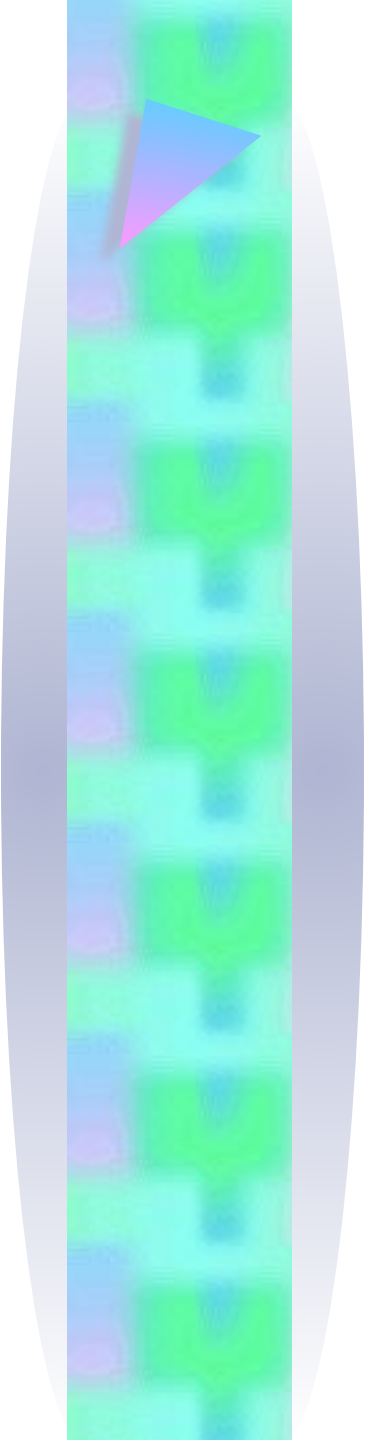
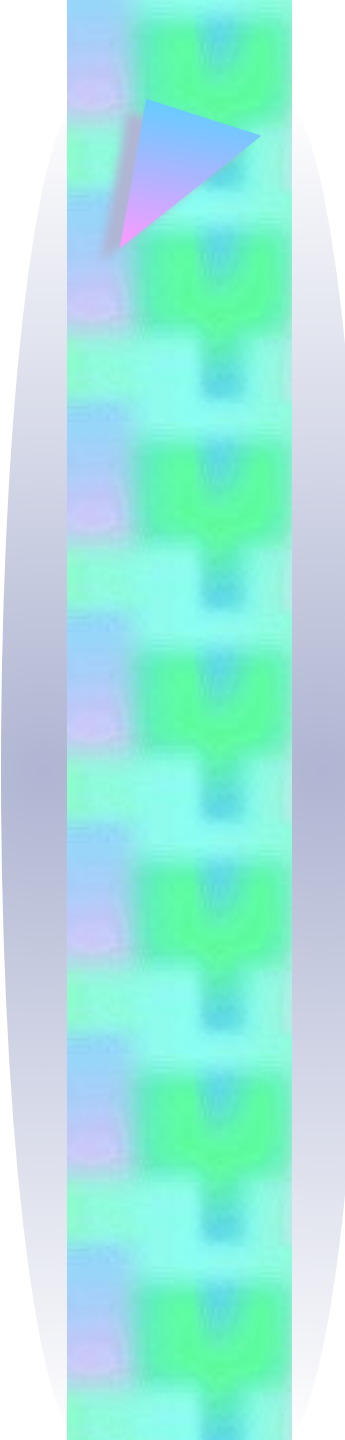




Тема лекции:

**Реляционная алгебра и
реляционное исчисление**



Структура занятия:

1. Постановка цели занятия
2. Актуализация знаний
3. Постановка проблемной задачи
4. Изучение нового материала
5. Подведение итогов занятия
6. Постановка задания для СРС

Цели:

- ✓ изучить возможности формирования запросов к реляционной базе данных, вычисляемых путем выполнения последовательности реляционных операторов, применяемым к отношениям;
- ✓ изучить возможности декларативной логической формулировки запросов, выполняемых путем интерпретации логических формул.



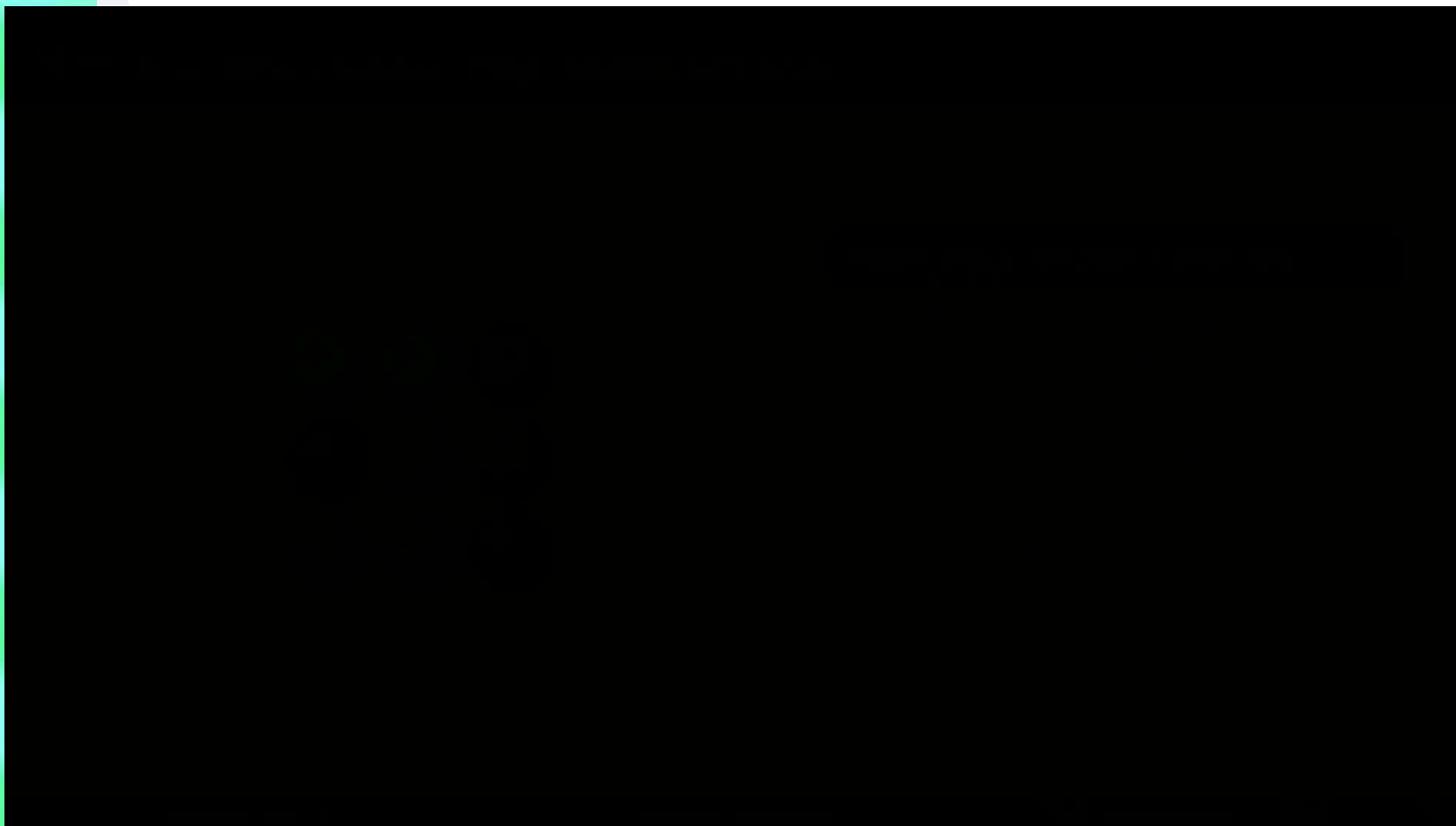
План лекции:

1. Классификация операций
реляционной алгебры.
2. Теоретико-множественные операции.
3. Специальные реляционные
операции.
4. Реляционное исчисление кортежей.
5. Реляционное исчисление доменов.

Актуализация знаний обучающихся

Реляционный термин	Соответствующий «табличный» термин
База данных	Набор
Схема базы данных	таблиц Набор заголовков
Отношение	таблиц Таблиц
Заголовок отношения	а Заголовок
Тело отношения	таблицы Тело
Атрибут отношения	таблицы Наименование столбца
Кортеж отношения	таблицы Строка
Степень (-арность) отношения	таблицы Количество столбцов
Мощность отношения	таблицы Количество строк
Домены и типы данных	таблицы Типы данные в ячейках таблицы

Актуализация знаний обучающихся





Постановка проблемной задачи

Одной из основных функций БД является возможность выполнения разнообразных запросов пользователей. Выясним, что лежит в основе языков построения запросов к набору связанных отношений, в результате выполнения которых будет получено одно или несколько других отношений с искомыми данными в нужном сочетании.

Реляционная алгебра

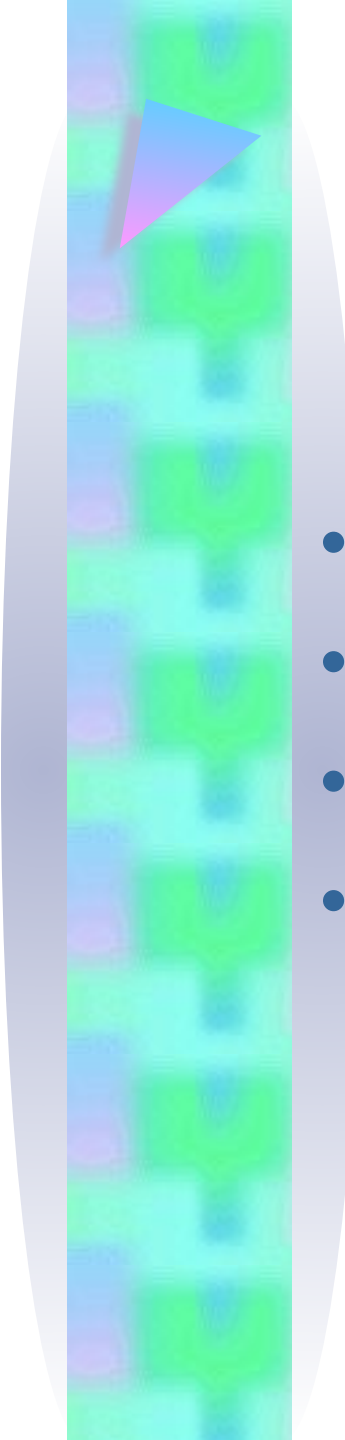
Т.к. отношения являются множествами, то средства манипулирования отношениями могут базироваться на традиционных теоретико-множественных операциях, дополненных некоторыми специальными операциями, специфичными для баз данных.



Реляционная алгебра

Начальный вариант алгебры был предложен Коддом.

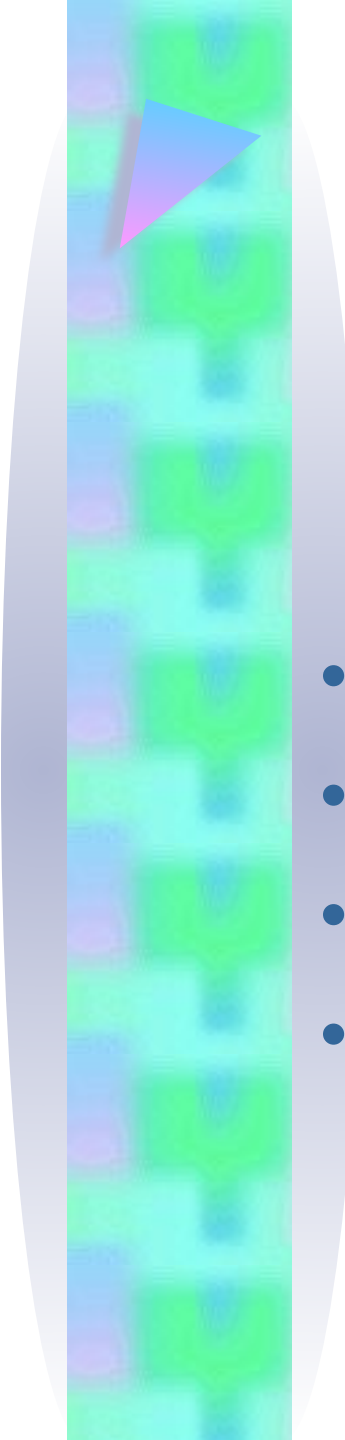
Набор основных алгебраических операций состоит из восьми операций, которые делятся на два класса - теоретико-множественные операции и специальные реляционные операции.



Реляционная алгебра

В состав теоретико-множественных операций входят операции:

- объединения отношений;
- пересечения отношений;
- взятия разности отношений;
- прямого произведения отношений.



Реляционная алгебра

Специальные реляционные операции включают:

- ограничение отношения;
- проекцию отношения;
- соединение отношений;
- деление отношений.



Реляционная алгебра

В состав алгебры включается:

- операция присваивания, позволяющая сохранить в базе данных результаты вычисления алгебраических выражений,
- операция переименования атрибутов, дающая возможность корректно сформировать заголовок результирующего отношения.



Реляционная алгебра

Реляционная алгебра представляет собой набор операторов, использующих отношения в качестве аргументов, и возвращающие отношения в качестве результата.

Реляционный оператор f - функция с отношениями в качестве аргументов:

$$R = f(R1, R2, ..., Rn)$$

Реляционная алгебра

Реляционная алгебра является замкнутой, т.к. в качестве аргументов в реляционные операторы можно подставлять другие реляционные операторы, подходящие по типу:

$$R = f(f_1(R_{11}, R_{12}, \dots), f_2(R_{21}, R_{22}, \dots), \dots)$$

Реляционная алгебра

Каждое отношение обязано иметь уникальное имя в пределах базы данных. Имя отношения, полученного в результате выполнения реляционной операции, определяется в левой части равенства.

Не требуется наличия имен от отношений, полученных в результате реляционных выражений, если эти отношения подставляются в качестве аргументов в другие реляционные выражения. Такие отношения называют неименованными отношениями.

Неименованные отношения реально не существуют в базе данных, а только вычисляются в момент вычисления значения реляционного оператора.

Реляционная алгебра

Отношения называют совместимыми по типу, если они имеют идентичные заголовки, а именно:

- отношения имеют одно и то же множество имен атрибутов;
- атрибуты с одинаковыми именами определены на одних и тех же доменах.

Оператор переименования атрибутов

Синтаксис оператора переименования атрибутов :

R RENAME Atr1, Atr2,... AS NewAtr1,New Atr2,...

где

R – отношение;

Atr1, Atr2, - исходные имена атрибутов;

NewAtr1,New Atr2 - новые имена атрибутов.

Объединение

Объединением двух совместимых по типу отношений A и B называется отношение с тем же заголовком, что и у отношений A и B , и телом, состоящим из строк (кортежей), принадлежащих или A , или B , или обоим отношениям.

Синтаксис операции объединения:

$A \text{ UNION } B$

Таблица 1. Отношение A

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Таблица 2. Отношение B

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Пушников	2500
4	Сидоров	3000

Таблица 3. Отношение A UNION B

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000
2	Пушников	2500
4	Сидоров	3000

Пересечение

Пересечением двух совместимых по типу отношений A и B называется отношение с тем же заголовком, что и у отношений A и B , и телом, состоящим из кортежей, принадлежащих одновременно обоим отношениям A и B .

Синтаксис операции пересечения:

$A \text{ INTERSECT } B$

Таблица 1. Отношение A

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Таблица 2. Отношение B

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Пушников	2500
4	Сидоров	3000

Таблица 4. Отношение A INTERSECT B

Табельный номер	Фамилия	Зарплата
1	Иванов	1000

Вычитание

Вычитанием двух совместимых по типу отношений A и B называется отношение с тем же заголовком, что и у отношений A и B , и телом, состоящим из кортежей, принадлежащих отношению A и не принадлежащих отношению B .

Синтаксис операции вычитания:
 $A \text{ MINUS } B$

Таблица 1. Отношение А

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Таблица 2. Отношение В

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Пушников	2500
4	Сидоров	3000

Таблица 5. Отношение А MINUS В

Табельный номер	Фамилия	Зарплата
2	Петров	2000
3	Сидоров	3000

Декартово произведение

Декартовым произведением двух отношений $A(A_1, A_2, \dots, A_n)$ и $B(B_1, B_2, \dots, B_n)$ называется отношение, заголовок которого является сцеплением заголовков отношений A и B : $(A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_n)$, а тело состоит из кортежей, являющихся сцеплением кортежей отношений A и B : $(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n)$, таких, что

$$\begin{aligned}(a_1, a_2, \dots, a_n) &\in A, \\ (b_1, b_2, \dots, b_n) &\in B\end{aligned}$$

Синтаксис операции декартового произведения: $A \text{ TIMES } B$

Таблица 6. Отношение A (Поставщики)

Номер поставщика	Наименование поставщика
1	Завод 1
2	Завод 2
3	Завод 3

Таблица 7. Отношение B (Детали)

Номер детали	Наименование детали
1	Болт
2	Гайка
3	Винт

Декартово произведение отношений A и B будет иметь вид:

Таблица 8. Отношение A TIMES B

Номер поставщика	Наименование поставщика	Номер детали	Наименование детали
1	Завод 1	1	Болт
1	Завод 1	2	Гайка
1	Завод 1	3	Винт
2	Завод 2	1	Болт
2	Завод 2	2	Гайка
2	Завод 2	3	Винт
3	Завод 3	1	Болт
3	Завод 3	2	Гайка
3	Завод_3	3	Винт

Выборка (ограничение, селекция)

Выборкой на отношении A с условием z называется отношение с тем же заголовком, что и у отношения A , и телом, состоящем из кортежей, значения атрибутов которых при подстановке в условие z дают значение ИСТИНА. « Z » представляет собой логическое выражение, в которое могут входить атрибуты отношения A и скалярные выражения.

В простейшем случае условие z имеет вид $X\Theta Y$, где Θ - один из операторов сравнения, а X и Y - атрибуты отношения A или скалярные значения. Такие выборки называются Θ -выборки (тэта-выборки) или Θ -ограничения, Θ -селекции.

Синтаксис операции выборки:

A WHERE z или A WHERE $X\Theta Y$

Пример. Пусть дано отношение A с информацией о сотрудниках:

Таблица 9. Отношение A

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Результат выборки A WHERE *Зарплата* <3000 будет иметь вид:

Таблица 10. Отношение A WHERE *Зарплата*<3000

Табельный номер	Фамилия	Зарплата
1	Иванов	1000
2	Петров	2000

Проекция

Проекцией отношения A по атрибутам $X, Y, \dots, Z$, где каждый из атрибутов принадлежит отношению A , называется отношение с заголовком $(X, Y, \dots, Z)$ и телом, содержащим множество кортежей вида $(x, y, \dots, z)$, таких, для которых в отношении A найдутся строки (кортежи) со значением атрибута X равным x , значением атрибута Y равным y , ..., значением атрибута Z равным z .

Синтаксис операции проекции:

$A[X, Y, \dots, Z]$



Пример. Пусть дано отношение A с информацией о поставщиках, включающих их наименование и месторасположение:

Таблица 11. Отношение A (Поставщики)

Номер поставщика	Наименование поставщика	Город поставщика
1	Завод_1	Уфа
2	Завод_2	Москва
3	Завод_3	Москва
4	Завод_4	Челябинск

Проекция A [Город поставщика] будет иметь вид:

Таблица 12. Отношение A[Город поставщика]

Город поставщика
Уфа
Москва
Челябинск

Соединение

Соединением отношений A и B по условию z называется отношение

$(A \text{ TIMES } B) \text{ WHERE } z$

где z представляет собой логическое выражение, в которое могут входить атрибуты отношений A и B и скалярные выражения.

Таблица 13 Отношение А (Поставщики)

Номер поставщика	Наименование поставщика
1	Иванов
2	Петров
3	Сидоров

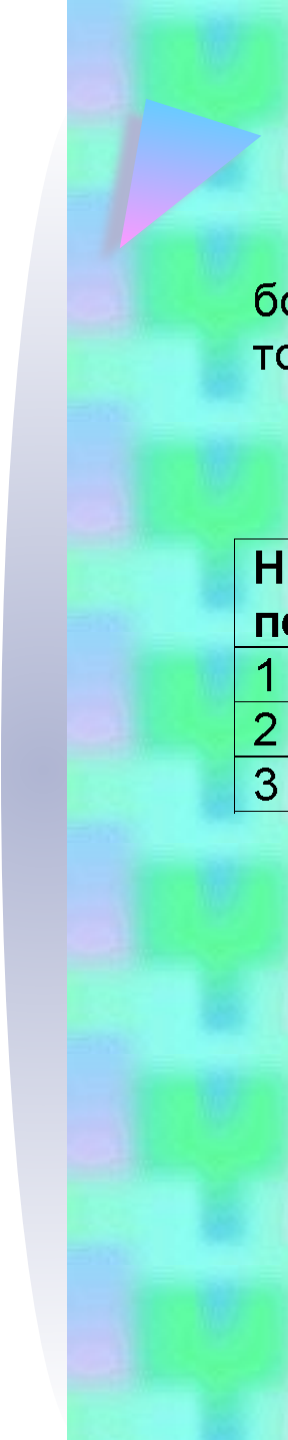
Таблица 14. Отношение В (Детали)

Номер детали	Наименование детали
1	Болт
2	Гайка
3	Винт

Декартово произведение отношений А и В будет иметь вид:

Таблица 15. Отношение А TIMES В

Номер поставщика	Наименование поставщика	Номер детали	Наименование детали
1	Иванов	1	Болт
1	Иванов	2	Гайка
1	Иванов	3	Винт
2	Петров	1	Болт
2	Петров	2	Гайка
2	Петров	3	Винт
3	Сидоров	1	Болт
3	Сидоров	2	Гайка
3	Сидоров	3	Винт



Если к нему приложить условие «Какие поставщики поставляют болты» то есть «Какие поставщики поставляют детали с номером 1», то в результате получим отношение

Таблица 16. Отношение A TIMES B WHERE Номер детали = 1

Номер поставщика	Наименование поставщика	Номер детали	Наименование детали
1	Иванов	1	Болт
2	Петров	1	Болт
3	Сидоров	1	Болт

Тэта-соединение

Пусть отношение A содержит атрибут X , отношение B содержит атрибут Y , а Θ - один из операторов сравнения (и т.д.).

Тогда Θ -соединением отношения A по атрибуту X с отношением B по атрибуту Y называют отношение

$(A \text{ TIMES } B) \text{ WHERE } X\Theta Y.$

Синтаксис операции Θ -соединения :

$A [X\Theta Y] B.$

Таблица 17. Отношение A (Поставщики)

Номер поставщика	Наименование поставщика	Статус поставщика (X)
1	Иванов	4
2	Петров	1
3	Сидоров	2

Таблица 18. Отношение B (Детали)

Номер детали	Наименование детали	Статус детали (Y)
1	Болт	3
2	Гайка	2
3	Винт	1

Ответ на вопрос "какие поставщики имеют право поставлять какие детали?" дает Θ -соединение $A [X \geq Y]$

Таблица 19. Отношение $A [X \geq Y]$ B

Номер поставщика	Наименование поставщика	X (Статус поставщика)	Номер детали	Наименование детали	Y (Статус детали)
1	Иванов	4	1	Болт	3
1	Иванов	4	2	Гайка	2
1	Иванов	4	3	Винт	1
2	Петров	1	3	Винт	1
3	Сидоров	2	2	Гайка	2
3	Сидоров	2	3	Винт	1

Экви-соединение

Наиболее важным частным случаем Θ - соединения является случай, когда Θ есть просто равенство.

Синтаксис операции экви-соединения :
 $A [X=Y] B.$

Таблица 20. Отношение A (Поставщики)

Номер поставщика	Наименование поставщика	Статус поставщика (X)
1	Иванов	4
2	Петров	1
3	Сидоров	2

Таблица 21. Отношение B (Детали)

Номер детали	Наименование детали	Статус изделия (Y)
1	Болт	3
2	Гайка	2
3	Винт	1

Таблица 22. Отношение A [X=Y] B

Номер поставщика	Наименование поставщика	X (Статус поставщика)	Номер детали	Наименование детали	Y (Статус изделия_1)
3	Сидоров	2	2	Гайка	2

Естественное соединение

Пусть даны отношения A ($A_1, A_2, \dots, A_n, X_1, X_2, \dots, X_n$) и $B(X_1, X_2, \dots, X_n, B_1, B_2, \dots, B_n)$, имеющие одинаковые атрибуты $X_1, X_2, \dots, X_n$.

Тогда естественным соединением отношений A и B называется отношение с заголовком ($A_1, A_2, \dots, A_n, X_1, X_2, \dots, X_n, B_1, B_2, \dots, B_n$) и телом, содержащим множество кортежей $(a_1, a_2, \dots, a_n, x_1, x_2, \dots, x_n, b_1, b_2, \dots, b_n)$, таких, что $(a_1, a_2, \dots, a_n, x_1, x_2, \dots, x_n) \in A$ и $(x_1, x_2, \dots, x_n, b_1, b_2, \dots, b_n) \in B$.

Синтаксис: $A \text{ JOIN } B$

Таблица 23. Отношение Р (Поставщики)

Номер поставщика (PNUM)	Наименование поставщика (PNAME)
1	Иванов
2	Петров
3	Сидоров

Таблица 24. Отношение D (Детали)

Номер детали (DNUM)	Наименование детали (DNAME)
1	Болт
2	Гайка
3	Винт

Таблица 25. Отношение PD (Поставки)

Номер поставщика (PNUM)	Номер детали (DNUM)	Поставляемое количество VOLUME
1	1	100
1	2	200
1	3	300
2	1	150
2	2	250
3	1	1000

Ответ на вопрос "какие детали поставляются поставщиками", можно записать в виде естественного соединения трех отношений *P JOIN PD JOIN D*):

Таблица 26. Отношение *P JOIN PD JOIN D*

Номер поставщика PNUM	Наименование поставщика PNAME	Номер детали DNUM	Наименование детали DNAME	Поставляемое количество VOLUME
1	Иванов	1	Болт	100
1	Иванов	2	Гайка	200
1	Иванов	3	Винт	300
2	Петров	1	Болт	150
2	Петров	2	Гайка	250
3	Сидоров	1	Болт	1000

Деление

Пусть отношение A определено на множестве атрибутов X , а отношение B - на множестве атрибутов Y , причем Y является подмножеством X .

Пусть $Z = X - Y$, то есть Z является множеством атрибутов отношения A , которые не являются атрибутами отношения B .

Результатом операции деления является набор кортежей отношения A , определенных на множестве атрибутов Z , которые соответствуют комбинации всех строк отношения B .

Синтаксис операции деления:

$A \text{ DEVIDBY } B$

Пример. В примере с поставщиками, деталями и поставками ответим на вопрос, "какие поставщики поставляют все детали?".

В качестве делимого возьмем проекцию $X = PD [PNUM, DNUM]$, содержащую номера поставщиков и номера поставляемых ими деталей:

Таблица 27. Проекция $X = PD [PNUM, DNUM]$

Номер поставщика PNUM	Номер детали DNUM
1	1
1	2
1	3
2	1
2	2
3	1

В качестве делителя возьмем проекцию $Y = D [DNUM]$, содержащую список номеров всех деталей (не обязательно поставляемых кем-либо):

Таблица 29. Проекция $Y = D [DNUM]$

Номер детали DNUM
1
2
3

Деление $X \text{ DEVIDEBY } Y$ дает список номеров поставщиков, поставляющих все детали:

Таблица 29. Отношение $X \text{ DEVIDEBY } Y$

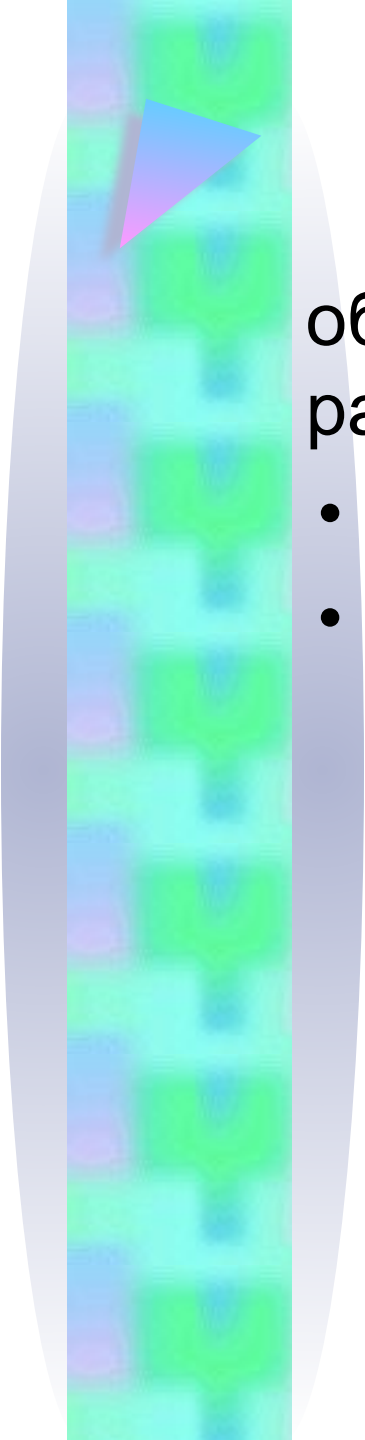
Номер поставщика PNUM
1

Оказалось, что только поставщик с номером 1 поставляет все детали.

Реляционное исчисление

Реляционное исчисление является прикладной ветвью формального механизма исчисления предикатов первого порядка.

В основе исчисления лежит понятие переменной с определенной для нее областью допустимых значений и понятие правильно построенной формулы, опирающейся на переменные, предикаты и кванторы.



Реляционное исчисление

В зависимости от того, что является областью определения переменной, различают:

- исчисление кортежей
- исчисление доменов.

Реляционное исчисление

В исчислении кортежей областями определения переменных являются тела отношений базы данных, т. е. допустимым значением каждой переменной является кортеж некоторого отношения.

В исчислении доменов областями определения переменных являются домены, на которых определены атрибуты отношений базы данных, т. е. допустимым значением каждой переменной является значение некоторого домена.

Исчисление кортежей

СЛУЖАЩИЕ

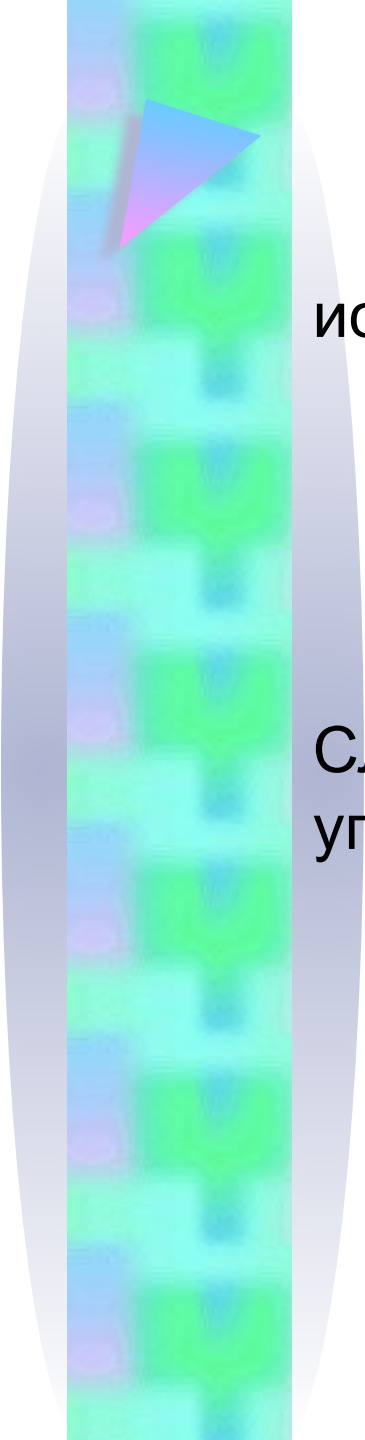
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2935	Петров	29600.00	1
2936	Сидоров	18000.00	1
2937	Федоров	20000.00	1
2938	Иванова	22000.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	2
2939	Сидоренко	18000.00	2
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	2

ПРОЕКТЫ

ПРО_НОМ	ПРОЕКТ_РУК
1	Иванов
2	Иваненко

НОМЕРА_ПРОЕКТОВ

ПРО_НОМ
1
2



Исчисление кортежей

Для определения кортежной переменной используется оператор RANGE.

RANGE СЛУЖАЩИЙ IS СЛУЖАЩИЕ

Чтобы сослаться на значение атрибута СЛУ_ИМЯ переменной СЛУЖАЩИЙ, нужно употребить конструкцию

СЛУЖАЩИЙ.СЛУ_ИМЯ.

Правильно построенные формулы

Правильно построенная формула (Well-Formulated Formula, WFF) служит для выражения условий, накладываемых на кортежные переменные.

Простые условия представляют собой операции сравнения скалярных значений.

Сложные варианты строятся с помощью логических связок NOT, AND, OR и IF ... THEN с учетом обычных приоритетов операций (NOT > AND > OR) и возможности расстановки скобок.

Правильно построенные формулы (WFF)

СЛУЖАЩИЕ

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2935	Петров	29600.00	1
2936	Сидоров	18000.00	1
2937	Федоров	20000.00	1
2938	Иванова	22000.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	2
2939	Сидоренко	18000.00	2
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	2

ПРОЕКТЫ

ПРО_НОМ	ПРОЕКТ_РУК
1	Иванов
2	Иваненко

НОМЕРА_ПРОЕКТОВ

ПРО_НОМ
1
2

СЛУЖАЩИЙ.
СЛУ_ИМЯ=
ПРОЕКТ.
ПРОЕКТ_РУК

СЛУЖАЩИЙ. СЛУ_НОМЕР	СЛУЖАЩИЙ. СЛУ_ИМЯ	СЛУЖАЩИЙ. СЛУ_ЗАРП	СЛУЖАЩИЙ. ПРО_НОМ	ПРОЕКТ. ПРО_НОМ	ПРОЕКТ. ПРОЕКТ_ РУК
2934	Иванов	22400.00	1	1	Иванов
2941	Иваненко	22000.00	2	2	Иваненко

Кванторы

- При построении *WFF* допускается использование кванторов существования (*EXISTS*) и всеобщности (*FORALL*).
- Формула *EXISTS var* принимает значение *true* только в том случае, если в области определения переменной *var* найдется хотя бы один кортеж, для которого *WFF* принимает значение *true*.
- Формула *FORALL var* принимает значение *true*, если для всех значений переменной *var* из ее области определения *WFF* принимает значение *true*.

Кванторы

Пусть СЛУ1 и СЛУ2 представляют собой две кортежные переменные, определенные на отношении СЛУЖАЩИЕ. Тогда WFF

EXISTS СЛУ2 (СЛУ1.СЛУ_ЗАРП > СЛУ2.СЛУ_ЗАРП)

принимает значение **true** только для тех значений кортежной переменной **СЛУ1**, которые соответствуют служащим, не получающим минимальную зарплату.

**FORALL СЛУ2 (СЛУ1.СЛУ_ЗАРП >= СЛУ2.
СЛУ_ЗАРП)**

принимает значение **true** только для тех значений кортежной переменной **СЛУ1**, которые соответствуют служащим, получающим максимальную зарплату.

Кванторы

(a) Область истинности WFF

EXISTS СЛУ2 (СЛУ1.СЛУ_ЗАРП > СЛУ2.СЛУ_ЗАРП)

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2935	Петров	29600.00	1
2937	Федоров	20000.00	1
2938	Иванова	22000.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	2
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	2

(b) Область истинности WFF

FORALL СЛУ2 (СЛУ1.СЛУ_ЗАРП ≥ СЛУ2.СЛУ_ЗАРП)

СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2935	Петров	29600.00	1
2935	Петров	29600.00	2

Реляционное исчисление

СЛУЖАЩИЕ			
СЛУ_НОМЕР	СЛУ_ИМЯ	СЛУ_ЗАРП	ПРО_НОМ
2934	Иванов	22400.00	1
2935	Петров	29600.00	1
2936	Сидоров	18000.00	1
2937	Федоров	20000.00	1
2938	Иванова	22000.00	1
2934	Иванов	22400.00	2
2935	Петров	29600.00	2
2939	Сидоренко	18000.00	2
2940	Федоренко	20000.00	2
2941	Иваненко	22000.00	2

ПРОЕКТЫ	
ПРО_НОМ	ПРОЕКТ_РУК
1	Иванов
2	Иваненко

НОМЕРА_ПРОЕКТОВ

ПРО_НОМ
1
2

Выдать имена и номера служащих, которые являются руководителями проектов со средней заработной платой, превышающей 18000.

Запрос на языке реляционной алгебры

**(СЛУЖАЩИЕ JOIN ПРОЕКТЫ WHERE (СЛУ_ИМЯ = ПРОЕКТ_РУК AND ПРО_ЗАРП > 18000.00))
ПРОЕКТ (СЛУ_ИМЯ, СЛУ_НОМ)**

- выполнить *эквисоединение* отношений СЛУЖАЩИЕ и ПРОЕКТЫ по условию СЛУ_ИМЯ = ПРОЕКТ_РУК ;
- ограничить полученное отношение по условию ПРО_ЗАРП > 18000.00 ;
- спроецировать результат предыдущей операции на атрибут СЛУ_ИМЯ, СЛУ_НОМ.

Запрос на языке реляционного исчисления

Переменные:

RANGE СЛУЖАЩИЙ IS СЛУЖАЩИЕ

RANGE ПРОЕКТ IS ПРОЕКТЫ

Выражение:

СЛУЖАЩИЙ.СЛУ_ИМЯ, СЛУЖАЩИЙ.СЛУ_НОМ
WHERE EXISTS (СЛУЖАЩИЙ.СЛУ_ИМЯ =
ПРОЕКТ.ПРОЕКТ_РУК AND ПРОЕКТ.ПРО_ЗАРП >
18000.00).

Исчисление доменов

В *исчислении* доменов областью определения переменных являются не отношения, а домены.

Применительно к базе данных СЛУЖАЩИЕ-ПРОЕКТЫ можно говорить о доменных переменных ИМЯ (значения – допустимые имена) или НОСЛУ (значения – допустимые номера служащих).

Отличием исчисления доменов от исчисления кортежей является наличие дополнительного множества предикатов, позволяющих выражать условия членства.

Если R – это n -арное отношение с атрибутами $a_1, a_2, \dots, a_n$, то условие членства имеет вид $R(a_{i_1} : v_{i_1}, a_{i_2} : v_{i_2}, \dots, a_{i_m} : v_{i_m})$ ($m \leq n$), где v_{ij} – это литерально задаваемая константа либо имя доменной переменной.

Исчисление доменов

СЛУЖАЩИЕ (СЛУ_НОМ:2934, СЛУ_ИМЯ:'Иванов',
СЛУ_ЗАРП:22400.00, ПРО_НОМ:1)

Областью истинности является значение доменной переменной:

<2934, 'Иванов', 22400.00, 1>.

СЛУЖАЩИЕ (СЛУ_НОМ:2934, СЛУ_ИМЯ:'Иванов',
СЛУ_ЗАРП:22400.00, ПРО_НОМ:ПРО_НОМ)

Областью истинности являются два набора значений доменных переменных:

<2934, 'Иванов', 22400.00, 1> и

<2934, 'Иванов', 22400.00, 2>.

Заключение

Знание реляционной алгебры и реляционного исчисления на практике позволяет строить оптимальные и быстрые запросы к реляционным БД.

Язык доступа к данным реляционных БД называется реляционно полным, если он по выразительной силе не уступает реляционной алгебре или реляционному исчислению. Именно таким и является язык SQL.

Задание для СРС

Перечисленные ниже таблицы образуют часть реляционной БД.

Гостиницы (ГостиницаНомер, ГостиницаНазвание, Город)

Комнаты (КомнатаНомер, ГостиницаНомер, Тип, Цена)

Бронь (ГостиницаНомер, ГостьНомер, ДатаПриезда, ДатаОтъезда, КомнатаНомер)

Гость (ГостьНомер, ГостьИмя, ГостьАдрес)

Сформируйте выражения реляционной алгебры , а также реляционного исчисления кортежей и доменов для следующих запросов:

Составить список всех отдельных номеров стоимостью меньше 3000 тенге в сутки;

Составить список всех постояльцев с указанием фамилий и городов, откуда они прибыли;

Составить список всех номеров в гостинице «Казахстан» с указанием стоимости и типа;

Составить список всех постояльцев, которые в настоящее время проживают в гостинице «Казахстан».

Рекомендуемая литература

1. Карпова Т.С. Базы данных: модели, разработка. - СПб.: Питер, 2011, 304 с.
2. Кренке Д. Теория и практика построения баз данных: [пер.с англ] / Д. Кренке. - 9 - е изд. - СПб.: Питер, 2010. - 858 с.
3. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. проф. А.Д. Хомоненко. - СПб.: КОРОНА принт, 2008. - 416с.
4. Мейер Д. Теория реляционных баз данных. - М.: Мир, 2004. - 608с.
5. Хансен Г., Хансен Д. Базы данных: разработка и управление. - М.: БИНОМ, 2006.
6. Гусева Т.И., Башин Ю.Б. Проектирование баз данных в примерах и задачах. - М.: Радио и связь, 2008. - 160 с., ил.
7. Дейт К. Введение в системы баз данных: Пер. с англ. - М.: Наука, 2010. - 464 с.
8. Кириллов В.В. Основы проектирования баз данных. - М.: Финансы и статистика, 2007. - 204 с.