

- Просмотр доступных баз данных.

```
SHOW DATABASES;
```

- Создание новых баз данных.

```
CREATE DATABASE имя_табл;
```

- Выбирается база данных, необходимая для дальнейшей работы с ней.

```
USE имя_табл;
```

- Команда для удаления целой базы данных.

```
DROP DATABASE имя_табл;
```

- Создания новой таблицы в базе данных.

```
CREATE TABLE имя_табл(  
    столб_1 тип_данных_1,  
    столб_2 тип_данных_2,  
    столб_n тип_данных_n,  
);
```

- Изменения значений в записях таблицы.

```
UPDATE имя_табл  
SET поле_1= значение_1  
WHERE поле_1= значение_2;
```

- Ввод новых данных.

```
INSERT INTO имя_табл  
VALUES (значение_1, 'значение_2', значение_3);
```

- Используется для удаления существующих записей в таблице.

```
DELETE FROM имя_табл WHERE поле_n='значение';
```

- Добавить/удалить строку.

```
ALTER TABLE имя_табл  
ADD/DROP COLUMN столб_n тип_данных_n;
```

- Сортировка таблицы по полю

```
SELECT * FROM имя_табл
ORDER BY поле_сортир ASC/DESC;
```

(по умолчанию по возрастанию)

ASC – по возрастанию

DESC – по убыванию

- Оператор LIKE используется для поиска заданного шаблона в столбце.

В сочетании с оператором LIKE используются два подстановочных знака:

% - Знак процента представляет нулевой, один или несколько символов

_ - Подчеркнутый символ представляет собой один символ

```
SELECT column1, column2, ...
```

```
FROM table_name
```

```
WHERE columnn LIKE 'a%'; – найти значения с первой буквой а
```

Выражение	Описание
'%a'	Последняя буква а
'%a%'	Буква а есть в слове
'_a%'	Вторая буква а
'[abc]%'	Первая буква а/б/с
'[!abc]%'	Первая буква не а/б/с
'a%k'	Начинается с а заканчивается k

- Оператор WHERE служит для задания дополнительного условия выборки.

```
SELECT * FROM имя_табл
```

```
WHERE поле = значение; – отбираются значения равные указанному
```

>= – больше или равно указанному

<= – меньше или равно указанному

> – больше указанного

< – меньше указанного

- Отбираются значения не равные указанному.

```
SELECT * FROM имя_табл  
WHERE поле != значение;
```

- Отбираются значения, находящиеся между указанными.

```
SELECT * FROM имя_табл  
WHERE поле_n BETWEEN значение_1 AND значение_2;
```

- Отбираются значения, находящиеся не между указанными.

```
SELECT * FROM имя_табл  
WHERE поле_n NOT BETWEEN значение_1 AND значение_2;
```

- Отбираются значения, соответствующие указанным.

```
SELECT * FROM имя_табл  
WHERE поле_n IN (значение_1, значение_2);
```

- Отбираются значения, кроме указанных.

```
SELECT * FROM имя_табл  
WHERE поле_n NOT IN (значение_1, значение_2);
```

- Оператор AND отображает запись, если значения, разделенные символом AND, является TRUE.

```
SELECT поле_1, поле_2, ...  
FROM имя_табл  
WHERE значение_1 AND значение_2;
```

- Оператор OR отображает запись, если значения, разделенные OR, является TRUE.

```
SELECT поле_1, поле_2, ...  
FROM имя_табл  
WHERE значение_1 OR значение_2;
```

- Для получения только неповторяющихся данных.

```
SELECT DISTINCT поле FROM имя_табл;
```

- Вывод определённое количество записей.

```
SELECT * FROM имя_табл  
LIMIT числовое_значение;
```

- Оператор SUM() — функция, возвращающая сумму значений столбца таблицы. Используется только для числовых столбцов.

```
SELECT SUM(поле)  
FROM имя_табл;
```

- Оператор MAX() — функция возвращающая макс значение столбца таблицы.

```
SELECT MAX(поле)  
FROM имя_табл;
```

- Оператор MIN() — функция возвращающая миним значение столбца.

```
SELECT MIN(поле)  
FROM имя_табл;
```

- Оператор AVG() — Функция возвращающая среднее значение столбца. Применима только для числовых столбцов.

```
SELECT AVG(поле)  
FROM имя_табл;
```

- Оператор COUNT() — функция возвращающая количество записей (строк) таблицы. Запись функции с указанием столбца (синтаксис ниже) вернет количество записей конкретного столбца за исключением NULL записей.

```
SELECT COUNT(поле)  
FROM имя_табл;
```

- Оператор FULL JOIN осуществляет формирование таблицы из записей двух или нескольких таблиц.

```
SELECT  
    имя_табл_1.поле_1, имя_табл_2.поле_1, ...  
FROM имя_табл_1  
    FULL JOIN имя_табл_2;
```

- Оператор RIGHT JOIN осуществляет формирование таблицы из записей двух или нескольких таблиц. В операторе RIGHT JOIN, как и в операторе LEFT JOIN, важен порядок следования таблиц, так как от этого будет зависеть полученный результат.

```
SELECT
    имя_табл_1.поле_1, имя_табл_2.поле_1, ...
FROM имя_табл_1
    RIGHT JOIN имя_табл_2;
```

- Оператор LEFT JOIN осуществляет формирование таблицы из записей двух или нескольких таблиц. В операторе LEFT JOIN, как и в операторе RIGHT JOIN, важен порядок следования таблиц, так как от этого будет зависеть полученный результат.

```
SELECT
    имя_табл_1.поле_1, имя_табл_2.поле_1, ...
FROM имя_табл_2
    LEFT JOIN имя_табл_1;
```

- Оператор INNER JOIN формирует таблицу из записей двух или нескольких таблиц.

```
SELECT
    имя_табл_1.поле_1, имя_табл_2.поле_1, ...
FROM имя_табл_1
    INNER JOIN имя_табл_2;
```

Типы данных

В SQL типы данных разделяются на три группы: целые числа, дата и время, строковые, с плавающей точкой (дробные числа).

Целые числа, дата и время:

Типы данных SQL	Описание
DATE()	Дата в формате ГГГГ-ММ-ДД
DATETIME()	Дата и время в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС
TIMESTAMP()	Дата и время в формате timestamp. Однако при получении значения поля оно отображается не в формате timestamp, а в виде ГГГГ-ММ-ДД ЧЧ:ММ:СС
TIME()	Время в формате ЧЧ:ММ:СС
YEAR()	Год в двух значной или в четырехзначном формате.

Строковые типы:

Типы данных SQL	Описание
CHAR(size)	Строки фиксированной длиной (могут содержать буквы, цифры и специальные символы). Фиксированный размер указан в скобках. Можно записать до 255 символов
VARCHAR(size)	Может хранить не более 255 символов.
TINYTEXT	Может хранить не более 255 символов.
TEXT	Может хранить не более 65 535 символов.
BLOB	Может хранить не более 65 535 символов.
MEDIUMTEXT	Может хранить не более 16 777 215 символов.
MEDIUMBLOB	Может хранить не более 16 777 215 символов.
LONGTEXT	Может хранить не более 4 294 967 295 символов.
LOBLOB	Может хранить не более 4 294 967 295 символов.
ENUM(x,y,z,etc.)	Позволяет вводить список допустимых значений. Можно ввести до 65535 значений в SQL Тип данных ENUM список. Если при вставке значения не будет присутствовать в списке <i>ENUM</i> , то мы получим пустое значение. Ввести возможные значения можно в таком формате: ENUM ('X', 'Y', 'Z')
SET	SQL Тип данных SET напоминает <i>ENUM</i> за исключением того, что <i>SET</i> может содержать до 64 значений.

Типы с плавающей точкой (дробные числа) и целые числа:

Типы данных SQL	Описание
TINYINT(size)	Может хранить числа от -128 до 127
SMALLINT(size)	Диапазон от -32 768 до 32 767
MEDIUMINT(size)	Диапазон от -8 388 608 до 8 388 607
INT(size)	Диапазон от -2 147 483 648 до 2 147 483 647
BIGINT(size)	Диапазон от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807
FLOAT(size,d)	Число с плавающей точкой небольшой точности.
DOUBLE(size,d)	Число с плавающей точкой двойной точности.
DECIMAL(size,d)	Дробное число, хранящееся в виде строки.