

# Шпаргалка по стандартным SQL операторам

## Выборка данных:

- **SELECT:** Извлекает данные из таблицы.
  - **Синтаксис:** `SELECT column1, column2, ... FROM table_name;`
  - **Пример:** `SELECT * FROM customers;` (извлекает все столбцы из таблицы `customers`)
- **DISTINCT:** Удаляет дубликаты из результата запроса.
  - **Синтаксис:** `SELECT DISTINCT column1, column2, ... FROM table_name;`
  - **Пример:** `SELECT DISTINCT city FROM customers;` (извлекает список уникальных городов из таблицы `customers`)
- **WHERE:** Фильтрует данные, удовлетворяющие заданному условию.
  - **Синтаксис:** `SELECT column1, column2, ... FROM table_name WHERE condition;`
  - **Пример:** `SELECT * FROM customers WHERE country = 'USA';` (извлекает информацию о клиентах из США)
- **ORDER BY:** Сортирует результат запроса.
  - **Синтаксис:** `SELECT column1, column2, ... FROM table_name ORDER BY column1 [ASC|DESC], column2 [ASC|DESC], ...;`
  - **Пример:** `SELECT * FROM products ORDER BY price ASC;` (сортирует товары по цене по возрастанию)
- **LIMIT:** Ограничивает количество возвращаемых строк.
  - **Синтаксис:** `SELECT column1, column2, ... FROM table_name LIMIT number;`
  - **Пример:** `SELECT * FROM orders LIMIT 10;` (извлекает 10 первых заказов)
- **OFFSET:** Пропускает заданное количество строк перед получением результата.
  - **Синтаксис:** `SELECT column1, column2, ... FROM table_name LIMIT number OFFSET offset;`
  - **Пример:** `SELECT * FROM orders LIMIT 10 OFFSET 5;` (извлекает 10 заказов, начиная с 6-го)

## Изменение данных:

- **INSERT:** Вставляет новые строки в таблицу.
  - **Синтаксис:** `INSERT INTO table_name (column1, column2, ...) VALUES (value1, value2, ...);`
  - **Пример:** `INSERT INTO customers (first_name, last_name, email) VALUES ('John', 'Doe', 'john.doe@example.com');` (добавляет нового клиента в таблицу)
- **UPDATE:** Обновляет значения в существующих строках.
  - **Синтаксис:** `UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;`
  - **Пример:** `UPDATE customers SET email = 'john.doe@new.com' WHERE id = 1;` (обновляет электронную почту клиента с ID 1)
    - **Синтаксис:** `DELETE FROM table_name WHERE condition;`
- **DELETE:** Удаляет строки из таблицы.
  - **Пример:** `DELETE FROM customers WHERE id = 1;` (удаляет клиента с ID 1)

## Другие операторы:

- **GROUP BY:** Группирует строки по одному или нескольким столбцам.
  - **Синтаксис:** `SELECT column1, column2, ... FROM table_name GROUP BY column1, column2, ...;`
  - **Пример:** `SELECT country, COUNT(*) FROM customers GROUP BY country;` (считает количество клиентов в каждой стране)
- **HAVING:** Фильтрует результаты группировки.
  - **Синтаксис:** `SELECT column1, column2, ... FROM table_name GROUP BY column1, column2, ... HAVING condition;`
  - **Пример:** `SELECT country, COUNT(*) FROM customers GROUP BY country HAVING COUNT(*) > 10;` (извлекает страны, в которых более 10 клиентов)
- **JOIN:** Соединяет строки из двух или более таблиц.
  - **Синтаксис:** `SELECT column1, column2, ... FROM table1 JOIN table2 ON condition;`
  - **Пример:** `SELECT customers.first_name, orders.order_date FROM customers JOIN orders ON customers.id = orders.customer_id;` (извлекает имя клиента и дату заказа для каждого заказа)
- **UNION:** Объединяет результаты нескольких запросов.
  - **Синтаксис:** `SELECT column1, column2, ... FROM table1 UNION SELECT column1, column2, ... FROM table2;`
  - **Пример:** `SELECT * FROM orders WHERE order_date < '2023-01-01' UNION SELECT * FROM orders WHERE order_date > '2023-01-31';` (извлекает заказы, даты которых не входят в январь 2023 года)