Day 43 - 90 days of Analytics: Date & Time Functions

In today's first video, we looked at the commonly used date and time functions with SQL queries

The following were mentioned

-The most difficult part when working with dates is to be sure that the format of the date you are trying to insert, matches the format of the date column in the database.

-MySQL comes with the following data types for storing a date or a date/time value in the database:

- **DATE** - format YYYY-MM-DD
- **DATETIME** - format: YYYY-MM-DD HH:MI:SS
- **TIMESTAMP** - format: YYYY-MM-DD HH:MI:SS
- **YEAR** - format YYYY or YY

-Some functions include

- **NOW ()**: Returns the current date and time as a value in **'YYYY-MM-DD HH:MM:SS'** or **YYYYMMDDHHMMSS** format, depending on whether the function is used in a string or numeric context. This value is expressed in the current time zone. Example
  `SELECT NOW();`
- **CURDATE()** : Returns the current date as a value in 'YYYY-MM-DD' or YYYYMMDD format, depending on whether the function is used in a string or in a numeric context. SELECT CURDATE();
- **DAY(date)** : Returns the day for date, in the range 1 to 31. Example
  `SELECT DAY('1998-02-05');`
- **MONTH(date)**: Returns the month for date, in the range 0 to 12. Example
  `SELECT MONTH('1998-02-03');`
- **YEAR(date)** : Returns the year for date, in the range 1000 to 9999, or 0 for the .zero. date. Example
  `SELECT YEAR('98-02-03');`
- **DATE_ADD(date,INTERVAL expr unit), DATE_SUB(date,INTERVAL expr unit)** : These functions perform date arithmetic.
  -The **date** is a DATETIME or DATE value specifying the starting date. The **expr** is an expression specifying the interval value to be added or subtracted from the starting date. The expr is a string; it may start with a '-' for negative intervals.
  -A **unit** is a keyword indicating the units in which the expression should be interpreted. (DAY, WEEK, MONTH, YEAR, …)
  -The INTERVAL keyword and the unit specifier are not case sensitive.
  Example `SELECT DATE_ADD('1999-01-01', INTERVAL 1 DAY);`
- **DATEDIFF(expr1,expr2)** : Returns the difference between two dates expressed as a value in days from one date to the other. Both expr1 and expr2 are date or date-and-time expressions. Only the date parts of the values are used in the calculation. Example
  `SELECT DATEDIFF('1997-12-31','1997-12-30');`

Link to the YouTube Recording: https://www.youtube.com/watch?v=i_dsi1PhIl4

#90daysofanalytics #community #dataanalysis #dataanalyst #microsoft #msexcel #SQL

Day 43 - 90 days of Analytics: String functions

In today's second video, we looked at string functions in SQL queries

The following were mentioned

-The **LENGTH()** function returns the length of a string (in bytes). Syntax

```
LENGTH(string)
```

Example

```
SELECT FirstName, LENGTH(FirstName) AS Length_FirstName
FROM staff_db.staffdemographic;
```

-The **CONCAT()** function adds two or more expressions together. Syntax

```
CONCAT(expression1, expression2, expression3,...)
```

Example

```
SELECT FirstName, LastName , CONCAT(FirstName," ",LastName) AS FullName
FROM staff_db.staffdemographic;
```

-The **SUBSTRING()** function extracts a substring from a string (starting at any position). Syntax

```
SUBSTRING(string, start, length)
```

- **string** : Required. The string to extract from
- **start**: Required. The start position. Can be both a positive or negative number. If it is a positive number, this function extracts from the beginning of the string. If it is a negative number, this function extracts from the end of the string
- **length**: Optional. The number of characters to extract. If omitted, the whole string will be returned (from the start position)

Example

```
SELECT JobTitle, SUBSTRING(JobTitle,2,5) AS Substring_JobTitle
FROM staff_db.staffsalary;
```

-The **RIGHT()** function extracts a number of characters from a string (starting from right). Syntax

```
RIGHT(string, number_of_chars)
```

- **String** : Required. The string to extract from
- **number_of_chars** : Required. The number of characters to extract. If this parameter is larger than the number of characters in string, this function will return **string**

Example

```
SELECT JobTitle, RIGHT(JobTitle,2) AS LastTwoCharacters
FROM staff_db.staffsalary;
```

-The **LEFT()** function extracts a number of characters from a string (starting from left). Syntax

```
LEFT(string, number_of_chars)
```

- **String** : Required. The string to extract from
- **number_of_chars**: Required. The number of characters to extract. If this parameter is larger than the number of characters in string, this function will return **string**

Example

```
SELECT JobTitle, LEFT(JobTitle,2) AS FirstTwoCharacters
FROM staff_db.staffsalary;
```

-The **LOCATE()** function returns the position of the first occurrence of a substring in a string. If the substring is not found within the original string, this function returns 0. The function performs a **case-insensitive** search.

Syntax

```
LOCATE(substring, string, start)
```

- **Substring** : Required. The substring to search for in string
- **String** : Required. The string that will be searched
- **Start** : Optional. The starting position for the search. Position 1 is default

Example

```
SELECT LOCATE("days","90daysofanalytics") AS TextLocated;
```

-The **REPLACE()** function replaces all occurrences of a substring within a string, with a new substring. The function performs a case-sensitive replacement. Syntax

```
REPLACE(string, from_string, new_string)
```

- **string**: Required. The original string
- **from_string** : Required. The substring to be replaced
- **new_string** : Required. The new replacement substring

Example

```
SELECT REPLACE("90daysofanalytics","analytics","SQL") AS ReplacedText;
```

Link to the YouTube Recording: https://www.youtube.com/watch?v=s30Dmo2de9I

#90daysofanalytics #community #dataanalysis #dataanalyst #microsoft #msexcel #SQL