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.

```
USE [test_db]
GO
/****** Object:  UserDefinedFunction [dbo].[emploeeyProjects]    Script Date: 10.11.2014 23:03:51 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER FUNCTION [dbo].[emploeeyProjects]
(
    @emp_n INT
)
RETURNS TABLE
AS
RETURN
(
    SELECT
    (SELECT emp_fname FROM employee WHERE emp_no=@emp_n) AS 'Firstname',
    (SELECT emp_lname FROM employee WHERE emp_no=@emp_n) AS 'Lastname',project_name
    FROM project WHERE project_no IN (SELECT project_no FROM works_no WHERE emp_no=@emp_n)
)

```

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

```
USE [test_db]
GO

SELECT * FROM [dbo].[emploeeyProjects] (2)
GO
```

Results | Messages

Firstname	Lastname	project_name
Sergey	Lukanov	Excell
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.

```
USE [test_db]
GO
/****** Object:  UserDefinedFunction [dbo].[emp_dept]    Script Date: 10.11.2014 23:06:31 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER FUNCTION [dbo].[emp_dept] ()
RETURNS
@results TABLE
(
    deptName CHAR(25),
    employees CHAR(1000)
)
AS
BEGIN
    DECLARE @deptName CHAR(25), @deptNO CHAR(4), @employeeCell CHAR(1000), @fname CHAR(20), @lname CHAR(20)
    DECLARE @employees CURSOR
    DECLARE deptName CURSOR FOR SELECT dept_name, dept_no FROM department;
    OPEN deptName;
     FETCH NEXT FROM deptName INTO @deptName,@deptNo
      WHILE @@FETCH_STATUS=0
      BEGIN
         SET @employees = CURSOR FOR  SELECT emp_fname,emp_lname FROM employee WHERE dept_no=@deptNO;
         OPEN @employees;
         SET @employeeCell=''
         FETCH NEXT FROM @employees INTO @fname,@lname
             WHILE @@FETCH_STATUS=0
             BEGIN
             SET @employeeCell=@fname+' '+@lname+@employeeCell
              FETCH NEXT FROM @employees INTO @fname,@lname
                IF @@FETCH_STATUS=0
                BEGIN
                SET @employeeCell=', '+@employeeCell
                END
              END
              CLOSE @employees
              INSERT INTO @results VALUES(@deptName,@employeeCell)
         FETCH NEXT FROM deptName INTO @deptName,@deptNo
      END
      CLOSE deptName
    RETURN
END

```

```
USE [test_db]
GO

SELECT * FROM [dbo].[emp_dept] ()
GO

```

Results | Messages

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.

```
USE [test_db]
GO
/****** Object:  UserDefinedFunction [dbo].[shippedFututre]    Script Date: 10.11.2014 23:10:23 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER FUNCTION [dbo].[shippedFututre]()
RETURNS int
AS
BEGIN
    DECLARE @result INT


    SELECT @result=count(*) FROM orders WHERE shipdate>CONVERT(date,GETDATE())


    RETURN @result

END

```

And the output is correct.

```
USE [test_db]
GO

SELECT [dbo].[shippedFututre] ()
GO
```

Results | Messages

(No column name)
2

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