Выполнил: Ковешников Д.А.

Весь код указан в конце.

Создание таблиц

```
create table SELLER (
SNUM number(4) NOT NULL,
SNAME varchar2(10) NOT NULL,
CITY varchar2(10) NOT NULL,
COMM number(7,2) NOT NULL)


INSERT ALL
INTO SELLER (SNUM, SNAME, CITY, COMM) VALUES (1001, 'Peel', 'London', 0.12)
INTO SELLER (SNUM, SNAME, CITY, COMM) VALUES (1002, 'Serres', 'San Jose', 0.13)
INTO SELLER (SNUM, SNAME, CITY, COMM) VALUES (1004, 'Motica', 'London', 0.11)
INTO SELLER (SNUM, SNAME, CITY, COMM) VALUES (1007, 'Rifkin', 'Barcelona', 0.15)
INTO SELLER (SNUM, SNAME, CITY, COMM) VALUES (1003, 'Axelrod', 'New York', 0.10)
SELECT * FROM dual


create table CUSTOMER (
CNUM number(4) NOT NULL,
CNAME varchar2(10) NOT NULL,
CITY varchar2(10) NOT NULL,
RATING number(3) NOT NULL)

INSERT ALL
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2001, 'Hoffman', 'London', 100)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2002, 'Giovanni', 'Rome', 200)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2003, 'Liu', 'San Jose', 200)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2004, 'Grass', 'Berlin', 300)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2006, 'Clemens', 'London', 100)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2008, 'Cisneros', 'San Jose', 300)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2007, 'Pereira', 'Rome', 100)
SELECT * FROM dual

create table ORDERS (
ONUM number(4) NOT NULL,
AMT varchar2(10) NOT NULL,
ODATE date NOT NULL,
CNUM number(4) NOT NULL,
SNUM number(4) NOT NULL)


INSERT ALL
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3001, 18.69, TO_DATE ('03.01.2020', 'DD.MM.YYYY'), 2008,
1007)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3003, 767.19, TO_DATE ('03.01.2020', 'DD.MM.YYYY'), 2001,
1001)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3002, 1900.10, TO_DATE ('03.01.2020', 'DD.MM.YYYY'), 2007,
1004)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3005, 5160.45, TO_DATE ('03.01.2020', 'DD.MM.YYYY'), 2003,
1002)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3006, 1098.16, TO_DATE ('03.01.2020', 'DD.MM.YYYY'), 2008,
1007)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3009, 1713.23, TO_DATE ('04.01.2020', 'DD.MM.YYYY'), 2002,
1003)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3007, 75.75, TO_DATE ('04.01.2020', 'DD.MM.YYYY'), 2004,
1002)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3008, 4723.00, TO_DATE ('05.01.2020', 'DD.MM.YYYY'), 2006,
1001)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3010, 1309.95, TO_DATE ('06.01.2020', 'DD.MM.YYYY'), 2004,
1002)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3011, 9891.88, TO_DATE ('06.01.2020', 'DD.MM.YYYY'), 2006,
1001)
SELECT * FROM dual
```

Селекты

1. Выведите всех покупателей с рейтингом выше 100, проживающих в городах, в названии которых вторая буква не равна “o” , а четвертая буква не равна “e”.

```
SELECT * FROM CUSTOMER
WHERE City NOT LIKE '_o%' AND City NOT LIKE '___e%' AND RATING > 100
```

Снимок экрана:

```
SELECT * FROM CUSTOMER
WHERE City NOT LIKE '_o%' AND City NOT LIKE '___e%' AND RATING > 100
```

CNUM	CNAME	CITY	RATING
2003	Liu	San Jose	200
2004	Grass	Berlin	300
2008	Cisneros	San Jose	300

Download CSV

3 rows selected.

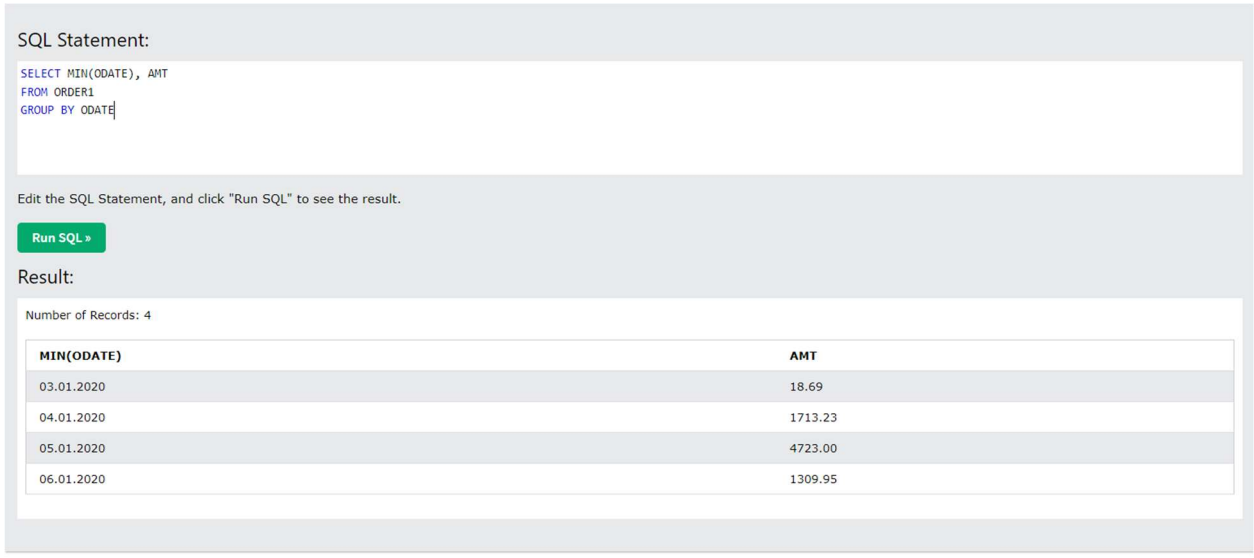
ИЛИ в w3school

SELECT MIN(ODATE), AMT

FROM ORDER1

GROUP BY ODATE

Снимок экрана:



2. Запросите двумя способами все заказы на 3 и 4 января.

1-й способ: `SELECT * FROM ORDERS`

`WHERE ODATE NOT LIKE '05%' AND ODATE NOT LIKE '06%'`

Снимок экрана:

```
SELECT * FROM ORDERS
WHERE ODATE NOT LIKE '05%' AND ODATE NOT LIKE '06%'
```

ONUM	AMT	ODATE	CNUM	SNUM
3001	18.69	03-JAN-20	2008	1007
3003	767.19	03-JAN-20	2001	1001
3002	1900.1	03-JAN-20	2007	1004
3005	5160.45	03-JAN-20	2003	1002
3006	1098.16	03-JAN-20	2008	1007
3009	1713.23	04-JAN-20	2002	1003
3007	75.75	04-JAN-20	2004	1002

Download CSV

2-й способ: `SELECT * FROM ORDERS`

`WHERE ODATE >= TO_DATE('03.01.2020', 'DD.MM.YYYY') AND ODATE < TO_DATE('05.01.2020', 'DD.MM.YYYY')`

Снимок экрана:

```
SELECT * FROM ORDERS
WHERE ODATE >= TO_DATE('03.01.2020', 'DD.MM.YYYY') AND ODATE < TO_DATE('05.01.2020'
```

ONUM	AMT	ODATE	CNUM	SNUM
3001	18.69	03-JAN-20	2008	1007
3003	767.19	03-JAN-20	2001	1001
3002	1900.1	03-JAN-20	2007	1004
3005	5160.45	03-JAN-20	2003	1002
3006	1098.16	03-JAN-20	2008	1007
3009	1713.23	04-JAN-20	2002	1003
3007	75.75	04-JAN-20	2004	1002
3001	18.69	03-JAN-20	2008	1007
3003	767.19	03-JAN-20	2001	1001
3002	1900.1	03-JAN-20	2007	1004
3005	5160.45	03-JAN-20	2003	1002
3006	1098.16	03-JAN-20	2008	1007
3009	1713.23	04-JAN-20	2002	1003
3007	75.75	04-JAN-20	2004	1002

Download CSV

14 rows selected.

3. Выведите сумму самого раннего заказа за каждую дату. (Задание решено криво).

```
SELECT AMT FROM ORDERS
WHERE ONUM NOT LIKE '%3' AND ONUM NOT LIKE '%2' AND ONUM NOT LIKE '%5' AND ONUM NOT LIKE '%6' AND ONUM NOT LIKE '%7' AND
ONUM NOT LIKE '%11'
ORDER BY ODATE
ASC
```

Снимок экрана:

```
SELECT AMT FROM ORDERS
WHERE ONUM NOT LIKE '%3' AND ONUM NOT LIKE '%2' AND ONUM NOT LIKE '%5' AND ONUM NOT
ORDER BY ODATE
ASC
```

AMT
18.69
18.69
1713.23
1713.23
4723
4723
1309.95
1309.95

Download CSV

8 rows selected.

4. Напишите запрос, который сосчитал бы сумму всех заказов для покупателей, которых обслуживает продавец с именем Peel.

Снимок экрана:

```
SELECT SUM(AMT)
FROM Orders, Seller
WHERE seller.snum = orders.snum and seller.sname LIKE '%Peel%'
```


```
SELECT SUM(AMT)
FROM Orders, Seller
WHERE seller.snum = orders.snum and seller.sname LIKE '%Peel%'
```


SUM(AMT)
61528.28

Download CSV

5. Напишите запрос, который выводит все заказы, сумма которых больше средней по все заказам, используя подзапрос.

```
SELECT *
FROM Orders
WHERE Orders.AMT > (SELECT AVG(AMT) AS avgPrice FROM Orders)
```

Снимок экрана:


```
SELECT *
FROM Orders
WHERE Orders.AMT > (SELECT AVG(AMT) AS avgPrice FROM Orders)
```


ONUM	AMT	ODATE	CNUM	SNUM
3005	5160.45	03-JAN-20	2003	1002
3008	4723	05-JAN-20	2006	1001
3011	9891.88	06-JAN-20	2006	1001
3005	5160.45	03-JAN-20	2003	1002
3008	4723	05-JAN-20	2006	1001
3011	9891.88	06-JAN-20	2006	1001

Download CSV

6 rows selected

6. Напишите запрос, который вывел бы для каждого заказа его номер, стоимость и имя заказчика. Данные вывести для заказчиков, размещенных не в Лондоне и не в Риме.

```
SELECT orders.onum, orders.amt, customer.cname
FROM orders, customer
WHERE customer.cnum = orders.cnum and customer.city not like '%London%' and customer.city not like '%Rome%'
```

Снимок экрана:

```
SELECT orders.onum, orders.amt, customer.cname
FROM orders, customer
WHERE customer.cnum = orders.cnum and customer.city not like '%London%' and custome
```

ONUM	AMT	CNAME
3001	18.69	Cisneros
3005	5160.45	Liu
3006	1098.16	Cisneros
3007	75.75	Grass
3010	1309.95	Grass

Download CSV

5 rows selected.

7. Вывести пары имен покупатель – продавец, которые совершили сделки не 4 и не 6 января, при этом сумма каждой сделки от 1000 до 5000. Отсортировать по возрастанию суммы. Без использования подзапросов.

```
SELECT seller.sname, customer.cname, orders.onum, orders.odate
FROM seller, orders, customer
where seller.snum = orders.snum and customer.cnum = orders.cnum and odate <> to_date ('04.01.2020', 'dd.mm.yyyy.') and
odate <> to_date ('06.01.2020', 'dd.mm.yyyy') and amt BETWEEN 1000 AND 5000
ORDER BY amt
```

Снимок экрана:

```
SELECT seller.sname, customer.cname, orders.onum, orders.odate
FROM seller, orders, customer
where seller.snum = orders.snum and customer.cnum = orders.cnum and odate <> to_dat
ORDER BY amt
```

SNAME	CNAME	ONUM	ODATE
Rifkin	Cisneros	3006	03-JAN-20
Motica	Pereira	3002	03-JAN-20
Peel	Clemens	3008	05-JAN-20

Download CSV

3 rows selected.

8. Вывести сумму сделки и имена покупателей, которые совершили сделку 3 января, но в городе отличном от города продавца.

```
SELECT orders.amt, customer.cname
FROM orders, customer, seller
where seller.snum = orders.snum and customer.cnum = orders.cnum and seller.city <> customer.city and odate = to_date
('03.01.2020', 'dd.mm.yyyy')
```

Снимок экрана:

```
SELECT orders.amt, customer.cname
FROM orders, customer, seller
where seller.snum = orders.snum and customer.cnum = orders.cnum and seller.city <>
```

AMT	CNAME
18.69	Cisneros
1098.16	Cisneros
1900.1	Pereira

Download CSV

3 rows selected.

Весь код

```
create table SELLER (
SNUM number(4) NOT NULL,
SNAME varchar2(10) NOT NULL,
CITY varchar2(10) NOT NULL,
COMM number(7,2) NOT NULL)
;

INSERT ALL
INTO SELLER (SNUM, SNAME, CITY, COMM) VALUES (1001, 'Peel', 'London', 0.12)
INTO SELLER (SNUM, SNAME, CITY, COMM) VALUES (1002, 'Serres', 'San Jose', 0.13)
INTO SELLER (SNUM, SNAME, CITY, COMM) VALUES (1004, 'Motica', 'London', 0.11)
INTO SELLER (SNUM, SNAME, CITY, COMM) VALUES (1007, 'Rifkin', 'Barcelona', 0.15)
INTO SELLER (SNUM, SNAME, CITY, COMM) VALUES (1003, 'Axelrod', 'New York', 0.10)
SELECT * FROM dual
;

create table CUSTOMER (
CNUM number(4) NOT NULL,
CNAME varchar2(10) NOT NULL,
CITY varchar2(10) NOT NULL,
RATING number(3) NOT NULL)



INSERT ALL
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2001, 'Hoffman', 'London', 100)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2002, 'Giovanni', 'Rome', 200)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2003, 'Liu', 'San Jose', 200)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2004, 'Grass', 'Berlin', 300)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2006, 'Clemens', 'London', 100)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2008, 'Cisneros', 'San Jose', 300)
INTO CUSTOMER (CNUM, CNAME, CITY, RATING) VALUES (2007, 'Pereira', 'Rome', 100)
SELECT * FROM dual;



create table ORDERS (
ONUM number(4) NOT NULL,
AMT varchar2(10) NOT NULL,
ODATE date NOT NULL,
CNUM number(4) NOT NULL,
SNUM number(4) NOT NULL)



INSERT ALL
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3001, 18.69, TO_DATE ('03.01.2020', 'DD.MM.YYYY'), 2008,
1007)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3003, 767.19, TO_DATE ('03.01.2020', 'DD.MM.YYYY'), 2001,
1001)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3002, 1900.10, TO_DATE ('03.01.2020', 'DD.MM.YYYY'), 2007,
1004)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3005, 5160.45, TO_DATE ('03.01.2020', 'DD.MM.YYYY'), 2003,
1002)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3006, 1098.16, TO_DATE ('03.01.2020', 'DD.MM.YYYY'), 2008,
1007)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3009, 1713.23, TO_DATE ('04.01.2020', 'DD.MM.YYYY'), 2002,
1003)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3007, 75.75, TO_DATE ('04.01.2020', 'DD.MM.YYYY'), 2004,
1002)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3008, 4723.00, TO_DATE ('05.01.2020', 'DD.MM.YYYY'), 2006,
1001)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3010, 1309.95, TO_DATE ('06.01.2020', 'DD.MM.YYYY'), 2004,
1002)
INTO ORDERS (ONUM, AMT, ODATE, CNUM, SNUM) VALUES (3011, 9891.88, TO_DATE ('06.01.2020', 'DD.MM.YYYY'), 2006,
1001)
SELECT * FROM dual;



SELECT * FROM CUSTOMER
WHERE City NOT LIKE '_o%' AND City NOT LIKE '___e%' AND RATING > 100



SELECT * FROM ORDERS
WHERE ODATE NOT LIKE '05%' AND ODATE NOT LIKE '06%'
```

```
SELECT * FROM ORDERS
WHERE ODATE >= TO_DATE('03.01.2020', 'DD.MM.YYYY') AND ODATE < TO_DATE('05.01.2020', 'DD.MM.YYYY')



SELECT AMT FROM ORDERS
WHERE ONUM NOT LIKE '%3' AND ONUM NOT LIKE '%2' AND ONUM NOT LIKE '%5' AND ONUM NOT LIKE '%6' AND ONUM NOT
LIKE '%7' AND ONUM NOT LIKE '%11'
ORDER BY ODATE
ASC



SELECT SUM(AMT)
FROM Orders, Seller
WHERE seller.snum = orders.snum and seller.sname LIKE '%Peel%'



SELECT *
FROM Orders
WHERE Orders.AMT > (SELECT AVG(AMT) AS avgPrice FROM Orders)



SELECT orders.onum, orders.amt, customer.cname
FROM orders, customer
WHERE customer.cnum = orders.cnum and customer.city not like '%London%' and customer.city not like '%Rome%'



SELECT seller.sname, customer.cname, orders.onum, orders.odate
FROM seller, orders, customer
where seller.snum = orders.snum and customer.cnum = orders.cnum and odate <> to_date ('04.01.2020',
'dd.mm.yyyy.') and odate <> to_date ('06.01.2020', 'dd.mm.yyyy') and amt BETWEEN 1000 AND 5000
ORDER BY amt
```

```
SELECT orders.amt, customer.cname
FROM orders, customer, seller
where seller.snum = orders.snum and customer.cnum = orders.cnum and seller.city <> customer.city and odate =
to_date ('03.01.2020', 'dd.mm.yyyy')
```