

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

КАФЕДРА «ЭВМ и СИСТЕМЫ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по дисциплине

«БАЗЫ И БАНКИ ДАННЫХ»

Часть 4. ОПЕРАЦИИ РЕЛЯЦИОННОЙ АЛГЕБРЫ

для студентов специальности
«Автоматизированные системы обработки информации»

Брест 2001

В данных методических указаниях приведены общие сведения о реляционной алгебре и рассмотрены традиционные и дополнительные операции реляционной алгебры. Дано определение и алгоритмы реализации этих операций и рассмотрено применение операций на примере конкретных отношений базы данных. Приведены методика и примеры построения запросов к базам данных на основе операций реляционной алгебры, а также предложены варианты заданий для построения запросов к базе данных.

Методические указания предназначены для использования студентами специальности «Автоматизированные системы обработки информации» Т.10.01.00 в ходе выполнения лабораторных и самостоятельных работ по дисциплине «Базы и банки данных».

Составители: В.Г.Брич, доцент, к.т.н.

В.И.Хвещук, доцент, к.т.н.

Рецензенты: Л.П.Матюшков, профессор, к.т.н., директор Брестского филиала Института Современных Знаний

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. ОБЩИЕ СВЕДЕНИЯ О РЕЛЯЦИОННОЙ АЛГЕБРЕ	5
2. ТРАДИЦИОННЫЕ ОПЕРАЦИИ РЕЛЯЦИОННОЙ АЛГЕБРЫ	8
2.1. ОПЕРАЦИЯ ОБЪЕДИНЕНИЯ	9
2.2. ОПЕРАЦИЯ ПЕРЕСЕЧЕНИЯ	10
2.3. ОПЕРАЦИЯ ВЫЧИТАНИЯ	11
2.4. ОПЕРАЦИЯ ПРОИЗВЕДЕНИЕ	12
3. СПЕЦИАЛЬНЫЕ ОПЕРАЦИИ РЕЛЯЦИОННОЙ АЛГЕБРЫ	14
3.1. ОПЕРАЦИЯ ВЫБОРКИ	14
3.2. ОПЕРАЦИЯ ПРОЕКЦИЯ	15
3.3. ЕСТЕСТВЕННОЕ СОЕДИНЕНИЕ	16
3.4. ОПЕРАЦИЯ СОЕДИНЕНИЯ	18
3.5. ОПЕРАЦИЯ ДЕЛЕНИЯ	19
4. ИСПОЛЬЗОВАНИЕ ОПЕРАЦИЙ РЕЛЯЦИОННОЙ АЛГЕБРЫ	21
4.1. ПРИМЕРЫ ФОРМУЛИРОВАНИЯ ЗАПРОСОВ	21
4.2. УПРАЖНЕНИЯ ДЛЯ СОСТАВЛЕНИЯ ЗАПРОСОВ	26
СПИСОК ЛИТЕРАТУРЫ	27
СПИСОК СОКРАЩЕНИЙ	27

ВВЕДЕНИЕ

Данная работа является логическим продолжением методических указаний «Проектирование реляционных баз данных» [1] и представляет собой введение в реляционную алгебру. В данных методических указаниях приведены общие сведения о реляционной алгебре (традиционные и дополнительные операции реляционной алгебры, их синтаксис и примеры применения операций, методика построения запросов к базам данных).

Основой современной технологии баз данных (БД) является реляционная модель данных (РМД). РМД включает три основных аспекта: *структуру данных* (объекты данных и связей), *целостность данных* и *обработку данных* (операторы или операции). В работе [1] рассмотрен первый аспект технологии реляционных БД, а в данных методических указаниях представлены основы реляционной алгебры, т.е. приведены определения и примеры использования реляционных операций.

Обработка данных в рамках реляционной модели данных реализуется с помощью операций, которые выполняются над отношениями. Реляционные операции делятся на две группы:

- *операции над множествами (объединение, пересечение, разность, деление и декартово произведение);*
- *специальные операции над отношениями (проекция, соединение, выбор и другие).*

В различных системах управления базами данных (СУБД) реализована некоторая часть операций над отношениями, определяющими в какой-то мере возможности данной СУБД и сложность реализации запросов к БД. В реляционных СУБД для выполнения операций над отношениями используется две группы языков, имеющих в качестве своей математической основы теоретические языки запросов, предложенные Э.Коддом: *реляционная алгебра* и *реляционное исчисление*. Именно реляционные языки оказались серьезным основанием для реляционной революции в БД в 1970-1971 годах благодаря работам Э.Кодда [2]. Кодд определил понятия реляционной модели данных и реляционных языков обработки данных — реляционную алгебру и реляционное исчисление. Наиболее важным аспектом этой модели были ее конкретные языки обработки данных, которые позволили манипулировать данными на основе только их логических характеристик.

Основным компонентом РМД является реляционная алгебра, которая состоит из набора операторов, использующих отношения в качестве операндов и возвращающих отношения в качестве результата.

Реляционная алгебра использует процедурный язык обработки реляционных таблиц. Это означает использование пошагового подхода к созданию реляционных таблиц, содержащих ответы на запросы к БД.

Реляционное исчисление использует непроцедурный язык. В реляционном исчислении запрос создается путем определения таблицы запроса за один шаг. Непроцедурный язык позволяет формулировать, что нужно сделать, а не как этого добиться.

Реляционная алгебра и реляционное исчисление логически эквивалентны. Это означает, что любой запрос к БД, который можно сформулировать при помощи логического исчисления, также можно сформулировать, пользуясь реляционной алгеброй, и наоборот. Если язык имеет как минимум те же возможности, что и реляционная алгебра, то он называется *реляционно полным*.

1. ОБЩИЕ СВЕДЕНИЯ О РЕЛЯЦИОННОЙ АЛГЕБРЕ

Реляционная алгебра как теоретический язык запросов по сравнению с реляционным исчислением более наглядно описывает выполняемые над отношениями действия. *Запрос* представляет собой специальным образом описанное требование, определяющее состав производимых над БД операций по выборке, удалению или модификации хранимых данных.

Вариант реляционной алгебры, предложенной Коддом, включает следующие группы операций:

1 *Операции математической теории множеств: объединение, пересечение, вычитание и декартово произведение (все они модифицированы с учетом того, что их операндами являются отношения, а не произвольные множества).*

2 *Специальные реляционные операции: выборка, проекция, соединение и деление.*

Операции реляционной алгебры манипулируют реляционными таблицами. Это значит, что операции используют одну или две из существующих таблиц для создания новой таблицы. Затем полученная новая таблица может использоваться в качестве входной таблицы для следующей операции. Операции реляционной алгебры упрощают создание запросов, поскольку можно экспериментировать с частичными решениями, пока не будет найден работающий способ решения.

Каждое отношение имеет две части - заголовок и тело, где заголовок - это множество атрибутов, а тело - это данные. При создании отношения становится известен его заголовок, а при заполнении и тело отношения. При выполнении операций реляционной алгебры над существующими отношениями всегда должны получать в результате отношения, к именам атрибутов которых можно обращаться в последующих операциях. Результатом произвольной реляционной операции является отношение, которое, в свою очередь, может участвовать в другой реляционной операции. Это свойство реляционной алгебры называется *свойством замкнутости*. Такое свойство предполагает возможность вложенности операций реляционной алгебры.

По справедливому замечанию Дейта, реляционная алгебра Кодда обладает несколькими недостатками [2].

Во-первых, восемь перечисленных операций по охвату своих функций избыточны, так как минимально необходимый набор составляет пять операций: объединение, вычитание, произведение, проекция и выборка. Три другие операции (пересечение, соединение и деление) можно определить через пять минимально необходимых.

Во-вторых, этих восьми операций недостаточно для построения реальной СУБД на принципах реляционной алгебры. Требуются расширения, включающие следующие операции: переименования атрибутов, образования новых вычисляемых атрибутов, вычисления итоговых функций, построения сложных алгебраических выражений, присвоения, сравнения и другие.

Дополнительные операции реляционной алгебры, предложенные Дейтом, включают следующие операции [2].

Операция **ПЕРИМЕНОВАНИЯ** позволяет изменить имя атрибута отношения и имеет вид:

RENAME <исходное отношение> <старое имя атрибута>

AS <новое имя атрибута>,

где <исходное отношение> задается именем отношения либо выражением

реляционной алгебры.

Операция **РАСШИРЕНИЯ** порождает новое отношение, похожее на исходное, но отличающееся наличием добавленного атрибута, значения которого получаются путем некоторых скалярных вычислений. Операция расширения имеет вид:

EXTEND <исходное отношение> *ADD* <выражение> *AS* <новый атрибут>,

где к исходному отношению добавляется (ключевое слово *ADD*) <новый атрибут> подсчитываемый по правилам, заданным <выражением>. Исходное отношение может быть задано именем отношения или с помощью выражения реляционной алгебры, заключенного в круглые скобки.

Помимо обычных арифметических операций и операций сравнения, в выражении можно использовать различные функции, называемые итоговыми, такие как:

COUNT (количество), *SUM* (сумма), *AVG* (среднее),

MAX (максимальное), *MIN* (минимальное).

Операция **ПОДВЕДЕНИЯ ИТОГОВ** выполняет «вертикальные» или групповые вычисления и имеет следующий формат:

SUMMARIZE <исходное отношение> *BY* <список атрибутов>

ADD <выражение> *AS* <новый атрибут>,

где исходное отношение задается именем отношения либо заключенным в круглые скобки выражением реляционной алгебры, <список атрибутов> представляет собой разделенные запятыми имена атрибутов исходного отношения X_1, X_2, \dots, X_N <выражение> — скалярное выражение, аналогичное выражению операции *EXTEND*, а <новый атрибут> — имя формируемого атрибута. В списке атрибутов и в выражении не должен использоваться <новый атрибут>. Результатом операции *SUMMARIZE* является отношение *R* с заголовком, состоящим из атрибутов списка, расширенного новым атрибутом. Для получения тела отношения *R* сначала выполняется проецирование (назовем проекцию *R1*) исходного отношения на атрибуты X_1, X_2, \dots, X_N , после чего каждый кортеж проекции расширяется новым ($N+1$)-м атрибутом. Поскольку проецирование, как правило, приводит к сокращению количества кортежей по отношению к исходному отношению (удаляются одинаковые кортежи), то можно считать, что происходит своеобразное группирование кортежей исходного отношения: одному кортежу отношения *R1* соответствует один или более (если было дублирование при проецировании) кортежей исходного отношения. Значение ($N+1$)-го атрибута каждого кортежа отношения *R* формируется путем вычисления выражения над соответствующей этому кортежу группой кортежей исходного отношения.

Операцию **ПРИСВОЕНИЯ** можно представить следующим образом:

<выражение-цель> := <выражение-источник>,

где оба выражения задают совместимые (точнее, эквивалентные) по структуре отношения. Типичный случай выражений: в левой части — имя отношения, а в правой — некоторое выражение реляционной алгебры. Выполнение операции присвоения сводится к замене предыдущего значения отношения на новое (начальное значение, если тело отношения было пустым), определенное выражением-источником. С помощью операции присвоения можно не только полностью заменить все значения отношения-цели, но и добавить или удалить кортежи. Более удобными операциями изменения тела отношения являются операции вставки, обновления и удаления.

Операция **ВСТАВКИ** имеет следующий вид:

INSERT <выражение-источник> *INTO* <выражение-цель>.

где оба выражения должны быть совместимы по структуре. Выполнение операции сводится к вычислению <выражение-источник> и вставке полученных кортежей в отношение, заданное <выражение-цель>.

Операция **ОБНОВЛЕНИЯ** имеет следующий вид:

UPDATE <выражение-цель> <список элементов>,

где <список элементов> представляет собой последовательность разделенных запятыми операций присвоения <атрибут>:= <скалярное выражение>. Результатом выполнения операции обновления является отношение, полученное после присвоения соответствующих значений атрибутам отношения, заданного целевым выражением.

Операция **УДАЛЕНИЯ** имеет следующий вид:

DELETE <выражение-цель>,

где <выражение-цель> представляет собой реляционное выражение, описывающее удаляемые кортежи.

Операция **РЕЛЯЦИОННОГО СРАВНЕНИЯ** может использоваться для прямого сравнения двух отношений. Она имеет синтаксис:

<выражение1> θ <выражение2>,

где оба выражения задают совместимые по структуре отношения, а знак θ — один из следующих операторов сравнения: равно, не равно и другие.

При записи произвольного выражения реляционной алгебры надо принимать во внимание следующее.

1. В реляционной алгебре должен быть определен приоритет выполнения операций (например, операция пересечение более приоритетна, чем операция объединения), который нужно учитывать при записи выражений. Для изменения порядка выполнения операций в выражениях можно использовать круглые скобки.
2. Существуют тождественные преобразования, позволяющие по-разному записывать одно и то же выражение.
3. Составляя выражение, нужно обеспечивать совместимость участвующих в операциях отношений. При необходимости изменения заголовков следует выполнять переименование атрибутов.

Рассмотрим логические аспекты операций реляционной алгебры на примере БД, состоящей из отношений **ПРЕПОДАВАТЕЛЬ**, **ПРЕДМЕТ**, **СТУДЕНЧЕСКАЯ ГРУППА** и связующего отношения **ПРЕПОДАВАТЕЛЬ ПРЕПОДАЕТ ПРЕДМЕТЫ В ГРУППАХ**. Конкретные значения записей перечисленных отношений БД приведены в таблицах 1.1.-1.4.

ПРЕПОДАВАТЕЛЬ

ТАБЛИЦА 1.1.

ПолноеИмя	Номер	Фамилия	Должность	Кафедра	Специальность	ТелефонДомаШный
101Л		Фролова	Доцент	ЭВМ	АСОИ	487
102Л		Костина	Доцент	ЭВМ	ЭВМ	483
103Л		Байко	Профессор	АСУ	АСОИ	112
104Л		Глазев	Ассистент	ТФ	СД	421
105Л		Петров	Ассистент	Экономия	Международная экономика	324

ПРЕДМЕТ

ТАБЛИЦА 1.2.

КодовыйНомерПредмета	НазваниеПредмета	КоличествоЧасов	Специальность	Семестр
12П	Матр ЭВМ	36	ЭВМ	1
14П	ПЭВМ	72	ЭВМ	2
17П	СУБД ПК	48	АСОИ	4
18П	ВКСС	36	АСОИ	6
34П	Финика	10	СД	6
22П	Аудит	24	Бухучет	3

СТУДЕНЧЕСКАЯ ГРУППА

ТАБЛИЦА 1.3.

Кодовый Номер Группы	Название Группы	Количество Человек	Специальность	Фамилия Староста
ЯГ	Э-12	18	ЭВМ	Иванова
ЗГ	Э-15	22	ЭВМ	Степан
4Г	АС-9	24	АСОИ	Балабанов
3Г	АС-8	20	АСОИ	Чижик
17Г	С-14	29	СД	Амурская
12Г	М-6	16	Менеджеры	Трунин
10Г	Б-4	21	Буучет	Зыткин

ТАБЛИЦА 1.4.

ПРЕПОДАВАТЕЛЬ ПРЕПОДАЕТ ПРЕДМЕТЫ В ГРУППАХ

Кодовый Номер Группы	Кодовый Номер Предмета	Личный Номер	Номер Аудитории
ЯГ	12П	222Л	112
ЯГ	14П	221Л	220
ЯГ	17П	222Л	112
ЗГ	14П	221Л	220
ЗГ	17П	222Л	241
ЗГ	18П	225Л	210
4Г	12П	222Л	112
4Г	18П	225Л	210
3Г	12П	222Л	112
3Г	17П	221Л	241
3Г	18П	225Л	210
17Г	12П	222Л	112
17Г	22П	110Л	220
17Г	34П	470Л	118
12Г	17П	222Л	112
12Г	22П	110Л	210
10Г	12П	222Л	210
10Г	22П	110Л	210

2. ТРАДИЦИОННЫЕ ОПЕРАЦИИ РЕЛЯЦИОННОЙ АЛГЕБРЫ

Для описания операций используем конкретные операторы языка SQL (Structured Query Language) - структурированный язык запросов [2].

К традиционным операциям над множествами относятся операции объединения, пересечения, вычитания и декартового произведения. Рассмотрим подробнее механизм выполнения реляционных операций на примере операции объединения.

В математике объединением двух множеств является множество, состоящее из всех элементов объединяемых множеств. Если отношение рассматривать как множество, то объединение двух отношений содержит множество всех принадлежащих им записей. Результатом объединения отношений **ПРЕПОДАВАТЕЛЬ** и **ПРЕДМЕТЫ** является множество записей, которое не является отношением, так как содержит записи разного типа. В соответствии с определением свойства замкнутости результатом объединения отношений должно быть отношение. Таким образом, объединение отношений в реляционной алгебре не полностью совпадает с понятием объединения множеств. Два исходных объединяемых отношения должны иметь одинаковую структуру. Это означает, что отношение **ПРЕПОДАВАТЕЛЬ** должно объединяться с таким же по структуре отношением **ПРЕПОДАВАТЕЛЬ**.

Два отношения совместимы по типу (структуре) если каждое имеет одинаковое число атрибутов (одну и ту же степень) и атрибуты также совместимы по типу. Совместимость по типу атрибутов означает, что они определены на одном и том же множестве допустимых значений. Кроме операции объединения,

совместимости по типу требуют операции пересечения и вычитания. Для выполнения перечисленных операций при различиях в именах атрибутов можно воспользоваться оператором **RENAME** для обеспечения совместимости имен атрибутов.

2.1. ОПЕРАЦИЯ ОБЪЕДИНЕНИЯ

ОБЪЕДИНЕНИЕМ (UNION) двух совместимых по типу отношений X и Y ($X \cup Y$) называется отношение R , которое содержит k записей из отношений X и Y , и имеет такой же заголовок, что и в отношениях X и Y .

Если отношение X содержит m записей, а отношение Y - n записей, то выражение $k = m + n$, где k - число записей отношения R , выполняется только в том случае, если отношения X и Y не содержат одинаковых записей. Если отношения X и Y содержат g одинаковых записей, то отношение R содержит $k = m + n - g$ записей. Графическая схема операции вычитания представлена на рис.2.4.

Используя оператор **UNION** языка SQL объединение отношений X и Y можно представить в виде: $X \cup Y$. Операция объединения ассоциативна, т.е. если X , Y и Z - произвольные совместимые по типу отношения, то выражения

$$(X \cup Y) \cup Z \text{ и } X \cup (Y \cup Z)$$

эквивалентны. Представленные формы записи выражений можно упростить:

$$X \cup Y \cup Z.$$

Операция объединения коммутативна, т.е. приведенные ниже выражения эквивалентны:

$$X \cup Y \text{ и } Y \cup X.$$

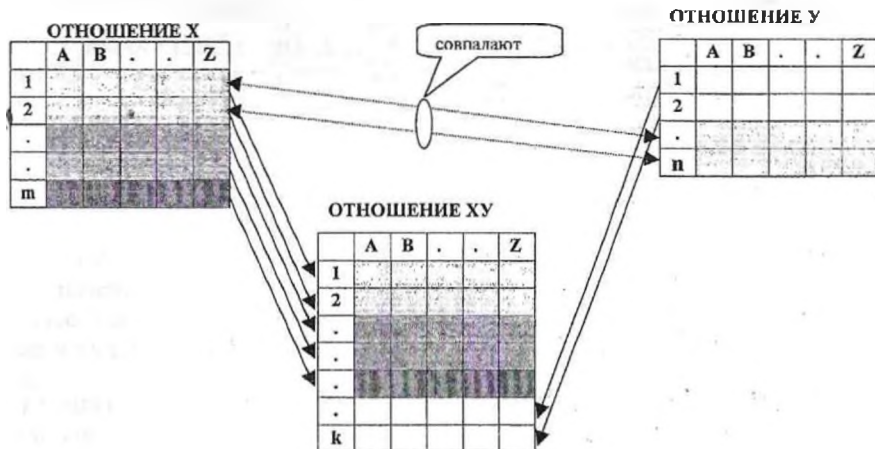


Рис.2.1. Схема операции объединения

Например, пусть имеется два отношения **X ПРЕПОДАВАТЕЛЬ** (табл.3.1.) и **Y ПРЕПОДАВАТЕЛЬ** (табл.3.2.), структура которых аналогична структуре отношения представленного в таблице 1.1. Первое отношение представляет преподавателей кафедр ЭВМ и АСУ, а второе - преподавателей ведущих свои занятия в аудитории с номером 241. Объединение этих отношений можно представить в виде: $X \cup Y$.

Результатом операции объединения является отношение **R ПРЕПОДАВАТЕЛЬ** (табл.2.3), которое содержит сведения о преподавателях, которые или работают на кафедре ЭВМ и АСУ или ведут занятия в 241 аудитории. Общее количество записей в отношении **R ПРЕПОДАВАТЕЛЬ** равно $k=3+3-1=5$. Количество записей в отношениях **X ПРЕПОДАВАТЕЛЬ** и **Y ПРЕПОДАВАТЕЛЬ** соответственно $m=3$ и $n=3$, а количество повторяющихся записей $g=1$.

X ПРЕПОДАВАТЕЛЬ

ТАБЛИЦА 2.1.

Личный Номер	Фамилия	Должность	Кафедра	Телефон Домашний
211Л	Фролов	Доцент	ЭВМ	487
145Л	Брунов	Доцент	ЭВМ	543
225Л	Бойко	Профессор	АСУ	112

Y ПРЕПОДАВАТЕЛЬ

ТАБЛИЦА 2.2.

Личный Номер	Фамилия	Должность	Кафедра	Телефон Домашний
211Л	Фролов	Доцент	ЭВМ	487
145Л	Брунов	Профессор	АСУ	312
338Л	Петров	Ассистент	ТФ	427

R ПРЕПОДАВАТЕЛЬ

ТАБЛИЦА 2.3.

Личный Номер	Фамилия	Должность	Кафедра	Телефон Домашний
211Л	Фролов	Доцент	ЭВМ	487
145Л	Брунов	Профессор	АСУ	312
222Л	Костин	Доцент	ЭВМ	543
225Л	Бойко	Профессор	АСУ	112
338Л	Петров	Ассистент	ТФ	427

2.2. ОПЕРАЦИЯ ПЕРЕСЕЧЕНИЯ

ПЕРЕСЕЧЕНИЕМ (INTERSECT) двух совместимых по типу отношений **X** и **Y** (**X INTERSECT Y**) является отношение **R** с таким же заголовком, как и в отношениях **X** и **Y**, и содержащем все те записи, которые одновременно принадлежат обоим отношениям **X** и **Y**. Графическая схема процесса выполнения операции пересечения представлена на рис.2.2.

Для обозначения операции пересечения используется оператор **INTERSECT**. Пересечение, как и операция объединения, обладает свойством коммутативности и ассоциативности для совместимых по типу отношений **X**, **Y**, **Z**. Выражения

(X INTERSECT Y) INTERSECT Z и

X INTERSECT (Y INTERSECT Z) эквивалентны.

Например, пусть имеется два отношения **X ПРЕПОДАВАТЕЛЬ** и **Y ПРЕПОДАВАТЕЛЬ**, (см. табл.2.1, 2.2). Результатом пересечения этих отношений является отношение **R ПРЕПОДАВАТЕЛЬ** с одной записью, которая содержит сведения о преподавателе с кафедры ЭВМ и ведущем занятию в 241 аудитории приведено в табл.1.4.

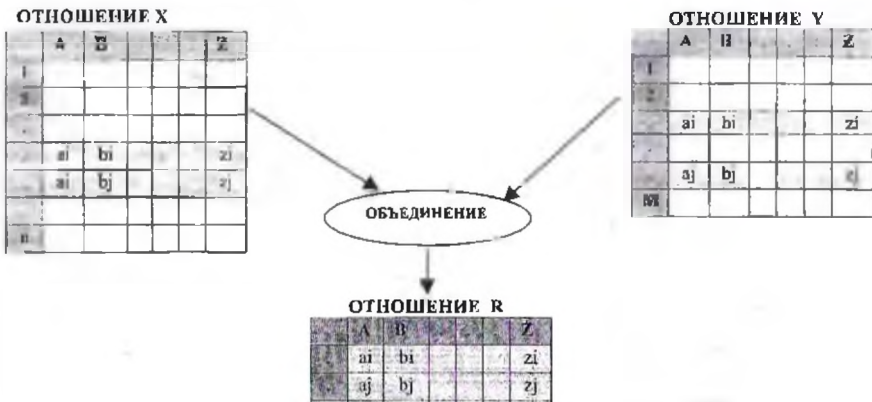


Рис.2.2. Схема выполнения операции объединения

R ПРЕПОДАВАТЕЛЬ (X ПРЕПОДАВАТЕЛЬ INTERSET Y ПРЕПОДАВАТЕЛЬ) ТАБЛИЦА 2.4.

Личный Номер	Фамилия	Должность	Кафедра	Телефон Домашний
221Л	Фролов	Доцент	ЭВМ	487

2.3. ОПЕРАЦИЯ ВЫЧИТАНИЯ

ВЫЧИТАНИЕМ (MINUS) двух совместимых по типу отношений X и Y (X MINUS Y) называется отношение R с тем же заголовком, что и отношениях X и Y, и R содержит записи, которые принадлежат отношению X и не принадлежат отношению Y. Пример операции вычитания представлен на рис.2.3.

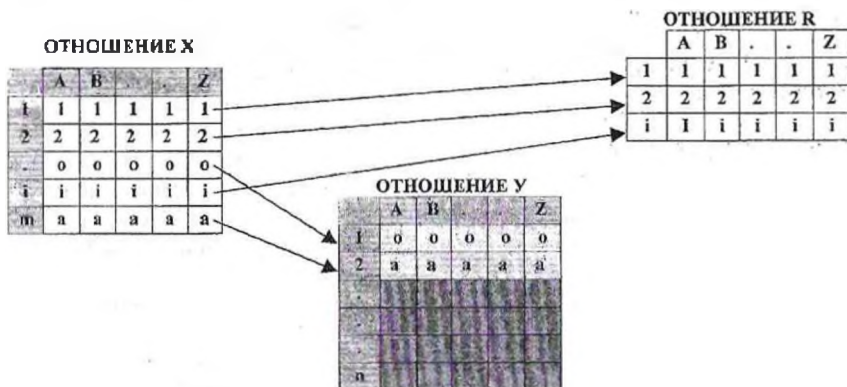


Рис.2.3. Схема выполнения операции вычитания

Оператором вычитания на языке SQL является MINUS. В качестве исходных отношений для операции вычитания используем два известных отношения

X ПРЕПОДАВАТЕЛЬ и **Y ПРЕПОДАВАТЕЛЬ**. Операцию вычитания можно представить в виде следующего выражения:

X ПРЕПОДАВАТЕЛЬ MINUS Y ПРЕПОДАВАТЕЛЬ

Результатом операции вычитания является отношение **R ПРЕПОДАВАТЕЛЬ** (табл.2.5.), в котором содержатся сведения о преподавателях с кафедр ЭВМ и АСУ, которые не проводят занятия в 241 аудитории.

Если в операции вычитания поменять местами операнды операции

Y ПРЕПОДАВАТЕЛЬ MINUS X ПРЕПОДАВАТЕЛЬ,

то получим результирующее отношение **R ПРЕПОДАВАТЕЛЬ 1** (табл.2.6), содержащее сведения о преподавателях ведущих занятия в аудитории 241, но не работающих на кафедрах ЭВМ и АСУ. Таким образом, при вычитании учитывается порядок расположения отношений в выражении.

R ПРЕПОДАВАТЕЛЬ ТАБЛИЦА 2.5
(**X ПРЕПОДАВАТЕЛЬ MINUS Y ПРЕПОДАВАТЕЛЬ**)

Личный Номер	Фамилия	Должность	Кафедра	Телефон Домашний
223Л	Костин	Доцент	ЭВМ	341
225Л	Бойко	Профессор	АСУ	112

R ПРЕПОДАВАТЕЛЬ 1 ТАБЛИЦА 2.6.
(**Y ПРЕПОДАВАТЕЛЬ MINUS X ПРЕПОДАВАТЕЛЬ**)

Личный Номер	Фамилия	Должность	Кафедра	Телефон Домашний
223Л	Бойко	Профессор	АСУ	112
338Л	Петров	Ассистент	1Ф	427

2.4. ОПЕРАЦИЯ ПРОИЗВЕДЕНИЕ

ДЕКАРТОВО ПРОИЗВЕДЕНИЕ (TIMES) двух множеств является множеством всех таких упорядоченных пар элементов, что первый элемент в каждой паре принадлежит первому множеству, а второй элемент в каждой паре второму множеству. Таким образом, декартово произведение двух отношений должно быть множеством упорядоченных пар записей. Но, согласно свойства замкнутости, результат должен содержать записи, а не упорядоченные пары записей. Поэтому, для реляционной алгебры декартово произведение представляет собой операцию, в которой каждая упорядоченная пара записей заменяется одной записью, образованной в результате объединения (в смысле теории множеств) (далее сцепления) двух записей этой пары. Графическая схема операции произведения приведена на рис.2.4.

Результирующее отношение декартового произведения должно иметь правильно сформированный заголовок, который представляет собой просто сцепление двух заголовков исходных отношений. Если два исходных отношения имеют заголовки с общими именами атрибутов, то результирующий заголовок будет иметь два одинаковых атрибута, а значит, будет "неверно сформированным". Чтобы избежать данной проблемы, необходимо прежде для переименования соответствующих атрибутов применить оператор RENAME.

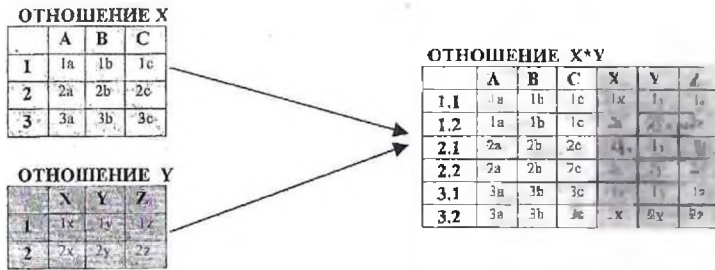


Рис.2.4. Схема выполнения операции произведения

Результатом **декартова произведения** двух отношений X и Y ($X \text{ TIMES } Y$), не имеющих общих имен: атрибутов, является отношение R с заголовком, который представляет собой сцепление двух заголовков исходных отношений, при этом отношение R состоит из множества всех записей $k = m * n$, где m - множество записей отношения X, а n - множество записей отношения Y. Общее количество атрибутов в отношении R равно сумме атрибутов отношений X и Y. В качестве оператора произведения в языке SQL используется оператор **TIMES**.

Операция декартова произведения не очень важна на практике, потому что в результате ее выполнения нет никакой дополнительной информации по сравнению с исходной. Декартово произведение используется при выполнении реляционной операции **тета-соединения**, о которой изложено ниже.

Следует отметить, что в теории множеств операция декартова произведения не является ни ассоциативной, ни коммутативной, но предложенный вариант операции для реляционной алгебры обладает свойством ассоциативности и коммутативности.

Например, пусть имеются отношения **ПРЕПОДАВАТЕЛЬ** и **ПРЕДМЕТ** (рис.2.5), где первое отношение содержит сведения о **ЛичныхНомерах** преподавателей, а второе - о **КодовыхНомерахПредметов**, которые они ведут. Результатом операции **ПРЕПОДАВАТЕЛЬ TIMES ПРЕДМЕТ** является отношение **ПРЕПОДАВАТЕЛЬ ПРЕДМЕТ**, содержащее все пары преподаватель - предмет и предмет - преподаватель.



Рис.2.5. Пример выполнения операции тета-соединения

3. СПЕЦИАЛЬНЫЕ ОПЕРАЦИИ РЕЛЯЦИОННОЙ АЛГЕБРЫ

3.1. ОПЕРАЦИЯ ВЫБОРКИ

Операция **ВЫБОРКИ** (**WHERE**) выполняется над одним отношением и возвращает отношение, содержащее все записи из исходного отношения, которые удовлетворяют определенным условиям. Пример выполнения операции приведен на рис.3.1. В качестве условий выборки могут использоваться скалярные операторы сравнения (=, ≤, <, ≥ и т.п.) или логические операции (**AND**, **OR**, **NOT** и др.).

Если **F** – логическое выражение или оператор сравнения для отношения **R**, то выборкой называется отношение **R1**, имеющее структуру отношения **R** и записи, удовлетворяющие истинности выражения **F**. В выражении **F** могут присутствовать в качестве операндов имена атрибутов (или номера столбцов), константы. С помощью оператора **WHERE** языка **SQL** можно записать операцию выборки следующим образом:

R WHERE F

Выборка из отношения **R** по атрибутам **x1** и **x2** при условии **θ** запишется в следующем виде: **R WHERE x1 θ x2**.

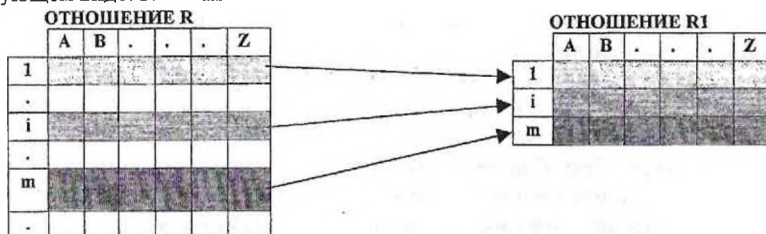


Рис.3.1. Схема выполнения операции выборка

При этом условие $x1 \theta x2$ должно иметь смысл, т.е. атрибуты $x1$ и $x2$ принадлежат одному типу данных, а условие θ имеет смысл для данного типа данных. Например, для атрибута **ВидСпорта** имеет смысл операция равно и не равно, но не имеют смысла операция меньше или больше.

Благодаря свойству замкнутости операция выборки может содержать произвольное число логических сочетаний простых сравнений. Таким образом, можно предложить следующие тождества, используя в них уже известные операции реляционной алгебры:

1. **R WHERE x1 AND x2** = **(R WHERE x1) INTERSECT (R WHERE x2)**
2. **R WHERE x1 OR x2** = **(R WHERE x1) UNION (R WHERE x2)**
3. **R WHERE NOT x1** = **R MINUS (R WHERE x1)**

Рассмотрим примеры операций выборки для выше рассмотренных отношений **ПРЕПОДАВАТЕЛЬ** и **ПРЕДМЕТ**. Пусть необходимо определить всех преподавателей, работающих на кафедре ЭВМ в отношении **ПРЕПОДАВАТЕЛЬ**. Данную операцию можно записать в виде:

ПРЕПОДАВАТЕЛЬ WHERE Кафедра = «ЭВМ»,

Результатом операции будет отношение **ПРЕПОДАВАТЕЛЬ 1** (табл. 3.1).

Личный Номер	Фамилия	Должность	Кафедра	Специальность	Телефон Домашний
221Л	Федотов	Доцент	ЭВМ	АСОИ	487
222Л	Костин	Доцент	ЭВМ	ЭВМ	543

В отношении ПРЕДМЕТ определим все сведения о предметах, изучаемых с 4 по 6 семестр, т.е. **ПРЕДМЕТ WHERE Семестр > 3 AND Семестр < 6.**

Результатом выполнения такой операции выборки является отношение ПРЕДМЕТ1.

ПРЕДМЕТ1

ТАБЛИЦА 3.2.

Кодовый Номер Предмета	Название Предмета	Количество Часов	Специальность	Семестр
17П	СУБД ПК	48	АСОИ	4
18П	ВКСС	32	АСОИ	6
34П	Физика	30	СД	6

3.2. ОПЕРАЦИЯ ПРОЕКЦИЯ

ПРОЕКЦИЕЙ отношения R по атрибутам X_1, X_2, \dots, X_n , которые принадлежат отношению R, называется отношение R1 с заголовком $\{X_1, X_2, \dots, X_n\}$, содержащим все записи отношения R после исключения из него атрибутов, которые не входят в список атрибутов X_1, X_2, \dots, X_n . Данное определение можно записать в виде: $R1 = R [X_1, X_2, \dots, X_n]$. Схема операции проекция изображена на рис.3.2.

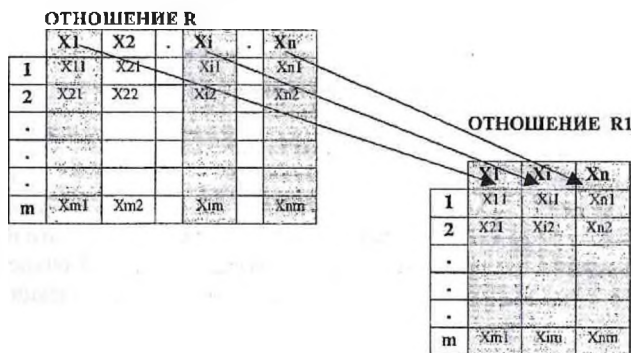


Рис.3.2. Схема выполнения операции проекция

С помощью оператора проекции получено "вертикальное" подмножество данного отношения, т.е. подмножество, получаемое исключением всех атрибутов, не указанных в списке атрибутов, и последующим исключением дублирующих записей из того, что остается. При этом каждый атрибут указывается в списке атрибутов не более одного раза.

Проекция, в которой список атрибутов не пропущен, т.е. такая $R []$, при которой список пустой, представляет собой "нулевую" проекцию. Если в списке указываются все атрибуты отношения R, то такая проекция эквивалентна проекции

без указания списка атрибутов и представляет собой тождественную проекцию. Таким образом, имя отношения является, допустимым реляционным выражением.

Рассмотрим несколько примеров проекций (см.рис.3.3) для отношений **ПРЕПОДАВАТЕЛЬ**, **СТУДЕНЧЕСКАЯ_ГРУППА** и **ПРЕДМЕТ**. В первом примере приведена проекция отношения **ПРЕПОДАВАТЕЛЬ** по атрибуту **Должность** – результатом выборки является отношение **ПРЕПОДАВАТЕЛЬ_1**, содержащее один атрибут **Должность** и три записи, поскольку исключены дублирующие записи. Во втором примере представлена проекция отношения **СТУДЕНЧЕСКАЯ_ГРУППА** по атрибутам **Специальность** и **КоличествоЧеловек** – результатом является отношение **СТУДЕНЧЕСКАЯ_ГРУППА_1** с количеством записей равным количеству исходного отношения. В третьем примере для отношения **ПРЕДМЕТ** использованы операции выборки и проекции совместно в виде следующей формулы:

(**ПРЕДМЕТ WHERE Семестр = «б»**) [**НазваниеПредмета**].

Результатом является отношение **ПРЕДМЕТ_1**, содержащее сведения о предметах, изучаемых в 6 семестре.

ПРЕПОДАВАТЕЛЬ_1		СТУДЕНЧЕСКАЯ_ГРУППА_1	
Должность		Количество Человек	Специальность
Доцент		18	ЭВМ
Профессор		22	ЭВМ
Ассистент		24	АСОИ
		20	АСОИ
		29	СД
		16	Международная экономика
		21	Бухучет

ПРЕДМЕТ_1	
Название Предмета	Семестр
ВКСС	б
Физика	б

Рис.3.3. Примеры операций проекции

На практике бывает удобно указывать не те атрибуты, по которым происходит проекция, а те, которые исключаются. Например, удобнее воспользоваться проекцией, исключающей атрибут **Специальность** из отношения **ПРЕПОДАВАТЕЛЬ**, чем проекцией отношения **ПРЕПОДАВАТЕЛЬ** по атрибутам [**ЛичныйНомер**, **Фамилия**, **Должность**, **Кафедра**, **ТелефонДомашний**]. Конкретного синтаксиса для этой альтернативной, но, эквивалентной версии проекции, не предлагается.

3.3. ЕСТЕСТВЕННОЕ СОЕДИНЕНИЕ

ЕСТЕСТВЕННОЕ СОЕДИНЕНИЕ (JOIN). Операция соединения имеет несколько разновидностей, однако, наиболее важным является естественное соединение. Пример реализации операции соединения приведен на рис.3.4.

Пусть имеется отношение **X** с заголовком $\{x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m\}$ и отношение **Y** с заголовком $\{y_1, y_2, \dots, y_m, z_1, z_2, \dots, z_l\}$, где атрибуты y_1, y_2, \dots, y_m общие для **X** и **Y** и совпадают по типу.

Тогда естественным соединением отношений **X** и **Y** (**X JOIN Y**) называется отношение **R** с заголовком $\{x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m, z_1, z_2, \dots, z_l\}$ и содержащим множество всех записей таких, для которых значение

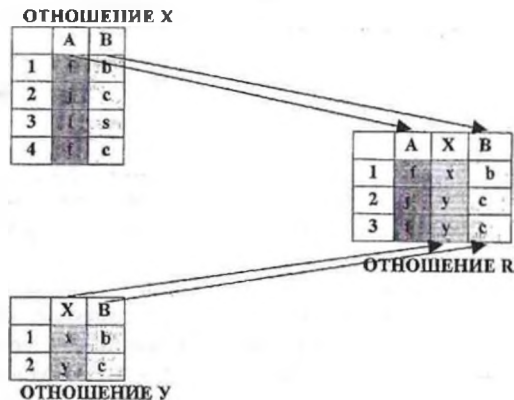


Рис.3.4. Пример выполнения операции естественное объединение

атрибута x_k ($k=1, \dots, n$) совпадает со значением x_k в отношении X, значение y_d ($d=1, \dots, m$) совпадает со значением y_d в отношении Y, значение z_s ($s=1, \dots, t$) совпадает со значением z_s в отношении Z.

Для записи операция естественного соединения на языке SQL используется ключевое слово **JOIN** (соединение). Естественное соединение обладает свойствами ассоциативности и коммутативности. Отсюда следует, что выражения:

(X JOIN Y) JOIN Z и X JOIN (Y JOIN Z)

могут быть приведены к следующему виду: X JOIN Y JOIN Z. Кроме того, выражения X JOIN Y и Y JOIN X эквивалентны.

Если отношения X и Y не имеют общих имен атрибутов, то естественное соединение превращается в этом случае в декартово произведение, т.е. выражение X JOIN Y эквивалентно выражению X TIMES Y.

Например, естественное соединение по атрибуту **Специальность** отношений **СТУДЕНЧЕСКАЯ ГРУППА** и **ПРЕДМЕТ** приведено в табл.3.3. (СТУДЕНЧЕСКАЯ ГРУППА JOIN ПРЕДМЕТ). Если значению атрибута **Специальность** отношения **СТУДЕНЧЕСКАЯ ГРУППА** не соответствует значение атрибута **Специальность** отношения **ПРЕДМЕТ** и, наоборот, значению атрибута **Специальность** отношения **ПРЕДМЕТ** не соответствует значение атрибута **Специальность** отношения **СТУДЕНЧЕСКАЯ ГРУППА**, то такие записи не присутствуют в полученном отношении.

СТУДЕНЧЕСКАЯ ГРУППА JOIN ПРЕДМЕТ

ТАБЛИЦА 3.3.

Кодовый Номер Группы	Название Группы	Количе- ство Человек	Фамилия Старосты	Специаль- ность	Кодовый Номер Предмета	Название Предмета	Кали- чество Часов	Семестр
ЭГ	Э-12	18	Иванова	ЭВМ	12П	Мини ЭВМ	76	1
ЭГ	Э-12	18	Иванова	ЭВМ	14П	ПЭВМ	72	2
ЭГ	Э-15	25	Степан	ЭВМ	12П	Мини ЭВМ	76	1
ЭГ	Э-15	22	Степан	ЭВМ	14П	ПЭВМ	72	2
АС	АС-9	24	Балабанов	АСОИ	17П	СУБД ПК	48	4
АС	АС-9	24	Балабанов	АСОИ	18П	ВКСС	52	6
АС	АС-8	20	Чирков	АСОИ	17П	СУБД ПК	48	4
АС	АС-8	20	Чирков	АСОИ	18П	ВКСС	52	6
АС	АС-14	20	Андреев	СД	44П	Финансы	80	4
АС	АС-4	21	Зяблицкин	бухучет	22П	Аудит	24	1

3.4. ОПЕРАЦИЯ СОЕДИНЕНИЯ

Операция **ТЕТА-СОЕДИНЕНИЕ** (θ -соединение) предназначается для тех случаев, когда нужно соединить вместе два отношения на основе некоторых условий, отличных от эквивалентности.

Пусть отношения X и Y не имеют общих имен атрибутов, и тета-соединение определяется так же, как и в операции выборки. Тогда θ -соединением отношения X по атрибуту x1 с отношением Y по атрибуту y1 называется результат вычисления следующего выражения:

(X TIMES Y) WHERE x1 θ y1

θ -соединение - это отношение R с тем же заголовком, что и при декартовом произведении отношений X и Y, и содержащим множество записей k, таких, что $k < s^*h$, где s - число записей отношения X, a h - число записей отношения Y, и вычисление условия "x1 θ y1" дает значение истина для этого кортежа. Атрибуты x1 и y1 должны быть совместимы по типу.

Операция θ -соединение всегда эквивалента двум операциям: нахождению расширенного декартова произведения двух отношений (при необходимости с переименованием соответствующих атрибутов) и последующему выполнению указанной выборки из полученного результата.

Например, предположим нам необходимо вычислить соединение отношения ПРЕДМЕТ с отношением СТУДЕНЧЕСКАЯ ГРУППА (в табл.3.4 приведен сокращенный вариант исходного отношения СТУДЕНЧЕСКАЯ ГРУППА для уменьшения объема записей) при выполнении условия, что значение атрибута Специальность отношения СТУДЕНЧЕСКАЯ ГРУППА расположено ниже в алфавитном порядке, чем значение атрибута отношения ПРЕДМЕТ. Это можно записать в следующем виде с условием переименования атрибута Специальность:

((СТУДЕНЧЕСКАЯ_ГРУППА RENAME Специальность AS СГ Специальность)
TIMES (ПРЕДМЕТ RENAME Специальность AS П Специальность))
WHERE П Специальность < СГ Специальность

Результирующее отношение ПРЕДМЕТ_СТУДЕНЧЕСКАЯ_ГРУППА приведено в табл.3.5.

СТУДЕНЧЕСКАЯ ГРУППА

ТАБЛИЦА 3.4.

Кодовый Номер Группы	Название Группы	Количество Человек	Специальность	Фамилия Старосты
4Г	АС-9	24	АСОИ	Баклабов
3Г	АС-8	20	АСОИ	Чижов
12Г	М-6	16	Механические машины	Трубин
10Г	Б-4	21	Бухучет	Зявоткин

ПРЕДМЕТ СТУДЕНЧЕСКАЯ ГРУППА

ТАБЛИЦА 3.5.

Кодовый Номер Предмета	Название Предмета	Коли- чество Часов	П. Специ- альность	Семестр	Кодовый Номер Группы	Название Группы	Коли- чество Человек	СГ Специ- альность	Фамилия старосты
17П	СУБД ПК	18	АСОИ	4	12Г	М-6	16	Механические машины	Трубин
18П	ВКСС	52	АСОИ	6	12Г	М-6	16	Механические машины	Трубин
22П	Аудит	24	Бухучет	3	12Г	М-6	16	Механические машины	Трубин
22П	Аудит	24	Бухучет	1	12Г	М-6	16	Механические машины	Трубин
17П	СУБД ПК	18	АСОИ	4	10Г	Б-4	21	Бухучет	Зявоткин
18П	ВКСС	52	АСОИ	6	10Г	Б-4	21	Бухучет	Зявоткин

Если в качестве θ -условия используется выражение "равно", то такое тета-соединение называется равно-соединение. Такое условие возможно только тогда, когда значения каждого из атрибутов x_1 и y_1 должны быть равны в каждой записи отношения. Если провести операцию проекции над полученным отношением с исключением атрибута x_1 или y_1 , то результатом будет просто естественное соединение. Таким образом, естественное соединение является проекцией выборки из произведения (с условием переименования атрибутов).

Например, выражение `СТУДЕНЧЕСКАЯ ГРУППА JOIN ПРЕДМЕТ`, представляющее естественное соединение отношений `СТУДЕНЧЕСКАЯ ГРУППА` и `ПРЕДМЕТ` эквивалентно следующему сложному выражению:

```
((СТУДЕНЧЕСКАЯ_ГРУППА TIMES (ПРЕДМЕТ RENAME Специальность AS
П_Специальность)) WHERE Специальность = П_Специальность)
[КодовыйНомерГруппы, НазваниеГруппы, КоличествоЧеловек, ФамилияСтаросты,
Специальности, КодовыйНомерПредмета, НазваниеПредмета, КоличествоЧасов,
Семестр]
```

3.5. ОПЕРАЦИЯ ДЕЛЕНИЯ

ДЕЛЕНИЕ (DIVIDE). Пусть имеется отношение X с заголовком $\{x_1, x_2, \dots, x_n\}$, $y_1, y_2, \dots, y_m\}$ и отношение Y с заголовком $\{y_1, y_2, \dots, y_m\}$, где атрибуты y_1, y_2, \dots, y_m общие для отношений X, Y и совпадают по типу, т.е. могут принимать значения из одного и того множества. Отношение X имеет дополнительные атрибуты x_1, x_2, \dots, x_n , а отношение Y не имеет дополнительных атрибутов.

Тогда делением отношений X и Y ($X \text{ DIVIDE } Y$) называется отношение R с заголовком $\{x_1, x_2, \dots, x_n\}$, содержащим множество всех записей таких, что для значений атрибутов x_k ($k=1, \dots, n$) отношения R соответствуют значения атрибутов x_k, y_d ($k=1, \dots, n, d=1, \dots, m$) в отношении X совпадающих со всеми значениями y_d ($d=1, \dots, m$) в отношении Y . Пример операции деления приведен на рис.3.5.

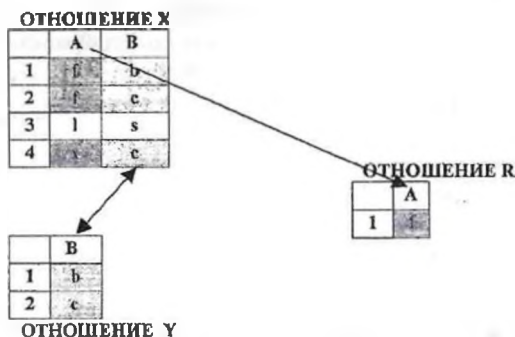


Рис.3.5. Схема выполнения операции деления.

По-другому это можно сформулировать следующим образом. Отношение R содержит такие значения x_k ($k=1, \dots, n$) из отношения X , для которых соответствующие значения y_d ($d=1, \dots, m$) (из X) включают все значения y_d ($d=1, \dots, m$) из отношения Y .

Для записи операции деления на языке SQL используется ключевое слово **DIVIDEBY** (деление).

Необходимо отметить, что использование операции деления возникают трудности в связи с пустыми отношениями, поэтому нужно быть внимательным при выполнении ее применения.

В качестве делимого воспользуемся отношением **ПРЕПОДАВАТЕЛЬ_ПРЕПОДАЕТ_ПРЕДМЕТЫ_В_ГРУППАХ** с атрибутами **КодовыйНомерГруппы** и **ЛичныйНомер**, а в качестве делителей - отношение **СТУД_ГРУППА** с атрибутом **КодовыйНомерГруппы** и отношение **S_ПРЕПОДАВАТЕЛЬ** с атрибутом **ЛичныйНомер**. Результатом выполнения операции деления являются соответственно отношения **R1_ПРЕПОДАВАТЕЛЬ** и **R2_СТУД_ГРУППА** (рис.3.6).

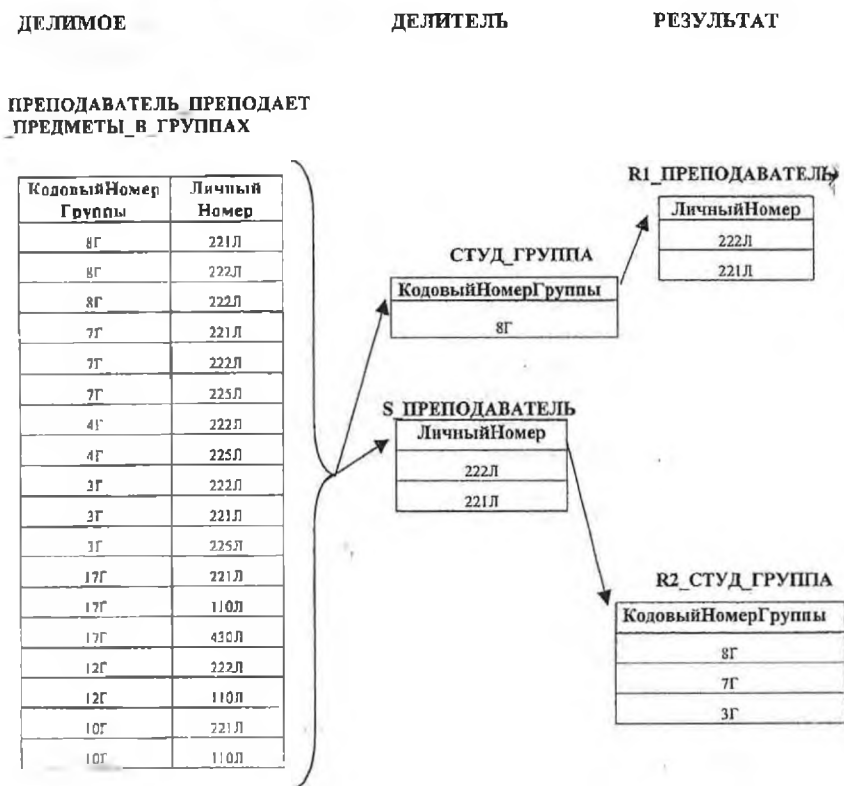


Рис.3.6. Пример выполнения операции деления

4. ИСПОЛЬЗОВАНИЕ ОПЕРАЦИЙ РЕЛЯЦИОННОЙ АЛГЕБРЫ

4.1. ПРИМЕРЫ ФОРМУЛИРОВАНИЯ ЗАПРОСОВ

Для иллюстрации примеров применения операций реляционной алгебры для описания запросов воспользуемся БД, отношения которой представлены в таблицах 4.1-4.4. Модель схемы БД представлена на рис.4.1.

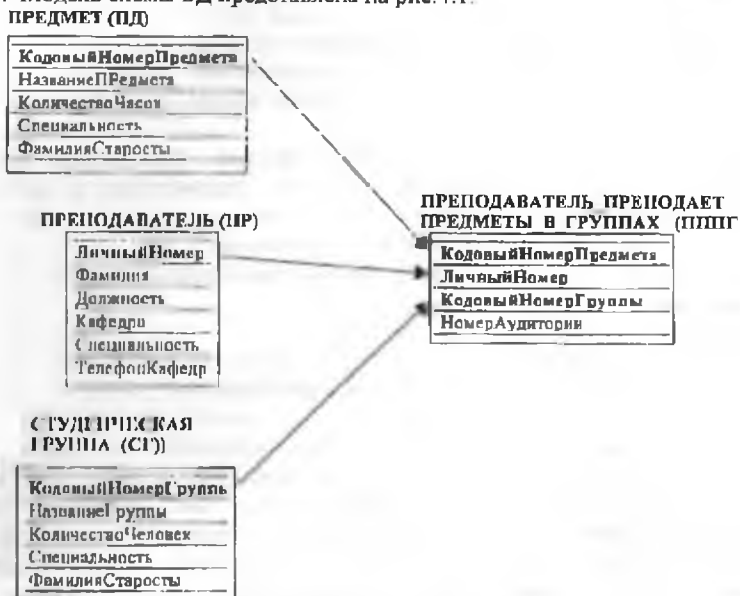


Рис.4.1. Схема базы данных

Приведенная схема и пример конкретных записей отношений БД служит концептуальной основой для составления «словесных запросов» к БД.

Под запросом к БД будем понимать постановку задачи на решение информационной задачи. Формулирование запросов заключается в определении результатов или выходных данных запроса к БД, которые необходимо получить после его выполнения. При этом не определяется способ их получения. Входной информацией для запросов является информация, хранящаяся в БД (отношения, атрибуты, связи и конкретные значения атрибутов).

Описание реализации запроса с использованием возможностей операторов реляционной алгебры заключается в определении процедуры получения выходных данных запроса на основе тех данных, которые хранятся в БД.

Процедура запроса представляется в виде совокупности реляционных операций над компонентами БД, т.е. над отношениями, записями, атрибутами и значениями атрибутов. Реляционные операции записываются в виде формул, с учетом правил использования операторов реляционной алгебры и свойства замкнутости. С целью компактности записи формул запросов воспользуемся следующими сокращенными названиями отношений рассматриваемой БД:

ПРЕПОДАВАТЕЛЬ - ПД, ПРЕДМЕТ - ПР, СТУДЕНЧЕСКАЯ ГРУППА - СТ,
ПРЕПОДАВАТЕЛЬ ПРЕПОДАЕТ ПРЕДМЕТЫ В ГРУППАХ - ПППГ.

Кроме этого, в качестве имен атрибутов отношений и конкретных их значений используется та же БД.

Методику составления запросов к БД можно представить как последовательность следующих действий:

- анализ формулировки «словесного запроса» с целью определения тех компонентов БД, которые являются входной информацией для решения поставленной в запросе задачи;
- анализ формулировки «словесного запроса» с целью определения выходной информации запроса, которая может представлять собой как конкретную информацию хранящуюся в БД, так и итоговые данные являющиеся результатом преобразования информации из БД;
- построение процедуры преобразования входной информации запроса в выходную информацию путем применения традиционных и специальных операций реляционной алгебры и представление запроса в виде формулы, объединяющей необходимые реляционные операции.

Применение данной методики рассмотрим на конкретных примерах составления запросов к БД.

Пример. 4.1.1. Необходимо определить список **Фамилий преподавателей, которые преподают предмет с КодовымНомеромПредмета = «14П»**.

В соответствие с предложенной методикой построения запросов определим входную информацию для запроса, сформулированного в примере 4.1.1. В качестве входной информации выступают отношения ПД и ПППГ, атрибуты которых определены в условии задачи (атрибут **Фамилия** преподавателей из отношения ПР и атрибут **КодовыеНомераПредмета** из отношения ПППГ, а также конкретное значение последнего атрибута = «14П»). Необходимый фрагмент схемы БД для построения запроса представлен на рис.4.2.

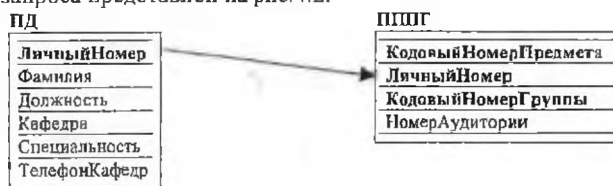


Рис.4.2. Схема связей между отношениями ПД и ПППГ.

В качестве результатов выполнения запроса должен быть список значений атрибута **Фамилия** из отношения ПД, который удовлетворяет заданным условиям.

Общую схему алгоритма процесса решения данной задачи можно представить в виде следующих действий:

1. Определение всех **КодовыхНомеровПредметов**, которые ведут преподаватели. Это можно реализовать путем соединения отношений ПД и ПППГ по атрибуту **ЛичныйНомер**, который является связующим между этими отношениями. Выполнение реляционной операции соединения ПД и ПППГ отношений дополнит каждую запись отношения ПППГ соответствующей информацией о преподавателе (т.е. **Фамилия, Должность, Кафедра** и т.д.).

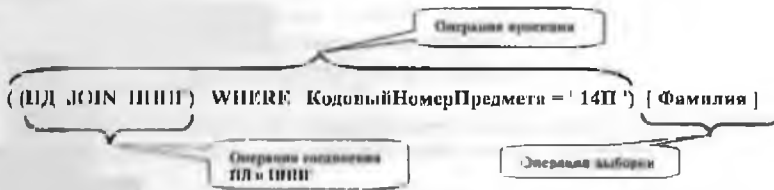
- Выборка из записей, полученных при выполнении первого действия, тех записей, для которых выполняется условие **КодовыйНомерПредмета = 14П**.
- Для записей, полученных после выполнения действия 2, выполняется операция проекции выборки по атрибуту **Фамилия**.

Конечным результатом решения задачи является отношение ПД 1, которое состоит из одного атрибута, содержащего список значений атрибута **Фамилия** преподавателей преподающих предмет с **КодовымНомеромПредмета = «14П»**. Для рассматриваемой БД результат запроса представлен в таблице 4.1.

ПД 1 ТАБЛИЦА 4.1.

Фамилия
Фролов
Костин

Данный алгоритм процесса решения поставленной задачи можно представить в виде запроса к БД, который можно определить формулой, использующей рассмотренные в разделах 2 и 3 традиционных и специальных реляционных операций:



Пример 4.1.2. *Определить список Фамилий преподавателей, которые ведут предмет с НазваниемПредмета = «МиниЭВМ».*

По аналогии с предыдущим примером запрос к БД можно представить в виде формулы, используя уже известные операции:

(((IP WHERE НазваниеПредмета = 'МиниЭВМ') JOIN ППП) [ЛичныйНомер]
JOIN ПД) [Фамилия]

Результатом выполнения запроса является отношение с единственным атрибутом **Фамилия** для вновь построенного отношения ПД 2.

Для данного запроса можно привести и другую формулировку процесса решения той же задачи. Например:

(((IP WHERE НазваниеПредмета = 'МиниЭВМ') [КодовыйНомерПредмета] JOIN ППП)
JOIN ПД) [Фамилия]

Таким образом, рассмотренный пример подчеркивает одно важное обстоятельство: возможность сформулировать один и тот же запрос несколькими способами. Результат выполнения запроса для нашего примера представлен в таблице 4.2.

ПД 2 ТАБЛИЦА 4.2.

Фамилия
Костин

Пример. 4.1.3. Определить список Фамилий преподавателей, которые преподают какой-либо предмет во всех студенческих группах.

Для последовательного формулирования алгоритма процесса решения задачи или построения результирующей формулы запроса к БД воспользуемся пошаговым способом описания процесса решения задачи, обозначая каждый шаг через T_n, где n- номер шага.

Тогда на первом шаге T₁ для данного запроса из примера 4.1.3 произведем проекцию над отношением ШПП по атрибутам КодовыйНомерГруппы и ЛичныйНомер. Эту операцию можно представить следующим образом:

T₁: ШПП [КодовыйНомерГруппы, ЛичныйНомер]

На втором шаге произведем операцию проекции отношения СГ по атрибуту КодовыйНомерГруппы. Выражение, соответствующее этой операции имеет вид:

T₂: СГ [КодовыйНомерГруппы]

На третьем шаге разделим результат первого шага на результат второго шага. Таким образом, получим значения ЛичныхНомеров преподавателей, которые ведут занятия во всех группах. Эту операцию можно записать в виде:

T₃: ШППГ [КодовыйНомерГруппы, ЛичныйНомер]
 DIVIDE BY СГ [КодовыйНомерГруппы]

На четвертом шаге соединим отношение, полученное на шаге 3 с отношением преподаватель – ПД, таким образом к каждой записи результирующего отношения шага 3 добавятся сведения о преподавателях:

T₄: (ШППГ [КодовыйНомерГруппы, ЛичныйНомер]
 DIVIDE BY СГ [КодовыйНомерГруппы]) JOIN ПД

На последнем пятом шаге сделаем проекцию отношения, полученного на шаге 4, по атрибуту Фамилия:

T₅:
 ((ШППГ[КодовыйНомерГруппы,ЛичныйНомер] DIVIDE BY ПД[ЛичныйНомер]) JOIN ПД [Фамилия])



Формула представленная на шаге 5 представляет собой запрос, который реализует задачу, описанную в примере 4.1.3. Результатом выполнения данного запроса для рассматриваемого примера БД является отношение с атрибутом Фамилия, имеющим единственное значение, совпадающее с результатом приведенным в примере 4.1.2. (табл. 4.2).

Пример. 4.1.4. Определить список Фамилий преподавателей, которые преподают такой же предмет, что и преподаватель с ЛичнымНомером = «222Л».

Пошаговую запись процесса решения задачи в виде совокупности реляционных операций для данного запроса можно представить следующим образом:

T1: ПППГ [КодовыйНомерПредмета, ЛичныйНомер]

T2: ПД [ЛичныйНомер] WHERE ЛичныйНомер = «222Л»

T3: ПППГ [КодовыйНомерПредмета, ЛичныйНомер]
 DIVIDEBY (ПД [ЛичныйНомер] WHERE ЛичныйНомер = «222Л»)

T4: (ПППГ [КодовыйНомерПредмета, ЛичныйНомер]
 DIVIDEBY (ПД [ЛичныйНомер] WHERE ЛичныйНомер = «222Л»))
 JOIN ПППГ

T5: ((ПППГ [КодовыйНомерПредмета, ЛичныйНомер]
 DIVIDEBY (ПД [ЛичныйНомер] WHERE ЛичныйНомер = «222Л»))
 JOIN ПППГ) JOIN ПД

T6: (((ПППГ [КодовыйНомерПредмета, ЛичныйНомер]
 DIVIDEBY (ПД [ЛичныйНомер] WHERE ЛичныйНомер = «222Л»)) JOIN ПППГ)
 JOIN ПД) [Фамилия]

В результате выполнения шага 6 полностью реализуется указанный в примере запрос, результатом которого является отношение ПД_3 (табл. 4.3.).

ПД_3 ТАБЛИЦА 4.3.

Фамилия
Фролов

Пример. 4.1.5. Определить список Фамилий преподавателей, которые не преподают предмет с НазваниемПредмета = «14П».

Решение данной задачи можно представить в виде следующей совокупности шагов.

T1: ПППГ WHERE КодовыйНомерПредмета = «14П»

T2: (ПППГ WHERE КодовыйНомерПредмета = «14П») [ЛичныйНомер]

T3: ПД [ЛичныйНомер]

T4: ПД [ЛичныйНомер]
 MINUS (ПППГ WHERE КодовыйНомерПредмета = «14П») [ЛичныйНомер]

T5: (ПД [ЛичныйНомер] MINUS
 (ПППГ WHERE КодовыйНомерПредмета = «14П») [ЛичныйНомер]) JOIN ПД

T6: ((ПД [ЛичныйНомер] MINUS
 (ПППГ WHERE КодовыйНомерПредмета = «14П») [ЛичныйНомер]) JOIN ПД)
 [Фамилия]

Формула, представленная на шаге 6, является завершающей и реализует полностью задачу из примера 4.1.5. Результаты выполнения запроса для рассматриваемой БД представлены в таблице 4.4.

ПД 4 ТАБЛИЦА 4.4.

Фамилия
Бойко
Глазов
Петров

4.2. УПРАЖНЕНИЯ ДЛЯ СОСТАВЛЕНИЯ ЗАПРОСОВ

Приведенный ниже список упражнений представляет собой формулировки «словесных запросов» к БД (схема отношений БД представлена на рис.4.1) или постановки задач на обработку информации, которая хранится в БД.

На основе этих «запросов» необходимо составить алгоритм решения поставленной задачи и представить его в виде формулы запроса, состоящей из последовательности реляционных операций над отношениями БД.

Для проверки правильности составленных запросов можно воспользоваться примером записей отношений БД из раздела 1 (табл. 1.1.-1.4.).

СПИСОК УПРАЖНЕНИЙ ДЛЯ СОСТАВЛЕНИЯ ЗАПРОСОВ К БД

- 4.2.1. Определить всю информацию о преподавателях.
- 4.2.2. Определить всю информацию о студенческих группах по **Специальности** с названием «ЭВМ».
- 4.2.3. Определить **ЛичныйНомерПреподавателя** и перечень **НомеровАудиторий**, в которых этот преподаватель преподает предмет с **КодовымНомеромПредмета** равным «18П».
- 4.2.4. Определить перечень **КодовыхНомеровПредметов** и **НазванийПредметов**, которые ведет преподаватель с **Фамилией** «Костин».
- 4.2.5. Определить перечень **КодовыхНомеровГрупп**, в которых преподаватель с **Фамилией** «Фролов» ведет предметы.
- 4.2.6. Определить всю информацию о предметах, которые проводятся для студентов по **Специальности** «АСОИ».
- 4.2.7. Определить всю информацию о преподавателях, которые ведут предметы для студентов по **Специальности** «АСОИ».
- 4.2.8. Определить фамилии преподавателей, которые ведут предметы в аудитории **НомерАудитории** = «210».
- 4.2.9. Определить перечень **НазванийПредметов** и **НазванийГруппы** студентов, для которых занятия проводятся в аудиториях начиная с **НомерАудитории**=100 и по **НомерАудитории**=200 включительно.
- 4.2.10. Определить общее количество студентов, которые обучающихся по **Специальности** с названием «ЭВМ».
- 4.2.11. Определить перечень **ЛичныхНомеров** преподавателей, обучающихся студентов по **Специальности** с названием «Бухучет».
- 4.2.12. Определить список **НазванийПредметов**, которые изучаются всеми студенческими группами.
- 4.2.13. Определить список **Фамилий** преподавателей, преподающих те же предметы, что и преподаватель преподающий предмет с **КодовымНомеромПредмета** равным «14П».

- 4.2.14. Определить информацию о предметах, которые не ведет преподаватель с **ЛичнымНомером** равным «2211».
- 4.2.15. Определить информацию о предметах, которые не изучаются в группе с **НазваниемГруппы** равным «М-6».
- 4.2.16. Определить информацию о доцентах, преподающих в группах с **КодовымНомеромГруппы** «ЗГ» и «ВГ».
- 4.2.17. Определить список **КодовыхНомеровПредметов**, **ЛичныхНомеровПреподавателей** и **КодовыйНомерГруппы**, в которых преподаватели с кафедры «ЭВМ» ведут занятия и имеют образование по **Специальности** «АСОИ».
- 4.2.18. Определить список **КодовыхНомеровГрупп** студентов, у которых название **Специальности** совпадает с названием **Специальности** у преподавателей, ведущие у них занятия.
- 4.2.19. Определить список **КодовыхНомеровПреподавателей** с кафедры «ЭВМ», преподающих предметы по **Специальности**, совпадающей со **Специальностью** студенческой группы.
- 4.2.20. Определить список **Специальностей** студентов, в которых преподаватели кафедры «АСУ» проводят занятия.
- 4.2.21. Определить список **КодовыхНомеровПредметов** для предметов, которые изучает студенческая группа с **НазваниемГруппы** равным «АС-8»
- 4.2.22. Определить список **КодовыхНомеровГруппы** студентов, которые изучают те же предметы, что и студенческая группа с **НазваниемГруппы** «С-4»
- 4.2.23. Определить список **КодовыхНомеровГрупп** студентов, которые не изучают предметы, преподаваемые в студенческой группе с **НазваниемГруппы** «Б-8»
- 4.2.24. Определить список **КодовыхНомеровГруппы** студентов, которые не изучают предметы, преподаваемые преподавателем с **ЛичнымНомером** равным «430Д».
- 4.2.25. Определить список **ЛичныхНомеров** преподавателей, которые проводят занятия в группе с **НазваниемГруппы** «Э-15», но не преподающих предмет с **КодовымНомеромПредмета** равным «121»

СПИСОК ЛИТЕРАТУРЫ

1. Брич В.Г., Хвещук В.И. Методические указания по дисциплине «Базы и банки данных», Часть3. «Проектирование реляционных баз данных» для студентов специальности «Автоматизированные системы обработки информации»: - Брест, БГТУ, 2001 - 32с.
2. Дейт К.Дж. Введение в системы баз данных: Пер. с англ. - 6-е изд. - К.:Диалектика, 1998.-784с.
3. Базы данных: Учебник для высших учебных заведений: Под ред.А.Д.Хомоненко. -СПб.: КОРОНА принт, 2000.-416с.
4. Цикритзис Д., Лоховски Ф. Модели данных: Пер.с англ. - М.:Финансы и статистика, 1985.- 344с.
5. Ульман Дж. Основы баз данных: Пер. с англ. - М.: Финансы и статистика, 1983 -334с

СПИСОК СОКРАЩЕНИЙ

- БД - база данных
 ПД - сокращенное название таблицы ПРЕПОДАВАТЕЛЬ
 ПП - сокращенное название таблицы ПРЕДМЕТ
 СГ - сокращенное название таблицы СТУДЕНЧЕСКАЯ_ГРУППА
 ППП - сокращенное название таблицы ПРЕПОДАВАТЕЛЬ_ПРЕПОДАЕТ_ПРЕДМЕТЫ_В_ГРУППАХ
 РМД - реляционная модель данных
 СУБД - система управления базами данных

УЧЕБНОЕ ИЗДАНИЕ

Составители: Виктор Григорьевич Брич
Владимир Иванович Хвещук

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по дисциплине
«БАЗЫ И БАНКИ ДАННЫХ»
Часть 4. Операции реляционной алгебры

для студентов специальности
«Автоматизированные системы обработки информации»

Ответственный за выпуск: Брич В.Г.

Редактор: Строкач Т.В.

Подписано к печати 13.04.2001г. Офсетная печать. Бумага писч. Формат 60x84 1/16.
Заказ № 246. Усл.печ.л. 1,6. Уч.изд.л. 1,75. Тираж 100. Отпечатано на ризографе
Брестского государственного технического университета. 224017, Брест, Московская, 267.