

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем  
Факультет автоматики и вычислительной техники  
Кафедра электронных вычислительных машин

Отчет по лабораторной работе №1  
по дисциплине  
«Управление данными»

Выполнил студент гр. ИВТб-2301-05-00	_____ /Макаров С.А./
Преподаватель	_____ /Клюкин В.Л./

Киров 2025

## Цель

Цель лабораторной работы: познакомиться со схемами, пользователями и ролями в PostgreSQL, познакомиться с типами данных в PostgreSQL, освоить основные варианты DDL-запросов в PostgreSQL, закрепить знания по проектированию структуры реляционной БД, создать рабочий материал для следующих лабораторных работ.

## Задание

1. Разработать структуру базы данных на любую выбранную тему. Структура должна отвечать следующим условиям:
  - должно быть не меньше пяти таблиц;
  - хотя бы одна таблица должна содержать колонку с числовыми данными;
  - структура БД не должна быть связь много-ко-многим.
2. Создать нового пользователя и пустую БД. Подключиться к созданной БД.
3. Написать и выполнить SQL-скрипт, создающий таблицы согласно разработанной структуре БД. Созданный в п.2 пользователь должен иметь все права на созданные объекты. В этом же скрипте должны создаваться нужные ограничения и индексы:
  - обязательно должны быть созданы внешние ключи для поддержания ссылочной целостности;
  - желательно должны быть проставлены ограничения и уникальные индексы для поддержания консистентности данных;
  - желательно должны быть проставлены индексы для производительности там, где они могут помочь.

## Решение

Выберем для структуры базы данных на тему «Сервис по доставке еды». Данная тема должна содержать пользователя, продукты, способы оплаты. Каждый пользователь имеет номер телефона, имя. Также у пользователя может выбирать способы оплаты. Продукты следует разделять по категориям. Помимо этого каждый продукт может иметь несколько вариаций и свои ингредиенты (обязательные, по выбору). Пользователь может добавлять продукты в корзину и оформлять заказ.

Таблица «users» содержит пользователей. Включает в себя столбцы «id» – уникальный идентификатор пользователя, «username» имя пользователя.

Таблица «payments» содержит доступные способы оплаты. Включает в себя столбцы «id» – уникальный идентификатор способа оплаты, «title» название способа оплаты.

Таблица «user payments» реализует связь многие ко многим между пользователем и способами оплаты. Включает в себя столбцы «id» – уникальный идентификатор, «user id» – идентификатор пользователя, «payment id» – идентификатор способа оплаты, «card number» – номер карты пользователя.

Таблица «categories» содержит категории продуктов. Включает в себя столбцы «id» – уникальный идентификатор категории, «title» – название категории.

Таблица «products» содержит продукты. Включает в себя столбцы «id» – уникальный идентификатор продукта, «title» – название продукта.

Таблица «ingredients» содержит ингредиенты. Включает в себя столбцы «id» – уникальный идентификатор ингредиента, «title» – название ингредиента, «price» – цена ингредиента.

Таблица «product ingredients» реализует связь многие ко многим между продуктами и ингредиентами. Включает в себя столбцы «id» – уникальный идентификатор, «product id» – идентификатор продукта, «ingredients id» – идентификатор ингредиента, «is required» – обязателен ли ингредиент.

Таблица «variants» содержит варианты продуктов. Включает в себя

столбцы «id» – уникальный идентификатор варианта продукта, «title» – название варианта, «value» – значение варианта.

Таблица «product variants» реализует связь многие ко многим между продуктами и вариантами продукта. Включает в себя столбцы «id» – уникальный идентификатор, «product id» – идентификатор продукта, «variant id» – идентификатор варианта продукта, «image url» – ссылка на изображение варианта продукта, «price» – цена варианта продукта.

Таблица «carts» содержит корзины пользователей. Включает в себя столбцы «id» – уникальный идентификатор корзины, «user id» – идентификатор пользователя.

Таблица «cart products» содержит товары корзины. Включает в себя столбцы «id» – уникальный идентификатор продукта, «cart id» – идентификатор корзины, «product variant id» – идентификатор варианта продукта, «quantity» – количество продукта.

Таблица «cart product ingredients» содержит дополнительные ингредиенты для продукта. Включает в себя столбцы «id» – уникальный идентификатор, «cart product id» – идентификатор продукта в корзине, «product ingredient id» – идентификатор ингредиента продукта.

Таблица «orders» содержит заказы пользователей. Включает в себя столбцы «id» – уникальный идентификатор заказа, «user id» – идентификатор пользователя, «payment id» – идентификатор способа оплаты, «status» – статус заказа, «address» – адрес для доставки заказа, «username» – имя заказчика, «cost» – стоимость заказа, «comment» – комментарий к заказу.

Таблица «order products» содержит продукты заказа. Включает в себя столбцы «id» – уникальный идентификатор продукта, «order id» – идентификатор заказа, «product variant id» – идентификатор варианта продукта, «quantity» – количество продукта.

Таблица «order product ingredients» содержит дополнительные ингредиенты для продукта. Включает в себя столбцы «id» – уникальный идентификатор, «order product id» – идентификатор продукта в заказе, «product ingredient id» – идентификатор ингредиента продукта.

Для данной базы данных была разработана ER диаграмма, представленная на рисунке 1.

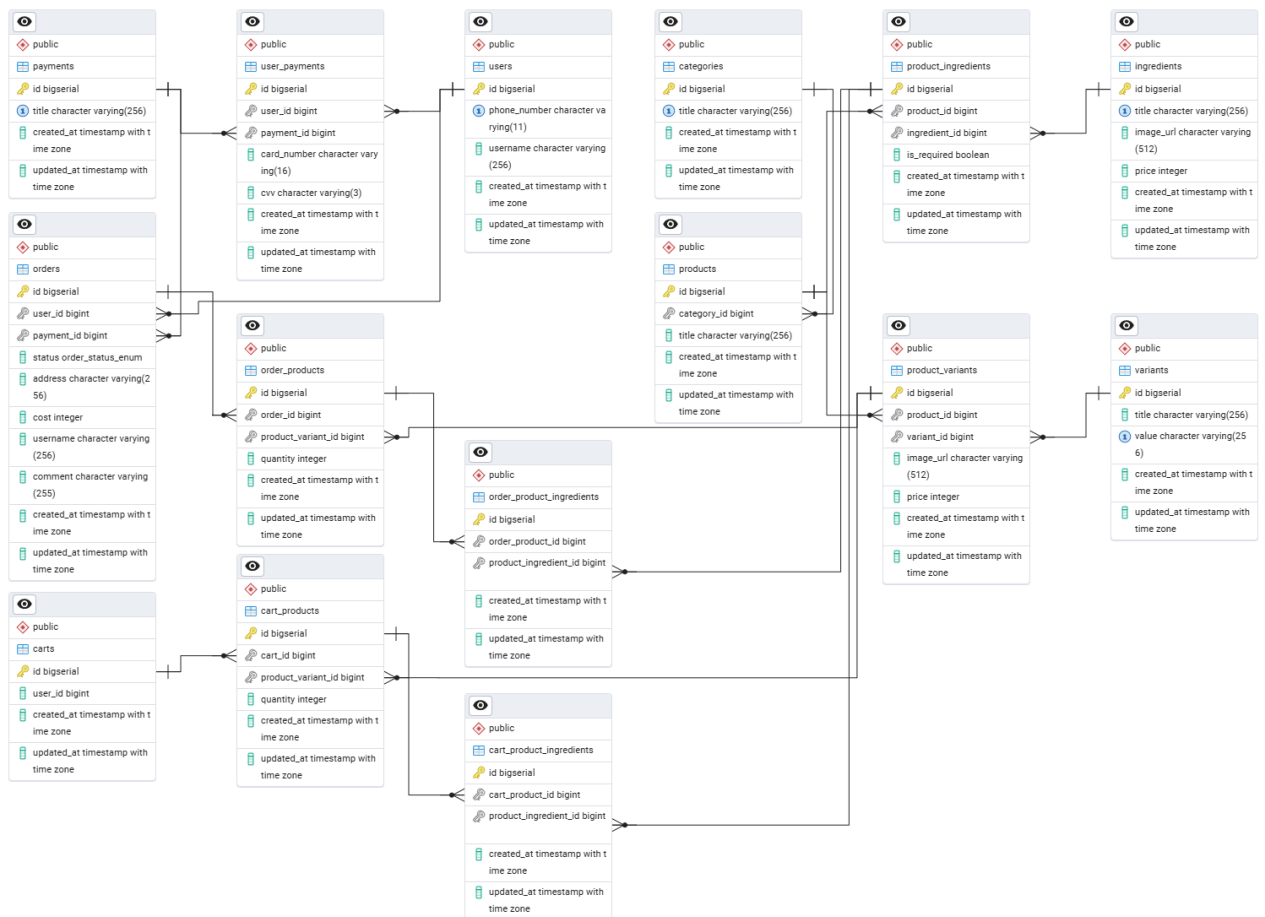


Рисунок 1 – ER - диаграмма базы данных

SQL-скрипт для создания таблиц базы данных представлен ниже:

```

CREATE TABLE "users" (
    "id" BIGSERIAL PRIMARY KEY,
    "phone_number" VARCHAR(11) NOT NULL UNIQUE,
    "username" VARCHAR(256),
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);
  
```

```

CREATE TABLE "payments" (
    "id" BIGSERIAL PRIMARY KEY,
    "title" VARCHAR(256) NOT NULL UNIQUE,
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);
  
```

```

CREATE TABLE "user_payments" (
    "id" BIGSERIAL PRIMARY KEY,
    "user_id" BIGINT NOT NULL,
    "payment_id" BIGINT NOT NULL,
    "card_number" VARCHAR(16),
    "cvv" VARCHAR(3),
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY ("user_id") REFERENCES "users"("id") ON DELETE CASCADE,
    FOREIGN KEY ("payment_id") REFERENCES "payments"("id") ON DELETE CASCADE
);

```

```

CREATE TABLE "categories" (
    "id" BIGSERIAL PRIMARY KEY,
    "title" VARCHAR(256) NOT NULL UNIQUE,
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

```

```

CREATE TABLE "products" (
    "id" BIGSERIAL PRIMARY KEY,
    "category_id" BIGINT NOT NULL,
    "title" VARCHAR(256) NOT NULL,
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY ("category_id") REFERENCES "categories"("id")
    ON DELETE RESTRICT
);

```

```

CREATE TABLE "ingredients" (
    "id" BIGSERIAL PRIMARY KEY,
    "title" VARCHAR(256) NOT NULL UNIQUE,
    "image_url" VARCHAR(512) NOT NULL,
    "price" INT NOT NULL CHECK (price >= 0),
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

```

```

CREATE TABLE "product_ingredients" (
    "id" BIGSERIAL PRIMARY KEY,
    "product_id" BIGINT NOT NULL,
    "ingredient_id" BIGINT NOT NULL,
    "is_required" BOOLEAN NOT NULL DEFAULT FALSE,
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY ("product_id") REFERENCES "products"("id") ON DELETE RESTRICT,
    FOREIGN KEY ("ingredient_id") REFERENCES "ingredients"("id")
    ON DELETE RESTRICT
);

CREATE TABLE "variants" (
    "id" BIGSERIAL PRIMARY KEY,
    "title" VARCHAR(256) NOT NULL,
    "value" VARCHAR(256) NOT NULL UNIQUE,
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE "product_variants" (
    "id" BIGSERIAL PRIMARY KEY,
    "product_id" BIGINT NOT NULL,
    "variant_id" BIGINT NOT NULL,
    "image_url" VARCHAR(512) NOT NULL,
    "price" INT NOT NULL CHECK (price >= 0),
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY ("product_id") REFERENCES "products"("id") ON DELETE RESTRICT,
    FOREIGN KEY ("variant_id") REFERENCES "variants"("id") ON DELETE RESTRICT
);

CREATE TABLE "carts" (
    "id" BIGSERIAL PRIMARY KEY,
    "user_id" BIGINT NOT NULL,
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP

```

);

```
CREATE TABLE "cart_products" (  
    "id" BIGSERIAL PRIMARY KEY,  
    "cart_id" BIGINT NOT NULL,  
    "product_variant_id" BIGINT NOT NULL,  
    "quantity" INT NOT NULL DEFAULT 1 CHECK (quantity >= 1),  
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,  
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,  
  
    FOREIGN KEY ("cart_id") REFERENCES "carts"("id") ON DELETE RESTRICT,  
    FOREIGN KEY ("product_variant_id") REFERENCES "product_variants"("id")  
    ON DELETE RESTRICT  
);
```

```
CREATE TABLE "cart_product_ingredients" (  
    "id" BIGSERIAL PRIMARY KEY,  
    "cart_product_id" BIGINT NOT NULL,  
    "product_ingredient_id" BIGINT NOT NULL,  
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,  
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,  
  
    FOREIGN KEY ("cart_product_id") REFERENCES "cart_products"("id")  
    ON DELETE RESTRICT,  
    FOREIGN KEY ("product_ingredient_id")  
    REFERENCES "product_ingredients"("id") ON DELETE RESTRICT  
);
```

```
CREATE TYPE ORDER_STATUS_ENUM AS ENUM (  
    'pending', 'succeeded', 'canceled'  
);
```

```
CREATE TABLE "orders" (  
    "id" BIGSERIAL PRIMARY KEY,  
    "user_id" BIGINT NOT NULL,  
    "payment_id" BIGINT NOT NULL,  
    "status" ORDER_STATUS_ENUM DEFAULT 'pending',  
    "address" VARCHAR(256) NOT NULL,  
    "cost" INT NOT NULL CHECK (cost >= 0),  
    "username" VARCHAR(256),  
    "comment" VARCHAR(255),  
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
```



```

"updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,

FOREIGN KEY ("user_id") REFERENCES "users"("id") ON DELETE CASCADE,
FOREIGN KEY ("payment_id") REFERENCES "payments"("id") ON DELETE CASCADE
);

CREATE TABLE "order_products" (
    "id" BIGSERIAL PRIMARY KEY,
    "order_id" BIGINT NOT NULL,
    "product_variant_id" BIGINT NOT NULL,
    "quantity" INT NOT NULL CHECK (quantity >= 1),
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY ("order_id") REFERENCES "orders"("id") ON DELETE RESTRICT,
    FOREIGN KEY ("product_variant_id") REFERENCES "product_variants"("id")
    ON DELETE RESTRICT
);

CREATE TABLE "order_product_ingredients" (
    "id" BIGSERIAL PRIMARY KEY,
    "order_product_id" BIGINT NOT NULL,
    "product_ingredient_id" BIGINT NOT NULL,
    "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    "updated_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY ("order_product_id") REFERENCES "order_products"("id")
    ON DELETE RESTRICT,
    FOREIGN KEY ("product_ingredient_id")
    REFERENCES "product_ingredients"("id") ON DELETE RESTRICT
);

```

## Вывод

В ходе выполнения лабораторной работы были освоены схемы, пользователи и роли в PostgreSQL, изучены типы данных в PostgreSQL, освоены основные варианты DDL-запросов, закреплены знания по проектированию структуры реляционной базы данных. В результате выполнения разработана база данных для сервиса по доставке еды.