Built-in Functions

Functions in SQL Server

Table of Contents

Functions in SQL Server

Overview

SQL Functions

- String Functions – for manipulating text
  - Either Table values or user input
  - E.g. concatenate column values
- Math Functions – calculations and working with aggregate data
  - E.g. perform geometry and currency operations
- Date and Time Functions
  - E.g. find length of timespan
- Miscellaneous Functions

String Functions

CONCAT, SUBSTRING, Etc.

String Functions

- Concatenation – combine strings

```
SELECT FirstName + ' ' + LastName
    AS [Full Name]
  FROM Employee
```

```
SELECT CONCAT(FirstName, ' ', LastName)
    AS [Full Name]
  FROM Employee
```

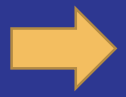
- **CONCAT** replaces **NULL** values it with empty string

String Functions (2)

- **SUBSTRING** – extract part of a string

```
SUBSTRING(String, StartIndex, Length)
```

```
SUBSTRING('SoftUni', 5, 3)
```



```
Uni
```

Index is 1-based!

- Example: get short summary of article

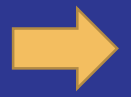
```
SELECT ArticleId, Author, Content,
       SUBSTRING(Content, 1, 200) + '...' AS Summary
  FROM Articles
```

String Functions (3)

- **REPLACE** – replace specific string with another

```
REPLACE(String, Pattern, Replacement)
```

```
REPLACE('SoftUni', 'Soft', 'Hard')
```



```
HardUni
```

- Example: censor the word blood from album names

```
SELECT REPLACE(Title, 'blood', '*****')
    AS Title
  FROM Album
```

String Functions (4)

- **LTRIM** & **RTRIM** – remove spaces from either side of string

```
LTRIM(String)
RTRIM(String)
```

- **LEN** – counts the number of characters

```
LEN(String)
```

- **DATALENGTH** – get number of used bytes (double for Unicode)

```
DATALENGTH(String)
```

String Functions (5)

- **LEFT** & **RIGHT** – get characters from beginning or end of string

```
LEFT(String, Count)
RIGHT(String, Count)
```

- Example: name shorthand (first 3 letters)

```
SELECT Id, Start,
       LEFT(Name, 3) AS Shorthand
  FROM Games
```

Problem: Obfuscate CC Numbers

- Our database contains credit card details for customers
- Provide a summary without revealing the serial numbers

ID	FirstName	LastName	PaymentNumber
1	Guy	Gilbert	5645322227179083
2	Kevin	Brown	4417937746396076
…	…		…

ID	FirstName	LastName	PaymentNumber
1	Guy	Gilbert	564532**********
2	Kevin	Brown	441793**********
…	…		…

Solution : Obfuscate CC Numbers

- We reveal the first 6 digits and obfuscate the rest

```
SELECT CustomerID,
       FirstName,
       LastName,
       LEFT(PaymentNumber, 6) + '**********'
  FROM Customers
```

- Bonus – create View for use by client apps

```
CREATE VIEW v_PublicPaymentInfo AS
…
```

String Functions (6)

- **LOWER** & **UPPER** – change letter casing

```
LOWER(String)
UPPER(String)
```

- **REVERSE** – reverse order of all characters in string

```
REVERSE(String)
```

- **REPLICATE** – repeat string

```
REPLICATE(String, Count)
```

String Functions (7)

- **CHARINDEX** – locate specific pattern (substring) in string

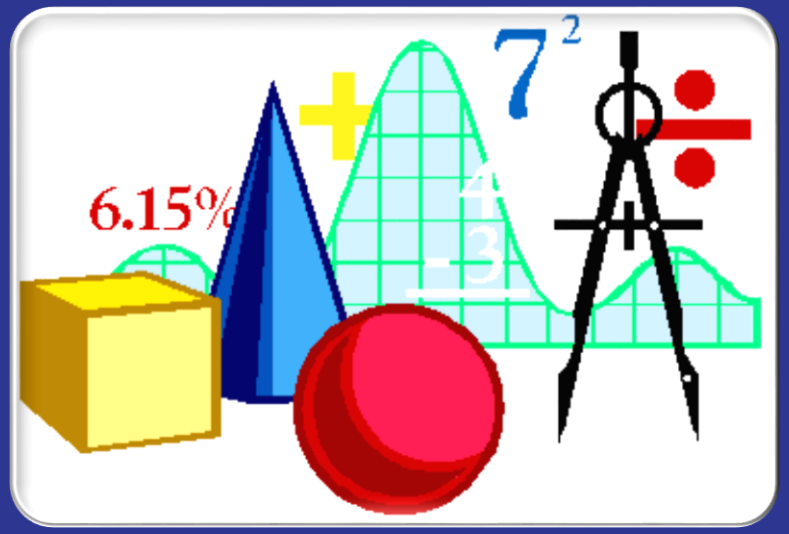
Optional, begins at 1

```
CHARINDEX(Pattern, String, [StartIndex])
```

- **STUFF** – insert substring at specific position

```
STUFF(String, StartIndex, Length, Substring)
```

Number of chars to delete



Math Functions

Arithmetic, PI, ABS, ROUND, etc.

Math Functions

- SQL Server supports basic arithmetic operations
  - Addition, subtraction, etc.
- Example: find area of triangles by given side and height

Id	A	H
1	2	4
2	1	18
3	4.5	3
4	8	12

Id	Area
1	4
2	9
3	6.75
4	48

```
SELECT Id,
       (A*H)/2 AS Area
  FROM Triangles
```

Math Functions (2)

- **PI** – get the value of Pi as float (15 –digit precision)

```
SELECT PI() --3.14159265358979
```

- **ABS** – absolute value

```
ABS(Value)
```

- **SQRT** – square root (result will be float)

```
SQRT(Value)
```

- **SQUARE** – raise to power of two

```
SQUARE(Value)
```

Example: Line Length

- Find the length of a line by given coordinates of end points

Id	X1	Y1	X2	Y2
1	0	0	10	0
2	0	0	5	3
4	-1	5	8	-3
5	18	23	8882	134

Id	Length
1	10
2	5.8309518948453
4	12.041594578792 3
5	8864.69497501183

```
SELECT Id,
       SQRT(SQUARE(X1-X2) + SQUARE(Y1-Y2))
    AS Length
  FROM Lines
```

Math Functions (3)

- **POWER** – raise value to desired exponent

```
POWER(Value, Exponent)
```

- **ROUND** – obtain desired precision
  - Negative precision rounds characters before decimal point

```
ROUND(Value, Precision)
```

- **FLOOR** & **CEILING** – return the nearest integer

```
FLOOR(Value)
CEILING(Value)
```

Problem: Pallets

- Calculate the required number of pallets to ship each item
  - BoxCapacity specifies how many items can fit in one box
  - PalletCapacity specifies how many boxes can fit in a pallet

Id	Name	Quantity	BoxCapacity	PalletCapacity
1	Perlenbacher 500ml	108	6	18
2	Perlenbacher 500ml	10	6	18
3	Chocolate Chips	350	24	3
4	Oil Pump	100	1	12

Number of pallets
1
1
5
9

Solution: Pallets

- Since we can't use half a box or half a pallet, we need to round up to the nearest integer value

```
SELECT
  CEILING(
    CEILING(
      CAST(Quantity AS float) /
      BoxCapacity) / PalletCapacity)
    AS [Number of pallets]
  FROM Products
```

Math Functions (4)

- **SIGN** – returns 1, -1 or 0, depending on value sign

```
SIGN(Value)
```

- **RAND** – get a random float value in range [0,1)
  - If **Seed** is not specified, one is assigned at random

```
RAND()
RAND(Seed)
```

Date Functions

GETDATE, DATEDIFF, DATEPART, etc.

Date Functions

- **DATEPART** – extract a segment from a date as an integer
  - Part can be any part and format of date or time

```
DATEPART(Part, Date)
```

year, yyyy, yy	YEAR(*Date*)
month, mm, m	MONTH(*Date*)
day, dd, d	DAY(*Date*)

- For a full list, see the official documentation

Problem: Quarterly Report

- Prepare sales data for aggregation by displaying yearly quarter, month, year and day of sale

InvoiceId	InvoiceDate	Total
1	2009-01-01 00:00:00.000	1.98
2	2009-01-02 00:00:00.000	3.96
3	2009-01-03 00:00:00.000	5.94
4	2009-01-06 00:00:00.000	8.91

InvoiceId	Total	Quarter	Month	Year	Day
1	1.98	1	1	2009	1
2	3.96	1	1	2009	2
3	5.94	1	1	2009	3
4	8.91	1	1	2009	6

Solution: Quarterly Report

- Use **DATEPART** to get the relevant parts of the date

```
SELECT InvoiceId, Total,
       DATEPART(QUARTER, InvoiceDate) AS Quarter,
       DATEPART(MONTH, InvoiceDate) AS Month,
       DATEPART(YEAR, InvoiceDate) AS Year,
       DATEPART(DAY, InvoiceDate) AS Day
  FROM Invoice
```

- This statement might be useful as a View

Date Functions (2)

- **DATEDIFF** – find difference between two dates
  - Part can be any part and format of date or time

```
DATEDIFF(Part, FirstDate, SecondDate)
```

- Example: Show employee experience

```
SELECT ID, FirstName, LastName,
       DATEDIFF(YEAR, HireDate, '2017/01/25')
    AS [Years In Service]
  FROM Employees
```

Date Functions (3)

- **DATENAME** – get a string representation of a date's part

```
DATENAME(Part, Date)
```

```
SELECT DATENAME(weekday, '2017/01/27')
```

- **DATEADD** – perform date arithmetic
  - Part can be any part and format of date or time

```
DATEADD(Part, Number, Date)
```

- **GETDATE** – obtain current date and time

```
SELECT GETDATE()
```

Other Functions

CAST, CONVERT, OFFSET, FETCH

Other Functions

- CAST & CONVERT – convert between data types

```
CAST(Data AS NewType)
CONVERT(NewType, Data)
```

- ISNULL – swap **NULL** values with a specified default value

```
ISNULL(Data, DefaultValue)
```

- Example: Display "Not Finished" for projects with no **EndDate**

```
SELECT ProjectID, Name,
       ISNULL(CAST(EndDate AS varchar), 'Not Finished')
  FROM Projects
```

Other Functions(2)

- OFFSET & FETCH – get only specific rows from the result set
  - Used in combination with ORDER BY for pagination

```
 SELECT ID, FirstName, LastName
     FROM Employees
ORDER BY ID
  OFFSET 10 ROWS
    FETCH NEXT 5 ROWS ONLY
```

Rows to skip

Rows to include

Summary

- SQL Server provides various built-in functions
- String functions allow us to manipulate strings
  - **CONCAT**, **LEFT**/**RIGHT**, **REPLACE**, etc.
- Math functions allow us to do various calculations
  - **PI**, **ABS**, **POWER**, **ROUND**, etc.
- Date functions allow us to work with dates easier
  - **DATEPART**, **DATEDIFF**, **GETDATE**, etc.
- Using Wildcards, we can obtain results by partial string matches