МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГТУ», ВГТУ)

Факультет информационных технологий и компьютерной безопасности

Кафедра компьютерных интеллектуальных технологий проектирования

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

По дисциплине: «Системы хранения и обработки данных»

Тема: «Разработка физической структуры базы данных»

Выполнил работу студент группы змиИВТ-241: Москвитина А.Г.

подпись, дата

Принял: Короленко В.В.

подпись, дата

Воронеж 2024

Цель работы: изучить и освоить процесс разработки физической структуры базы данных с использованием системы управления базами данных Postgres.

Основные задачи:

* создание таблиц в СУБД Postgres с помощью SQL-запросов с атрибутами, связями, первичными и внешними ключами в соответствии с разработанной в предыдущей лабораторной работе логической структурой базы данных;
* построение графической диаграммы для отображения физической структуры базы данных.

**Ход работы:**

После изучения основ работы с СУБД PostgreSQL и инструментария для работы с СУБД DBeaver, были сформированы SQL-запросы для создания таблиц в СУБД Postgres, соответствующих логической структуре базы данных, разработанной в предыдущей лабораторной работе.

SQL-запросы:

CREATE TABLE Автомобиль (

Автомобиль\_ID SERIAL PRIMARY KEY,

Марка VARCHAR(50) NOT NULL,

Модель VARCHAR(50) NOT NULL,

Год\_выпуска INT NOT NULL,

Номерной\_знак VARCHAR(10) NOT NULL,

Владелец\_ID INT NOT NULL REFERENCES "Владелец автомобиля"(Владелец\_Автомобиля\_ID),

Страховая\_стоимость\_ID INT NOT NULL REFERENCES "Страховая стоимость автомобиля"(Страховая\_стоимость\_автомобиля\_ID)

);

Данный запрос создает таблицу "Автомобиль" в базе данных, определяет структуру таблицы с заданными полями и их типами данных.

Элементы запроса:

* CREATE TABLE – ключевое слово, указывающее на создание новой таблицы.
* Автомобиль\_ID — это столбец таблицы, который будет автоматически генерировать уникальные значения (с помощью типа данных SERIAL).
* SERIAL – тип данных, который обеспечивает автоинкремент и генерацию последовательных уникальных значений для столбца Автомобиль\_ID.
* PRIMARY KEY – ограничение, указывающее, что столбец Автомобиль\_ID является первичным ключом.
* Марка – столбец таблицы, который будет содержать значения марки автомобиля типа VARCHAR с максимальной длиной в 50 символов.
* Модель – аналогично столбцу Марка, но для хранения марки автомобиля.
* Год выпуска – столбец типа INT для хранения даты выпуска автомобиля.
* Номерной знак – столбец типа VARCHAR с максимальной длиной в 10 символов для хранения номера автомобиля.
* Владелец\_ID – столбец типа INT для хранения информации о владельце, является внешним ключом и указывает на идентификатор владельца из таблицы "Владелец автомобиля"

CREATE TABLE "Владелец автомобиля" (

Владелец\_Автомобиля\_ID SERIAL PRIMARY KEY,

Имя VARCHAR(50) NOT NULL,

Фамилия VARCHAR(50) NOT NULL,

Дата\_рождения DATE NOT NULL,

Адрес VARCHAR(100) NOT NULL,

Номер\_телефона VARCHAR(20) NOT NULL

);

Этот запрос создает таблицу "Владелец автомобиля" в БД.

Элементы запроса:

* CREATE TABLE - ключевое слово, указывающее на создание новой таблицы.
* Владелец\_Автомобиля\_ID: столбец типа SERIAL, который будет автоматически генерировать уникальные идентификаторы для каждого владельца.
* Имя: столбец типа VARCHAR(100), который будет хранить имя владельца.
* Фамилия: столбец типа VARCHAR(100), который будет хранить фамилию владельца автомобиля.
* Дата\_рождения: столбец типа DATE, который будет хранить информацию о дате рождения владельца.
* Адрес: столбец типа VARCHAR(100), который будет адрес владельца.
* Номер телефона; столбец типа VARCHAR(20), который будет хранить информацию о номере телефона владельца.

CREATE TABLE Водитель (

Водитель\_ID SERIAL PRIMARY KEY,

Имя VARCHAR(50) NOT NULL,

Фамилия VARCHAR(50) NOT NULL,

Дата\_Рождения DATE NOT NULL,

Адрес VARCHAR(100) NOT NULL,

Номер\_телефона VARCHAR(20) NOT NULL,

Автомобиль\_ID INT NOT NULL REFERENCES Автомобиль(Автомобиль\_ID)

);

Этот запрос, как и остальные создает таблицу, но со следующими столбцами:

* Водитель\_ID: столбец типа SERIAL, который будет автоматически генерировать уникальные идентификаторы для каждого водителя.
* Имя: столбец типа VARCHAR(100), который будет хранить имя владельца.
* Фамилия: столбец типа VARCHAR(100), который будет хранить фамилию владельца автомобиля.
* Дата\_рождения: столбец типа DATE, который будет хранить информацию о дате рождения владельца.
* Адрес: столбец типа VARCHAR(100), который будет адрес владельца.
* Номер телефона; столбец типа VARCHAR(20), который будет хранить информацию о номере телефона владельца.
* Автомобиль\_ID; столбец типа INT для хранения информации о автомобиле, является внешним ключом и указывает на идентификатор владельца из таблицы "Автомобиль"

CREATE TABLE "Нарушение ПДД" (

Нарушение\_ПДД\_ID SERIAL PRIMARY KEY,

Описание\_нарушения VARCHAR(100) NOT NULL,

Дата\_и\_время\_нарушения TIMESTAMP NOT NULL,

Место\_нарушения VARCHAR(100) NOT NULL,

Водитель\_ID INT NOT NULL REFERENCES Водитель(Водитель\_ID)

);

Данный запрос создает таблицу "Нарушение ПДД".

Элементы запроса:

* Нарушение\_ПДД\_ID: SERIAL PRIMARY KEY. Здесь элемент является первичным ключом. Он уникально идентифицирует каждую запись в таблице.
* Дата\_и\_время\_нарушения: TIMESTAMP. Это колонка, тип данных которой предназначен для хранения даты и времени.
* Место\_нарушения: VARCHAR(100). Этот столбец, который сохраняет информацию о месте нарушения.
* Водитель\_ID: INTEGER REFERENCES Водитель(Водитель\_ID). Это столбец, который является внешним ключом и ссылается на идентификатор абитуриента из таблицы "Водитель".

CREATE TABLE Штраф (

Штраф\_ID SERIAL PRIMARY KEY,

Сумма\_штрафа DECIMAL(10,2) NOT NULL,

Дата\_оплаты\_штрафа DATE NOT NULL,

Нарушение\_ПДД\_ID INT NOT NULL REFERENCES "Нарушение ПДД"(Нарушение\_ПДД\_ID)

);

Запрос создает таблицу "Штраф" со следующими составляющими:

* Штраф\_ID: столбец типа SERIAL, который будет использоваться в качестве первичного ключа таблицы.
* Сумма\_штрафа: столбец типа DECIMAL (10, 2), предназначенный для хранения суммы штрафа.
* Дата\_оплаты\_штрафа: столбец типа DATE, который будет хранить дату оплаты.
* Нарушение\_ПДД\_ID; столбец типа INTEGER, который будет ссылаться на столбец " Нарушение\_ПДД\_ID " в таблице " Нарушение ПДД ".

CREATE TABLE "Страховая стоимость автомобиля" (

Страховая\_стоимость\_автомобиля\_ID SERIAL PRIMARY KEY,

Сумма\_страхования DECIMAL(10,2) NOT NULL,

Дата\_начала\_страхования DATE NOT NULL,

Дата\_окончания\_страхования DATE NOT NULL

);

Последний запрос создает таблицу "Страховая стоимость автомобиля" со следующими составляющими:

* Страховая\_стоимость\_автомобиля\_ID: столбец типа SERIAL, который будет использоваться в качестве первичного ключа таблицы.
* Сумма\_страхования: столбец типа DECIMAL (10, 2), предназначенный для хранения суммы страхования.
* Дата\_начала\_страхования: столбец типа DATE, который будет хранить дату начала страхования.
* Дата\_окончания\_страхования; столбец типа DATA, который будет хранить информацию о дате окончания страхования.

После создания запросов необходимо обернуть созданную базу данных в docker-контейнер, содержимое файла docker-compose.yml указано ниже:

version: '3'

services:

postgres:

image: postgres:latest

environment:

POSTGRES\_DB: dbname

POSTGRES\_USER: dbuser

POSTGRES\_PASSWORD: dbpass

ports:

- "5432:5432"

volumes:

- pg\_data:/var/lib/postgresql/data

volumes:

pg\_data:

На рисунке 1 изображена диаграмма, отображающая физическую структуру базы данных.

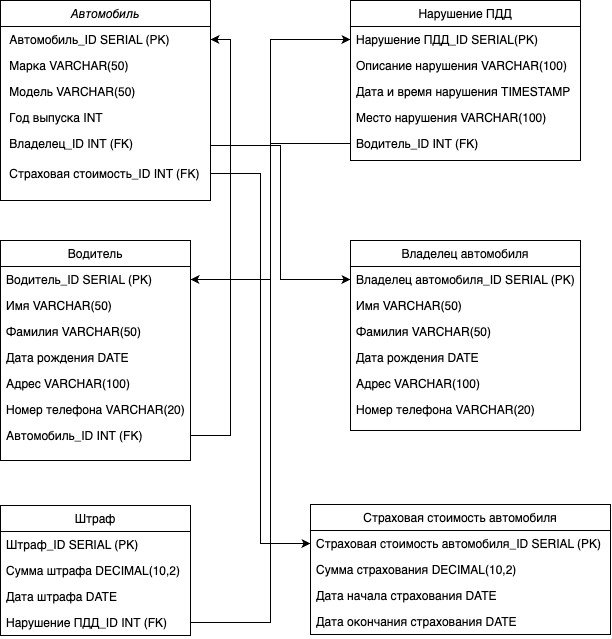


Рисунок 1 – Диаграмма

На диаграмме отображены таблицы, где каждый прямоугольник представляет отдельную таблицу. Внутри прямоугольника указаны атрибуты таблицы, такие как названия столбцов и их тип данных. Также были указаны индексы, которые отображают первичные и внешние ключи таблиц.

Связи между таблицами представляются линиями, указывающими на поля, которые являются внешними ключами. Это отображает связь между таблицами и определяет, как они связаны между собой.

Вывод: в данной работе были изучен и освоен процесс разработки физической структуры базы данных с использованием системы управления базами данных Postgres.

**Контрольные вопросы:**

1. Особенности PostgreSQL.

Зачастую выделяют следующие особенности PostgreSQL:

- Надежность и целостность данных

- Расширяемость и поддержка пользовательских типов данных

- Гибкость в моделях индексирования и запросов

- Масштабируемость и поддержка больших объемов данных

- Поддержка ACID-транзакций

- Множество расширений и дополнительных функций

- Открытый и бесплатный для использования.

1. Что такое DDL?

DDL (Data Definition Language) - это язык определения данных, который используется в системах управления базами данных для создания, изменения и удаления структур данных, таких как таблицы, индексы, представления и т. д. DDL включает в себя команды, такие как CREATE, ALTER и DROP, которые позволяют определить схему базы данных и ее объекты.

1. Что такое DML?

DML (Data Manipulation Language) - это язык манипулирования данными, который используется в СУБД для выполнения операций над данными. Он позволяет добавлять, изменять, удалять и извлекать данные из таблиц. Основные команды DML включают INSERT (добавление данных), UPDATE (изменение данных), DELETE (удаление данных) и SELECT (извлечение данных). DML позволяет пользователям работать с данными внутри базы данных, обеспечивая изменение и получение нужной информации.

1. Что такое DCL?

DCL (Data Control Language) - группа операторов определения доступа к данным. Иными словами, это операторы для управления разрешениями, с помощью них мы можем разрешать или запрещать выполнение определенных операций над объектами базы данных.

1. С какой команды, как правило, начинается запрос?

Запросы обычно начинаются с команды SELECT, которая используется для получения данных из таблицы.

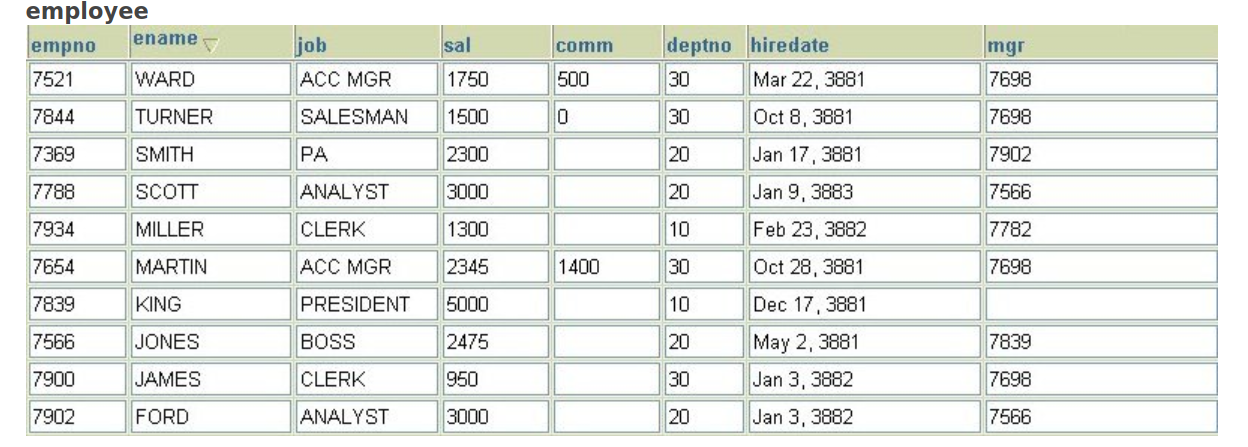
1. Какая команда используется для создания базы данных?

В PostgreSQL для создания новой базы данных используется команда CREATE DATABASE.

1. Какая команда используется для создания таблицы в базе данных?

Для создания новой таблицы используется команда CREATE TABLE.

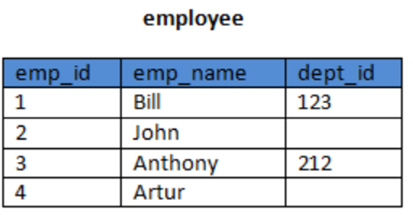
1. Существует следующая таблица:



Из таблицы employee необходимо показать сотрудников с empno 7844, 7788 и 7902 одним запросом. Какие из приведенных ниже запросов соответствуют данному условию?

* SELECT \* FROM employee WHERE sal IN (1500, 3000);

1. В базе данных компании хранятся данные о работниках и подразделениях, к которым они относятся.



Ваша задача: достать весь список работников, при этом все работники без подразделения должны быть в начале списка. Какой из представленных запросов выполнит эту задачу?

* SELECT \* FROM employee ORDER BY dept\_id NULLS FIRST;

1. Какие из следующих значений подходят для типа char в PostgreSQL?

* '1'
* 'a'
* 'char'

1. Какие из перечисленных конструкций запроса НЕ являются верными? Выбрать можно несколько вариантов ответа.

* Select список полей from список таблиц union Select список полей from список таблиц order by поля для сортировки
* Select список полей from список таблиц order by поля для сортировки union Select список полей from список таблиц order by поля для сортировки

1. Необходимо вычислить средний объем продаж (advance) книг и годовой доход (sales) от продажи книг по всем категориям (type) из таблицы titles. Какой из перечисленных фрагментов кода позволит решить поставленную задачу?

* Select avg (advance), sum (sales) from titles