Relational DB & SQL - C11

Dashboard / Courses / Miscellaneous / RDB & SQL - C11 / Functions / Date and Time Functions

Date and Time Functions

To do: Go through the activity to the end

Date Functions

`GETDATE()`

You can use the `GETDATE()` function to determine the current date and time of the computer running the current SQL instance. This function doesn't include the time zone difference, just returns **datetime** format.

Let's see a `GETDATE()` function example.

query:

```sql
SELECT GETDATE() AS now;

```

result:

```
now
-----------------------
2021-11-13 10:38:23.363
```

`DATENAME(datepart, date)`

The `DATENAME()` function returns the name or value of a specific part of the date in **nvarchar** format.

Let's see a `DATENAME()` function example.

query:

```sql
SELECT DATENAME(WEEKDAY, '2021-11-11') AS sample;

```

result:

```
sample
--------
Thursday
```

💡**Tips:** There ara fourty datepart tips in SQL Server you can use. Such as: DAY, HOUR, MINUTE, WEEKDAY, YEAR, DAYOFYEAR, MONTH, etc.

`DATEPART(datepart, date)`

The `DATEPART()` function returns the value of a specific part of the date in **integer** format.

Let's see a `DATEPART()` function example.

query:

```
SELECT DATEPART(MINUTE, GETDATE()) AS sample;

```

result:

```
sample
------
35
```

💡**Tips:** There ara fourty datepart tips in SQL Server you can use. Such as: DAY, HOUR, MINUTE, WEEKDAY, YEAR, DAYOFYEAR, MONTH, etc.

`DAY(date)`

The `DAY()` function returns the day of the date in **integer** format.

Let's see a `DAY()` function example.

query:

```
SELECT DAY('2021-11-19') AS sample;

```

result:

```
sample
------
19
```

`MONTH(date)`

The `MONTH()` function returns the month of the date in **integer** format.

Let's see a `MONTH()` function example.

query:

```
SELECT MONTH('2021-11-19') AS sample;

```

result:

```
sample
------
11
```

`YEAR(date)`

The `YEAR()` function returns the year of the date in **integer** format.

Let's see a `YEAR()` function example.

query:

```
SELECT YEAR('2021-11-19') AS sample;

```

result:

```
sample
------
2021
```

`DATEDIFF(datepart, startdate, enddate)`

The `DATEDIFF()` function returns the difference between two dates in **integer** format.

In the syntax, datepart is the parameter that specifies which part of the date you want to use to calculate difference. The datepart can be year, month, week, day, hour, minute, second, or milisecond. You then specify the start date in the startdate parameter and the end date in the enddate parameter for which you want to find the difference.

Let's see a `DATEDIFF()` function example.

query:

```
SELECT DATEDIFF(week, '2021-01-01', '2021-02-12') AS DateDifference


```

result:

```
DateDifference
--------------
6

```

`DATEADD(datepart, number, date)`

The `DATEADD()` function enables you to add an interval to part of a specific date.

Let's see a `DATEADD()` function example.

query:

```
SELECT DATEADD (SECOND, 1, '2021-12-31 23:59:59') AS NewDate


```

result:

```
NewDate
-----------------------
2022-01-01 00:00:00.000

```

`EOMONTH(startdate [, month to add])`

The `EOMONTH()` function returns the last day of the month containing a specified date, with an optional offset.

Let's see a `EOMONTH()` function example.

query:

```
SELECT EOMONTH('2021-02-10') AS EndofFeb


```

result:

```
EndofFeb
----------
2021-02-28

```

`ISDATE(expression)`

The `ISDATE()` returns 1 if the expression is a valid datetime value; otherwise, 0.

If your system language is us_english, the date format is "mdy" (month, day, year) by default. The ISDATE() function checks the expression according to this format.
If you need, you can change the date format as below:

```
SET DATEFORMAT DMY
```

Let's see a `ISDATE()` function example. (dateformat = mdy)

query:

```
SELECT ISDATE('2021-02-10') AS isdate_


```

result:

```
isdate_
-------
1

```

Let's see another `ISDATE()` function example.

query:

```
SELECT ISDATE('15/2008/04') AS isdate_


```

result:

```
isdate_
-------
0

```