National University of Computer and Emerging Sciences

Lab Manual

"SQL User Defined Functions"

Database Systems Lab

Spring 2024

Department of Computer Science
FAST-NU, Lahore, Pakistan

Table of Contents

1 Objective

We will also study User defined functions, it's comparison with stored procedures and views.

2 Task Distribution

Total Time	45 Minutes
UDF (Scalar and Inline)	15 Minutes
Exercise	15 Minutes

3

User defined functions

User defined functions are routines that encapsulates SQL logic inside it. Like stored procedures User defined functions can also be passed input parameters but user defined functions are compiled and executed at runtime. Therefore, they are slower than stored procedures.

Syntax:

```
CREATE FUNCTION dbo.Function
    (
    /*
    @parameter1 datatype = default value,
    @parameter2 datatype
    */
    )
  RETURNS /* datatype */
  AS
    BEGIN
     /* sql statement ... */
    RETURN/* value */
   END
```

Certain limitations for User defined functions:

i) UDF can't perform DML (data manipulation language) operations like Insertion, Update and Deletion on the base table.

ii) UDF can't return non deterministic values like GETDATE () etc.

iii) Stored procedure can't be called from inside a UDF whereas a stored procedure can call a user defined function or another stored procedure inside it.

There are two types of user defined functions:

- Scalar
- Inline

3.1 Scalar Functions (returns a single value)

Example:

```
CREATE FUNCTION EmployeeContactID(@Empid int)
RETURNS int
AS
BEGIN

Declare @returnvalue int
Select @returnvalue=Employee.ContactID from Employee where Employee.EmployeeID=@Empid
RETURN   @returnvalue
END
```

Execution:

```
select dbo.EmployeeContactID(1)
```

3.2 Inline Functions (returns a table)

Example:

```
CREATE FUNCTION dbo.GetEmployeeFunction(@empid int)
RETURNS TABLE
AS
RETURN SELECT*
FROM employee where employee.EmployeeID=@empid
```

Execution:

```
Select * from dbo.GetEmployeeFunction(1)
```

3.3 Difference between procedures and user defined functions

1. A stored procedure is pre compiled while a User defined function is compiled and executed at runtime.

2. A Stored procedure is more flexible than user defined function like you can write complex logic (for example exceptional handling using try catch block is possible in stored procedures which is not possible in user defined functions)
3. A stored procedure can call another stored procedure or user defined function inside it but a user defined function can't call stored procedure inside it.
4. A stored procedure can return non deterministic values but a user defined function can't return a non-deterministic value like GetDate() function.
5. A user defined functions does not support DML operations like insertion, deletion and update on the base table but it is possible via stored procedure.
6. A user defined function is easier to execute and can be used inside selection and even for joins but stored procedure can't be used inside selection queries and it can't be used to join with other tables.

3.4 Some comparison with views

If you think of view than a question might arise in your mind why we don't use views instead of stored procedures or user defined functions for basic SQL selection queries. Answer is flexibility. You can't pass parameters to views for selection of filtered queries but stored procedures and user defined functions provide you these features. Similarly, Multiple DML operations are restricted in views which are possible through stored procedures and user defined functions.

```
CREATE VIEW Designation_Of_Marital_Status AS
SELECT Employee.Name,Employee.Title
FROM Employee
WHERE Employee.MaritalStatus ='M'
```

Execute

```
Select *
From [Designation_Of_Marital_Status]
```