

Урок 17 - DDL и DML

Типы данных в sql

PostgreSQL поддерживает разнообразные типы данных. Вот основные:

Тип данных	Назначение	Пример использования
SERIAL	Целое число с автоинкрементом	id SERIAL PRIMARY KEY
INTEGER / INT	Обычное целое число	age INT
NUMERIC(p,s)	Точное число: p — всего цифр, s — после запятой	price NUMERIC(10,2)
FLOAT	Число с плавающей точкой (не точное)	rating FLOAT
BOOLEAN	Логическое значение: true, false	is_active BOOLEAN
VARCHAR(n)	Строка длиной до n символов	name VARCHAR(100)
TEXT	Строка неограниченной длины	description TEXT
DATE	Дата (без времени)	birthdate DATE
TIMESTAMP	Дата и время (без часового пояса)	created_at TIMESTAMP
UUID	Уникальный идентификатор (128-битный)	user_id UUID DEFAULT gen_random_uuid()

- Для финансов **лучше использовать NUMERIC**
- VARCHAR(n) ограничивает длину, TEXT — нет.
- BOOLEAN можно задавать как true/false, 't'/'f', 1/0.
- SERIAL автоматически создаёт последовательность для автоинкремента.

Правила наименования в sql

запомните что sql регистрационно-независимый язык (case-insensitive)

0. Зарезервированные слова

- всегда пишем в верхнем регистре "SELECT"

1. Названия таблиц и колонок:

- Всегда пишем в **snake_case**: user_profiles, created_at
- Избегаем пробелов, русских букв, спецсимволов

- Не используем зарезервированные слова: select, table, user и т.п.

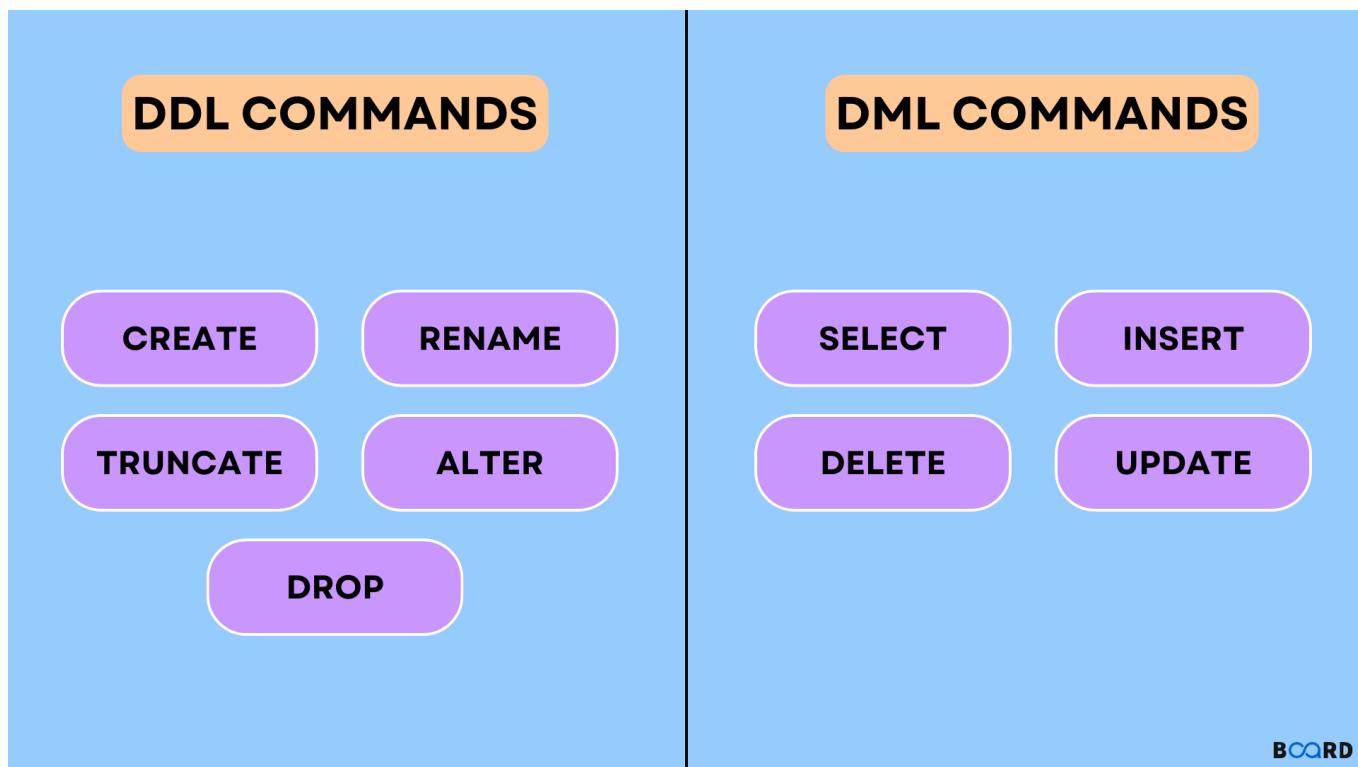
2. Типовые соглашения:

- Таблицы — **множественное число** (users, orders)
- Поля — **единичное число** (email, price)
- created_at, updated_at — стандартные для временных меток

3. Для внешних ключей:

- Названия полей: user_id, product_id
- Названия таблиц в связях: user_orders, product_reviews

DDL



DDL — это подмножество SQL, которое отвечает за **строктуру** базы данных.

Команды:

- CREATE — создать таблицу
- ALTER — изменить таблицу
- DROP — удалить таблицу

Пример:

```
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50),
```

```
    age INT  
);
```

DDL ШПАРГАЛКА

-- 1. CREATE TABLE – создать таблицу

```
CREATE TABLE users (  
    id SERIAL PRIMARY KEY,          -- Уникальный идентификатор,  
    автоинкремент  
    name VARCHAR(100) NOT NULL,     -- Обязательное поле  
    age INT,                      -- Целое число  
    email TEXT UNIQUE             -- Уникальное значение  
);
```

-- 2. ALTER TABLE – изменить таблицу

-- Добавить колонку

```
ALTER TABLE users ADD COLUMN address TEXT;
```

-- Переименовать колонку

```
ALTER TABLE users RENAME COLUMN address TO location;
```

-- Удалить колонку

```
ALTER TABLE users DROP COLUMN location;
```

-- Изменить тип данных у колонки

```
ALTER TABLE users ALTER COLUMN age TYPE VARCHAR(3);
```

-- 3. DROP TABLE – удалить таблицу (безвозвратно!)

```
DROP TABLE users;
```

-- 4. CREATE INDEX – создать индекс на колонку email

```
CREATE INDEX idx_users_email ON users(email);
```

-- 5. DROP INDEX – удалить индекс

```
DROP INDEX idx_users_email;
```

-- 6. RENAME TABLE – переименовать таблицу

```
ALTER TABLE users RENAME TO customers;
```

-- Быстрый пример: создать, изменить и удалить таблицу

```
CREATE TABLE products (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    price NUMERIC(10, 2) DEFAULT 0.00,
```

```
    created_at TIMESTAMP DEFAULT NOW()
);

-- Добавить колонку description
ALTER TABLE products ADD COLUMN description TEXT;

-- Удалить таблицу
DROP TABLE products;
```

DML

DML — это подмножество SQL, которое отвечает за **работу с данными** внутри таблиц.

Команды:

- INSERT — вставить данные
- UPDATE — изменить данные
- DELETE — удалить данные
- SELECT — получить данные

Примеры:

```
INSERT INTO users (name, age) VALUES ('Ali', 25);

SELECT * FROM users;

UPDATE users SET age = 26 WHERE name = 'Ali';

DELETE FROM users WHERE age < 18;
```

DML ШПАРГАЛКА

```
-- 1. INSERT – вставить данные в таблицу
INSERT INTO users (name, age, email)
VALUES ('Ali', 25, 'ali@example.com');

-- 2. SELECT – получить данные из таблицы
SELECT * FROM users;

-- Получить только имя и возраст пользователей старше 18 лет
SELECT name, age FROM users WHERE age > 18;

-- 3. UPDATE – изменить существующие данные
UPDATE users
```

```
SET age = 26
WHERE name = 'Ali';

-- 4. DELETE – удалить данные
DELETE FROM users
WHERE age < 18;

-- 5. Примеры с сортировкой и фильтрацией

-- Получить всех пользователей, отсортированных по возрасту по убыванию
SELECT * FROM users ORDER BY age DESC;

-- Получить всех пользователей с уникальными email
SELECT DISTINCT email FROM users;

-- 6. Работа с условиями

-- Найти пользователей с email, содержащим 'gmail.com'
SELECT * FROM users WHERE email LIKE '%gmail.com%';

-- Найти пользователей с возрастом от 20 до 30 включительно
SELECT * FROM users WHERE age BETWEEN 20 AND 30;

-- Найти пользователей с именем 'Ali' или 'Zara'
SELECT * FROM users WHERE name IN ('Ali', 'Zara');
```

Практика по DDL и DML в SQL

DDL (Data Definition Language)

1. Создайте таблицу `books` с полями:
 - `id` (автоинкремент, PRIMARY KEY)
 - `title` (обязательное поле)
 - `author`
 - `price` (с двумя знаками после запятой)
2. Добавьте в таблицу `books` колонку `published_at` типа `DATE`.
3. Переименуйте колонку `author` в `author_name`.
4. Удалите колонку `published_at`.
5. Измените тип поля `price` на `INTEGER`.
6. Переименуйте таблицу `books` в `library`.
7. Удалите таблицу `library`.
8. Создайте таблицу `students` с полями:

- id, name, age, email (email должен быть уникальным)
-

DML (Data Manipulation Language)

11. Вставьте 3 записи в таблицу `students`.
 12. Получите всех студентов из таблицы.
 13. Получите только имена студентов, старше 20 лет.
 14. Обновите возраст студента с именем 'Ali' до 25 лет.
 15. Удалите студентов младше 18 лет.
 16. Получите студентов, у которых email содержит `@gmail.com`.
 17. Получите студентов с возрастом от 18 до 30.
 18. Получите студентов по именам: 'Ali', 'Zara', 'John'.
 19. Получите студентов, отсортированных по возрасту по убыванию.
 20. Выведите только уникальные email студентов
-

Домашка

DDL (Data Definition Language)

1. Создайте таблицу `'books'` с полями:
 - `id` (автоинкремент, PRIMARY KEY)
 - `title` (обязательное поле)
 - `author`
 - `price` (с двумя знаками после запятой)
2. Добавьте в таблицу `'books'` колонку `'published_at'` типа `'DATE'`.
3. Переименуйте колонку `'author'` в `'author_name'`.
4. Удалите колонку `'published_at'`.

DML (Data Manipulation Language)

1. Вставьте 3 записи в таблицу `'students'`.
2. Получите всех студентов из таблицы.
3. Получите только имена студентов, старше 20 лет.
4. Обновите возраст студента с именем 'Ali' до 25 лет.
5. Удалите студентов младше 18 лет.