих символьные данные, могут быть заданы не только операции сравнения по равенству и неравенству значений. Если для данного домена задано лексикографическое упорядочение, то он имеет также полный спектр операций сравнения.

Схемы двух отношений называются *эквивалентными*, если они имеют одинаковую степень и возможно такое упорядочение атрибутов в схемах, что на одинаковых местах будут находиться сравнимые атрибуты, то есть атрибуты, принимающие значения из одного домена.

$S_{R_1} = (A_1, A_2, \dots, A_n)$ – схема отношения R_1 .

$S_{R_2} = (A_{i1}, A_{i2}, \dots, A_{im})$ – схема отношения R_2 после упорядочения имен атрибутов.

Тогда

$$S_{R_1} \sim S_{R_2} \Leftrightarrow \begin{cases} 1, n = m \\ 2, A_j, B_{ij} \subseteq D_j \end{cases}$$

Реляционная модель представляет базу данных в виде множества взаимосвязанных отношений. В отличие от теоретико-графовых моделей, в реляционной модели связи между отношениями поддерживаются неявным образом. Связи между отношениями являются иерархическим. В каждой связи одно отношение может выступать как основное, а другое отношение выступает в роли подчиненного. Это означает, что один кортеж основного отношения может быть связан с несколькими кортежами подчиненного отношения. Для поддержки этих связей оба отношения должны содержать

наборы атрибутов, по которым они связаны. В основном отношении это первичный ключ отношения (PRIMARY KEY), который однозначно определяет кортеж основного отношения. В подчиненном отношении для моделирования связи должен присутствовать набор атрибутов, соответствующий первичному ключу основного отношения. Этот набор атрибутов является вторичным ключом, то есть он определяет множество кортежей подчиненного отношения, которые связаны с единственным кортежем основного отношения. Данный набор атрибутов в подчиненном отношении принято называть внешним ключом (FOREIGN KEY).

1.2. Операции над отношениями

Можно показать, что множество отношений замкнуто относительно некоторых специальных операций, то есть образует вместе с этими операциями абстрактную алгебру. Это важнейшее свойство отношений было использовано в реляционной модели для разработки языка манипулирования данными, связанного с исходной алгеброй. Американский математик Э.Ф. Кодд в 1970 году впервые сформулировал основные понятия и ограничения реляционной модели, ограничив набор операций в ней семью основными и одной дополнительной операцией. В общем это множество избыточное, так как одни операции могут быть представлены через другие, однако множество операций выбрано из соображений максимального удобства при реализации произвольных запросов к базе данных (БД).

Теоретико-множественные операции реляционной алгебры

Объединением двух отношений называется отношение, содержащее множество кортежей, принадлежащих либо первому, либо второму исходным отношениям, либо обоим отношениям одновременно.

Пусть заданы два отношения $R_1 = \{r_1\}$, $R_2 = \{r_2\}$, где r_1 и r_2 – соответственно кортежи отношений R_1 и R_2 , то объединение

$$R_1 \cup R_2 = \{r \mid r \in R_1 \vee r \in R_2\},$$

где r - кортеж нового отношения, \vee - операция логического сложения (логическое «ИЛИ»).

Пример применения операции объединения. Исходными отношениями являются R_1 и R_2 , которые содержат перечни деталей, изготавливаемых соответственно на первом и втором участках цеха. Отношение R_3 содержит общий перечень деталей, изготавливаемых в цеху, то есть характеризует общую номенклатуру цеха.

R1	
Шифр детали	Название детали
00011073	Гайка М1
00011075	Гайка М2
00011076	Гайка М3
00011003	Болт М1
00011006	Болт М3
00013063	Шайба М1
00013066	Шайба М3

R2	
Шифр детали	Название детали
00011073	Гайка М1
00011076	Гайка М3
00011077	Гайка М4
00011004	Болт М2
00011006	Болт М3

R3	
Шифр детали	Название детали
00011073	Гайка М1
00011075	Гайка М2
00011076	Гайка М3
00011003	Болт М1
00011006	Болт М3
00013063	Шайба М1
00013066	Шайба М3
00011077	Гайка М4
00011004	Болт М2

Пересечением отношений R_1 и R_2 называется отношение, которое содержит множество кортежей, принадлежащих одновременно и первому и второму отношениям

$$R_3 = R_1 \cap R_2 = \{r \mid r \in R_1 \wedge r \in R_2\},$$

где \wedge - операция логического умножения (логическое «И»).

В отношении R_4 содержится перечень деталей, которые выпускаются одновременно на двух участках цеха.

R4	
Шифр детали	Название детали
00011073	Гайка М1
00011076	Гайка М3
00011006	Болт М3

Разностью отношений R_1 и R_2 называется отношение, содержащее множество кортежей, принадлежащих R_1 и не принадлежащих R_2

$$R_5 = R_1 \setminus R_2 = \{r \mid r \in R_1 \wedge r \notin R_2\}$$

Отношение R_5 содержит перечень деталей, изготавливаемых только на участке 1, отношение R_6 содержит перечень деталей, изготавливаемых только на участке 2

$$R_6 = R_2 \setminus R_1 = \{r \mid r \in R_2 \wedge r \notin R_1\}$$

R5	
Шифр детали	Название детали
00011075	Гайка M2
00011003	Болт M1
00013063	Шайба M1
00013063	Шайба M3

R6	
Шифр детали	Название детали
00011077	Гайка M4
00011004	Болт M2

Следует отметить, что первые две операции, объединение и пересечение, являются коммутативными, то есть результат операции не зависит от порядка аргументов в операции. Операция разности является принципиально несимметричной операцией, то есть результат операции будет различным для разного порядка аргументов, что и показывает сравнение отношений R_5 и R_6 .

В отличие от навигационных средств манипулирования данными в теоретико-графовых моделях операции реляционной алгебры позволяют получить сразу иной качественный результат, который является семантически гораздо более ценным и понятным пользователям. Например, сравнение результатов объединения и разности номенклатуры двух участков позволит оценить специфику производств: насколько оно уникально на каждом участке, и, в зависимости от необходимости, принять соответствующее решение по изменению номенклатуры.

Для демонстрации возможностей рассмотренных операций реляционной алгебры приведем пример из другой предметной области. Исходными отношениями являются три отношения R_1 , R_2 и R_3 . Все отношения имеют эквивалентные схемы.

- ° $R_1 = (\text{ФИО}, \text{Паспорт}, \text{Школа});$
- ° $R_2 = (\text{ФИО}, \text{Паспорт}, \text{Школа});$
- ° $R_3 = (\text{ФИО}, \text{Паспорт}, \text{Школа}).$

Рассмотрим ситуацию поступления в высшие учебные заведения, которая была характерна для периода, когда были разрешены так называемые репетиционные вступительные экзамены, которые сдавались раньше основных вступительных экзаменов в вуз. Отношение R_1 содержит список абитуриентов, сдававших репетиционные экзамены. Отношение R_2 содержит список абитуриентов, сдававших экзамены на общих условиях, отношение R_3 – список абитуриентов, принятых в институт. Будем считать, что при неудачной сдаче репетиционных экзаменов абитуриент мог делать вторую попытку и сдавать экзамены в общем потоке, поэтому некоторые абитуриенты могут присутствовать как в первом, так и во втором отношении.

Рассмотрим следующие запросы:

1. Список абитуриентов, которые поступали два раза и не поступили в вуз

$$R = R_1 \cap R_2 \setminus R_3$$

2. Список абитуриентов, которые поступили в вуз с первого раза, то есть они сдавали экзамены только один раз и были сразу зачислены в вуз

$$R = (R_1 \setminus R_2 \cap R_3) \cup (R_2 \setminus R_1 \cap R_3)$$

3. Список абитуриентов, которые поступили в вуз только со второго раза. Прежде всего это абитуриенты, которые присутствуют в отношениях R_1 и R_2 , потому что они поступали два раза, и присутствуют в отношении R_3 , потому что они поступили

$$R = R_1 \cap R_2 \cap R_3$$

4. Список абитуриентов, которые поступали один раз и не поступили. Это прежде всего те абитуриенты, которые присутствуют в R_1 и не присутствуют в R_2 , и те, кто присутствуют в R_2 и не присутствуют в R_1 , и разумеется, никто из них не присутствует в R_3

$$R = (R_1 \setminus R_2) \cup (R_2 \setminus R_1) \setminus R_3$$

В отсутствие скобок порядок выполнения операций реляционной алгебры естественный, поэтому сначала будут выполнены операции в скобках, затем будет выполнена последняя операция вычитания отношения R_3 .

Операции объединения, пересечения и разности применимы только к отношениям с эквивалентными схемами.

Кроме перечисленных трех операций в рамках реляционной алгебры определена еще одна теоретико-множественная операция – расширенное декартово произведение. Эта операция не накладывает никаких дополнительных условий на схемы исходных отношений, поэтому операция расширенного декартова произведения, обозначаемая $R_1 \otimes R_2$, допустима для любых двух отношений. Прежде чем определить саму операцию, введем дополнительно понятие конкатенции, или сцепления кортежей.

Сцеплением, или конкатенацией, кортежей $c = \langle c_1, c_2, \dots, c_n \rangle$ и $q = \langle q_1, q_2, \dots, q_m \rangle$ называется кортеж, полученный добавлением значений второго в конец первого. Сцепление кортежей c и q обозначается как (c, q)

$$(c, q) = \langle c_1, c_2, \dots, c_n, q_1, q_2, \dots, q_m \rangle.$$

Расширенным декартовым произведением отношения R_1 степени n со схемой

$$S_{R1} = (A_1, A_2, \dots, A_n)$$

и отношения R_2 со схемой

$$S_{R2} = (B_1, B_2, \dots, B_m)$$

называется отношение R_3 степени $n+m$ со схемой

$$S_{R3} = (A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m),$$

содержащее кортежи, полученные сцеплением каждого кортежа r отношения R_1 с каждым кортежем отношения R_2 .

Если $R_1 = \{r\}$, $R_2 = \{q\}$, то

$$R_1 \otimes R_2 = \{(r, q) \mid r \in R_1 \wedge q \in R_2\}.$$

Операция декартова произведения с учетом возможности перестановки атрибутов в отношении можно считать симметричной. Очень часто операция расширенного декартова произведения используется для получения некоторого универсума – то есть отношения, которое характеризует все возможные комбинации между элементами отдельных множеств. Однако самостоятельного значения результат выполнения операции обычно не имеет, он участвует в дальнейшей обработке. Например, на производстве в отношении R_7 задана обязательная номенклатура деталей для всех цехов, а в отношении R_8 содержится перечень всех цехов.

R7	
Шифр детали	Название детали
00011073	Гайка М1
00011075	Гайка М2
00011076	Гайка М3
00011003	Болт М1
00011006	Болт М3
00013063	Шайба М1
00013066	Шайба М3
00011077	Гайка М4
00011004	Болт М2
00011005	Болт М5
00011006	Болт М6
00013062	Шайба М2

R8
Цех
Цех 1
Цех 2
Цех 3

Тогда отношение R_9 , которое соответствует ситуации, когда каждый цех изготавливает все требуемые детали, будет выглядеть следующим образом:

R9		
Шифр детали	Название детали	Цех
00011073	Гайка М1	Цех 1
00011075	Гайка М2	Цех 1
00011076	Гайка М3	Цех 1
00011003	Болт М1	Цех 1
00011006	Болт М3	Цех 1
00013063	Шайба М1	Цех 1
00013066	Шайба М3	Цех 1
00011077	Гайка М4	Цех 1
00011004	Болт М2	Цех 1
00011005	Болт М5	Цех 1
00011006	Болт М6	Цех 1
00013062	Шайба М2	Цех 1
00011073	Гайка М1	Цех 2
00011075	Гайка М2	Цех 2
00011076	Гайка М3	Цех 2
00011003	Болт М1	Цех 2
00011006	Болт М3	Цех 2
00013063	Шайба М1	Цех 2
00013066	Шайба М3	Цех 2
00011077	Гайка М4	Цех 2
00011004	Болт М2	Цех 2
00011005	Болт М5	Цех 2
00011006	Болт М6	Цех 2
00013062	Шайба М2	Цех 2
00011073	Гайка М1	Цех 3
00011075	Гайка М2	Цех 3
00011076	Гайка М3	Цех 3
00011003	Болт М1	Цех 3
00011006	Болт М3	Цех 3
00013063	Шайба М1	Цех 3
00013066	Шайба М3	Цех 3
00011077	Гайка М4	Цех 3
00011004	Болт М2	Цех 3
00011005	Болт М5	Цех 3
00011006	Болт М6	Цех 3
00013062	Шайба М2	Цех 3

R10		
Шифр детали	Название детали	Цех
00011073	Гайка М1	Цех 1
00011075	Гайка М2	Цех 1
00011076	Гайка М3	Цех 1
00011003	Болт М1	Цех 1
00011006	Болт М3	Цех 1
00013063	Шайба М1	Цех 1
00013066	Шайба М3	Цех 1
00011077	Гайка М4	Цех 1
00011004	Болт М2	Цех 1
00011006	Болт М3	Цех 2

R10 (продолжение)		
00013063	Шайба М1	Цех 2
00013066	Шайба М3	Цех 2
00011077	Гайка М4	Цех 2
00011004	Болт М2	Цех 2
00011006	Болт М6	Цех 2
00013062	Шайба М2	Цех 2
00011073	Гайка М1	Цех 3
00011075	Гайка М2	Цех 3
00011076	Гайка М3	Цех 3
00011003	Болт М1	Цех 3
00011006	Болт М3	Цех 3
00013063	Шайба М1	Цех 3
00013066	Шайба М3	Цех 3
00011077	Гайка М4	Цех 3
00011005	Болт М5	Цех 3
00011006	Болт М6	Цех 3
00013062	Шайба М2	Цех 3

Расширенное декартово произведение моделирует некоторую ситуацию, которая характеризуется словом «все». Поэтому если нужно узнать, какие детали в каких цехах из общей обязательной номенклатуры не выпускаются, то можно вычесть из полученного отноршения R_9 отношение R_{10} , характеризующее реальный выпуск деталей в каждом цехе.

Отношение R_{11} , которое является результاتم выполнения этой операции, имеет вид

$$R_{11} = R_9 \setminus R_{10}$$

R11		
Шифр детали	Название детали	Цех
00011073	Гайка М1	Цех 2
00011075	Гайка М2	Цех 2
00011076	Гайка М3	Цех 2
00011004	Болт М2	Цех 3
00013062	Шайба М2	Цех 3
00011003	Болт М1	Цех 2
00011005	Болт М5	Цех 3

Специальные операции реляционной алгебры

Первой специальной операцией реляционной алгебры является *горизонтальный выбор*, или *операция фильтрации*, или *операция ограничения отношений*. Для определения этой операции введем дополнительные обозначения.

Пусть α - логическое выражение, составленное из термов сравнения с помощью связок И,ИЛИ,НЕ и, возможно, скобок. В качестве термов сравнения допускаются

а) терм A ос a ,

где A - имя некоторого атрибута, принимающего значения из домена D ; a – константа, принадлежащая домену D ; ос – одна из допустимых для данного домена операций сравнения;

б) терм A ос B ,

где A, B – имена некоторых θ -сравнимых атрибутов, то есть атрибутов, принимающих значения из одного и того же домена D .

Тогда результатом операции выбора, или фильтрации, заданной на отношении R в виде логического выражения, определенного на атрибутах отношения R , называется отношение $R[\alpha]$, включающее те кортежи из исходного отношения, для которых истинно условие выбора или фильтрации:

$$R[\alpha(r)] = \{ r \in R \wedge \alpha(r) = \text{"Истина"} \}$$

Операция фильтрации является одной из основных при работе с реляционной моделью данных. Условие α может быть сколь угодно сложным.

Например, выбрать из R_{10} детали с шифром «0011003»

$$R_{12} = R_{10} [\text{Шифр детали} = \text{"0011003"}]$$

R12		
Шифр детали	Название детали	Цех
00011003	Болт М1	Цех 1
00011003	Болт М1	Цех 3

Следующей специальной операцией является *операция проектирования*. Пусть R – отношение, $S_R = (A_1, A_2, \dots, A_n)$ – схема отношения R . Обозначим через V подмножество атрибутов из $\{A_i\}$. При этом пусть V^1 – множество из $\{A_i\}$, не вошедших в V .

Если $V = \{A_i^1, A_i^2, \dots, A_i^k\}$, $V^1 = \{A_j^1, A_j^2, \dots, A_j^k\}$ и $r = \langle a_i^1, a_i^2, \dots, a_i^k \rangle, a_i^k \in A_i^k$, то $r[V]$, $s = \langle a_j^1, a_j^2, \dots, a_j^m \rangle, a_j^m \in A_j^m$.

Проекцией отношения R на набор атрибутов V , обозначаемой $R[V]$, называется отношение со схемой, соответствующей набору атрибутов V $S_{R[V]} = V$, содержащем кортежи, получаемые из кортежей исходного отношения R путем удаления из них значений, не принадлежащих атрибутам из набора V .

$$R[V] = \{r[V]\}.$$

По определению отношений все дублирующие кортежи удаляются из результирующего отношения.

Операция проектирования, называемая иногда также операцией вертикального выбора, позволяет получить только требуемые характеристики моделируемого объекта. Чаще всего операция проектирования употребляется как промежуточный шаг в операциях горизонтального выбора, или фильтрации. Кроме того, они используются самостоятельно на заключительном этапе получения ответа на запрос.

Например, выберем все цеха, которые изготавливают деталь «Болт М1». Для этого необходимо из отношения R_{10} выбрать детали с заданным названием, а потом полученное отношение спроектировать на столбец «Цех». Результатом выполнения этих операций будет отношение R_{14}

$$R_{13} = R_{10} [\text{Название детали} = \text{“Болт М1”}]$$

$$R_{14} = R_{13} [\text{Цех}]$$

R13		
Шифр детали	Название детали	Цех
00011003	Болт М1	Цех 1
00011003	Болт М1	Цех 3

R14
Цех
Цех 1
Цех 3

Следующей специальной операцией реляционной алгебры является операция *условного соединения*.

В отличие от рассмотренных выше специальных операций реляционной алгебры: фильтрации и проектирования, которые являются унарными, то есть применяются к одному отношению, операция условного соединения является бинарной, то есть исходными для нее являются два отношения.

Пусть $R = \{r\}$, $Q = \{q\}$ – исходные отношения, S_R , S_q – схемы отношений R и Q соответственно

$$S_R = (A_1, A_2, \dots, A_k); S_q = (B_1, B_2, \dots, B_m),$$

где A_i, B_j – имена атрибутов в схемах отношений R и Q соответственно.

При этом полагаем, что заданы наборы атрибутов A и B

$$A \subseteq \{A_i\}, i = 1, k; B \subseteq \{B_j\}, j = 1, m$$

и эти наборы состоят из θ -сравнимых атрибутов.

Тогда соединением отношений R и Q при условии β будет подмножество декартова произведения отношений R и Q , кортежи которого удовлетворяют условию β , рассматриваемому как одновременное выполнение условий:

° $r.A_i \theta_i B_j : i=1, k$, где k – число атрибутов, входящих в наборы A и B ; θ_i – операция сравнения

° $A_i \theta_i B_i D$; θ_i - i -й предикат сравнения, определяемый из множества допустимых на домене D операций сравнения

$$R[\beta] Q = \{(r,q) \mid (r,q) \mid r.A \theta_i q.B = \text{'Истина'}, j=1,k\}$$

Например, рассмотрим следующий запрос. Пусть отношений R_{15} содержит перечень деталей с указанием материалов, из которых эти детали изготавливаются

R15		
Шифр детали	Название детали	Материал
00011073	Гайка М1	сталь-ст1
00011075	Гайка М2	сталь-ст2
00011076	Гайка М3	сталь-ст1
00011003	Болт М1	сталь-ст3
00011006	Болт М3	сталь-ст3
00013063	Шайба М1	сталь-ст1
00013066	Шайба М3	сталь-ст1
00011077	Гайка М4	сталь-ст2
00011004	Болт М2	сталь-ст3
00011005	Болт М5	сталь-ст3
00013062	Шайба М2	сталь-ст1

R16
Название детали
Гайка М1
Гайка М3
Шайба М1
Шайба М3
Шайба М2

Получим перечень деталей, которые изготавливаются в цехе 1 из материала «сталь-ст1»

$$R_{15} = (R_{15}[(R_{15}.\text{Шифр детали} = R_{10}.\text{Шифр детали}) \wedge R_{10}.\text{Цех} = \text{«Цех1»}] \wedge R_{15}.\text{Материал} = \text{«сталь-ст1»}) R_{10} [\text{Название детали}]$$

Последней операцией, включаемой в набор операций реляционной алгебры, является операция *деления*.

Для определения операции деления рассмотрим сначала понятие множества образов.

Пусть R – отношение со схемой $S_R = (A_1, A_2, \dots, A_k)$, A – некоторый набор атрибутов $A \subset \{A_i\} i=1, k$, A^1 – набор атрибутов, не входящих в множество A .

Пересечение множеств A и A^1 пусто: $A \cap A^1 = \emptyset$; объединение множеств равно множеству всех атрибутов исходного отношения $A \cup A^1 = S_R$.

Тогда множеством образов элемента y проекции $R[A]$ называется множество таких элементов y проекции $R[A^1]$, для которых сцепление (x,y) является кортежами отношения R , то есть

$$QA(x) = \{y \mid y \in R[A^1] \wedge (x,y) \in R\} - \text{множество образов.}$$

Например, множеством образов отношения R_{15} по материалу «сталь-ст2» будет множество кортежей

$$R_{15}.\text{Материал} = \{ \langle 00011075, \text{Гайка М2}, \text{«сталь-ст2»} \rangle, \langle 00011077, \text{Гайка М4}, \text{«сталь-ст2»} \rangle \}$$

Используя введенное понятие множества образов, можно дать следующее определение операции деления.

Пусть даны два отношения R и T соответственно со схемами

$$S_R = (A_1, A_2, \dots, A_k); S_T = (B_1, B_2, \dots, B_m),$$

где A и B – наборы атрибутов этих отношений одинаковой длины (без повторений);

$$A \subset S_R; B \subset S_T$$

Атрибуты A^1 – это атрибуты из R , не вошедшие в множество A .

Пересечение множеств $A \cap A^1 = \emptyset$ и $A \cup A^1 = S^R$. Проекции $R[A]$ и $T[B]$ совместимы по объединению, то есть имеют эквивалентные схемы. Тогда операция деления ставит в соответствие отношениям R и T отношение $Q = R[A:B]T$, кортежи которого являются теми элементами проекции $R[A^1]$, для которых $T[B]$ входит в построенное для них множество образов:

$$R[A:B]T = \{r \mid r \in R[A^1] \wedge T[B] \subseteq \{y \mid y \in R[A] \wedge (r,y) \in R\}\}.$$

Операция деления удобна тогда, когда требуется сравнить некоторое множество характеристик отдельных атрибутов. Например, пусть дано отношение R_7 , которое содержит номенклатуру всех выпускаемых на предприятии деталей. В отношении R_{10} хранятся сведения о том, что и в каких цехах действительно выпускается. Поставим задачу определить перечень цехов, в которых выпускается вся номенклатура деталей.

Тогда решением этой задачи будет операция деления отношения R_{10} на отношение R_7 по набору атрибутов (Шифр детали, Наименование детали).

$$R_{17} = R_{10}[\text{Шифр детали, Наименование детали} : \text{Шифр детали, Наименование детали}] R_7$$

R17
Цех
Цех 1

2. СТРУКТУРИРОВАННЫЙ ЯЗЫК ЗАПРОСОВ SQL

2.1. Структура SQL

В отличие от реляционной алгебры, где были представлены только операции запросов к БД, SQL является полным языком, в нем присутствуют не только операции запросов, но и операторы, соответствующие DDL – Data Definition Language – языку описания данных. Кроме того, язык содержит операторы, предназначенные для управления (администрирования) БД. SQL содержит разделы, представленные в таблице 1.1

Таблица 2.1 Операторы определения данных DDL

Оператор	Смысл	Действие
CREATE TABLE	Создать таблицу	Создает новую таблицу в БД
DROP TABLE	Удалить таблицу	Удаляет таблицу из БД
ALTER TABLE	Изменить таблицу	Изменяет структуру существующей таблицы или ограничения целостности, задаваемое для таблицы
CREATE VIEW	Создать представление	Создает виртуальную таблицу, соответствующую некоторому SQL-запросу
ALTER VIEW	Изменить представление	Изменяет ранее созданное Представление
DROP VIEW	Удалить представление	Удаляет ранее созданное представление
CREATE INDEX	Создать индекс	Создает индекс для некоторой таблицы с целью обеспечения быстрого доступа по атрибута, входящим в индекс
DROP INDEX	Удалить индекс	Удаляет ранее созданный индекс

Таблица 2.2 Операторы манипулирования данными Data Manipulation Language (DML)

Оператор	Смысл	Действие
DELETE	Удалить строки	Удаляет одну или несколько строк, соответствующих условиям фильтрации, из базовой таблицы
INSERT	Вставить строку	Вставляет одну строку в Базовую таблицу
UPDATE	Обновить строку	Обновляет значения одного или нескольких столбцов в одной или нескольких строках

Таблица 2.3 Язык запросов Data Query Language (DQL)

Оператор	Смысл	Действие
SELECT	Выбрать строки	Оператор, заменяющий все операторы реляционной алгебры и позволяющий сформировать результирующее отношение, соответствующее запросу

Таблица 2.4 Средства управления транзакциями

Оператор	Смысл	Действие
COMMIT	Завершить транзакцию	Завершить комплексную взаимосвязанную обработку информации объединенную в транзакцию
ROLLBACK	Откатить транзакцию	Отменить изменения, проведенные в ходе выполнения транзакции
SAVEPOINT	Сохранить промежуточную точку выполнения транзакции	Сохранить промежуточное состояние БД, пометить его для того, чтобы можно было в дальнейшем к нему вернуться

Таблица 2.5 Средства администрирования данных

Оператор	Смысл	Действие
ALTER DATABASE	Завершить транзакцию	Изменить набор основных объектов в базе данных, ограничений, касающихся всей базы данных
ALTER DBAREA	Откатить транзакцию	Изменить ранее созданную Область хранения
ALTER PASSWORD	Изменить пароль	Изменить пароль для всей базы данных
CREATE DATABASE	Создать БД	Создать новую базу данных, определив основные параметры для нее
CREATE DBAREA	Создать новую область хранения	Создать новую область хранения и сделать ее доступной для размещения данных
DROP DATABASE	Удалить БД	Удалить существующую базу данных
DROP DBAREA	Удалить область хранения БД	Удалить существующую область хранения (если в ней в данный момент не располагаются активные данные)
GRANT	Предоставить права	Предоставить права доступа на ряд действий над некоторым объектом БД

Продолжение таблицы 2.5

REVOKE	Лишить прав	Лишить прав доступа к некоторому объекту или некоторым действиям над объектом
--------	-------------	---

Таблица 2.6 Программный SQL

Оператор	Смысл	Действие
DECLARE	Определить курсор для запроса	Задает некоторое имя и определяет связанный с ним запрос к БД, который соответствует виртуальному набору данных
OPEN	Открыть курсор	Формирует виртуальный набор данных, соответствующий описанию указанного курсора и текущему состоянию БД
FETCH	Считать строку из множества строк, определенных курсором	Считывает строку, заданную переметром команды из виртуального набора данных, соответствующего открытому курсору
CLOSE	Закрыть курсор	Прекращает доступ к виртуальному набору данных, соответствующему указанному курсору
PREPARE	Подготовить оператор SQL к динамическому выполнению	Генерирует план выполнения запроса, соответствующего заданному оператору
EXECUTE	Выполнить оператор SQL, ранее подготовленный к динамическому выполнению	Выполняет ранее подготовленный план запроса

В коммерческих СУБД набор основных оператор расширен. В большинстве СУБД включены операторы определения и запуска хранимых процедур и операторы определения триггеров.

2.2. Операторы языка SQL

Оператор выбора SELECT

Язык запросов (Data Query Language) в SQL состоит из единственного оператора SELECT. Этот оператор поиска реализует все операции реляционной алгебры. Синтаксис оператора SELECT имеет следующий вид:

```
SELECT [ALL | DISTINCT] (<Список полей> | *)
FROM <Список таблиц>
[WHERE <Предикат-условие выборки или соединения>]
[GROUP BY <Список полей результата>]
```

[HAVING <Предикат-условие для группы>]
[ORDER BY <Список полей, по которым упорядочить вывод>]

Ключевое слово ALL означает, что в результирующий набор строк включаются все строки, удовлетворяющие условиям запроса. Значит, в результирующий набор могут попасть одинаковые строки, что является нарушением принципов теории отношений (в отличие от реляционной алгебры, где по умолчанию предполагается отсутствие дубликатов в каждом результирующем отношении). Ключевое слово DISTINCT означает, что в результирующий набор включаются только различные строки, то есть дубликаты строк результата не включаются в набор.

Символ * (звездочка) означает, что в результирующий набор включаются все столбцы из исходных таблиц запроса.

В разделе FROM задается перечень исходных отношений (таблиц) запроса. В разделе WHERE задаются условия отбора строк результата или условия соединения кортежей исходных таблиц, подобно операции условного соединения в реляционной алгебре.

В разделе GROUP BY задается список полей группировки.

В разделе HAVING задаются предикаты-условия, накладываемые на каждую группу. В части ORDER BY задается список полей упорядочения результата, то есть список полей, который определяет порядок сортировки в результирующем отношении. Например, если первым полем списка будет указана Фамилия, а вторым Номер группы, то в результирующем отношении сначала будут собраны в алфавитном порядке студенты, и если найдутся однофамильцы, то они будут расположены в порядке возрастания номеров групп.

Рассмотрим более подробно первые три строки оператора SELECT:

° SELECT – ключевое слово, которое сообщает СУБД, что эта команда – запрос. Все запросы начинаются этим словом с последующим пробелом. За ним может следовать способ выборки - с удалением дубликатов (DISTINCT) или без удаления (ALL, подразумевается по умолчанию). Затем следует столбцов, которые выбираются запросом из таблиц, или символ '*' (звездочка) для выбора всей строки. Любые столбцы, не перечисленные здесь, не будут включены в результирующее отношение, соответствующее выполнению команды. Это не значит, что они будут удалены или их информация будет стерта из таблиц, потому что запрос не воздействует на информацию в таблицах – он только показывает данные.

° FROM – ключевое слово, подобно SELECT, которое должно быть представлено в каждом запросе. Оно сопровождается пробелом и затем именами таблиц, используемых в качестве источника информации. В случае если указано более одного имени таблицы, неявно подразумевается, что над перечисленными таблицами осуществляется операция декартова произведения. Таблицам можно присвоить имена-псевдонимы, что бывает полез-

но для осуществления операции соединения таблицы с самой собою или доступа из вложенного подзапроса к текущей записи внешнего запроса.

Все последующие разделы оператора SELECT являются необязательными.

Самый простой запрос SELECT без необязательных частей соответствует просто декартову произведению. Например, выражение

```
SELECT *  
FROM R1,R2
```

соответствует декартову произведению таблиц R1 и R2.

Выражение

```
SELECT R1.A, R2.B  
FROM R1,R2
```

соответствует проекции декартова произведения двух таблиц на два столбца: .A из таблицы R1 и B из таблицы R2, при этом дубликаты всех строк сохранены, в отличие от операции проектирования в реляционной алгебре, где при проектировании по умолчанию все дубликаты кортежей уничтожаются.

° WHERE – ключевое слово, за которым следует предикат – условие, налагаемое на запись в таблице, которому она должна удовлетворять, чтобы попасть в выборку, аналогично операции селекции в реляционной алгебре.

Рассмотри базу данных, которая моделирует сдачу сдачи сессии в некотором учебном заведении. Пусть она состоит из трех отношений R₁, R₂ и R₃. Будем считать, что они представлены таблицами R1, R2 и R3 соответственно.

R1 = (ФИО,Дисциплина,Оценка); R2 = (ФИО,Группа);

R3 = (Группа,Дисциплина);

R1		
ФИО	Дисциплина	Оценка
Петров Ф.И.	Базы данных	5
Сидоров К.А.	Базы данных	4
Морозов А.В.	Базы данных	2
Степанова К.Е.	Базы данных	2
Крылова Т.С.	Базы данных	5
Сидоров К.А.	Теория информации	4
Степанова К.Е.	Теория информации	2
Крылова Т.С.	Теория информации	5
Миронов А.В.	Теория информации	Null
Владимиров В.А.	Базы данных	5
Трофимов П.А.	Сети и телекоммуникации	4
Иванова Е.А.	Сети и телекоммуникации	5
Уткина Н.В.	Сети и телекоммуникации	5
Владимиров В.А.	Английский язык	4
Трофимов П.А.	Английский язык	5

R1 продолжение		
Иванова Е.А.	Английский язык	3
Петров Ф.И.	Английский язык	5

R2	
ФИО	Группа
Петров Ф.И.	4906
Сидоров К.А.	4906
Морозов А.В.	4906
Крылова Т.С.	4906
Владимиров В.А.	4906
Трофимов П.А.	4807
Иванова Е.А.	4807
Уткина Н.В.	4807

R3	
Группа	Дисциплина
4906	Базы данных
4906	Теория информации
4906	Английский язык
4807	Английский язык
4807	Сети и телекоммуникации

Приведем несколько примеров использования оператора SELECT.

° Вывести список групп (без повторений), где должны пройти экзамены

```
SELECT DISTINCT Группы  
FROM R3
```

Результат:

Группа
4906
4807

° Вывести список студентов, которые сдали экзамен по дисциплине «Базы данных» на «отлично»

```
SELECT ФИО  
FROM R1  
WHERE Дисциплина = "Базы данных" AND Оценка = 5
```

Результат:

ФИО
Петров Ф.И.
Крылова Т.С.

° Вывести список всех студентов, которым надо сдавать экзамены с указанием названий дисциплин, по которым должны проводиться эти экзамены

```
SELECT ФИО,Дисциплина  
FROM R2,R3  
WHERE R2.Группа = R3.Группа
```

Здесь часть WHERE задает условия соединения отношений R2 и R3 , при отсутствии условий соединения в части WHERE результат будет эквивалентен расширенному декартову произведению, и в этом случае каждому студенту были бы приписаны все дисциплины из отношения R3, а не те, которые должна сдавать его группа.

В заключение заметим, что логика работы оператора выбора (декартово произведение-селекция-проекция) не совпадает с порядком описания в нем данных (сначала список полей для проекции, потом список таблиц для декартова произведения и условие соединения). Дело в том, что SQL изначально разрабатывался для применения конечными пользователями, и его стремились сделать возможно ближе к естественному языку, а не алгоритмическому языку.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Дейт К. Дж. Введение в системы баз данных.: Пер. с англ. 6-е изд. М.: Диалектика, 1998
2. Мартин Дж. Организация баз данных в вычислительных системах. Пер с англ. М.: Мир, 1980
3. Ульман Дж. Основы систем баз данных.- М.: Финансы и статистика, 1983
4. Мамаев Е.В. Администрирование MS SQL Server 7.0. СПб.: БХВ – Санкт-Петербург. 2000

ОГЛАВЛЕНИЕ

1. Реляционная модель данных.....	3
2. Основные определения.....	3
3. Операции над отношениями.....	6
4. Структурированный язык запросов SQL.....	16
5. Структура SQL.....	16
6. Операторы языка SQL.....	18