# **Федеральное государственное автономное образовательное учреждение высшего образования**

# **«Национальный исследовательский университет «Высшая школа экономики» Факультет компьютерных наук**

# 

# 

# 

# 

# **О Т Ч Е Т**

# **по дисциплине**

# **«Теория Баз Данных»** **Perevozka**

# 

# 

# 

# 

# **Выполнили студенты 184 гр**

# **Юдин Виктор**

# **Зюляев Илья**

# 

# **(ФИО)**

# **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# 

# ***(подпись)***

# 

# 

# 

# 

# **Проверил:**

# 

# *(должность, ФИО руководителя практики)*

# 

# 

# *17.12.2020*

# *(дата)*

# 

# 

# 

# 

# **2020 год**

## Содержание

[1. Описание предметной области](#_rizvf83fj4qk)

[2. Описание процесса построения инфологической модели](#_w99e5fqx8im9)

[ER-диаграмма (нотация Чена)](#_2raixz456qtk)

[3. Реляционная модель](#_yb2527dwdp5t)

[Процесс перехода к реляционной модели](#_lbhxthotjca8)

[TR-диаграмма (нотация Чена)](#_7gvkogy8nijh)

[Словарь данных:](#_vau20oomufon)

[4. Развертывание БД](#_50tpzz44chlk)

[DDL код создания:](#_yarpwia9n9f2)

[Пример DML операторов вставки:](#_b8gfc9egubfz)

[Пример триггера:](#_uekc5dx8uwzp)

[5. Разработка клиентского приложения](#_i00fspfmunah)

[Архитектура](#_dqhzn0y4z4f9)

[Сценарии использования](#_ufns6eff5rul)

[Организация доступа к данным и Интерфейс с пользователем](#_2rlal5ojolkd)

[Примеры запросов:](#_pj7ggffxvhc1)

[6. Заключение](#_bykw0jboud7s)

[Объемные характеристики разработки](#_2lkq7ch1ycg1)

[Авторский вклад и комментарии по выполнению проекта](#_dy3fxzuj48kd)

[7. Источники](#_5nxdh515gcpk)

# 1. Описание предметной области

В качестве предметной области мы рассмотрели такой кейс: Крупная компания, занимающаяся грузовыми перевозками. Специализация – перевозка сборных грузов по всей России. Услуги компании пользуются спросом благодаря минимальным срокам доставки груза, широкой филиальной сети, высокому уровню сервиса и оптимальной ценовой политике. Доставка осуществляется посредством автотранспорта. У вас есть собственные автопарки. Для перевозки принимаются как небольшие, так и крупногабаритные грузы весом до 20 тонн. Компания позволяет следить за перевозкой груза т.к. груз зачастую перевозится через промежуточные пункты сортировки.

В качестве итогового визуального решения (интерактивного) реализовали полноценныйы вебсервис с помощью Angular + Django который в онлайне позволяет не только менять базу данных, а и имитировать работу компании (перевозку грузов) и следить за путем перевозки.

# 2. Описание процесса построения инфологической модели

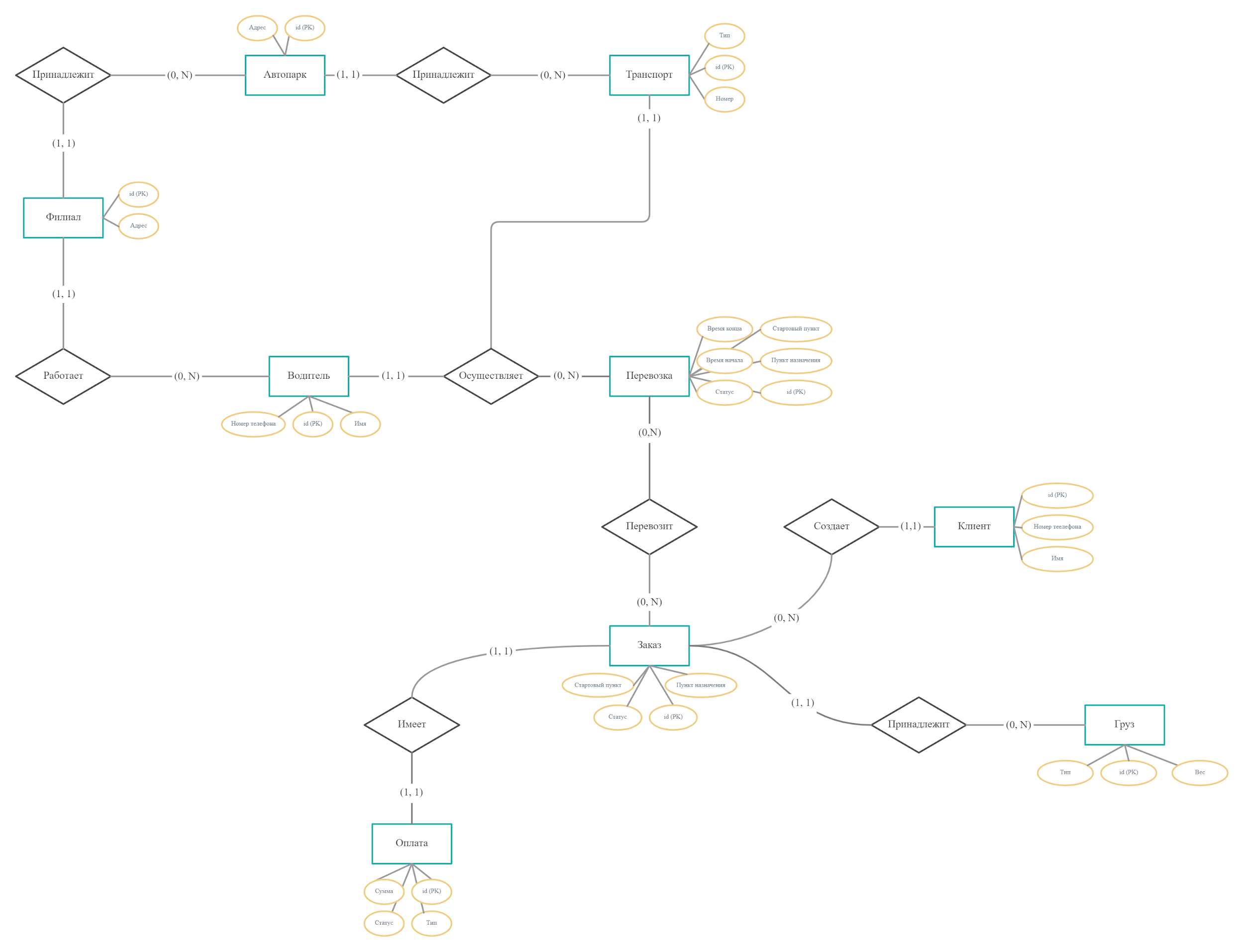
В процессе создания инфологической модели было выделено 9 сущностей. Для каждой из них заведем свой уникальный идентификатор. Уникальность подразумевается в рамках одной сущности, между объектами двух разных сущностей ключи могут совпадать. Поле “ключ” в описании сущностей опустим для простоты. А ссылки на другие таблицы будут видны из связей в дальнейшем. Давайте посмотрим на сущности и их атрибуты которые у нас получились:

* Сущность Клиент описывает клиента компании и включает в себя поля: имя клиента, номер телефона.
* Сущность Заказ описывает заказ который оставил клиент и содержит сущности: статус заказа, ключ клиента (который оставил заказ), пункт отправления и пункт назначения.
* Сущность Груз описывает груз который будет доставлен в рамках одного из заказа. В качестве атрибутов содержит: название груза, вес, тип.
* Сущность Оплата содержит информацию о оплате заказа. Содержит атрибуты: сумма, статус, тип.
* Сущность Перевозка - самая интересная сущность нашего проекта. Она содержит информацию о перевозке заказов между двумя пунктами. Т.е. перевозка осуществляется между двумя пунктами и некоторое кол-во заказов перемещается между этими пунктами (т.е. общий процесс доставки разбивается на такие мелкие перевозки между двумя соседними пунктами). Данная сущность содержит атрибуты: время отправки, время прибытия, стартовый пункт, пункт назначения, статус.
* Сущность Водитель описывает водителя, который занимается перевозкой заказов и включает в себя поля: имя, номер телефона.
* Сущность Филиал описывает филиал нашей компании и содержит поле адрес.
* Сущность Автопарк содержит информацию об автопарках и имеет единственное поле адрес.
* Сущность Транспорт содержит информацию о транспорте с помощью которого осуществляются перевозки. Имеющиеся атрибуты: серийный номер, тип.

Также в нашей базе данных представлены три типа вида связей: один-ко-многим, многие-ко-многим, один-к-одному (Одна из связей при этом тернарная). Рассмотрим их подробнее:

* Связь Создает (один ко многим) между Клиентом и Заказом означает, что клиент мог создать много заказов, а у заказа всегд есть определенный заказчик.
* Связь Принадлежит (один ко многим) между Заказом и Грузом показывает, что в заказе может быть много грузов, а конкретный груз содержится в конкретном заказе.
* Связь Имеет (один к одному) между Оплатой и Заказом показывает, что у заказа есть единственная информация об оплате и конкретная оплата происходит по определенному заказу.
* Связь Перевозит (многие ко многим) между Заказом и Перевозкой означает, что у заказа может быть несколько перевозок т.е. заказ побывает в промежуточных пунктах перед тем как прибыть на место назначения, при этом в одной перевозке может быть несколько заказов ведь за раз водитель перевозит сразу много заказов между двумя пунктами.
* Тернарная связь Осуществляется (один к одному ко многим) между Транспортом, Водителем и Перевозкой показывает, что перевозка осуществляется конкретным водителем на конкретном транспорте, однако в общем какой-то водитель и какой-то транспорт в разное время могут осуществлять разные перевозки.
* Связь Работает (один ко многим) между Филиалом и Водителем означает, что в филиале работает много водителей, а водитель числится за конкретным филиалом.
* Связь Принадлежит (один ко многим) между Филиалом и Автопарком означает, что у филиала может быть много автопарков, а определенный автопарк управляется определенным филиалом.
* Связь Принадлежит (один ко многим) между Автопарком и Транспортом показывает, что у автопарка много различного транспорта, а конкретный транспорт числится за конкретным автопарком.

## ER-диаграмма (нотация Чена)



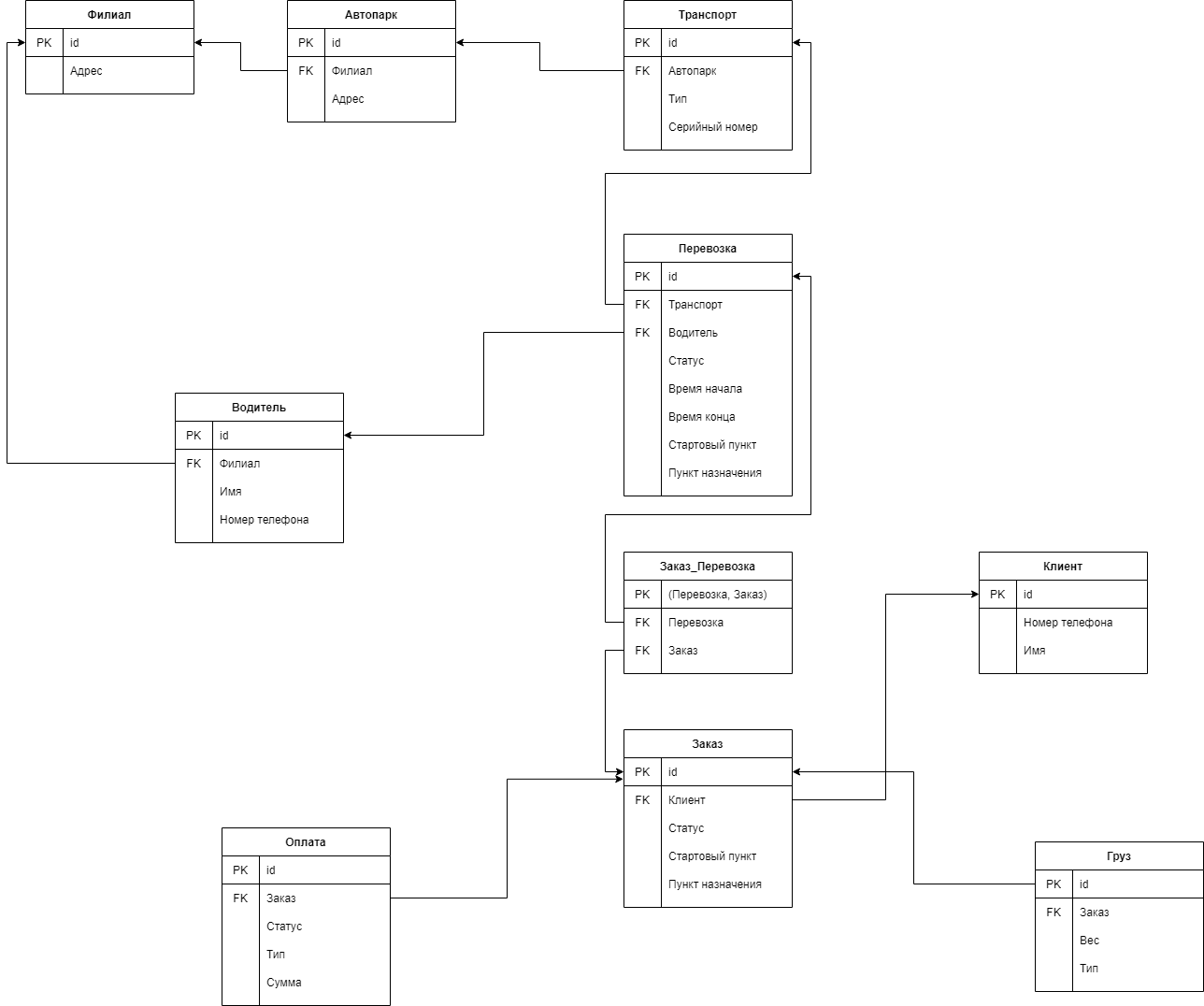
Ссылка на ER-диаграмму на GitHub: [ссылка](https://github.com/yudinvictor/DBMS/blob/master/ER.jpg)

# 3. Реляционная модель

## Процесс перехода к реляционной модели

При переходе к реляционной модели мы установили ограничения на связи. Для каждой связи в таблицах были установлены внешние ключи и ограничения множественности. Были добавлены таблицы пар ключей для реализации вида многие-ко многим. При связи один-к-одному внешний ключ помещался только в одну из таблиц (чтобы не было избыточности).

## TR-диаграмма (нотация Чена)



## Словарь данных:

Филиал, Автопарк, Водитель, Транспорт, Оплата, Заказ, Груз, Клиент, Перевозка

# 4. Развертывание БД

## DDL код создания:

CREATE TABLE branch (

id INT NOT NULL PRIMARY KEY,

address text NOT NULL

);

CREATE TABLE car\_park (

id INT NOT NULL PRIMARY KEY,

address text NOT NULL,

branch\_id INT NOT NULL REFERENCES branch (id)

);

CREATE TABLE client (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

phone\_number VARCHAR(255) NOT NULL

);

CREATE TABLE driver (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

phone\_number VARCHAR(255) NOT NULL,

branch\_id INT NOT NULL REFERENCES branch (id)

);

CREATE TABLE order (

id INT NOT NULL PRIMARY KEY,

status VARCHAR(255) NOT NULL,

departure\_address text NOT NULL,

destination\_address text NOT NULL,

client\_id INT NOT NULL REFERENCES client (id)

);

CREATE TABLE transport (

id INT NOT NULL PRIMARY KEY,

type VARCHAR(255) NOT NULL,

number VARCHAR(255) NOT NULL,

car\_park\_id INT NOT NULL REFERENCES car\_park (id)

);

CREATE TABLE shipping (

id INT NOT NULL PRIMARY KEY,

start\_time timestamp NOT NULL,

stop\_time timestamp NULL,

departure\_address text NOT NULL,

destination\_address text NOT NULL,

status VARCHAR(255) NOT NULL,

driver\_id INT NOT NULL REFERENCES driver (id) ,

transport\_id INT NOT NULL REFERENCES transport (id)

);

CREATE TABLE shipping\_orders (

shipping\_id INT NOT NULL REFERENCES shipping (id) ,

order\_id INT NOT NULL REFERENCES "order" (id) ,

PRIMARY KEY (shipping\_id, order\_id)

);

CREATE TABLE payment (

id INT NOT NULL PRIMARY KEY,

amount FLOAT NOT NULL,

type VARCHAR(255) NOT NULL,

status VARCHAR(255) NOT NULL,

order\_id INT NOT NULL UNIQUE REFERENCES "order" (id)

);

CREATE TABLE cargo (

id INT NOT NULL PRIMARY KEY,

weight FLOAT NOT NULL,

type VARCHAR(255) NOT NULL,

order\_id INT NOT NULL REFERENCES "order" (id)

);

## Пример DML операторов вставки:

INSERT INTO Order

(status, departure\_address, destination\_address, client)

VALUES ('Создан', 'Москва', 'Сочи',

(SELECT id FROM Client WHERE name = 'Вася' LIMIT 1))

INSERT INTO Payment(amount, type, status, order)

VALUES (1000, 'Apple Pay', 'Ожидание оплаты', 1)

## Пример триггера:

CREATE TRIGGER tr\_order ON Order

AFTER INSERT

DECLARE

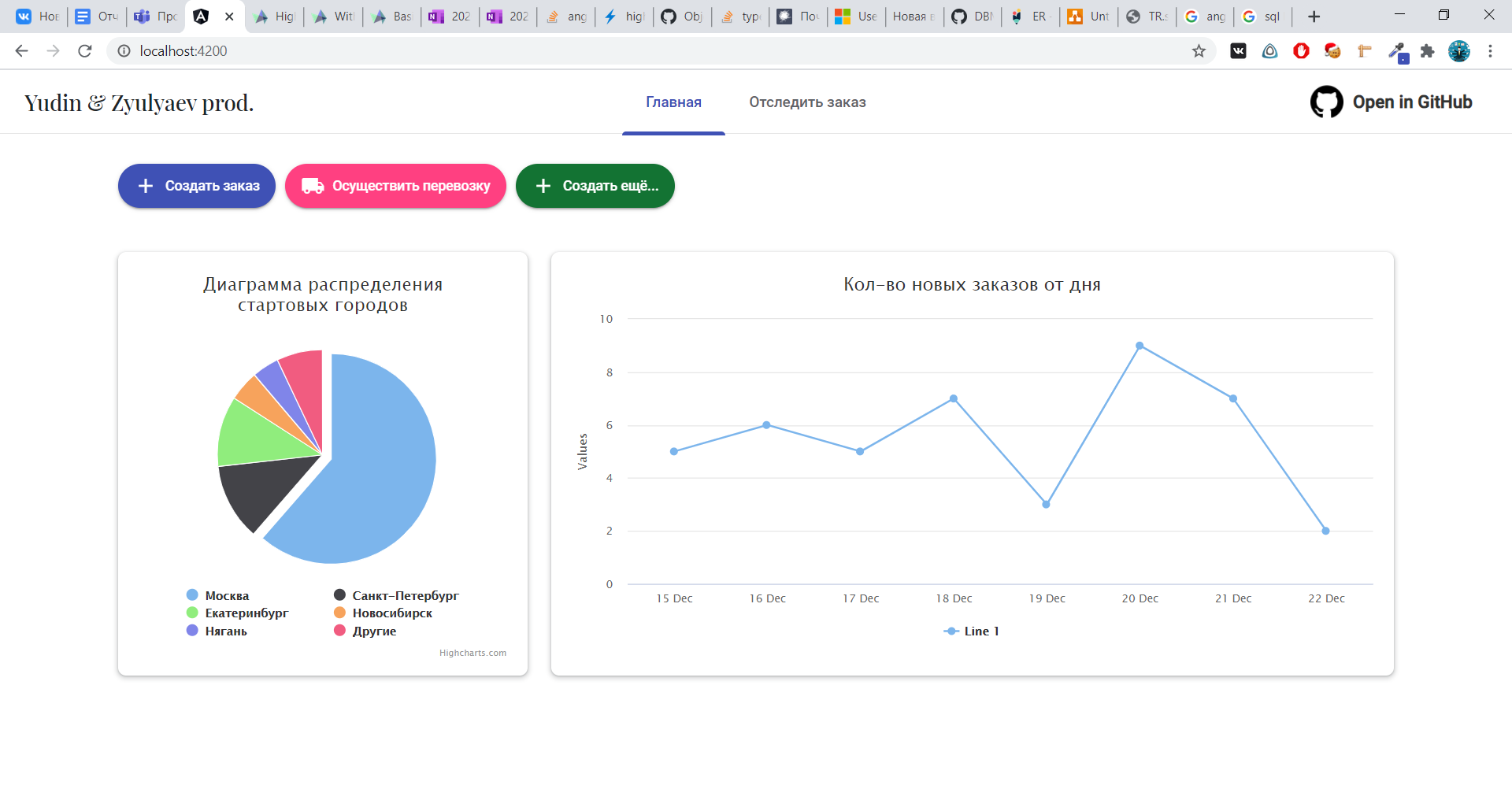
BEGIN

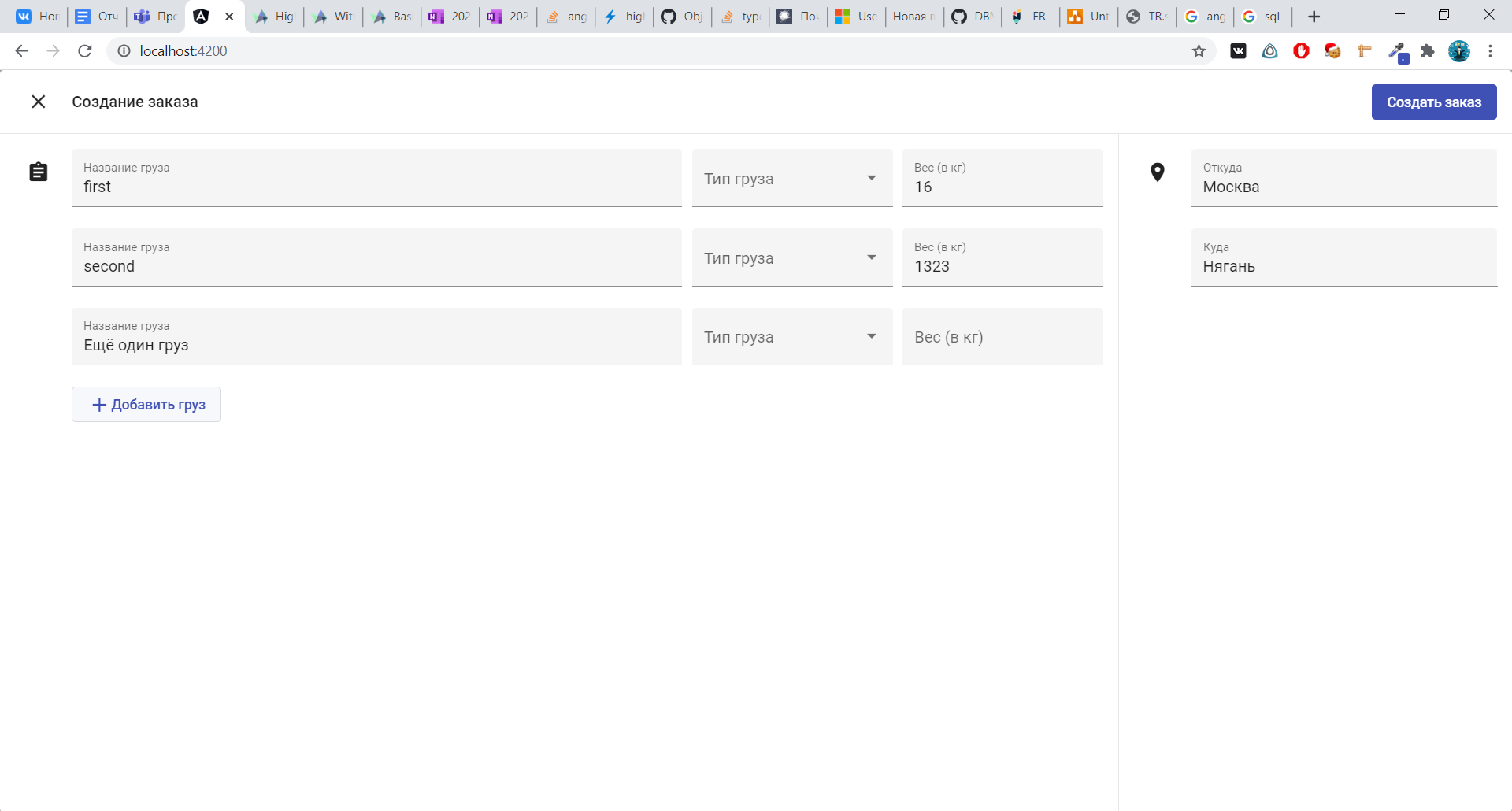
INSERT INTO Payment(amount, type, status, order)

VALUES (1000, 'Тип не выбран', 'Ожидание оплаты', (SELECT id FROM inserted))

END

# 5. Разработка клиентского приложения





## Архитектура

В основе нашего вебсайта лежит Angular 10 + Django. При разработке использовалась SQLite, при развертывании PostrgreSQL.

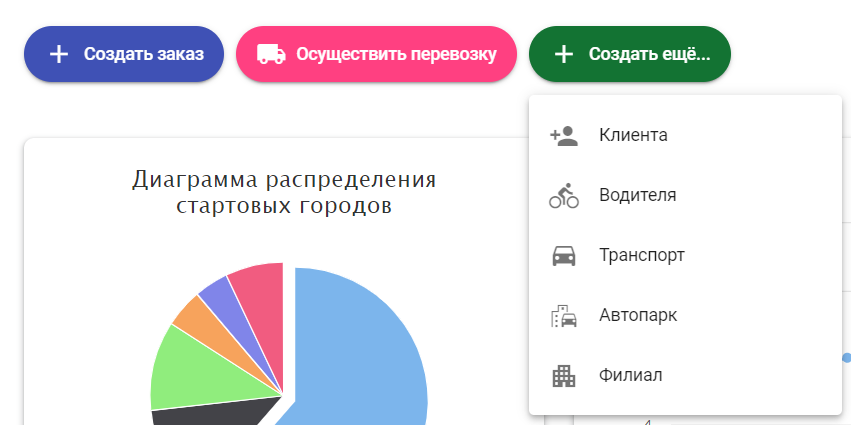
## Сценарии использования

Мы создали основу для полноценного SaaS. Уже сейчас можно добавлять новых водителей, транспорт, создавать заказы и управлять перевозками, отслеживать выбранный заказ.

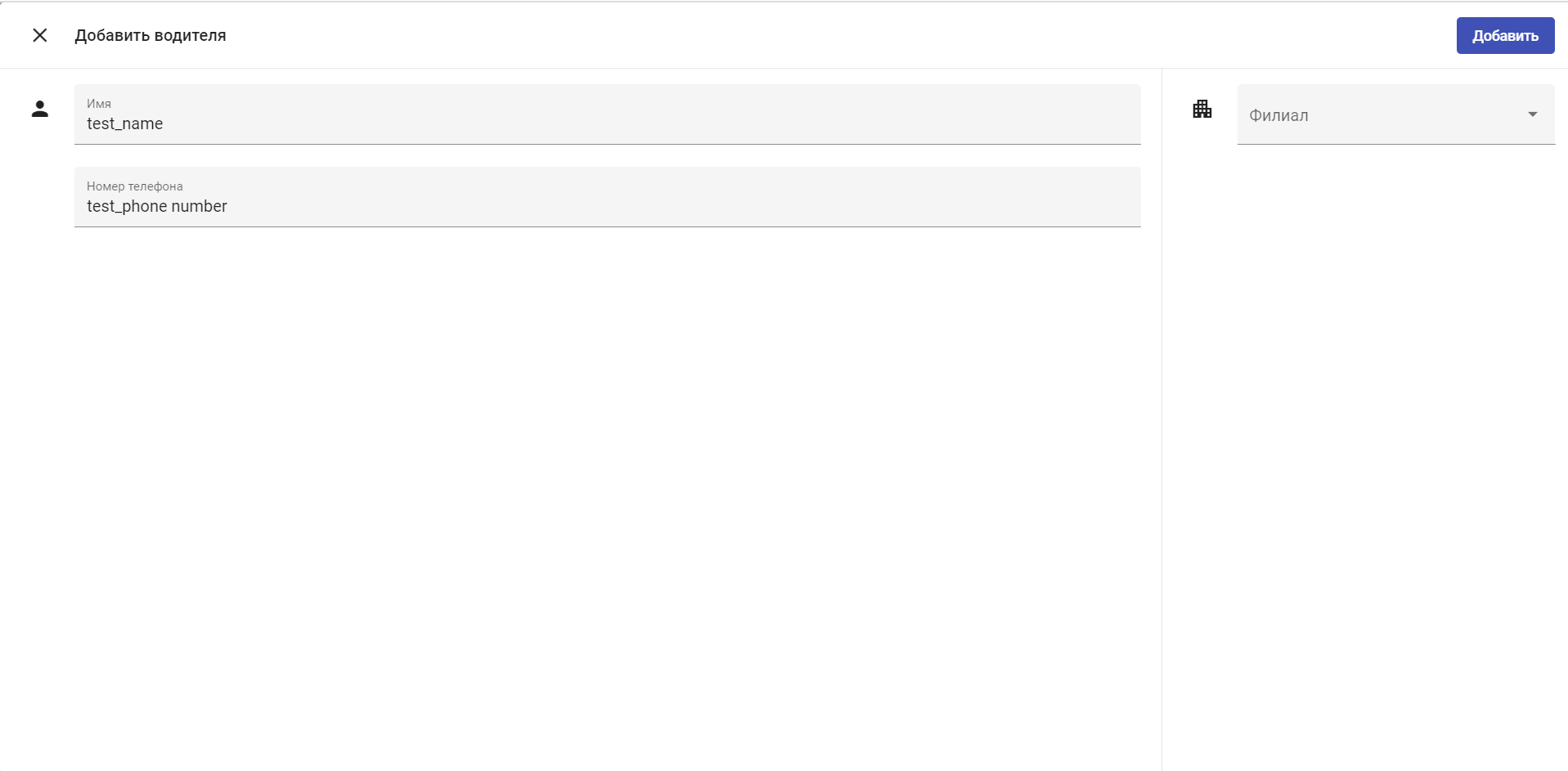
На презентации мы захостим сайт для общего доступа.

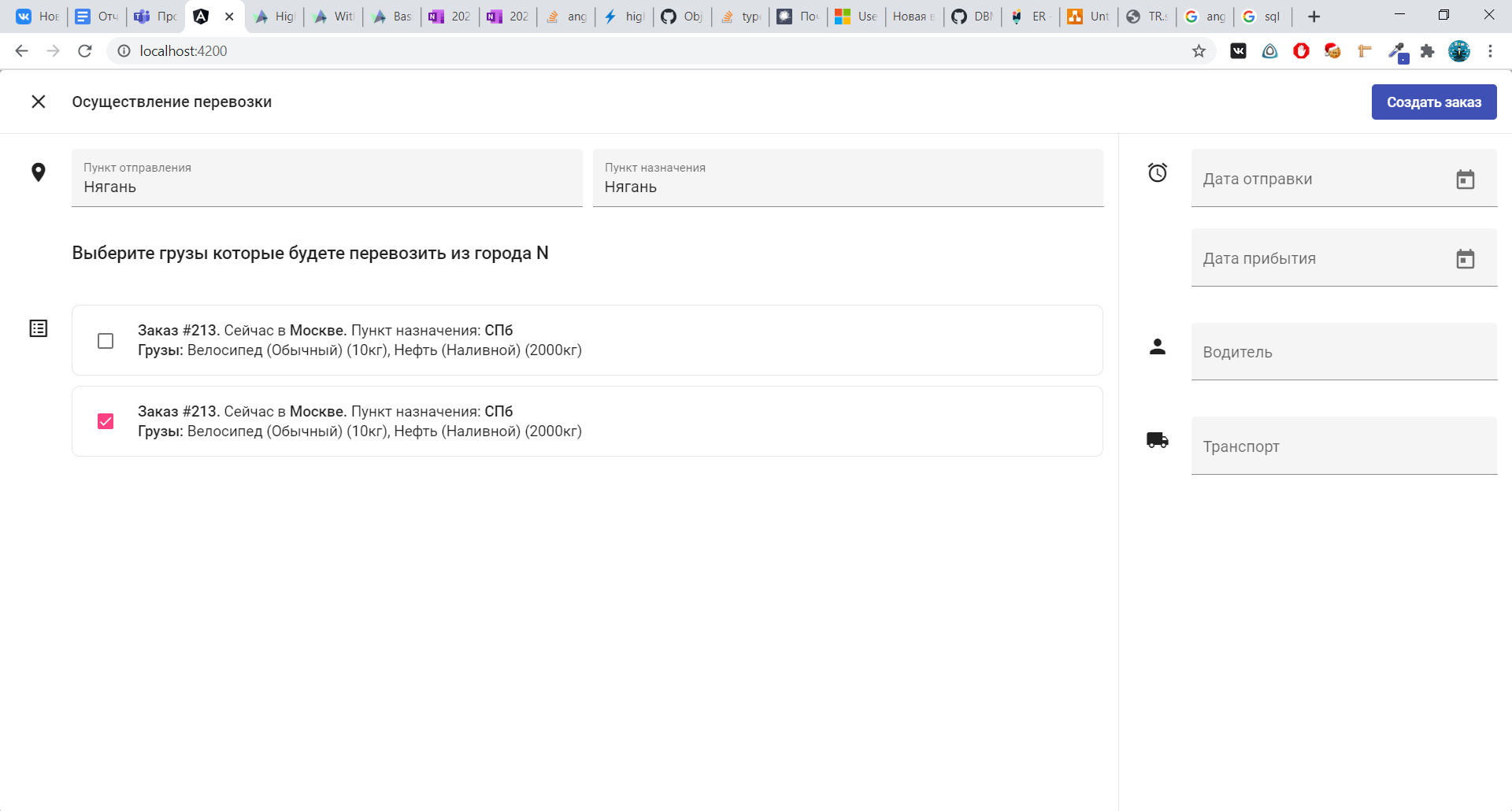
## Организация доступа к данным и Интерфейс с пользователем

Мы позволяем получить доступ не только с помощью написанного красивого UI интерфейса, но и извне предоставляя api (Однако для получения доступа необходимо заранее с нами связаться чтобы мы могли настроить порты и cors). На данном этапе мы позволяем создавать представленные сущности.

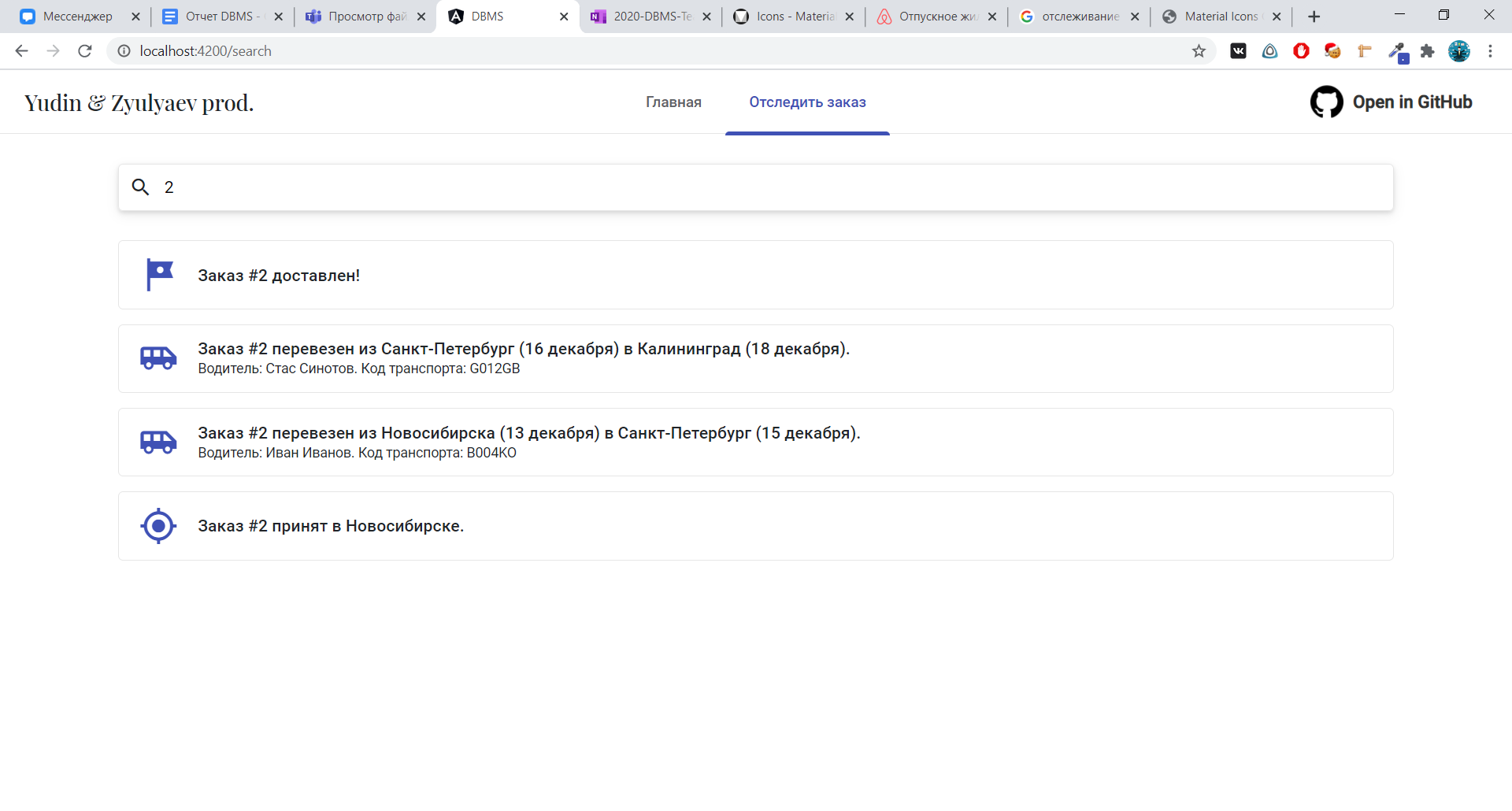


Все происходит динамически, без перезагрузки страницы:





На главном экране вы можете создать новые сущности, указав все необходимые параметры. Можно посмотреть на диаграммы которые в онлайн режиме обновляются. (К демонстрации мы расширим кол-во диаграмм и статистик и добавим фильтрацию).



## 

## Примеры запросов:

По заказу (ORDER\_ID) находим все перевозки в которых он участвовал

SELECT \* FROM shipping INNER JOIN shipping\_orders ON

(shipping.id = shipping\_orders.shipping\_id)

WHERE shipping\_orders.order\_id = ORDER\_ID

По клиенту (CLIENT\_ID) находим все заказы, которые когда либо были в Москве

SELECT \* FROM order INNER JOIN shipping\_orders ON

(order.id = shipping\_orders.order\_id) INNER JOIN shipping ON

(shipping\_orders.shipping\_id = shipping.id)

WHERE (order.client\_id = CLIENT\_ID AND shipping.departure\_address = 'Москва')

По водителю (DRIVER\_ID) находим все грузы, больше 100 кг, которые он перевозил

SELECT \* FROM cargo INNER JOIN order ON (cargo.order\_id = order.id)

INNER JOIN shipping\_orders ON (order.id = shipping\_orders.order\_id)

INNER JOIN shipping ON (shipping\_orders.shipping\_id = shipping.id)

WHERE (cargo.weight >= 100 AND shipping.driver\_id = DRIVER\_ID)

# 6. Заключение

В итоге мы сделали основу для будущего SaaS с возможностью создания различных объектов, отслеживания грузов и наглядного просмотра статистики на основе Angular + Material + Highcharts + Django + PostgreSQL.

## Объемные характеристики разработки

Как только вы зайдете и пощупаете сайт - вы поймете масштаб работы) Говорить о кол-ве строк глупо потому что наш проект весит 500Мб без учета базы данных). Считаем что непомерный вклад в работу внесли Юдин Виктор и Зюляев Илья. Индусы на аутсорсе почти ничего не внесли.

## Авторский вклад и комментарии по выполнению проекта

Юдин Виктор и Зюляев Илья:

* Почти все: от создания моделей и проектирования БД до написания фронта и бэка.

Индусы на аутсорсе:

* Все остальное

# 7. Источники

1. <https://teams.microsoft.com/_?tenantId=21f26c24-0793-4b07-a73d-563cd2ec235f#/pdf/viewer/teams/https:~2F~2Feduhseru.sharepoint.com~2Fsites~2F2020-DBMS-Team~2FShared%20Documents~2FGeneral~2FShare~2FBooks~2F%5B2016%5D%20Mancas%20-%20Conceptual%20Data%20Modeling%20and%20Database%20Design.pdf?threadId=19:bb25be41aac44ba29675f23b524862f4@thread.tacv2&baseUrl=https:~2F~2Feduhseru.sharepoint.com~2Fsites~2F2020-DBMS-Team&fileId=F4EED0F3-2297-466D-8DC9-E54709BED90A&viewerAction=view> - Лучшая книжка
2. Лучше книжки выше только Tinkoff Fintech