String and Scalar Functions

DSC 301: Lecture 6

February 10, 2021

Lecture Objectives

- String functions
- Arithmetic functions

String Functions

String functions include such functions concatenation, trimming, locating substrings, obtaining string length, replacing parts, capitalizing, padding, etc.

Concatenation

- `CONCAT(str1,str2)`: Concatenates two or more strings. If one of the strings is `null`, the entire result is `null`.

```
SELECT CONCAT(carrier,flight) as Flight from Flights;
```

  Note the use of aliasing in the example above.

- `CONCAT_WS(sep,str1,str2)`: Concatenates two or more strings with a separator (`sep`). If `sep` is null, the result is `null`. The example below will place a hyphen between the carrier and flight number.

```
SELECT CONCAT_WS('-',carrier,flight) as Flight from Flights;
```

Trimming

- `LTRIM(str)`: Returns the string (`str`) with left spaces removed. That is trims left spacing.

```
SELECT LTRIM('  American Airlines') as Airline;
```

- `RTRIM(str)`: Returns the string (`str`) with right spaces removed. That is trims right spacing.

```
SELECT RTRIM('  American Airlines  ') as Airline;
```

  Note left spaces still exist in the result of this example.

- `TRIM([both|leading|trailing] [remove FROM] str)`: Returns the string (`str`) with `remove` eliminated from the begining or ending (or both). If `remove` is omitted, only spaces are removed.

```
SELECT TRIM(LEADING '---' FROM '---American Airlines') as
    Airline;
```

Length of strings

- `LENGTH(str)`: returns number of characters in the string.

```
SELECT LENGTH('American Airlines') as 'Length of Name';
```

  Note difference in length between the previous query and the following.

```
SELECT LENGTH('   American Airlines') as 'Length of Name';
```

Substrings

- `LEFT(str,length)`: Returns a specified number of characters from the beginning of the string.

```
SELECT LEFT('American Airlines',4) as 'Begining 4 characters';
```

```
SELECT LEFT(tailnum,2) as 'Part tailnum' from Flights;
```

- `RIGHT(str,length)`: Returns a specified number of characters from the end of the string.

```
SELECT RIGHT('American Airlines',5) as 'Last 5 characters';
```

- `SUBSTRING(str,start,[length])`: Returns the a specified number of characters starting from the `start` position. If `length` is empty, it returns from the starting position to the end.

```
SELECT SUBSTRING('American Airlines', 4,5) as 'Substring';
```

```
SELECT SUBSTRING(tailnum, 2,3) as 'PartTail' from Flights;
```

- `SUBSTRING_INDEX(str,delimiter,number)`: Returns the substring before the specified `number` of occurences of the delimiter. When `number` is positive, it starts from the beginning and when negative from the end. Used to parse strings.

```
SELECT SUBSTRING_INDEX('https://www.une.edu', '.',-2) as '
    Domain Name';
```

- `LOCATE(lookfor,search,[start])`: Returns the **position** of the first occurrence of `lookfor` in the `search` string starting at the optional parameter `start`. If `start` is empty, the search starts at the beginning of the string.

```
SELECT LOCATE('irl','American Airlines') as 'Position of irl';
```

```
 SELECT LOCATE('O',dest,2) as 'Position of O' from Flights;
```

- `REPLACE(search,find, replace)`: Returns the `search` string with the characters `find` replaced by `replace`.

```
SELECT REPLACE('American Airlines','Air','Van') as 'New Str';
```

- `INSERT(str,start, length,insert)`: Returns the `search` string with the characters `find` replaced by `replace`.

```
SELECT INSERT('American Airlines',5,3,'Van') as 'New Str';
```

- `UPPER(str)`: Returns the `str` converted to uppercase letters.

```
SELECT UPPER('American Airlines') as 'All UPPERCASE';
```

- `LOWER(str)`: Returns the `str` converted to lowercase letters.

```
SELECT LOWER('American Airlines') as 'All lowercase';
```

- `LPAD(str, length, char)`: Returns a string of length `length` padded on the **left** with `char`. If `str` is longer than the `length`, the `str` is truncated and not padding occurs.

```
SELECT LPAD('American Airlines',20,'.') as 'Padded String';
```

```
SELECT LPAD('American Airlines',15,'.') as 'Truncated String';
```

- `RPAD(str, length, char)`: Returns a string of length `length` padded on the **right** with `char`. If `str` is longer than the `length`, the `str` is truncated and not padding occurs.

```
SELECT RPAD('American Airlines',20,'.') as 'Padded String';
```

- `SPACE(number)`: Returns the specified `number` of space characters.

```
SELECT CONCAT('American',  SPACE(10), 'Airlines') as 'New Str'
    ;
```

- `REPLACE(str,replace,with)`: Returns the string (`str`) with all occurances of the substring `replace`, replaced by `with`. For example, the code below will change 'American Airlines' to 'American Vanlines'.

```
SELECT REPLACE('American Airlines','Air','Van') as 'New Str';
```

- `REVERSE(str)`: Returns `str` with characters in reverse order.

```
SELECT REVERSE('American Airlines') as 'Reverse Str';
```

- `REPEAT(str,times)`: Returns the `str` repeated `times` number of times.

```
SELECT REPEAT('Air',3) as 'Repeat Str 3 times';
```

Numeric Functions

Basic Math

Basic math function include addition, subtraction, multiplication, division, modulo, and exponentiation.

- `+`: Adds values. For example to add 30 minutes to the air time of a flight, we write:

```
SELECT air_time + 30 as 'New Air Time' FROM Flights;
```

- `-`: Subtraction. Similar to subtract 30 minutes.

```
SELECT air_time - 30 as 'New Air Time' FROM Flights;
```

- `*`: Multiply. Scales values. For example, to convert to flight distance in miles to kilometers.

```
SELECT 1.609344*distance as 'Distance (km)' FROM Flights;
```

- `/`: Divide. Similar usage to multiplication. For example, lets determine the flight time in hours.

```
SELECT air_time / 60 as 'Travel time (hours)' FROM Flights;
```

Scalar Functions

- `abs(x)`: absolute value, e.g., abs(-17.4)
- `cbrt(x)`: cube root, e.g., cbrt(8)
- `ceil(x)`: smallest integer not less than argument, e.g., ceil(-42.8)
- `ceiling(x)`:same as `ceil(x)`
- `degree(x)`: radians to degrees
- `floor(x)`: largest integer not greater than argument, e.g., floor(-42.8)
- `exp(x)`: exp(1) is the natural number e
- `ln(x)`: natural logarithm, e.g., ln(2.0)
- `log(x)`: base 10 logarithm, e.g., log(100)
- `mod(y,x)`: remainder of y/x, e.g., mod(9,4)
- `pi()`: π
- `pow(x,y)`: raise a number to exponent y, e.g., pow(9.0, 3.0)
- `power(x,y)`: same as `pow`
- `radians(x)`: degree to radians
- `rand()`: random value between 0.0 and 1.0
- `round(x,p)`: round to p decimal places, e.g., round(42.4382, 2)
- `sign(x)`: sign of the argument, i.e., $(-1, 0, +1)$
- `sqrt(x)`: square root, e.g., sqrt(2.0)
- `truncate(x,p)`: truncate to s decimal places, trunc(42.4382,2)

Date Functions

Not covered in this lecture.

Next time

Aggregate functions