Practice 10.

Control statements, cursor, UDF, Stored Procedures

USE the ScottDB database to solve the exercises.

1. Declare a MinIncome variable and assign to it *the minimum income* from the **emp** table.
2. Write a T-SQL script to create a copy of the emp table to the TempEmp table and then increase the salary with 500 $ of those employees who work as a clerk. Create a **cursor** to solve the problem. Test your solution. Drop the TempEmp table.
3. Write a T-SQL script to copy the emp table into the TempEmp table. Increase the salary of those employees with a given percent (%), whose salary is less than the average salary. Use **cursor** to solve the problem. Test your solution! Drop the TempEmp table.

Create User Defined Functions

Use the Northwind database. Scalar-value functions

4. Create a function which returns the maximum UnitPrice value of a given product category. Test your solution. Drop the function.
5. Create a scalar function which returns the age! The function should have two input parameters, the one is the current date and the other one is the birthdate. Test the function! Display the employees' employeeID, full name, birthdate fields from Employees table and calculate the age of every employees using the function. Drop the function!

Table-valued UDF

1. Create a function to return the list of orders sent in a given time period! Test your function! Drop the function.
2. Create a function to return the list of products (ProductID, ProductName and the UnitPrice fields) which UnitPrice value is in a range given by the user. Test the function. Drop the function.

Stored Procedure – Use ScottDB database

3. Create a stored procedure which retrieve the data of an employee given by the empno input parameter. Test your solution! Drop the stored procedure.
4. Create a stored procedure to list the employees, whose income is between a given income interval. The result should be contain the empname, job, deptname, loc and income values. Ordered the result according to the deptname and the empname.
5. Create a stored procedure which retrieve employees who works for a given department. The department name (dname) is input parameter. Test your solution! Drop the stored procedure.
6. Create a stored procedure to list the employees, whose hiredate is between a given time interval. The result should be contain the empname, job, deptname, and hiredate values. Ordered the result according to the deptname and the empname.
7. Create a stored procedure, which insert a new employee to the emp table.

Stored Procedure – Use Northwind

8. Create a stored procedure to retrieve the five cheapest products from Products table!
9. Create a copy of the Products table to the Prod table! Create a Stored procedure, which increase the UnitsInStock values of each products with an input parameter value. Test your solution. Drop the Prod table, drop the stored procedure.
10. Create a stored procedure that accepts a ProductID for a parameter and has an OUTPUT parameter that returns the total number sold for the product (Use the OrderDetails table). Test the stored procedure. Drop the stored procedure.