

ФГБОУ ВО

«Заполярный государственный университет им. Н.М. Федоровского»

Кафедра __ИСиТ__

Специальность ИЭ-21

ОТЧЕТ

о выполнении лабораторной работы №1

Выполнил:

Быков В. В.

Дата:

« 3 » апреля 2023 г.

Лабораторная работа №1

Тема: проектирование реляционной базы данных PostgreSQL

Вариант 8. Таксопарк

Цель работы: Получение практических навыков проектирования реляционных баз данных и работы в CASE-средстве.

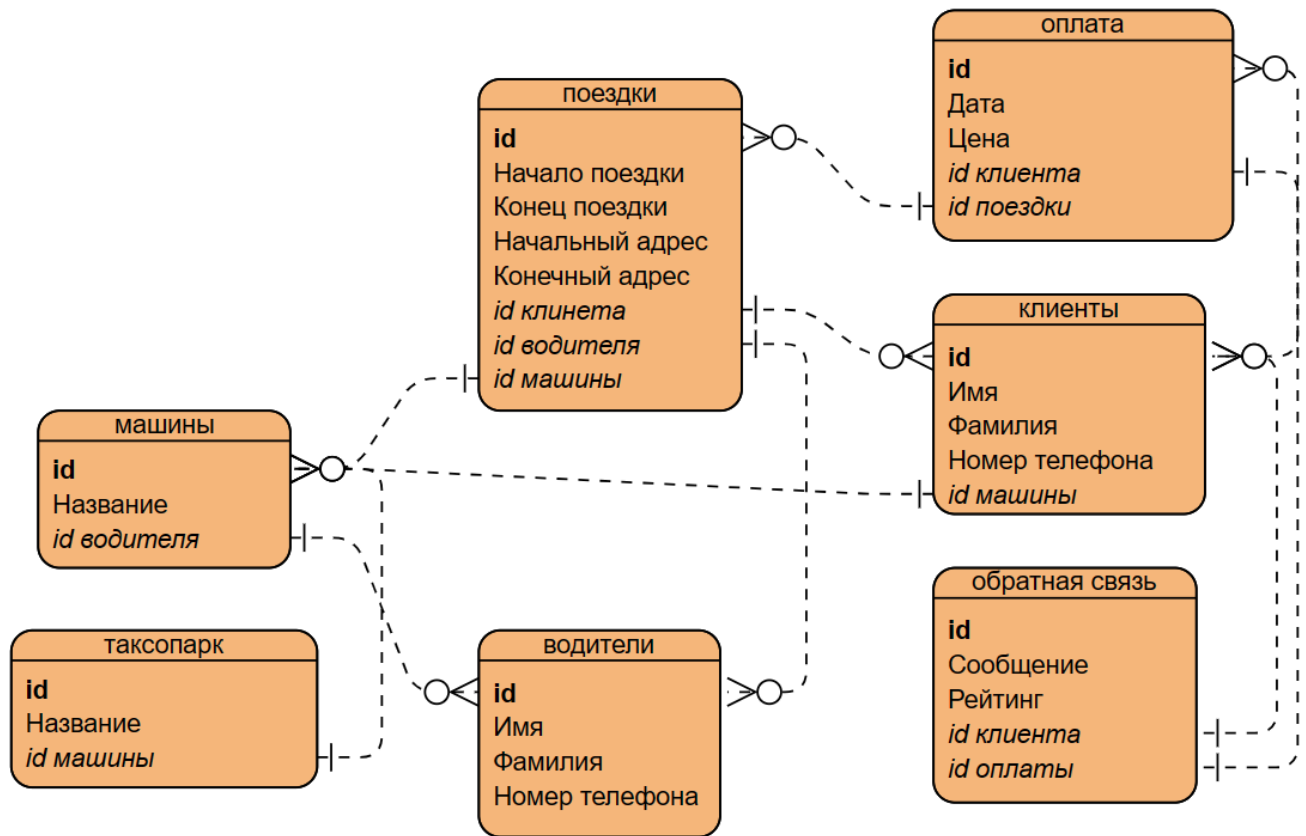
Задание: Спроектировать согласно варианту реляционную базу данных, используя методологию IDEF1X, для этого необходимо:

- построить ER-модель (логическую и физическую) с помощью CASEсредства, например dbdiagram.io или Lucidchart (не менее 7 сущностей),
- описать ограничения целостности (CONSTRAINT, UNIQUE)
- вывести программный код создания БД на языке целевой СУБД (прямое проектирование),
- перенести код в СУБД (выполнить),
- * внести изменения в схему БД и построить новую физическую модель измененной базы данных (обратное проектирование), если доступно создание ODBC источника данных

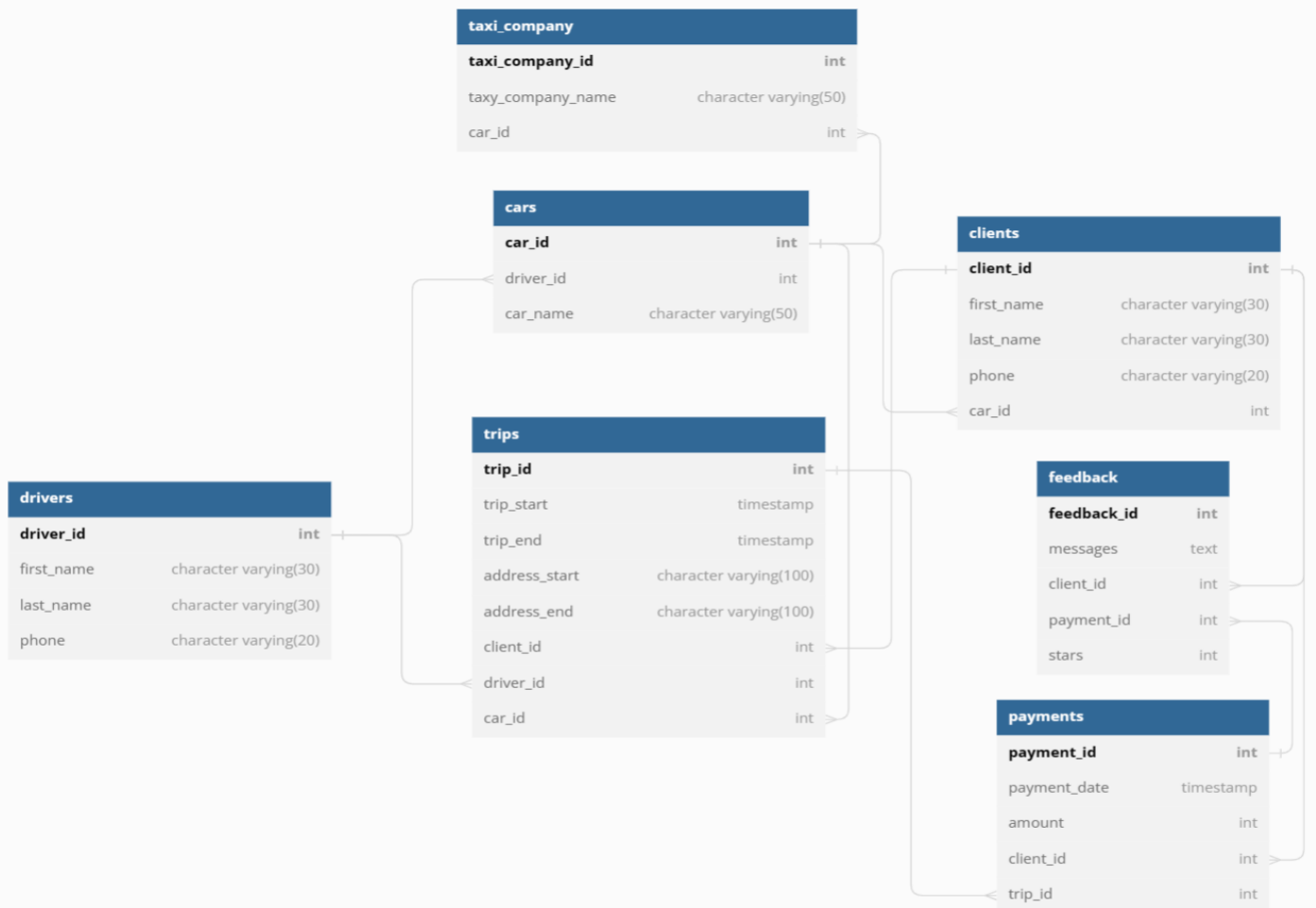
Описываемая база данных таксопарка, выделенные сущности:

- **Таксопарк(taxi_company)** - это организация, которая занимается предоставлением услуг такси. Она имеет свой парк машин, на которых работают водители.
- **Водители(drivers)** - это люди, которые управляют автомобилями компании и выполняют заказы клиентов.
- **Клиенты(clients)** - это люди, которые пользуются услугами таксопарка.
- **Машины(cars)** - это транспортные средства, которые используются для перевозки клиентов.
- **Поездка(trips)** - это процесс перевозки клиента из одного места в другое. Клиенты могут заказывать поездки на определенное время, место и с указанием пункта назначения.
- **Оплата(payments)** - это процесс, связанный с оплатой услуг таксопарка.
- **Обратная связь(feedback)** - это механизм, позволяющий клиентам оставлять свои комментарии и оценки о качестве услуг таксопарка.

Логическая модель базы данных:



Физическая модель базы данных:



Код создания базы данных:

```
CREATE TABLE drivers
(
    driver_id int PRIMARY KEY,
    first_name character varying(30),
    last_name character varying(30),
    phone character varying(20),
    CONSTRAINT driver_phone UNIQUE (phone)
);
CREATE TABLE cars
(
    car_id int PRIMARY KEY,
    driver_id int,
    car_name character varying(50),
    FOREIGN KEY (driver_id) REFERENCES drivers(driver_id)
);
CREATE TABLE taxi_company
(
    taxi_company_id int PRIMARY KEY,
    taxy_company_name character varying(50),
    car_id int,
    FOREIGN KEY (car_id) REFERENCES cars(car_id)
);
CREATE TABLE clients
(
    client_id int PRIMARY KEY,
    first_name character varying(30),
    last_name character varying(30),
    phone character varying(20),
    car_id int,
    CONSTRAINT client_phone UNIQUE (phone),
    FOREIGN KEY (car_id) REFERENCES cars(car_id)
);
CREATE TABLE trips
(
    trip_id int PRIMARY KEY,
    trip_start timestamp,
    trip_end timestamp,
    address_start character varying(100),
    address_end character varying(100),
    client_id int,
    driver_id int,
    car_id int,
    FOREIGN KEY (client_id) REFERENCES clients(client_id),
    FOREIGN KEY (driver_id) REFERENCES drivers(driver_id),
    FOREIGN KEY (car_id) REFERENCES cars(car_id)
);
CREATE TABLE payments
(
    payment_id int PRIMARY KEY,
    payment_date timestamp,
    amount int,
    client_id int,
    trip_id int,
    FOREIGN KEY (client_id) REFERENCES clients(client_id),
    FOREIGN KEY (trip_id) REFERENCES trips(trip_id)
);
CREATE TABLE feedback
(
    feedback_id int PRIMARY KEY,
    messages text,
    client_id int,
    payment_id int,
    stars int,
    FOREIGN KEY (client_id) REFERENCES clients(client_id),
    FOREIGN KEY (payment_id) REFERENCES payments(payment_id)
);
```

Скриншот выполнения кода:

Создание базы данных:

The screenshot shows a PostgreSQL client interface with the following components:

- Header:** A toolbar with icons for connection, save, edit, filter, and execution. The connection string is `taxi/postgres@PostgreSQL 15`.
- Query Editor:** A text area containing SQL code for creating a database schema. The code is as follows:

```
51 amount int,  
52 client_id int,  
53 trip_id int,  
54 FOREIGN KEY (client_id) REFERENCES clients(client_id),  
55 FOREIGN KEY (trip_id) REFERENCES trips(trip_id)  
56 );  
57 CREATE TABLE feedback  
58 (  
59     feedback_id int PRIMARY KEY,  
60     messages text,  
61     client_id int,  
62     payment_id int,  
63     stars int,  
64     FOREIGN KEY (client_id) REFERENCES clients(client_id),  
65     FOREIGN KEY (payment_id) REFERENCES payments(payment_id)  
66 );  
67
```
- Execution Results:** A tabbed interface with three tabs: "Data Output", "Сообщения" (Messages), and "Notifications". The "Сообщения" tab is active, displaying the message "CREATE TABLE".
- Status Bar:** A message at the bottom states: "Запрос завершён успешно, время выполнения: 89 мсек." (Query completed successfully, execution time: 89 ms).