**Database Management System – cs422 DE**

**Lab 3 – Week 7**

-----------------------------------------------------------------------------------------------------------------

**This Lab is based on Transact-SQL.**

* Submit your *own work* on time. No credit will be given if the lab is submitted after the due date.
* Note that the completed lab should be submitted in .doc, .docx, .rtf, .pdf or .zip format only.   
  ---------------------------------------------------------------------------------------------------------------

1. [3] Write and execute a T-SQL stored procedure *Factorial*(*n*), which computes and outputs the factorial of the input parameter *n*. If *n* is negative, then the procedure prints an error message.

Attach the screenshots of the output and the command which you used to execute the SP.

ANS:

CREATE PROCEDURE Factorial (@param INT) AS

BEGIN

DECLARE @fact bigint

IF(@param < 0)

BEGIN

print 'Error: The value should not the negative.'

END

ELSE

BEGIN

SET @fact = 1

IF(@param = 0)

BEGIN

SET @fact = 1

END

ELSE

BEGIN

WHILE(@param >0)

BEGIN

SET @fact = @fact \* @param

SET @param = @param -1

END

END

print @fact

END

END

**Factorial(5)**

Graphical user interface, text, application, email

Description automatically generated

**Factorial(-3)**

Graphical user interface, text, application, email

Description automatically generated

1. [7] Create a Table *Employee* with the fields: social security no. (primary key), name, position, no. of dependents, annual salary.

Write and execute a T-SQL procedure *Compute\_Tax* to do the following:

* Create a new table *Tax* with fields: social security no., income tax.
* Fill the table *Tax* with data by computing the income tax for each person in the Employee Table.

The income tax is computed from the annual salary S and the number of dependents D.

Net Salary: S - (7000 + D\*950)

Tax Computed as follows:

* 10% of the first 15,000 of net salary;
* plus 15% of the next 15,000 of net salary;
* plus 28% of any net salary over 30,000.

For getting full credit for this problem, you need to show me the complete code for the *Compute\_Tax* SP. Also attach the screenshots of the *Employee* and the new *Tax* table.

ANS:

CREATE PROCEDURE compute\_Tax AS

BEGIN

DECLARE @SSN VARCHAR(30)

DECLARE @NoOfDependent INT

DECLARE @AnnualSalary float

DECLARE @NetSalary float

DECLARE @Tax float

DECLARE @Done bit

DECLARE empCur CURSOR FOR

SELECT SSN, NoOfDependent, AnnualSalary FROM Employee;

OPEN empCur;

FETCH next from empCur INTO @SSN, @NoOfDependent, @AnnualSalary;

IF not exists (select \* from sysobjects where Name='TAX')

Begin

CREATE TABLE TAX (SSN VARCHAR(30), incomeTax float);

End

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @NetSalary = @AnnualSalary - (7000 + @NoOfDependent \* 950);

IF (@NetSalary < 0 )

begin

BREAK;

End

IF (@NetSalary < 15000)

begin

SET @Tax = 0.1 \* @NetSalary;

end

ELSE

begin

IF (@NetSalary > 15000)

begin

IF (@NetSalary > 30000)

begin

SET @Tax = 0.10 \* 15000 + 0.15 \* 15000 + 0.28 \* (@NetSalary - 30000);

end

ELSE

begin

SET @Tax = 0.10 \* 15000 + 0.15 \* (@NetSalary -15000);

end

end

end

INSERT INTO TAX(SSN, incomeTax) VALUES(@SSN, @Tax);

FETCH next from empCur INTO @SSN, @NoOfDependent, @AnnualSalary;

END

CLOSE empCur;

DEALLOCATE empCur;

END;

**Employee Table:**

Table

Description automatically generated

**Tax Table:**

Table

Description automatically generated