

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

**Отчет**

По лабораторной работе №1

«Создание БД в СУБД PostgreSQL.

Резервное копирование и восстановление БД»  
по дисциплине «Проектирование и реализация баз данных»

Вариант 10. БД «Автовокзал»

Автор: Ле Хоанг Чыонг

Факультет: ИКТ

Группа: К32392

Преподаватель: Говорова М. М.

Санкт-Петербург, 2023

# 1 Описание работы

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

**Оборудование:** компьютерный класс.

**Программное обеспечение:** СУБД PostgreSQL 1X, pgAdmin 4.

**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.
  - i. *Указание:*
  - ii. *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.

## **2 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.

### 3 Выполнение работы

Наименование БД: Bus  
ERD диаграмма:

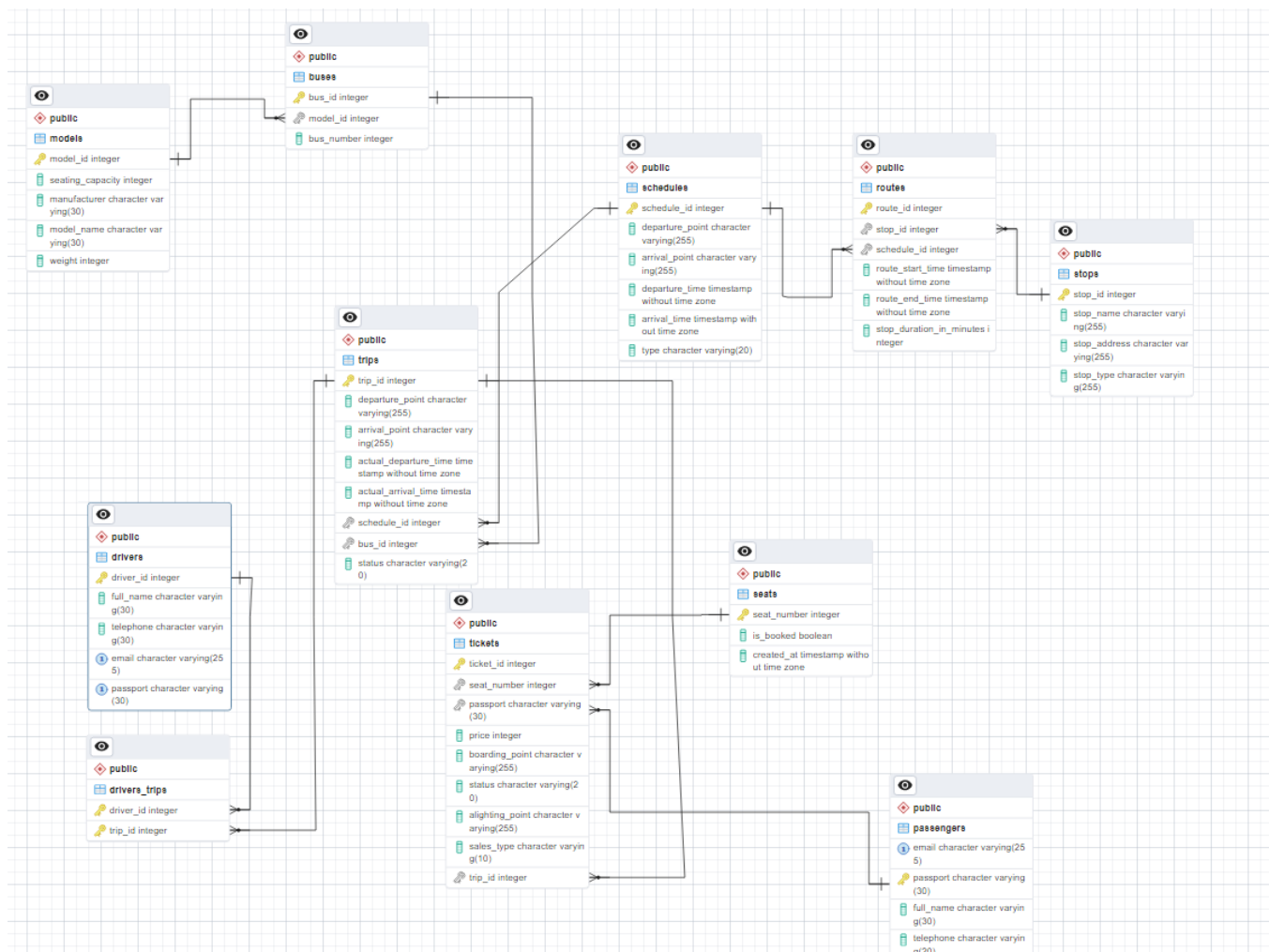


Рисунок 1 – ERD диаграмм

## 4 Dump, содержащий скрипты работы с БД

### Скрипты работы с БД для создания таблиц:

```
CREATE TABLE IF NOT EXISTS passengers(  
    email VARCHAR(255) NOT NULL UNIQUE,  
    passport VARCHAR(30) PRIMARY KEY,  
    full_name VARCHAR(30) NOT NULL,  
    telephone VARCHAR(20) NOT NULL CHECK (telephone ~ '^\\d{10}$'),  
    CONSTRAINT chk_email CHECK (email LIKE '%_@__%.__%')  
);  
  
CREATE TABLE IF NOT EXISTS seats(  
    seat_number INT PRIMARY KEY,  
    is_booked BOOLEAN NOT NULL DEFAULT FALSE,  
    created_at TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP  
);  
  
CREATE TABLE IF NOT EXISTS schedules (  
    schedule_id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    departure_point VARCHAR(255) NOT NULL,  
    arrival_point VARCHAR(255) NOT NULL,  
    departure_time TIMESTAMP NOT NULL CHECK (departure_time < arrival_time),  
    arrival_time TIMESTAMP NOT NULL CHECK (arrival_time > departure_time),  
    type VARCHAR(20) NOT NULL CHECK (type IN ('intercity', 'international'))  
    DEFAULT 'intercity'  
);  
  
CREATE TABLE IF NOT EXISTS models(  
    model_id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    seating_capacity INT NOT NULL,  
    manufacturer VARCHAR(30) NOT NULL,
```

```
    model_name VARCHAR(30) NOT NULL,  
    weight INT NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS buses(  
    bus_id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    model_id INT NOT NULL,  
    bus_number INT NOT NULL,  
    FOREIGN KEY (model_id) REFERENCES models(model_id) ON DELETE  
CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS trips(  
    trip_id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    departure_point VARCHAR(255) NOT NULL,  
    arrival_point VARCHAR(255) NOT NULL,  
    actual_departure_time TIMESTAMP NOT NULL CHECK (actual_arrival_time >  
actual_departure_time),  
    actual_arrival_time TIMESTAMP NOT NULL CHECK (actual_arrival_time >  
actual_departure_time),  
    schedule_id INT NOT NULL,  
    bus_id INT NOT NULL,  
    status VARCHAR(20) NOT NULL CHECK (status IN ('Pending', 'In progress',  
'Completed'))),  
    FOREIGN KEY (schedule_id) REFERENCES schedules(schedule_id),  
    FOREIGN KEY (bus_id) REFERENCES buses(bus_id)  
);
```

```
CREATE TABLE IF NOT EXISTS tickets(  
    ticket_id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    seat_number INT NOT NULL,  
    passport VARCHAR(30) NOT NULL,
```

```
price INT NOT NULL,  
boarding_point VARCHAR(255) NOT NULL,  
status VARCHAR(20) NOT NULL CHECK (status IN ('paid', 'unpaid', 'cancelled')),  
alighting_point VARCHAR(255) NOT NULL,  
sales_type VARCHAR(10) NOT NULL CHECK (sales_type IN('online','offline')),  
trip_id INT NOT NULL,  
FOREIGN KEY (trip_id) REFERENCES trips(trip_id) ON DELETE CASCADE,  
FOREIGN KEY (passport) REFERENCES passengers(passport),  
FOREIGN KEY (seat_number) REFERENCES seats(seat_number)  
);
```

```
CREATE TABLE IF NOT EXISTS stops (  
    stop_id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    stop_name VARCHAR(255) NOT NULL,  
    stop_address VARCHAR(255) NOT NULL,  
    stop_type VARCHAR(255) NOT NULL DEFAULT 'unknown'  
);
```

```
CREATE TABLE routes(  
    route_id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
    stop_id INT NOT NULL,  
    schedule_id INT NOT NULL,  
    route_start_time TIMESTAMP NOT NULL,  
    route_end_time TIMESTAMP NOT NULL,  
    stop_duration_in_minutes INT NOT NULL,  
    FOREIGN KEY (stop_id) REFERENCES stops(stop_id),  
    FOREIGN KEY (schedule_id) REFERENCES schedules(schedule_id),  
    CONSTRAINT chk_route_time CHECK (route_start_time < route_end_time)  
);
```

```
CREATE TABLE IF NOT EXISTS drivers (  
    driver_id INT GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
```

```
full_name VARCHAR(30) NOT NULL,  
telephone VARCHAR(30) NOT NULL,  
email VARCHAR(255) NOT NULL UNIQUE,  
passport VARCHAR(30) NOT NULL UNIQUE,  
CONSTRAINT chk_email CHECK (email LIKE '%_@__%.__%')  
);
```

```
CREATE TABLE drivers_trips (  
    driver_id INT NOT NULL,  
    trip_id INT NOT NULL,  
    PRIMARY KEY (driver_id, trip_id),  
    FOREIGN KEY (driver_id) REFERENCES drivers (driver_id) ON DELETE  
CASCADE,  
    FOREIGN KEY (trip_id) REFERENCES trips (trip_id) ON DELETE CASCADE  
);
```



**Скрипты работы с БД для вставки данных:**

```
INSERT INTO passengers (email, passport, full_name, telephone)
VALUES
```

```
    ('email1@example.com', '234567890', 'passenger1', '0123456789'),
    ('email2@example.com', '345678901', 'passenger2', '0987654321'),
    ('email3@example.com', '456789012', 'passenger3', '0123456789');
```

```
INSERT INTO seats (seat_number, is_booked)
VALUES
```

```
    (4, false),
    (5, false),
    (6, true);
```

```
INSERT INTO schedules (departure_point, arrival_point, departure_time, arrival_time,
type)
VALUES
```

```
    ('Hanoi', 'Danang', '2023-03-26 08:00:00', '2023-03-26 14:00:00', 'intercity'),
    ('Hanoi', 'Bangkok', '2023-03-27 10:00:00', '2023-03-27 13:00:00', 'international');
```

```
INSERT INTO models (seating_capacity, manufacturer, model_name, weight)
VALUES
```

```
    (30, 'Toyota', 'Coaster', 4000),
    (80, 'Scania', 'Touring', 12000);
```

```
INSERT INTO buses (model_id, bus_number)
VALUES
```

```
(1, 2001),  
(1, 2002),  
(2, 3001),  
(2, 3002);
```

```
INSERT INTO trips (departure_point, arrival_point, actual_departure_time,  
actual_arrival_time, schedule_id, bus_id, status)
```

```
VALUES
```

```
('Hanoi', 'Danang', '2023-03-26 08:05:00', '2023-03-26 14:05:00', 1, 3, 'Pending'),  
('Hanoi', 'Bangkok', '2023-03-27 10:05:00', '2023-03-27 13:05:00', 2, 4, 'Pending');
```

```
INSERT INTO tickets (seat_number, passport, price, boarding_point, status,  
alighting_point, sales_type, trip_id)
```

```
VALUES
```

```
(6, '234567890', 150000, 'Hanoi', 'unpaid', 'Danang', 'online', 1),  
(5, '345678901', 200000, 'Hanoi', 'paid', 'Danang', 'offline', 1),  
(4, '456789012', 500000, 'Hanoi', 'unpaid', 'Bangkok', 'online', 2);
```

```
INSERT INTO stops (stop_name, stop_address, stop_type)
```

```
VALUES
```

```
('Stop 3', '789 Oak Street', 'unknown'),  
('Stop 4', '101 Main Street', 'unknown');
```

```
INSERT INTO routes (stop_id, schedule_id, stop_duration_in_minutes)
```

```
VALUES (1, 2, 30),
```

```
(2, 1, 15),  
(1, 2, 45),  
(2, 1, 30);
```

```
INSERT INTO drivers (full_name, telephone, email, passport)
```

```
VALUES
```

```
('driver1', '0987654321', 'driver1@example.com', '876543210'),
```

```
('driver2', '0123456789', 'driver2@example.com', '765432109');
```

```
INSERT INTO drivers_trips (driver_id, trip_id)
VALUES
  (1, 1),
  (2, 2)
```

## Процесс создания резервной копии и восстановления базы данных:

Backup (Database: Lab1\_database)

General

Data/Objects

Options

Filename

C:\Users\natib\Downloads\Bus\_backup\_custom.sql

Format

Custom

Compression ratio

Encoding

UTF8

X

Number of jobs

Role name

Select an item...

i

?

X Close

Reset

Backup

Backup (Database: Lab1\_database) ↗ ✕

General Data/Objects Options

Filename  📁

Format  | ▼

Compression ratio

Encoding  ✕ | ▼

Number of jobs

Role name  | ▼

ℹ ? ✕ Close ↺ Reset 💾 Backup

Рисунок 3 - Создание резервной копии БД

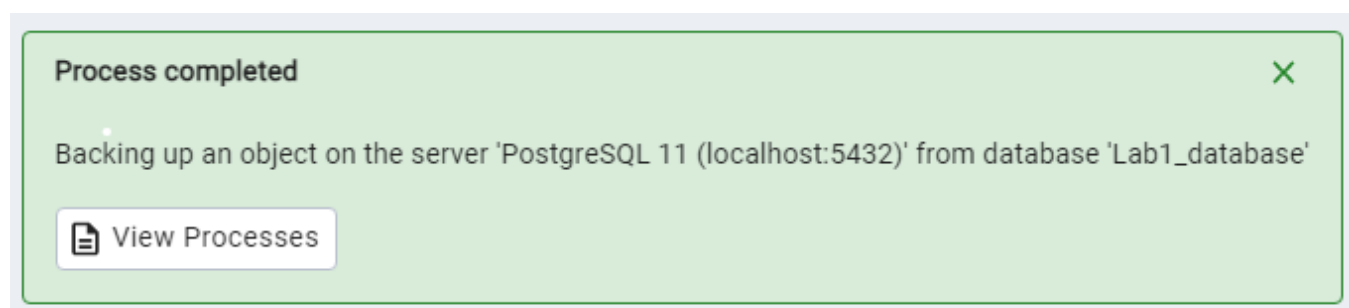


Рисунок 4 - Успешное создание резервной копии базы данных

Dashboard Properties SQL Statistics Dependencies Dependents <u>Processes</u>									
<div> <div></div> <div></div> </div>		<div>Search</div>							
			PID	Type	Server	Object	Start Time ▾	Status	Time Taken (sec)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4472	Backup Object	PostgreSQL 11 (localhost:5432)	Lab_database	3/21/2023, 8:54:52 PM	Finished	0.24
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	21816	Backup Object	PostgreSQL 11 (localhost:5432)	Lab_database	3/21/2023, 8:54:19 PM	Finished	0.35

Рисунок 5 - Резервное копирование базы дан

## **5 Выводы**

В результате выполнения лабораторной работы были получены навыки работы с PostgreSQL и средой разработки pgAdmin 4. Были изучены основные команды для создания таблиц, установки ограничений на таблицы, а также создания и восстановления резервных копий баз данных. Была создана структура базы данных для системы автобусного транспорта, которая включает таблицы для пассажиров, мест, моделей автобусов, автобусов, расписаний, рейсов, билетов, остановок и водителей. Также были добавлены тестовые данные для проверки корректности структуры таблиц. В целом, выполнение лабораторной работы позволило углубить знания в области работы с базами данных и получить практические навыки работы с PostgreSQL и pgAdmin 4.