

# Урок 3. SQL – выборка данных, сортировка, агрегатные функции

## ▼ Задание (первая часть):

1. Напишите запрос, который вывел бы таблицу со столбцами в следующем порядке: city, sname, snum, comm. (к первой или второй таблице, используя SELECT)
2. Напишите команду SELECT, которая вывела бы оценку(rating), сопровождаемую именем каждого заказчика в городе San Jose. ("заказчики")
3. Напишите запрос, который вывел бы значения snum всех продавцов из таблицы заказов без каких бы то ни было повторений. (уникальные значения в "snum" "Продавцы")
- 4\*. Напишите запрос, который бы выбирал заказчиков, чьи имена начинаются с буквы G. Используется оператор "LIKE": ("заказчики") <https://dev.mysql.com/doc/refman/8.0/en/string-comparison-functions.html>
5. Напишите запрос, который может дать вам все заказы со значениями суммы выше чем \$1,000. ("Заказы", "amt" - сумма)
6. Напишите запрос который выбрал бы наименьшую сумму заказа.  
(Из поля "amt" - сумма в таблице "Заказы" выбрать наименьшее значение)
7. Напишите запрос к таблице "Заказчики", который может показать всех заказчиков, у которых рейтинг больше 100 и они находятся не в Риме.

## ▼ Таблицы для домашнего задания (первая часть)

===== ТАБЛИЦА 1: ПРОДАВЦЫ (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	.10

===== ТАБЛИЦА 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

===== ТАБЛИЦА 3: ЗАКАЗЫ (ORDERS) =====

onum	amt	odate	cnum	snum
3001	18.69	10/03/1990	2008	1007
3003	767.19	10/03/1990	2001	1001
3002	1900.10	10/03/1990	2007	1004
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	4723.00	10/05/1990	2006	1001
3010	1309.95	10/06/1990	2004	1002
3011	9891.88	10/06/1990	2006	1001

#### ▼ Задание (часть вторая)

1. Отсортируйте поле “зарплата” в порядке убывания и возрастания
2. \* Отсортируйте по возрастанию поле “Зарплата” и выведите 5 строк с наибольшей заработной платой (возможен подзапрос)
3. Выполните группировку всех сотрудников по специальности , суммарная зарплата которых превышает 100000

#### ▼ Таблица для домашнего задания (часть вторая)

+ Параметры

	id	name	surname	specialty	seniority	salary	age
<input type="checkbox"/>	1	Вася	Васькин	начальник	40	100000	60
<input type="checkbox"/>	2	Петя	Петькин	начальник	8	70000	30
<input type="checkbox"/>	3	Катя	Каткина	инженер	2	70000	25
<input type="checkbox"/>	4	Саша	Сашкин	инженер	12	50000	35
<input type="checkbox"/>	5	Иван	Иванов	рабочий	40	30000	59
<input type="checkbox"/>	6	Петр	Петров	рабочий	20	25000	40
<input type="checkbox"/>	7	Сидор	Сидоров	рабочий	10	20000	35
<input type="checkbox"/>	8	Антон	Антонов	рабочий	8	19000	28
<input type="checkbox"/>	9	Юра	Юркин	рабочий	5	15000	25
<input type="checkbox"/>	10	Максим	Воронин	рабочий	2	11000	22
<input type="checkbox"/>	11	Юра	Галкин	рабочий	3	12000	24
<input type="checkbox"/>	12	Люся	Люськина	уборщик	10	10000	49

↑ ☐ Отметить все С отмеченными:

## Решение (часть первая):

Подготовка таблиц:

```
CREATE DATABASE db_less3;
USE db_less3;

CREATE TABLE salespeople(
    snum INTEGER PRIMARY KEY UNIQUE NOT Null,
    sname VARCHAR(12) NOT Null,
    city VARCHAR(24),
    comm DECIMAL (3, 2)
);
INSERT INTO salespeople (snum, sname, city, comm) VALUES
(1001, 'Peel', 'London', 0.12),
(1002, 'Serres', 'San Jose', 0.13),
(1004, 'Motika', 'London', 0.11),
(1007, 'Rifkin', 'Barcelona', 0.15),
(1003, 'Axelrod', 'New York', 0.10);
```

```
SELECT * FROM salespeople;
```

Результат:

	snum	sname	city	comm
▶	1001	Peel	London	0.12
	1002	Serres	San Jose	0.13
	1003	Axelrod	New York	0.10
	1004	Motika	London	0.11
	1007	Rifkin	Barcelona	0.15
*	NULL	NULL	NULL	NULL

```
CREATE TABLE customers(
  cnum INTEGER PRIMARY KEY UNIQUE NOT Null,
  cname VARCHAR(12) NOT Null,
  city VARCHAR(24),
  rating INTEGER,
  snum INTEGER NOT Null,
  FOREIGN KEY (snum) REFERENCES salespeople (snum)
);
INSERT INTO customers (cnum, cname, city, rating, snum) VALUES
  (2001, 'Hoffman', 'London', 100, 1001),
  (2002, 'Giovanni', 'Rome', 200, 1003),
  (2003, 'Liu', 'San Jose', 200, 1002),
  (2004, 'Grass', 'Berlin', 300, 1002),
  (2006, 'Clemens', 'London', 100, 1001),
  (2008, 'Cisneros', 'San Jose', 300, 1007),
  (2007, 'Pereira', 'Rome', 100, 1004);

SELECT * FROM customers;
```

Результат:

	cnum	cname	city	rating	snum
▶	2001	Hoffman	London	100	1001
	2002	Giovanni	Rome	200	1003
	2003	Liu	San Jose	200	1002
	2004	Grass	Berlin	300	1002
	2006	Clemens	London	100	1001
	2007	Pereira	Rome	100	1004
	2008	Cisneros	San Jose	300	1007
*	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE orders(
  onum INTEGER PRIMARY KEY UNIQUE NOT Null,
  amt DECIMAL(7, 2),
  odate DATE,
  cnum INTEGER NOT Null,
  snum INTEGER NOT Null,
  FOREIGN KEY (snum) REFERENCES salespeople (snum),
  FOREIGN KEY (cnum) REFERENCES customers (cnum)
);
INSERT INTO orders (onum, amt, odate, cnum, snum) VALUES
  (3001, 18.69, '1990-03-10', 2008, 1007),
  (3003, 767.19, '1990-03-10', 2001, 1001),
  (3002, 1900.10, '1990-03-10', 2007, 1004),
  (3005, 5160.45, '1990-03-10', 2003, 1002),
  (3006, 1098.16, '1990-03-10', 2008, 1007),
  (3009, 1713.23, '1990-04-10', 2002, 1003),
```

```
(3007, 75.75, '1990-04-10', 2004, 1002),
(3008, 4723.95, '1990-05-10', 2006, 1001),
(3010, 1309.95, '1990-06-10', 2004, 1002),
(3011, 9891.88, '1990-06-10', 2006, 1001);
```

```
SELECT onum, amt, DATE_FORMAT(odate, '%d/%m/%Y') AS odate, cnum, snum FROM orders;
```

Результат:

	onum	amt	odate	cnum	snum
▶	3001	18.69	10/03/1990	2008	1007
	3002	1900.10	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
	3007	75.75	10/04/1990	2004	1002
	3008	4723.95	10/05/1990	2006	1001
	3009	1713.23	10/04/1990	2002	1003
	3010	1309.95	10/06/1990	2004	1002
	3011	9891.88	10/06/1990	2006	1001

**1. Напишите запрос, который вывел бы таблицу со столбцами в следующем порядке: city, sname, snum, comm. (к первой или второй таблице, используя SELECT)**

```
SELECT city, sname, snum, comm FROM salespeople;
```

Результат:

	city	sname	snum	comm
▶	London	Peel	1001	0.12
	San Jose	Serres	1002	0.13
	New York	Axelrod	1003	0.10
	London	Motika	1004	0.11
	Barcelona	Rifkin	1007	0.15
*	NULL	NULL	NULL	NULL

**2. Напишите команду SELECT, которая вывела бы оценку(rating), сопровождаемую именем каждого заказчика в городе San Jose. (“заказчики”)**

```
SELECT rating, cname FROM customers WHERE city = 'San Jose';
```

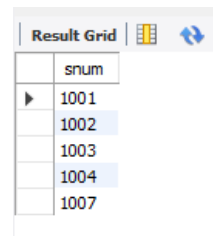
Результат:

	rating	cname
▶	200	Liu
	300	Cisneros

**3. Напишите запрос, который вывел бы значения snum всех продавцов из таблицы заказов без каких бы то ни было повторений. (уникальные значения в “snum” “Продавцы”)**

```
SELECT DISTINCT snum FROM orders;
```

Результат:




snum
1001
1002
1003
1004
1007

**4. Напишите запрос, который бы выбирал заказчиков, чьи имена начинаются с буквы G. Используется оператор "LIKE": (“заказчики”)**

```
SELECT cname FROM customers WHERE cname LIKE 'G%';
```

Результат:

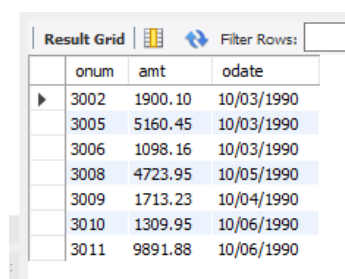


cname
Giovanni
Grass

**5. Напишите запрос, который может дать вам все заказы со значениями суммы выше чем \$1,000. (“Заказы”, “amt” - сумма)**

```
SELECT onum, amt, DATE_FORMAT(odate, '%d/%m/%Y') AS odate  
FROM orders WHERE amt > 1000;
```

Результат:



onum	amt	odate
3002	1900.10	10/03/1990
3005	5160.45	10/03/1990
3006	1098.16	10/03/1990
3008	4723.95	10/05/1990
3009	1713.23	10/04/1990
3010	1309.95	10/06/1990
3011	9891.88	10/06/1990

**6. Напишите запрос который выбрал бы наименьшую сумму заказа.**

(Из поля “amt” - сумма в таблице “Заказы” выбрать наименьшее значение)

```
SELECT MIN(amt) FROM orders;
```

Результат:

	MIN(amt)
	18.69

**7. Напишите запрос к таблице “Заказчики”, который может показать всех заказчиков, у которых рейтинг больше 100 и они находятся не в Риме.**

```
SELECT rating, cname, city FROM customers
WHERE (NOT city = 'Rome') AND (rating > 100);
```

Результат:

	rating	cname	city
	200	Liu	San Jose
	300	Grass	Berlin
	300	Cisneros	San Jose

## Решение (часть вторая):

Подготовка таблицы:

```
USE db_less3;
CREATE TABLE sotrudniki (
    id SERIAL PRIMARY KEY,
    name VARCHAR (12) NOT Null,
    surname VARCHAR (12) NOT Null,
    speciality VARCHAR (12),
    seniority INTEGER,
    salary DECIMAL (8),
    age INTEGER
);

INSERT INTO sotrudniki (name, surname, speciality, seniority, salary, age) VALUES
    ('Вася', 'Васькин', 'начальник', 40, 100000, 60),
    ('Петя', 'Петькин', 'начальник', 8, 70000, 30),
    ('Катя', 'Каткина', 'инженер', 2, 70000, 25),
    ('Саша', 'Сашкин', 'инженер', 12, 50000, 35),
    ('Иван', 'Иванов', 'рабочий', 40, 30000, 59),
    ('Петр', 'Петров', 'рабочий', 20, 25000, 40),
    ('Сидор', 'Сидоров', 'рабочий', 10, 20000, 35),
    ('Антон', 'Антонов', 'рабочий', 8, 19000, 28),
    ('Юра', 'Юркин', 'рабочий', 5, 15000, 25),
    ('Максим', 'Воронин', 'рабочий', 2, 11000, 22),
    ('Юра', 'Галкин', 'рабочий', 3, 12000, 24),
    ('Люся', 'Люськина', 'уборщик', 10, 10000, 49);

SELECT * FROM sotrudniki;
```

Результат:

Result Grid							
Filter Rows:							
	id	name	surname	speciality	seniority	salary	age
▶	1	Вася	Васькин	начальник	40	100000	60
	2	Петя	Петькин	начальник	8	70000	30
	3	Катя	Каткина	инженер	2	70000	25
	4	Саша	Сашкин	инженер	12	50000	35
	5	Иван	Иванов	рабочий	40	30000	59
	6	Петр	Петров	рабочий	20	25000	40
	7	Сидор	Сидоров	рабочий	10	20000	35
	8	Антон	Антонов	рабочий	8	19000	28
	9	Юра	Юркин	рабочий	5	15000	25
	10	Максим	Воронин	рабочий	2	11000	22
	11	Юра	Галкин	рабочий	3	12000	24
	12	Люся	Люськина	уборщик	10	10000	49
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## 1. Отсортируйте поле “зарплата” в порядке убывания и возрастания

```
SELECT * FROM sotrudniki ORDER BY salary;
```

Результат:

(в порядке возрастания)

Result Grid							
Filter Rows:							
	id	name	surname	speciality	seniority	salary	age
▶	12	Люся	Люськина	уборщик	10	10000	49
	10	Максим	Воронин	рабочий	2	11000	22
	11	Юра	Галкин	рабочий	3	12000	24
	9	Юра	Юркин	рабочий	5	15000	25
	8	Антон	Антонов	рабочий	8	19000	28
	7	Сидор	Сидоров	рабочий	10	20000	35
	6	Петр	Петров	рабочий	20	25000	40
	5	Иван	Иванов	рабочий	40	30000	59
	4	Саша	Сашкин	инженер	12	50000	35
	2	Петя	Петькин	начальник	8	70000	30
	3	Катя	Каткина	инженер	2	70000	25
	1	Вася	Васькин	начальник	40	100000	60
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT * FROM sotrudniki ORDER BY salary DESC;
```

Результат:

(в порядке убывания)

Result Grid							
Filter Rows:							
	id	name	surname	speciality	seniority	salary	age
▶	1	Вася	Васькин	начальник	40	100000	60
	2	Петя	Петькин	начальник	8	70000	30
	3	Катя	Каткина	инженер	2	70000	25
	4	Саша	Сашкин	инженер	12	50000	35
	5	Иван	Иванов	рабочий	40	30000	59
	6	Петр	Петров	рабочий	20	25000	40
	7	Сидор	Сидоров	рабочий	10	20000	35
	8	Антон	Антонов	рабочий	8	19000	28
	9	Юра	Юркин	рабочий	5	15000	25
	11	Юра	Галкин	рабочий	3	12000	24
	10	Максим	Воронин	рабочий	2	11000	22
	12	Люся	Люськина	уборщик	10	10000	49
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**2\*. Отсортируйте по возрастанию поле “Зарплата” и выведите 5 строк с наибольшей заработной платой (возможен подзапрос)**

```
SELECT * FROM (SELECT * FROM sotrudniki ORDER BY salary DESC LIMIT 5) q ORDER BY q.salary;
```

Результат:

Result Grid							
Filter Rows:							
	id	name	surname	speciality	seniority	salary	age
▶	5	Иван	Иванов	рабочий	40	30000	59
	4	Саша	Сашкин	инженер	12	50000	35
	2	Петя	Петькин	начальник	8	70000	30
	3	Катя	Каткина	инженер	2	70000	25
	1	Вася	Васькин	начальник	40	100000	60

**3. Выполните группировку всех сотрудников по специальности, суммарная зарплата которых превышает 100000**

```
SELECT speciality, COUNT(*) AS count, SUM(salary) AS summa FROM sotrudniki GROUP BY speciality HAVING SUM(salary) > 100000;
```

Результат:

Result Grid			
Filter Rows:			
	speciality	count	summa
▶	начальник	2	170000
	инженер	2	120000
	рабочий	7	132000