

# Базы данных

Гаврилова Юлия Михайловна

2019

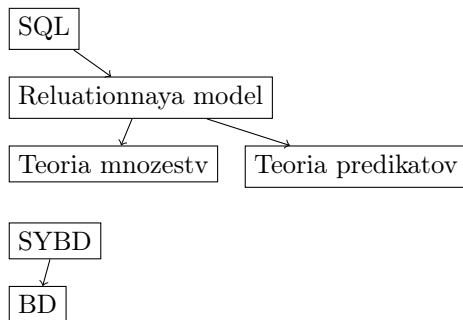
# Оглавление

<b>1</b>	<b>Введение</b>	<b>2</b>
1.1	Реляционная модель . . . . .	2
1.1.1	Структурная часть . . . . .	2
1.1.2	Целостная часть . . . . .	3
1.1.3	Манипуляционная часть . . . . .	3
1.2	Реляционная алгебра . . . . .	4
1.2.1	GROUP . . . . .	6
1.2.2	Summarize . . . . .	6
1.2.3	UNGROUP . . . . .	6
1.3	Реляционное сравнение . . . . .	6
1.3.1	Агрегатные сравнения . . . . .	7
1.4	Исчисление доменов . . . . .	7
<b>2</b>	<b>Теория проектирования реляционных баз данных</b>	<b>10</b>

# Глава 1

## Введение

Способы организации	
OLAP (online analytic processor)	OLTP (jnlne transaction processor)
Время отклика	Быстрая вставка
3NF	1NF
Нормальная форма	Для сбора статистики



### 1.1 Реляционная модель

1. Структурная часть: как построена модель
2. Целостная часть: какие ограничения, как должны быть организованы данные
3. Манипуляционная: обработка данных

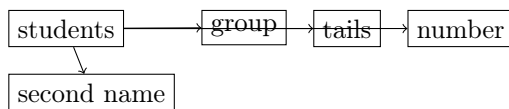
#### 1.1.1 Структурная часть

- Тип int, char
- домен - надстройка над типом, набор ограничений/правил (положительные четные для int), можно объявить над типом или над доменом
- атрибут - упорядоченная пара ( $\langle \text{имя}, \text{тип или домен} \rangle$ )
- заголовок (схема) отношения - множество всех пар атрибутов  $\{ \langle \text{имя атрибута}_1, \text{значение}_1 \rangle, \dots, \langle \text{имя атрибута}_N, \text{значение}_N \rangle \}$   
 $\{ \langle a_1, \text{int} \rangle, \langle a_2, \text{float} \rangle, \langle a_3, \text{char} \rangle, \langle a_4, \text{varchar} \rangle \}$
- кортеж над схемой  
 $\{ \langle a_1, 1 \rangle, \langle a_2, 1.4 \rangle, \langle a_3, 'a' \rangle, \langle a_4, 'aaa' \rangle \}$
- отношение

$a_1$	$a_2$	$a_3$	$a_4$
1	1.4	'a'	'aaa'

## ER-модель

- отношение/сущность



Здесь студент сущность сильная. Если студент зависит, то студент - слабая сущность

- связь 1 - 1 (Студент → зачетка)
- связь 1 ко многим (Студенты → группа)
- многие ко многим (Студенты → курс)

### Лабораторная работа 1

- Подобрать предметную область на весь семестр
- ER модель (не менее 3х самостоятельных сущностей)
- Создать свою БД (не менее 1000 записей на таблицу)

#### Защита:

- Добавить связь/атрибут
- Создать ссылку

### 1.1.2 Целостная часть

- целостность сущностей/отношений
- целостность ссылок

id	ФИО	Age
1	Иванов	10
2	Петров	15
3	Иванов	45

Потенциальный ключ:

- однозначная идентификация записи
- никаких подмножеств не должно быть под ключом

id	ФИО	id группы
1	Петров	1

↓ Внешняя ссылка

id	Название
1	ИУ7-53

Ссылочная целостность - нельзя сослаться на несуществующий объект

### 1.1.3 Манипуляционная часть

- Реляционная алгебра
- Реляционные исчисления

## 1.2 Реляционная алгебра

id	name
1	a
2	b

id	name
2	b
3	c

### 1. Традиционные - работа с множеством

- Объединение (UNION)

id	name
1	a
2	b
3	c

- Пересечение (INTERSECT)

id	name
2	b

- Вычитание (MINUS)

id	name
1	a

id	name
3	c

- Декартово произведение (TIMES) - все возможные комбинации атрибутов

### 2. Специальные

- Соединение (JOIN)

id	name1	name2
2	b	b

- Ограничение (WHERE)
- Проекция (PROJECT)
- Деление (DIVIDE BY)

Реляционное выражение = унарное выражение (бинарное выражение)

### Унарные выражения

- Проекция  
терм | терм[список атрибутов]
  - Ограничение  
терм WHERE логическое\_выражение
  - Переименование  
терм RENAME old\_name TO new\_name
- терм - имя\_отношения | (реляционное\_выражение)

### Бинарные выражения

- Объединение
- Пересечение
- Вычитание
- Декартово произведение

- Соединение

бинарные операции = проекция бинарная\_операция реляционное\_выявление

S JOIN P[P..,S..]

Поставщик
S

↓ Многие ко многим SP

Детали
P

$S(Sno : integer, Sname : string, Status : integer, City : string)$

$P(Pno : integer, Pname : string, Color : string, Weight : real, City : string)$

$SP(Sno : integer, Pno : integer, Quantity : integer)$

S

Sno	Sname	Status	City
1	Алмаз	20	Смоленск
2	Дельта	10	Владимир
3	Орион	30	Смоленск

P

Pno	Pname	Color	Weight	City
1	Гайка	К	12.0	Смоленск
2	Болт	С	17.1	Рязань
3	Винт	З	15.47	Владимир
4	Винт	К	18	Москва
5	Шайба	З	25	Смоленск

SP

Sno	Pno	Quantity
1	1	25
1	2	14
2	4	2

1. Имена всех поставщиков детали под номером 2

$$\underbrace{\underbrace{((SP \text{ join } S) \text{ where } Pno = 2)}_{\text{рел. выр.}}}_{\text{реляционное выражение}} \underbrace{[Sname]}_{\text{лог. выр.}}$$

select Sname

from SP inner join S on SP.Sno = S.Sno

where SP.Pno = 2

2. Вывести все имена поставщиков, которые поставляют как минимум одну красную деталь

$$(((P \text{ where } Color = 'K') \text{ join } SP) \text{ join } S)[Sname]$$

3. Получить имена поставщиков, которые поставляют все детали

$A(X_1, \dots, X_n, Y_1, \dots, Y_n)$

$B(Y_1, \dots, Y_n)$

$A \text{ divide by } B = (X_1, \dots, X_n)$

Sno	Pno
1	1
1	2
1	3
2	2
2	3
3	1

$P[Pno]$

$SP$  divide by  $P[Pno]$

$((SP \text{ divide by } P[Pno]) \text{ join } S)[Sname]$

4. Все поставщики, которые поставляют только красные детали

$(SP \text{ divide by } (P \text{ where Color} = 'K')[Pno])[Sname]$

5. Переименовать города из первой таблицы во вторые

$(S \text{ rename } Sno \text{ to } firstName)[firstName, City] \text{ join}$

$(S \text{ rename } Sno \text{ to } secondName)[secondName, City]) \text{ where } secondName > firstName \text{ join } S$

firstName	C
1	C
2	B
3	C

secondName	C
1	C
2	B
3	C

firstName	secondName	C
1	1	C
1	3	C
2	2	B
3	1	C
3	3	C

6. Поставщики, которые не поставляют деталь номер 2

$((S[Sno] \text{ minus } (SP \text{ where } Pno = 2)[Sno]) \text{ join } S)[Sname]$

### 1.2.1 GROUP

SP group (Pno, Qty) as PQ - группирует

Sno	Pno	Qty
1	1	10
1	2	15
2	1	5

↓

Sno	PQ
1	1-10 2-15
2	1-5

### 1.2.2 Summarize

summarize SP per SP {Pno} add sum(Qty) as sQty

Pno	sQty
1	15
1	16

extend S add 'Поставщик' as Sname2

extend SP add Qty\*100 as Qty2

### 1.2.3 UNGROUP

## 1.3 Реляционное сравнение

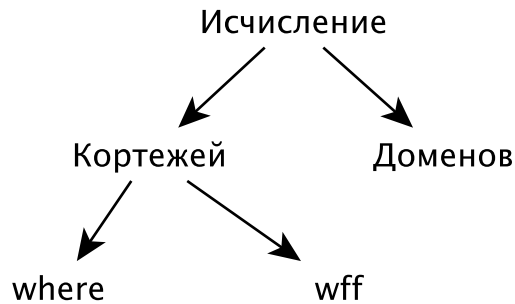
$S(Sno) = SP(Sno)$

is\_empty(реляционное выражение)

$t \text{ in } R \Leftrightarrow \text{RELATION}\{t\} \leq R$

t - Кортеж

R - отношение



объявление = range of переменная is список  
 область = отношение | реляционное выражение  
 реляционное выражение = (список целевых элементов) [where(wff)]  
 целевой элемент = переменная | переменная атрибут [as имя]  
 wff = условие | not условие | условие and (or) wff | if условие then(wff)

### Примеры

range of SX is S  
 range of SPX is SP  
 range of SY is (SX) where SX.City = 'Смоленск', (SX) where exists SPX(SPX.Sno=SX.Sno and SPX.Pno=1)

Задачи как в реляционной алгебре

1. range of SX is S (SPX.Sno) where SPX.Pno = 2 (SX.Sname) where exists SPX(SPX.Sno = SX.Sno and SPX.Pno = 2)
2. range of SX os P (SX.Sname) where exists SPX(SPX.Sno = SX.Sno and exists PX(SPX.Pno = PS.Pno and PX.Color = 'K'))  
 (SX.Sname) where exists SPX(where exists PX(SPX.Sno = SX.Sno and SPX.Pno = SX.Pno and PX.Color = 'K'))  
 range of PX is (Pno) where P.Color = 'K'
3. (SX.Sname) where forall PX(exists SPX(SPX.Pno = SX.Pno and SPX.Sno = SX.Sno))
4.  $\langle S_1, S_2 \rangle$   
 range of SY is S (SX.Sname as FirstName, SY.Sname as SecondName) SX.City = SY.City and SX.Sno > SY.Sno
5. (SX.Sname) where not exists SPX(SPX.Sno = SX.Sno and SPX.Pno = 2)

### 1.3.1 Агрегатные сравнения

(sum(SPX.Qty) as Total)  
 агрегатная функция((атрибуты) where f[атрибуты])

## 1.4 Исчисление доменов

R(pair, pair...) - условие принадлежности в общем виде  
 R - имя отношения, pair = A:v, где A - атрибут отношения R, v - или переменная домена, или литерал

SP(Sno : 1, Pno:1) - истина если есть кортеж с Sno = 1 и Pno = 1

SP(Sno: SX, Pno: PX) - только если в отношении SP есть кортеж

1. (SX) - множество всех номеров поставщиков
2. (SX) where S(Sno : SX) - множество всех номеров поставщиков в отношении S
3. (SX) where S(Sno : SX, City : 'Смоленск') - подмножество номеров поставщиков из города Смоленска



4. (SX, CityX) where S(Sno : SX, City : CityX) and SP(Sno : SX, Pno : 2) - запрос на получение номера поставщиков поставляющих деталь под номером 2
5. (SX, PX) where S(Sno : Sx, City : CityX) and P(Pno : PX, City : CityY) and CityX <> CityY - получение пар номер поставщика и детали где поставщики и детали находятся не в одном городе
1. Получить номера поставщиков из Смоленска у которых статус больше 20  
SX where exists StatusX (StatusX > 20 and S(Sno : SX, Status : StatusX, City : 'Смоленск'))
2. Получить все пары поставщиков, что два поставщика размещаются в одном городе  
(SX as FirstSno, SY as SecondSno) where exists CityZ(S(Sno : SX, City : CityZ) and S(Sno : SY, City : CityZ) and SX < SY)
3. Получить имена поставщиков которые поставляют как минимум одну красную деталь  
NameX where exists SX exists PX (S(Sno : SY, Sname : NameX) and SP(Sno : SX, Pno : PX) and P(Pno : PX, Color = 'Красный'))
4. Получить имена поставщиков которые поставляют хотя бы одну деталь поставляемую поставщиком под номером 2  
NameX where exists SX exists PX (S(Sno : SX, Sname : NameX) and SP(Sno : SX, Pno : PX) and SP(Sno : 2, Pno : PX))
5. Получить имена поставщиков которые поставляют все типы деталей  
NameX where exists SX(S(Sno : SXm Sname : NameX) and forall PX(if P(Pno : PX) then (Sno : SX, Pno : PX)))
6. Получить имена поставщиков которые не поставляют деталь с номером 2  
NameX where exists SX(S(Sno : SX, Sname : NameX) and not SP(Sno : SX, Pno : 2))
7. Получить номера поставщиков которые поставляют как минимум все типы деталей поставляемыми поставщиком с номером 2  
SX where forall PX(if SP(Sno : 2, Pno : PX) then SP(Sno : SX, Pno : PX))
8. Получить номера деталей которые не весят больше 16 фунтов или поставляются с поставщиком под номером 2, или и то и другое  
PX where exists WeightX(P(Pno : PX, WeightX) and WeightX > 16) or SP(Sno : 2, Pno : PX))

#### Поставщики (S)

Sno	Sname	Status	City
1	Алмаз	20	Смоленск
2	Циклон	10	Владивосток
⋮			

#### Детали (P)

Pno	Pname	Color	Weight	City
1	Гайка	Красный	12	Смоленск
2	Болт	Зеленый	17	Владимир
⋮				

#### Проекты (J)

Jno	Jname	City
1	Ангара	Владимир
2	Алтай	Рязань
⋮		

#### Поставки (SPJ)

Sno	Pno	Jno	Qty
1	1	1	200
1	1	4	700
⋮			

1. (SX,Name, SX.City) where exists JX forall PX exists PSJX (JX.City = 'Ярославль' and JX.Jno = SPJX.Jno and PX.Pno = SPJX.Pno and SX.Sno = SPJX and SPJX.Qty >= 50)

SX : все кортежи отношения S (5 шт)

PX : все кортежи отношения P (6 шт)

JX : все кортежи отношения J, в которых City = 'Ярославль' (2шт)

SPJX : все кортежи отношения SPJ, d которых Qty >= 50 (24 шт)

2. JX.JN = SPJX.Jno and PX.Pno = SPJX.Ono and SX.Xno = SPJX.Sno
3. exists RX, forall RX
4. exists JX forall PX exists SPJX

1. exists SPJX исключая SPJ (SPJ.Sno, SPJ.Pno, SPJ.Jno и SPJ.Qty)

Sno	Sname	City	Pno	Pname	Color	weight	City	Jno	Jname	City

2. forall PX Делим на P

Sno	Sname	Status	City	Jno	Jname	City

3. exists JX исключаем J(J.Jno, J.Jname, J.City)

Sno	Sname	Status	City

5. SX.Sname, SX.City

Sname	City

## Глава 2

# Теория проектирования реляционных баз данных

Есть две проблемы: как повысить эффективность и как представить реальные объекты.

Классический подход это выделение решений и их реализация. Нормальные формы. каждая НФ - набор ограничений. Каждая следующая НФ лучше предыдущей.

Следующие нормальные формы:

1. 1 NF
2. 2 NF
3. 3 NF
4. 4 NF
5. PSNF - форма проекций соединения

### Свойства НФ

1. Каждая следующая НФ лучше предыдущей
2. При переходе к следующей НФ, свойства предыдущей сохраняются