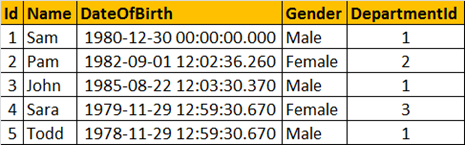
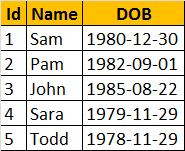
We have discussed about [**scalar functions in Part 29**](http://csharp-video-tutorials.blogspot.com/2012/09/scalar-user-defined-functions-in-sql.html) and [**Inline Table Valued functions in Part 30**](http://csharp-video-tutorials.blogspot.com/2012/09/inline-table-valued-functions-part-31.html). In this video session, we will discuss about Multi-Statement Table Valued functions.  
  
   
  
   
  
   
  
Multi statement table valued functions are very similar to Inline Table valued functions, with a few differences. Let's look at an example, and then note the differences.   
  
**Employees Table:**  
  
  
**Let's write an Inline and multi-statement Table Valued functions that can return the output shown below.**  
  
  
  
**Inline Table Valued function(ILTVF):**  
Create Function fn\_ILTVF\_GetEmployees()  
Returns Table  
as  
Return (Select Id, Name, Cast(DateOfBirth as Date) as DOB  
        From tblEmployees)  
  
  
**Multi-statement Table Valued function(MSTVF):**  
Create Function fn\_MSTVF\_GetEmployees()  
Returns @Table Table (Id int, Name nvarchar(20), DOB Date)  
as  
Begin  
 Insert into @Table  
 Select Id, Name, Cast(DateOfBirth as Date)  
 From tblEmployees  
   
 Return  
End  
  
**Calling the Inline Table Valued Function:**  
Select \* from fn\_ILTVF\_GetEmployees()  
  
**Calling the Multi-statement Table Valued Function:**  
Select \* from fn\_MSTVF\_GetEmployees()  
  
**Now let's understand the differences between Inline Table Valued functions and Multi-statement Table Valued functions**  
1. In an Inline Table Valued function, the RETURNS clause cannot contain the structure of the table, the function returns. Where as, with the multi-statement table valued function, we specify the structure of the table that gets returned  
2. Inline Table Valued function cannot have BEGIN and END block, where as the multi-statement function can have.  
3. Inline Table valued functions are better for performance, than multi-statement table valued functions. If the given task, can be achieved using an inline table valued function, always prefer to use them, over multi-statement table valued functions.  
4. It's possible to update the underlying table, using an inline table valued function, but not possible using multi-statement table valued function.  
  
**Updating the underlying table using inline table valued function:**  
This query will change **Sam** to **Sam1**, in the underlying table **tblEmployees**. When you try do the same thing with the multi-statement table valued function, you will get an error stating 'Object 'fn\_MSTVF\_GetEmployees' cannot be modified.'  
Update fn\_ILTVF\_GetEmployees() set Name='Sam1' Where Id = 1  
  
**Reason for improved performance of an inline table valued function:**  
Internally, SQL Server treats an inline table valued function much like it would a view and treats a multi-statement table valued function similar to how it would a stored procedure.

[Email This](https://www.blogger.com/share-post.g?blogID=6082652835152798567&postID=6501043248302489021&target=email)[BlogThis!](https://www.blogger.com/share-post.g?blogID=6082652835152798567&postID=6501043248302489021&target=blog)[Share to Twitter](https://www.blogger.com/share-post.g?blogID=6082652835152798567&postID=6501043248302489021&target=twitter)[Share to Facebook](https://www.blogger.com/share-post.g?blogID=6082652835152798567&postID=6501043248302489021&target=facebook)[Share to Pinterest](https://www.blogger.com/share-post.g?blogID=6082652835152798567&postID=6501043248302489021&target=pinterest)