

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего  
образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

## **Отчет**

по лабораторной работе «Создание БД в СУБД PostgreSQL. Резервное  
копирование и восстановление БД» по дисциплине «Проектирование и  
реализация баз данных»

Вариант 10. БД «Автовокзал»

Автор: Чан Дык Минь

Факультет: ИКТ

Группа: К32392

Преподаватель: Говорова М. М.



Санкт-Петербург 2023

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	4
СХЕМА ЛОГИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ, СГЕНЕРИРОВАННАЯ В GENERATE ERD.....	5
DUMP, СОДЕРЖАЩИЙ СКРИПТЫ РАБОТЫ С БД.....	6
ВЫВОДЫ.....	11

## ВВЕДЕНИЕ

**Цель работы:** овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

### **Практическое задание:**

1. Создать базу данных с использованием pgAdmin 4 (согласно индивидуальному заданию).
2. Создать схему в составе базы данных.
3. Создать таблицы базы данных.
4. Установить ограничения на данные: Primary Key, Unique, Check, Foreign Key.
5. Заполнить таблицы БД рабочими данными.
6. Создать резервную копию БД.

Создать две резервные копии:

- с расширением CUSTOM для восстановления БД;
  - с расширением PLAIN для листинга (в отчете);
  - при создании резервных копий БД настроить параметры Dump options для Type of objects и Queries.
7. Восстановить БД.

## ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

### Вариант 10. БД «Автовокзал»

Описание предметной области: С автовокзала ежедневно отправляется несколько междугородных/международных автобусных рейсов. Номер рейса определяется маршрутом и временем отправления. По всем промежуточным остановкам на маршруте известны название, тип населенного пункта, время прибытия, отправления, время стоянки.

Автобусы курсируют по расписанию, но могут назначаться дополнительные рейсы на заданный период или определенные даты.

Билеты могут продаваться предварительно, но не ранее чем за 10 суток. В билете указывается номер места в автобусе. На каждый рейс может продаваться не более 10 билетов без места, цена на которые снижается на 10%. Пунктами отправления и назначения, согласно билету, могут быть промежуточные остановки.

Билеты могут продаваться в кассе автовокзала или онлайн.

На каждый рейс формируется экипаж из двух водителей.

БД должна содержать следующий минимальный набор сведений: Номер рейса. Номер водителя. Номер автобуса. Паспортные данные водителя. Пункт отправления. Пункт назначения. Промежуточные остановки. Дата отправления. Время отправления. Время в пути. Тип автобуса. Количество мест в автобусе. Страна. Производитель. Год выпуска. Номер билета. Номер места в автобусе (при наличии). Цена билета. ФИО пассажира. Паспортные данные пассажира.

# СХЕМА ЛОГИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ, СГЕНЕРИРОВАННАЯ В GENERATE ERD

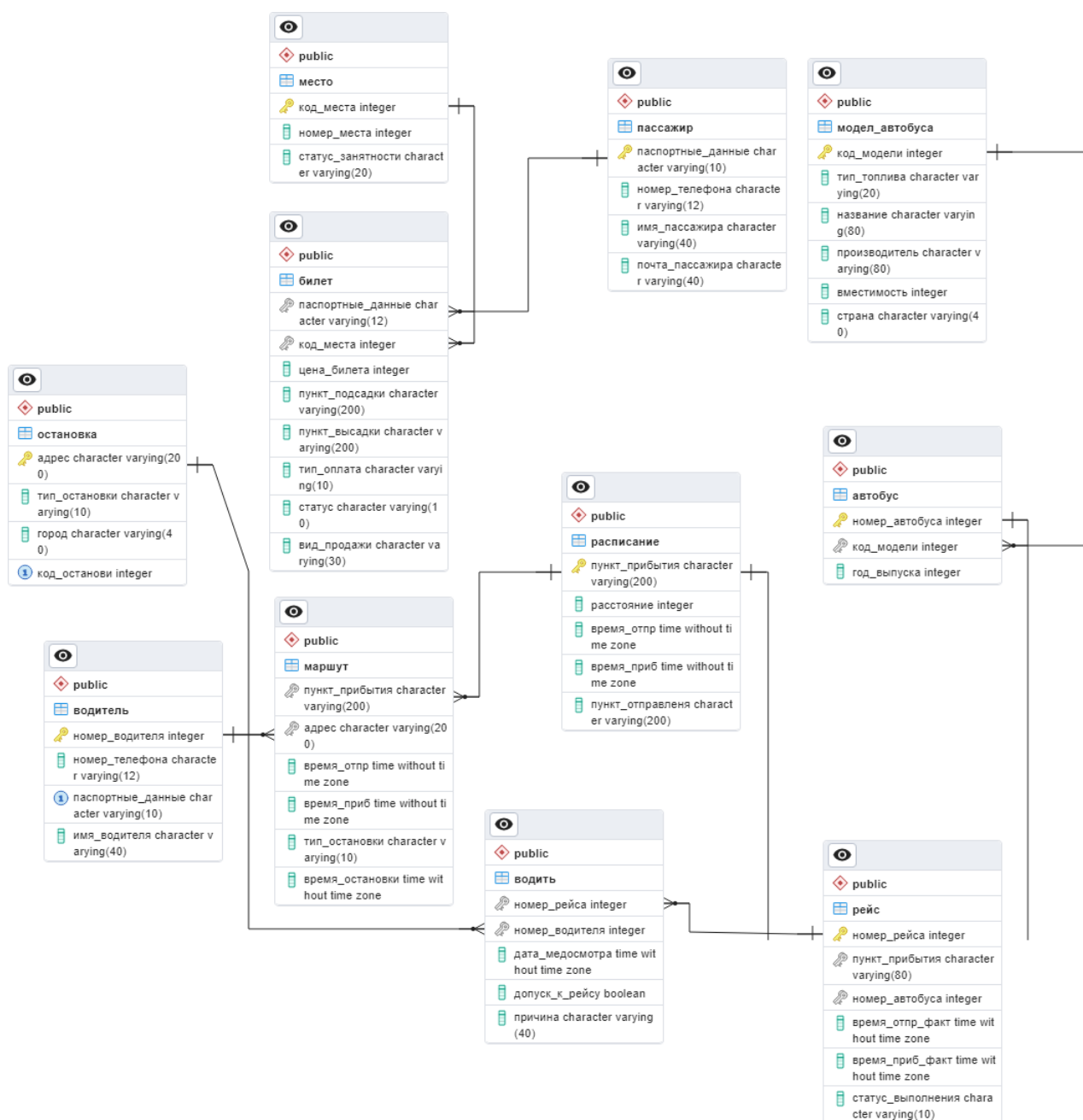


Рисунок 1 – Схема ЛМ БД в Generate ERD

## DUMP, СОДЕРЖАЩИЙ СКРИПТЫ РАБОТЫ С БД

```
BEGIN;
CREATE TABLE IF NOT EXISTS public."модел_автобуса" (
  "код_модели" integer NOT NULL UNIQUE PRIMARY KEY,
  "тип_топлива" varchar(20) NOT NULL CHECK(
    тип_топлива IN ('бензин', 'мазут')
  ),
  "название" varchar(80) NOT NULL,
  "производитель" varchar(80) NOT NULL,
  "вместимость" integer NOT NULL CHECK(вместимость > 0),
  "страна" varchar(40) NOT NULL
);
CREATE TABLE IF NOT EXISTS public."автобус" (
  "номер_автобуса" integer NOT NULL UNIQUE GENERATED ALWAYS
  AS IDENTITY PRIMARY KEY,
  "код_модели" integer NOT NULL,
  "год_выпуска" integer NOT NULL CHECK (
    год_выпуска < DATE_PART('Year', NOW())
  ),
  FOREIGN KEY (код_модели) REFERENCES
  модел_автобуса(код_модели)
);
CREATE TABLE IF NOT EXISTS public."водитель" (
  "номер_водителя" integer NOT NULL UNIQUE GENERATED ALWAYS
  AS IDENTITY PRIMARY KEY,
  "номер_телефона" varchar(12) NOT NULL,
  "паспортные_данные" varchar(10) NOT NULL UNIQUE,
  "имя_водителя" varchar(40) NOT NULL
);
CREATE TABLE IF NOT EXISTS public."пассажир" (
  "паспортные_данные" varchar(10) NOT NULL UNIQUE PRIMARY KEY,
  "номер_телефона" varchar(12) NOT NULL,
  "имя_пассажира" varchar(40) NOT NULL,
  "почта_пассажира" varchar(40) NOT NULL CHECK(
    почта_пассажира LIKE '%_@_%._%'
  )
);
```

```

CREATE TABLE IF NOT EXISTS public."остановка" (
  "адрес" varchar(200) NOT NULL UNIQUE PRIMARY KEY,
  "тип_остановки" varchar(10) NOT NULL CHECK(
    тип_остановки IN(
      'быстрый', 'длительный'
    )
  ),
  "город" varchar(40) NOT NULL,
  "код_остановки" integer NOT NULL UNIQUE GENERATED ALWAYS AS
IDENTITY
);
CREATE TABLE IF NOT EXISTS public."место" (
  "код_места" integer NOT NULL UNIQUE GENERATED ALWAYS AS
IDENTITY PRIMARY KEY,
  "номер_места" integer NOT NULL,
  "статус_занятности" varchar(20) NOT NULL CHECK(
    статус_занятности IN(
      'Забронировано',
      'пусто', 'отменено'
    )
  )
);
CREATE TABLE IF NOT EXISTS public."расписание" (
  "пункт_прибытия" varchar(200) NOT NULL UNIQUE PRIMARY KEY,
  "расстояние" integer NOT NULL,
  "время_отпр" TIME NOT NULL,
  "время_приб" TIME NOT NULL,
  "пункт_отправления" varchar(200) NOT NULL
);
CREATE TABLE IF NOT EXISTS public."рейс" (
  "номер_рейса" integer NOT NULL UNIQUE GENERATED ALWAYS AS
IDENTITY PRIMARY KEY,
  "пункт_прибытия" varchar(80) NOT NULL,
  "номер_автобуса" integer NOT NULL,
  "время_отпр_факт" TIME NOT NULL CHECK(
    время_отпр_факт < время_приб_факт
  ),

```

```

"время_приб_факт" TIME NOT NULL CHECK(
    время_отпр_факт < время_приб_факт
),
"статус_выполнения" varchar(10) NOT NULL CHECK(
    статус_выполнения IN (
        'прибыли', 'не прибыли'
    )
),
CONSTRAINT fk_пункт_прибытия FOREIGN KEY(пункт_прибытия)
REFERENCES расписание(пункт_прибытия),
CONSTRAINT fk_номер_автобуса FOREIGN KEY(номер_автобуса)
REFERENCES автобус(номер_автобуса)
);
CREATE TABLE IF NOT EXISTS public."водить" (
    "номер_рейса" integer NOT NULL UNIQUE,
    "номер_водителя" integer NOT NULL UNIQUE,
    "дата_медосмотра" TIME NOT NULL,
    "допуск_к_рейсу" boolean NOT NULL DEFAULT FALSE,
    "причина" varchar(40) NOT NULL,
    CONSTRAINT fk_номер_рейса FOREIGN KEY(номер_рейса)
REFERENCES рейс(номер_рейса),
    CONSTRAINT fk_номер_водителя FOREIGN KEY(номер_водителя)
REFERENCES водитель(номер_водителя)
);
CREATE TABLE IF NOT EXISTS public."билет" (
    "паспортные_данные" varchar(12) NOT NULL,
    "код_места" integer NOT NULL,
    "цена_билета" integer NOT NULL CHECK(цена_билета >= 0),
    "пункт_подсадки" varchar(200) NOT NULL,
    "пункт_высадки" varchar(200) NOT NULL,
    "тип_оплата" varchar(10) NOT NULL CHECK(
        тип_оплата IN(
            'карта', 'наличные'
        )
    ),
),

```



```

"статус" varchar(10) NOT NULL CHECK(
    статус IN(
        'оплачен', 'бронь', 'возврат'
    )
),
"вид_продажи" varchar(30) NOT NULL CHECK(
    вид_продажи IN(
        'продавать напрямую',
        'продавать по телефону',
        'продавать через киоск'
    )
),
CONSTRAINT fk_паспортные_данные FOREIGN KEY(
    паспортные_данные
) REFERENCES пассажир(
    паспортные_данные
),
CONSTRAINT fk_код_места FOREIGN KEY(код_места) REFERENCES
место(код_места)
);
CREATE TABLE IF NOT EXISTS public."маршрут" (
    "пункт_прибытия" varchar(200) NOT NULL,
    "адрес" varchar(200) NOT NULL,
    "время_отпр" TIME NOT NULL CHECK(
        время_отпр < время_приб
    ),
    "время_приб" TIME NOT NULL CHECK(
        время_отпр < время_приб
    ),
    "тип_остановки" varchar(10) NOT NULL CHECK(
        тип_остановки IN(
            'быстрый', 'длительный'
        )
    ),
    "время_остановки" TIME NOT NULL,
    CONSTRAINT fk_пункт_прибытия FOREIGN KEY(пункт_прибытия)
REFERENCES расписание(пункт_прибытия),

```

```
CONSTRAINT fk_адрес FOREIGN KEY(адрес) REFERENCES  
остановка(адрес)  
);  
END;
```

## **ВЫВОДЫ**

В рамках этой лаборатории практические навыки были получены путем создания таблиц базы данных PostgreSQL, заполнения их рабочими данными, резервного копирования и восстановления базы данных.

Создайте диаграмму ERD для базы данных. Узнайте, как использовать код для создания таблиц и данных в базе данных. Кроме того, получить опыт работы в использовании программы Pgadmin 4.