

РЕЛЯЦИОННЫЕ ОПЕРАТОРЫ И ЯЗЫК SQL

Основные определения

- Реляционная модель - множество взаимосвязанных отношений.
- В каждой связи одно отношение может выступать как **основное**, а другое отношение выступает в роли **подчиненного**.
- 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. Данный набор атрибутов в подчиненном отношении принято называть **внешним ключом (FOREIGN KEY)**.

ОГРАНИЧЕНИЯ ЦЕЛОСТНОСТИ ДАННЫХ

База данных поддерживает следующие виды ограничений:

- **1) PRIMARY KEY**
- **2) UNIQUE**
- **3) FOREIGN KEY**
- **4) CHECK,**
- **5) NOT NULL**

Основные определения



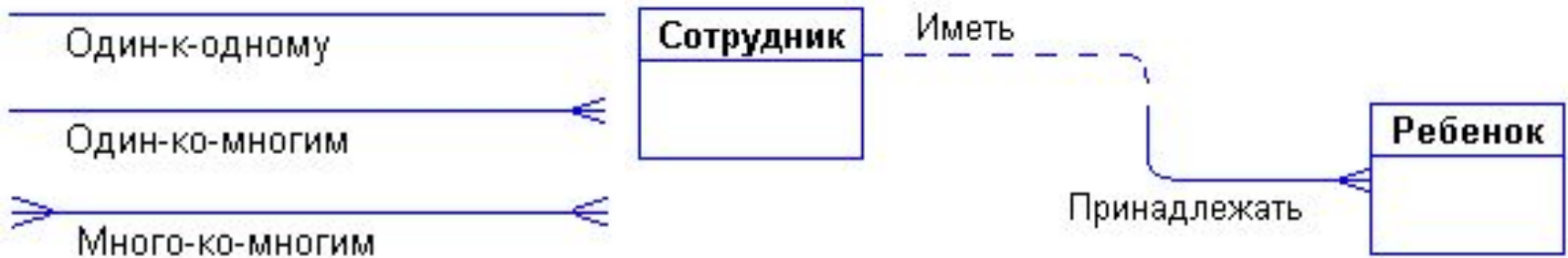
Атрибут **Паспорт** PRIMARY KEY отношения «Сотрудник»
и FOREIGN KEY для отношения «Карьера».

Типы связей

5

Одна сущность может быть связана с другой сущностью или сама с собою.

Левая сущность (со стороны "один") называется *родительской*, правая (со стороны "много") - *дочерней*.



Реляционная

6 алгебра

В университете учатся студенты Иванов, Петров и Сидоров. Лекции им читают преподаватели Пушников, Цыганов и Шарипов, причем известны следующие факты:

- Пушников читает лекции по алгебре и базам данных, соответственно, 40 и 80 часов в семестр.
- Цыганов читает лекции по геометрии, 50 часов в семестр.
- Шарипов читает лекции по алгебре и геометрии, соответственно, 40 и 50 часов в семестр.
- Студент Иванов посещает лекции по алгебре у Шарипова и по базам данных у Пушникова.
- Студент Петров посещает лекции по алгебре у Пушникова и по геометрии у Цыганова.
- Студент Сидоров посещает лекции по геометрии у Цыганова и по базам данных у Пушникова.

Реляционная алгебра

- Множество преподавателей

$A = \{\text{Пушников, Цыганов, Шарипов}\}.$

- Множество предметов

$B = \{\text{Алгебра, Геометрия, Базы данных}\}.$

- Множество студентов

$C = \{\text{Иванов, Петров, Сидоров}\}.$

Упорядоченная тройка

$(x, y, n) \in R_1$ а и только

тогда принадлежит отношению , когда

преподаватель x читает лекции по предмету y в количестве n часов в семестр.

Реляционная

алгебра

A (Преподаватель)	B (Предмет)	Q (Количество часов)
<i>Пушников</i>	<i>Алгебра</i>	<i>40</i>
<i>Пушников</i>	<i>Базы данных</i>	<i>80</i>
<i>Цыганов</i>	<i>Геометрия</i>	<i>50</i>
<i>Шарипов</i>	<i>Алгебра</i>	<i>40</i>
<i>Шарипов</i>	<i>Геометрия</i>	<i>50</i>

Реляционная алгебра

9

Упорядоченная тройка $(z, y, x) \in R_2$, когда студент z посещает лекции по предмету y у преподавателя x .

С (студент)	В (предмет)	А (Преподаватель)
Иванов	Алгебра	Шарипов
Иванов	Базы данных	Пушников
Петров	Алгебра	Пушников
Петров	Геометрия	Цыганов
Сидоров	Геометрия	Цыганов
Сидоров	Базы данных	Пушников

Реляционная алгебра

10

Каждому отношению можно поставить в соответствие некоторое логическое выражение $P(x_1, x_2, \dots, x_n)$, зависящее от n параметров (n -местный предикат) и определяющее, будет ли кортеж (a_1, a_2, \dots, a_n) принадлежать отношению R .

Это логическое выражение называют предикатом отношения R .

Отношение R на декартовом произведении двух множеств $A_1 \times A_2$ называется **функциональным отношением**, если оно обладает следующим свойством:

Если $(x, y) \in R$ и $(x, z) \in R$, то $y = z$ (однозначность функции).

Потенциальные ключи

- Каждый кортеж должен обладать свойством уникальности (свойством уникальности в пределах отношения могут обладать отдельные атрибуты или группы атрибутов- **потенциальные ключи**).
- Подмножество атрибутов K отношения R будем называть **потенциальным ключом**, если выполнено:
- **Свойство уникальности** - в отношении не может быть двух различных кортежей, с одинаковым значением K .
- **Свойство избыточности** - никакое подмножество в K не обладает свойством уникальности.

Потенциальные

12

КЛЮЧИ

- отношение имеет, по крайней мере, один потенциальный ключ.
- если никакой атрибут или группа атрибутов не являются потенциальным ключом, то, в силу уникальности кортежей, все атрибуты вместе образуют потенциальный ключ.
- потенциальный ключ, состоящий из одного атрибута, называется **простым**, а из нескольких - **составным**.
- отношение может иметь несколько потенциальных ключей: один из потенциальных ключей объявляется **первичным**, а остальные - **альтернативными**.

Восемь реляционных операторов

13

- **Теоретико-множественные операторы:**
 - Декартово произведение
 - Объединение
 - Пересечение
 - Вычитание
- **Специальные реляционные операторы:**
 - Выборка
 - Проекция
 - Соединение
 - Деление

Реляционная

14

алгебра

- Отношение состоит из двух частей - **заголовка отношения и тела отношения.**
- Количество атрибутов называется **степенью отношения.**
- Тело отношения состоит из кортежей.
- Количество кортежей отношения называется **мощностью отношения.**

Декартово произведение

15

Основной структурой данных в модели является **отношение**, именно поэтому модель получила название **реляционной** (от английского **relation** — отношение).
Отношение в реляционной базе данных – подмножество прямого (декартова) произведения множества атрибутов.

N-арным отношением R называют подмножество декартова произведения $D_1 \times D_2 \times \dots \times D_n$ множеств D_1, D_2, \dots, D_n ($n > 1$), необязательно различных.

Исходные множества D_1, D_2, \dots, D_n называют в модели **доменами**.

$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

Декартово произведение

16

Введем дополнительно понятие **конкатенации**, или **сцепления**, кортежей.

Сцеплением, или **конкатенацией**, кортежей

$$c = \langle c_1, c_2, \dots, c_n \rangle \text{ и}$$

$$q = \langle q_1, q_2, \dots, q_m \rangle$$

называется *кортеж*, полученный добавлением значений второго в конец первого.

Сцепление кортежей c и q обозначается как (c, q) .

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

Здесь n — число элементов в первом кортеже c , m — число элементов во втором кортеже q .

Декартово произведение

17

- **Определение** Декартовым произведением двух отношений $A(A_1, A_2, \dots, A_N)$ и $B(B_1, B_2, \dots, B_M)$ называется отношение, заголовок которого является сцеплением заголовков отношений A и B :
$$(A_1, A_2, \dots, A_N, B_1, B_2, \dots, B_M),$$
- а тело состоит из кортежей, являющихся сцеплением кортежей отношений A и B :
- $(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m),$
- таких, что $(a_1, a_2, \dots, a_n) \in A, (b_1, b_2, \dots, b_m) \in B.$
- Синтаксис операции декартового произведения:
$$A \text{ TIMES } B$$

Декартово произведение

18

Отношение А

Номер поставщика	Наименование поставщика
1	Иванов
2	Петров
3	Сидоров

Отношение В

Номер детали	Наименование детали
1	Болт
2	Гайка
3	Винт

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

Отношения, совместимые по типу

19

Отношения **совместимы по типу**, если они имеют идентичные заголовки, а именно:

- Отношения имеют одно и то же множество имен атрибутов, т.е. для любого атрибута в одном отношении найдется атрибут с таким же наименованием в другом отношении,
- Атрибуты с одинаковыми именами определены на одних и тех же доменах.

Объединение отношений

20

- 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 кортежей, принадлежащих A или B , или обоим отношениям.
- Синтаксис операции объединения: $A \text{ UNION } B$

$$R_1 \cup R_2 = \{r | r \in R_1 \vee r \in R_2\}$$

Объединение отношений

21

Отношение А

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

Отношение В

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

Отношение А UNION В

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

Пересечение отношений

22

- 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:
 $A \text{ INTERSECT } B$

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

Пересечение отношений

23

Отношение А

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

Отношение В

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

Отношение А INTERSECT В

Табельный номер	Фамилия	Зарплата
1	Иванов	1000

Вычитание отношений

24

- 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:

$A \text{ MINUS } B$

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

Вычитание отношений

25

Отношение А

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

Отношение В

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

Отношение А MINUS В

Табельный номер	Фамилия	Зарплата
2	Петров	2000
3	Сидоров	3000

Выборка

- **Выборкой** на отношении **A** с условием **C** называется отношение с тем же заголовком, что и у отношения **A**, и телом, состоящим из кортежей, значения атрибутов которых при подстановке в условие **C** дают значение «Истина».
- **C** представляет логическое выражение, в которое могут входить атрибуты отношения **A** и(или) скалярные выражения.

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

Выборка

27

- В простейшем случае условие s имеет вид $X\theta Y$, где θ - один из операторов сравнения ($=, >, <$ и т.д.), 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 s , или A WHERE $X\theta Y$

$$\theta = \{=, <>, \geq, \leq, <, >\}$$

Выборка

28

Отношение A

<i>Табельный номер</i>	<i>Фамилия</i>	<i>Зарплата</i>
1	Иванов	1000
2	Петров	2000
3	Сидоров	3000

Отношение A WHERE Зарплата < 3000

<i>Табельный номер</i>	<i>Фамилия</i>	<i>Зарплата</i>
1	Иванов	1000
2	Петров	2000

Трехзначная логика (3VL)

29

AND	F	T	U
F	F	F	F
T	F	T	U
U	F	U	U

Таблица 1 Таблица истинности **AND**

OR	F	T	U
F	F	T	U
T	T	T	T
U	U	T	U

Таблица 2 Таблица истинности **OR**

NOT	
F	T
T	F
U	U

Таблица 3 Таблица истинности **NOT**

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 кортежи со значением атрибута X равным x , значением атрибута Y равным y, \dots , значением Z равным z .
- Проекция дает «вертикальный срез», в котором удалены все возникшие при таком срезе дубликаты кортежей.
 - Синтаксис операции проекции:
 $A[X, Y, \dots, Z]$

Проекция

Отношение A (Поставщики)

<i>Номер поставщика</i>	<i>Наименование поставщика</i>	<i>Город поставщика</i>
1	Иванов	Уфа
2	Петров	Москва
3	Сидоров	Москва
4	Сидоров	Челябинск

Отношение A[Город поставщика]

Город поставщика
Уфа
Москва
Челябинск

Соединение

Соединением отношений **A** и **B** по условию **C** (логическое выражение, в которое могут входить атрибуты отношений и (или) скалярные выражения) называется отношение, полученное путем последовательного применения операций декартова произведения и выборки.

(A TIMES B) WHERE c

Соединение

33

- *Тэта-соединение*
- *Эквисоединение*
- *Естественное соединение*
- *Внешнее соединение*

Тэта-соединение

34

- *Определение.* 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 TIMES B) WHERE X θ Y

- Это частный случай операции общего соединения.

$$\theta = \{=, <, >, \geq, \leq, <, >\}$$

Экви-соединение

35

(A TIMES B) WHERE X=Y

Естественное соединение

36

- *Определение.* Пусть даны отношения $A(A_1, A_2, \dots, A_N, X_1, \dots, X_P)$ и $B(X_1, \dots, X_P, B_1, B_2, \dots, B_M)$, имеющие одинаковые атрибуты X_1, \dots, X_P (т.е. атрибуты с одинаковыми именами и определенные на одинаковых доменах).
- Тогда **естественным соединением** отношений A и B называется отношение с заголовком $(A_1, A_2, \dots, A_N, X_1, \dots, X_P, B_1, B_2, \dots, B_M)$ и телом, содержащим множество кортежей $(a_1, a_2, \dots, a_N, x_1, \dots, x_P, b_1, b_2, \dots, b_M)$, таких, что $(a_1, a_2, \dots, a_N, x_1, \dots, x_P) \in A$ и $(a_1, a_2, \dots, a_N, x_1, \dots, x_P, b_1, b_2, \dots, b_M) \in B$.
- Естественное соединение настолько важно, что для него используют специальный синтаксис:

A JOIN B

Естественное соединение

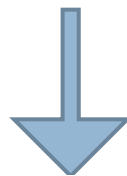
37

СТУДЕНТ

Кст	Фио	Нгр
105	Петров	3
318	Иванов	2
216	Сидоров	2
418	Алексеев	3

ГРУППА

Нгр	Спец	Кст
2	Физика	216
1	Химия	318
3	Матем	105



СТУДЕНТ .Кст	Фио	СТУДЕНТ.Нгр	ГРУППА.Нгр	Спец	ГРУППА Кст
105	Петров	3		Матем	
216	Сидоров	2		Физика	

Внешнее соединение

38

Типы внешних соединений:

- ▣ **левое внешнее соединение** - внутреннее соединение + строки левой таблицы, которым нет соответствия в правой таблице;
- ▣ **правое внешнее соединение** - внутреннее соединение + строки правой таблицы, которым нет соответствия в левой таблице;
- ▣ **полное внешнее соединение** - внутреннее соединение и строки правой таблицы, которым нет соответствия в левой таблице, и строки правой таблицы, которым нет соответствия в левой таблице.

Внешнее соединение

39

Таблица отделов **DEP** - с таким содержимым:

DEP_ID	NAME
12	SALES
13	OPERATIONS
14	RESEACH

Таблица менеджеров отделов - **MNG** - с таким содержимым:

DEP ID	M NAME
12	JONES
13	SMITH
15	BROWN

1). Выбрать названия отделов и имена их менеджеров - **естественное (внутреннее) соединение**

```
name      m_name
-----  -
SALES     JONES
OPERATIONS SMITH
```

2). Выбрать названия отделов и имена их менеджеров, включая те отделы, в которых нет менеджеров, - **левое внешнее соединение**

```
name      m_name
-----  -
SALES     JONES
OPERATIONS SMITH
RESEACH   -
```

Внешнее соединение

40

Таблица отделов **DEP** - с таким содержимым:

DEP_ID	NAME
12	SALES
13	OPERATIONS
14	RESEACH

Таблица менеджеров отделов - **MNG** - с таким содержимым:

DEP_ID	M_NAME
12	JONES
13	SMITH
15	BROWN

3). Выбрать названия отделов и имена их менеджеров, включая тех менеджеров, которые не приписаны к отделам,- **правое внешнее соединение**

```
name      m_name
-----  -
SALES     JONES
OPERATIONS SMITH
-         BROWN
```

4). Выбрать названия отделов и имена их менеджеров, включая те отделы, в которых нет менеджеров, и тех менеджеров, которые не приписаны к отделам - **полное внешнее соединение**

```
name      m_name
-----  -
SALES     JONES
OPERATIONS SMITH
RESEACH   -
-         BROWN
```


Деление

41

- *Определение.* Пусть даны отношения $A(X_1, X_2, \dots, X_N, Y_1, Y_2, \dots, Y_M)$ и $B(Y_1, Y_2, \dots, Y_M)$, причем атрибуты Y_1, Y_2, \dots, Y_M - общие для двух отношений. **Делением** отношений A на B называется отношение с заголовком (X_1, X_2, \dots, X_N) и телом, содержащим множество кортежей (x_1, x_2, \dots, x_n) , таких, что для всех кортежей $(y_1, y_2, \dots, y_m) \in B$ в отношении A найдется кортеж $(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_m)$.
- Отношение A выступает в роли делимого, отношение B выступает в роли делителя. Деление отношений аналогично делению чисел с остатком.

Синтаксис операции деления:

A DIVIDE BY B

Деление

42

Проекция X=PD[PNUM,DNUM]

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

Проекция Y=D[DNUM]

Номер детали DNUM
1
2
3

Отношение X DEVIDEBY Y дает список номеров поставщиков,
поставляющих все детали

Номер поставщика PNUM
1

ЯЗЫК SQL

- **Structured Query Language (SQL)**— это не процедурный язык, используемый для формулировки запросов к данным в большинстве современных СУБД и являющийся индустриальным стандартом (ANSI, ISO)
- Существует много его диалектов
- Операторы языка делятся на группы

ОПЕРАТОРЫ ОПРЕДЕЛЕНИЯ ДАННЫХ (DATA DEFINITION LANGUAGE DDL):

- **CREATE** создает объект БД;
- **ALTER** изменяет объект;
- **DROP** удаляет объект;

ОПЕРАТОРЫ ОПРЕДЕЛЕНИЯ ДОСТУПА К ДАННЫМ (DATA CONTROL LANGUAGE DCL):

- **GRANT** предоставляет пользователю (группе) разрешения на определенные операции с объектом;
- **REVOKE** отзывает ранее выданные разрешения;

ОПЕРАТОРЫ УПРАВЛЕНИЯ ТРАНЗАКЦИЯМИ (TRANSACTION CONTROL LANGUAGE, TCL)

- ❑ **COMMIT** применяет транзакцию;
- ❑ **ROLLBACK** откатывает все изменения, сделанные в контексте текущей транзакции;
- ❑ **SAVEPOINT** делит транзакцию на более мелкие участки, применяется для отметки логических точек разрыва в границах транзакции;
- ❑ **SET TRANSACTION** начинает транзакцию и определяет ее поведение.

ОПЕРАТОРЫ МАНИПУЛЯЦИИ ДАННЫМИ (DATA MANIPULATION LANGUAGE DML):

- **SELECT** считывает данные, удовлетворяющие заданным условиям;
- **INSERT** добавляет новые данные;
- **UPDATE** изменяет существующие данные;
- **DELETE** удаляет данные;

Порядок выполнения оператора SELECT

48

```
SELECT (DISTINCT)
FROM список_таблиц ← - - - - - Начало внешнего
                               оператора SELECT
WHERE
{выражение { [NOT] IN | оператор_сравнения [ANY | ALL] }
  | [NOT] EXISTS}
{SELECT (DISTINCT) список_выбора_подзапроса } Подзапрос,
  FROM список_таблиц } заключенный в скобки
  WHERE условия)

[GROUP BY список_группировки
[HAVING условия]] ← - - - - - Необязательные части
[ORDER BY порядок] внешнего оператора SELECT
```


Порядок обработки элементов оператора SELECT

49

- **FROM** - Определяются имена используемой таблицы или нескольких таблиц.
- **WHERE** – накладывается условие отбора данных.
- **GROUP BY** – образуются группы строк, имеющие одинаковые значения в указанном столбце.
- **HAVING** – накладывается условие на отбор сгруппированных строк.
- **SELECT** – определяются столбцы, которые нужно отобразить в результате.
- **ORDER BY** – отобранные данные сортируются по указанным столбцам.

Порядок выполнения оператора SELECT

50

- Шаг 1 (**FROM**). Вычисляется прямое декартовое произведение всех таблиц, указанных в обязательном разделе FROM.
- Шаг 2 (**WHERE**). Для каждой строки из таблицы A вычисляется условное выражение, приведенное в разделе WHERE. Только те строки, для которых условное выражение возвращает значение TRUE, включаются в результат.

Порядок выполнения оператора SELECT

51

- Шаг 3 (**GROUP BY**). Строки таблицы, полученной на втором шаге, группируются в соответствии со списком группировки, приведенным в разделе **GROUP BY**. К группам можно применять функции агрегирования.
- Если раздел **GROUP BY** опущен, то сразу переходим к шагу 4.

Порядок выполнения оператора SELECT

52

- Шаг 4 (**HAVING**). Если в операторе SELECT присутствует раздел HAVING, то группы, не удовлетворяющие условному выражению, приведенному в разделе HAVING, исключаются. Если раздел HAVING опущен, то сразу переходим к шагу 5.

Порядок выполнения оператора SELECT

53

- Шаг 5(**ORDER BY**) . Упорядочение результатов запроса по нескольким полям с возрастанием или убыванием (**ORDER BY... ASC (DESC)**).

PNUM	DNUM	VOLUME
1	1	100
2	1	150
3	1	1000
1	2	200
2	2	250
1	3	300

- `SELECT PD.PNUM, PD.DNUM, PD.VOLUME FROM PD ORDER BY DNUM;`
- ASC** (с возрастанием) по умолчанию

Что такое подзапрос?

54

Подзапросы в зависимости от элементов в предложении **WHERE**:

- могут не возвращать ни одного или возвращать несколько элементов (**начинаются с IN, оператора сравнения, ANY, ALL**);
- возвращать единственное значение (**с оператора сравнения**);
- представлять собой тест на существование (**EXISTS**)

Предикаты, используемые в условии отбора с WHERE

55

Условие поиска	Ключевые слова, специальные символы
Сравнение	>,<,<=>(или !=)
Диапазон	BETWEEN/NOT BETWEEN
Принадлежность к множеству	IN/NOT IN
Соответствие шаблону	LIKE/NOT LIKE
Значение NULL	IS NULL/IS NOT NULL
	ANY/ALL
	EXISTS

Оператор LIKE

56

Образец строки	Обработанный критерий	Примеры возвращенных значений
'shaX'	Строка символов может быть любой длины, но должна начинаться с "sha".	Shannon, Sharon, Shawn
'Xson'	Строка символов может быть любой длины, но должна заканчиваться "son".	Benson, Johnson, Morrison
'XhanX'	Строка символов может быть любой длины, но должна содержать "han".	Buchanan, Handel, Johansen, Nathanson
'go_'	Строка символов может иметь длину только в три символа, а первая и вторая буквы должны быть "go".	Rob, Ron, Roy
'_im'	Строка символов может иметь длину только в три символа, а вторая и третья буквы должны быть "im".	Jim, Kim, Tim
'_ar_'	Строка символов может иметь длину только в четыре символа, а вторая и третья буквы должны быть "ar".	Bart, Gary, Mark
'_atX'	Строка символов может быть любой длины, но вторая и третья буквы должны быть "at".	Gates, Matthews, Patterson
'Xac_'	Строка символов может быть любой длины, но вторая и третья буквы от конца должны быть "ac".	Apodaca, Tracy, Wallace

SELECT site, url FROM wwwsites

WHERE url LIKE '%my@_works%' ESCAPE '@';

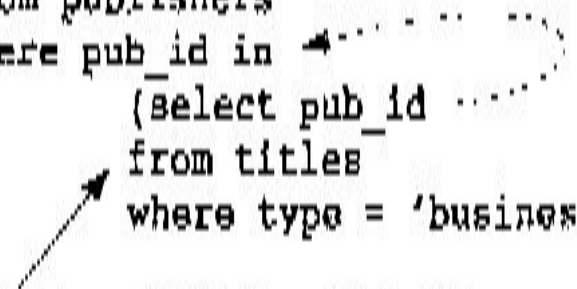
Пример подходящего шаблона — '12my_works'

Что такое подзапрос?

57

SQL. (некоррелированный):

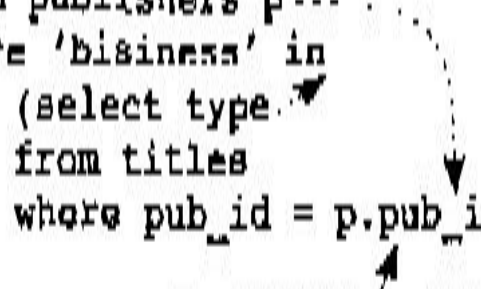
```
select pub_name  
from publishers  
where pub_id in  
    (select pub_id  
     from titles  
     where type = 'business')
```



Внутренний запрос выполняется независимо, передавая результаты во внешний запрос

SQL. (коррелированный):

```
select pub_name  
from publishers p  
where 'business' in  
    (select type  
     from titles  
     where pub_id = p.pub_id)
```



Внутренний запрос для своего выполнения должен получить данные из внешнего запроса

Результат:

Pub_name

New Age Books

Algodata Infosystems

Примеры выполнения SELECT

58

```
SELECT * FROM CUSTOMERS WHERE RATING > ALL(SELECT RATING  
FROM CUSTOMERS WHERE CITY='ROME' )
```

CNUM	CNAME	CITY	RATING	SNUM
2008	Cisneros	San Jose	300	1007
2004	Grass	Berlin	300	1002

Таблица 1.2: Заказчики (Customers)

CNUM	CNAME	CITY	RATING	SNUM
2001	Hoffman	London	100	1001
2002	Giovanni	Rome	200	1003
2003	Liu	SanJose	200	1002
2004	Grass	Berlin	300	1002
2006	Clemens	London	100	1001
2008	Cisneros	SanJose	300	1007
2007	Pereira	Rome	100	1004

Примеры выполнения SELECT

59

```
SELECT * FROM CUSTOMERS WHERE RATING > ANY(SELECT  
RATING FROM CUSTOMERS WHERE CITY='ROME' )
```

CNUM	CNAME	CITY	RATING	SNUM
2004	Grass	Berlin	300	1002
2008	Cisneros	San Jose	300	1007
2003	Liu	San Jose	200	1002
2002	Giovanni	Rome	200	1003

Таблица 1.2: Заказы (Customers)

CNUM	CNAME	CITY	RATING	SNUM
2001	Hoffman	London	100	1001
2002	Giovanni	Rome	200	1003
2003	Liu	SanJose	200	1002
2004	Grass	Berlin	300	1002
2006	Clemens	London	100	1001
2008	Cisneros	SanJose	300	1007
2007	Pereira	Rome	100	1004

Примеры выполнения SELECT

60

```
SELECT * FROM CUSTOMERS WHERE RATING > SOME(SELECT  
RATING FROM CUSTOMERS WHERE CITY='ROME' )
```

CNUM	CNAME	CITY	RATING	SNUM
2004	Grass	Berlin	300	1002
2008	Cisneros	San Jose	300	1007
2003	Liu	San Jose	200	1002
2002	Giovanni	Rome	200	1003

Таблица 1.2: Заказчики (Customers)

CNUM	CNAME	CITY	RATING	SNUM
2001	Hoffman	London	100	1001
2002	Giovanni	Rome	200	1003
2003	Liu	SanJose	200	1002
2004	Grass	Berlin	300	1002
2006	Clemens	London	100	1001
2008	Cisneros	SanJose	300	1007
2007	Pereira	Rome	100	1004

Примеры выполнения SELECT

61

```
orcl | D  
0,00813952 seconds  
select distinct n from pd where not exists(select * from d  
where not exists(select * from pd pd2 where d.dn=pd2.dn and pd2.n=pd.n ))
```

Выбрать номера поставщиков, для которых не существовало бы ни одной детали так, чтобы эта деталь не была бы в поставках у данного поставщика

Results:

AZ	N
1	1

AZ	N	AZ	DN
1	1	1	1
2	1	2	2
3	1	3	3
4	2	1	1
5	2	2	2
6	3	1	1

AZ	DN
1	1
2	2
3	3

Восемь реляционных операторов в SQL

62

- **Теоретико-множественные операторы:**
 - Объединение **Union**
 - Пересечение **Intersect**
 - Вычитание **Minus**
 - Декартово произведение **Cross join**
- **Специальные реляционные операторы:**
 - Выборка **where**
 - Проекция
 - Соединение **inner join, outer join**
 - Деление **exists**

Union

63

```
SQL> select snum from orders union select snum from salespeople;
```

```
-----  
SNUM
```

```
1001  
1002  
1003  
1004  
1007
```

SQL Plus

```
SQL> select * from salespeople;
```

```
-----  
SNUM SNAME CITY COMM  
-----  
1001 Peel London .12  
1002 Serres San Jose .13  
1004 Motika London .11  
1007 Rifkin Barcelona .15  
1003 Axelrod New York .1
```

```
SQL> select * from orders;
```

```
-----  
ONUM AMT ODATE CNUM SNUM  
-----  
3001 18.69 10.03.1990 2008 1007  
3002 1900.1 10.03.1990 2007 1004  
3003 767.19 10.03.1990 2001 1001  
3005 5160.45 10.03.1990 2003 1002  
3006 1098.16 10.03.1990 2008 1007  
3009 1713.23 10.04.1990 2002 1003  
3007 75.75 10.04.1990 2004 1002  
3008 4723 10.05.1990 2006 1001  
3010 1309.95 10.06.1990 2004 1002  
3011 9891.88 10.06.1990 2006 1001
```

Union

64

Результат:

discount	title	old	new
10% off	Cooking with Computers: Surreptitious Balance Sheets	11.95	10.755
10% off	Emotional Security: A New Algorithm	7.99	7.191
10% off	Fifty Years in Buckingham Palace Kitchens	11.95	10.755
10% off	Is Anger the Enemy?	10.95	9.855
10% off	Life Without Fear	7	6.3
10% off	Sushi, Anyone?	14.99	13.491
20% off	The Gourmet Microwave	2.99	2.392
20% off	You Can Combat Computer Stress!	2.99	2.392

Intersect

65

```
SQL> select snum from orders intersect select snum from salespeople;
```

```
SNUM
```

```
-----  
1001  
1002  
1003  
1004  
1007
```

SQL Plus

```
SQL> select * from salespeople;
```

SNUM	SNAME	CITY	COMM
1001	Peel	London	.12
1002	Serres	San Jose	.13
1004	Motika	London	.11
1007	Rifkin	Barcelona	.15
1003	Axelrod	New York	.1

```
SQL> select * from orders;
```

ONUM	AMT	ODATE	CNUM	SNUM
3001	18.69	10.03.1990	2008	1007
3002	1900.1	10.03.1990	2007	1004
3003	767.19	10.03.1990	2001	1001
3005	5160.45	10.03.1990	2003	1002
3006	1098.16	10.03.1990	2008	1007
3009	1713.23	10.04.1990	2002	1003
3007	75.75	10.04.1990	2004	1002
3008	4723	10.05.1990	2006	1001
3010	1309.95	10.06.1990	2004	1002
3011	9891.88	10.06.1990	2006	1001

Minus

66

```
SQL> select snum from orders minus select snum from salespeople;  
no rows selected
```

SQL Plus

```
SQL> select * from salespeople;
```

SNUM	SNAME	CITY	COMM
1001	Peel	London	.12
1002	Serres	San Jose	.13
1004	Motika	London	.11
1007	Rifkin	Barcelona	.15
1003	Axelrod	New York	.1

```
SQL> select * from orders;
```

ONUM	AMT	ODATE	CNUM	SNUM
3001	18.69	10.03.1990	2008	1007
3002	1900.1	10.03.1990	2007	1004
3003	767.19	10.03.1990	2001	1001
3005	5160.45	10.03.1990	2003	1002
3006	1098.16	10.03.1990	2008	1007
3009	1713.23	10.04.1990	2002	1003
3007	75.75	10.04.1990	2004	1002
3008	4723	10.05.1990	2006	1001
3010	1309.95	10.06.1990	2004	1002
3011	9891.88	10.06.1990	2006	1001

Join

67

Существует три типа join-выражений:

- cross join;
- inner join;
- outer join;

Cross join

68

```
SQL> select o.snum,amt from orders o, salespeople;
```

SNUM	AMT
1007	18.69
1007	18.69
1007	18.69
1007	18.69
1007	18.69
1004	1900.1
1004	1900.1
1004	1900.1
1004	1900.1
1004	1900.1
1001	767.19

```
SQL> select s.snum from orders cross join salespeople s;
```

Соединение

69

Inner join

Join ... using on (...);

Natural join;

Outer join

Right join | Left join | Full join ... on () ...;

Inner join

70

```
SQL> select s.snum,amt from orders o, salespeople s where o.snum=s.snum;
```

SNUM	AMT
1007	18.69
1004	1900.1
1001	767.19
1002	5160.45
1007	1098.16
1003	1713.23
1002	75.75
1001	4723
1002	1309.95
1001	9891.88

10 rows selected.

SNUM

1007	18
1004	190
1001	767.19
1002	5160.45
1007	1098.16
1003	1713.23
1002	75.75
1001	4723
1002	1309.95
1001	9891.88

10 rows selected.

```
select * from (customers join orders using (snum)) NATURAL join salespeople;
```

Results Script Output Explain Autotrace DBMS Output OWA Output

Results:

	SNUM	CITY	CNUM	CNAME	RATING	ONUM	AMT	ODATE	CNUM_1	SNAME	COMM
1	1001	London	2001	Hoffman	100	3011	9891,88	10.06.90	2006	Peel	0,12
2	1001	London	2001	Hoffman	100	3008	4723	10.05.90	2006	Peel	0,12
3	1001	London	2001	Hoffman	100	3003	767,19	10.03.90	2001	Peel	0,12
4	1002	San Jose	2003	Liu	200	3010	1309,95	10.06.90	2004	Serres	0,13
5	1002	San Jose	2003	Liu	200	3007	75,75	10.04.90	2004	Serres	0,13
6	1002	San Jose	2003	Liu	200	3005	5160,45	10.03.90	2003	Serres	0,13
7	1001	London	2006	Clemens	100	3011	9891,88	10.06.90	2006	Peel	0,12
8	1001	London	2006	Clemens	100	3008	4723	10.05.90	2006	Peel	0,12
9	1001	London	2006	Clemens	100	3003	767,19	10.03.90	2001	Peel	0,12

Outer join

71

- **Outer join** может быть **left**, **right** и **full** (слово **outer** обычно опускается);
- Конструкция **join** располагается сразу после **select-выражения**.

... **join_type join table_name on condition ...**

join_type - тип join-выражения (**left**, **right** и **full**),

table_name - имя таблицы, которая
присоединяется к результату,

condition - условие объединения таблиц.

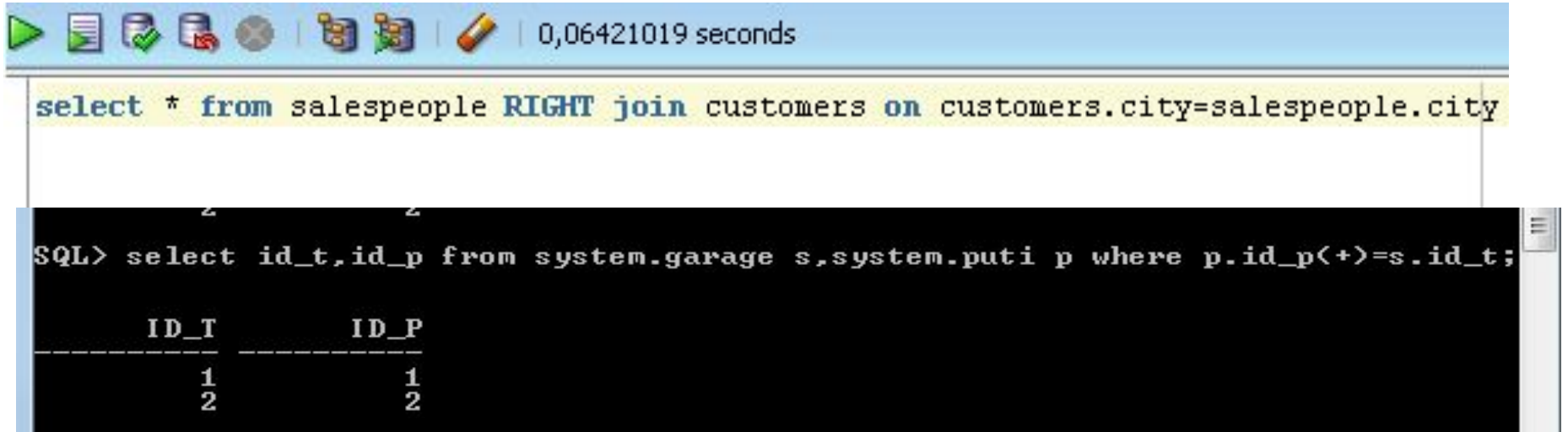
Outer join

72

- В случае с **Left join** из главной таблицы будут выбраны все записи, даже если в присоединяемой таблице нет совпадений, то есть условие `condition` не учитывает присоединяемую (правую) таблицу.
- **Right join** отображает все строки удовлетворяющие правой части условия `condition`, даже если они не имеют соответствия в главной (левой) таблице
- **Full outer join** необходим для отображения всех возможных комбинаций строк из нескольких

Outer join

73



A screenshot of a SQL IDE window. The title bar shows a green play button, a document icon, a green checkmark, a red 'X', a grey 'X', a database icon, a green arrow, and a pencil icon, followed by the text '0,06421019 seconds'. The main area contains a SQL query: `select * from salespeople RIGHT join customers on customers.city=salespeople.city`. Below the query, a black terminal window shows the execution of a different query: `SQL> select id_t,id_p from system.garage s,system.puti p where p.id_p(+)=s.id_t;`. The result is a table with two columns, ID_T and ID_P, and two rows of data.

```
select * from salespeople RIGHT join customers on customers.city=salespeople.city
```

```
SQL> select id_t,id_p from system.garage s,system.puti p where p.id_p(+)=s.id_t;
```

ID_T	ID_P
1	1
2	2

```
select * from orders full join customers on orders.cnum= customers.cnum
```

Деление

74

The screenshot displays the Oracle SQL Developer interface. At the top, a toolbar shows execution icons and a timer indicating 0,00813952 seconds. The main window contains a SQL query:

```
select distinct n from pd where not exists(select * from d  
where not exists(select * from pd pd2 where d.dn=pd2.dn and pd2.n=pd.n ))
```

Below the query, a toolbar offers options for Results, Script Output, Explain, Autotrace, DBMS Output, and OWA Output. The Results tab is active, showing a table with the following data:

N
1

Two smaller windows are also visible. The first, titled 'PD', shows a table with columns 'N' and 'DN':

N	DN
1	1
2	1
3	1
4	2
5	2
6	3

The second window, titled 'D', shows a table with columns 'DN' and 'N':

DN	N
1	1
2	2
3	3

СПАСИБО ЗА ВНИМАНИЕ!