

Лабораторная работа 4 Создание основных объектов базы данных в PostgreSQL с использованием DBeaver

1. Тема

Создание основных объектов реляционной базы данных (схем, таблиц) в СУБД PostgreSQL с использованием графического интерфейса DBeaver. Реализация логической модели в физической БД.

2. Цель работы

- **Теоретическая:** Закрепить понятия схемы, таблицы, атрибута (столбца), типа данных, первичного и внешнего ключа.
- **Практическая:** Приобрести навыки создания схем, таблиц, определения типов данных, первичных и внешних ключей в графическом интерфейсе DBeaver. Научиться переносить логическую модель данных в физическую базу данных.

3. Задачи

1. Подключиться к серверу PostgreSQL (192.168.1.14:5432) с помощью DBeaver.
2. Создать новую схему с уникальным именем.
3. В своей схеме создать таблицы в соответствии с выданным вариантом.
4. Определить первичные ключи для таблиц.
5. Установить связи между таблицами с помощью внешних ключей.
6. Убедиться в правильности созданной структуры базы данных.

4. Оборудование и программное обеспечение

- **Персональный компьютер.**
- **Программное обеспечение:** DBeaver.
- **Сервер БД:** PostgreSQL на 192.168.1.14:5432.
- **Учетная запись:** Предоставляется преподавателем.

5. Краткие теоретические сведения

Схема (Schema) — это пространство имен в базе данных, которое позволяет группировать объекты (таблицы, views, функции и т.д.) и управлять правами доступа. Каждый студент будет создавать объекты в своей собственной схеме.

Таблица (Table) — это основная структурная единица базы данных, состоящая из строк и столбцов. Каждая таблица соответствует одной сущности из ER-диаграммы.

Атрибут (Attribute) / Столбец (Column) — это свойство сущности. Каждый столбец имеет определенный тип данных (например, `integer`, `varchar`, `date`).

Первичный ключ (Primary Key, PK) — это столбец или набор столбцов, которые однозначно идентифицируют каждую строку в таблице.

Внешний ключ (Foreign Key, FK) — это столбец, который ссылается на первичный ключ в другой таблице, устанавливая связь между ними.

6. Порядок выполнения работы

1. Запустите DBeaver и создайте новое подключение к PostgreSQL, используя данные:
 - **Хост:** 192.168.1.14
 - **Порт:** 5432
 - **База данных:** (используйте базу по умолчанию, например, `postgres`)
 - **Пользователь/Пароль:** (предоставленные преподавателем)
2. В дереве подключения (слева) раскройте вашу базу данных -> **Схемы**.
3. **Создайте новую схему:** Щелкните правой кнопкой мыши на "Схемы", выберите "Создать новую схему". Введите имя схемы в формате `фамилия_инициалы` (например, `иванов_ai`). Нажмите "ОК".
4. Получите у преподавателя номер варианта задания (см. п. 9).
5. **Создавайте таблицы:** В своей схеме щелкните правой кнопкой мыши на "Таблицы" и выберите "Создать новую таблицу".
 - На вкладке **"Колонки"** добавляйте столбцы, указывая их Имя, Тип данных и Ограничения (галочка `Not Null` для обязательных полей).
 - **"Первичный ключ"** выберите столбец, который будет первичным ключом (обычно это `_id`), для него укажите в поле Key - Unique type значение `PRIMARY KEY`.
 - Нажмите "Сохранить", чтобы создать таблицу.
6. Повторите шаг 5 для всех таблиц вашего варианта.
7. **Создайте связи (Внешние ключи):**
 - Щелкните правой кнопкой мыши на таблице, которая должна ссылаться на другую (таблица с внешним ключом), выберите "Создать -> Внешний ключ".
 - Дайте имя внешнему ключу (например, `fk_orders_customer_id`).
 - На вкладке "Столбцы" выберите столбец из текущей таблицы (внешний ключ) и целевую таблицу со столбцом (первичный ключ), на который он ссылается.
 - Нажмите "Сохранить".

8. Убедитесь, что все таблицы и связи созданы корректно, просмотрев их в дереве объектов DBeaver.
9. Постройте диаграмму для вашей схемы, для этого в древе объектов щелкните правой кнопкой мыши по схеме и выберите пункт View Diagram.

7. Контрольные вопросы

1. Что такое схема в PostgreSQL и зачем она нужна?
2. Какие типы данных в PostgreSQL вы использовали при создании таблиц?
3. Что такое первичный ключ и каково его назначение?
4. Что такое внешний ключ и каково его назначение? Как он обеспечивает целостность данных?
5. Чем отличается процесс создания таблицы через графический интерфейс от написания SQL-запроса `CREATE TABLE` ?
6. Почему важно правильно выбирать типы данных для столбцов?

8. Содержание отчета

- Номер варианта и формулировка задания.
- Скриншоты:
 - Подключения к серверу в DBeaver.
 - Созданной схемы в древе объектов.
 - Диалоговых окон создания каждой таблицы (вкладка "Столбцы").
 - Диалоговых окон создания первичных ключей.
 - Диалоговых окон создания внешних ключей.
 - Итогового дерева объектов вашей схемы с всеми созданными таблицами.
- Ответы на контрольные вопросы.
- Выводы по работе.

9. Варианты заданий

Вариант 1: Онлайн-библиотека (упрощенная)

Создайте в своей схеме следующие таблицы:

- **authors**
 - `author_id` (integer, PRIMARY KEY)
 - `first_name` (varchar(50), NOT NULL)
 - `last_name` (varchar(50), NOT NULL)
- **books**
 - `book_id` (integer, PRIMARY KEY)
 - `title` (varchar(200), NOT NULL)

- `publication_year` (integer)
- `author_id` (integer, FOREIGN KEY ссылается на `authors(author_id)`)

Вариант 2: Система записи к врачу (упрощенная)

Создайте в своей схеме следующие таблицы:

- **patients**
 - `patient_id` (integer, PRIMARY KEY)
 - `full_name` (varchar(100), NOT NULL)
 - `date_of_birth` (date)
- **doctors**
 - `doctor_id` (integer, PRIMARY KEY)
 - `full_name` (varchar(100), NOT NULL)
 - `specialization` (varchar(100))
- **appointments**
 - `appointment_id` (integer, PRIMARY KEY)
 - `patient_id` (integer, FOREIGN KEY ссылается на `patients(patient_id)`)
 - `doctor_id` (integer, FOREIGN KEY ссылается на `doctors(doctor_id)`)
 - `appointment_date` (timestamp)

Вариант 3: Учет сотрудников и отделов

Создайте в своей схеме следующие таблицы:

- **departments**
 - `department_id` (integer, PRIMARY KEY)
 - `department_name` (varchar(100), NOT NULL)
- **employees**
 - `employee_id` (integer, PRIMARY KEY)
 - `first_name` (varchar(50), NOT NULL)
 - `last_name` (varchar(50), NOT NULL)
 - `position` (varchar(100))
 - `department_id` (integer, FOREIGN KEY ссылается на `departments(department_id)`)

Вариант 4: Кинотеатр (упрощенная)

Создайте в своей схеме следующие таблицы:

- **movies**
 - `movie_id` (integer, PRIMARY KEY)
 - `title` (varchar(150), NOT NULL)
 - `duration_minutes` (integer)

- **halls**

- `hall_id` (integer, PRIMARY KEY)
- `hall_name` (varchar(50))

- **sessions**

- `session_id` (integer, PRIMARY KEY)
- `movie_id` (integer, FOREIGN KEY ссылается на `movies(movie_id)`)
- `hall_id` (integer, FOREIGN KEY ссылается на `halls(hall_id)`)
- `start_time` (timestamp)

Вариант 5: Социальная сеть (упрощенная)

Создайте в своей схеме следующие таблицы:

- **users**

- `user_id` (integer, PRIMARY KEY)
- `username` (varchar(50), NOT NULL)
- `email` (varchar(100))

- **posts**

- `post_id` (integer, PRIMARY KEY)
- `user_id` (integer, FOREIGN KEY ссылается на `users(user_id)`)
- `post_text` (text)
- `created_at` (timestamp)

- **comments**

- `comment_id` (integer, PRIMARY KEY)
- `post_id` (integer, FOREIGN KEY ссылается на `posts(post_id)`)
- `user_id` (integer, FOREIGN KEY ссылается на `users(user_id)`)
- `comment_text` (text)