

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3**

**«СОЗДАНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ PostgreSQL. ЗАПОЛНЕНИЕ ТАБЛИЦ РАБОЧИМИ
ДАНЫМИ»**

по дисциплине «Проектирование и реализация баз данных»

Обучающийся (Синюков Лев Владимирович)

Факультет прикладной информатики

Группа __3240__

Направление подготовки 09.03.03 Прикладная информатика

Образовательная программа Мобильные и сетевые технологии 2023

Преподаватель Говорова Марина Михайловна

Санкт-Петербург
2024/2025

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.

Указание:

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.

Индивидуальное задание:

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. Билет может быть приобретен в кассе автовокзала или онлайн. В билете указывается номер места в автобусе (необязательно). 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.

Необходимо учитывать, что местом посадки и высадки пассажира могут быть промежуточные остановки согласно купленному билету.

На каждый рейс формируется экипаж из двух водителей. Необходимо хранить данные о прохождении медосмотра перед рейсом (дата, статус, причина недопуска).

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.

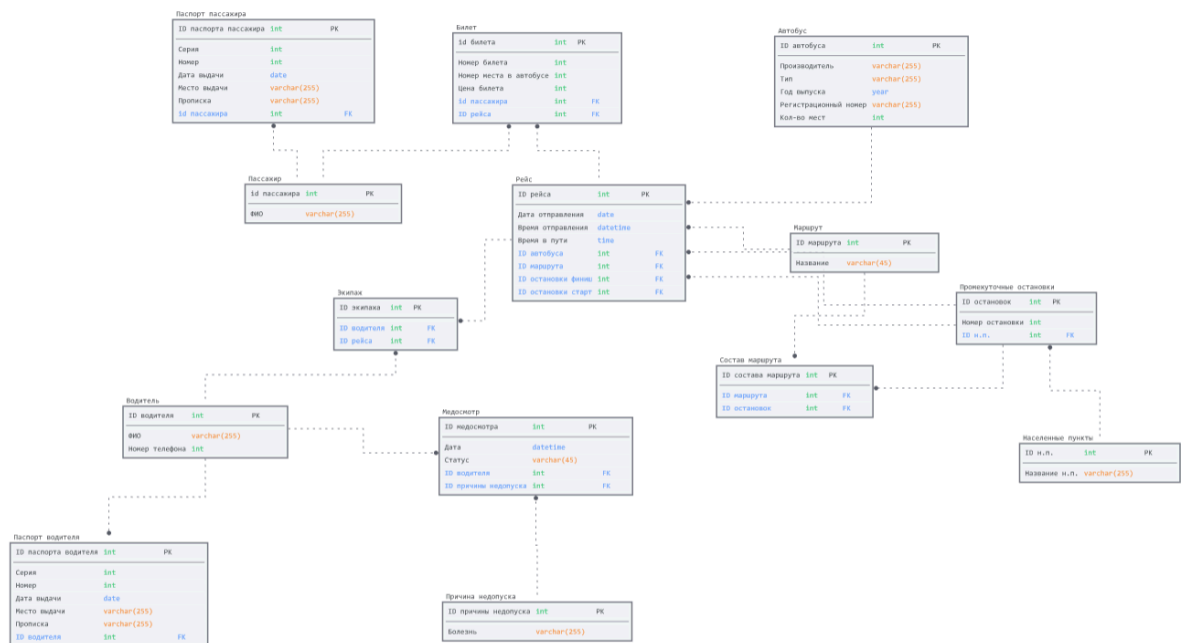
Дополните состав атрибутов на основе анализа предметной области.

Выполнение

БД «Автовокзал»

1. Паспорт пассажира (ID паспорта пассажира, серия, номер, дата выдачи, место выдачи, прописка, ID пассажира)
2. Пассажир (ID пассажира, ФИО)
3. Билет (ID билета, номер билета, номер места в автобусе, цена билета, ID пассажира, ID рейса)
4. Автобус (ID автобуса, производитель, тип, год выпуска, регистрационный номер, количество мест)
5. Рейс (ID рейса, дата отправления, время отправления, время прибытия, ID остановки финиш, ID остановки старт, ID автобуса, ID маршрута)
6. Экипаж (ID экипажа, ID водителя)
7. Водитель (ID водителя, ФИО, номер телефона)
8. Паспорт водителя (ID паспорта водителя, серия, номер, дата выдачи, место выдачи, прописка, ID водителя)
9. Маршрут (ID маршрута, название)
10. Состав маршрута (ID состава маршрута, ID маршрута, ID остановок)
11. Промежуточные остановки (ID остановок, номер остановки, ID н. п.)
12. Населенные пункты (ID н. п., название н. п.)
13. Медосмотр (ID медосмотра, дата, статус, ID водителя, ID причины недопуска)
14. Причина недопуска (ID причины недопуска, болезнь)

Схема инфологической модели данных БД в нотации IDEF1X



Создание БД:

```
-- Создание базы данных
CREATE DATABASE bus_system;
\c bus_system;
```

Добавление таблиц:

```
-- Населенные пункты
CREATE TABLE settlement (
    settlement_id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL
);

-- Маршрут
CREATE TABLE route (
    route_id SERIAL PRIMARY KEY,
    name VARCHAR(45) NOT NULL
);

-- Автобус
CREATE TABLE bus (
    bus_id SERIAL PRIMARY KEY,
    manufacturer VARCHAR(255),
    model_type VARCHAR(255),
    production_year INT,
    registration_number VARCHAR(255),
    seats_number INT NOT NULL
);

-- Данные пассажира
CREATE TABLE passenger (
    passenger_id SERIAL PRIMARY KEY,
    full_name VARCHAR(255) NOT NULL
);

-- Данные паспорта пассажира
CREATE TABLE passport_passenger (
    passport_id SERIAL PRIMARY KEY,
    series INT,
    number INT,
    issue_date DATE,
    issue_place VARCHAR(255),
    registration_address VARCHAR(255),
    passenger_id INT UNIQUE REFERENCES passenger(passenger_id)
);
```

```

-- Билет
CREATE TABLE ticket (
    ticket_id SERIAL PRIMARY KEY,
    ticket_number INT,
    seat_number INT,
    price NUMERIC(10, 2),
    passenger_id INT REFERENCES passenger(passenger_id),
    trip_id INT
);

-- Рейс (данные рейса)
CREATE TABLE trip (
    trip_id SERIAL PRIMARY KEY,
    departure_date DATE,
    departure_time TIMESTAMP,
    travel_time INTERVAL,
    bus_id INT REFERENCES bus(bus_id),
    route_id INT REFERENCES route(route_id),
    start_stop_id INT,
    final_stop_id INT
);

-- Промежуточные остановки
CREATE TABLE intermediate_stop (
    stop_id SERIAL PRIMARY KEY,
    stop_number INT,
    settlement_id INT REFERENCES settlement(settlement_id)
);

-- Состав маршрута
CREATE TABLE route_composition (
    route_composition_id SERIAL PRIMARY KEY,
    route_id INT REFERENCES route(route_id),
    stop_id INT REFERENCES intermediate_stop(stop_id)
);

-- Данные водителя
CREATE TABLE driver (
    driver_id SERIAL PRIMARY KEY,
    full_name VARCHAR(255),
    phone_number VARCHAR(20)
);

-- Данные паспорта водителя
CREATE TABLE passport_driver (

```

```

passport_id SERIAL PRIMARY KEY,
series INT,
number INT,
issue_date DATE,
issue_place VARCHAR(255),
registration_address VARCHAR(255),
driver_id INT UNIQUE REFERENCES driver(driver_id)
);

-- Экипаж
CREATE TABLE crew (
    crew_id SERIAL PRIMARY KEY,
    driver_id INT REFERENCES driver(driver_id),
    trip_id INT REFERENCES trip(trip_id)
);

-- Причины недопуска по медосмотру
CREATE TABLE reason_for_disqualification (
    reason_id SERIAL PRIMARY KEY,
    disease VARCHAR(255)
);

-- Медицинский осмотр
CREATE TABLE medical_checkup (
    checkup_id SERIAL PRIMARY KEY,
    checkup_date TIMESTAMP,
    status VARCHAR(45),
    driver_id INT REFERENCES driver(driver_id),
    reason_id INT REFERENCES reason_for_disqualification(reason_id)
);

```

Далее проверим запросы на вставку данных:

```

INSERT INTO settlement (name) VALUES ('Moscow'), ('St. Petersburg'),
('Pskov');

INSERT INTO route (name) VALUES ('Moscow - St. Petersburg'), ('Pskov -
Moscow');

INSERT INTO bus (manufacturer, model_type, production_year,
registration_number, seats_number)
VALUES ('GAZ', 'Gazelle', 2020, 'A777AA77', 12);

INSERT INTO passenger (full_name) VALUES ('Andrey Sahur');

INSERT INTO passport_passenger (series, number, issue_date, issue_place,
registration_address, passenger_id)

```

```

VALUES (1111, 777111, '2002-02-01', 'Moscow', 'Moscow, Lesnaya st. 13',
1);

INSERT INTO ticket (ticket_number, seat_number, price, passenger_id)
VALUES (1001, 5, 1500.00, 1);

INSERT INTO trip (departure_date, departure_time, travel_time, bus_id,
route_id)
VALUES ('2025-05-06', '2025-05-06 08:00:00', '07:00', 1, 1);

INSERT INTO intermediate_stop (stop_number, settlement_id)
VALUES (1, 1), (2, 2);

INSERT INTO route_composition (route_id, stop_id)
VALUES (1, 1), (1, 2);

INSERT INTO driver (full_name, phone_number)
VALUES ('Ivan Ivanich', '79992220222');

INSERT INTO passport_driver (series, number, issue_date, issue_place,
registration_address, driver_id)
VALUES (1234, 123456, '1999-05-01', 'Pskov', 'Pskov, Krasniy ave. 1', 1);

INSERT INTO crew (driver_id, trip_id)
VALUES (1, 1);

INSERT INTO reason_for_disqualification (disease)
VALUES ('Heart');

INSERT INTO medical_checkup (checkup_date, status, driver_id, reason_id)
VALUES ('2025-04-01 09:00:00', 'Passed', 1, NULL);

```

Схема логической модели базы данных, сгенерированная в Generate ERD:

