SQL запросы

Задание 1

Таблицы, с которыми предстоит работать: заказы (Orders) и клиенты (Customers), найти их можно тут:
https://www.w3schools.com/sql/trysql.asp?filename=trysql_asc

Необходимо вывести id клиента и имя клиента, сделавших больше 5-ти заказов, а также количество заказов и дату последнего заказа. Список должен идти по убыванию от клиентов с большим количеством заказов к клиентам с меньшим количеством заказов

Решение:

```
SELECT Customers. CustomerID, count(*) as "count", Customers. CustomerName, MAX(Orders. OrderDate) as "Orders. OrderDate"
FROM Customers
JOIN Orders ON Customers. CustomerID = Orders. CustomerID
Group by Customers. CustomerID
Having count(*)>=5
ORDER BY "count" DESC
```

Задание 2

Вводные данные:

Имеется таблица клиентов, в которой хранятся данные по компании, сотрудникам и их заявкам.
Структура таблицы client:
Id – уникальный идентификатор компании
company_name – название компании
user_number – уникальный идентификатор сотрудника
user_name – фамилия и инициалы сотрудников
amount_of_applications – количество заявок
Задание:

Вывести ФИО сотрудника и название его компании, количество заявок от которых принадлежит интервалу от 2 до 14 (включая границы). Информацию отсортировать сначала по полю ФИО сотрудника (в обратном алфавитном порядке), а затем по названию компании (по алфавиту).

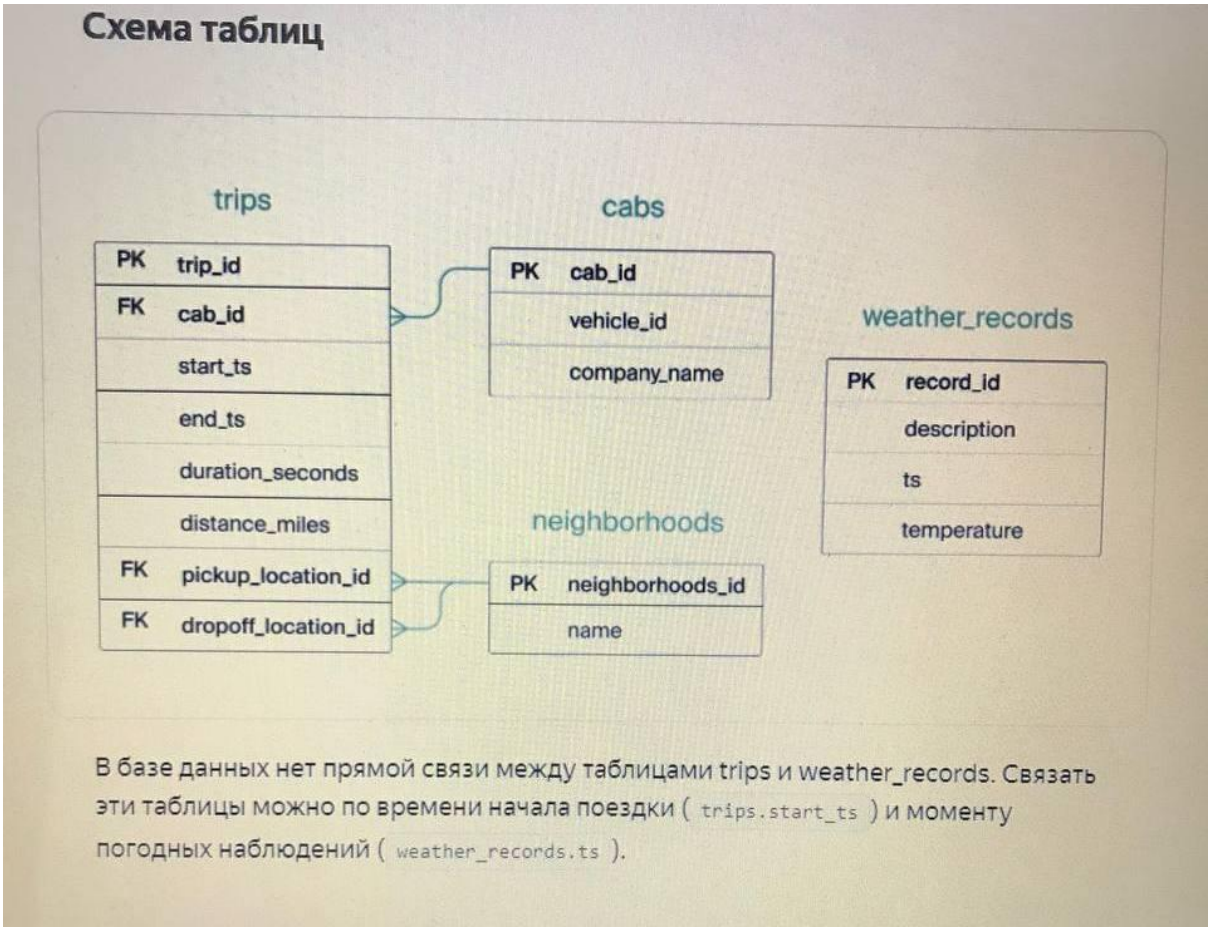
Решение:

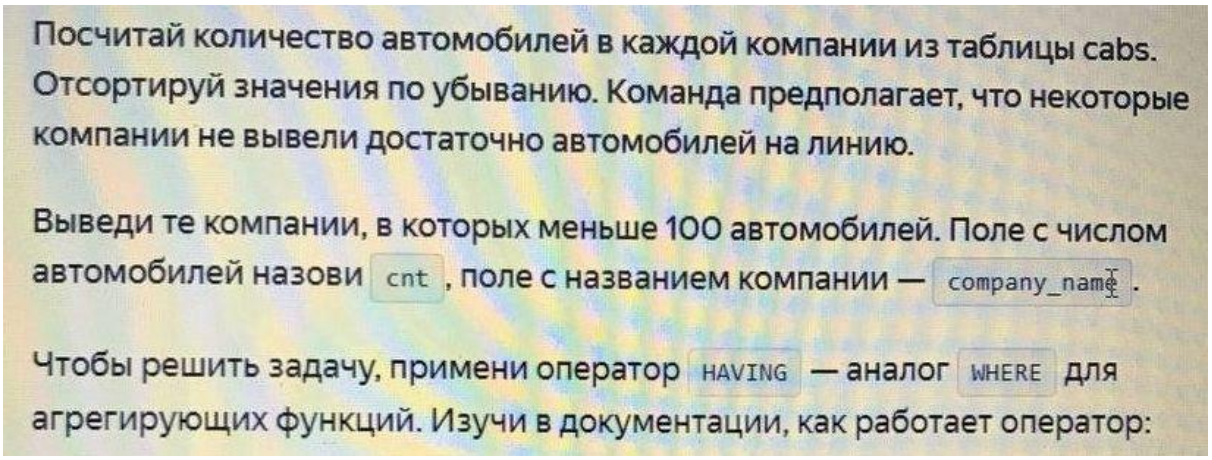
```
SELECT user_name, company_name, amount_of_applications FROM client
WHERE amount_of_applications BETWEEN 2 and 14
ORDER by user_name DESC
```

```
SELECT user_name, company_name, amount_of_applications FROM client
```

WHERE amount_of_applications BETWEEN 2 and 14
ORDER by company_name ASC

Задание 3





Решение:

SELECT company_name, COUNT(vehicle_id) AS cnt

FROM cabs

GROUP BY company_name

HAVING COUNT(vehicle_id) < 100

ORDER BY cnt DESC

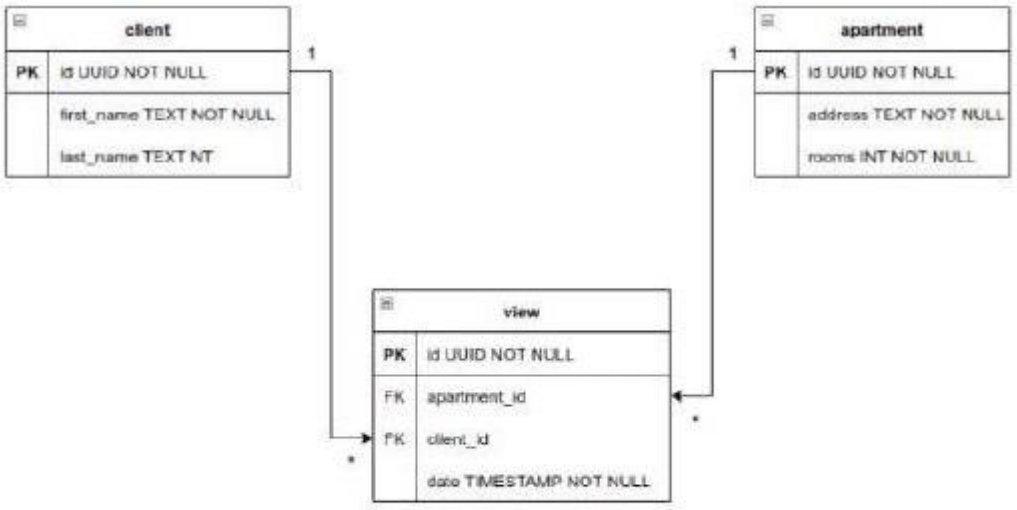
Задание 4

1) Таблица "client", которая содержит информацию о клиентах компании. Она содержит поля:
"id" – уникальный идентификатор клиента;
"first_name" – имя клиента;
"last_name" – фамилия клиента.
2) Таблица "apartment", которая содержит информацию о продаваемых квартирах. Она содержит поля:
"id" – уникальный идентификатор квартиры;
"address" – адрес квартиры;
"rooms" – количество комнат в квартире.
3) Таблица "view", которая содержит информацию о записи на просмотр квартир клиентами. Она содержит поля:
"id" – уникальный идентификатор записи;
"apartment_id" – идентификатор квартиры;
"client_id" – идентификатор клиента, записанного на просмотр;
"date" – дата просмотра квартиры.

Обратите внимание, что на просмотр одной квартиры могут записаться несколько клиентов, а один клиент может записаться на просмотр нескольких квартир.

Ниже представлена схема описанной базы данных.

Ниже представлена схема описанной базы данных.



Исходя из описания и схемы базы данных, составьте SQL-запрос, результатом которого будет список фамилий клиентов, записанных на просмотр двух и более трехкомнатных квартир.

Решение:

```
SELECT client.last_name,  COUNT(*) as "count"
FROM client
JOIN view ON client.id=view.client_id
JOIN apartment ON view.apartment_id=apartment.id
GROUP BY last_name,apartment.rooms
HAVING apartment.rooms=3 AND COUNT(*)>1
```

Задание 5

```
CREATE TABLE users (
id integer GENERATED ALWAYS AS IDENTITY,
user_name VARCHAR,
level_id INTEGER,
skill INTEGER
 )

INSERT INTO users (user_name, level_id, skill)
 VALUES ('Anton', 1, 900000);

 INSERT INTO users (user_name, level_id, skill)
 VALUES ('Denis', 3, 4000);

 INSERT INTO users (user_name, level_id, skill)
 VALUES ('Petr', 2, 50000);

INSERT INTO users (user_name, level_id, skill)
```

```
 VALUES ('Andrey', 4, 20);


INSERT INTO users (user_name, level_id, skill)
 VALUES ('Olga', 1, 600000);


INSERT INTO users (user_name, level_id, skill)
VALUES ('Anna', 1, 1600000);


CREATE TABLE levels (
id integer GENERATED ALWAYS AS IDENTITY,
level_name VARCHAR (100)
 )


 INSERT INTO levels (level_name)
 VALUES ('admin');


 INSERT INTO levels (level_name)
 VALUES ('power_user');


 INSERT INTO levels (level_name)
 VALUES ('user');


 INSERT INTO levels (level_name)
VALUES ('guest');
```

ЗАДАНИЯ:

1. Отобрать из таблицы user всех пользователей, у которых level_id=1, skill> 799000 и в имени встречается буква а.

```
SELECT user_name

FROM users

WHERE level_id=1 AND skill > 799000 AND user_name LIKE '%a%';
```

2. Удалить всех пользователей, у которых skill меньше 100000

```
DELETE

FROM users

WHERE skill<100000
```

3. Вывести все данные из таблицы user в порядке убывания по полю skill

```
SELECT user_name FROM users

ORDERBY skill DESC
```

4. Добавить в таблицу user нового пользователя по имени Oleg, с уровнем 4 и skill =10

```
INSERT INTO users (user_name, level_id, skill)

VALUES ('Oleg', 4, 10)
```

5. Обновить данные в таблице user - для пользователей с level_id меньше 2 проставить skill 2000000

```
UPDATE users

 SET skill='2000000'

 WHERE level_id<2
```

6. Выбрать user_name всех пользователей уровня admin используя подзапрос

```
SELECT users.user_name, (SELECT levels.level_name FROM levels
WHERE users.level_id=levels.id AND levels.level_name='admin' )
FROM users
```

7. Выбрать user_name всех пользователей уровня admin используя join

```
SELECT users.user_name
FROM users
JOIN levels ON users.level_id=levels.id
WHERE levels.level_name='admin'
```