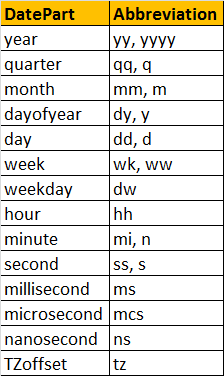
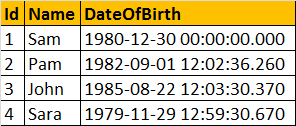
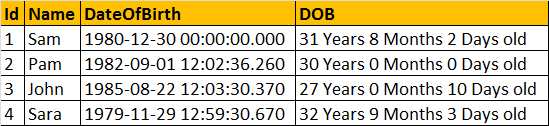
**DatePart, DateAdd and DateDiff functions in SQL Server - Part 27**

**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  
  
Consider the emaployees table below.  
  
  
Write a query to compute the age of a person, when the date of birth is given. The output should be as shown below.  
  
  
  
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  
  
**Using the function in a query to get the expected output along with the age of the person.**  
Select Id, Name, DateOfBirth, dbo.fnComputeAge(DateOfBirth) as Age from tblEmployee