



ИНТЕРНЕТ-СЕРВИС

# Основы SQL для аналитиков

Чулков В.А.

Чебоксары 2023

Занятие 1

# Основные понятия, реляционная модель данных



## Данные представлены в виде плоских таблиц, состоящих из столбцов и строк

	2			3					4
	123 employee_id	ABC first_name	ABC last_name	ABC email	ABC phone_number	hire_date	ABC job_id	123 salary	123 department_id
1	100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24 000	90
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17 000	90
3	102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17 000	90
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9 000	60
5	104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6 000	60
6	105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4 800	60
7	106	Valli	Pataballa	VPATABAL	590.423.4560	1998-02-08	IT_PROG	4 800	60
8	107	Diana	Lorentz	DLORENTZ	590.423.5567	1999-02-05	IT_PROG	4 200	60
9	108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17	FI_MGR	12 000	100
10	109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-16	FI_ACCOUNT	9 000	100
11	110	John	Chen	JCHEN	515.124.4269	1997-09-28	FI_ACCOUNT	8 200	100
12	111	Ismael	Sciarra	ISCIARRA	515.124.4369	1997-09-30	FI_ACCOUNT	7 700	100
13	112	Jose Manuel	Urman	JMURMAN	515.124.4469	1998-03-07	FI_ACCOUNT	7 800	100
14	113	Luis	Popp	LPOPP	515.124.4567	1999-12-07	FI_ACCOUNT	6 900	100
15	114	Den	Raphaely	DRAPHEAL	515.127.4561	1994-12-07	PU_MAN	11 000	30
16	115	Alexander	Khoo	AKHOO	515.127.4562	1995-05-18	PU_CLERK	3 100	30
17	116	Shelli	Baida	SBAIDA	515.127.4563	1997-12-24	PU_CLERK	2 900	30
18	117	Sigal	Tobias	STOBIAS	515.127.4564	1997-07-24	PU_CLERK	2 800	30
19	118	Guy	Himuro	GHIMURO	515.127.4565	1998-11-15	PU_CLERK	2 600	30
20	119	Karen	Colmenares	KCOLMENA	515.127.4566	1999-08-10	PU_CLERK	2 500	30

1. Строка (row)
2. Столбец, являющийся первичным ключом (primary key)
3. Столбец, не являющийся ключом
4. Столбец, являющийся внешним ключом (foreign key)
5. Поле (значение)

## Выборка данных с использованием команды SELECT языка SQL

Общий вид запроса:

```
SELECT <выражения для столбцов>
```

```
FROM <источник>
```

```
WHERE <условия выборки>
```

```
ORDER BY <столбцы сортировки>
```

## Практическое занятие 1

### 1.1. Будет ли успешно выполнен запрос?

```
select last_name, job_id, salary as Sal  
from employees;
```

### 1.2. Будет ли успешно выполнен запрос?

```
select *  
from job_grades
```

## Практическое занятие 1

### 1.4. Получить данные о всех должностях из таблицы employees.

	ABC job_id
1	SH_CLERK
2	AD_VP
3	SA_MAN
4	PU_MAN
5	IT_PROG
6	ST_CLERK
7	FI_MGR
8	PU_CLERK
9	HR_REP
10	ST_MAN
11	MK_MAN
12	AC_MGR
13	SA_REP
14	AD_ASST
15	PR_REP
16	MK_REP
17	AD PRES
18	FI_ACCOUNT
19	AC_ACCOUNT

## Практическое занятие 1

1.5. Получить данные о всех сотрудниках и идентификаторах их должностей. Нужно вывести фамилию, соединенную с идентификатором должности через запятую и пробел. Назвать столбец Employee and Title.

	ABC Employee and Title
1	King, AD_PRES
2	Kochhar, AD_VP
3	De Haan, AD_VP
4	Hunold, IT_PROG
5	Ernst, IT_PROG
6	Austin, IT_PROG
7	Pataballa, IT_PROG
8	Lorentz, IT_PROG
9	Greenberg, FI_MGR
10	Faviet, FI_ACCOUNT
11	Chen, FI_ACCOUNT
12	Sciarra, FI_ACCOUNT
13	Urman, FI_ACCOUNT
14	Popp, FI_ACCOUNT
15	Raphaely, PU_MAN
...	
105	Baer, PR_REP
106	Higgins, AC_MGR
107	Gietz, AC_ACCOUNT

## Задания на дом

...





## Занятие 2

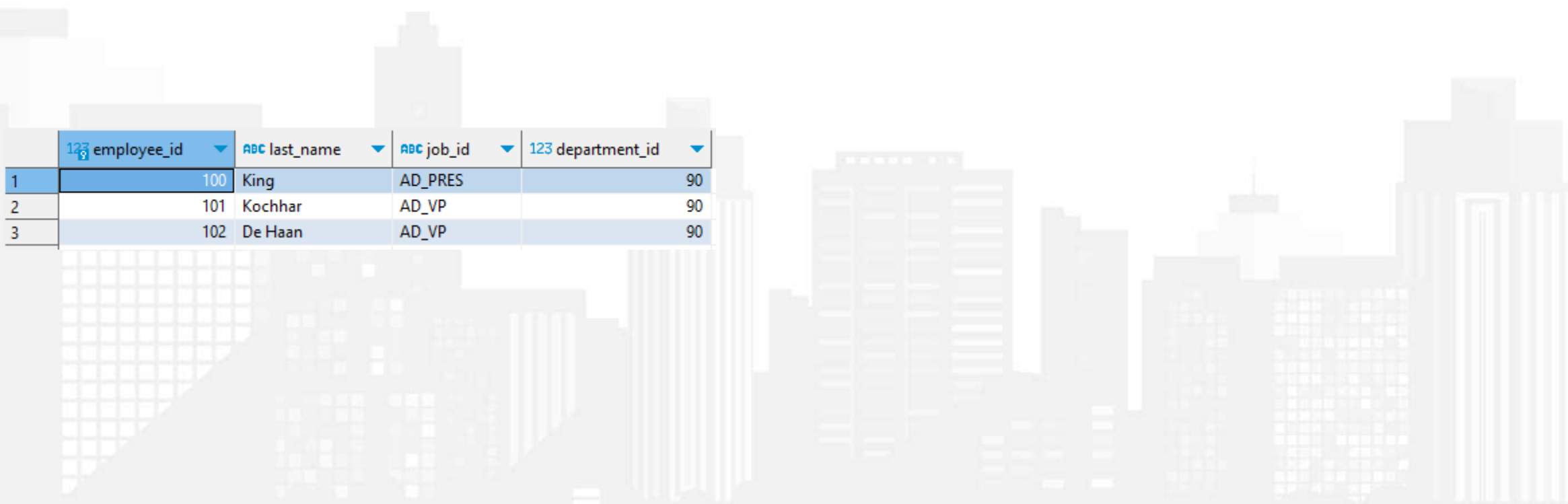
# Ограничение и сортировка выходных данных

Ограничение данных при помощи предложения WHERE

Сортировка данных при помощи предложения ORDER BY

## Использование предложения WHERE


```
select employee_id, last_name, job_id, department_id  
from employees  
where department_id = 90;
```



	123 employee_id ▼	ABC last_name ▼	ABC job_id ▼	123 department_id ▼
1	100	King	AD_PRES	90
2	101	Kochhar	AD_VP	90
3	102	De Haan	AD_VP	90

## Символьные строки

```
select last_name, job_id, department_id  
from employees  
where last_name = 'Whalen'
```



	ABC last_name ▼	ABC job_id ▼	123 department_id ▼
1	Whalen	AD_ASST	10

## Логические операторы

=

Равно

!=, <>

Не равно

>

Больше

>=

Больше или равно

<

Меньше

<=

Меньше или равно

between .. and ..

Принадлежит диапазону

in (...)

Принадлежит множеству

like

Похож

is null

Является Null

is not null

Не является Null

## Использование оператора сравнения

```
select last_name, salary  
from employees  
where salary < 3000;
```

	ABC last_name	123 salary
1	Baida	2 900
2	Tobias	2 800
3	Himuro	2 600
4	Colmenares	2 500
5	Mikkilineni	2 700
6	Landry	2 400
7	Markle	2 200
8	Atkinson	2 800
9	Marlow	2 500
10	Olson	2 100
11	Rogers	2 900
12	Gee	2 400
13	Philtanker	2 200
14	Seo	2 700
15	Patel	2 500
16	Matos	2 600
17	Vargas	2 500
18	Sullivan	2 500
19	Geoni	2 800
20	Gates	2 900
21	Perkins	2 500
22	Jones	2 800
23	OConnell	2 600
24	Grant	2 600

## Использование оператора between

```
select last_name, salary  
from employees  
where salary between 2500 and 3500;
```

	ABC last_name ▼	123 salary ▼
1	Khoo	3 100
2	Baida	2 900
3	Tobias	2 800
4	Himuro	2 600
5	Colmenares	2 500
6	Nayer	3 200
7	Mikkilineni	2 700
8	Bissot	3 300
9	Atkinson	2 800
10	Marlow	2 500
11	Mallin	3 300
12	Rogers	2 900
13	Stiles	3 200
14	Seo	2 700
15	Patel	2 500
16	Rajs	3 500
17	Davies	3 100
18	Matos	2 600
19	Vargas	2 500

...

## Использование оператора in

```
select employee_id, last_name, salary, manager_id
from employees
where manager_id in (100, 101, 201);
```

	123 employee_id ▼	ABC last_name ▼	123 salary ▼	123 manager_id ▼
1	101	Kochhar	17 000	100
2	102	De Haan	17 000	100
3	108	Greenberg	12 000	101
4	114	Raphaely	11 000	100
5	120	Weiss	8 000	100
6	121	Frapp	8 200	100
7	122	Kaufling	7 900	100
8	123	Vollman	6 500	100
9	124	Mourgos	5 800	100
10	145	Russell	14 000	100
11	146	Partners	13 500	100
12	147	Errazuriz	12 000	100
13	148	Cambrault	11 000	100
14	149	Zlotkey	10 500	100
15	200	Whalen	4 400	101
16	201	Hartstein	13 000	100
17	202	Fay	6 000	201
18	203	Mavris	6 500	101
19	204	Baer	10 000	101
20	205	Higgins	12 000	101

## Использование оператора like

% – Представляет любую последовательность из нуля или более символов

\_ и ? – Представляет любой одиночный символ

```
select first_name
```

```
from employees
```


```
where first_name like 'S%';
```

	ABC first_name
1	Steven
2	Shelli
3	Sigal
4	Shanta
5	Steven
6	Stephen
7	Sarath
8	Sundar
9	Sundita
10	Sarah
11	Samuel
12	Susan
13	Shelley



## Использование оператора like


```
select first_name  
from employees  
where first_name like '_o%';
```



ABC first_name	
1	John
2	Jose Manuel
3	Mozhe
4	John
5	Joshua
6	John
7	Louise
8	Jonathon
9	Donald
10	Douglas

## Использование оператора IS NULL

```
select last_name, manager_id  
from employees  
where manager_id is null;
```



	ABC last_name ▼	123 manager_id ▼
1	King	[NULL]

## Логический оператор AND

Оператор AND требует выполнения обоих условий

```
select employee_id, last_name, job_id, salary
from employees
where salary > 10000
and job_id like '%MAN%';
```

	123 employee_id ▼	ABC last_name ▼	ABC job_id ▼	123 salary ▼
1	114	Raphaely	PU_MAN	11 000
2	145	Russell	SA_MAN	14 000
3	146	Partners	SA_MAN	13 500
4	147	Errazuriz	SA_MAN	12 000
5	148	Cambrault	SA_MAN	11 000
6	149	Zlotkey	SA_MAN	10 500
7	201	Hartstein	MK_MAN	13 000

## Логический оператор OR

Оператор OR требует выполнения любого из условий

```
select employee_id, last_name, job_id, salary
from employees
where salary > 10000
or job_id like '%MAN%';
```

	123 employee_id ▼	ABC last_name ▼	ABC job_id ▼	123 salary ▼
1	100	King	AD_PRES	24 000
2	101	Kochhar	AD_VP	17 000
3	102	De Haan	AD_VP	17 000
4	108	Greenberg	FI_MGR	12 000
5	114	Raphaely	PU_MAN	11 000
6	120	Weiss	ST_MAN	8 000
7	121	Fripp	ST_MAN	8 200
8	122	Kauffling	ST_MAN	7 900
9	123	Vollman	ST_MAN	6 500
10	124	Mourgos	ST_MAN	5 800
11	145	Russell	SA_MAN	14 000
12	146	Partners	SA_MAN	13 500
13	147	Errazuriz	SA_MAN	12 000
14	148	Cambrault	SA_MAN	11 000
15	149	Zlotkey	SA_MAN	10 500
16	162	Vishney	SA_REP	10 500
17	168	Ozer	SA_REP	11 500
18	174	Abel	SA_REP	11 000
19	201	Hartstein	MK_MAN	13 000
20	205	Higgins	AC_MGR	12 000

## Логический оператор NOT

```
select last_name, job_id
from employees
where job_id not in ('IT_PROG', 'ST_CLERK', 'SA_REP');
```

	ABC last_name	ABC job_id
1	King	AD_PRES
2	Kochhar	AD_VP
3	De Haan	AD_VP
4	Greenberg	FI_MGR
5	Faviet	FI_ACCOUNT
6	Chen	FI_ACCOUNT
7	Sciarra	FI_ACCOUNT
8	Urman	FI_ACCOUNT
9	Popp	FI_ACCOUNT
10	Raphaely	PU_MAN
11	Khoo	PU_CLERK
12	Baida	PU_CLERK
13	Tobias	PU_CLERK
14	Himuro	PU_CLERK
15	Colmenares	PU_CLERK

...

## Приоритеты операторов

Арифметические операторы

Операторы конкатенации

Операторы сравнения

is null, is not null, like, in

between

NOT

AND

OR

Для управления последовательностью вычисления  
используются скобки

## Сортировка

```
select last_name, job_id, department_id, hire_date
from employees
order by hire_date;
```

	ABC last_name ▼	ABC job_id ▼	123 department_id ▼	🕒 hire_date ▼
1	King	AD_PRES	90	1987-06-17
2	Whalen	AD_ASST	10	1987-09-17
3	Kochhar	AD_VP	90	1989-09-21
4	Hunold	IT_PROG	60	1990-01-03
5	Ernst	IT_PROG	60	1991-05-21
6	De Haan	AD_VP	90	1993-01-13
7	Higgins	AC_MGR	110	1994-06-07
8	Gietz	AC_ACCOUNT	110	1994-06-07
9	Mavris	HR_REP	40	1994-06-07
10	Baer	PR_REP	70	1994-06-07
11	Faviet	FI_ACCOUNT	100	1994-08-16
12	Greenberg	FI_MGR	100	1994-08-17
13	Raphaely	PU_MAN	30	1994-12-07
14	Kaufling	ST_MAN	50	1995-05-01
15	Khoo	PU_CLERK	30	1995-05-18
16	Ladwig	ST_CLERK	50	1995-07-14
17	Rajs	ST_CLERK	50	1995-10-17

## Сортировка в порядке убывания

```
select last_name, job_id, department_id, hire_date
from employees
order by hire_date desc;
```

	ABC last_name ▼	ABC job_id ▼	123 department_id ▼	🕒 hire_date ▼
1	Kumar	SA_REP	80	2000-04-21
2	Banda	SA_REP	80	2000-04-21
3	Ande	SA_REP	80	2000-03-24
4	Markle	ST_CLERK	50	2000-03-08
5	Lee	SA_REP	80	2000-02-23
6	Philtanker	ST_CLERK	50	2000-02-06
7	Geoni	SH_CLERK	50	2000-02-03
8	Zlotkey	SA_MAN	80	2000-01-29
9	Marvins	SA_REP	80	2000-01-24
10	Grant	SH_CLERK	50	2000-01-13
11	Johnson	SA_REP	80	2000-01-04
12	Perkins	SH_CLERK	50	1999-12-19
13	Gee	ST_CLERK	50	1999-12-12
14	Popp	FI_ACCOUNT	100	1999-12-07
15	Tuvault	SA_REP	80	1999-11-23
16	Mourgos	ST_MAN	50	1999-11-16
17	Cambrault	SA_MAN	80	1999-10-15
18	Colmenares	PU_CLERK	30	1999-08-10



## Сортировка по псевдониму столбца

```
select last_name, job_id, department_id, hire_date hd
from employees
order by hd;
```

	ABC last_name ▼	ABC job_id ▼	123 department_id ▼	🕒 hd ▼
1	King	AD_PRES	90	1987-06-17
2	Whalen	AD_ASST	10	1987-09-17
3	Kochhar	AD_VP	90	1989-09-21
4	Hunold	IT_PROG	60	1990-01-03
5	Ernst	IT_PROG	60	1991-05-21
6	De Haan	AD_VP	90	1993-01-13
7	Higgins	AC_MGR	110	1994-06-07
8	Gietz	AC_ACCOUNT	110	1994-06-07
9	Mavris	HR_REP	40	1994-06-07
10	Baer	PR_REP	70	1994-06-07
11	Faviet	FI_ACCOUNT	100	1994-08-16
12	Greenberg	FI_MGR	100	1994-08-17
13	Raphaely	PU_MAN	30	1994-12-07
14	Kaufling	ST_MAN	50	1995-05-01
15	Khoo	PU_CLERK	30	1995-05-18
16	Ladwig	ST_CLERK	50	1995-07-14
17	Rajs	ST_CLERK	50	1995-10-17

## Сортировка по нескольким столбцам

```
select last_name, job_id, department_id, hire_date
from employees
order by department_id desc, last_name;
```

	ABC last_name ▼	ABC job_id ▼	123 department_id ▼	🕒 hire_date ▼
1	Grant	SA_REP	[NULL]	1999-05-24
2	Gietz	AC_ACCOUNT	110	1994-06-07
3	Higgins	AC_MGR	110	1994-06-07
4	Chen	FI_ACCOUNT	100	1997-09-28
5	Faviet	FI_ACCOUNT	100	1994-08-16
6	Greenberg	FI_MGR	100	1994-08-17
7	Popp	FI_ACCOUNT	100	1999-12-07
8	Sciarra	FI_ACCOUNT	100	1997-09-30
9	Urman	FI_ACCOUNT	100	1998-03-07
10	De Haan	AD_VP	90	1993-01-13
11	King	AD_PRES	90	1987-06-17
12	Kochhar	AD_VP	90	1989-09-21
13	Abel	SA_REP	80	1996-05-11
14	Ande	SA_REP	80	2000-03-24
15	Banda	SA_REP	80	2000-04-21
16	Bates	SA_REP	80	1999-03-24
17	Bernstein	SA_REP	80	1997-03-24

## Практическое занятие 2

2.1. Вывести фамилии(`last_name`) и оклады(`salary`) служащих, получающих более 12000

	ABC last_name ▼	123 salary ▼
1	King	24 000
2	Kochhar	17 000
3	De Haan	17 000
4	Russell	14 000
5	Partners	13 500
6	Hartstein	13 000

## Практическое занятие 2 (продолжение)

2.2. Вывести фамилию (`last_name`) и номер отдела (`department_id`) служащего под номером 176

	ABC last_name ▼	123 department_id ▼
1	Taylor	80

## Практическое занятие 2 (продолжение)

2.3. Вывести фамилии (`last_name`) и оклады (`salary`) всех служащих, чей оклад не входит в диапазон от 5000 до 12000

	ABC last_name ▼	123 salary ▼
1	King	24 000
2	Kochhar	17 000
3	De Haan	17 000
4	Austin	4 800
5	Pataballa	4 800
6	Lorentz	4 200
7	Khoo	3 100
8	Baida	2 900
9	Tobias	2 800
10	Himuro	2 600
11	Colmenares	2 500
12	Nayer	3 200
13	Mikkilineni	2 700
14	Landry	2 400
15	Markle	2 200
...		
53	Grant	2 600
54	Whalen	4 400
55	Hartstein	13 000

## Практическое занятие 2 (продолжение)

2.4. Вывести фамилию (`last_name`), идентификатор должности (`job_id`) и дату найма (`hire_date`) всех служащих с фамилиями Matos и Taylor. Отсортируйте данные в порядке возрастания даты найма.

	ABC last_name ▼	ABC job_id ▼	🕒 hire_date ▼
1	Taylor	SH_CLERK	1998-01-24
2	Matos	ST_CLERK	1998-03-15
3	Taylor	SA_REP	1998-03-24

## Практическое занятие 2 (продолжение)

2.5. Вывести фамилию (`last_name`) и номер отдела (`department_id`) всех служащих из отделов 20 и 50. Отсортируйте данные по фамилиям в алфавитном порядке.

	ABC last_name ▼	123 department_id ▼
1	Atkinson	50
2	Bell	50
3	Bissot	50
4	Bull	50
5	Cabrio	50
6	Chung	50
7	Davies	50
8	Dellinger	50
9	Dilly	50
10	Everett	50
11	Fay	20
12	Feeney	50
13	Fleur	50
14	Fripp	50
15	Gates	50
...		
45	Vollman	50
46	Walsh	50
47	Weiss	50

## Практическое занятие 2 (продолжение)

2.6. Вывести фамилии (`last_name`) и оклады (`salary`) служащих отделов 20 и 50, зарабатывающих от 5000 до 12000.

Назовите столбцы `Employee` и `Monthly Salary`, соответственно.

	ABC Employee ▾	123 Monthly Salary ▾
1	Weiss	8 000
2	Fripp	8 200
3	Kaufling	7 900
4	Vollman	6 500
5	Mourgos	5 800
6	Fay	6 000




## Практическое занятие 2 (продолжение)

2.7. Вывести фамилии (`last_name`) и даты найма (`hire_date`) всех служащих, нанятых в 1994 году.

	ABC last_name ▼	🕒 hire_date ▼
1	Greenberg	1994-08-17
2	Faviet	1994-08-16
3	Raphaely	1994-12-07
4	Mavris	1994-06-07
5	Baer	1994-06-07
6	Higgins	1994-06-07
7	Gietz	1994-06-07

## Практическое занятие 2 (продолжение)

2.8. Вывести фамилии (`last_name`) и должности (`job_id`) всех служащих, не имеющих менеджера.



	ABC last_name ▼	ABC job_id ▼
1	King	AD_PRES

## Практическое занятие 2 (продолжение)

2.9. Вывести фамилии (`last_name`), оклады (`salary`) и комиссионные всех служащих, зарабатывающих комиссионные.

Отсортируйте данные в порядке убывания окладов и КОМИССИОННЫХ.

	ABC last_name ▼	123 salary ▼	123 commission_pct ▼
1	Russell	14 000	0,4
2	Partners	13 500	0,3
3	Errazuriz	12 000	0,3
4	Ozer	11 500	0,25
5	Cambrault	11 000	0,3
6	Abel	11 000	0,3
7	Vishney	10 500	0,25
8	Zlotkey	10 500	0,2
9	King	10 000	0,35
10	Tucker	10 000	0,3
11	Bloom	10 000	0,2
12	Fox	9 600	0,2
13	Sully	9 500	0,35
14	Bernstein	9 500	0,25
15	Greene	9 500	0,15
...			
33	Banda	6 200	0,1
34	Johnson	6 200	0,1
35	Kumar	6 100	0,1

## Практическое занятие 2 (продолжение)

2.10. Вывести все фамилии служащих, в которых третья буква – **a**

	ABC last_name ▼
1	Grant
2	Grant
3	Whalen

## Практическое занятие 2 (продолжение)

2.11. Вывести все фамилии служащих, в которых есть буквы **а** и **е**

	ABC last_name ▼
1	De Haan
2	Faviet
3	Raphaely
4	Colmenares
5	Nayer
6	Markle
7	Philtanker
8	Patel
9	Davies
10	Partners
11	Sewall
12	Bates
13	Fleaur
14	Gates
15	Whalen
16	Hartstein
17	Baer

## Практическое занятие 2 (продолжение)

2.12. Вывести фамилии, должности и оклады всех служащих, работающих торговыми представителями (SA\_REP) или клерками на складе (ST\_CLERK), у которых оклад не равен 2500, 3500, 7000.

	ABC last_name ▼	ABC job_id ▼	123 salary ▼
1	Nayer	ST_CLERK	3 200
2	Mikkilineni	ST_CLERK	2 700
3	Landry	ST_CLERK	2 400
4	Markle	ST_CLERK	2 200
5	Bissot	ST_CLERK	3 300
6	Atkinson	ST_CLERK	2 800
7	Olson	ST_CLERK	2 100
8	Mallin	ST_CLERK	3 300
9	Rogers	ST_CLERK	2 900
10	Gee	ST_CLERK	2 400
11	Philtanker	ST_CLERK	2 200
12	Ladwig	ST_CLERK	3 600
13	Stiles	ST_CLERK	3 200
14	Seo	ST_CLERK	2 700
15	Davies	ST_CLERK	3 100
...			
41	Taylor	SA_REP	8 600
42	Livingston	SA_REP	8 400
43	Johnson	SA_REP	6 200

## Практическое занятие 2 (продолжение)

2.13. Вывести фамилии, оклады и комиссионные всех служащих, у которых сумма комиссионных составляет 20%.

	ABC last_name ▼	123 salary ▼	123 commission_pct ▼
1	Zlotkey	10 500	0,2
2	Olsen	8 000	0,2
3	Cambrault	7 500	0,2
4	Bloom	10 000	0,2
5	Fox	9 600	0,2
6	Taylor	8 600	0,2
7	Livingston	8 400	0,2

## Занятие 3

# Однострочные функции

Символьные функции

Числовые функции

Функции для работы с датами

Функции преобразования

Выражение CASE



## Символьные функции

-- **lower(str)** - преобразует алфавитные символы в нижний регистр

```
select lower('abABcde efGH');
```

-- **upper(str)** - преобразует алфавитные символы в верхний регистр

```
select upper('abABcde efGH');
```

-- **initcap(str)** - преобразует алфавитные символы: первую букву

-- каждого слова в верхний регистр, остальные - в нижний

```
select initcap('abABcde efGH');
```

## Символьные функции (продолжение)

```
select
```

```
-- сцепляет переданные значения в один текст
```

```
concat('abc', 123, 'efg'),
```

```
-- substr(str, m, [n]), возвращает n-символов (или оставшуюся часть строки),
```

```
-- начиная с m-го символа
```

```
substr('Hello, SQL-function', 8, 3),
```

```
--      123456789
```

```
-- length(str) - возвращает длину строки
```

```
length('abc');
```

## Символьные функции (продолжение)

```
select
```

```
-- дополняет символьное значение, выровненное справа, до заданной длины.
```

```
lpad('abcd', 6, '1'),
```

```
-- дополняет символьное значение, выровненное слева, до заданной длины.
```

```
rpadd('abcd', 6, '1'),
```

```
-- удаляет из символьной строки начальные и конечные символы.
```

```
trim('--abc--', '-'),
```

```
-- Заменяет последовательность символов в строке на новую
```

```
replace('abcdabcd', 'cd', 'KL'),
```

```
-- возвращает указанное количество символов строки слева
```

```
left('abcd',2),
```

```
-- возвращает указанное количество символов строки справа
```

```
right('abcd', 3);
```

## Пример вложенных вызовов функции

```
select  
concat(last_name, ' ', first_name, ', длина: ',length(last_name)) fio,  
length(last_name)  
from employees
```

## Числовые функции

```
select
-- округление до целого
round(1.23456) r1,
-- округление до 3-го знака
round(1.23456, 3) r2,
-- округление вниз до 3-го знака
trunc(1.23456, 3) t,
-- остаток от деления
mod(34, 10) m;
```

## Функции для работы с датами

```
select
-- текущая дата
now(),
-- год
extract('year' from now()),
-- месяц
extract('month' from now()),
-- день
extract('day' from now());
```

## Функции преобразования

```
select  
to_char(12345.78901234567, '99999999990D999'),  
TO_DATE('20170103','YYYYMMDD'),  
to_char(TO_DATE('20170103','YYYYMMDD'), 'YYYY.MM.DD')
```

## Выражение CASE. Вариант 1

```
case значение  
when вариант1 then результат1  
...  
when вариантN then результатN  
else иначе  
end
```



## Выражение CASE. Вариант 1 (продолжение)

Пример:

```
select last_name,  
case last_name  
when 'King' then 'КИНГ'  
when 'Atkinson' then 'АТКИНСОН'  
else 'Не знаю кто' end who  
from employees;
```

## Выражение CASE. Вариант 2

```
case  
when условие1 then результат1  
...  
when условиеN then результатN  
else иначе  
end
```

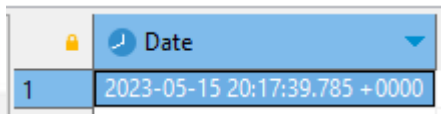
## Выражение CASE. Вариант 2 (продолжение)

Пример:

```
select last_name,  
case when hire_date < '1993-01-01' then 'давно работает'  
when hire_date between '1993-01-01' and '1997-01-01' then  
'работает не очень давно'  
else 'работает недавно'  
end as Wwork  
from employees;
```

## Практическое занятие 3

3.1. Напишите запрос для вывода текущей даты. Назовате столбец Date.



The screenshot shows a single-row query result. The first column is labeled '1' and the second column is labeled 'Date'. The value in the 'Date' column is '2023-05-15 20:17:39.785 +0000'.

	Date
1	2023-05-15 20:17:39.785 +0000

## Практическое занятие 3

3.2. Вывести номер служащего, его фамилию, оклад и новый оклад, повышенный на 15.5% и округленный до целого.

Столбец, содержащий новый оклад, должен иметь имя New Salary.

	123 employee_id	ABC last_name	123 salary	123 New Salary
1	100	King	24 000	27 720
2	101	Kochhar	17 000	19 635
3	102	De Haan	17 000	19 635
4	103	Hunold	9 000	10 395
5	104	Ernst	6 000	6 930
6	105	Austin	4 800	5 544
7	106	Pataballa	4 800	5 544
8	107	Lorentz	4 200	4 851
9	108	Greenberg	12 000	13 860
10	109	Faviet	9 000	10 395
11	110	Chen	8 200	9 471
12	111	Sciarra	7 700	8 894
13	112	Urman	7 800	9 009
14	113	Popp	6 900	7 970
15	114	Raphaely	11 000	12 705
...				
105	204	Baer	10 000	11 550
106	205	Higgins	12 000	13 860
107	206	Gietz	8 300	9 587

## Практическое занятие 3

3.3. В запрос к предыдущему заданию добавьте столбец, который будет содержать результат вычитания старого оклада из нового. Назовите столбец Increase.

	123 employee_id	ABC last_name	123 salary	123 New Salary	123 Increase
1	100	King	24 000	27 720	3 720
2	101	Kochhar	17 000	19 635	2 635
3	102	De Haan	17 000	19 635	2 635
4	103	Hunold	9 000	10 395	1 395
5	104	Ernst	6 000	6 930	930
6	105	Austin	4 800	5 544	744
7	106	Pataballa	4 800	5 544	744
8	107	Lorentz	4 200	4 851	651
9	108	Greenberg	12 000	13 860	1 860
10	109	Faviet	9 000	10 395	1 395
11	110	Chen	8 200	9 471	1 271
12	111	Sciarra	7 700	8 894	1 194
13	112	Urman	7 800	9 009	1 209
14	113	Popp	6 900	7 970	1 070
15	114	Raphaely	11 000	12 705	1 705
...					
105	204	Baer	10 000	11 550	1 550
106	205	Higgins	12 000	13 860	1 860
107	206	Gietz	8 300	9 587	1 287

## Практическое занятие 3

3.4. Выведите фамилии служащих (первая буква фамилии должна быть заглавной, а остальные – строчными) и длину каждой фамилии для тех служащих, фамилия которых начинается с символа J, A или M. Столбцы назовите Name и Length соответственно. Отсортируйте результат по фамилиям служащих.

	ABC Name	123 Length
1	Austin	6
2	Mourgos	7
3	Mikkilineni	11
4	Markle	6
5	Atkinson	8
6	Marlow	6
7	Mallin	6
8	Matos	5
9	Mcewen	6
10	Marvins	7
11	Ande	4
12	Abel	4
13	Johnson	7
14	Mccain	6
15	Jones	5
16	Mavris	6

## Практическое занятие 3

3.5. Для каждого служащего выведите фамилию, дату найма и год найма (назовите `hire_year`).

	ABC last_name ▼	🕒 hire_date ▼	123 hire_year ▼
1	King	1987-06-17	1 987
2	Kochhar	1989-09-21	1 989
3	De Haan	1993-01-13	1 993
4	Hunold	1990-01-03	1 990
5	Ernst	1991-05-21	1 991
6	Austin	1997-06-25	1 997
7	Pataballa	1998-02-05	1 998
8	Lorentz	1999-02-07	1 999
9	Greenberg	1994-08-17	1 994
10	Faviet	1994-08-16	1 994
11	Chen	1997-09-28	1 997
12	Sciarra	1997-09-30	1 997
13	Urman	1998-03-07	1 998
14	Popp	1999-12-07	1 999
15	Raphaely	1994-12-07	1 994
...			
105	Baer	1994-06-07	1 994
106	Higgins	1994-06-07	1 994
107	Gietz	1994-06-07	1 994



## Практическое занятие 3

3.6. Вывести по каждому служащему отчет в следующем виде:

<фамилия> зарабатывает <оклад> в месяц, но желает <утроенный оклад>. Назовите столбец Dream Salaries. Для преобразования числа в текст используйте функцию `to_char(salary, 'FM99999999999999999999')`

	ABC Dream Salaries
1	King зарабатывает 24000 в месяц, но желает 72000
2	Kochhar зарабатывает 17000 в месяц, но желает 51000
3	De Haan зарабатывает 17000 в месяц, но желает 51000
4	Hunold зарабатывает 9000 в месяц, но желает 27000
5	Ernst зарабатывает 6000 в месяц, но желает 18000
6	Austin зарабатывает 4800 в месяц, но желает 14400
7	Pataballa зарабатывает 4800 в месяц, но желает 14400
8	Lorentz зарабатывает 4200 в месяц, но желает 12600
9	Greenberg зарабатывает 12000 в месяц, но желает 36000
10	Faviet зарабатывает 9000 в месяц, но желает 27000
11	Chen зарабатывает 8200 в месяц, но желает 24600
12	Sciarra зарабатывает 7700 в месяц, но желает 23100
13	Urman зарабатывает 7800 в месяц, но желает 23400
14	Popp зарабатывает 6900 в месяц, но желает 20700
15	Raphaely зарабатывает 11000 в месяц, но желает 33000
...	
105	Багг зарабатывает 10000 в месяц, но желает 30000
106	Higgins зарабатывает 12000 в месяц, но желает 36000
107	Gietz зарабатывает 8300 в месяц, но желает 24900

## Практическое занятие 3

3.7. Вывести фамилии и оклады всех служащих. Столбец с окладом назовите Salary. Длина столбца должна быть 15 символов с дополненными слева символами \$. Для преобразования числа в текст используйте функцию `to_char(salary, 'FM999999999999999999')`.

	ABC last_name	ABC Salary
1	King	\$\$\$\$\$\$\$\$\$24000
2	Kochhar	\$\$\$\$\$\$\$\$\$17000
3	De Haan	\$\$\$\$\$\$\$\$\$17000
4	Hunold	\$\$\$\$\$\$\$\$\$9000
5	Ernst	\$\$\$\$\$\$\$\$\$6000
6	Austin	\$\$\$\$\$\$\$\$\$4800
7	Pataballa	\$\$\$\$\$\$\$\$\$4800
8	Lorentz	\$\$\$\$\$\$\$\$\$4200
9	Greenberg	\$\$\$\$\$\$\$\$\$12000
10	Faviet	\$\$\$\$\$\$\$\$\$9000
11	Chen	\$\$\$\$\$\$\$\$\$8200
12	Sciarra	\$\$\$\$\$\$\$\$\$7700
13	Urman	\$\$\$\$\$\$\$\$\$7800
14	Popp	\$\$\$\$\$\$\$\$\$6900
15	Raphaely	\$\$\$\$\$\$\$\$\$11000
...		
105	Baer	\$\$\$\$\$\$\$\$\$10000
106	Higgins	\$\$\$\$\$\$\$\$\$12000
107	Gietz	\$\$\$\$\$\$\$\$\$8300

## Практическое занятие 3

3.8.\* Вывести фамилии и суммы комиссионных каждого служащего. Если служащий не зарабатывает комиссионных, укажите в столбце "No Commission". Назовите столбец Comm.

Для преобразования числа в текст используйте функцию `to_char` с маской `'FM999999999999999999999999D00'`

	ABC last_name	ABC Comm
1	King	No Commission
2	Kochhar	No Commission
3	De Haan	No Commission
4	Hunold	No Commission
5	Ernst	No Commission
6	Austin	No Commission
7	Pataballa	No Commission
8	Lorentz	No Commission
9	Greenberg	No Commission
10	Faviet	No Commission
11	Chen	No Commission
12	Sciarra	No Commission
13	Urman	No Commission
14	Popp	No Commission
15	Raphaely	No Commission
...		
105	Baer	No Commission
106	Higgins	No Commission
107	Gietz	No Commission

## Практическое занятие 3

3.9.\* Выведите первые восемь букв фамилии сотрудников, затем пробел, затем заработную плату в виде гистограммы, состоящей из звездочек, где каждая звездочка означает \$1000. Результат отсортируйте по убыванию заработной платы. Пример: King \*\*\*\*\*

Результат должен быть выведен одним столбцом, озаглавленным как `employees_and_their_salaries`

Для вычисления потребуется явное преобразование типа в целое число, используйте `salary::int`

	ABC employees_and_their_salaries
1	King *****
2	Kochhar *****
3	De Haan *****
4	Russell *****
5	Partners *****
...	
105	Philtank **
106	Markle **
107	Olson **

## Практическое занятие 3

3.10.\* Используя оператор CASE, напишите запрос для отображения должности сотрудника и ее разряда (grade), строки не должны повторяться.

Соответствие типа должности job\_id:

AD\_PRES – A

ST\_MAN – B

IT\_PROG – C

SA\_REP – D

ST\_CLERK – E

Другая – 0

	ABC job_id	ABC grade
1	PU_MAN	0
2	FI_MGR	0
3	HR_REP	0
4	ST_CLERK	E
5	AD_ASST	0
6	SA_MAN	0
7	AD_PRES	A
8	AC_MGR	0
9	MK_REP	0
10	ST_MAN	B
11	FI_ACCOUNT	0
12	PR_REP	0
13	MK_MAN	0
14	SA_REP	D
15	PU_CLERK	0
16	IT_PROG	C
17	AC_ACCOUNT	0
18	AD_VP	0
19	SH_CLERK	0

Занятие 4

# Агрегирование данных

Групповые функции

Группировка при помощи предложения group by

Предложение having

## Общий вид запроса

SELECT <выражения для групповых столбцов>

FROM <источник>

WHERE <условия выборки>

**GROUP BY** <столбцы группировки>

**HAVING** <условия выборки сгруппированных данных>

ORDER BY <столбцы сортировки>

Групповые функции работают с множеством строк и возвращают один результат на группу.

	123 employee_id	ABC last_name	ABC job_id	123 salary	123 department_id
1	200	Whalen	AD_ASST	4 400	10
2	201	Hartstein	MK_MAN	13 000	20
3	202	Fay	MK_REP	6 000	20
4	114	Raphaely	PU_MAN	11 000	30
5	115	Khoo	PU_CLERK	3 100	30
6	116	Baida	PU_CLERK	2 900	30
7	117	Tobias	PU_CLERK	2 800	30
8	118	Himuro	PU_CLERK	2 600	30
9	119	Colmenares	PU_CLERK	2 500	30
10	100	King	AD_PRES	24 000	90
11	101	Kochhar	AD_VP	17 000	90
12	102	De Haan	AD_VP	17 000	90
13	205	Higgins	AC_MGR	12 000	110
14	206	Gietz	AC_ACCOUNT	8 300	110

max(salary)

	123 max
1	24 000



## Вызов групповой функции на всех значениях

НАЗВАНИЕ\_ФУНКЦИИ (ВЫРАЖЕНИЕ)



## Вызов групповой функции без учета дубликатов

НАЗВАНИЕ\_ФУНКЦИИ (DISTINCT ВЫРАЖЕНИЕ)



## Групповая функция count

**count(expr)** – подсчет количества строк с непустым (не null) значением expr;

**count(distinct expr)** – подсчет количества уникальных значений expr, null не учитывается;

**count(\*)** – подсчет количества всех строк.

## Другие групповые функции

**min** – минимальное значение выражения;

**max** – максимальное значение выражения;

**avg** – среднее значение выражения;

**sum** – сумма значений.

Эти функции игнорируют значения `null`, допускают использование ключевого слова `distinct` и не допускают использование оператора `*`.

## Пример запроса:

```
select sum(salary) from employees;
```

	123 employee_id	ABC last_name	ABC job_id	123 salary	123 department_id
1	200	Whalen	AD_ASST	4 400	10
2	201	Hartstein	MK_MAN	13 000	20
3	202	Fay	MK_REP	6 000	20
4	114	Raphaely	PU_MAN	11 000	30
5	115	Khoo	PU_CLERK	3 100	30
6	116	Baida	PU_CLERK	2 900	30
7	117	Tobias	PU_CLERK	2 800	30
8	118	Himuro	PU_CLERK	2 600	30
9	119	Colmenares	PU_CLERK	2 500	30
10	100	King	AD_PRES	24 000	90
11	101	Kochhar	AD_VP	17 000	90
12	102	De Haan	AD_VP	17 000	90
13	205	Higgins	AC_MGR	12 000	110
14	206	Gietz	AC_ACCOUNT	8 300	110

sum(salary)

Аккумулятор

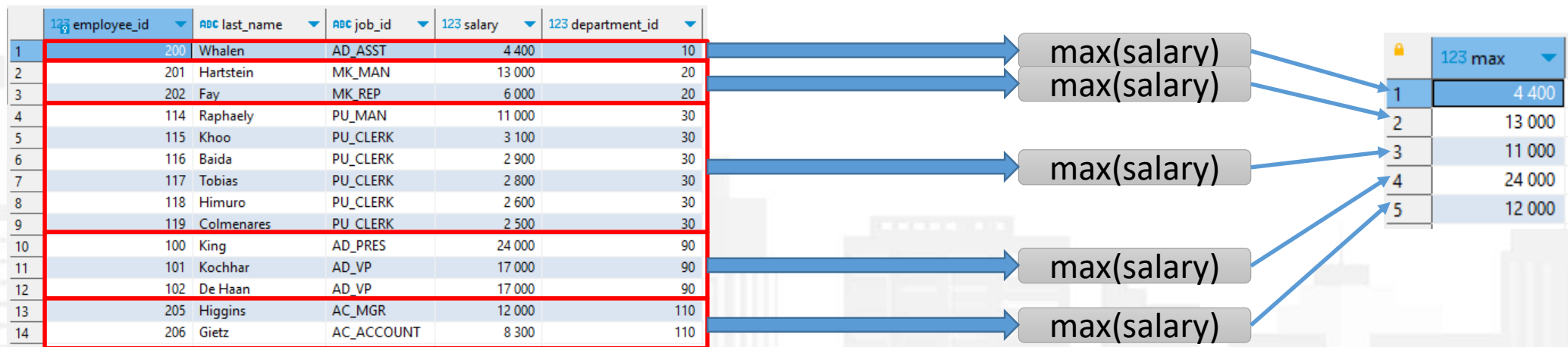
	123 sum
1	126 600

## Практическое занятие. Использование групповых функций.

1. Вывести минимальный, максимальный, средний оклад служащих, сумму окладов и количество
2. Вывести среднее значение комиссионных, количество служащих получающих комиссионные

## Создание групп данных

Выполнение групповой функции на нескольких группах



## Использование предложения GROUP BY

```
SELECT department_id, MAX(salary) max_salary
FROM employees
GROUP BY department_id;
```

	123 employee_id	ABC last_name	ABC job_id	123 salary	123 department_id
1	200	Whalen	AD_ASST	4 400	10
2	201	Hartstein	MK_MAN	13 000	20
3	202	Fay	MK_REP	6 000	20
4	114	Raphaely	PU_MAN	11 000	30
5	115	Khoo	PU_CLERK	3 100	30
6	116	Baida	PU_CLERK	2 900	30
7	117	Tobias	PU_CLERK	2 800	30
8	118	Himuro	PU_CLERK	2 600	30
9	119	Colmenares	PU_CLERK	2 500	30
10	100	King	AD_PRES	24 000	90
11	101	Kochhar	AD_VP	17 000	90
12	102	De Haan	AD_VP	17 000	90
13	205	Higgins	AC_MGR	12 000	110
14	206	Gietz	AC_ACCOUNT	8 300	110



	123 department_id	123 max_salary
1	10	4 400
2	20	13 000
3	30	11 000
4	90	24 000
5	110	12 000



## Группировка по нескольким столбцам

```
SELECT department_id, job_id, sum(salary) AS sum_salary
FROM employees
GROUP BY department_id, job_id;
```

	123 employee_id	ABC last_name	ABC job_id	123 salary	123 department_id
1	200	Whalen	AD_ASST	4 400	10
2	201	Hartstein	MK_MAN	13 000	20
3	202	Fay	MK_REP	6 000	20
4	114	Raphaely	PU_MAN	11 000	30
5	115	Khoo	PU_CLERK	3 100	30
6	116	Baida	PU_CLERK	2 900	30
7	117	Tobias	PU_CLERK	2 800	30
8	118	Himuro	PU_CLERK	2 600	30
9	119	Colmenares	PU_CLERK	2 500	30
10	100	King	AD_PRES	24 000	90
11	101	Kochhar	AD_VP	17 000	90
12	102	De Haan	AD_VP	17 000	90
13	205	Higgins	AC_MGR	12 000	110
14	206	Gietz	AC_ACCOUNT	8 300	110



	123 department_id	ABC job_id	123 sum_salary
1	90	AD_PRES	24 000
2	110	AC_MGR	12 000
3	30	PU_CLERK	13 900
4	10	AD_ASST	4 400
5	30	PU_MAN	11 000
6	90	AD_VP	34 000
7	110	AC_ACCOUNT	8 300
8	20	MK_REP	6 000
9	20	MK_MAN	13 000

## Недействительные запросы с группировкой

### 1. Использование в выборке столбца не участвующего в группировке

```
select department_id dept, count(last_name) cnt  
from employees;
```

### 2. Использование групповых функций в предложении where

```
select department_id dept, avg(salary) avg_salary  
from employees  
where avg(salary) > 8000;
```

## Исключение групп

```
SELECT department_id, max(salary) AS max_salary
FROM employees
GROUP BY department_id
HAVING avg(salary) < 10000;
```

	123 employee_id	ABC last_name	ABC job_id	123 salary	123 department_id
1	200	Whalen	AD_ASST	4 400	10
2	201	Hartstein	MK_MAN	13 000	20
3	202	Fay	MK_REP	6 000	20
4	114	Raphaely	PU_MAN	11 000	30
5	115	Khoo	PU_CLERK	3 100	30
6	116	Baida	PU_CLERK	2 900	30
7	117	Tobias	PU_CLERK	2 800	30
8	118	Himuro	PU_CLERK	2 600	30
9	119	Colmenares	PU_CLERK	2 500	30
10	100	King	AD_PRES	24 000	90
11	101	Kochhar	AD_VP	17 000	90
12	102	De Haan	AD_VP	17 000	90
13	205	Higgins	AC_MGR	12 000	110
14	206	Gietz	AC_ACCOUNT	8 300	110



	123 department_id	123 max_salary
1	10	4 400
2	20	13 000
3	30	11 000
4	90	24 000
5	110	12 000

## Последовательность выборки

```
select department_id, sum(salary * 1.3) new_salary
from employees
where salary < 10000
group by department_id
having sum(salary * 1.3) < 8000
order by new_salary desc
```

	123 employee_id	ABC first_name	ABC last_name	ABC email	ABC phone_number	hire_date	ABC job_id	123 salary
1	100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24 000
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17 000
3	102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17 000
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9 000
5	104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6 000
6	105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4 800
7	106	Valli	Pataballa	VPATABAL	590.423.4560	1998-02-05	IT_PROG	4 800
8	107	Diana	Lorentz	DLORENTZ	590.423.5567	1999-02-07	IT_PROG	4 200
9	108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17	FI_MGR	12 000
10	109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-16	FI_ACCOUNT	9 000
11	110	John	Chen	JCHEN	515.124.4269	1997-09-28	FI_ACCOUNT	8 200
12	111	Ismael	Sciarra	ISCIARRA	515.124.4369	1997-09-30	FI_ACCOUNT	7 700
13	112	Jose Manuel	Urman	JMURMAN	515.124.4469	1998-03-07	FI_ACCOUNT	7 800
14	113	Luis	Popp	LOPP	515.124.4567	1999-12-07	FI_ACCOUNT	6 900

## Последовательность выборки

select department\_id, sum(salary \* 1.3) new\_salary  
from employees

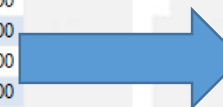
where salary < 10000

group by department\_id

having sum(salary \* 1.3) < 8000

order by new\_salary desc

	123 employee_id	ABC first_name	ABC last_name	ABC email	ABC phone_number	hire_date	ABC job_id	123 salary
1	100	Steven	King	SKING	515.123.4567	1987-06-17	AD_PRES	24 000
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21	AD_VP	17 000
3	102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13	AD_VP	17 000
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03	IT_PROG	9 000
5	104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21	IT_PROG	6 000
6	105	David	Austin	DAUSTIN	590.423.4569	1997-06-25	IT_PROG	4 800
7	106	Valli	Pataballa	VPATABAL	590.423.4560	1998-02-05	IT_PROG	4 800
8	107	Diana	Lorentz	DLORENTZ	590.423.5567	1999-02-07	IT_PROG	4 200
9	108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17	FI_MGR	12 000
10	109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-16	FI_ACCOUNT	9 000
11	110	John	Chen	JCHEN	515.124.4269	1997-09-28	FI_ACCOUNT	8 200
12	111	Ismael	Sciarra	ISCIARRA	515.124.4369	1997-09-30	FI_ACCOUNT	7 700
13	112	Jose Manuel	Urman	JMURMAN	515.124.4469	1998-03-07	FI_ACCOUNT	7 800
14	113	Luis	Popp	LPOPP	515.124.4567	1999-12-07	FI_ACCOUNT	6 900



	123 department_id	123 ?column?
1	60	11 700
2	60	7 800
3	60	6 240
4	60	6 240
5	60	5 460
6	100	11 700
7	100	10 660
8	100	10 010
9	100	10 140
10	100	8 970
11	30	4 030
12	30	3 770
13	30	3 640
14	30	3 380

## Последовательность выборки

```
select department_id, sum(salary * 1.3) new_salary
from employees
where salary < 10000
group by department_id
having sum(salary * 1.3) < 8000
order by new_salary desc;
```

	123 department_id	123 ?column?
1	60	11 700
2	60	7 800
3	60	6 240
4	60	6 240
5	60	5 460
6	100	11 700
7	100	10 660
8	100	10 010
9	100	10 140
10	100	8 070
11	30	4 030
12	30	3 770
13	30	3 640
14	30	3 380

## Последовательность выборки

```
select department_id, sum(salary * 1.3) new_salary
from employees
where salary < 10000
group by department_id
having sum(salary * 1.3) < 8000
order by new_salary desc
```

	123 department_id	123 ?column?
1	60	11 700
2	60	7 800
3	60	6 240
4	60	6 240
5	60	5 460
6	100	11 700
7	100	10 660
8	100	10 010
9	100	10 140
10	100	8 070
11	30	4 030
12	30	3 770
13	30	3 640
14	30	3 380

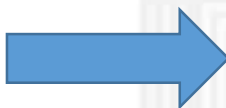


123 department_id	123 new_salary
20	7 800
50	203 320
30	18 070
110	10 790
100	51 480
[NULL]	9 100
80	234 650
60	37 440
10	5 720
40	8 450

## Последовательность выборки

```
select department_id, sum(salary * 1.3) new_salary
from employees
where salary < 10000
group by department_id
having sum(salary * 1.3) < 8000
order by new_salary desc
```

123 department_id	123 new_salary
20	7 800
50	203 320
30	18 070
110	10 790
100	51 480
[NULL]	9 100
80	234 650
60	37 440
10	5 720
40	8 450



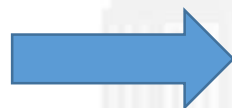
	123 department_id	123 new_salary
1	20	7 800
2	10	5 720



## Последовательность выборки

```
select department_id, sum(salary * 1.3) new_salary  
from employees  
where salary < 10000  
group by department_id  
having sum(salary * 1.3) < 8000  
order by new_salary desc
```

	123 department_id	123 new_salary
1	20	7 800
2	10	5 720



	123 department_id	123 new_salary
1	20	7 800
2	10	5 720

## Практическое занятие 4

4.1. Вывести самый высокий, самый низкий и средний оклад по всем служащим, а также суммы всех окладов. Назовите столбцы Maximum, Minimum, Average и Sum соответственно.

	123 Maximum ▼	123 Minimum ▼	123 Average ▼	123 Sum ▼
1	24 000	2 100	6 461,6822429907	691 400

## Практическое занятие 4

4. 2. Вывести самый высокий, самый низкий и средний оклад, а также суммы окладов по всем служащим по каждой должности. Назовите столбцы job\_id Maximum, Minimum, Average и Sum соответственно.

	ABC job_id ▼	123 Maximum ▼	123 Minimum ▼	123 Average ▼	123 Sum ▼
1	SH_CLERK	4 200	2 500	3 215	64 300
2	AD_VP	17 000	17 000	17 000	34 000
3	SA_MAN	14 000	10 500	12 200	61 000
4	PU_MAN	11 000	11 000	11 000	11 000
5	IT_PROG	9 000	4 200	5 760	28 800
6	ST_CLERK	3 600	2 100	2 785	55 700
7	FI_MGR	12 000	12 000	12 000	12 000
8	PU_CLERK	3 100	2 500	2 780	13 900
9	HR_REP	6 500	6 500	6 500	6 500
10	ST_MAN	8 200	5 800	7 280	36 400
11	MK_MAN	13 000	13 000	13 000	13 000
12	AC_MGR	12 000	12 000	12 000	12 000
13	SA_REP	11 500	6 100	8 350	250 500
14	AD_ASST	4 400	4 400	4 400	4 400
15	PR_REP	10 000	10 000	10 000	10 000
16	MK_REP	6 000	6 000	6 000	6 000
17	AD PRES	24 000	24 000	24 000	24 000
18	FI_ACCOUNT	9 000	6 900	7 920	39 600
19	AC_ACCOUNT	8 300	8 300	8 300	8 300

## Практическое занятие 4

4. 3. Напишите запрос для вывода должности и количества служащих, занимающих каждую должность. Назовите столбцы `job_id` и `count` соответственно.

	ABC job_id ▼	123 count ▼
1	SH_CLERK	20
2	AD_VP	2
3	SA_MAN	5
4	PU_MAN	1
5	IT_PROG	5
6	ST_CLERK	20
7	FI_MGR	1
8	PU_CLERK	5
9	HR_REP	1
10	ST_MAN	5
11	MK_MAN	1
12	AC_MGR	1
13	SA_REP	30
14	AD_ASST	1
15	PR_REP	1
16	MK_REP	1
17	AD PRES	1
18	FI_ACCOUNT	5
19	AC_ACCOUNT	1

## Практическое занятие 4

4. 4. Получите количество служащих, имеющих подчиненных, без их перечисления. Столбец назовите Number Of Managers

Подсказка: используйте столбец manager\_id

	123 Number Of Managers
1	18

## Практическое занятие 4

4. 5. Напишите запрос для вывода разности между самым высоким и самым низким окладами, назовите столбец difference

	123 difference
1	21 900

## Практическое занятие 4

4. 6.\* Напишите запрос для вывода номера каждого менеджера и заработной платы самого низкооплачиваемого из его подчиненных. Исключите всех, для которых неизвестных менеджеры. Исключите все группы, где минимальный оклад составляет менее 6000. Отсортируйте выходные строки в порядке убывания оклада. Столбцы назовите `manager_id` и `min_salary`.

	123 manager_id	123 min_salary
1	102	9 000
2	205	8 300
3	145	7 000
4	146	7 000
5	108	6 900
6	147	6 200
7	149	6 200
8	148	6 100
9	201	6 000

## Практическое занятие 4

4. 7.\* Напишите запрос для вывода общего количества служащих и количества служащих, нанятых в 1995, 1996, 1997 и 1998 годах. Столбцы назовите Total, 1995, 1996, 1997, 1998. Подсказка: для получения года из даты используйте функцию `extract('Year' from hire_date)`

	123 Total	123 1995	123 1996	123 1997	123 1998
1	107	4	10	28	23



## Практическое занятие 4

4. 8.\* Напишите матричный запрос для вывода всех должностей и суммы заработной платы служащих, работающих в этой должности в отделах 20, 50, 80 и 90. Последний столбец должен содержать сумму заработной платы служащих этих отделов, занимающих каждую конкретную должность. Дайте столбцам соответствующие заголовки. Столбцы назовите Job, Dept20, Dept50, Dept80, Dept90, Total.

	ABC Job ▼	123 Dept20 ▼	123 Dept50 ▼	123 Dept80 ▼	123 Dept90 ▼	123 Total ▼
1	SH_CLERK	0	64 300	0	0	64 300
2	AD_VP	0	0	0	34 000	34 000
3	SA_MAN	0	0	61 000	0	61 000
4	ST_CLERK	0	55 700	0	0	55 700
5	ST_MAN	0	36 400	0	0	36 400
6	MK_MAN	13 000	0	0	0	13 000
7	SA_REP	0	0	243 500	0	243 500
8	MK_REP	6 000	0	0	0	6 000
9	AD_PRES	0	0	0	24 000	24 000

## Занятие 5

# Выборка данных из нескольких таблиц

Соединение при помощи оператора join

Оператор using

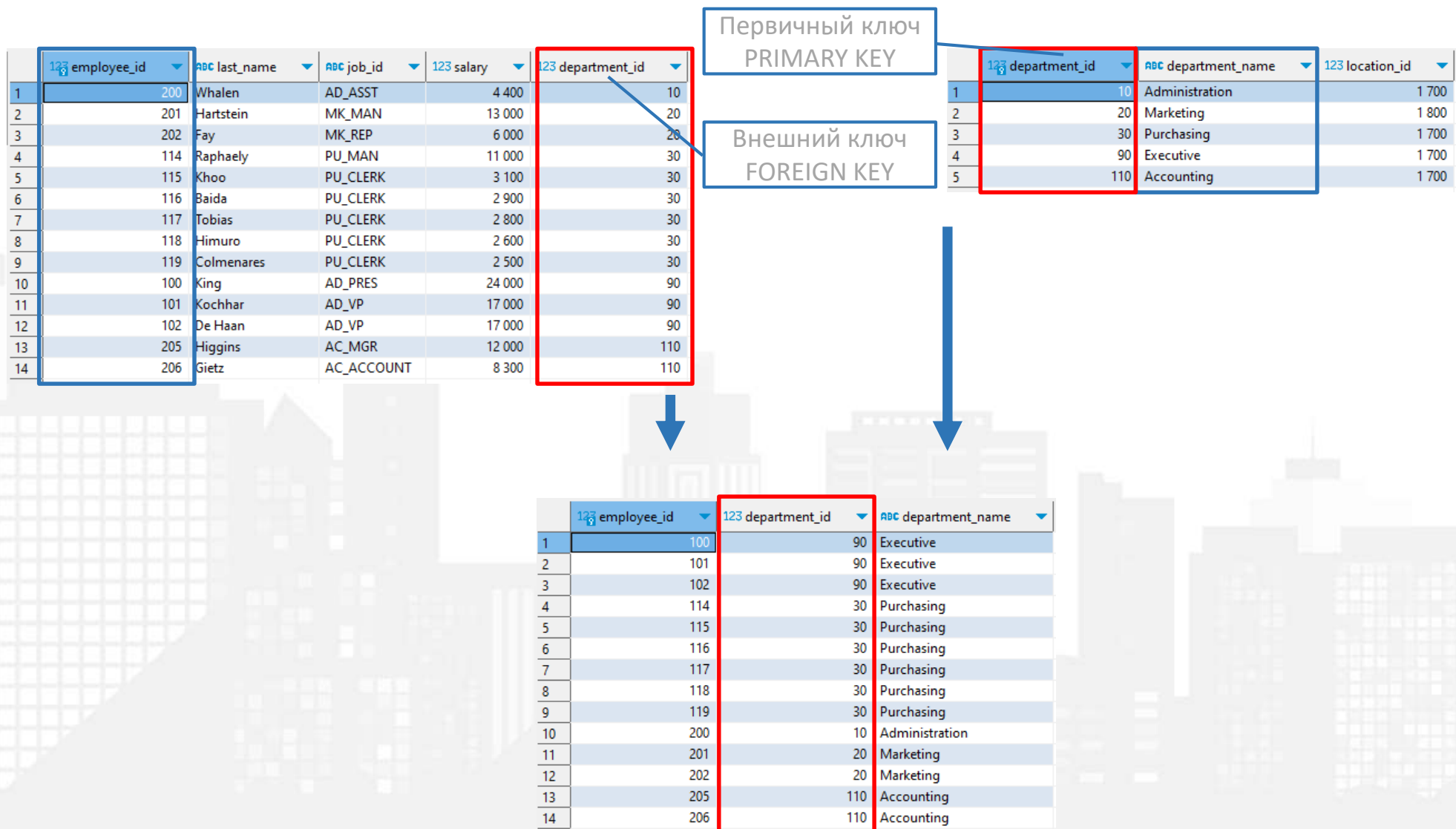
Внешние соединения

Кросс-соединение

## Общий вид запроса

```
SELECT <выражения для групповых столбцов>  
FROM <источник1>  
[INNER JOIN] | [LEFT|RIGHT|FULL OUTER JOIN] <источник2>  
[USING (<список_столбцов>) ] | [ON (<список_условий>)]  
WHERE <условия выборки>  
GROUP BY <столбцы группировки>  
HAVING <условия выборки сгруппированных данных>  
ORDER BY <столбцы сортировки>
```

## Суть объединения таблиц



## Виды соединений

INNER JOIN – внутреннее

LEFT OUTER JOIN – левое внешнее

RIGHT OUTER JOIN – правое внешнее

FULL OUTER JOIN – полное внешнее

CROSS JOIN – декартово произведение

## Внутреннее соединение

```
select *  
from employees  
inner join departments  
on (departments.department_id = employees.department_id)
```

## Внутреннее соединение с использованием предложения using

```
select *  
from employees  
inner join departments  
using (department_id)
```

## Использование псевдонимов таблиц

```
select e.employee_id, d.department_id, d.department_name  
from employees as e  
inner join departments as d  
on (e.department_id = d.department_id)
```

```
select e.employee_id, department_id, d.department_name  
from employees as e  
inner join departments as d  
using (department_id)
```



## Рефлексивное соединение (соединение таблицы с собой)

```
select e.last_name emp, m.last_name mgr  
from employees as e  
inner join employees as m  
on (e.manager_id = d.employee_id)
```

## Внешнее соединение

Что будет если отдел не имеет сотрудников или сотрудник не относится к какому-либо отделу?



## Внешнее левое соединение

Возвращаются все служащие, в том числе не относящиеся к какому-либо отделу

```
select *  
from employees  
left outer join departments using(department_id)
```

## Внешнее правое соединение

Возвращаются все отделы, в том числе не имеющие служащих

```
select * from employees  
right outer join departments using(department_id)
```

## Внешнее полное соединение

Возвращаются все служащие и отделы

```
select * from employees  
full outer join departments using(department_id)
```

## Перекрестное соединение

### Декартово произведение двух таблиц

```
select last_name, department_name  
from employees  
cross join departments
```

## Практическое занятие 5

5.1. Вывести адреса всех отделов. Используйте таблицы `locations` и `countries`. Вывести код локации (`location_id`), адрес (`street_address`), город (`city`), регион (`state_province`) и страну (`country_name`).

	location_id	street_address	city	state_province	country_name
1	1 300	9450 Kamiya-cho	Hiroshima	[NULL]	Japan
2	1 500	2011 Interiors Blvd	South San Francisco	California	United States of America
3	2 600	9702 Chester Road	Stretford	Manchester	United Kingdom
4	2 700	Schwanthalerstr. 7031	Munich	Bavaria	Germany
5	2 400	8204 Arthur St	London	[NULL]	United Kingdom
6	2 500	Magdalen Centre, The Oxford Science Park	Oxford	Oxford	United Kingdom
7	1 100	93091 Calle della Testa	Venice	[NULL]	Italy
8	1 400	2014 Jabberwocky Rd	Southlake	Texas	United States of America
9	1 600	2007 Zagora St	South Brunswick	New Jersey	United States of America
10	1 700	2004 Charade Rd	Seattle	Washington	United States of America
11	2 800	Rua Frei Caneca 1360	Sao Paulo	Sao Paulo	Brazil
12	1 200	2017 Shinjuku-ku	Tokyo	Tokyo Prefecture	Japan
13	1 900	6092 Boxwood St	Whitehorse	Yukon	Canada
14	2 900	20 Rue des Corps-Saints	Geneva	Geneve	Switzerland
15	1 000	1297 Via Cola di Rie	Roma	[NULL]	Italy
...					
21	2 100	1298 Vileparle (E)	Bombay	Maharashtra	India
22	3 100	Pieter Breughelstraat 837	Utrecht	Utrecht	Netherlands
23	2 200	12-98 Victoria Street	Sydney	New South Wales	Australia

## Практическое занятие 5


5.2. Вывести фамилию (`last_name`), номер отдела (`department_id`) и название отдела (`department_name`) для всех сотрудников.

	ABC last_name ▼	123 department_id ▼	ABC department_name ▼
1	King	90	Executive
2	Kochhar	90	Executive
3	De Haan	90	Executive
4	Hunold	60	IT
5	Ernst	60	IT
6	Austin	60	IT
7	Pataballa	60	IT
8	Lorentz	60	IT
9	Greenberg	100	Finance
10	Faviet	100	Finance
11	Chen	100	Finance
12	Sciarra	100	Finance
13	Urman	100	Finance
14	Popp	100	Finance
15	Raphaely	30	Purchasing
...			
104	Baer	70	Public Relations
105	Higgins	110	Accounting
106	Gietz	110	Accounting



## Практическое занятие 5

5.3. Напишите запрос для вывода фамилии, должности, номера отдела и названия отдела всех служащих, работающих в городе Toronto.



	ABC last_name ▼	ABC job_id ▼	123 department_id ▼	ABC department_name ▼
1	Hartstein	MK_MAN	20	Marketing
2	Fay	MK_REP	20	Marketing

## Практическое занятие 5

5.4. Выведите фамилии и номера служащих вместе с фамилиями и номерами их менеджеров. Назовите столбцы: Employee, Emp#, Manager, Mgr#.

	ABC Employee ▼	123 Emp# ▼	ABC Manager ▼	123 Mgr# ▼
1	Kochhar	101	King	100
2	De Haan	102	King	100
3	Hunold	103	De Haan	102
4	Ernst	104	Hunold	103
5	Austin	105	Hunold	103
6	Pataballa	106	Hunold	103
7	Lorentz	107	Hunold	103
8	Greenberg	108	Kochhar	101
9	Faviet	109	Greenberg	108
10	Chen	110	Greenberg	108
11	Sciarra	111	Greenberg	108
12	Urman	112	Greenberg	108
13	Popp	113	Greenberg	108
14	Raphaely	114	King	100
15	Khoo	115	Raphaely	114
...				
104	Baer	204	Kochhar	101
105	Higgins	205	Kochhar	101
106	Gietz	206	Higgins	205

## Практическое занятие 5

5.5. Измените запрос задания 4 так, чтобы в выборке были и сотрудники, не имеющие менеджера. Упорядочьте результат по возрастанию номера служащего.

	ABC Employee ▼	123 Emp# ▼	ABC Manager ▼	123 Mgr# ▼
1	King	100	[NULL]	[NULL]
2	Kochhar	101	King	100
3	De Haan	102	King	100
4	Hunold	103	De Haan	102
5	Ernst	104	Hunold	103
6	Austin	105	Hunold	103
7	Pataballa	106	Hunold	103
8	Lorentz	107	Hunold	103
9	Greenberg	108	Kochhar	101
10	Faviet	109	Greenberg	108
11	Chen	110	Greenberg	108
12	Sciarra	111	Greenberg	108
13	Urman	112	Greenberg	108
14	Popp	113	Greenberg	108
15	Raphaely	114	King	100
...				
105	Baer	204	Kochhar	101
106	Higgins	205	Kochhar	101
107	Gietz	206	Higgins	205

## Практическое занятие 5

5. 6. Вывести номер отдела, фамилию служащего и фамилии всех служащих, работающих в одном отделе с данным служащим. Назовите столбцы: department, employee, colleague.

	123 department ▼	ABC employee ▼	ABC colleague ▼
1	90	King	De Haan
2	90	King	Kochhar
3	90	Kochhar	De Haan
4	90	Kochhar	King
5	90	De Haan	Kochhar
6	90	De Haan	King
7	60	Hunold	Lorentz
8	60	Hunold	Pataballa
9	60	Hunold	Austin
10	60	Hunold	Ernst
11	60	Ernst	Lorentz
12	60	Ernst	Pataballa
13	60	Ernst	Austin
14	60	Ernst	Hunold
15	60	Austin	Lorentz
...			
3190	20	Fay	Hartstein
3191	110	Higgins	Gietz
3192	110	Gietz	Higgins

## Практическое занятие 5

5.7. Выведите фамилию, должность, название отдела, оклада и категорию (grade), столбцы назовите соответственно: last\_name, job\_id, department\_name, salary, gra.

	ABC last_name ▼	ABC job_id ▼	ABC department_name ▼	123 salary ▼	ABC gra ▼
1	Baida	PU_CLERK	Purchasing	2 900	A
2	Tobias	PU_CLERK	Purchasing	2 800	A
3	Himuro	PU_CLERK	Purchasing	2 600	A
4	Colmenares	PU_CLERK	Purchasing	2 500	A
5	Mikkilineni	ST_CLERK	Shipping	2 700	A
6	Landry	ST_CLERK	Shipping	2 400	A
7	Markle	ST_CLERK	Shipping	2 200	A
8	Atkinson	ST_CLERK	Shipping	2 800	A
9	Marlow	ST_CLERK	Shipping	2 500	A
10	Olson	ST_CLERK	Shipping	2 100	A
11	Rogers	ST_CLERK	Shipping	2 900	A
12	Gee	ST_CLERK	Shipping	2 400	A
13	Philtanker	ST_CLERK	Shipping	2 200	A
14	Seo	ST_CLERK	Shipping	2 700	A
15	Patel	ST_CLERK	Shipping	2 500	A
...					
104	King	AD_PRES	Executive	24 000	E
105	Kochhar	AD_VP	Executive	17 000	E
106	De Haan	AD_VP	Executive	17 000	E

## Практическое занятие 5

5.8\*. Вывести фамилию и дату найма служащих нанятых после Davies.

	ABC last_name	hire_date
1	Austin	1997-06-25
2	Pataballa	1998-02-05
3	Lorentz	1999-02-07
4	Chen	1997-09-28
5	Sciarra	1997-09-30
6	Urman	1998-03-07
7	Popp	1999-12-07
8	Baida	1997-12-24
9	Tobias	1997-07-24
10	Himuro	1998-11-15
11	Colmenares	1999-08-10
12	Fripp	1997-04-10
13	Vollman	1997-10-10
14	Mourgos	1999-11-16
15	Nayer	1997-07-16
...		
76	OConnell	1999-06-21
77	Grant	2000-01-13
78	Fay	1997-08-17

## Практическое занятие 5

5.9\*. Вывести сведения о служащих, нанятых раньше своих менеджеров. Нужно вывести фамилию, дату найма, а также фамилию и дату найма их менеджеров.

	ABC last_name ▼	🕒 hire_date ▼	ABC last_name ▼	🕒 hire_date ▼
1	Hunold	1990-01-03	De Haan	1993-01-13
2	Faviet	1994-08-16	Greenberg	1994-08-17
3	Marlow	1997-02-16	Fripp	1997-04-10
4	Ladwig	1995-07-14	Vollman	1997-10-10
5	Rajs	1995-10-17	Mourgos	1999-11-16
6	Davies	1997-01-29	Mourgos	1999-11-16
7	Matos	1998-03-15	Mourgos	1999-11-16
8	Vargas	1998-07-09	Mourgos	1999-11-16
9	King	1996-01-30	Partners	1997-01-05
10	Sully	1996-03-04	Partners	1997-01-05
11	McEwen	1996-08-01	Partners	1997-01-05
12	Ozer	1997-03-11	Cambrault	1999-10-15
13	Bloom	1998-03-23	Cambrault	1999-10-15
14	Fox	1998-01-24	Cambrault	1999-10-15
15	Smith	1999-02-23	Cambrault	1999-10-15
...				
28	Feeney	1998-05-23	Mourgos	1999-11-16
29	OConnell	1999-06-21	Mourgos	1999-11-16
30	Whalen	1987-09-17	Kochhar	1989-09-21

## Занятие 6

# Использование операторов над множествами

Описание операторов над множествами

Использование операторов над множествами для объединения запросов





## Общий шаблон запроса

SELECT *<выражения для столбцов>* ...

**UNION [ALL] | INTERSECT | EXCEPT**

SELECT *<выражения для столбцов>* ...

...

## Суть объединения таблиц при помощи JOIN

(горизонтальное соединение)

	123 employee_id	ABC last_name	ABC job_id	123 salary	123 department_id
1	200	Whalen	AD_ASST	4 400	10
2	201	Hartstein	MK_MAN	13 000	20
3	202	Fay	MK_REP	6 000	20
4	114	Raphaely	PU_MAN	11 000	30
5	115	Khoo	PU_CLERK	3 100	30
6	116	Baida	PU_CLERK	2 900	30
7	117	Tobias	PU_CLERK	2 800	30
8	118	Himuro	PU_CLERK	2 600	30
9	119	Colmenares	PU_CLERK	2 500	30
10	100	King	AD_PRES	24 000	90
11	101	Kochhar	AD_VP	17 000	90
12	102	De Haan	AD_VP	17 000	90
13	205	Higgins	AC_MGR	12 000	110
14	206	Gietz	AC_ACCOUNT	8 300	110



	123 department_id	ABC department_name	123 location_id
1	10	Administration	1 700
2	20	Marketing	1 800
3	30	Purchasing	1 700
4	90	Executive	1 700
5	110	Accounting	1 700

	123 employee_id	123 department_id	ABC department_name
1	100	90	Executive
2	101	90	Executive
3	102	90	Executive
4	114	30	Purchasing
5	115	30	Purchasing
6	116	30	Purchasing
7	117	30	Purchasing
8	118	30	Purchasing
9	119	30	Purchasing
10	200	10	Administration
11	201	20	Marketing
12	202	20	Marketing
13	205	110	Accounting
14	206	110	Accounting

## Суть объединения таблиц при помощи UNION

(вертикальное соединение)

	employee_id	last_name	job_id	salary	department_id
1	200	Whalen	AD_ASST	4 400	10
2	201	Hartstein	MK_MAN	13 000	20
3	202	Fay	MK_REP	6 000	20
4	114	Raphaely	PU_MAN	11 000	30
5	115	Khoo	PU_CLERK	3 100	30
6	116	Baida	PU_CLERK	2 900	30
7	117	Tobias	PU_CLERK	2 800	30

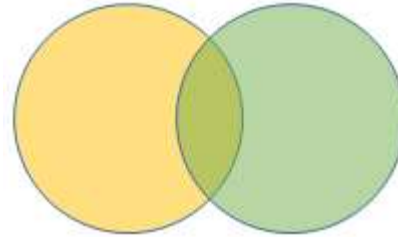


	employee_id	last_name	job_id	salary	department_id
8	118	Himuro	PU_CLERK	2 600	30
9	119	Colmenares	PU_CLERK	2 500	30
10	100	King	AD PRES	24 000	90
11	101	Kochhar	AD_VP	17 000	90
12	102	De Haan	AD_VP	17 000	90
13	205	Higgins	AC_MGR	12 000	110
14	206	Gietz	AC_ACCOUNT	8 300	110

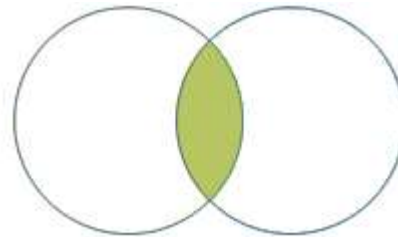
	employee_id	last_name	job_id	salary	department_id
1	200	Whalen	AD_ASST	4 400	10
2	201	Hartstein	MK_MAN	13 000	20
3	202	Fay	MK_REP	6 000	20
4	114	Raphaely	PU_MAN	11 000	30
5	115	Khoo	PU_CLERK	3 100	30
6	116	Baida	PU_CLERK	2 900	30
7	117	Tobias	PU_CLERK	2 800	30
8	118	Himuro	PU_CLERK	2 600	30
9	119	Colmenares	PU_CLERK	2 500	30
10	100	King	AD PRES	24 000	90
11	101	Kochhar	AD_VP	17 000	90
12	102	De Haan	AD_VP	17 000	90
13	205	Higgins	AC_MGR	12 000	110
14	206	Gietz	AC_ACCOUNT	8 300	110

## Операции над множествами

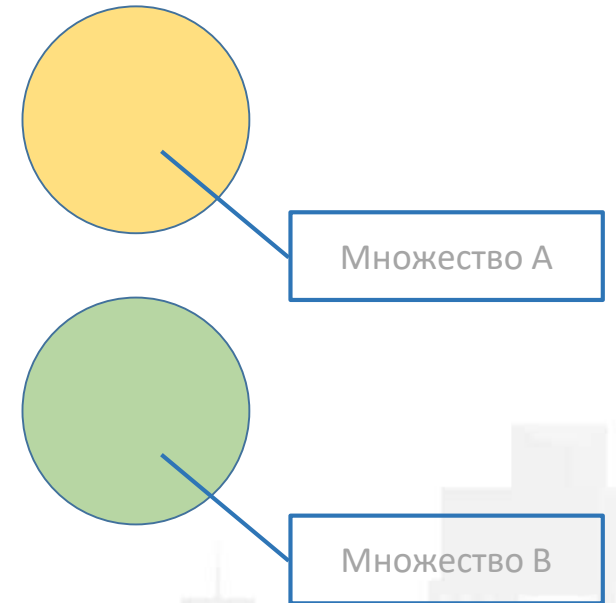
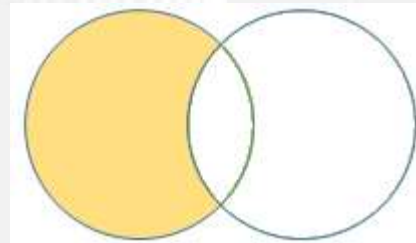
**UNION [ALL]**



**INTERSECT [ALL]**



**EXCEPT [ALL]**



## Используемые в занятии таблицы

EMPLOYEES, JOB\_HISTORY



## Использование оператора UNION

Вывод текущей и предыдущих должностей всех сотрудников.  
Выходные строки не повторяются.

```
SELECT employee_id, job_id FROM employees
UNION
SELECT employee_id, job_id FROM job_history;
```

## Использование оператора UNION ALL

Вывод текущей и предыдущих должностей всех сотрудников.  
Выходные строки могут повторяться.

```
SELECT employee_id, job_id, department_id  
FROM employees
```

### **UNION ALL**

```
SELECT employee_id, job_id , department_id  
FROM job_history;
```

## Использование оператора INTERSECT

Вывод номеров сотрудников и их должностей, если текущие должности совпадают с должностями, которые они уже занимали.

```
SELECT employee_id, job_id FROM employees
```

**INTERSECT**

```
SELECT employee_id, job_id FROM job_history;
```



## Операторы над множествами: указания

- Количество выражений в списках SELECT и их типы данных должны совпадать;
- Предложение ORDER BY может присутствовать только в самом конце всей команды;
- Команды с ALL работают быстрее, т.к. не требуется дополнительных ресурсов для обеспечения уникальности выходных данных;

## Согласование команд SELECT

При необходимости можно использовать фиктивные столбцы. Для всех сотрудников вывести их номер, идентификаторы всех их должностей и оклад.

```
SELECT employee_id, job_id, salary FROM employees
UNION
SELECT employee_id, job_id, 0 FROM job_history;
```

## Сортировка

```
SELECT employee_id, job_id, salary
FROM employees
UNION
SELECT employee_id, job_id , 0
FROM job_history
ORDER BY employee_id, salary desc;
```

## Практическое занятие 6

6.1. Вывести список номеров отделов, в которых нет сотрудников с идентификатором должности ST\_CLERK. Используйте операторы над множествами.

	123 department_id
1	210
2	60
3	120
4	180
5	130
6	30
7	220
8	80
9	260
10	90
11	190
12	10
13	40
14	250
15	160
...	
24	140
25	100
26	70

## Практическое занятие 6

6.2. Вывести идентификаторы и названия стран, в которых не располагаются подразделения компании.

	ABC country_id	ABC country_name
1	MX	Mexico
2	KW	Kuwait
3	JP	Japan
4	CN	China
5	CH	Switzerland
6	HK	HongKong
7	IN	India
8	AR	Argentina
9	AU	Australia
10	NL	Netherlands
11	DK	Denmark
12	NG	Nigeria
13	IT	Italy
14	SG	Singapore
15	BR	Brazil
...		
19	EG	Egypt
20	ZM	Zambia
21	ZW	Zimbabwe

## Практическое занятие 6

6.3. Используя операторы над множествами, вывести список должностей отделов 10, 50 и 20 в таком же порядке отделов. Включите в результат столбцы `job_id` и `department_id`.

	ABC job_id ▼	123 department_id ▼
1	AD_ASST	10
2	ST_CLERK	50
3	ST_MAN	50
4	SH_CLERK	50
5	MK_MAN	20
6	MK_REP	20

## Практическое занятие 6

6.4. Выведите номер сотрудника и идентификатор его должности, если его текущая должность совпадает с той, которую он уже занимал, работая в компании.

	123 employee_id	ABC job_id
1	200	AD_ASST
2	176	SA_REP

## Практическое занятие 6

6.5. Вывести в одном запросе:

- фамилии и отделы всех сотрудников из таблицы EMPLOYEES, независимо от того, относятся ли они к какому-либо отделу,
- номера и наименования всех отделов из таблицы DEPARTMENTS, независимо от того, есть ли в них сотрудники или нет.

	ABC last_name	123 department_id	ABC department_name
1	King	90	
2	Kochhar	90	
3	De Haan	90	
...			
107	Gietz	110	
108		10	Administration
109		20	Marketing
...			
132		250	Retail Sales
133		260	Recruiting
134		270	Payroll



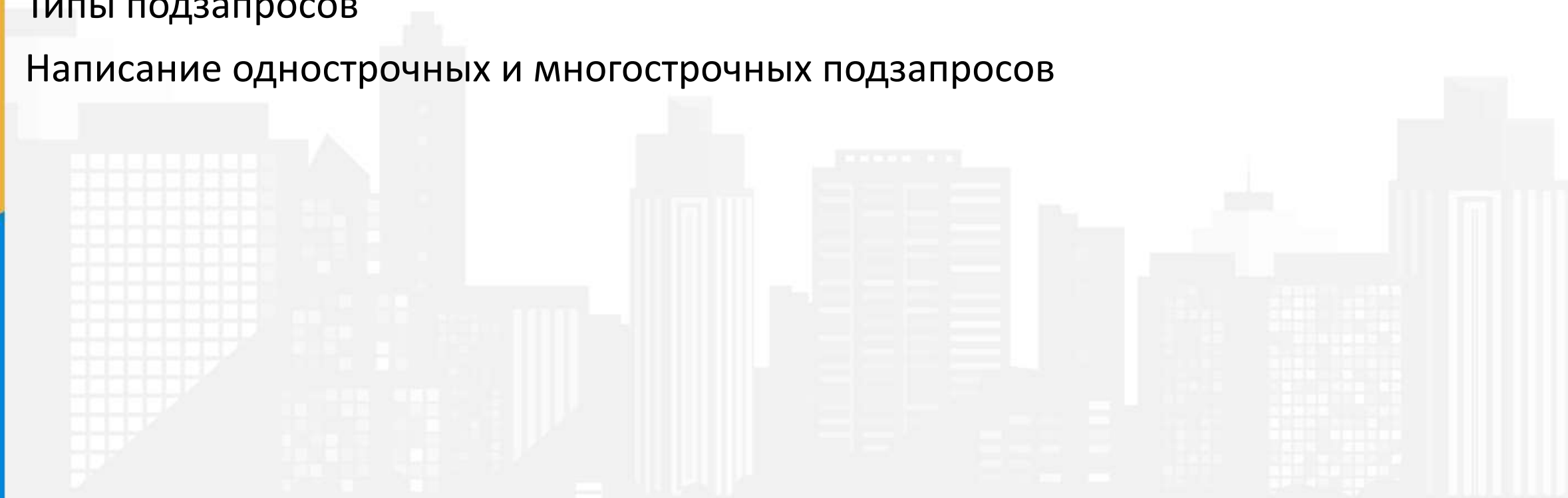
## Занятие 7

# Использование подзапросов

Определение подзапросов

Типы подзапросов

Написание однострочных и многострочных подзапросов



## Задача

У кого оклад больше, чем у Абеля?

Главный вопрос: у кого из служащих оклад больше, чем у Абеля?

Вспомогательный вопрос: каков оклад у Абеля?

## Ответ на вспомогательный вопрос

Главный вопрос: у кого из служащих оклад больше, чем у Абе́ля?

**Вспомогательный вопрос: каков оклад у Абе́ля?**

```
select salary from employees where last_name = 'Abel';
```

## Ответ на главный вопрос

**Главный вопрос: у кого из служащих оклад больше, чем у Абеля?**

```
select last_name, salary from employees  
where salary > Abel_salary
```

**Вспомогательный вопрос: каков оклад у Абеля?**

```
select salary from employees where last_name = 'Abel';
```

## Решение с использованием подзапроса

У кого оклад больше, чем у Абеля?

```
select last_name, salary from employees  
where salary > (select salary  
                from employees  
                where last_name = 'Abel');
```

## Однострочные (скалярные) подзапросы

Возвращают единственное значение, можно использовать везде, где используется значения, в том числе с операциями:

`=, !=, >, >=, <, <=`

```
select last_name, salary from employees
where salary >= (select salary
                  from employees
                  where last_name = 'Abel');
```

## Несколько подзапросов

```
select last_name, job_id, salary
from employees
where job_id = (select job_id
                from employees
                where employee_id = 141)
and salary > (select salary
              from employees
              where employee_id = 143);
```

## Групповая функция в подзапросе

Вывести служащих, получающих оклад равный минимальному окладу отдела 50

```
select last_name, job_id, salary, department_id
from employees
where salary = (select min(salary)
                from employees
                where department_id = 50);
```



## Подзапрос в предложении HAVING

Вывести отделы, минимальные оклады в которых превышают минимальный оклад в отделе 50

```
select department_id, min(salary)
from employees
group by department_id
having min(salary) > (select min(salary)
                      from employees
                      where department_id = 50)
```

## подзапрос в предложении SELECT

Просто отобразить в отдельном столбце минимальный оклад в отделе 50

```
select department_id, min(salary), (select min(salary)  
from employees where department_id = 50) as min_salary50  
from employees  
group by department_id  
having min(salary) > (select min(salary)  
                        from employees  
                        where department_id = 50)
```

## Возможные ошибки в однострочных подзапросах

1. Подзапрос возвращает несколько строк
2. Подзапрос не возвращает значений

## Использование значений из главного запроса

Вывести фамилию, должность, оклад, отдел и максимальный оклад в отделе, в котором он работает

```
select last_name, job_id, salary, department_id,  
(select max(salary) from employees where department_id =  
e.department_id) maxSalary  
from employees as e;
```

Вывести фамилию, должность, оклад сотрудников, у которых оклад ниже среднего по отделу

```
select last_name, job_id, salary, department_id  
from employees as e  
where salary < (select avg(salary) from employees where  
department_id = e.department_id);
```

## Многострочные подзапросы

Возвращают несколько строк, можно использовать везде, где используется список значений, в том числе с операциями:

`in, not in, any, all.`

Вывести фамилии служащих, не имеющих подчиненных

```
select last_name  
from employees  
where employee_id not in (select manager_id  
                           from employees  
                           where manager_id is not null)
```

## Значения null в подзапросе

Если в предыдущем примере в подзапросе убрать условие `where manager_id is not null`, то запрос не вернет ни одной строки.

```
select last_name  
from employees  
where employee_id not in (select manager_id  
                           from employees  
                           where manager_id is not null)
```

## Многострочные подзапросы

Пример с оператором ANY

Вывести служащих, у которых оклад меньше, чем у какого-либо служащего с должностью IT\_PROG

```
select last_name, job_id, salary, department_id
from employees
where job_id != 'IT_PROG'
      and salary < any (select salary from employees where job_id =
'IT_PROG');
```

## Многостолбцовые подзапросы

Возвращают результат с несколькими столбцами. Могут использоваться, если требуется сравнение значений двух и более столбцов.

Вывести сведения о сотрудниках, которые подчиняются такому же менеджеру и работают в том же отделе, что и сотрудники с номерами 199 или 174.

```
select employee_id, manager_id, department_id
from employees
where (manager_id, department_id) in
      (select manager_id , department_id
       from employees
       where employee_id in (199, 174))
and employee_id not in (199, 174);
```



## Использование оператора EXISTS

Проверяет существование строк в наборе результатов подзапроса

Поиск служащих, которым подчиняется хотя бы один служащий

```
select employee_id, last_name, job_id, department_id
from employees as e
where exists (select 'x'
              from employees
              where manager_id = e.employee_id);
```

## Использование оператора NOT EXISTS

Поиск отделов, в которых нет служащих

```
select department_id, department_name  
from departments as d  
Where not exists (select 'x'  
                    from employees  
                    where department_id = d.department_id);
```

## Использование подзапросов в качестве источников данных

```
select *  
from  
(  
    select employee_id as id, first_name as "firstName",  
           last_name as "lastName", job_id as "jobId",  
           department_id as "departmentId"  
    from employees  
) as e  
where e."departmentId" in (90, 100);
```

## Практическое занятие 7

7.1. Вывести фамилии и даты найма служащих, работающих в одном отделе вместе с Zlotkey.

	ABC last_name ▼	🕒 hire_date ▼
1	Russell	1996-10-01
2	Partners	1997-01-05
3	Errazuriz	1997-03-10
4	Cambrault	1999-10-15
5	Zlotkey	2000-01-29
6	Tucker	1997-01-30
7	Bernstein	1997-03-24
8	Hall	1997-08-20
9	Olsen	1998-03-30
10	Cambrault	1998-12-09
11	Tuvault	1999-11-23
12	King	1996-01-30
13	Sully	1996-03-04
14	McEwen	1996-08-01
15	Smith	1997-03-10
...		
32	Taylor	1998-03-24
33	Livingston	1998-04-23
34	Johnson	2000-01-04

## Практическое занятие 7

7.2. Вывести номера, фамилии и оклады, оклад которых выше среднего. Отсортируйте данные в порядке увеличения окладов.

	123 employee_id ▼	ABC last_name ▼	123 salary ▼
1	203	Mavris	6 500
2	123	Vollman	6 500
3	165	Lee	6 800
4	113	Popp	6 900
5	155	Tuvault	7 000
6	178	Grant	7 000
7	161	Sewall	7 000
8	164	Marvins	7 200
9	172	Bates	7 300
10	171	Smith	7 400
11	154	Cambrault	7 500
12	160	Doran	7 500
13	111	Sciarra	7 700
14	112	Urman	7 800
15	122	Kauffling	7 900
...			
49	102	De Haan	17 000
50	101	Kochhar	17 000
51	100	King	24 000

## Практическое занятие 7

7.3. Вывести номера, фамилии всех служащих, работающих в одном отделе с любым служащим, фамилия которого содержит букву и.

	123 employee_id	ABC last_name
1	103	Hunold
2	104	Ernst
3	105	Austin
4	106	Pataballa
5	107	Lorentz
6	114	Raphaely
7	115	Khoo
8	116	Baida
9	117	Tobias
10	118	Himuro
11	119	Colmenares
12	120	Weiss
13	121	Fripp
14	122	Kaufling
15	123	Vollman
...		
88	197	Feeney
89	198	OConnell
90	199	Grant

## Практическое занятие 7

7.4. Вывести фамилии, номера отделов и должности служащих, идентификатор местоположения которых равен 1700.

	ABC last_name ▼	123 department_id ▼	ABC job_id ▼
1	King	90	AD_PRES
2	Kochhar	90	AD_VP
3	De Haan	90	AD_VP
4	Greenberg	100	FI_MGR
5	Faviet	100	FI_ACCOUNT
6	Chen	100	FI_ACCOUNT
7	Sciarra	100	FI_ACCOUNT
8	Urman	100	FI_ACCOUNT
9	Popp	100	FI_ACCOUNT
10	Raphaely	30	PU_MAN
11	Khoo	30	PU_CLERK
12	Baida	30	PU_CLERK
13	Tobias	30	PU_CLERK
14	Himuro	30	PU_CLERK
15	Colmenares	30	PU_CLERK
16	Whalen	10	AD_ASST
17	Higgins	110	AC_MGR
18	Gietz	110	AC_ACCOUNT

## Практическое занятие 7

7.5. Вывести фамилии и оклады всех служащих подчиненных King.


	ABC last_name ▼	123 salary ▼
1	Kochhar	17 000
2	De Haan	17 000
3	Raphaely	11 000
4	Weiss	8 000
5	Fripp	8 200
6	Kaufling	7 900
7	Vollman	6 500
8	Mourgos	5 800
9	Russell	14 000
10	Partners	13 500
11	Errazuriz	12 000
12	Cambrault	11 000
13	Zlotkey	10 500
14	Hartstein	13 000



## Практическое занятие 7

7.6. Вывести номер отдела, фамилию и должность для служащих, работающих в администрации

(department\_name = 'Executive')



	123 department_id ▼	ABC last_name ▼	ABC job_id ▼
1	90	King	AD_PRES
2	90	Kochhar	AD_VP
3	90	De Haan	AD_VP

## Практическое занятие 7

7.7\*. Выведите номера, фамилии и оклады всех служащих при условии, что оклады превышают средний и служащие работают в одном отделе с любым служащим, у которого фамилия содержит букву u.

	123 employee_id	ABC last_name	123 salary
1	103	Hunold	9 000
2	114	Raphaely	11 000
3	120	Weiss	8 000
4	121	Fripp	8 200
5	122	Kauffling	7 900
6	123	Vollman	6 500
7	145	Russell	14 000
8	146	Partners	13 500
9	147	Errazuriz	12 000
10	148	Cambrault	11 000
11	149	Zlotkey	10 500
12	150	Tucker	10 000
13	151	Bernstein	9 500
14	152	Hall	9 000
15	153	Olsen	8 000
...			
34	175	Hutton	8 800
35	176	Taylor	8 600
36	177	Livingston	8 400

## Список литературы

ORACLE. Основы SQL. Руководство слушателя.

<https://metanit.com/sql/postgresql>

<https://habr.com>

