**SQL Server Date and Time Functions**

SQL Server provides several types of date and time functions such as SYSUTCDATETIME(), CURRENT\_TIMESTAMP, GETDATE(), DAY(), MONTH(), YEAR(), DATEFROMPARTS (), DATETIME2FROMPARTS(), TIMEFROMPARTS (), DATEDIFF(), DATEADD(), ISDATE(), etc. These functions are used to perform operations on date and time input.

SYSUTCDATETIME(): This function is used to return the date and time values of the computer on which the instance of SQL Server is running in the UTC format.

Syntax: SYSUTCDATETIME()

Example: SELECT SYSUTCDATETIME() AS [SYSUTCDATETIME()]

CURRENT\_TIMESTAMP: This function is used to return the datetime value containing the date and time values of the computer on which the instance of SQL Server is running. The function returns datetime value as the result, excluding the time zone offset.

Syntax: CURRENT\_TIMESTAMP

Example: SELECT CURRENT\_TIMESTAMP AS [CURRENT\_TIMESTAMP]

GETDATE(): This function is used to return the datetime value containing the date and time values of the computer on which the instance of SQL Server is running. The function returns datetime value as the result, excluding the time zone offset. The difference between CURRENT\_TMESTAMP and GETDATE() is that CURRENT\_TIMESTAMP is an ANSI SQL function whereas GETDATE is the T-SQL version of that same function.

In most scenarios, CURRENT\_TIMESTAMP is recommended to use as it is portable to any ANSI-compliant database, whereas GETDATE() is not.

Syntax: GETDATE()

Example: SELECT GETDATE() AS [GETDATE()]

DAY(): This function is used to return an integer which represents the day (day of the month) part of the specified date. The DAY function takes *date*as an argument. The provided argument can be either of the types datetime, date, datetime2, time and small datetime. Users can pass a column expression, expression, string literal or user-defined variable as an argument to the DAY function.

Note: If the provided date argument only contains a time part, then the DAY function will return 1 – the base day as result.

Syntax: DAY(*date*)

Example 1: SELECT DAY('2022-09-02') AS [DAY]

Example 2: SELECT DAY('2022-09-12 01:01:01.2234567') AS [DAY]

MONTH(): This function is used to return an integer which represents the month part of the specified date. The MONTH function takes date as an argument. The provided argument can be either of the types datetimeoffset, time, date, datetime, smalldatetime or datetime2. Users can pass a column expression, expression, string literal or user-defined variable as an argument to the MONTH function.

Note: If the provided date argument only contains a time part, then the MONTH function will return 1, the base month.

Syntax: MONTH(date)

Example 1: SELECT MONTH ('2022-10-02') AS [MONTH]

Example 2: SELECT MONTH ('2022-01-12 01:01:01.2234567') AS [MONTH]

YEAR(): This function is used to return an integer which represents the year part of the specified date. The YEAR function takes date as an argument. The provided argument can be either of the types datetimeoffset, time, date, datetime, smalldatetime or datetime2. Users can pass a column expression, expression, string literal or user-defined variable as an argument to the YEAR function.

Note: If the provided date argument only contains a time part, then the YEAR function will return 1900, the base year.

Syntax: YEAR(date)

Example 1: SELECT YEAR ('1998-10-02') AS [YEAR]

Example 2: SELECT YEAR('2022-01-12 01:01:01.2234567') AS [YEAR]

DATEFROMPARTS(): This function is used to return date for the specified year, month, and day. The DATEFROMPARTS function takes *year*(specifies the year), *month* (specifies the month) and *day* (specifies the day) as the required arguments. If an invalid *year*, *month* or *day* is provided, then DATEFROMPARTS function will raise an error. Null is returned from DATEFROMPARTS function if at least one required argument has a null value.

Syntax: DATEFROMPARTS(year, month, day)

Example: SELECT DATEFROMPARTS (2012, 09, 15) AS Result;

DATETIME2FROMPARTS(): This function is used to return date for the specified date and time, with the specified precision. The DATETIME2FROMPARTS function takes *year*(specifies the year), *month* (specifies the month), *day* (specifies the day), *hour*(specifies the hours), *minute*(specifies the minutes), *seconds*(specifies the seconds), *fractions* (specifies the fractional seconds value) and *precision* (specifies the precision of the datetime2 value returned by the DATETIME2FROMPARTS function) as the required arguments. If an invalid required argument is provided, then DATETIME2FROMPARTS function will raise an error. Null is returned from DATETIME2FROMPARTS function if at least one required argument has a null value. However, if the *precision* argument has a null value, an error is raised by the DATETIME2FROMPARTS function.

Syntax: DATETIME2FROMPARTS (year, month, day, hour, minute, seconds, fractions, precision)

Example: SELECT DATETIME2FROMPARTS ( 2022, 09, 15, 12, 34, 56, 7,2 ) AS Result;

DATEDIFF(): This function is used to return the count of the specified datepart boundaries, crossed between two specified dates. The DATEDIFF function takes *datepart*(specifies the units in which DATEDIFF reports the difference between the *startdate*and *enddate*), *startdate* and *enddate* as arguments. The *startdate* and *enddate* can be either of the type datetime, date, datetime2, datetimeoffset, time and smalldatetime.

Syntax: DATEDIFF(datepart, startdate, endate)

Example: SELECT DATEDIFF(MONTH, '2010-08-01', '2022-02-28');