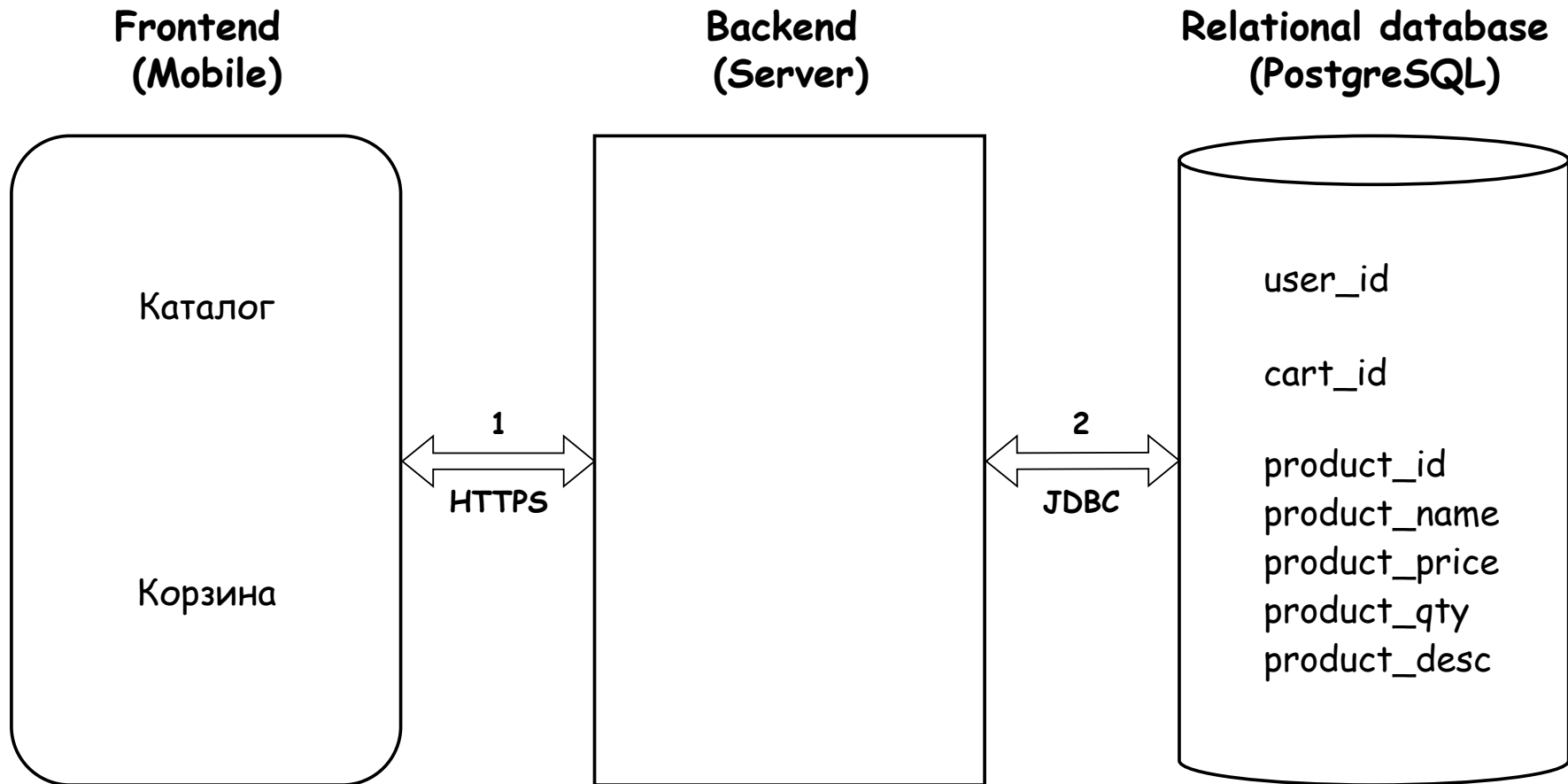
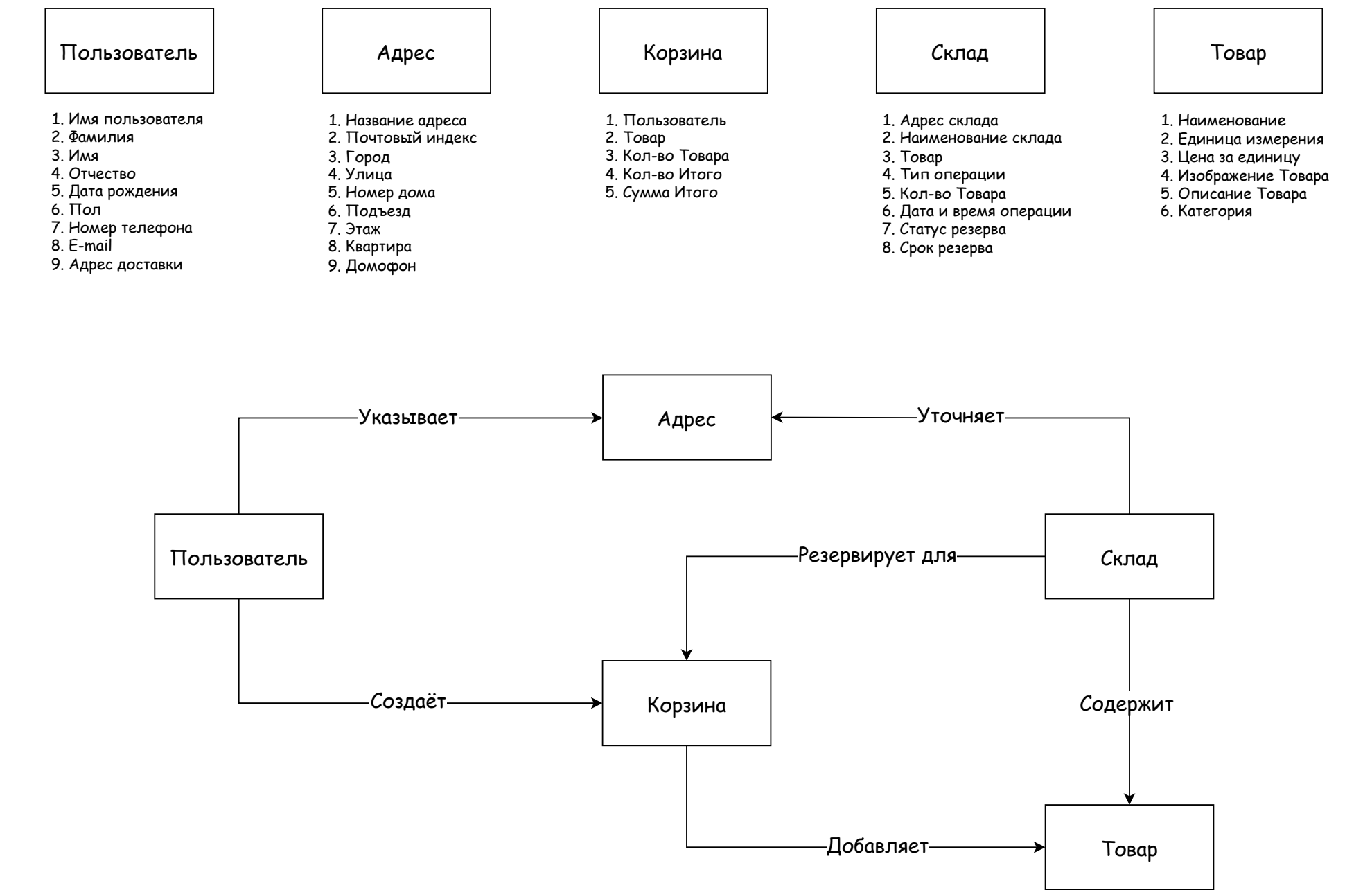


## Задание 5. Архитектура



# Задание 6. Моделирование данных



Объект: Пользователь (родительская сущность)

Атрибут	Описание
username	Имя пользователя
surname	Фамилия
name	Имя
patronymic	Отчество
birthdate	Дата рождения
gender	Пол
phone_number	Контактный номер телефона
email	Адрес электронной почты
shipping_address	Адрес (дочерняя сущность)

Объект: Корзина (дочерняя сущность)

Атрибут	Описание
user	Пользователь (родительская сущность)
product	Товар (дочерняя сущность)
item_qty	Кол-во выбранного Товара в Корзине
total_qty	Общее кол-во Товаров в Корзине
total_cost	Общая стоимость Корзины

Объект: Товар (дочерняя сущность)

Атрибут	Описание
name	Наименование
unit	Единица измерения
price	Цена за единицу
photo	Изображение Товара
desc	Описание Товара
category	Группировка Товаров по смыслу

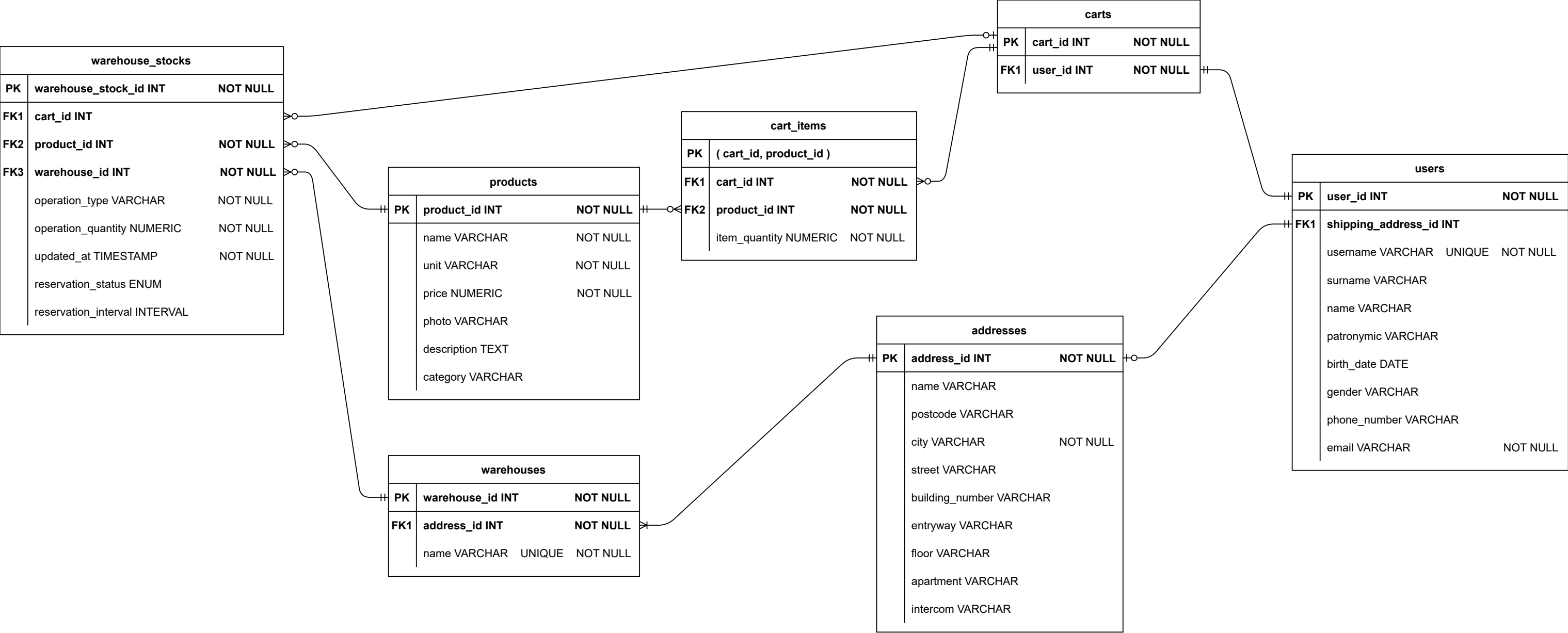
Объект: Адрес (дочерняя сущность)

Атрибут	Описание
name	Название адреса
postcode	Почтовый индекс
city	Город
street	Улица
building_number	Номер дома
entryway	Подъезд
floor	Этаж
apartment	Квартира
intercom	Домофон

Объект: Склад (родительская сущность)

Атрибут	Описание
warehouse_address	Адрес (дочерняя сущность)
name	Наименование склада
cart	Корзина (дочерняя сущность)
product	Товар (дочерняя сущность)
operation_type	Тип операции
operation_qty	Количество Товара
updated_at	Дата и время операции
reservation_status	Статус резерва
reservation_interval	Срок резерва

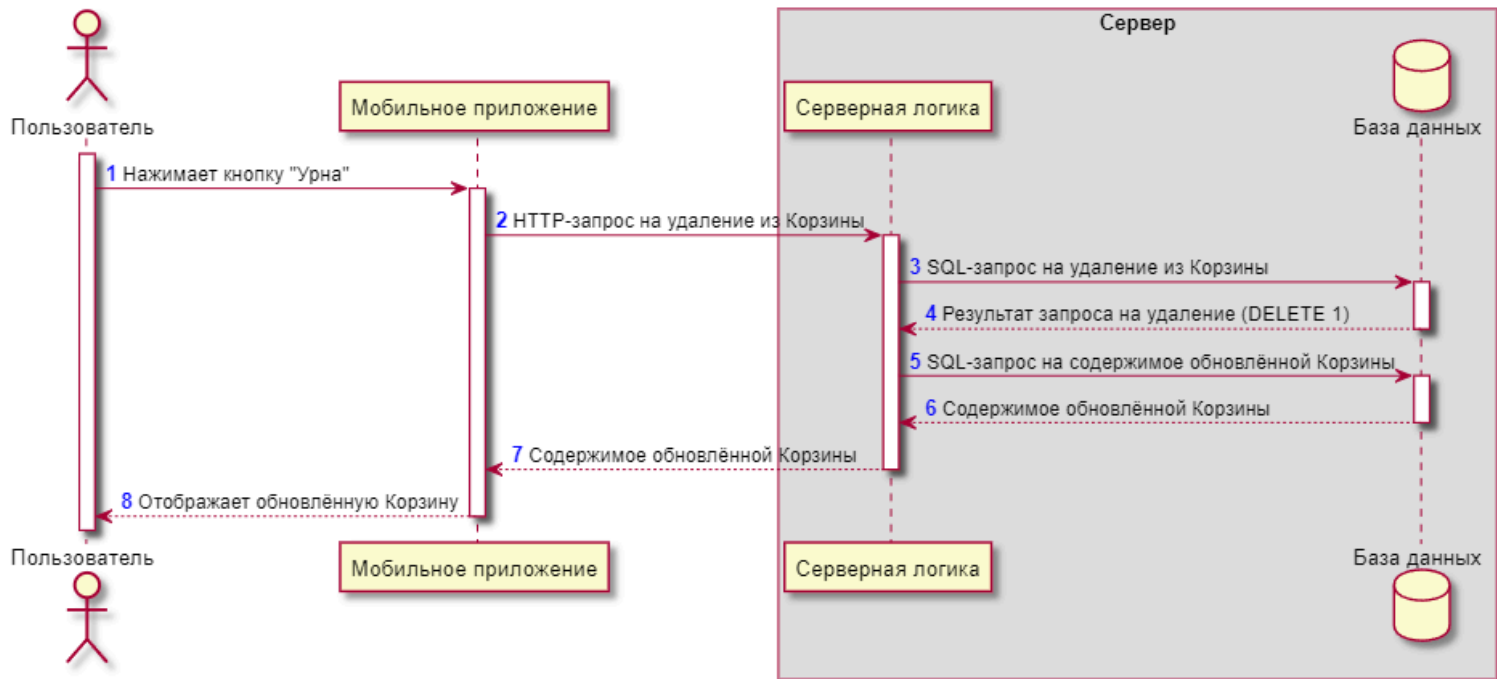
# Задание 7.1. ER-диаграмма



# Задание 7.2. SQL-запросы

# Задание 8. Sequence-диаграмма

[МегаМаркет]: Удаление товара из корзины



Шаг	Описание
1	Пользователь, находясь на экране Корзины, решил удалить Товар из Корзины и нажимает кнопку "Урна" рядом с Товаром
2	Мобильное приложение отправляет HTTP-запрос Серверу на удаление Товара из Корзины
3	Сервер отправляет SQL-запрос в Базу данных на удаление записи из таблицы <b>cart_items</b> с последующей отменой резерва соответствующего Товара в таблице <b>warehouse_stocks</b>
4	База данных возвращает Серверу SQL-ответ с успешным удалением записи (DELETE 1)
5	Сервер отправляет SQL-запрос в Базу данных на содержимое обновлённой Корзины
6	База данных возвращает Серверу SQL-ответ с содержимым обновлённой Корзины
7	Сервер возвращает Мобильному приложению HTTP-ответ с содержимым обновлённой Корзины (HTTP 200 OK)
8	Мобильное приложение отображает Пользователю экран с содержимым обновлённой Корзины

# Задание 10. API

## GET /v1/users/{userId}/cart

Получение содержимого Корзины пользователя

### Request

Название параметра	Местонахождение	Тип данных	Описание	Обязательность параметра
userId	path	number	Уникальный ID Пользователя	Да

### Response

Название параметра	Местонахождение	Тип данных	Описание	Обязательность параметра
cartId	body	number	Уникальный ID Корзины	Да
cartItems	body	array	Массив Товаров	Да
cartItem	body	object	Объект Товара	Нет
productId	body	number	Уникальный ID Товара	Да
name	body	string	Наименование Товара	Да
unit	body	string	Единица измерения Товара	Да
price	body	number	Цена за единицу Товара	Да
photo	body	string	Изображение Товара	Нет
description	body	string	Описание Товара	Нет
category	body	string	Категория Товара	Нет
itemQuantity	body	number	Количество Товара в Корзине	Да

Response code < 200 OK >

## POST /v1/users/{userId}/cart/item

Добавление выбранного Товара в Корзину пользователя

### Request

Название параметра	Местонахождение	Тип данных	Описание	Обязательность параметра
userId	path	number	Уникальный ID Пользователя	Да
productId	body	number	Уникальный ID Товара	Да
itemQuantity	body	number	Количество добавляемого Товара	Да

### Response

Название параметра	Местонахождение	Тип данных	Описание	Обязательность параметра
message	body	string	Сообщение о результате выполнения	Да

Response code < 201 Created >

# Задание 11. Swagger

The image shows the Swagger Editor interface. On the left, a YAML file is being edited, defining an API for 'МегаМаркет API'. The file includes metadata like version (1.1.1) and a base URL, followed by two endpoints: a GET endpoint for retrieving a user's cart and a POST endpoint for adding items to the cart. The right panel displays the rendered API documentation. It features a title 'МегаМаркет API' with version and OAS version indicators, a brief description, a server selection dropdown, and a section titled 'Управление Корзиной'. This section lists the two endpoints with their methods (GET, POST) and descriptions. Below this, the 'Schemas' section defines the data structures: a 'Cart' object with a 'cartId' and an array of 'cartItems', and an 'Error' object with a 'code' and a 'message'.

```
1 openapi: 3.0.3
2 info:
3   title: МегаМаркет API
4   description: API для управления содержимым Корзины Пользователя
5   version: 1.1.1
6 servers:
7   - url: https://api.megamarket.com/v1
8
9 paths:
10  /users/{userId}/cart:
11    get:
12      tags:
13        - Управление Корзиной
14      summary: Получение содержимого Корзины Пользователя
15      description: Позволяет получить список Товаров в Корзине Пользователя
16      parameters:
17        - name: userId
18          in: path
19          description: Уникальный ID Пользователя
20          example: 111
21          required: true
22          schema:
23            type: number
24      responses:
25        '200':
26          description: OK
27          content:
28            application/json:
29              schema:
30                $ref: '#/components/schemas/Cart'
31        '400':
32          description: Некорректный запрос. Проверьте параметры запроса.
33          content:
34            application/json:
35              schema:
36                $ref: '#/components/schemas/Error'
37        '404':
38          description: Пользователь не найден.
39          content:
40            application/json:
41              schema:
42                $ref: '#/components/schemas/Error'
43        '500':
44          description: Внутренняя ошибка сервера. Попробуйте позже.
45          content:
46            application/json:
47              schema:
48                $ref: '#/components/schemas/Error'
49
50  /users/{userId}/cart/item:
51    post:
52      tags:
53        - Управление Корзиной
54      summary: Добавление Товара в Корзину
55      description: Позволяет добавить выбранный Товар в Корзину Пользователя
56      parameters:
57        - name: userId
58          in: path
59          description: Уникальный ID Пользователя
60          example: 111
61          required: true
62          schema:
63            type: number
64      requestBody:
65        description: Данные Товара для добавления в Корзину
66        required: true
67        content:
68          application/json:
69            schema:
70              type: object
71              properties:
```

**МегаМаркет API** 1.1.1 OAS 3.0

API для управления содержимым Корзины Пользователя

Servers

https://api.megamarket.com/v1

### Управление Корзиной

**GET** /users/{userId}/cart Получение содержимого Корзины Пользователя

**POST** /users/{userId}/cart/item Добавление Товара в Корзину

### Schemas

**Cart** {  
 cartId\*  
 cartItems\*  
}

number  
example: 123  
Уникальный ID Корзины

[ {  
 productId\*  
 name\*  
 unit\*  
 price\*  
 photo  
 description  
 category  
 itemQuantity\*  
}]

**Error** {  
 code\*  
 message\*  
}

number  
example: 418  
Код ошибки

string  
example: Я - чайник  
Сообщение об ошибке

YAML-файл