

**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.....	10

## **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."bus\_models"

```
(  
    "model_code" integer NOT NULL unique PRIMARY KEY,  
    "fuel_type" varchar(20) NOT NULL CHECK(fuel_type IN ('gasoline', 'oil')),  
    "name" varchar(80) NOT NULL,  
    "manufacturer" varchar(80) NOT NULL,  
    "capacity" integer NOT NULL CHECK(capacity>0),  
    "country" varchar(40) NOT NULL  
);
```

CREATE TABLE IF NOT EXISTS public."buses"

```
(  
    "bus_id" integer NOT NULL unique GENERATED ALWAYS AS  
IDENTITY PRIMARY KEY,  
    "model_code" integer NOT NULL,  
    "release_year" integer NOT NULL CHECK  
(release_year<DATE_PART('Year', NOW())),  
    foreign key (model_code) references bus_models(model_code)  
);
```

CREATE TABLE IF NOT EXISTS public."drivers"

```
(  
    "driver_id" integer NOT NULL unique GENERATED ALWAYS AS  
IDENTITY PRIMARY KEY,  
    "phone_id" varchar(12) NOT NULL,  
    "passport" varchar(10) NOT NULL unique,  
    "driver_name" varchar(40) NOT NULL  
);
```

CREATE TABLE IF NOT EXISTS public."passengers"

```
(  
    "passport" varchar(10) NOT NULL unique PRIMARY KEY,  
    "phone_number" varchar(12) NOT NULL,  
    "passenger_name" varchar(40) NOT NULL,
```

```
    "passenger_email" varchar(40) NOT NULL CHECK(passenger_email  
LIKE '%_@_%._%')  
);
```

```
CREATE TABLE IF NOT EXISTS public."stops"  
(  
    "address" varchar(200) NOT NULL unique PRIMARY KEY,  
    "stop_type" varchar(10) NOT NULL CHECK(stop_type IN('short', 'long')),  
    "city" varchar(40) NOT NULL,  
    "stop_code" integer NOT NULL unique GENERATED ALWAYS AS  
IDENTITY  
);
```

```
CREATE TABLE IF NOT EXISTS public."seats"  
(  
    "seat_code" integer NOT NULL unique GENERATED ALWAYS AS  
IDENTITY PRIMARY KEY,  
    "seat_id" integer NOT NULL,  
    "book_status" varchar(20) NOT NULL CHECK(book_status IN('booked',  
'empty', 'canceled'))  
);
```

```
CREATE TABLE IF NOT EXISTS public."schedules"  
(  
    "arrival_point" varchar(200) NOT NULL unique PRIMARY KEY,  
    "distance" integer NOT NULL,  
    "departure_time" time NOT NULL,  
    "arrival_time" time NOT NULL,  
    "departure_point" varchar(200) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS public."trips"  
(  
    "trip_id" integer NOT NULL unique GENERATED ALWAYS AS  
IDENTITY PRIMARY KEY,  
    "arrival_point" varchar(80) NOT NULL,  
    "bus_id" integer NOT NULL,
```

```

        "actual_departure_time" time NOT NULL
CHECK(actual_departure_time<actual_arrival_time),
        "actual_arrival_time" time NOT NULL
CHECK(actual_departure_time<actual_arrival_time),
        "status" varchar(10) NOT NULL CHECK(status IN ('arrived', 'not
arrived')),
        FOREIGN KEY(arrival_point) REFERENCES schedules(arrival_point),
        FOREIGN KEY(bus_id) REFERENCES buses(bus_id)

```

```
);
```

```
CREATE TABLE IF NOT EXISTS public."drive"
```

```

(
    "trip_id" integer NOT NULL unique,
    "driver_id" integer NOT NULL unique,
    "admission_to_trip" boolean NOT NULL DEFAULT FALSE,
    PRIMARY KEY (trip_id, driver_id),
    FOREIGN KEY(trip_id) REFERENCES trips(trip_id) ON DELETE
CASCADE,
    FOREIGN KEY(driver_id) REFERENCES drivers(driver_id) ON
DELETE CASCADE
);

```

```
CREATE TABLE IF NOT EXISTS public."tickets"
```

```

(
    "passport" varchar(12) NOT NULL,
    "seat_code" integer NOT NULL,
    "ticket_price" integer NOT NULL CHECK(ticket_price >= 0),
    "landing_point" varchar(200) NOT NULL,
    "drop_point" varchar(200) NOT NULL,
    "payment_type" varchar(10) NOT NULL CHECK(payment_type
IN('card', 'cash')),
    "status" varchar(10) NOT NULL CHECK(status IN('payed', 'waiting',
'refund')),
    "sale_type" varchar(30) NOT NULL CHECK(sale_type IN('sell directly',
'sell by phone', 'sell through a kiosk')),
    PRIMARY KEY (passport, seat_code),

```



```
        FOREIGN KEY(passport) REFERENCES passengers(passport) ON  
DELETE CASCADE,  
        FOREIGN KEY(seat_code) REFERENCES seats(seat_code) ON  
DELETE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS public."routes"  
(  
    "arrival_point" varchar(200) NOT NULL,  
    "address" varchar(200) NOT NULL,  
    "departure_time" time NOT NULL  
CHECK(departure_time<arrival_time),  
    "arrival_time" time NOT NULL CHECK(departure_time<arrival_time),  
    "stop_type" varchar(10) NOT NULL CHECK(stop_type IN('short',  
'long')),  
    "stop_time" time NOT NULL,  
    PRIMARY KEY (arrival_point, address),  
    FOREIGN KEY(arrival_point) REFERENCES schedules(arrival_point)  
ON DELETE CASCADE,  
    FOREIGN KEY(address) REFERENCES stops(address) ON DELETE  
CASCADE  
);  
  
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.