

**Правительство Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет
«Высшая школа экономики»**

Московский институт электроники и математики им. А.Н. Тихонова

Департамент компьютерной инженерии

Отчёт по домашней работе по дисциплине

«Базы данных»

На тему «Проектирование реляционной базы данных аренды уличного
транспорта»

Выполнили:
студенты группы БИВ173
Куркин А.Н., Клюваков Д.Д.

Проверила:
Доц., к.т.н. Карпова И.П.

Москва, 2019

О г л а в л е н и е

Задание	3
Инфологическое проектирование	3
Логическое проектирование реляционной БД	5
Составление реляционных отношений	6
Нормализация	8
Ограничение целостности	10
Описание групп пользователей и прав доступа	11
Создание таблиц	11
Создание представлений	13
Назначение прав доступа	14
Назначение индексов	14

Задание

Предметная область – аренда городского транспорта.

База данных будет содержать данные клиентов, точки для парковки, виды транспортных средств, транспортные средства, договор.

Инфологическое проектирование

Анализ предметной области

- В договоре может быть указан только один клиент, одно транспортное средство;
- Вид ТС представлены как обычными самокатами/велосипедами, так и электро;
- Каждое ТС, точка парковки и договор имеет свой уникальный номер;

Сущности предметной области:

1. **Клиент.** Атрибуты: ФИО, серия, номер паспорта, номер телефона, номер карты, электронная почта;
2. **Вид ТС.** Атрибуты: вид, цена за час, грузоподъемность, название модели;
3. **ТС.** Атрибуты: id ТС, уровень заряда, состояние;
4. **Точка.** Атрибуты: id, вместимость(количество парковочных мест), координаты, ближайшая станция метро;
5. **Договор.** Атрибуты: id договора, дата начала, дата окончания, итоговая стоимость, примечания.

Исходя из выявленных сущностей, получаем ER-диаграмму (См. Рис.1):

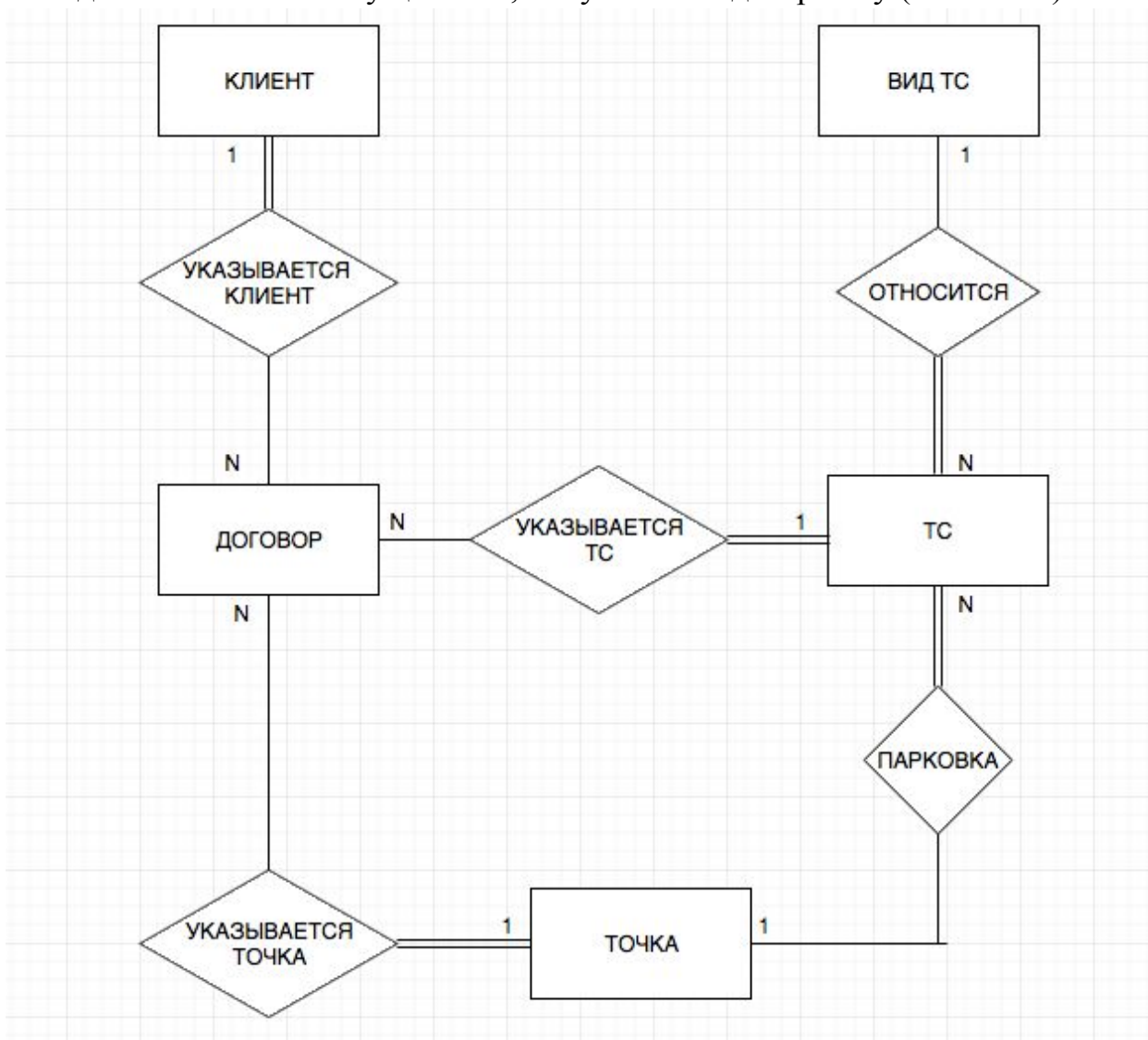


Рис. 1 ER-диаграмма

Логическое проектирование реляционной БД

Преобразуем ER-диаграмму в схему базы данных:

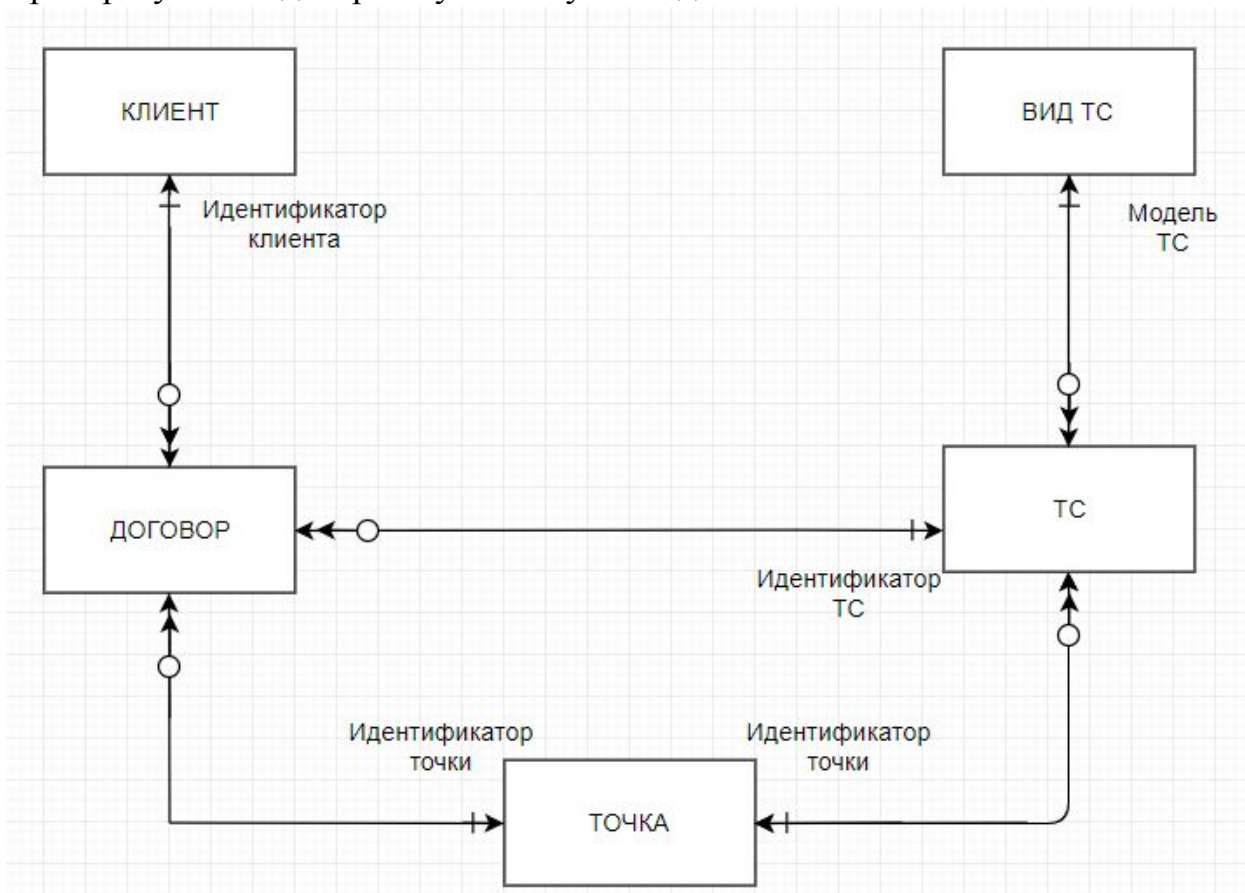


Рис. 2 Схема РБД, полученная из ER-диаграммы

Составление реляционных отношений

Таблица №1 Клиенты(Clients)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип данных</i>	<i>Примечания</i>
ФИО	fio	V(100)	Обязательное составное поле
Серия+номер паспорта	pass	N(10)	Первичный ключ
Телефон	phn	N(11)	Обязательное уникальное поле
Номер карты	crd	N(20)	Обязательное поле
Электр. почта	email	V(100)	Обязательное поле

Таблица №2 Виды ТС(VTypes)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип данных</i>	<i>Примечания</i>
Грузоподъемность	wght	N(3)	Обязательное поле
Цена за час	price	N(3)	Обязательное поле
Название модели	model	V(100)	Первичный ключ
Тип ТС	ts_type	V(16)	Обязательное поле
Длительность поездки на одном заряде аккумулятора	dur_charge	N(2)	

Таблица №3 Транспортные средства(Vehicles)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип данных</i>	<i>Примечания</i>
Идентификатор	id	N(4)	Первичный ключ
Уровень заряда	charge	N(3)	
Точка	Id_p	N(3)	(ВнК Parking)
Состояние	comments	V(100)	Обязательное поле
Тип ТС	ts_type	V(16)	(ВнК Types)

Таблица №4 Точки парковки(Parking)

Содержание поля	Имя поля	Тип данных	Примечания
Идентификатор	id	N(3)	Первичный ключ
Адрес	Address	V(100)	Обязательное составное поле
Вместимость	cont	N(2)	Обязательное поле

Отношение №5 Поездка (Trip)

Содержание поля	Имя поля	Тип данных	Примечания
Идентификатор	Id	N(6)	Первичный ключ
Дата начала	Date_start	Date	Обязательное поле
Дата окончания	Date_end	Date	Обязательное поле
Итоговая стоимость	Sumprice	N(5)	Обязательное поле
Примечания	comments	V(100)	
Идентификатор ТС	Id_ts	N(4)	Внешний ключ(к Vehicles)
Клиент	Id_cl	N(10)	Внешний ключ (к Clients)
Начальная Точка	Id_p1	N(3)	Внешний ключ(к Parking)
Конечная точка	Id_p2	N(3)	Внешний ключ(к Parking)
Дистанция(в км)	distation	N(3,2)	Обязательное

При любом тарифе клиент заранее не знает, какая будет финальная точка и, соответственно, время поездки. В начале поездки оформляется доступ на пользование (на сутки/месяц и тд) и фиксируются время начала поездки и начальная точка. Чек (данные о поездке) будет составлен по факту сдачи (будь то промежуточная, чтобы уложиться в 30 минут или итоговая). Таблица представляет из себя данные о поездках между двумя конкретными парковками. Поле точка и дата окончания будет автоматически заполняться при сдаче ТС(когда сработает фиксатор парковочной ячейки, информация.

Нормализация

1НФ:

Разделим в таблице «Клиенты» атрибут Фамилия, имя, отчество на два атрибута: *Фамилия* и *ИмяОтчество (ИО)*

Разделим атрибут Адрес на три атрибута: точные GPS координаты по *широте* и *долготе* и *ближайшая станция метро*. По аналогу существующих приложений, используем тип данных Numeric(6,4) для максимальной эффективности в точности и объёме данных.

Далее, ввиду, активного развития сети метрополитена и большего количества станций целесообразно для удобства **создать** справочную таблицу «Станции метро», включающую два атрибута: *Идентификатор*, *Название станции*.

Выделим атрибут *Тип ТС* в справочную таблицу «Типы ТС», включающую два атрибута: *Идентификатор типа ТС*, *Название типа*.

Таблица №6 Клиенты(Clients)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип данных</i>	<i>Примечания</i>
Фамилия	f	V(100)	Обязательное поле
ИО	Io	V(100)	Обязательное поле
Серия+номер паспорта	pass	N(10)	Первичный ключ
Телефон	phn	N(11)	Обязательное уникальное поле
Номер карты	crd	N(20)	Обязательное поле
Электр. почта	email	V(100)	Обязательное поле

Таблица №7 Точки парковки(Parking)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип данных</i>	<i>Примечания</i>
Идентификатор	id	N(3)	Первичный ключ
Ближайшее метро	Id m	N(3)	(ВнК Metro)
Точные координаты(долгота)	gpsy	N(6,4)	Обязательное поле
Точные координаты(широта)	gpsx	N(6,4)	Обязательное поле
Вместимость	cont	N(2)	Обязательное поле

Таблица №8 Станции метро(Metro)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип данных</i>	<i>Примечания</i>
Идентификатор метро	id_m	N(3)	Первичный ключ
Название станции	station	V(100)	Обязательное поле

Таблица №9 Виды ТС(VTypes)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип данных</i>	<i>Примечания</i>
Грузоподъемность	wght	N(3)	Обязательное поле
Цена за час	price	N(3)	Обязательное поле
Название модели	model	V(100)	Первичный ключ
Тип ТС	ts_type	N(2)	ВнК TStyle
Длительность поездки на одном заряде аккумулятора	dur_charge	N(2)	

Таблица №10 Типы ТС(TStyle)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип данных</i>	<i>Примечания</i>
Идентификатор типа ТС	id_tts	N(2)	Первичный ключ
Название типа	typen	V(100)	Обязательное поле

2-4НФ:

Все отношения удовлетворяют требованиям 2,3,4 нормальных форм.

Итоговая схема базы данных(См. Рис 3):

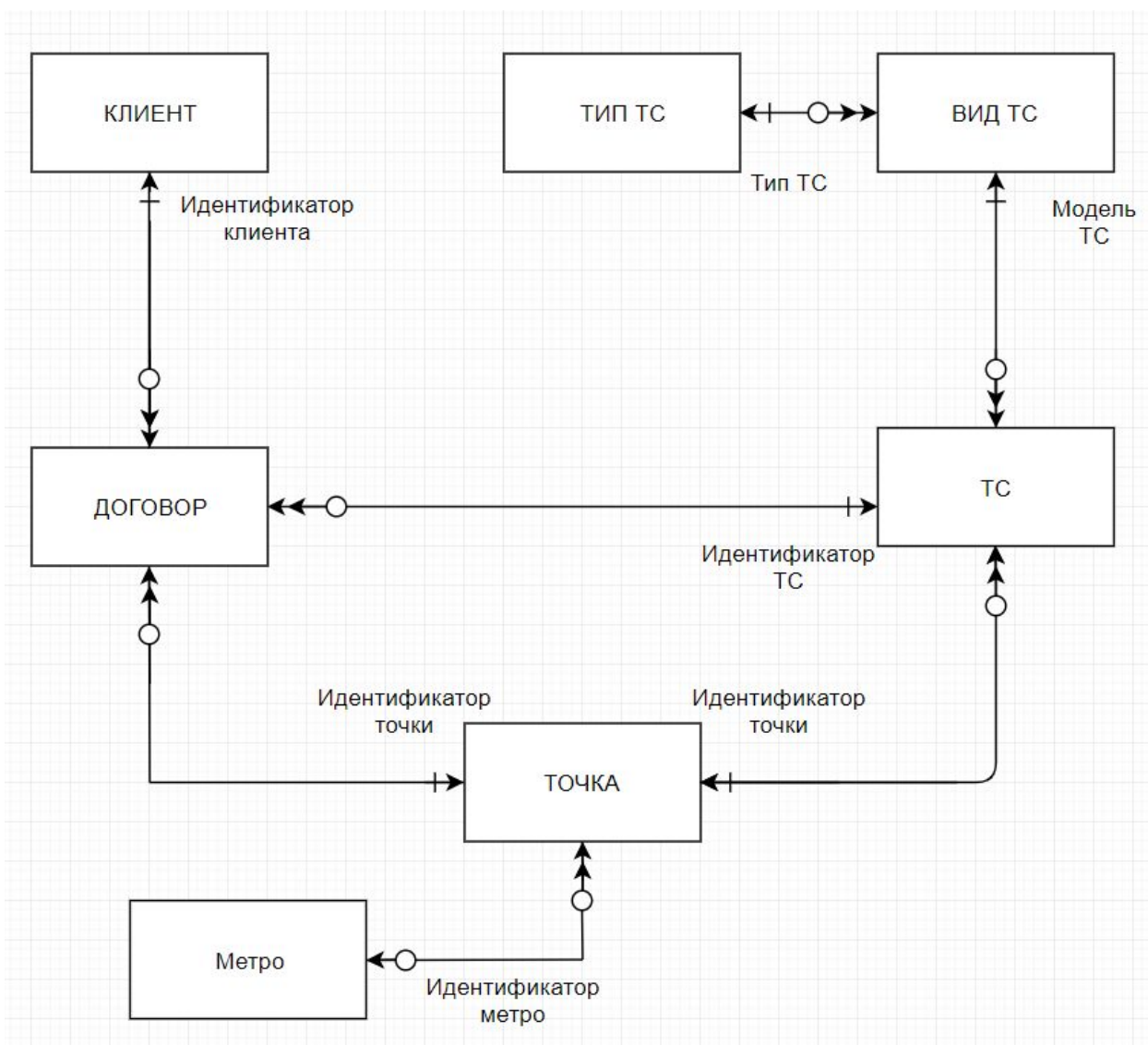


Рис. 3 Окончательная схема БД

Ограничение целостности

1. Атрибут Название типа в таблице «Типы ТС» может принимать одно из следующих значений: 'велосипед', 'электровелосипед', 'самокат', 'электросамокат'.
2. Атрибут Точные координаты широты/долготы в таблице «Точки парковки» должен заполняться по общепринятым стандартам, чтобы в дальнейшем использовать эту информацию для отображения их в мобильном приложении.
3. Дата окончания договора должна быть больше даты начала договора
4. Атрибут Итоговая стоимость в таблице «Поездка» высчитывается на основе длительности поездки и тарифа ТС, причем округляется не в сторону пользователя

Ограничение 4 нельзя реализовать в схеме отношений. Они будут реализовываться вручную, либо программно.

Описание групп пользователей и прав доступа

s – чтение данных (select);
i – добавление данных (insert);
u – модификация данных (update);
d – удаление данных(delete).

Таблицы	Группы пользователей (роли)	
	Клиент (пользователь)	Администратор приложения
Виды ТС	S	SIUD
ТС	S	SUID
Точки	S	SUID
Поездка	SI	SUID
Клиенты	SI	SUID
Метро	S	SUID

Любые запросы по поводу обслуживания транспорта будут проходить по запросам через администратора приложения. Клиент может просматривать только свой договор, select по паспорту. Комментарий клиент оставляет в поле *Примечание* таблицы «Поездка»

Создание таблиц

1. Отношение Клиент (Clients):

```
create table clients (
  f varchar(100) not null,
  io varchar(100) not null,
  pass numeric(4) primary key,
  phn numeric(11) not null unique,
  crd numeric (20) not null,
  email varchar (100) not null)
engine =innodb;
```

2. Отношение виды транспортных средств (Vtypes):

```
create table vtypes (
```

```

model varchar(100) primary key,
ts_type numeric(2) not null references tstype,
wght numeric(3) not null,
price numeric(3) not null,
dur_charge numeric(2))
engine =innodb;

```

3. **Отношение транспортные средства (Vehicles):**

```

create table vehicles (
id numeric(4) primary key,
charge numeric(3) not null,
id_p numeric(3) not null references parking,
comments varchar(100) not null,
ts_type numeric(2) not null references vtypes)
engine =innodb;

```

4. **Отношение типы транспортных средств(TStype):**

```

Create table tstype (
id_tts numeric (2) primary key,
typen varchar(100) not null check(typen in
('самокат', 'велосипед', 'электровелосипед',
'электросамокат'));
engine =innodb;

```

5. **Отношение точки (parking):**

```

create table parking (
id numeric(3) primary key,
gpsy numeric(6,4) not null,
gpsx numeric(6,4) not null,
id_m numeric(3) not null references metro,
cont numeric(2) not null)
engine =innodb;

```

6. **Отношение Метро (metro):**

```

Create table metro (
id numeric(3) primary key,
station varchar(100) not null)
engine =innodb;

```

7. **Отношение Поездка (trip):**

```

create table trip (
id numeric(6) primary key,
date_start date not null,
date_end date not null,
sumprice numeric(5) not null,
comments varchar(100),
id_ts numeric(4) not null references vehicles,
id_cl numeric(10) not null references clients,
id_p1 numeric(3) not null references parking,
id_p2 numeric(3) not null references parking,
distation numeric(3,2) not null,
check (date_start<date_end),
)
engine =innodb;

```

Создание представлений

1. Вывести идентификаторы клиентов, сумму оплаты за поездку и конечную точку парковки, которые совершали поездки вчера
Create or replace view YstrdCntrcts as
select id_cl, sumprice, id_p2
from trip
where date_start = (currrdate()-1) and date_end = (currrdate()-1);
2. Вывести электровелосипеды расположенные возле м. "Кропоткинская" :
create or replace view MetroNear as
select v.id, v.charge, v.ts_type
from metro m, vehicles v, vtypes vt, TStype t,
parking p
where m.id=p.id_m and v.id_p=p.id and
t.id_tts=vt.ts_type v.ts_type=vt.ts_type and
and t.typen='электровелосипед' and
m.station='Кропоткинская';
3. Вывести все элетровелосипеды и электросамокаты, которые могут проехать более 6 часов на одной зарядке (примем что идентификатор типа ТС для электровелосипеда равен 1, для электросамоката равен 2)
create or replace view onecharge as
select v.model, t.id
from vehicles v, vtypes vt, tstype t
where v.model=t.id_tts and vt. =t.and (vt.typen

- ```
= 'электровелосипед' or
vt.type='электросамокат') and dur_charge>6;
```
4. Вывести клиентов, которые не возвращали ТС более двух дней (для штрафных санкций)

```
create or replace view penalty as
select id_cl, sumrpice
from trip
where (date_end - date_start >2)
```
  5. Данные о свободных и занятых местах на точке (в нашем случае точка под номером 340)

```
create or replace view freepoint as
select p.cont, count(*)
from parking p, vehicles v
where p.id=v.id_p and v.id_p=340
group by p.cont ;
```

## Назначение прав доступа

Права доступа пользователей предоставляются с помощью команды GRANT. Рассмотрим для примера права клиента (*client\_1*), который является пользователем приложения. Права доступа к отношениям Vehicles, Types, Clients, Trip, Parking, Metro могут быть описаны следующим образом:

```
grant select on parking to client_1;
grant select on vehicles to client_1;
grant select on types to client_1;
grant select, insert on clients to client_1;
grant select, insert on trip to client_1;
grant select on metro to client_1;
```

Права доступа Администраторов приложения (**admin**) к представлению ко всем отношениям могут быть описаны следующим образом:

```
grant select, insert, update, delete on parking to admin;
grant select, insert, update, delete on vehicles to admin;
grant select, insert, update, delete on types to admin;
grant select, insert, update, delete on clients to admin;
```

```
grant select, insert, update, delete on trip to admin;
grant select, insert, update, delete on metro to admin;
```

### **Назначение индексов**

Создание индексов для внешних ключей:

```
create index id_p on vehicles (id_p);
create index ts_type on vehicles (ts_type);
create index id_m on parking (id_m);
create index id_ts on trip(id_ts);
create index id_cl on trip (id_cl);
create index id_p1 on trip (id_p1);
create index id_p2 on trip (id_p2);
```