

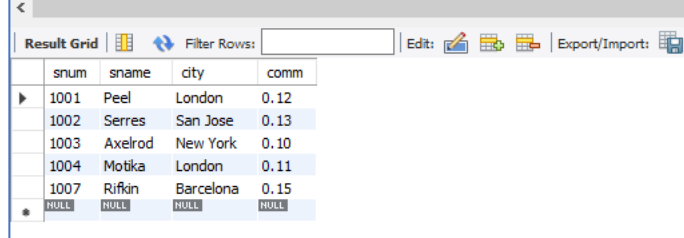
# SQL. Семинар 3. Домашняя работа

## • ЧАСТЬ 1

### 1. Создать и заполнить таблицы salespeople, customers, orders.

- создание и заполнение таблицы salespeople:

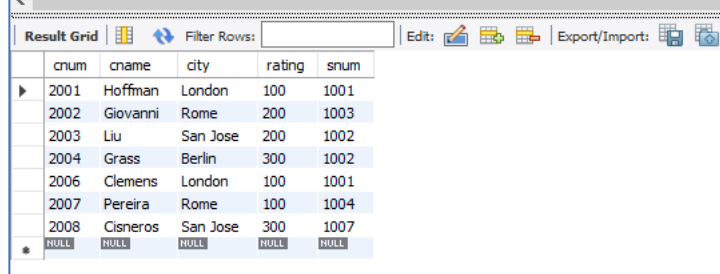
```
6 • CREATE TABLE salespeople(  
7     snum INT PRIMARY KEY,  
8     sname VARCHAR(50) NOT NULL,  
9     city VARCHAR(50),  
10    comm DECIMAL(3, 2)  
11 );  
12 • INSERT INTO salespeople(snum, sname, city, comm)  
13 VALUES  
14 (1001, 'Peel', 'London', '.12'),  
15 (1002, 'Serres', 'San Jose', '.13'),  
16 (1004, 'Motika', 'London', '.11'),  
17 (1007, 'Rifkin', 'Barcelona', '.15'),  
18 (1003, 'Axelrod', 'New York', '.10');  
19 • SELECT * FROM salespeople;  
20
```



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

- создание и заполнение таблицы customers:

```
22 • CREATE TABLE customers(  
23     cnum INT PRIMARY KEY,  
24     cname VARCHAR(50) NOT NULL,  
25     city VARCHAR(50),  
26     rating INT,  
27     snum INT  
28 );  
29 • INSERT INTO customers(cnum, cname, city, rating, snum)  
30 VALUES  
31 (2001, 'Hoffman', 'London', 100, 1001),  
32 (2002, 'Giovanni', 'Rome', 200, 1003),  
33 (2003, 'Liu', 'San Jose', 200, 1002),  
34 (2004, 'Grass', 'Berlin', 300, 1002),  
35 (2006, 'Clemens', 'London', 100, 1001),  
36 (2008, 'Cisneros', 'San Jose', 300, 1007),  
37 (2007, 'Pereira', 'Rome', 100, 1004);  
38 • SELECT * FROM customers;  
39
```



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

- создание и заполнение таблицы orders:

```

41 • CREATE TABLE orders(
42     onum INT PRIMARY KEY,
43     amt DECIMAL(10, 2),
44     odate DATE,
45     cnum INT,
46     snum INT
47 );
48 • INSERT INTO orders(onum, amt, odate, cnum, snum)
49 VALUES
50 (3001, '18.69', '1990-01-10', 2008, 1007),
51 (3003, '767.19', '1990-03-10', 2001, 1001),
52 (3002, '1900.10', '1990-03-10', 2007, 1004),
53 (3005, '5160.45', '1990-03-10', 2003, 1002),
54 (3006, '1098.16', '1990-03-10', 2008, 1007),
55 (3009, '1713.23', '1990-04-10', 2002, 1003),
56 (3007, '75.75', '1990-04-10', 2004, 1002),
57 (3008, '4723.00', '1990-05-10', 2006, 1001),
58 (3010, '1309.95', '1990-06-10', 2004, 1002),
59 (3011, '9891.88', '1990-06-10', 2006, 1001);
60 • SELECT * FROM orders;

```

onum	amt	odate	cnum	snum
3001	18.69	1990-01-10	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
NULL	NULL	NULL	NULL	NULL

## Выполнение заданий:

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

```

78 • SELECT city, sname, snum, comm
79 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
NULL	NULL	NULL	NULL

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

```

85 • SELECT cname, rating
86 FROM customers
87 WHERE city = 'San Jose';
88

```

cname	rating
Liu	200
Cisneros	300

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”)

```

93 • SELECT DISTINCT snum AS "Продавцы"
94 FROM orders
95 ORDER BY snum;

```

Продавцы
1001
1002
1003
1004
1007

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

```

101 • SELECT cname
102 FROM customers
103 WHERE cname LIKE 'G%'
104 ORDER BY cname;

```

cname
Giovanni
Grass

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

```

110 • SELECT onum, amt
111 FROM orders
112 WHERE amt > 1000
113 ORDER BY onum;

```

onum	amt
3002	1900.10
3005	5160.45
3006	1098.16
3008	4723.00
3009	1713.23
3010	1309.95
3011	9891.88
NULL	NULL

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

```

119 • SELECT MIN(amt) AS 'MIN_order'
120 FROM orders;

```

MIN_order
18.69

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

```
126 • SELECT cname, city, rating
127 FROM customers
128 WHERE rating > 100 AND NOT city = 'Rome'
129 ORDER BY cname;
```

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

## • ЧАСТЬ 2

### Таблица для работы (из классной работы)

```
133 • SELECT * FROM staff;
134
```

	id	firstname	lastname	post	seniority	salary	age
▶	1	Вася	Петров	Начальник	40	100000	60
	2	Петр	Власов	Начальник	8	70000	30
	3	Катя	Катина	Инженер	2	70000	25
	4	Саша	Сасин	Инженер	12	50000	35
	5	Иван	Иванов	Рабочий	40	30000	59
	6	Петр	Петров	Рабочий	20	25000	40
	7	Сидр	Сидоров	Рабочий	10	20000	35
	8	Антон	Антонов	Рабочий	8	19000	28
	9	Юрий	Юрков	Рабочий	5	15000	25
	10	Максим	Максимов	Рабочий	2	11000	22
	11	Юрий	Галкин	Рабочий	3	12000	24
	12	Людмила	Маркина	Уборщик	10	10000	49
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Выполнение заданий:

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

```
138 -- ASC сортировка (по возрастанию)
139 • SELECT * FROM staff
140 ORDER BY salary;
```

	id	firstname	lastname	post	seniority	salary	age
▶	12	Людмила	Маркина	Уборщик	10	10000	49
	10	Максим	Максимов	Рабочий	2	11000	22
	11	Юрий	Галкин	Рабочий	3	12000	24
	9	Юрий	Юрков	Рабочий	5	15000	25
	8	Антон	Антонов	Рабочий	8	19000	28
	7	Сидр	Сидоров	Рабочий	10	20000	35
	6	Петр	Петров	Рабочий	20	25000	40
	5	Иван	Иванов	Рабочий	40	30000	59
	4	Саша	Сасин	Инженер	12	50000	35
	2	Петр	Власов	Начальник	8	70000	30
	3	Катя	Катина	Инженер	2	70000	25
	1	Вася	Петров	Начальник	40	100000	60

```
141 -- DESC сортировка (по убыванию)
142 • SELECT * FROM staff
143 ORDER BY salary DESC;
```

	id	firstname	lastname	post	seniority	salary	age
▶	1	Вася	Петров	Начальник	40	100000	60
	2	Петр	Власов	Начальник	8	70000	30
	3	Катя	Катина	Инженер	2	70000	25
	4	Саша	Сасин	Инженер	12	50000	35
	5	Иван	Иванов	Рабочий	40	30000	59
	6	Петр	Петров	Рабочий	20	25000	40
	7	Сидр	Сидоров	Рабочий	10	20000	35
	8	Антон	Антонов	Рабочий	8	19000	28
	9	Юрий	Юрков	Рабочий	5	15000	25
	11	Юрий	Галкин	Рабочий	3	12000	24
	10	Максим	Максимов	Рабочий	2	11000	22
	12	Людмила	Маркина	Уборщик	10	10000	49

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

```
149 • SELECT *
150 FROM (SELECT * FROM staff ORDER BY salary DESC LIMIT 5) AS quarry
151 ORDER BY salary;
```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

	id	firstname	lastname	post	seniority	salary	age
▶	5	Иван	Иванов	Рабочий	40	30000	59
	4	Саша	Сасин	Инженер	12	50000	35
	2	Петр	Власов	Начальник	8	70000	30
	3	Катя	Катина	Инженер	2	70000	25
	1	Вася	Петров	Начальник	40	100000	60

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

```
157 • SELECT post, SUM(salary) as sum_salary
158 FROM staff
159 GROUP BY post
160 HAVING sum_salary > 100000;
```

<

Result Grid | Filter Rows: | Export: | Wra

	post	sum_salary
▶	Начальник	170000
	Инженер	120000
	Рабочий	132000