Functions that return system date and time

SQL SERVER FUNCTIONS FOR MANIPULATING DATA

Ana Voicu
Data Engineer

Common mistakes when working with dates and time

- Inconsistent date time formats or patterns
- Arithmetic operations
- Issues with time zones

Time zones in SQL Server

- Local time zone
- UTC time zone (Universal Time Coordinate)

Functions that return the date and time of the operating system

Higher-precision

- SYSDATETIME()
- SYSUTCDATETIME()
- SYSDATETIMEOFFSET()

timezone offset

Lower-precision

- GETDATE()
- GETUTCDATE()
- CURRENT_TIMESTAMP

Higher-precision functions example

```
SELECT
    SYSDATETIME() AS [SYSDATETIME],
    SYSDATETIMEOFFSET() AS [SYSDATETIMEOFFSET],
    SYSUTCDATETIME() AS [SYSUTCDATETIME];
```

```
| [SYSDATETIME]               | [SYSDATETIMEOFFSET]                | [SYSUTCDATETIME]            |
|-----------------------------|------------------------------------|-----------------------------|
| 2019-04-15 00:35:38.8740380 | 2019-04-15 00:35:38.8740380 +03:00 | 2019-04-14 21:35:38.8740380 |
```

Lower-precision functions example

```
SELECT
    CURRENT_TIMESTAMP AS [CURRENT_TIMESTAMP],
    GETDATE() AS [GETDATE] ,
    GETUTCDATE() AS [GETUTCDATE];
```

```
| [CURRENT_TIMESTAMP]     | [GETDATE]               | [GETUTCDATE]            |
|-------------------------|-------------------------|-------------------------|
| 2019-04-15 06:42:56.010 | 2019-04-15 06:42:56.010 | 2019-04-15 03:42:56.010 |
```

These functions are similar to the previous ones, but they return a datetime type instead of datetime2, so their precision is not that accurate.

Retrieving only the date

```
SELECT
    CONVERT(date, SYSDATETIME()) AS [SYSDATETIME],
    CONVERT(date, SYSDATETIMEOFFSET()) AS [SYSDATETIMEOFFSET],
    CONVERT(date, SYSUTCDATETIME()) AS [SYSUTCDATETIME],
    CONVERT(date, CURRENT_TIMESTAMP) AS [CURRENT_TIMESTAMP],
    CONVERT(date, GETDATE()) AS [GETDATE],
    CONVERT(date, GETUTCDATE()) AS [GETUTCDATE];
```

```
| [SYSDATETIME] | [SYSDATETIMEOFFSET] | [SYSUTCDATETIME] | [CURRENT_TIMESTAMP] | [GETDATE]   |[GETUT
|---------------|---------------------|------------------|---------------------|-------------|------
| 2018-11-22    | 2018-11-22          | 2018-11-22       | 2018-11-22          | 2018-11-22  | 2018-
```

Retrieving only the time

```
SELECT
    CONVERT(time, SYSDATETIME()) AS [SYSDATETIME],
    CONVERT(time, SYSDATETIMEOFFSET()) AS [SYSDATETIMEOFFSET],
    CONVERT(time, SYSUTCDATETIME()) AS [SYSUTCDATETIME],
    CONVERT(time, CURRENT_TIMESTAMP) AS [CURRENT_TIMESTAMP],
    CONVERT(time, GETDATE()) AS [GETDATE],
    CONVERT(time, GETUTCDATE()) AS [GETUTCDATE]
```

```
| [SYSDATETIME]    | [SYSDATETIMEOFFSET] | [SYSUTCDATETIME] | [CURRENT_TIMESTAMP] | [GETDATE]        | [GETUTCDATE]
|------------------|---------------------|------------------|---------------------|------------------|--------------
| 07:03:13.4127151 | 07:03:13.4142034    | 04:03:13.4142034 | 07:03:13.4133333    | 07:03:13.4133333 | 04:03:13.41333
```

Let's practice!

SQL SERVER FUNCTIONS FOR MANIPULATING DATA

Functions returning date and time parts

SQL SERVER FUNCTIONS FOR MANIPULATING DATA

Ana Voicu
Data Engineer

YEAR(date)

- Returns the year from the specified date

```
SELECT
    first_name,
    first_vote_date,
    YEAR(first_vote_date) AS first_vote_year
FROM voters;
```

```
| first_name | first_vote_date | first_vote_year |
|------------|-----------------|-----------------|
| Carol      | 2015-03-09      | 2015            |
| Ana        | 2015-01-17      | 2015            |
| Melissa    | 2015-04-09      | 2015            |
```

MONTH(date)

- Returns the month from the specified date

```
SELECT
    first_name,
    first_vote_date,
    YEAR(first_vote_date) AS first_vote_year,
    MONTH(first_vote_date) AS first_vote_month
FROM voters;
```

```
| first_name | first_vote_date | first_vote_year | first_vote_month |
|------------|-----------------|-----------------|------------------|
| Carol      | 2015-03-09      | 2015            | 3                |
| Ana        | 2015-01-17      | 2015            | 1                |
| Melissa    | 2015-04-09      | 2015            | 4                |
```

DAY(date)

- Returns the day from the specified date

```
SELECT
    first_name,
    first_vote_date,
    YEAR(first_vote_date) AS first_vote_year,
    MONTH(first_vote_date) AS first_vote_month,
    DAY(first_vote_date) AS first_vote_day
FROM voters;
```

```
| first_name | first_vote_date | first_vote_year | first_vote_month | first_vote_day |
|------------|-----------------|-----------------|------------------|----------------|
| Carol      | 2015-03-09      | 2015            | 3                | 9              |
| Ana        | 2015-01-17      | 2015            | 1                | 17             |
| Melissa    | 2015-04-09      | 2015            | 4                | 9              |
```

DATENAME(datepart, date)

- Returns a character string representing the specified date part of the given date

datepart	abbreviations
year	yy, yyyy
month	mm, m
dayofyear	dy, y
week	wk, ww
weekday	dw, w

DATENAME() example

```
DECLARE @date datetime = '2019-03-24'
SELECT
    YEAR(@date) AS year,
    DATENAME(YEAR, @date) AS year_name,
    MONTH(@date) AS month,
    DATENAME(MONTH, @date) AS month_name,
    DAY(@date) AS day,
    DATENAME(DAY, @date) AS day_name,
    DATENAME(WEEKDAY, @date) AS weekday
```

```
| year | year_name | month | month_name | day | day_name | weekday |
|------|-----------|-------|------------|-----|----------|---------|
| 2019 | 2019      | 3     | March      | 24  | 24       | Sunday  |
```

DATEPART(datepart, date)

- It is similar to `DATENAME()`
- Returns an integer representing the specified part of the given date

```
 DECLARE @date datetime = '2019-03-24'
SELECT
    DATEPART(YEAR, @date) AS year_name,
    DATENAME(YEAR, @date) AS year_name,
    DATEPART(MONTH, @date) AS month_name,
    DATENAME(MONTH, @date) AS month_name
```

```
| year | year_name | month | month_name |
|------|-----------|-------|------------|
| 2019 | 2019      | 3     | March      |
```

DATEFROMPARTS(year, month, day)

- Receives 3 parameters: year, month, and day values
- Generates a **date**

```
SELECT
    DATEFROMPARTS(2019, 3, 5) AS new_date;
```

```
| new_date   |
|------------|
| 2019-03-05 |
```

DATEFROMPARTS(year, month, day)

```
SELECT
    YEAR('2019-03-05')  AS date_year,
    MONTH('2019-03-05') AS date_month,
    DAY('2019-03-05')   AS date_day,
    DATEFROMPARTS(YEAR('2019-03-05'), MONTH('2019-03-05'), DAY('2019-03-05')) AS reconstructed_date;
```

a string of characters

```
| date_year | date_month | date_day | reconstructed_date |
|-----------|------------|----------|--------------------|
| 2019      | 3          | 5        | 2019-03-05         |
```

date

Let's practice!

SQL SERVER FUNCTIONS FOR MANIPULATING DATA

Performing arithmetic operations on dates

SQL SERVER FUNCTIONS FOR MANIPULATING DATA

Ana Voicu
Data Engineer

Types of operations with dates

- Operations using arithmetic operators (+, -)
- Modify the value of a date - `DATEADD()`
- Return the difference between two dates - `DATEDIFF()`

Arithmetic operations

```
DECLARE @date1 datetime = '2019-01-01';
DECLARE @date2 datetime = '2020-01-01';

SELECT
    @date2 + 1 AS add_one,
    @date2 - 1 AS subtract_one,
    @date2 + @date1 AS add_dates,
    @date2 - @date1 AS subtract_date;
```

In SQL Server, the date is first converted to an integer and then it is being added to the initial date as an increase in number of days.

```
| add_one                 | subtract_one            | add_dates               | subtract_date
|-------------------------|-------------------------|-------------------------|---------------
| 2020-01-02 00:00:00.000 | 2019-12-31 00:00:00.000 | 2139-01-01 00:00:00.000 | 1901-01-01 00:00:00.
```

DATEADD(datepart, number, date)

- Add or subtract a number of time units from a date

```
SELECT
    first_name,
    birthdate,
    DATEADD(YEAR, 5, birthdate) AS fifth_birthday,
    DATEADD(YEAR, -5, birthdate) AS subtract_5years,
    DATEADD(DAY, 30, birthdate) AS add_30days,
    DATEADD(DAY, -30, birthdate) AS subtract_30days
FROM voters;
```

```
| first_name | birthdate  | fifth_birthday | subtract_5years | add_30days | subtract_30days |
|------------|------------|----------------|-----------------|------------|-----------------|
| Carol      | 1989-01-15 | 1994-01-15     | 1984-01-15      | 1989-02-14 | 1988-12-16      |
| Dennis     | 1972-03-11 | 1977-03-11     | 1967-03-11      | 1972-04-10 | 1972-02-10      |
```

DATEDIFF(datepart, startdate, enddate)

- Returns the number of units between two dates

```
SELECT
    first_name,
    birthdate,
    first_vote_date,
    DATEDIFF(YEAR, birthdate, first_vote_date) AS age_years,
    DATEDIFF(QUARTER, birthdate, first_vote_date) AS age_quarters,
    DATEDIFF(DAY, birthdate, first_vote_date) AS age_days,
    DATEDIFF(HOUR, birthdate, first_vote_date) AS age_hours
  FROM voters;
```

```
| first_name | birthdate  | first_vote_date | age_years | age_quarters | age_days | age_hours |
|------------|------------|-----------------|-----------|--------------|----------|-----------|
| Carol      | 1989-01-15 | 2015-03-09      | 26        | 104          | 9549     | 229176    |
| Dennis     | 1972-03-11 | 2013-10-29      | 41        | 167          | 15207    | 364968    |
```

Let's practice!

SQL SERVER FUNCTIONS FOR MANIPULATING DATA

Validating if an expression is a date

SQL SERVER FUNCTIONS FOR MANIPULATING DATA

Ana Voicu
Data Engineer

ISDATE(expression)

- Determines whether an expression is a valid date data type

ISDATE() expression	Return type
date, time, datetime	1
datetime2	0
other type	0

ISDATE(expression)

```
DECLARE @date1 NVARCHAR(20) = '2019-05-05'
DECLARE @date2 NVARCHAR(20) = '2019-01-XX'
DECLARE @date3 CHAR(20) = '2019-05-05 12:45:59.9999999'    a datetime2 value
DECLARE @date4 CHAR(20) = '2019-05-05 12:45:59'

SELECT
    ISDATE(@date1) AS valid_date,
    ISDATE(@date2) AS invalid_date,
    ISDATE(@date3) AS valid_datetime2,
    ISDATE(@date4) AS valid_datetime;
```

```
| valid_date | invalid_date |valid_datetime2 | valid_datetime |
|------------|--------------|----------------|----------------|
| 1          | 0            | 0              | 1              |
```

SET DATEFORMAT

```
SET DATEFORMAT {format}
```

- Sets the order of the date parts for interpreting strings as dates
- Valid formats:
  - `mdy` , `dmy` , `ymd` , `ydm` , `myd` , `dym`

SET DATEFORMAT

```
DECLARE @date1 NVARCHAR(20) = '12-30-2019'
DECLARE @date2 NVARCHAR(20) = '30-12-2019'

SET DATEFORMAT dmy;
SELECT
    ISDATE(@date1) AS invalid_dmy,
    ISDATE(@date2) AS valid_dmy;
```

```
|invalid_dmy| valid_dmy |
|-----------|-----------|
| 0         | 1         |
```

SET LANGUAGE

```
SET LANGUAGE {language}
```

- Sets the language for the session
- Implicitly sets the setting of `SET DATEFORMAT`
- Valid languages: English, Italian, Spanish, etc.

SET LANGUAGE

```
SET LANGUAGE English;
SELECT
    ISDATE('12-30-2019') AS mdy,
    ISDATE('30-12-2019') AS dmy;
```

```
| mdy | dmy |
|-----|-----|
| 1   | 0   |
```

SET LANGUAGE

```
SET LANGUAGE French;
SELECT
    ISDATE('12-30-2019') AS mdy,
    ISDATE('30-12-2019') AS dmy;
```

```
| mdy | dmy |
|-----|-----|
| 0   | 1   |
```

Let's practice!

SQL SERVER FUNCTIONS FOR MANIPULATING DATA