

# Лабораторная работа №5 DDL - язык определения данных в PostgreSQL

## Цель работы:

Освоить основные команды DDL (Data Definition Language) в PostgreSQL: CREATE, ALTER, DROP. Научиться создавать и модифицировать структуру базы данных.

**Программное обеспечение:** PostgreSQL, DBeaver

---

## Подготовительный этап

### 1. Установка соединения и выбор схемы

Перед началом работы выполните в DBeaver:

```
-- Просмотр текущей схемы
SELECT current_schema();

-- Установка вашей личной схемы как схемы по умолчанию
SET search_path TO ваша_схема;

-- Проверка
SELECT current_schema();
```

---

## Пример выполнения работы

### Создание связанных таблиц

```
-- 1. Создание таблицы "Фильмы"
CREATE TABLE movies (
    movie_id SERIAL PRIMARY KEY,
    title VARCHAR(150) NOT NULL,
    director VARCHAR(100),
    release_year INTEGER,
    duration_minutes INTEGER,
    budget DECIMAL(12,2)
```

```

);

-- 2. Создание таблицы "Кинозалы"
CREATE TABLE cinema_halls (
    hall_id SERIAL PRIMARY KEY,
    hall_name VARCHAR(50) NOT NULL,
    capacity INTEGER NOT NULL,
    has_3d BOOLEAN DEFAULT FALSE,
    vip_seats BOOLEAN DEFAULT FALSE
);

-- 3. Создание таблицы "Сеансы" (связывает фильмы и кинозалы)
CREATE TABLE screenings (
    screening_id SERIAL PRIMARY KEY,
    movie_id INTEGER REFERENCES movies(movie_id),
    hall_id INTEGER REFERENCES cinema_halls(hall_id),
    screening_date DATE NOT NULL,
    screening_time TIME NOT NULL,
    ticket_price DECIMAL(8,2) NOT NULL
);

```

## Изменение таблиц через ALTER TABLE

```

-- 1. Добавление нового столбца с рейтингом
ALTER TABLE movies ADD COLUMN rating DECIMAL(3,1);

-- 2. Добавление ограничения CHECK для года выпуска
ALTER TABLE movies ADD CONSTRAINT year_check
CHECK (release_year BETWEEN 1888 AND EXTRACT(YEAR FROM CURRENT_DATE));

-- 3. Изменение типа данных столбца
ALTER TABLE cinema_halls ALTER COLUMN hall_name TYPE VARCHAR(100);

-- 4. Добавление ограничения UNIQUE
ALTER TABLE cinema_halls ADD CONSTRAINT unique_hall_name UNIQUE
(hall_name);

-- 5. Добавление столбца с DEFAULT значением
ALTER TABLE screenings ADD COLUMN is_sold_out BOOLEAN DEFAULT FALSE;

```

## Работа с временной таблицей

```
-- Создание пробной таблицы
CREATE TABLE test_table (
    test_id SERIAL PRIMARY KEY,
    test_name VARCHAR(50),
    created_at TIMESTAMP DEFAULT NOW()
);

-- Вставка тестовых данных
INSERT INTO test_table (test_name) VALUES ('Test record 1');

-- Просмотр данных
SELECT * FROM test_table;

-- Удаление таблицы
DROP TABLE test_table;
```

---

## Варианты заданий для студентов

### Вариант 1: Библиотечная система

Создать таблицы:

1. authors (author\_id, name, country)
2. books (book\_id, title, year, author\_id) - ссылка на authors
3. readers (reader\_id, name, phone, registration\_date)

ALTER операции:

1. Добавить столбец "email" в таблицу readers
2. Сделать столбец "phone" в readers UNIQUE
3. Добавить CHECK ограничение на год издания (1900-текущий год)
4. Переименовать таблицу readers в library\_users
5. Удалить столбец "country" из authors

### Вариант 2: Учебный центр

Создать таблицы:

1. teachers (teacher\_id, name, specialization, hire\_date)
2. courses (course\_id, title, hours, teacher\_id) - ссылка на teachers
3. students (student\_id, name, group\_name, enrollment\_date)

ALTER операции:

1. Добавить столбец "price" в courses
2. Изменить тип столбца "hours" на DECIMAL(5,2)
3. Добавить DEFAULT значение для hire\_date (CURRENT\_DATE)
4. Добавить CHECK ограничение на hours (>0)
5. Переименовать столбец "group\_name" в "student\_group"

## Вариант 3: Интернет-магазин

Создать таблицы:

1. customers (customer\_id, name, email, registration\_date)
2. products (product\_id, name, price, category)
3. orders (order\_id, customer\_id, product\_id, quantity, order\_date)

ALTER операции:

1. Добавить столбец "description" в products
2. Сделать столбец "email" в customers UNIQUE и NOT NULL
3. Добавить CHECK ограничение на price (>0)
4. Добавить столбец "status" в orders с DEFAULT значением 'new'
5. Удалить столбец "category" из products

## Вариант 4: Больница

Создать таблицы:

1. doctors (doctor\_id, name, specialization, license\_number)
2. patients (patient\_id, name, birth\_date, phone)
3. appointments (appointment\_id, doctor\_id, patient\_id, appointment\_date, diagnosis)

ALTER операции:

1. Добавить столбец "department" в doctors
2. Сделать столбец "license\_number" UNIQUE
3. Добавить CHECK ограничение на birth\_date (должен быть в прошлом)
4. Переименовать таблицу appointments в medical\_visits
5. Добавить столбец "created\_at" с DEFAULT NOW() во все таблицы

## Вариант 5: Фитнес-центр

Создать таблицы:

1. trainers (trainer\_id, name, specialization, experience\_years)

2. members (member\_id, name, membership\_type, join\_date)
3. sessions (session\_id, trainer\_id, member\_id, session\_date, duration)

ALTER операции:

1. Добавить столбец "phone" в members
  2. Изменить тип столбца "duration" на INTEGER
  3. Добавить CHECK ограничение на experience\_years (>=0)
  4. Добавить столбец "price" в sessions
  5. Сделать столбец "membership\_type" NOT NULL
- 

## Пошаговый план выполнения работы

### Шаг 1: Подготовка

1. Откройте DBeaver и подключитесь к PostgreSQL
2. Выполните команду для установки вашей схемы:

```
SET search_path TO ваша_схема;  
SELECT current_schema(); -- для проверки
```

### Шаг 2: Создание таблиц

1. Создайте 3 таблицы согласно вашему варианту
2. Убедитесь, что таблицы созданы
3. Проверьте структуру таблиц

### Шаг 3: Модификация таблиц

1. Выполните 5 операций ALTER TABLE согласно вашему варианту
2. После каждой операции проверяйте изменения структуры
3. Фиксируйте возможные ошибки и их решения

### Шаг 4: Работа с временной таблицей

1. Создайте пробную таблицу
2. Добавьте в нее тестовые данные
3. Просмотрите данные
4. Удалите таблицу

### Шаг 5: Проверка и оформление отчета

1. Убедитесь, что все операции выполнены успешно

- 
2. Сохраните SQL-скрипт с выполненными командами
  3. Сделайте скриншоты структуры таблиц
- 

## Требования к отчету

1. **SQL-скрипт** со всеми выполненными командами
  2. **Скриншоты:**
    - Дерева объектов DBeaver с созданными таблицами
    - Структуры каждой таблицы после создания
    - Структуры таблиц после изменений
  3. **Описание возникших проблем и их решений**
- 

## Контрольные вопросы

1. Чем отличается **DDL** от **DML**?
2. Что такое **SERIAL** в PostgreSQL и для чего он используется?
3. Какие типы ограничений (**CONSTRAINTS**) вы знаете и для чего они нужны?
4. В чем разница между **DROP TABLE** и **DELETE FROM**?
5. Что делает команда **ALTER TABLE**?
6. Для чего используется **search\_path** в PostgreSQL?
7. Какие операции **ALTER TABLE** нельзя выполнить на заполненной таблице?
8. Что такое внешний ключ (**FOREIGN KEY**) и для чего он используется?