

ЗАДАНИЕ № 1

Создаем таблицы SalesPeople, Customers, Orders

```
mysql> CREATE TABLE SalesPeople (snum INT PRIMARY KEY AUTO_INCREMENT, sname VARCHAR(20) NOT NULL, city VARCHAR(20) NOT NULL, comm DECIMAL (5,2));
Query OK, 0 rows affected (0,02 sec)
```

```
mysql> 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');
Query OK, 5 rows affected, 5 warnings (0,01 sec)
Records: 5 Duplicates: 0 Warnings: 5
```

```
mysql> SELECT * FROM SalesPeople;
+-----+-----+-----+-----+
| 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 |
+-----+-----+-----+-----+
5 rows in set (0,00 sec)
```

```
mysql> CREATE TABLE Customers (cnum INT PRIMARY KEY AUTO_INCREMENT, cname VARCHAR(20), city VARCHAR(20), rating INT, snum INT);
Query OK, 0 rows affected (0,02 sec)
```

```
mysql> 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');
Query OK, 7 rows affected (0,02 sec)
Records: 7 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT * FROM Customers;
+-----+-----+-----+-----+-----+
| 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 |
+-----+-----+-----+-----+-----+
7 rows in set (0,02 sec)
```

```
mysql> CREATE TABLE Orders (onum INT PRIMARY KEY AUTO_INCREMENT, amt DECIMAL (7,2), odate DATE, cnum INT, snum INT);
Query OK, 0 rows affected (0,01 sec)
```

```
mysql> 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', '4723.00', '1990-05-10', '2006', '1001'),
-> ('3010', '1309.95', '1990-06-10', '2004', '1002'),
-> ('3011', '9891.88', '1990-06-10', '2006', '1001');
Query OK, 10 rows affected (0,00 sec)
Records: 10 Duplicates: 0 Warnings: 0

```

```

mysql> SELECT * FROM Orders;
+-----+-----+-----+-----+-----+
| onum | amt   | odate   | cnum | snum |
+-----+-----+-----+-----+-----+
| 3001 | 18.69 | 1990-03-10 | 2008 | 1007 |
| 3002 | 1900.10 | 1990-03-10 | 2007 | 1004 |
| 3003 | 767.19 | 1990-03-10 | 2001 | 1001 |
| 3005 | 5160.45 | 1990-03-10 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-03-10 | 2008 | 1007 |
| 3007 | 75.75 | 1990-04-10 | 2004 | 1002 |
| 3008 | 4723.00 | 1990-05-10 | 2006 | 1001 |
| 3009 | 1713.23 | 1990-04-10 | 2002 | 1003 |
| 3010 | 1309.95 | 1990-06-10 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-06-10 | 2006 | 1001 |
+-----+-----+-----+-----+-----+
10 rows in set (0,00 sec)

```

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

```

mysql> SELECT city, sname, snum, comm FROM SalesPeople;
+-----+-----+-----+-----+
| 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 |
+-----+-----+-----+-----+
5 rows in set (0,00 sec)

```

```

mysql> SELECT city, cname, snum FROM Customers;
+-----+-----+-----+
| city   | cname   | snum |
+-----+-----+-----+
| London | Hoffman | 1001 |
| Rome   | Giovanni | 1003 |
| San Jose | Liu     | 1002 |
| Berlin | Grass   | 1002 |
| London | Clemens | 1001 |
| Rome   | Pereira | 1004 |
| San Jose | Cisneros | 1007 |
+-----+-----+-----+
7 rows in set (0,01 sec)

```

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

```

mysql> SELECT rating, cname, city FROM Customers WHERE city = 'San Jose';
+-----+-----+-----+
| rating | cname   | city   |
+-----+-----+-----+
| 200    | Liu     | San Jose |
| 300    | Cisneros | San Jose |
+-----+-----+-----+
2 rows in set (0,00 sec)

```

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”)

```

mysql> SELECT DISTINCT snum FROM Orders;
+-----+
| snum |
+-----+

```

```
+-----+
| 1007 |
| 1004 |
| 1001 |
| 1002 |
| 1003 |
+-----+
5 rows in set (0,00 sec)
```

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

```
mysql> SELECT * FROM Customers WHERE cname LIKE 'G%';
+-----+-----+-----+-----+-----+
| cnum | cname   | city   | rating | snum |
+-----+-----+-----+-----+-----+
| 2002 | Giovanni | Rome   | 200    | 1003 |
| 2004 | Grass    | Berlin | 300    | 1002 |
+-----+-----+-----+-----+-----+
2 rows in set (0,00 sec)
```

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

```
mysql> SELECT * FROM Orders WHERE amt >= 1000 LIMIT 0, 1000;
+-----+-----+-----+-----+-----+
| onum | amt      | odate      | cnum | snum |
+-----+-----+-----+-----+-----+
| 3002 | 1900.10 | 1990-03-10 | 2007 | 1004 |
| 3005 | 5160.45 | 1990-03-10 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-03-10 | 2008 | 1007 |
| 3008 | 4723.00 | 1990-05-10 | 2006 | 1001 |
| 3009 | 1713.23 | 1990-04-10 | 2002 | 1003 |
| 3010 | 1309.95 | 1990-06-10 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-06-10 | 2006 | 1001 |
+-----+-----+-----+-----+-----+
7 rows in set (0,00 sec)
```

6. Напишите запрос который выбрал бы наименьшую сумму заказа.
(Из поля "amt" - сумма в таблице "Заказы" выбрать наименьшее значение)

```
mysql> SELECT MIN(amt) FROM Orders;
+-----+
| MIN(amt) |
+-----+
| 18.69    |
+-----+
1 row in set (0,02 sec)
```

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

```
mysql> SELECT * FROM customers WHERE rating > 100 AND NOT city = 'Rome';
+-----+-----+-----+-----+-----+
| cnum | cname   | city      | rating | snum |
+-----+-----+-----+-----+-----+
| 2003 | Liu     | San Jose | 200    | 1002 |
| 2004 | Grass   | Berlin   | 300    | 1002 |
| 2008 | Cisneros | San Jose | 300    | 1007 |
+-----+-----+-----+-----+-----+
3 rows in set (0,00 sec)
```

ЗАДАНИЕ № 2

Создаем таблицу Workers

```
mysql> CREATE TABLE Workers (id INT PRIMARY KEY AUTO_INCREMENT, name VARCHAR(30), surname VARCHAR(30), speciality VARCHAR(30), seniority INT, salary INT, age INT);
Query OK, 0 rows affected (0,03 sec)
```

```
mysql> INSERT INTO Workers (id, name, surname, speciality, seniority, salary, age)
-> VALUES
-> ('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');
```

Query OK, 12 rows affected (0,00 sec)

Records: 12 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM Workers;

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

12 rows in set (0,01 sec)

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

mysql> SELECT * FROM Workers ORDER BY salary;

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

12 rows in set (0,00 sec)

mysql> SELECT * FROM Workers ORDER BY salary DESC;

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

12 rows in set (0,01 sec)

1 2 ** Отсортируйте по возрастанию поле “Зарплата” и выведите 5 строк с наибольшей заработной платой (возможен подзапрос)

```
mysql> SELECT * FROM workers ORDER BY salary DESC LIMIT 5;
```

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

5 rows in set (0,00 sec)

1 3 Выполните группировку всех сотрудников по специальности , суммарная зарплата которых превышает 100000

```
mysql> SELECT speciality, SUM(salary) AS SalarySum
-> FROM Workers
-> GROUP BY speciality
-> HAVING SUM(salary) > 100000;
```

speciality	SalarySum
начальник	170000
инженер	120000
рабочий	132000

3 rows in set (0,00 sec)

```
mysql> SELECT speciality, COUNT(*) AS SpecialityCount FROM Workers WHERE salary >= 100000 GROUP BY speciality;
```

speciality	SpecialityCount
начальник	1

1 row in set (0,00 sec)

```
mysql> SELECT speciality, COUNT(*) AS SpecialityCount FROM Workers WHERE salary >= 50000 GROUP BY speciality;
```

speciality	SpecialityCount
начальник	2
инженер	2

2 rows in set (0,00 sec)