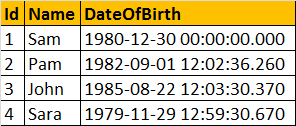
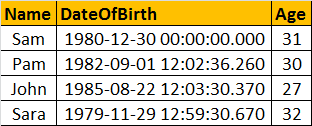
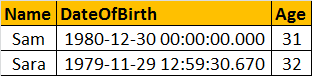
**Scalar User Defined Functions in sql server - Part 30**

From **Parts 22 to 29**, we have learnt how to use many of the built-in system functions that are available in SQL Server. In this session, we will turn our attention, to creating **user defined functions**. In short **UDF**.  
  
**We will cover**  
1. User Defined Functions in sql server  
2. Types of User Defined Functions  
3. Creating a Scalar User Defined Function  
4. Calling a Scalar User Defined Function  
5. Places where we can use Scalar User Defined Function  
6. Altering and Dropping a User Defined Function  
  
  
  
  
  
  
  
**In SQL Server there are 3 types of User Defined functions**  
1. Scalar functions  
2. Inline table-valued functions  
3. Multistatement table-valued functions  
  
**Scalar functions** may or may not have parameters, but always return a single (scalar) value. The returned value can be of any data type, except **text, ntext, image, cursor, and timestamp**.  
  
**To create a function, we use the following syntax:**  
CREATE FUNCTION Function\_Name(@Parameter1 DataType, @Parameter2 DataType,..@Parametern Datatype)  
RETURNS Return\_Datatype  
AS  
BEGIN  
    Function Body  
    Return Return\_Datatype  
END  
  
Let us now create a function which calculates and returns the age of a person. To compute the age we require, date of birth. So, let's pass date of birth as a parameter. So, AGE() function returns an integer and accepts date parameter.  
CREATE FUNCTION Age(@DOB Date)    
RETURNS INT    
AS    
BEGIN    
 DECLARE @Age INT    
 SET @Age = DATEDIFF(YEAR, @DOB, GETDATE()) - CASE WHEN (MONTH(@DOB) > MONTH(GETDATE())) OR (MONTH(@DOB) = MONTH(GETDATE()) AND DAY(@DOB) > DAY(GETDATE())) THEN 1 ELSE 0 END    
 RETURN @Age    
END  
  
  
**When calling a scalar user-defined function**, you must supply a two-part name, **OwnerName.FunctionName**. **dbo** stands for database owner.  
Select dbo.Age( dbo.Age('10/08/1982')  
  
**You can also invoke it using the complete 3 part name**, DatabaseName.OwnerName.FunctionName.  
Select SampleDB.dbo.Age('10/08/1982')  
  
**Consider the Employees table below.**  
  
  
**Scalar user defined functions can be used in the Select clause** as shown below.  
Select Name, DateOfBirth, dbo.Age(DateOfBirth) as Age from tblEmployees  
  
  
**Scalar user defined functions can be used in the Where clause**, as shown below.  
Select Name, DateOfBirth, dbo.Age(DateOfBirth) as Age   
from tblEmployees  
Where dbo.Age(DateOfBirth) > 30  
  
  
  
**A stored procedure** also can accept DateOfBirth and return Age, but you cannot use stored procedures in a **select or where clause**. This is just one difference between a function and a stored procedure. There are several other differences, which we will talk about in a later session.  
  
To alter a function we use ALTER FUNCTION FuncationName statement and to delete it, we use DROP FUNCTION FuncationName.  
  
To view the text of the function use sp\_helptext FunctionName