

Строковые функции

CONCAT: объединяет строки. В качестве параметра принимает от 2-х и более строк, которые надо соединить:

```
SELECT CONCAT('Tom', ' ', 'Smith') -- Tom Smith
```

При этом в функцию можно передавать не только непосредственно строки, но и числа, даты - они будут преобразовываться в строки и также объединяться.

CONCAT_WS: также объединяет строки, но в качестве первого параметра принимает разделитель, который будет соединять строки: `SELECT CONCAT_WS(' ', 'Tom', 'Smith', 'Age:', 34)` -- Tom Smith Age: 34

LENGTH: возвращает количество символов в строке. В качестве параметра в функцию передается строка, для которой надо найти длину: `SELECT LENGTH('Tom Smith')` -- 9

LTRIM: удаляет начальные пробелы из строки. В качестве параметра принимает строку: `SELECT LTRIM(' Apple')`

RTRIM: удаляет конечные пробелы из строки. В качестве параметра принимает строку: `SELECT RTRIM(' Apple ')`

TRIM: удаляет начальные и конечные пробелы из строки. В качестве параметра принимает строку: `SELECT TRIM(' Tom Smith ')`

С помощью дополнительного оператора можно задать где имеется удалить пробелы: **BOTH** (в начале и в конце), **TRAILING** (только в конце), **LEADING** (только в начале): `SELECT TRIM(BOTH FROM ' Tom Smith ')`

LOCATE(find, search [, start]): возвращает позицию первого вхождения подстроки find в строку search. Дополнительный параметр start позволяет установить позицию в строке search, с которой начинается поиск подстроки find. Если подстрока search не найдена, то возвращается 0: `SELECT LOCATE('om', 'Tom Smith');` -- 2
`SELECT LOCATE('m', 'Tom Smith');` -- 3

LEFT: вырезает с начала строки определенное количество символов. Первый параметр функции - строка, а второй - количество символов, которые надо вырезать сначала строки: `SELECT LEFT('Apple', 3)` -- App

RIGHT: вырезает с конца строки определенное количество символов. Первый параметр функции - строка, а второй - количество символов, которые надо вырезать сначала строки: `SELECT RIGHT('Apple', 3)` -- ple

SUBSTRING(str, start [, length]): вырезает из строки str подстроку, начиная с позиции start. Третий необязательный параметр передает количество вырезаемых символов:
`SELECT SUBSTRING('Galaxy S8 Plus', 8),` -- S8 Plus
`SELECT SUBSTRING('Galaxy S8 Plus', 8, 2);` -- S8

SUBSTRING_INDEX(str, delimiter, count): вырезает из строки str подстроку. Параметр delimiter определяет разделитель внутри строки. А параметр count определяет, до какого вхождения разделителя надо вырезать подстроку. Если count положительный, то подстрока вырезается с начала, если count отрицательный, то с конца строки str:

```
1  SELECT SUBSTRING_INDEX('Galaxy S8 Plus', ' ', 1),      -- Galaxy
2  (SELECT SUBSTRING_INDEX('Galaxy S8 Plus', ' ', 2) ),    -- Galaxy S8
3  (SELECT SUBSTRING_INDEX('Galaxy S8 Plus', ' ', -2) );    -- S8 Plus
```

REPLACE(search, find, replace): заменяет в строке find подстроку search на подстроку replace. Первый параметр функции - строка, второй - подстрока, которую надо заменить, а третий - подстрока, на которую надо заменить:
`SELECT REPLACE('Galaxy S8 Plus', 'S8 Plus', 'Note 8')` -- Galaxy Note 8

INSERT(str, start, length, insert): вставляет в строку str, заменяя length символов с позиции start подстрокой insert. Первый параметр функции - строка, второй - позиция, с которой надо заменить, третий - сколько символов с позиции start надо заменить вставляемой подстрокой, четвертый параметр - вставляемая подстрока: `SELECT INSERT('Galaxy S9', 8, 3, 'Note 9'); -- Galaxy Note 9`

REVERSE: переворачивает строку наоборот: `SELECT REVERSE('123456789') -- 987654321`

LOWER: переводит строку в нижний регистр: `SELECT LOWER('Apple') -- apple`

UPPER: переводит строку в верхний регистр: `SELECT UPPER('Apple') -- APPLE`

SPACE: возвращает строку, которая содержит определенное количество пробелов

REPEAT(str, count): возвращает строку, которая содержит определенное количество повторов подстроки str. Количество повторов задается через параметр count: `SELECT REPEAT('ab', 5); -- ababababab`

LPAD(str, length, pad): добавляет слева от строки str некоторое количество символов, которые определены в параметре pad. Количество добавляемых символов вычисляется по формуле $length - LENGTH(str)$. Если параметр length меньше длины строки str, то эта строка усекается до length символов. `SELECT LPAD('Tom Smith', 13, '*');` --
*****Tom Smith

RPAD(str, length, pad): добавляет справа от строки str некоторое количество символов, которые определены в параметре pad. Количество добавляемых символов вычисляется по формуле $length - LENGTH(str)$. Если параметр length меньше длины строки str, то эта строка усекается до length символов. `SELECT RPAD('Tom Smith', 13, '*');` --
Tom Smith****

Например, возьмем таблицу:

```
1 CREATE TABLE Products
2 (
3     Id INT AUTO_INCREMENT PRIMARY KEY,
4     ProductName VARCHAR(30) NOT NULL,
5     Manufacturer VARCHAR(20) NOT NULL,
6     ProductCount INT DEFAULT 0,
7     Price DECIMAL NOT NULL
8 );
```

И при извлечении данных применим строковые функции:

```
1 SELECT UPPER(LEFT(Manufacturer,2)) AS Abbreviation,
2     CONCAT(ProductName, ' - ', Manufacturer) AS FullProdName
3 FROM Products
4 ORDER BY Abbreviation
```

	Abbreviation	FullProdName
	AP	iPhone X - Apple
	AP	iPhone 8 - Apple
	AP	iPhone 7 - Apple
	HM	Nokia 8 - HMD Global
	HU	Honor 10 - Huawei
	SA	Galaxy S9 - Samsung
	SA	Galaxy S8 - Samsung