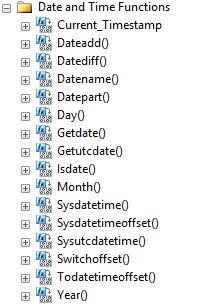
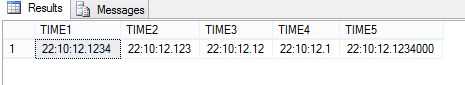
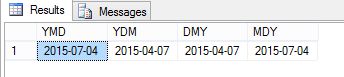
Date and Time functions of SQL Server. SQL Server provides the following functions for date and time.  
  
  
  
Before starting any discussion about the Date and Time functions we will get a basic understanding of the Date and Time data types.  
  
**TIME**  
The TIME data type defines a time of a day. The time is independent of the time zone and is based on a 24-hour clock.  
  
**Syntax**  
*time [ (fractional second precision) ]*  
**fractional seconds precision:**Specifies the number of digits for the fractional part of the seconds. This can be an integer from 0 to 7. The default fractional precision is 7 (100ns).  
**Default string literal format:** hh:mm:ss[.nnnnnnn]  
**Storage size:**5 bytes  
**Range:**00:00:00.0000000 through 23:59:59.9999999  
  
**Example**

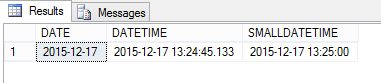
1. **DECLARE** @TIME1 **TIME**(4);
2. **DECLARE** @TIME2 **TIME**(3);
3. **DECLARE** @TIME3 **TIME**(2);
4. **DECLARE** @TIME4 **TIME**(1);
5. **DECLARE** @TIME5 **TIME**;
6. **SET** @TIME1='22:10:12.1234';
7. **SET** @TIME2=@TIME1;
8. **SET** @TIME3=@TIME1;
9. **SET** @TIME4=@TIME1;
10. **SET** @TIME5=@TIME1;
12. **SELECT** @TIME1 **AS** TIME1,@TIME2 **AS** TIME2,@TIME3 **AS** TIME3,@TIME4 **AS** TIME4 , @TIME5 **AS** TIME5;

**Output**  
  
  
  
**DATE**  
DATE defines a date in SQL Server.  
  
**Syntax***date*  
  
**Default string literal format:**YYYY-MM-DD  
**Range:**0001-01-01 through 9999-12-31  
**Default value:** 1900-01-01  
**Storage size:**3 bytes, fixed  
  
**Example**

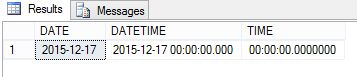
1. **DECLARE** @DATE1 **DATE**;
2. **DECLARE** @DATE2 **DATE**;
3. **DECLARE** @DATE3 **DATE**;
4. **DECLARE** @DATE4 **DATE**;
6. **SET** @DATE1='2015-7-04'; /\* YYYY-MM-DD; \*/
7. **SET** @DATE2='2015-04-7'; /\*YYYY-DD-MM\*/
8. **SET** @DATE3='04-7-2015'; /\*DD-MM-YYYY\*/
9. **SET** @DATE4='7-04-2015'; /\*MM-DD-YYYY\*/
10. **SELECT** @DATE1 **AS** YMD,@DATE2 **AS** YDM,@DATE3 **AS** DMY,@DATE4 **AS** MDY

**Output**  
  
  
  
**SMALLDATETIME**  
smalldatetime defines a date that is combined with a time of day. The time is based on a 24-hour day, with seconds always zero (:00) and without fractional seconds.  
  
**Syntax**  
  
*smalldatetime*  
**Date range:** 1900-01-01 through 2079-06-06  
**Time range:**00:00:00 through 23:59:59  
**Storage size:**4 bytes, fixed.  
  
**Example**

1. **DECLARE** @**DATE** **DATE**;
2. **DECLARE** @DATETIME [datetime];
3. **DECLARE** @SMALLDATETIME smalldatetime;
4. **SET** @DATETIME='2015-12-17 13:24:45.133';
5. **SET** @**DATE**=@DATETIME;
6. **SET** @SMALLDATETIME=@DATETIME;
8. **SELECT** @**DATE** **AS** [**DATE**] , @DATETIME **AS** [DATETIME], @SMALLDATETIME **AS** [SMALLDATETIME]

**Output**  
  
  
  
**DATETIME**  
Datetime defines a date that is combined with a time of day with fractional seconds that is based on a 24-hour clock.  
  
**Syntax**  
 *datetime*  
**Date range:**January 1, 1753, through December 31, 9999  
**Time range:** 00:00:00 through 23:59:59.997  
**Time zone offset range:**None  
**Storage size:** 8 bytes  
  
**Example**

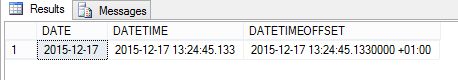
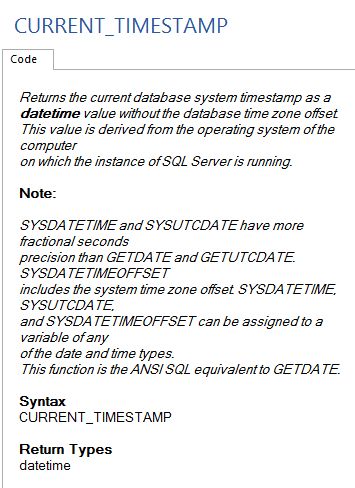
1. **DECLARE** @**DATE** **DATE**;
2. **DECLARE** @DATETIME [datetime];
3. **DECLARE** @**TIME** **TIME**;
4. **SET** @**DATE**='2015-12-17';
5. **SET** @DATETIME=@**DATE**;
6. **SET** @**TIME**=@DATETIME;
7. **SELECT** @**DATE** **AS** [**DATE**] , @DATETIME **AS** [DATETIME], @**TIME** **AS** [**TIME**]

**Output**  
  
  
  
**DATETIME2**  
Datetime2 defines a date that is combined with a time of day that is based on a 24-hour clock. datetime2 can be considered as an extension of the existing datetime type that has a larger date range, a larger default fractional precision and optional user-specified precision.  
  
**Syntax***datetime2 [ (fractional seconds precision) ]*  
  
**Default string literal format:**YYYY-MM-DD hh:mm:ss[.fractional seconds]  
**Date range:**0001-01-01 through 9999-12-31  
**Time range:** 00:00:00 through 23:59:59.9999999  
**Default value:**1900-01-01 00:00:00  
**Storage size:**6 bytes for precisions less than 3; 7 bytes for precisions 3 and 4. All other precisions require 8 bytes.  
  
**Example**

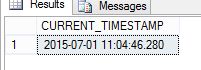
1. **DECLARE** @**DATE** **DATE**;
2. **DECLARE** @DATETIME2 [datetime2];
3. **DECLARE** @DATETIME [datetime];
4. **SET** @DATETIME2='2015-12-17 12:14:16.1234560';
5. **SET** @DATETIME=@DATETIME2;
6. **SET** @**DATE**=@DATETIME;
7. **SELECT** @**DATE** **AS** [**DATE**] , @DATETIME2 **AS** [DATETIME2], @DATETIME **AS** [DATETIME]

**Output**  
  
precisions   
  
**DATETIMEOFFSET**  
datetimeoffset defines a date that is combined with a time of a day that has time zone awareness and is based on a 24-hour clock.  
  
**Syntax**  
  
*datetimeoffset [ (fractional seconds precision) ]*  
  
**Default string literal formats (used for down-level client):** YYYY-MM-DD hh:mm:ss[.nnnnnnn] [{+|-}hh:mm]  
**Date range:** 0001-01-01 through 9999-12-31  
**Time range:**00:00:00 through 23:59:59.9999999  
**Time zone offset range:** -14:00 through +14:00  
**Storage size:**10 bytes  
  
**Example**

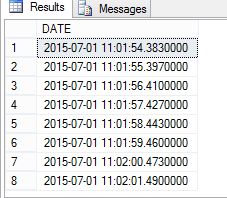
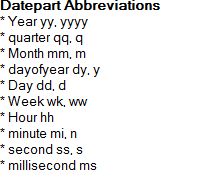
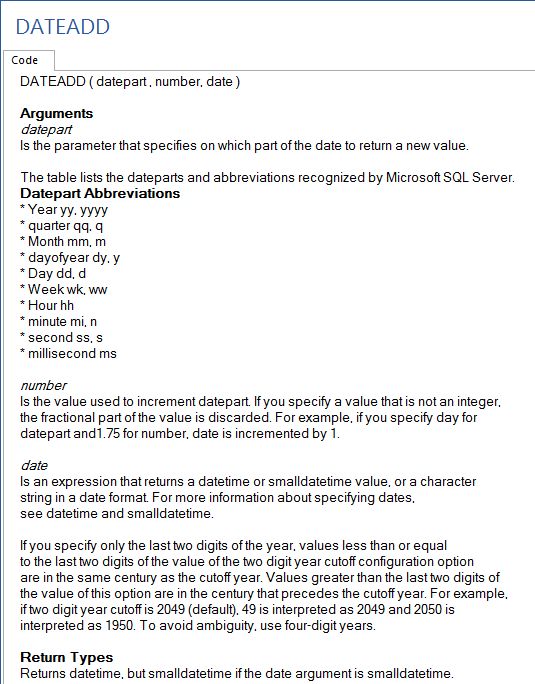
1. **DECLARE** @**DATE** **DATE**;
2. **DECLARE** @DATETIME [datetime];
3. **DECLARE** @DATETIMEOFFSET DATETIMEOFFSET;
4. **SET** @DATETIMEOFFSET='2015-12-17 13:24:45.133 +01:00';
5. **SET** @**DATE**=@DATETIMEOFFSET;
6. **SET** @DATETIME=@DATETIMEOFFSET;
8. **SELECT** @**DATE** **AS** [**DATE**] , @DATETIME **AS** [DATETIME], @DATETIMEOFFSET **AS** [DATETIMEOFFSET]

**Output**  
  
  
  
Now we read each Date and Time function one by one.  
  
**CURRENT\_TIMESTAMP**  
CURRENT\_TIMESTAMP returns the current database system timestamp as a datetime value without the database time zone offset. This value is derived from the operating system of the computer on which the instance of SQL Server is running. Transact-SQL statements can refer to CURRENT\_TIMESTAMP anywhere they can refer to a datetime expression. CURRENT\_TIMESTAMP is a nondeterministic function. Views and expressions that reference this column cannot be indexed.  
  
**Syntax**  
  
*CURRENT\_TIMESTAMP*  
  
**Return Type:** datetime  
  
  
  
**Example**

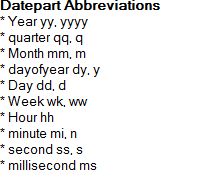
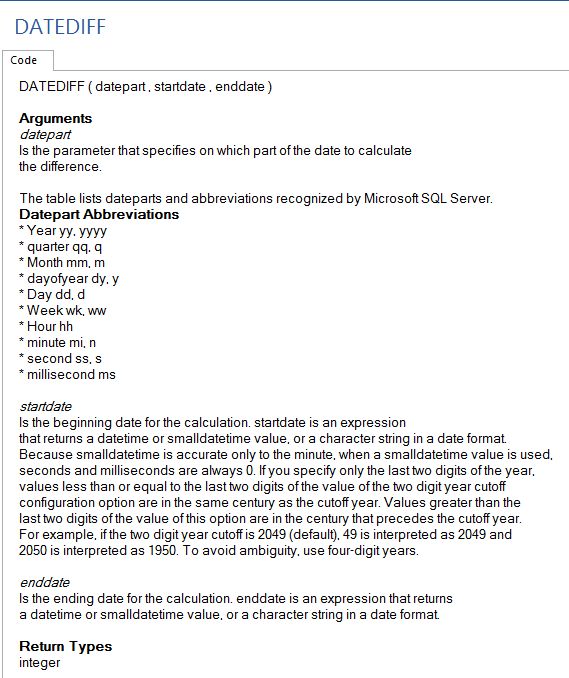
1. **SELECT** CURRENT\_TIMESTAMP **AS** [CURRENT\_TIMESTAMP];

**Output**  
  
**Example**

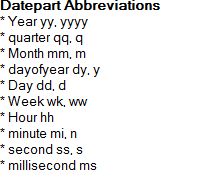
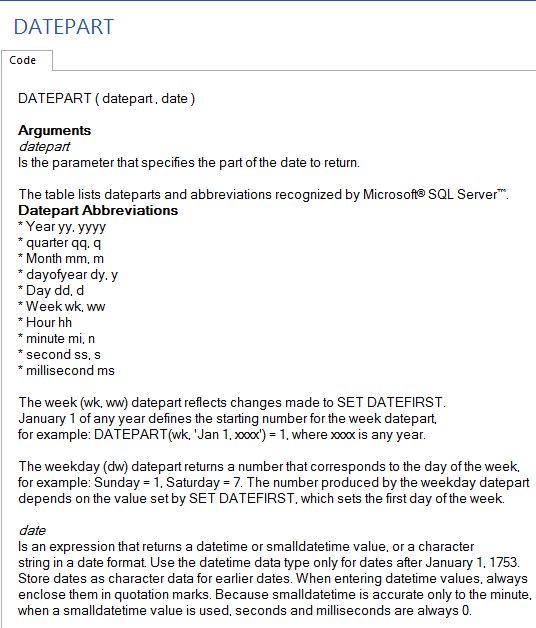
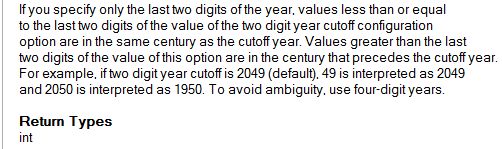
1. **CREATE** **TABLE** #**TEMP**
2. (
3. [**DATE**] [datetime2]
4. )
6. **DECLARE** @COUNT **INT**;
7. **SET** @COUNT=1;
8. WHILE @COUNT<=8
9. **BEGIN**
10. **INSERT** **INTO** #**TEMP**
11. **SELECT** CURRENT\_TIMESTAMP
12. WAITFOR DELAY '00:00:01';
13. **SET** @COUNT=@COUNT+1;
14. **END**
16. **SELECT** \* **FROM** #**TEMP** t
17. **DROP** **TABLE** #**TEMP**

**Output**   
  
Here we use “*WAITFOR DELAY '00:00:01*'” to provide a delay of one second after each insertion of data into the table.  
  
**DATEADD**  
The DATEADD function returns a specified date with the specified number interval (signed integer) added to a specified datepart of that date.  
  
**Syntax**  
*DATEADD (datepart , number , date )*  
  
  
  
**Number:**Is an expression that can be resolved to an int that is added to a datepart of date. User-defined variables are valid. If you specify a value with a decimal fraction, the fraction is truncated and not rounded.  
  
**Date:** Is an expression that can be resolved to a time, date, smalldatetime, datetime, datetime2, or datetimeoffset value.  
  
**Return Type:**The return data type is the data type of the date argument, except for string literals.   
  
  
  
**Example**

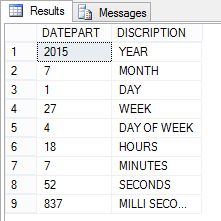
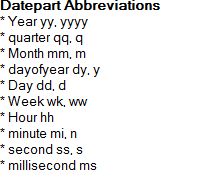
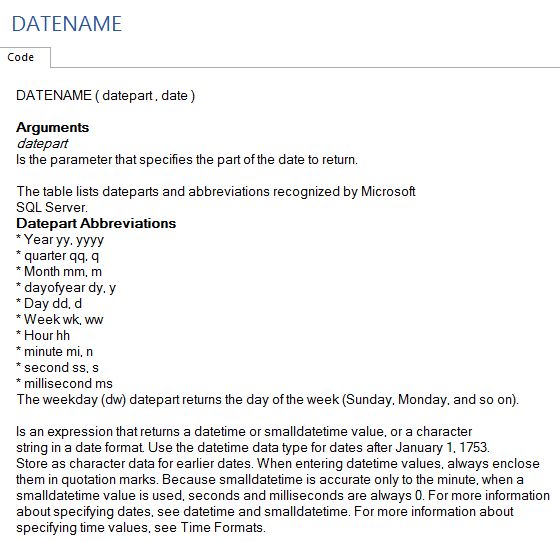
1. **DECLARE** @**DATE** [datetime2];
2. **SET** @**DATE**='2015-12-17 12:14:16.1234560';
4. **SELECT** DATEADD(YY,1,@**DATE**) **AS** [**DATE**], 'ADD YEAR' **AS** DISCRIPTION **UNION** ALL
5. **SELECT** DATEADD(MM,2,@**DATE**), 'ADD MONTH' **UNION** ALL
6. **SELECT** DATEADD(DD,10,@**DATE**) , 'ADD DAY' **UNION** ALL
7. **SELECT** DATEADD(WK,2,@**DATE**) , 'ADD WEEK' **UNION** ALL
8. **SELECT** DATEADD(DW,2,@**DATE**) , 'ADD WEEK DAY' **UNION** ALL
9. **SELECT** DATEADD(HH,2,@**DATE**), 'ADD HOUR' **UNION** ALL
10. **SELECT** DATEADD(MM,10,@**DATE**) , 'ADD MINUTE' **UNION** ALL
11. **SELECT** DATEADD(SS,11,@**DATE**) , 'ADD SECOND' **UNION** ALL
12. **SELECT** DATEADD(MS,100,@**DATE**), 'ADD MILLI SECOND'

**Output**  
  
  
  
**DATEDIFF**  
The DATEDIFF function returns the count (signed integer) of the specified datepart boundaries crossed between the specified start date and end date.  
 **Syntax**  
  
*DATEDIFF ( datepart , startdate , enddate )*  
  
  
 **Start date:** Is an expression that can be resolved to a time, date, smalldatetime, datetime, datetime2, or datetimeoffset value. date can be an expression, column expression, user-defined variable or string literal. startdate is subtracted from enddate.  
  
**Enddate:**It is the ending date of the calculation.  
  
**Return Type:**int  
  
  
 **Example**

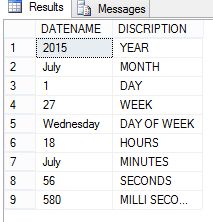
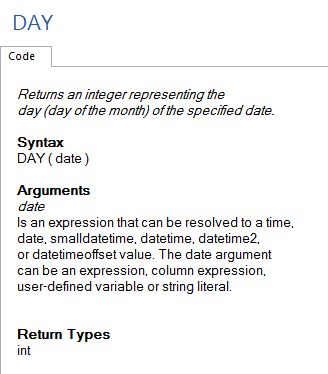
1. **DECLARE** @DATE1 [datetime2];
2. **DECLARE** @DATE2 [datetime2];
3. **SET** @DATE1='2015-12-17 12:14:16.323456789';
4. **SET** @DATE2='2013-08-13 8:10:26.223456789';
6. **SELECT** DATEDIFF(YY,@DATE2,@DATE1) **AS**[DATE\_DIFF], 'YEAR DIFF' **AS** DISCRIPTION **UNION** ALL
7. **SELECT** DATEDIFF(MM,@DATE2,@DATE1) **AS**[DATE\_DIFF], 'MONTH DIFF' **AS** DISCRIPTION **UNION** ALL
8. **SELECT** DATEDIFF(DD,@DATE2,@DATE1) **AS**[DATE\_DIFF], 'DAY DIFF' **AS** DISCRIPTION **UNION** ALL
9. **SELECT** DATEDIFF(WK,@DATE2,@DATE1) **AS**[DATE\_DIFF], 'WEEK DIFF' **AS** DISCRIPTION **UNION** ALL
10. **SELECT** DATEDIFF(DW,@DATE2,@DATE1) **AS**[DATE\_DIFF], 'WEAKDAY DIFF' **AS** DISCRIPTION **UNION** ALL
11. **SELECT** DATEDIFF(HH,@DATE2,@DATE1) **AS**[DATE\_DIFF], 'HOURS DIFF' **AS** DISCRIPTION **UNION** ALL
12. **SELECT** DATEDIFF(MM,@DATE2,@DATE1) **AS**[DATE\_DIFF], 'MINUTE DIFF' **AS** DISCRIPTION **UNION** ALL
13. **SELECT** DATEDIFF(SS,@DATE2,@DATE1) **AS**[DATE\_DIFF], 'SECONDS DIFF' **AS** DISCRIPTION

**Output**  
  
  
 **DATEPART**  
The DATEPART function returns an integer that represents the specified datepart of the specified date. Each datepart and its abbreviations return the same value. DATEPART can be used in the select list, WHERE, HAVING, GROUP BY and ORDER BY clauses.  
  
**Syntax**  
*DATEPART ( datepart , date )*  
  
  
 **Date:**an expression that can be resolved to a time, date, smalldatetime, datetime, datetime2, or datetimeoffset value. date can be an expression, column expression, user-defined variable, or string literal.  
  
**Return Type:**int   
  
  
  
  
**Example**

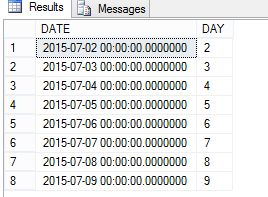
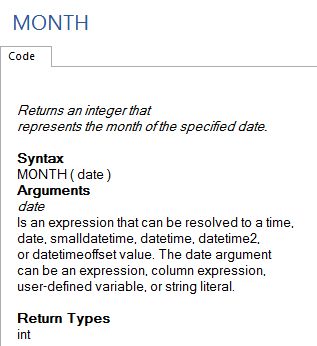
1. **DECLARE** @**DATE** [datetime2];
2. **SET** @**DATE**=GETDATE();
4. **SELECT** DATEPART(YY,@**DATE**) **AS** [DATEPART], 'YEAR' **AS** DISCRIPTION **UNION** ALL
5. **SELECT** DATEPART(MM,@**DATE**) ,'MONTH'**UNION** ALL
6. **SELECT** DATEPART(DD,@**DATE**),'DAY' **UNION** ALL
7. **SELECT** DATEPART(WW,@**DATE**) , 'WEEK'**UNION** ALL
8. **SELECT** DATEPART(DW,@**DATE**) , 'DAY OF WEEK '**UNION** ALL
9. **SELECT** DATEPART(HH,@**DATE**),'HOURS' **UNION** ALL
10. **SELECT** DATEPART(MM,@**DATE**),'MINUTES' **UNION** ALL
11. **SELECT** DATEPART(SS,@**DATE**),'SECONDS' **UNION** ALL
12. **SELECT** DATEPART(MS,@**DATE**) ,'MILLI SECONDS'

**Output**  
  
  
  
**DATENAME**  
The DATENAME function returns a character string that represents the specified datepart of the specified date. Each datepart and its abbreviations return the same value. DATENAME can be used in the select list, WHERE, HAVING, GROUP BY and ORDER BY clauses.  
 **Syntax**  
*DATENAME ( datepart , date )*  
  
**Date:** Is an expression that can be resolved to a time, date, smalldatetime, datetime, datetime2, or datetimeoffset value. date can be an expression, column expression, user-defined variable, or string literal.  
  
**Return Type:** nvarchar  
  
  
  
 **Example**

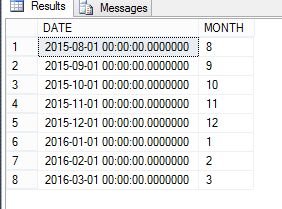
1. **DECLARE** @**DATE** [datetime2];
2. **SET** @**DATE**=GETDATE();
4. **SELECT** DATENAME(YY,@**DATE**) **AS** [DATENAME], 'YEAR' **AS** DISCRIPTION **UNION** ALL
5. **SELECT** DATENAME(MM,@**DATE**) ,'MONTH'**UNION** ALL
6. **SELECT** DATENAME(DD,@**DATE**),'DAY' **UNION** ALL
7. **SELECT** DATENAME(WW,@**DATE**) , 'WEEK'**UNION** ALL
8. **SELECT** DATENAME(DW,@**DATE**) , 'DAY OF WEEK '**UNION** ALL
9. **SELECT** DATENAME(HH,@**DATE**),'HOURS' **UNION** ALL
10. **SELECT** DATENAME(MM,@**DATE**),'MINUTES' **UNION** ALL
11. **SELECT** DATENAME(SS,@**DATE**),'SECONDS' **UNION** ALL
12. **SELECT** DATENAME(MS,@**DATE**) ,'MILLI SECONDS'

**Output**  
  
  
  
**DAY**  
The DAY function returns an integer representing the day (day of the month) of the specified date. DAY returns the same value as the DATEPART (day, date). If date contains only a time part, the return value is 1, the base day.  
 **Syntax:**  
*DAY ( date )*  
  
**Return Type:** int  
  
  
  
**Example**

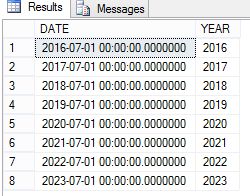
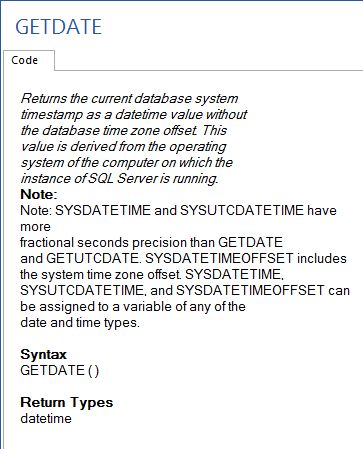
1. **CREATE** **TABLE** #**TEMP**
2. (
3. [**DATE**] [datetime2],
4. [DAY] **INT**
5. )
7. **DECLARE** @COUNT **INT**;
8. **DECLARE** @**DATE** [**date**];
9. **SET** @**DATE**=GETDATE()
10. **SET** @COUNT=1;
11. WHILE @COUNT<=8
12. **BEGIN**
13. **INSERT** **INTO** #**TEMP**
14. **SELECT** DATEADD(DD,@COUNT,@**DATE**) , DAY(DATEADD(DD,@COUNT,@**DATE**))
16. **SET** @COUNT=@COUNT+1;
17. **END**
19. **SELECT** \* **FROM** #**TEMP** t
20. **DROP** **TABLE** #**TEMP**

**Output**  
  
  
  
**MONTH**  
The MONTH function returns an integer that represents the month of the specified date. MONTH returns the same value as DATEPART(month, date). If date contains only a time part, the return value is 1, the base month.  
  
**Syntax**  
*MONTH ( date )*  
  
**Return Type:**int  
  
  
  
**Example**

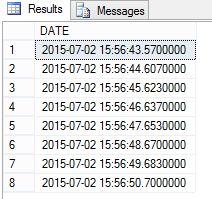
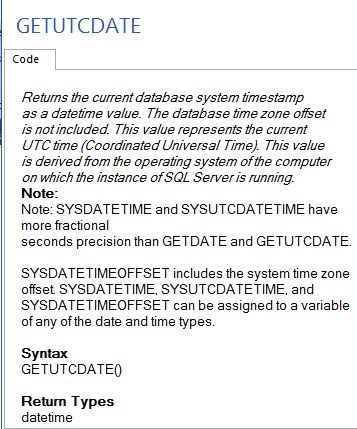
1. **CREATE** **TABLE** #**TEMP**
2. (
3. [**DATE**] [datetime2],
4. [MONTH] **INT**
5. )
7. **DECLARE** @COUNT **INT**;
8. **DECLARE** @**DATE** [**date**];
9. **SET** @**DATE**=GETDATE()
10. **SET** @COUNT=1;
11. WHILE @COUNT<=8
12. **BEGIN**
13. **INSERT** **INTO** #**TEMP**
14. **SELECT** DATEADD(MM,@COUNT,@**DATE**) , MONTH(DATEADD(MM,@COUNT,@**DATE**))
16. **SET** @COUNT=@COUNT+1;
17. **END**
19. **SELECT** \* **FROM** #**TEMP** t
20. **DROP** **TABLE** #**TEMP**

**Output**  
  
  
  
**YEAR**  
The YEAR function returns an integer that represents the year of the specified date. YEAR returns the same value as DATEPART (year, date). If date only contains a time part, the return value is 1900, the base year.  
  
**Syntax**  
*YEAR ( date )*  
  
**Return type:**int  
  
**Example**

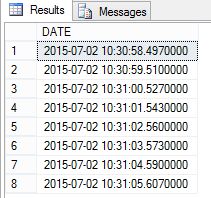
1. **CREATE** **TABLE** #**TEMP**
2. (
3. [**DATE**] [datetime2],
4. [YEAR] **INT**
5. )
7. **DECLARE** @COUNT **INT**;
8. **DECLARE** @**DATE** [**date**];
9. **SET** @**DATE**=GETDATE()
10. **SET** @COUNT=1;
11. WHILE @COUNT<=8
12. **BEGIN**
13. **INSERT** **INTO** #**TEMP**
14. **SELECT** DATEADD(YY,@COUNT,@**DATE**) , YEAR(DATEADD(YY,@COUNT,@**DATE**))
16. **SET** @COUNT=@COUNT+1;
17. **END**
19. **SELECT** \* **FROM** #**TEMP** t
20. **DROP** **TABLE** #**TEMP**

**Output**  
  
  
  
**GETDATE**  
The GETDATE function returns the current database system timestamp as a datetime value without the database time zone offset. This value is derived from the operating system of the computer on which the instance of SQL Server is running. Transact-SQL statements can refer to GETDATE anywhere they can refer to a datetime expression. GETDATE is a nondeterministic function. Views and expressions that reference this function in a column cannot be indexed.  
  
**Syntax**  
*GETDATE ( )*  
  
**Return Type:** datetime  
  
  
  
**Example**

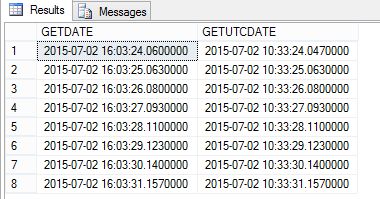
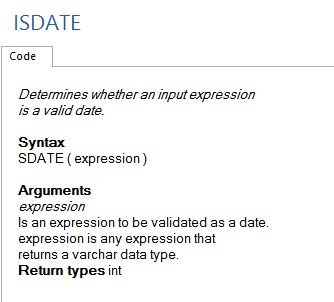
1. **CREATE** **TABLE** #**TEMP**
2. (
3. [**DATE**] [datetime2]
4. )
6. **DECLARE** @COUNT **INT**;
7. **SET** @COUNT=1;
8. WHILE @COUNT<=8
9. **BEGIN**
10. **INSERT** **INTO** #**TEMP**
11. **SELECT** GETDATE()
12. WAITFOR DELAY '00:00:01';
13. **SET** @COUNT=@COUNT+1;
14. **END**
16. **SELECT** \* **FROM** #**TEMP** t
17. **DROP** **TABLE** #**TEMP**

**Output**  
  
  
  
**GETUTCDATE**  
The GETUTCDATE function returns the current database system timestamp as a datetime value. The database time zone offset is not included. This value represents the current Coordinated Universal (UTC) time. This value is derived from the operating system of the computer on which the instance of SQL Server is running. Transact-SQL statements can refer to GETUTCDATE anywhere they can refer to a datetime expression. GETUTCDATE is a nondeterministic function. Views and expressions that reference this function in a column cannot be indexed.  
  
**Syntax**  
*GETUTCDATE()*  
  
**Return Type:**datetime  
  
  
  
**Example**

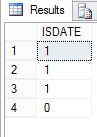
1. **CREATE** **TABLE** #**TEMP**
2. (
3. [**DATE**] [datetime2]
4. )
6. **DECLARE** @COUNT **INT**;
7. **SET** @COUNT=1;
8. WHILE @COUNT<=8
9. **BEGIN**
10. **INSERT** **INTO** #**TEMP**
11. **SELECT** GETUTCDATE()
12. WAITFOR DELAY '00:00:01';
13. **SET** @COUNT=@COUNT+1;
14. **END**
16. **SELECT** \* **FROM** #**TEMP** t
17. **DROP** **TABLE** #**TEMP**

**Output**  
  
   
  
**Example**

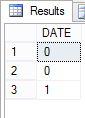
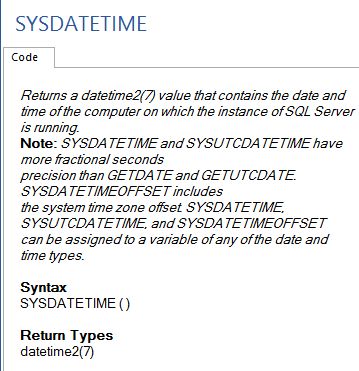
1. **CREATE** **TABLE** #**TEMP**
2. (
3. [GETDATE] [datetime2],
4. [GETUTCDATE] [datetime2]
6. )
8. **DECLARE** @COUNT **INT**;
9. **SET** @COUNT=1;
10. WHILE @COUNT<=8
11. **BEGIN**
12. **INSERT** **INTO** #**TEMP**
13. **SELECT** GETDATE() , GETUTCDATE()
14. WAITFOR DELAY '00:00:01';
15. **SET** @COUNT=@COUNT+1;
16. **END**
18. **SELECT** \* **FROM** #**TEMP** t
19. **DROP** **TABLE** #**TEMP**

**Output**  
  
**ISDATE**  
The ISDATE function returns 1 if the expression is a valid date, time, or datetime value, otherwise 0. ISDATE returns 0 if the expression is a datetime2 value. ISDATE is deterministic only if you use it with the CONVERT function, if the CONVERT style parameter is specified and style is not equal to 0, 100, 9, or 109. The return value of ISDATE depends on the settings set by SET DATEFORMAT, SET LANGUAGE and the default language option.  
  
**Syntax***ISDATE ( expression )*  
  
**Return Type:** int  
  
  
  
**Example**

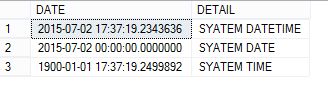
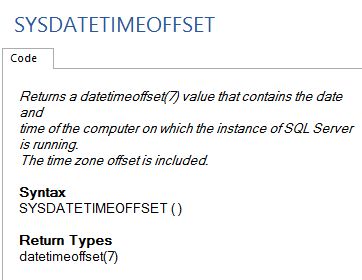
1. **SET** LANGUAGE us\_english;
2. **SET** DATEFORMAT mdy;
4. **SELECT** ISDATE('03/18/2010') **AS** [ISDATE] **UNION** ALL
5. **SELECT** ISDATE('03-15-2010') **UNION** ALL
6. **SELECT** ISDATE('03.15.2010') **UNION** ALL
7. **SELECT** ISDATE('24/2010/04')

**Output**  
  
  
  
**Example**

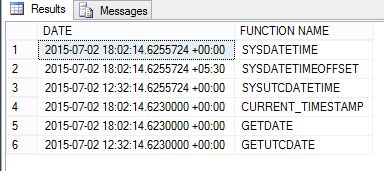
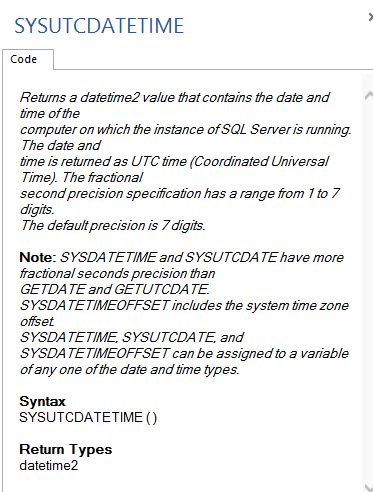
1. **SET** DATEFORMAT mdy;
2. **SELECT** ISDATE('15/04/2008') **AS** [**DATE**] **UNION** ALL
3. **SELECT** ISDATE('2008/15/04') **UNION** ALL
4. **SELECT** ISDATE('12/2004/15')

**Output**  
  
  
  
**SYSDATETIME**  
The SYSDATETIME function returns a datetime2(7) value that contains the date and time of the computer on which the instance of SQL Server is running. Transact-SQL statements can refer to SYSDATETIME anywhere they can refer to a datetime2(7) expression. SYSDATETIME is a nondeterministic function. Views and expressions that reference this function in a column cannot be indexed.  
  
**Syntax**  
*SYSDATETIME ( )*  
  
**Return Type:** datetime2(7)  
  
  
  
**Example**

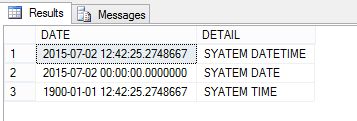
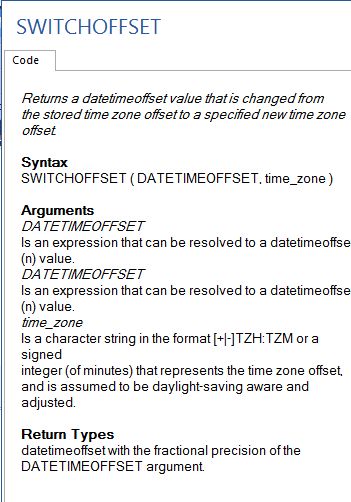
1. **SELECT** SYSDATETIME() **AS** [**DATE**] , 'SYATEM DATETIME' **AS** [DETAIL] **UNION** ALL
2. **SELECT** CONVERT (**DATE**,SYSDATETIME()) **AS** [**DATE**] , 'SYATEM DATE' **AS** [DETAIL] **UNION** ALL
3. **SELECT** CONVERT (**TIME**,SYSDATETIME()) **AS** [**DATE**] , 'SYATEM TIME' **AS** [DETAIL]

**Output**  
  
  
  
**SYSDATETIMEOFFSET**  
The SYSDATETIMEOFFSET function returns a datetimeoffset(7) value that contains the date and time of the computer on which the instance of SQL Server is running. The timezone offset is included. Transact-SQL statements can refer to SYSDATETIMEOFFSET anywhere they can refer to a datetimeoffset expression. SYSDATETIMEOFFSET is a nondeterministic function. Views and expressions that reference this function in a column cannot be indexed.  
  
**Syntax**  
*SYSDATETIMEOFFSET ( )*  
  
**Return Type:** datetimeoffset(7)  
  
  
  
**Example**

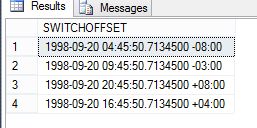
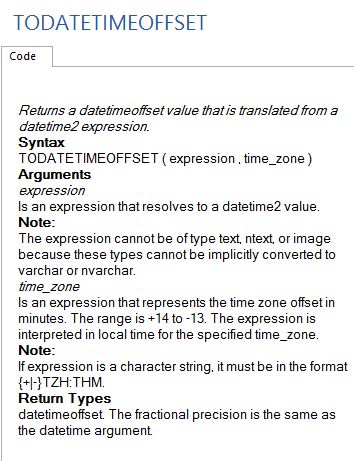
1. **SELECT** SYSDATETIME() **AS** [**DATE**], 'SYSDATETIME' **AS** [**FUNCTION** **NAME** ] **UNION** ALL
2. **SELECT** SYSDATETIMEOFFSET() **AS** [**DATE**], 'SYSDATETIMEOFFSET' **AS** [**FUNCTION** **NAME** ] **UNION** ALL
3. **SELECT** SYSUTCDATETIME() **AS** [**DATE**], 'SYSUTCDATETIME' **AS** [**FUNCTION** **NAME** ] **UNION** ALL
4. **SELECT** CURRENT\_TIMESTAMP **AS** [**DATE**], 'CURRENT\_TIMESTAMP' **AS** [**FUNCTION** **NAME** ] **UNION** ALL
5. **SELECT** GETDATE() **AS** [**DATE**], 'GETDATE' **AS** [**FUNCTION** **NAME** ] **UNION** ALL
6. **SELECT** GETUTCDATE() **AS** [**DATE**], 'GETUTCDATE' **AS** [**FUNCTION** **NAME** ]

**Output**  
  
  
  
**SYSUTCDATETIME**  
The SYSUTCDATETIME function returns a datetime2 value that contains the date and time of the computer on which the instance of SQL Server is running. The date and time is returned as the Coordinated Universal (UTC) time. The fractional second precision specification has a range from 1 to 7 digits. The default precision is 7 digits. Transact-SQL statements can refer to SYSUTCDATETIME anywhere they can refer to a datetime2 expression. SYSUTCDATETIME is a nondeterministic function. Views and expressions that reference this function in a column cannot be indexed.  
  
**Syntax**  
*SYSUTCDATETIME ( )*  
  
**Return Type:**datetime2  
  
  
  
**Example**

1. **SELECT** SYSUTCDATETIME() **AS** [**DATE**] , 'SYATEM DATETIME' **AS** [DETAIL] **UNION** ALL
2. **SELECT** CONVERT (**DATE**,SYSUTCDATETIME()) **AS** [**DATE**] , 'SYATEM DATE' **AS** [DETAIL] **UNION** ALL
3. **SELECT** CONVERT (**TIME**,SYSUTCDATETIME()) **AS** [**DATE**] , 'SYATEM TIME' **AS** [DETAIL]

**Output  
  
**  
  
**SWITCHOFFSET**  
The SWITCHOFFSET function returns a datetimeoffset value that is changed from the stored time zone offset to a specified new time zone offset.  
  
**Syntax**  
*SWITCHOFFSET ( DATETIMEOFFSET, time\_zone )*  
  
**DATETIMEOFFSET:**Is an expression that can be resolved to a datetimeoffset(n) value.  
  
**time\_zone:**Is a character string in the format [+|-]TZH:TZM or a signed integer (of minutes) that represents the time zone offset and is assumed to be daylight-saving aware and adjusted.  
  
**Return Type:** datetimeoffset with the fractional precision of the DATETIMEOFFSET argument.  
  
  
 **Example**

1. **CREATE** **TABLE** #**TEMP**
2. (
3. ColDatetimeoffset datetimeoffset
4. );
5. GO
6. **INSERT** **INTO** #**TEMP**
7. **VALUES** ('1998-09-20 7:45:50.71345 -5:00');
9. **SELECT** SWITCHOFFSET (ColDatetimeoffset, '-08:00') **AS** [SWITCHOFFSET] **FROM** #**TEMP** **UNION** ALL
10. **SELECT** SWITCHOFFSET (ColDatetimeoffset, '-03:00') **FROM** #**TEMP** **UNION** ALL
11. **SELECT** SWITCHOFFSET (ColDatetimeoffset, '+08:00') **FROM** #**TEMP** **UNION** ALL
12. **SELECT** SWITCHOFFSET (ColDatetimeoffset, '+04:00') **FROM** #**TEMP**
14. **DROP** **TABLE** #**TEMP**

**Output**  
  
  
  
**TODATETIMEOFFSET**  
  
The TODATETIMEOFFSET function returns a datetimeoffset value that is translated from a datetime2 expression.  
  
**Syntax** *TODATETIMEOFFSET ( expression , time\_zone )*  
  
**Return Type:**datetimeoffset. The fractional precision is the same as the datetime argument.  
  
  
  
**Example**

1. **CREATE** **TABLE** #**TEMP**
2. (
3. ColDatetimeoffset [datetime2]
4. );
5. GO
6. **INSERT** **INTO** #**TEMP**
7. **VALUES** ('1998-09-20 7:45:50.71345');
9. **SELECT** TODATETIMEOFFSET (ColDatetimeoffset, '-08:00') **AS** TODATETIMEOFFSET **FROM** #**TEMP** **UNION** ALL
10. **SELECT** TODATETIMEOFFSET (ColDatetimeoffset, '-03:00') **FROM** #**TEMP** **UNION** ALL
11. **SELECT** TODATETIMEOFFSET (ColDatetimeoffset, '+08:00') **FROM** #**TEMP** **UNION** ALL
12. **SELECT** TODATETIMEOFFSET (ColDatetimeoffset, '+04:00') **FROM** #**TEMP**
14. **DROP** **TABLE** #**TEMP**

**Output**  
  
