**Part 26**

**ISDATE**() - Checks if the given value, is a valid date, time, or datetime. Returns 1 for success, 0 for failure.  
  
**Examples:**  
Select ISDATE('PRAGIM') -- returns 0  
Select ISDATE(Getdate()) -- returns 1  
Select ISDATE('2012-08-31 21:02:04.167') -- returns 1  
  
**Note**: For datetime2 values, IsDate returns ZERO.  
  
**Example**:  
Select ISDATE('2012-09-01 11:34:21.1918447') -- returns 0.

**Day**() - Returns the **'Day number of the Month'** of the given date  
  
**Examples**:  
Select DAY(GETDATE()) -- Returns the day number of the month, based on current system datetime.  
Select DAY('01/31/2012') -- Returns 31  
  
**Month**() - Returns the **'Month number of the year'** of the given date  
  
**Examples**:  
Select Month(GETDATE()) -- Returns the **Month number of the year**, based on the current system date and time  
Select Month('01/31/2012') -- Returns 1  
  
**Year**() - Returns the**'Year number'**of the given date  
  
**Examples:**  
Select Year(GETDATE()) -- Returns the year number, based on the current system date  
Select Year('01/31/2012') -- Returns 2012  
  
**DateName**(DatePart, Date) - Returns a string, that represents a part of the given date. This functions takes 2 parameters. The first parameter **'DatePart'** specifies, the part of the date, we want. The second parameter, is the actual date, from which we want the part of the Date.

**Examples:**  
Select DATENAME(Day, '2012-09-30 12:43:46.837') -- Returns 30  
Select DATENAME(WEEKDAY, '2012-09-30 12:43:46.837') -- Returns Sunday

Select DATENAME(MONTH, '2012-09-30 12:43:46.837') -- Returns September

**DatePart**(DatePart, Date) - Returns an integer representing the specified DatePart. This function is simialar to DateName(). DateName() returns nvarchar, where as DatePart() returns an integer. The valid DatePart parameter values are shown below.

**Examples:**  
Select DATEPART(weekday, '2012-08-30 19:45:31.793') -- returns 5  
Select DATENAME(weekday, '2012-08-30 19:45:31.793') -- returns Thursday

**DATEADD**(datepart, NumberToAdd, date) - Returns the DateTime, after adding specified NumberToAdd, to the datepart specified of the given date.  
  
**Examples:**  
Select DateAdd(DAY, 20, '2012-08-30 19:45:31.793')   
-- Returns 2012-09-19 19:45:31.793  
Select DateAdd(DAY, -20, '2012-08-30 19:45:31.793')   
-- Returns 2012-08-10 19:45:31.793  
  
**DATEDIFF**(datepart, startdate, enddate) - Returns the count of the specified datepart boundaries crossed between the specified startdate and enddate.  
  
**Examples:**  
Select DATEDIFF(MONTH, '11/30/2005','01/31/2006') -- returns 2  
Select DATEDIFF(DAY, '11/30/2005','01/31/2006') -- returns 62

CREATE FUNCTION fnComputeAge(@DOB DATETIME)  
RETURNS NVARCHAR(50)  
AS  
BEGIN  
  
DECLARE @tempdate DATETIME, @years INT, @months INT, @days INT  
SELECT @tempdate = @DOB  
  
SELECT @years = DATEDIFF(YEAR, @tempdate, GETDATE()) - CASE WHEN (MONTH(@DOB) > MONTH(GETDATE())) OR (MONTH(@DOB) = MONTH(GETDATE()) AND DAY(@DOB) > DAY(GETDATE())) THEN 1 ELSE 0 END  
SELECT @tempdate = DATEADD(YEAR, @years, @tempdate)

SELECT @months = DATEDIFF(MONTH, @tempdate, GETDATE()) - CASE WHEN DAY(@DOB) > DAY(GETDATE()) THEN 1 ELSE 0 END  
SELECT @tempdate = DATEADD(MONTH, @months, @tempdate)  
  
SELECT @days = DATEDIFF(DAY, @tempdate, GETDATE())  
  
DECLARE @Age NVARCHAR(50)  
SET @Age = Cast(@years AS  NVARCHAR(4)) + ' Years ' + Cast(@months AS  NVARCHAR(2))+ ' Months ' +  Cast(@days AS  NVARCHAR(2))+ ' Days Old'  
RETURN @Age  
  
End