|  |  |
| --- | --- |
| **Function Category** | **Description** |
| [**Conversion Functions**](https://msdn.microsoft.com/en-us/library/hh231076.aspx) | Support data type casting and converting.  CAST ( expression AS data\_type [ ( length ) ] ) |
| [**Date and Time Functions**](https://msdn.microsoft.com/en-us/library/ms186724.aspx)  *Perform operations on a date and time input values and return string, numeric, or date and time values* | |  |  |  | | --- | --- | --- | | GETDATE | GETDATE() | Returns a datetime value that contains date and time of the underlying OS. | | DAY,  [MONTH](https://msdn.microsoft.com/en-us/library/ms187813.aspx),  [YEAR](https://msdn.microsoft.com/en-us/library/ms186313.aspx) | DAY (date ),  MONTH (date ),  YEAR (date ) | Returns an integer that represents the day, month, or year part of a specified date. | | DAYNAME | DAYNAME(*datepart, date*) | Returns name of the specified part of a specific date. | | [DATEDIFF](https://msdn.microsoft.com/en-us/library/ms189794.aspx) | DATEDIFF (datepart ,startdate ,enddate ) | Returns the number of date or time datepart boundaries that are crossed between two specified dates. | |
| **Logical Functions**  *Perform logical operations* | |  |  |  | | --- | --- | --- | | [IIF](https://msdn.microsoft.com/en-us/library/ms189794.aspx) | IIF ( boolean\_expression, true\_value, false\_value ) | Returns one of two values, depending on whether the Boolean expression evaluates to true or false.  Returns the data type with the highest precedence from the types in true\_value and false\_value. | |
| **String Functions**  *Perform operations on a string (****char*** *or* ***varchar****) input value and return a string or numeric value.* | |  |  |  | | --- | --- | --- | | LEFT  RIGHT  SUBSTRING | LEFT(*expr, n*) RIGHT(*expr, n*)  SUBSTRING(expr,   start, length) | Returns a part of a character string as specified:   * LEFT/RIGHT: left or right n characters; or * SUBSTRING: length characters starting from the start point. | | Str | STR(*float\_expr*) | Returns a character expression of the numeric data. | | LOWER  UPPER | LOWER(*string\_expr*) UPPER(*string\_expr*) | Returns a character expression with letters in the specified case. | | LEN | LEN(*string\_expr*) | Returns the number of characters of the specified string expression. | |

/\* Q8.1 \*/

SELECT **SYSDATETIME()** AS [System Date & Time],

**GETDATE()** AS [Current Date & Time]

What is the diff: GETDATE() vs. SYSDATETIME()?

/\* Q8.2 \*/

SELECT **YEAR(SYSDATETIME())** AS Year,

**DATENAME(MONTH, GETDATE())** AS Month,

**DAY(GETDATE())** AS Day,

**DATENAME(WEEKDAY, GETDATE())** AS Weekday

What other datepart can be used?

/\* Q8.3a \*/

SELECT **Year(**InvoiceDate**)** AS Year,

**Month(**InvoiceDate**)** AS Month

FROM Invoices

/\* Q8.3b \*/

SELECT DATENAME(Month, InvoiceDate) AS Month,

COUNT(\*) [Number of Invoices]

FROM Invoices

WHERE YEAR(InvoiceDate) = 2010

GROUP BY DATENAME(Month, InvoiceDate)

order by 2 DESC

/\* Q8.3c \*/

SELECT Month(InvoiceDate) AS [ ],

DATENAME(Month, InvoiceDate) AS Month,

COUNT(\*) [Number of Invoices]

FROM Invoices

GROUP BY Month(InvoiceDate), DATENAME(Month, InvoiceDate)

ORDER BY 1

/\* Q8.4 \*/

SELECT InvoiceID,

**DateDiff(day, InvoiceDate, PaymentDate)**

AS [Payment Elapse]

FROM Invoices

ORDER BY 2 DESC

What if …

* Swap the two dates?
* Use year/month as datepart?

/\* Q8.5 \*/

SELECT InvoiceID,

DateDiff(day, InvoiceDate, PaymentDate)

AS [Payment Elapse],

DueDays

**FROM Invoices i JOIN Terms t ON i.TermsID = t.TermsID**

**WHERE DateDiff(DAY, InvoiceDate, PaymentDate) > DueDays + 5**

ORDER BY 2 DESC

Try it again with all six clauses used…

SELECT **IIF(PaymentDate IS NULL, 'Unpaid', 'Paid')**

AS [Payment Status]

FROM Invoices

/\* Q8.6 – w/ error! \*/

SELECT InvoiceID, IIF(PaymentDate IS NULL, 'Unpaid',

DateDiff(day, InvoiceDate, PaymentDate) )

FROM Invoices

WHERE YEAR(InvoiceDate) = 2010

Not work… Why?

/\* Q8.6 \*/

SELECT InvoiceID, IIF(PaymentDate IS NULL, 'Unpaid',

**CAST(**DateDiff(day, InvoiceDate, PaymentDate)

**AS char(4))**)

AS [Payment Elapse]

FROM Invoices

WHERE YEAR(InvoiceDate) = 2010

/\* Q8.8 \*/

SELECT Name, **LEN(Name)** [# chars]

FROM Vendors

/\* Q8.9 \*/

SELECT Name, **LEFT(Name, 1)** L1,

**SUBSTRING(Name, LEN(Name)/2, 1)** M1,

**RIGHT(Name, 1)** R1

FROM Vendors

/\* Q8.10 \*/

SELECT InvoiceTotal,

**STR(InvoiceTotal, 10, 0)** [Whole Dollar Amt]

FROM Invoices

WHERE InvoiceTotal > 10000

Use the cast (as decimal(10, 0))

/\* Q8.11 \*/

SELECT Name, CONCAT(LEFT(Name, 1),

**LOWER(**SUBSTRING(Name, LEN(Name)/2, 1)**)**,

**UPPER(**RIGHT(Name, 1)**)**, '-',

**LTRIM(STR(VendorID, 3))**) [Vendor Code]

FROM Vendors

/\* Q8.12 \*/

SELECT InvoiceID, [Payment Term] =

**CASE** TermsID

**WHEN** 1 **THEN** '10 days'

**WHEN** 2 **THEN** '20 days'

**WHEN** 3 **THEN** '30 days'

**WHEN** 4 **THEN** '60 days'

**WHEN** 5 **THEN** '90 days'

**ELSE** 'Invalid code!'

**END**

FROM Invoices

WHERE YEAR(InvoiceDate) = 2010

/\* Q8.13 \*/

SELECT InvoiceID, [Payment Status] =

**CASE**

**WHEN** PaymentDate IS NULL **THEN** 'Not paid'

**WHEN** DateDiff(day, InvoiceDate, PaymentDate) <

DateDiff(day, InvoiceDate, DueDate)

**THEN** 'On time'

**WHEN** DateDiff(day, InvoiceDate, PaymentDate) <

DateDiff(day, InvoiceDate, DueDate) + 5

**THEN** 'Marginal'

**ELSE** 'Past due!'

**END**

FROM Invoices

WHERE YEAR(InvoiceDate) = 2010