

# **Тема 4. Язык SQL- Structured Query Language**

## **Часть 2 Встроенные функции MS SQL SERVER**

# Встроенные функции

---

- Имеются следующие типы функций одной строки:
  - функции для работы датой/временем,
  - символьные функции,
  - числовые функции,
  - функции преобразования

# Функции для работы с датой и временем

---

# Функции для работы с датой и временем

---

- **current\_timestamp** – возвращает системную дату и время
- **GETDATE()** – функция T-SQL, аналогична **current\_timestamp**.

## Пример

```
SELECT GETDATE()
```

```
SELECT current_timestamp
```

# Функции для работы с датой и временем

---

- **DATEADD (datepart, number, date)** – возвращает значение типа datetime, которое получается добавлением к date количества интервалов типа datepart, равного number.
- Допустимые значения datepart:
  - Год – yy, yuuu, year
  - Месяц – mm, m, month
  - День – dd, d, day
  - и др.

# Функции для работы с датой и временем

---

- **Пример.**

**Определить дату, которая наступит через неделю**

```
SELECT DATEADD(day, 7, GETDATE())
```

```
SELECT DATEADD(ww, 1, GETDATE())
```

# Функции для работы с датой и временем

---

- **DATEDIFF (datepart, startdate, enddate)** – возвращает интервал времени, прошедшего между двумя временными отметками
- Допустимые значения datediff аналогичны вариантам datepart функции DATEADD

# Функции для работы с датой и временем

---

- **Пример.**

**Определить количество дней,  
прошедших после оформления заказа**

```
SELECT onum, amt,  
        DATEDIFF(day, odate, GETDATE())  
FROM Orders
```



# Функции для работы с датой и временем

---

- **DATEPART (datepart, date)** – возвращает целое число, представляющее собой указанную аргументом *datepart* часть указанной даты *date*
- Допустимые значения *datepart* аналогичны вариантам *datepart* функции DATEADD, и значение *dw* – день недели

# Функции для работы с датой и временем

---

- **Пример.**

**Определить дни недели, в которые заключались заказы**

```
SELECT DISTINCT DATEPART(dw,  
    odate)  
FROM Orders
```

# Функции для работы с датой и временем

---

- **DAY(date)** — целочисленное представление дня указанной даты.
- **MONTH(date)** — целочисленное представление месяца указанной даты.
- **YEAR(date)** — целочисленное представление года указанной даты.

# Функции для работы с датой и временем

---

- **DAY(date)** — DATEPART(dd, date).
- **MONTH (date)** — DATEPART(mm, date)
- **YEAR(date)** — DATEPART(yy, date).

# Функции для работы с датой и временем

---

- **DATENAME(datepart, date)** — возвращает символьное представление составляющей (datepart) указанной даты (date).
- **SELECT**  
**DATENAME(weekday, '20191105' )+', '**  
**+ DATENAME(day, '20191105') +' '**  
**+ DATENAME(month, '20191105')**

# Функции для работы с датой и временем

---

- **DATENAME(datepart, date)** — возвращает символьное представление составляющей (datepart) указанной даты (date).
- **SELECT DATENAME(day, '20181231')**  
*Вывод: 31*
- **SELECT DATENAME(dayofyear, '20181231')**  
*Вывод: 365*

# Функции для работы с датой и временем

---

- **Пример.**

**Определить количество заказов по дням недели. Отсортировать по убыванию количества**

```
SELECT DATEPART(dw, odate),  
        COUNT(*) AS ordersCount  
FROM Orders  
GROUP BY DATEPART(dw, odate)  
ORDER BY ordersCount DESC
```

# Символьные функции

---



# Символьные функции

---

- **Имеются два типа СИМВОЛЬНЫХ функций:**
  - **возвращающие СИМВОЛЬНОЕ значение**
  - **возвращающие ЧИСЛОВОЕ значение.**

# Символьные функции

---

- **ASCII (строка)** – ASCII-код первого символа строки
- **CHAR (число)** - символ по коду ASCII
- **UNICODE(строка)** - Unicode-код первого символа строки
- **NCHAR(число)** – символ по коду Unicode
- **QUOTENAME(строка)** - конвертирует строку в формат Unicode

# Символьные функции

---

- **Пример.**

**Определить кол-во разных букв, с которых начинаются названия городов поставщиков**

```
SELECT COUNT(DISTINCT ASCII(city))  
FROM Salespeople
```

# Символьные функции

---

- **Пример.**

**Определить набор разных букв, с которых начинаются названия городов поставщиков**

```
SELECT DISTINCT CHAR(ASCII(city))  
FROM Salespeople
```

# Символьные функции

---

- Пример.

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

Результат: 224 1072

```
SELECT CHAR(ASCII('a')),
```

```
CHAR(UNICODE('a'))
```

Результат: a NULL

```
SELECT CHAR(ASCII('a')),
```

```
NCHAR(UNICODE('a'))
```

Результат: a a

# Символьные функции

---

- **CHARINDEX(подстрока, строка [, стартовая\_позиция ])**  
определяет порядковый номер символа, с которого начинается вхождение подстроки в строку
- **PATINDEX( '%образец%' , строка )**  
выполняет поиск подстроки в строке по указанному шаблону
- **LEN(строка)** - возвращает длину строки

# Символьные функции

---

- **Пример.**

**Определить поставщиков, в именах которых есть строка “abc”**

```
SELECT *  
FROM Salespeople  
WHERE CHARINDEX('abc',sname)>0
```

# Символьные функции

---

- **Пример.**

**Определить поставщиков, в именах которых есть строка “abc”**

```
SELECT *  
FROM Salespeople  
WHERE PATINDEX('%abc%',sname)>0
```

*Замечание:* концевые знаки % - обязательны. В шаблоне используются символы «%» и «\_»



# Символьные функции

---

- **Пример.**

**Определить поставщиков, имена которых содержат не менее 2-х букв «е», расположенных через один СИМВОЛ**

```
SELECT *  
FROM Salespeople  
WHERE PATINDEX('%e_e%',sname)>0
```

# Символьные функции

---

- **LEFT(строка, число)** - возвращает указанное число символов с начала строки
- **RIGHT(строка, число)** - возвращает указанное число символов с конца строки
- **SUBSTRING(выражение, начальная\_позиция, длина)**  
возвращает для строки подстроку указанной длины с заданного символа

# Символьные функции

---

- **Пример.**

**Определить поставщиков, имена которых начинаются и заканчиваются на одну и ту же букву**

**SELECT \***

**FROM Salespeople**

**WHERE**

**LEFT(sname,1)=RIGHT(sname,1)**

# Символьные функции

---

- **REVERSE(строка)** – возвращает строку, символы которой записаны в обратном порядке
- **REPLACE(строка1, строка2, строка3)** - заменяет в строке1 все вхождения строки2 на строку3
- **REPLICATE(строка, число)** - выполняет тиражирование строки определенное число раз

# Символьные функции

---

- **STUFF(строка1, стартовая\_позиция, длина, строка2)** - удаляет указанное число символов, заменяя новой подстрокой
- **STR(число с плавающей точкой [, длина [, число десятич. знаков ] ] )** - выполняет конвертирование значения числового типа в символьный формат

# Символьные функции

---

- **SPACE(число)** - возвращает указанное число пробелов
- **LTRIM(строка)** - удаляет пробелы в начале строки
- **RTRIM(строка)** - удаляет пробелы в конце строки
- **UPPER(строка) / LOWER(строка)** - переводит все символы строки в верхний / нижний регистр

# Функции преобразования данных

---

# **Функции преобразования данных**

**Для преобразования выражения  
одного типа данных в другой  
используется одна из двух  
функции SQL Server:**

- **CAST**
- **CONVERT.**



# Функции преобразования данных

Синтаксис **CAST** :

**CAST** (**expression** **AS** **data\_type** [(length)])

Пример:

```
SELECT CAST(odate AS varchar(50))  
FROM Orders
```

# Функции преобразования данных

Синтаксис **CONVERT**:

**CONVERT** (**data\_type** [(length)],  
**expression** [, **style**])

*Пример:*

```
SELECT CONVERT(varchar(50), odate)  
FROM Orders
```

# Функции преобразования данных

Функция **CONVERT**:

**CONVERT** (**data\_type** [(length)],  
**expression** [, **style**])

**style** - целочисленное выражение,  
определяющее стиль преобразования  
параметра **expression**.

Если стиль имеет значение NULL,  
возвращается NULL.

Диапазон определяется параметром  
*data\_type*

# CONVERT DateTime To Char /VarChar

---

Стиль	Схема	Результат
<b>0</b> или <b>100</b>	Default	мес дд гггг чч:мм АМ (РМ)
<b>1</b> или <b>101</b>	США	мм/дд/гггг
<b>2</b> или <b>102</b>	ANSI	гг.мм.дд
<b>3</b> или <b>103</b>	Британ. / франц.	дд/мм/гггг
<b>4</b> или <b>104</b>	Немецкий	дд.мм.гг
<b>14</b> или <b>114</b>	-	чч:мм:сс:ммм (24- часовой формат)

# Функции преобразования данных

Функция **CONVERT**:

*Пример*

Вывести данные по заказам продавца с идентификатором 2004, формат даты: без времени, дд.мм.гггг

```
SELECT *, CONVERT(varchar(50), odate,  
104)
```

```
FROM Orders
```

```
WHERE snum = 2004
```

# **CONVERT** Float/real To Char/VarChar

---

Значение	Выход
0 (по умолчанию)	Не более 6 разрядов. По необходимости используется экспоненциальное представление чисел.
1	Всегда 8 разрядов в экспоненциальном формате.
2	Всегда 16 разрядов в экспоненциальном формате.

# CONVERT Money To Char/VarChar

Значение	Выходные данные
0	<b>Без разделение на группы разрядов, после десятичного разделителя – 2 знака. <i>Пример 4235.98. <u>(по умолчанию)</u></i></b>
1	<b>Разделение на группы запятыми, после десятичного разделителя – 2 знака. <i>Пример 3,510.92.</i></b>
2	<b>Без разделение на группы разрядов, , после десятичного разделителя – 4 знака, <i>Пример 4235.9819.</i></b>

# Числовые функции

---



# Числовые функции

---

- **ABS(N)** - модуль  $n$ .  
**SELECT ABS(-15) as "Absolute"**
- **CEILING(N)** - наименьшее целое  $\geq n$ .  
**SELECT CEILING(15.7) as "Ceil"**
- **FLOOR(N)** - наибольшее целое  $\leq n$ .  
**SELECT FLOOR(15.7) as "Floor"**

# Числовые функции

---

- **SIN(N), COS(N), TAN(N)** - sin, cos или tan  $n$  (угол в радианах).  
`SELECT SIN(30*3.1415/180)`
- **EXP(N)** -  $e$  в степени  $n$ , где  $e = 2.71828183 \dots$   
`SELECT EXP(4) AS "e to the 4th power"`
- **LOG(N)** - Возвращает натуральный логарифм  $n \geq 0$ .  
`SELECT LOG(95) AS "Natural log of 95"`

# Числовые функции

---

- **LOG10(N)** - Возвращает десятичный логарифм *n*.

*n* может быть любым положительным числом.

```
SELECT LOG10(100) as "Log base 10 of  
100"
```

# Числовые функции

---

- **$M \% N$**  - Возвращает остаток от деления  $m$  на  $n$ .

**SELECT 11 % 4**

- **$\text{POWER}(M, N)$**  - Возвращает  $m$  в степени  $n$ . Основание  $m$  и показатель степени  $n$  любые числа, однако, если  $m < 0$ , то  $n$  должно быть целым.

**SELECT POWER(3,2)**

# Числовые функции

---

- **SIGN(N)** - Если  $n < 0$ , то функция возвращает -1. Если  $n = 0$ , то функция возвращает 0. Если  $n > 0$ , то функция возвращает 1.

**SELECT SIGN(-15) "Sign"**

- **SQUARE(N)** - Возвращает квадрат  $n$ .

**SELECT SQUARE(26)**

# Числовые функции

---

- **SQRT(N)** - Возвращает корень квадратный от  $n$ . Значение  $n$  не может быть отрицательным. SQRT возвращает в качестве результата число типа "real".

**SELECT SQRT(26)**

# Числовые функции

---

- **ROUND(N [,M])** - Возвращает *n* округленное до *m* позиции справа от десятичной точки. Если *m* опущено, *n* округляется до позиции 0. *m* может быть отрицательным для округления до позиции слева от десятичной точки. *m* должно быть целым числом.

```
SELECT ROUND(15.193,1) AS "Round"
```

# Числовые функции

<del>ACOS</del>	<del>вычисляет арккосинус</del>
ASIN	вычисляет арксинус
ATAN	вычисляет арктангенс
ATN2	вычисляет арктангенс с учетом квадратов
COT	котангенс угла
DEGREES	радианы в градусы
PI	значение "пи"
RADIANS	градусы в радианы
RAND	случайное число