Данный оператор позволяет создавать и определять таблицу, синтаксис: CREATE TABLE [название таблицы] ([название столбца 1] [аргументы], [название столбца 2] [аргументы], ...);

Данный оператор используется для вставки новых строк в таблицу (внесения данных в таблицу), синтаксис: INSERT INTO [название таблицы] ([название столбца 1], [название столбца 2], ...) VALUES ([значение 1], [значение 2], ...);

Данный оператор используется для запросов к базе данных с целью извлечения из нее информации (вывод на экран значения/значений таблицы), синтаксис: SELECT [название столбца 1, название столбца 2, ...] FROM [название таблицы] WHERE ... ORDER BY ...;

Оператор WHERE задает условие, по которому будут выбираться строки из базы данных. Данная команда может использоваться для выборки строк с помощью SELECT, удаления строк с помощью DELETE, редактирования строк с помощью UPDATE.

Команда ORDERBY позволяет сортировать записи по определенному полю при выборе из базы данных, используется с оператором SELECT.

\* используется в операторе SELECT для выбора всей записей таблицы, пример использования: SELECT \* FROM months;

DISTINCT используется с SELECT для вывода значений из столбца без повторений, пример использования: SELECT DISTINCT name FROM months;

AUTOINCREMENT - свойство, которое позволяет создавать уникальный номер для каждой записи автоматически. По умолчанию начальное значения для атрибута автоинкремента равно 1, для каждой новой записи значение увеличивается на 1. Пример использования: CREATE TABLE months (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, days INTEGER);

PRIMARY KEY — первичный ключ, ограничение, позволяющее однозначно идентифицировать каждую запись в таблице SQL (определяет ключевое поле), пример использования: CREATE TABLE users (id INTEGER PRIMARY KEY); Дополнение: СУБД не позволит вам создать копию значения из ключевого поля, так как каждое значение ключевого поля уникально. Также не получится записать значение NULL в ключевое поле. Поле с атрибутом PRIMARY KEY автоматически имеет ограничение NOT NULL.

Данный оператор производит изменения в уже существующей записи или во множестве записей (обновляет элемент в таблице), синтаксис: UPDATE [название таблицы] SET [название столбца] = ... WHERE [условие обзора];

Данный оператор удаляет одну или несколько строк в таблице, синтаксис: DELETE FROM [название таблицы] WHERE [условие обзора];

Данный оператор удаляет таблицу, синтаксис: DROP TABLE [название таблицы];

Функция MIN возвращает минимальное значение поля среди найденных строк, синтаксис: SELECT MIN(поле) FROM имя\_таблицы WHERE условие;

Функция MAX возвращает максимальное значение поля среди найденных строк, синтаксис: SELECT MAX(поле) FROM имя\_таблицы WHERE условие;

Функция COUNT подсчитывает количество записей в таблице, синтаксис: SELECT COUNT(поле) FROM имя\_таблицы WHERE условие

Функция SUM суммирует значения указанного поля по всем выбранным строкам, синтаксис: SELECT SUM(поле) FROM имя\_таблицы WHERE условие;

Функция AVG возвращает среднее арифметическое по всем найденным записям, синтаксис: SELECT AVG(поле) FROM имя\_таблицы WHERE условие;

Команда GROUP BY позволяет группировать результаты при выборке из базы данных, синтаксис SELECT \* FROM имя\_таблицы WHERE условие GROUP BY поле\_для\_группировки

Команда LIMIT задает ограничение на количество записей, выбираемых из базы данных. Данная команда может использоваться совместно с командой SELECT, командой DELETE, и командой UPDATE.

Использование команды OFFSET позволяет пропустить указанное количество строк перед тем как выводить результаты запроса, пример использования: SELECT \* FROM название таблицы LIMIT 10 OFFSET 14;

Команда ALTER TABLE используется для добавления (используем ADD), удаления (используем DROP) или модификации (например, переименовываем, используя RENAME TO) колонки в уже существующей таблице, примеры использования: ALTER TABLE имя таблицы RENAME TO новое имя таблицы; ALTER TABLE имя таблицы ADD COLUMN название столбца [тип данных столбца];

NOT NULL - ограничение, при котором запись или записи не могут равняться значению NULL. СУБД не позволит вам создать запись со значением NULL в конкретном поле, если у данного поля имеется атрибут NOT NULL. Примеры использования:

Создание таблицы с полем, требующим отсутствия незаполненных ячеек в рамках данного поля: CREATE TABLE users (id INTEGER PRIMARY KEY, contact TEXT NOT NULL);

Выборка всех записей из таблицы users, у которых в поле contact отсутствует значение NULL: SELECT \* FROM users WHERE contact IS NOT NULL;

Атрибут UNIQUE - ограничение, при котором создание двух одинаковых ячеек в рамках одного поля становится невозможным. Значения ячеек в рамках поля с атрибутом UNIQUE могут равняться NULL. В этом состоит его главное отличие от атрибута ключевого поля. Пример использования: CREATE TABLE users (id INTEGER PRIMARY KEY, contact TEXT UNIQUE);

Оператор LIKE используется в предложении WHERE для поиска заданного шаблона в столбце. Синтаксис: SELECT [название столбцов] FROM [название таблицы] WHERE [название столбца] LIKE [условие];

Оператор GLOB используется для сопоставления только текстовых значений с шаблоном с использованием подстановочных знаков. Синтаксис: SELECT [название столбцов] FROM [название таблицы] WHERE [название столбца] GLOB [условие];

Оператор UNION используется для объединения результирующего набора из двух или более операторов SELECT. Чтобы использовать оператор UNION, каждый SELECT обязан иметь одинаковое количество выбранных столбцов с одинаковыми типами данных. Записи выведутся без повторений. Синтаксис:

SELECT [название столбцов] FROM [название таблицы] [WHERE [название столбца]]

UNION

SELECT [название столбцов] FROM [название таблицы] [WHERE [название столбца]];

Оператор UNION по умолчанию выбирает только разные значения. Чтобы разрешить повторяющиеся значения, используйте UNION ALL. Синтаксис:

SELECT [название столбцов] FROM [название таблицы] [WHERE [название столбца]]

UNION ALL

SELECT [название столбцов] FROM [название таблицы] [WHERE [название столбца]];