

Functions

for EFP0700 Database Servers



Alexey Tukalo,
EFA12SF,
Information Technology,
Savonia University of Applied Sciences

November 11, 2014

1 SQL Server Functions

1.1 The first function

The first function should return a table with all project of the employee whose number was send to the function as an argument.

```

1  USE [test_db]
2  GO
3  /***** Object: UserDefinedFunction [dbo].[employeeProjects]    Script Date: 10.11.2014 23:03:51 *****/
4  SET ANSI_NULLS ON
5  GO
6  SET QUOTED_IDENTIFIER ON
7  GO
8  ALTER FUNCTION [dbo].[employeeProjects]
9  (
10     @emp_n INT
11 )
12 RETURNS TABLE
13 AS
14 RETURN
15 (
16     SELECT
17         (SELECT emp_fname FROM employee WHERE emp_no=@emp_n) AS 'Firstname',
18         (SELECT emp_lname FROM employee WHERE emp_no=@emp_n) AS 'Lastname', project_name
19     FROM project WHERE project_no IN (SELECT project_no FROM works_no WHERE emp_no=@emp_n)
20 )
21

```

The output of the function for argument equal 2 is like that:

```

1  USE [test_db]
2  GO
3
4  SELECT * FROM [dbo].[employeeProjects] (2)
5  GO
6

```

Firstname	Lastname	project_name
Sergey	Lukanov	Excel
Sergey	Lukanov	PowerPoint

1.2 The second function

The second task was the most advanced one, which I have made during whole the course, because it requires to use two cursors at the same time. The code you can see below.

There is also very tricky moment with string summering, because the statements like $a=a+b$ does not work, but $a=b+a$ works well and actually return the output which should be produced by the first kind of statements.

```

1  USE [test_db]
2  GO
3  /***** Object:  UserDefinedFunction [dbo].[emp_dept]    Script Date: 10.11.2014 23:06:31 *****/
4  SET ANSI_NULLS ON
5  GO
6  SET QUOTED_IDENTIFIER ON
7  GO
8  ALTER FUNCTION [dbo].[emp_dept] ()
9  RETURNS
10 @results TABLE
11 (
12     deptName CHAR(25),
13     employees CHAR(1000)
14 )
15 AS
16 BEGIN
17     DECLARE @deptName CHAR(25), @deptNO CHAR(4), @employeeCell CHAR(1000), @fname CHAR(20), @lname CHAR(20)
18     DECLARE @employees CURSOR
19     DECLARE deptName CURSOR FOR SELECT dept_name, dept_no FROM department;
20     OPEN deptName;
21     FETCH NEXT FROM deptName INTO @deptName, @deptNO
22     WHILE @@FETCH_STATUS=0
23     BEGIN
24         SET @employees = CURSOR FOR SELECT emp_fname, emp_lname FROM employee WHERE dept_no=@deptNO;
25         OPEN @employees;
26         SET @employeeCell=''
27         FETCH NEXT FROM @employees INTO @fname, @lname
28         WHILE @@FETCH_STATUS=0
29         BEGIN
30             SET @employeeCell=@fname+' '+@lname+@employeeCell
31             FETCH NEXT FROM @employees INTO @fname, @lname
32             IF @@FETCH_STATUS=0
33             BEGIN
34                 SET @employeeCell=', '+@employeeCell
35             END
36         END
37         CLOSE @employees
38         INSERT INTO @results VALUES(@deptName, @employeeCell)
39         FETCH NEXT FROM deptName INTO @deptName, @deptNO
40     END
41     CLOSE deptName
42     RETURN
43 END
44

```

```

1  USE [test_db]
2  GO
3
4  SELECT * FROM [dbo].[emp_dept] ()
5  GO
6
7

```

deptName	employees
Tuke	Daria Doncova
Bird	Artur Doile, Sergey Lukanov
Car	
Horse	Leo Tolstoy
Three	Josev Stalin

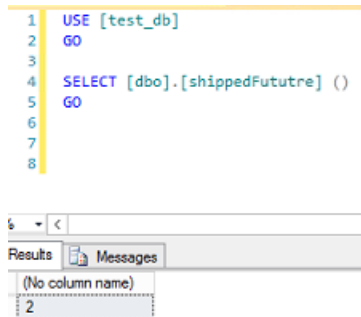
The output of the function follows task requirements.

1.3 The third function

The third task was really easy and the code is very short.

```
1  USE [test_db]
2  GO
3  /***** Object: UserDefinedFunction [dbo].[shippedFututre]    Script Date: 10.11.2014 23:10:23 *****/
4  SET ANSI_NULLS ON
5  GO
6  SET QUOTED_IDENTIFIER ON
7  GO
8  ALTER FUNCTION [dbo].[shippedFututre]()
9  RETURNS int
10 AS
11 BEGIN
12     DECLARE @result INT
13
14
15     SELECT @result=count(*) FROM orders WHERE shipdate>CONVERT(date,GETDATE())
16
17
18     RETURN @result
19
20 END
21
```

And the output is correct.



2 Tables

dept_no	dept_name	location
1	Tuke	Kuopio ...
2	Bird	Mikkeli ...
3	Car	Helsinki ...
4	Horse	Turku ...
5	Three	Jounnsue ...

Figure 1: department table

emp_no	emp_fname	emp_lname	dept_no
1	Daria	Doncova	1
2	Sergey	Lukanov	2
3	Artur	Doile	2
4	Leo	Tolstoy	4
5	Josev	Stalin	5

Figure 2: employee table

orederid	customerid	orderdate	shipdate	city	freighy	shipname	shipaddress	fax	quantity
3	1	2014-03-01	2014-03-08	Kuopio	2,5000	Jon	Sun st. 7	234534536 ...	3
4	1	2014-03-01	2014-12-08	Kuopio	3,0000	Jill	See st. 5	231234532 ...	2
5	1	2003-05-06	2015-12-09	Turku	4,0000	Jordon	Look st. 13	123453435 ...	5

Figure 3: orders table

emp_no	project_no	job	enter_date
2	5	3	2012-07-21
2	3	5	2014-06-03
1	2	4	2013-05-05
3	4	6	2012-05-03
4	1	2	2030-08-06

Figure 4: works_no