

# Лекция 12

## **Функции работы со строками в MS SQL SERVER**

---

# Перчень функций

<b>ASCII</b>	<b>REPLACE</b>	<b>STUFF</b>
<b>CHAR</b>	<b>LEFT</b>	<b>STR</b>
<b>UNICODE</b>	<b>RIGHT</b>	<b>SPACE</b>
<b>NCHAR</b>	<b>LEN</b>	<b>LTRIM</b>
<b>CHARINDEX</b>	<b>DATALength</b>	<b>RTRIM</b>
<b>PATINDEX</b>	<b>REVERSE</b>	<b>LOWER</b>
<b>SUBSTRING</b>	<b>REPLICATE</b>	<b>UPPER</b>
<b>CONCAT</b>		

Функция **ASCII** возвращает ASCII-код крайнего левого символа строкового выражения, являющегося аргументом функции

Функция **CHAR** возвращает символ по известному ASCII-коду (от 0 до 255)

**Какой результат выполнения запросов?**

```
SELECT COUNT(DISTINCT ASCII(name))  
FROM Ships;
```

```
SELECT DISTINCT CHAR(ASCII(name))  
FROM Ships  
ORDER BY 1;
```

Функция **UNICODE**(*строковое выражение*) возвращает номер в кодировке Unicode первого символа строкового выражения.

Функция **NCHAR**(*целое*) возвращает символ по его номеру в кодировке Unicode. Приведем несколько примеров.

```
SELECT ASCII('a'), UNICODE('a');
```

```
SELECT CHAR(ASCII('a')), NCHAR(UNICODE('a'));
```

Функция CHARINDEX( **искомое\_выражение**,  
**строковое\_выражение** [, **стартовая\_позиция** ] )  
возвращает номер позиции в строковом  
выражении, с которой начинается искомое  
выражение, или ноль.

Необязательный целочисленный параметр стартовая\_позиция определяет позицию в строковом выражении, начиная с которой выполняется поиск. Если этот параметр опущен, поиск выполняется от начала строкового\_выражения.

**Какой результат выполнения запросов?**

```
SELECT name, CHARINDEX('sh', name)
```

```
FROM Ships
```

```
WHERE CHARINDEX('sh', name) > 0
```

```
SELECT CHARINDEX('a',name)
FROM Ships
WHERE name='California';
```

```
SELECT CHARINDEX('a', name, 3)
FROM Ships
WHERE name='California';
```

```
SELECT CHARINDEX('a',name) +1
FROM Ships
WHERE name='California';
```

```
SELECT CHARINDEX('a',name) first_a,
CHARINDEX('a', name, CHARINDEX('a', name)+1) second_a
FROM Ships
WHERE name='California';
```

Функция **PATINDEX** ( '%образец%' ,  
строковое\_выражение )

- только два аргумента;
- искомое\_выражение с символами % и могут  
включать РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ.

## Результаты одинаковые

```
SELECT name, PATINDEX('%sh%', name)  
FROM Ships  
WHERE PATINDEX('%sh%', name) > 0;
```

```
SELECT name, CHARINDEX('sh', name)  
FROM Ships  
WHERE CHARINDEX('sh', name) > 0
```

name	
Kirishima	5
Musashi	5
Washington	3

# Использование **регулярных** ОТНОШЕНИЙ

```
SELECT name  
FROM Ships  
WHERE PATINDEX('%e_e%', name) > 0;
```

<u>Name</u>
Revenge
Royal Sovereign

# Использование **регулярных** выражений

```
SELECT name, PATINDEX('%[a, o, u, e, i]%', name)  
FROM Ships  
WHERE PATINDEX('%[a, o, u, e, i]%', name) > 0;
```

name	
California	2
Haruna	2
Hiei	2
Iowa	1
Kirishima	2
...	...

# Использование **регулярных** выражений

```
SELECT name
```

```
FROM Ships
```

```
WHERE PATINDEX('%e_e%', name) > 0;
```

<u>Name</u>
Revenge
Royal Sovereign

## Последовательности в регулярных отношениях

```
SELECT name, PATINDEX('%[a, o, u, e, i]%', name)
FROM Ships
WHERE PATINDEX('%[a, o, u, e, i]%', name) > 0
```

```
SELECT name, PATINDEX('%[^a, o, u, e, i]%', name)
FROM Ships
WHERE PATINDEX('%[^a, o, u, e, i]%', name) > 0
```

```
SELECT name, PATINDEX('%[A-Z]%', name)
FROM Ships
WHERE PATINDEX('%[A-Z]%', name) > 0
```

```
SELECT name, PATINDEX('%[^A-Z]%', name)
FROM Ships
WHERE PATINDEX('%[^A-Z]%', name) > 0
```

**Функция SUBSTRING**(выражение, начальная позиция, длина) возвращает из выражения его часть заданной длины, начиная от заданной начальной позиции

**Какой результат выполнения запросов?**

```
SELECT name, SUBSTRING(name, 2, 3)
```

```
FROM Ships;
```

name	
California	ali
Haruna	aru
Hiei	iei
Iowa	owa
....	...

## Задача 136(3)

Для каждого корабля из таблицы Ships, в имени которого есть символы, не являющиеся латинской буквой, вывести: имя корабля, позиционный номер первого небуквенного символа в имени и сам СИМВОЛ.

# Решение задачи 136

```
SELECT  
name,  
PATINDEX('%[^A-Za-z]%', name) as npos,  
SUBSTRING(name, PATINDEX('%[^A-Za-z]%',  
name), 1) as litter  
FROM Ships  
WHERE PATINDEX('%[^A-Za-z]%', name) > 0
```

**Функция LEFT(строковое выражение,  
целочисленное выражение)**

вырезает заданное вторым аргументом  
число символов слева из строки,  
являющейся первым аргументом

**Функция RIGHT(строковое выражения,  
число символов)** возвращает заданное  
число символов справа из строкового  
выражен

## Какой результат выполнения запросов?

```
SELECT DISTINCT LEFT(name, 1)
FROM Ships
ORDER BY 1;
```

```
SELECT name
FROM Ships
WHERE LEFT(name, 1) = RIGHT(name, 1);
```

```
SELECT *
FROM (SELECT class +' '+ name AS cn
FROM Ships) x
WHERE LEFT(cn, 1) = RIGHT(cn, 1);
```

## Какой результат выполнения запросов?

```
SELECT DISTINCT LEFT(name, 1)
FROM Ships
ORDER BY 1;
```

```
SELECT name
FROM Ships
WHERE LEFT(name, 1) = RIGHT(name, 1);
```

```
SELECT *
FROM (SELECT class +' '+ name AS cn
FROM Ships) x
WHERE LEFT(cn, 1) = RIGHT(cn, 1);
```

cn
Iowa Missouri
North Carolina Washington
n

**Функция REPLACE(строка1, строка2, строка3)**

заменяет в *строке1* все

вхождения *строки2* на *строку3*

**Какой результат выполнения запросов?**

```
SELECT name, REPLACE(name, 'a', 'aa')
```

```
FROM Ships;
```

```
SELECT name, REPLACE(name, ' ', '-')
```

```
FROM Ships;
```

name	
California	Caaliforniaa
Haruna	Haarunaa
Hiei	Hiei
Iowa	Iowaa
Kirishima	Kirishimaa
Kongo	Kongo
Missouri	Missouri
Musashi	Musaashi

name	
Kongo	Kongo
Missouri	Missouri
Musashi	Musashi
New Jersey	New-Jersey
North Carolina	North-Carolina
Ramillies	Ramillies
Renown	Renown
Repulse	Repulse
Resolution	Resolution
Revenge	Revenge
Royal Oak	Royal-Oak
Royal Sovereign	Royal-Sovereign

# Задача 86

Для каждого производителя перечислить в алфавитном порядке с разделителем "/" все типы выпускаемой им продукции.  
Вывод: maker, список типов продукции

# Сформируем три вычисляемых столбца-флаги

maker	a	b	c
A			xPrinterx
A			xPrinterx
A			xPrinterx
A		xPCx	
A		xPCx	
A	xLaptopx		
A	xLaptopx		
B		xPCx	
B	xLaptopx		
C	xLaptopx		
D			xPrinterx
D			xPrinterx
E			xPrinterx
E		xPCx	
E		xPCx	
E		xPCx	

```
with t as(select maker,  
            max(case when type = 'Laptop' then 'xLaptopx' else '' end) a,  
            max(case when type = 'PC' then 'xPCx' else '' end) b,  
            max(case when type = 'Printer' then 'xPrinterx' else '' end) c  
            from Product  
            group by maker)  
select maker, a, b, c from t
```

maker	a	b	c
A	xLaptopx	xPCx	xPrinterx
B	xLaptopx	xPCx	
C	xLaptopx		
D			xPrinterx
E		xPCx	xPrinterx

```
with t as(select maker,  
max(case when type = 'Laptop' then 'xLaptopx' else '' end) a,  
max(case when type = 'PC' then 'xPCx' else '' end) b,  
max(case when type = 'Printer' then 'xPrinterx' else '' end) c  
from Product  
group by maker)
```

select maker, **a+ b+c** from t **Доработайте выражение для получения нужного результата**

maker	
A	xLaptopxxPCxxPrinterx
B	xLaptopxxPCx
C	xLaptopx
D	xPrinterx
E	xPCxxPrinterx

select maker, replace(a+b+c,'xx','/') types from t

maker	types
A	xLaptop/PC/Printerx
B	xLaptop/PCx
C	xLaptopx
D	xPrinterx
E	xPC/Printerx

select maker, replace(replace(replace(a+b+c,'xx','/'),'x','')) types from t

maker	types
A	Laptop/PC/Printer
B	Laptop/PC
C	Laptop
D	Printer
E	PC/Printer

**Функция LEN(строковое выражение)**  
возвращает число символов в строке (в MS  
SQL Server она не учитывает концевые  
пробелы)

**Функция DATALENGTH (строковое  
выражение)** возвращает число символов в  
строке с концевыми пробелами

```
SELECT name, DATALENGTH(name), Len(name )
FROM Passenger
```

name		
Alan Rickman	20	12
Angelina Jolie	20	14
Antonio Banderas	20	16
Brad Pitt	20	9
Bruce Willis	20	12
Bruce Willis	20	12
Catherine Zeta-Jones	20	20
Clint Eastwood	20	14

## Использование

Задачи, в которых требуется упорядочить (найти максимум и т. д.) в числовом порядке значения, представленные в текстовом формате. Например, номер места в самолете (2d) или скорость привода CD-ROM (24x).

```
SELECT '1a' AS place  
UNION ALL  
SELECT '2a'  
UNION ALL  
SELECT '11a'  
ORDER BY 1;
```

Проблема сортировки текста.

Места в самолете отсортированные по возрастанию :

**11a**

**1a**

**2a**

## Идея алгоритма

- Извлечь число из строки `LEFT(place, LEN(place) - 1)`.
- Привести его к числовому формату `CAST (LEFT(place, LEN(place) - 1) AS INT)`.
- Выполнить сортировку по приведенному значению `ORDER BY CAST(LEFT(place, LEN(place) - 1) AS INT)`.

```
SELECT *  
FROM (SELECT '1a' AS place  
      UNION ALL  
      SELECT '2a'  
      UNION ALL  
      SELECT '11a'  
      ) x  
ORDER BY CAST(LEFT(place, LEN(place) - 1) AS INT);
```

Задача. Найти наибольшее значение CD компьютеров.

Идея алгоритма

- Удалить из строки 'x'  
`replace(cd, 'x', '')` .
- Привести его к числовому формату  
`cast( replace(cd, 'x', '') as int))` .
- Найти наибольшее значение числовой части CD  
`Max(cast( replace(cd, 'x', '') as int)))` .
- Преобразовать числовое значение в строковый тип  
`cast(max( cast( replace(cd, 'x', '') as int)) as varchar(10))`.
- Выполнить конкатенацию с 'x'  
`cast(max( cast( replace(cd, 'x', '') as int)) as varchar(10)) + 'x'`

```
Select cast(max( cast( replace(cd, 'x', '') as int)) as varchar(10)) + 'x'  
From PC
```

## Задачи на подсчет количества букв

Сколько раз в названии корабля используется буква «а».

**Идея:** заменим каждую искомую букву двумя любыми символами, после чего посчитать разность длин полученной и искомой строки.

```
SELECT name, LEN(REPLACE(name, 'a', 'aa')) - LEN(name)
FROM Ships;
```

# Задача 131

Выбрать из таблицы Trip такие города, названия которых содержат минимум 2 разные буквы из списка (a,e,i,o,u) и все имеющиеся в названии буквы из этого списка встречаются одинаковое число раз.

1. Собрать города из town\_out и town\_to в одну таблицу.
2. Объединить списки городов, в которых встречается каждая буква, с указанием количества.
3. Сгруппировать по городам и отобрать города, которые встретились более одного раза и минимум и максимум количеств букв совпадают.

With

t as

```
(select town_from town from trip  
union select town_to town from trip),
```

r as

```
(SELECT town, 'a' b,  
LEN(REPLACE(town, 'a', 'aa'))-LEN(town) AS k  
FROM t where LEN(REPLACE(town, 'a', 'aa'))-LEN(town)>0  
union  
SELECT town, 'e' b,  
LEN(REPLACE(town, 'e', 'ee')) -LEN(town)AS k  
FROM t where LEN(REPLACE(town, 'e', 'ee'))-LEN(town)>0  
.....)
```

```
select town
```

```
from r
```

```
.....
```

```
having .....
```

select town, b, k  
from r

town	b	k
London	o	2
Moscow	o	2
Paris	a	1
Paris	i	1
Rostov	o	2
Singapore	a	1
Singapore	e	1
Singapore	i	1
Singapore	o	1
Vladivostok	a	1
Vladivostok	i	1
Vladivostok	o	2

select town  
from r

.....

having .....

```
select town
from r
group by town
having min(k)= max(k) and count(k)>1
```

Функция **REVERSE(выражение)**  
возвращает перевернутое выражение

Задача

Требуется определить позицию не первого, а последнего вхождения некоторого символа (или последовательности символов) в строке.

```
SELECT CHARINDEX('a', name) first_a  
FROM Ships  
WHERE name = 'California';
```

```
SELECT LEN(name) + 1 - CHARINDEX('a',  
REVERSE(name)) last_a  
FROM Ships  
WHERE name = 'California';
```

Функция **REPLICATE** (**выражение**, **количество**) повторяет первый аргумент такое число раз, которое задается вторым аргументом.

```
SELECT name,  
(LEN(REPLACE(name, 'a', REPLICATE('a', 2))) -  
LEN(name))  
FROM Ships
```

Функция **STUFF** (**строка1**, **стартовая позиция**, **L** , **строка2**) заменяет подстроку длиной **L**, которая начинается со стартовой позиции в строке1 на строку2.

```
SELECT STUFF(STUFF('20121119',5,0,'-'),8,0,'-');
```

Нуль означает вставку *строки2* в *строку1*, начиная с позиции, заданной вторым параметром

# Функции STR, SPACE

**STR(число с плавающей точкой [, длина [, число десятичных знаков ] ] )**

Функция **STR()** преобразует число к его символьному представлению.

При этом преобразовании выполняется округление, а длина задает длину результирующей строки.

<b>STR(3.3456, 5, 1)</b>	<b>3.3</b>
<b>STR(3.3456, 5, 2)</b>	<b>3.35</b>
<b>STR(3.3456, 5, 3)</b>	<b>3.346</b>
<b>STR(3.3456, 5, 4)</b>	<b>3.346</b>

**SPACE(число  
пробелов)**

**Задача**

**Изменить имя корабля: оставив в его имени 5 первых символов, дописать «\_» (нижнее подчеркивание) и год спуска на воду. Если в имени менее 5 символов, дополнить его пробелами.**

```
SELECT name, STUFF(name + SPACE(6), 6,  
LEN(name), '_' + STR(launched,4))  
FROM Ships;
```

Функции **LTRIM**(строковое выражение)  
и **RTRIM**(строковое выражение)

Отсекают, соответственно, начальные и  
конечные пробелы строкового выражения,  
которое неявно приводится к типу varchar.

Задача

Требуется построить такую строку:

*<имя пассажира>\_<идентификатор  
пассажира>*

```
SELECT name + '_' + CAST(id_psg AS VARCHAR)
FROM Passenger;
```

Alan Rickman	_9
--------------	----

Angelina Jolie	_15
----------------	-----

Antonio Banderas	_22
------------------	-----

Brad Pitt	_27
-----------	-----

Bruce Willis	_1
--------------	----

Bruce Willis	_31
--------------	-----

Catherine Zeta-Jones	_21
----------------------	-----

```
SELECT RTRIM(name) + '_' + CAST(id_psg AS  
VARCHAR)  
FROM Passenger;
```

Alan Rickman\_9

Angelina Jolie\_15

Antonio Banderas\_22

Brad Pitt\_27

Bruce Willis\_1

Bruce Willis\_31

Catherine Zeta-Jones\_21

Функции **LOWER**(*строковое выражение*)  
и **UPPER**(*строковое выражение*)  
преобразуют все символы аргумента,  
соответственно, к нижнему и верхнему  
регистру.

Эти функции оказываются полезными при  
сравнении регистрозависимых строк.

## Задачи

1. Вывести названия каждого сражения в таблице Battles, перенеся все конечные пробелы в начало названия.
2. Вывести названия кораблей из таблицы Ships, заменив любое количество повторяющихся пробелов в названиях кораблей на один пробел.
3. Вывести модель и cd с максимальной скоростью среди всех ПК;

## Указания к решению:

1. `SPACE(DATALength(name) - LEN(name))+RTRIM(name)`

ИЛИ

`REPLICATE(' ', 20-LEN(name))+RTRIM(name)`

2. `replace(replace(replace(replace(name, ' ', ''), ' ', ''), ' ', ''), ' ', '')`

3. `cast(max(cast(replace(cd, 'x', '') as int)) as varchar(10)) + 'x' cd`