

# Лабораторная работа №6 Основы DML в PostgreSQL

## Цель работы:

Освоить основные команды DML (Data Manipulation Language): INSERT, SELECT, UPDATE, DELETE. Научиться добавлять, выбирать, изменять и удалять данные в таблицах.

**Программное обеспечение:** PostgreSQL, DBeaver

**Время:** 2 академических часа

---

## Теоретическая справка

**DML (Data Manipulation Language)** - язык управления данными:

- **INSERT** - добавление новых записей
  - **SELECT** - выборка данных
  - **UPDATE** - изменение существующих данных
  - **DELETE** - удаление записей
- 

## Пример выполнения работы (Вариант 0: Магазин электроники)

### 1. Подготовка базы данных

```
-- Установка схемы
SET search_path TO ваш_логин;

-- Создание таблиц
CREATE TABLE categories (
    category_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    description TEXT
);

CREATE TABLE products (
```

```
product_id SERIAL PRIMARY KEY,  
name VARCHAR(200) NOT NULL,  
price DECIMAL(10,2) NOT NULL,  
category_id INTEGER REFERENCES categories(category_id),  
in_stock BOOLEAN DEFAULT TRUE  
);
```

## 2. Оператор INSERT - добавление данных

```
-- Добавление данных в таблицу categories  
INSERT INTO categories (name, description) VALUES  
( 'Смартфоны', 'Мобильные телефоны и смартфоны'),  
( 'Ноутбуки', 'Портативные компьютеры'),  
( 'Планшеты', 'Планшетные компьютеры');  
  
-- Добавление данных в таблицу products  
INSERT INTO products (name, price, category_id, in_stock) VALUES  
( 'iPhone 14', 799.99, 1, TRUE),  
( 'Samsung Galaxy S23', 699.99, 1, TRUE),  
( 'MacBook Air', 999.99, 2, TRUE),  
( 'iPad Air', 599.99, 3, FALSE),  
( 'Dell XPS 13', 899.99, 2, TRUE);
```

## 3. Оператор SELECT - выборка данных

```
-- Выбор всех данных из таблицы  
SELECT * FROM products;  
  
-- Выбор конкретных столбцов  
SELECT product_id, name, price FROM products;  
  
-- Выбор с условием WHERE  
SELECT * FROM products WHERE price > 700;  
  
-- Выбор с сортировкой ORDER BY  
SELECT * FROM products ORDER BY price DESC;  
  
-- Выбор с ограничением количества строк LIMIT  
SELECT * FROM products LIMIT 3;  
  
-- Выбор с использованием логических операторов
```

```
SELECT * FROM products
WHERE price BETWEEN 600 AND 900
AND in_stock = TRUE;
```

-- Поиск по шаблону LIKE

```
SELECT * FROM products WHERE name LIKE '%Air%';
```

## 4. Оператор UPDATE - изменение данных

-- Обновление одной записи

```
UPDATE products SET price = 849.99 WHERE product_id = 1;
```

-- Обновление нескольких записей

```
UPDATE products SET in_stock = FALSE WHERE category_id = 3;
```

-- Обновление с использованием вычислений

```
UPDATE products SET price = price * 0.9 WHERE category_id = 2;
```

-- Обновление нескольких столбцов одновременно

```
UPDATE products
SET price = 749.99, in_stock = TRUE
WHERE product_id = 2;
```

## 5. Оператор DELETE - удаление данных

-- Удаление конкретной записи

```
DELETE FROM products WHERE product_id = 4;
```

-- Удаление по условию

```
DELETE FROM products WHERE in_stock = FALSE;
```

-- Удаление всех данных из таблицы (осторожно!)

```
-- DELETE FROM products;
```

---

## Варианты заданий для студентов

### Вариант 1: Библиотека

```
-- Таблицы:
CREATE TABLE books (
    book_id SERIAL PRIMARY KEY,
    title VARCHAR(200) NOT NULL,
    author VARCHAR(100),
    year_published INTEGER,
    is_available BOOLEAN DEFAULT TRUE
);

CREATE TABLE readers (
    reader_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100)
);
```

Задания:

1. INSERT: Добавить 5 книг и 3 читателей
2. SELECT:
  - Все доступные книги
  - Книги определенного автора
  - Книги, изданные после 2020 года
  - Читателей в алфавитном порядке
3. UPDATE:
  - Изменить статус книги на "не доступна"
  - Обновить email читателя
  - Увеличить год издания на 1 для всех книг
4. DELETE:
  - Удалить книгу по ID
  - Удалить всех читателей без email

## Вариант 2: Студенты и курсы

```
-- Таблицы:
CREATE TABLE students (
    student_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    group_name VARCHAR(20)
);
```

```
CREATE TABLE courses (  
    course_id SERIAL PRIMARY KEY,  
    course_name VARCHAR(100) NOT NULL,  
    credits INTEGER  
);
```

Задания:

1. INSERT: Добавить 4 курса и 6 студентов
2. SELECT:
  - Всех студентов определенной группы
  - Курсы с количеством кредитов больше 3
  - Студентов, отсортированных по фамилии
  - Первые 3 курса
3. UPDATE:
  - Изменить группу студента
  - Увеличить кредиты курса на 1
  - Исправить опечатку в названии курса
4. DELETE:
  - Удалить курс по ID
  - Удалить студентов без группы

## Вариант 3: Ресторан

```
-- Таблицы:  
CREATE TABLE menu_items (  
    item_id SERIAL PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    category VARCHAR(50),  
    price DECIMAL(8,2) NOT NULL,  
    is_vegetarian BOOLEAN DEFAULT FALSE  
);  
  
CREATE TABLE orders (  
    order_id SERIAL PRIMARY KEY,  
    table_number INTEGER,  
    order_date DATE DEFAULT CURRENT_DATE  
);
```

Задания:

1. INSERT: Добавить 6 блюд и 4 заказа
2. SELECT:
  - Все вегетарианские блюда
  - Блюда определенной категории
  - Блюда дороже 500 рублей
  - Заказы за сегодня
3. UPDATE:
  - Изменить цену блюда
  - Сделать блюдо вегетарианским
  - Изменить номер стола в заказе
4. DELETE:
  - Удалить блюдо по ID
  - Удалить заказы определенного стола

## Вариант 4: Фитнес-клуб

```
-- Таблицы:
CREATE TABLE trainers (
    trainer_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    specialization VARCHAR(100),
    hourly_rate DECIMAL(8,2)
);

CREATE TABLE members (
    member_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    membership_type VARCHAR(50),
    join_date DATE
);
```

Задания:

1. INSERT: Добавить 3 тренера и 5 клиентов
2. SELECT:
  - Тренеров определенной специализации
  - Клиентов с абонементом "премиум"
  - Тренеров с почасовой ставкой больше 1000
  - Клиентов, вступивших в этом месяце
3. UPDATE:

- Увеличить почасовую ставку тренера
- Изменить тип абонеента клиента
- Исправить специализацию тренера

#### 4. DELETE:

- Удалить тренера по ID
- Удалить клиентов без типа абонеента

## Вариант 5: Автомастерская

```
-- Таблицы:
CREATE TABLE services (
    service_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    duration_hours INTEGER,
    cost DECIMAL(8,2) NOT NULL
);

CREATE TABLE cars (
    car_id SERIAL PRIMARY KEY,
    brand VARCHAR(50) NOT NULL,
    model VARCHAR(50),
    year INTEGER,
    license_plate VARCHAR(20)
);
```

### Задания:

1. INSERT: Добавить 5 услуг и 4 автомобиля

2. SELECT:

- Услуги дешевле 5000 рублей
- Автомобили определенной марки
- Услуги, занимающие больше 2 часов
- Автомобили старше 2015 года

3. UPDATE:

- Увеличить стоимость услуги
- Изменить госномер автомобиля
- Уменьшить длительность услуги

4. DELETE:

- Удалить услугу по ID
- Удалить автомобили без госномера

---

# Пошаговый план выполнения работы

## Шаг 1: Подготовка

1. Установите схему: `SET search_path TO ваш_логин;`
2. Создайте таблицы согласно вашему варианту

## Шаг 2: Оператор INSERT

1. Добавьте тестовые данные в обе таблицы
2. Проверьте корректность добавления: `SELECT * FROM таблица;`

## Шаг 3: Оператор SELECT

Выполните не менее 5 различных запросов SELECT:

- Простая выборка всех данных
- Выборка с условием WHERE
- Выборка с сортировкой ORDER BY
- Выборка с ограничением LIMIT
- Выборка с использованием BETWEEN, LIKE

## Шаг 4: Оператор UPDATE

Выполните не менее 3 операций UPDATE:

- Обновление одной записи
- Обновление нескольких записей по условию
- Обновление с вычислениями

## Шаг 5: Оператор DELETE

Выполните 2 операции DELETE:

- Удаление по конкретному условию
- Удаление по общему условию

## Шаг 6: Проверка и отчет

Сохраните все выполненные команды и сделайте скриншоты результатов

---

## Требования к отчету



1. **SQL-скрипт** со всеми выполненными командами
  2. **Скриншоты** результатов выполнения:
    - Данных после INSERT
    - Результатов различных SELECT запросов
    - Состояния данных после UPDATE и DELETE
  3. **Описание** выполненных операций
- 

## Контрольные вопросы

1. В чем разница между DDL и DML?
2. Какие основные операторы DML вы знаете?
3. Как добавить несколько записей одним оператором INSERT?
4. Для чего используется оператор WHERE?
5. Чем отличается UPDATE от DELETE?
6. Что делает оператор ORDER BY?
7. Как ограничить количество возвращаемых строк?
8. Какие операторы используются в условиях WHERE?