**LAB - 9**

**Stored Procedure**

1. Write 3 stored procedures :

* 1st to Insert Employee detail

**Query:**

CREATE PROC EMP @LastName NVARCHAR(20), @FirstName NVARCHAR(10), @employeeid INT AS INSERT INTO Employees VALUES (@employeeid, @LastName, @FirstName)

EXEC Emp 'Hassan', 'Ali', '123';

EXEC Emp 'shah', 'amir', '163';

**Result:**



* 2nd to Update Employee detail &

**Query:**

ALTER PROC EMP @LastName NVARCHAR(20), @FirstName NVARCHAR(10), @employeeid INT AS SELECT \* FROM Employees WHERE lastname = @LastName

EXEC EMP 'shah', 'amir', '163';

**Result:**



* 3rd to Delete Employee

**Query:**

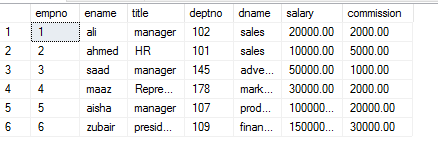
ALTER PROC EMP @LastName NVARCHAR(20), @FirstName NVARCHAR(10), @employeeid INT AS DELETE FROM Employees WHERE lastname = @LastName

EXEC emp 'Hassan', 'Ali', '123';

**Result:**



1. Create a stored procedure **DISPLAY** without parameters. The procedure must display empno, ename and title of all the employees of DEPTNO = 101.



**Query:**

CREATE PROC DISPLAY AS SELECT empno, ename, title, deptno FROM EMP WHERE deptno = 101

EXEC DISPLAY

**Result:**



1. Create a stored procedure **DISPLAY2** with parameters. It must take DEPTNO as an input and must return the DNAME and TOTAL SALARY of the input department number.

**Query:**

CREATE PROC Display2 @dept INT AS SELECT dname, salary+commission AS TotalSalary FROM EMP WHERE deptno = @dept

EXEC display2 '107'

**Result:**



1. Create a stored procedure **DISPLAY3** with parameters. It must take DEPTNO as an input and must return the DNAME, SMALLEST and HIGHEST SALARIES of the input department number. DISPLAY3 must also display empno,ename,total salary (sal+comm) of all the employees of the input department number.

**Query:**

CREATE PROC Display3 @dept INT AS SELECT dname, salary + commission AS TotalSalary, deptno, empno, ename, salary = CASE WHEN salary >= (SELECT AVG(salary) FROM EMP) THEN 'HighestSalary'

ELSE 'SmallestSalary'

END FROM EMP WHERE deptno = @dept

**Output:**

**For Highest:**

exec display3 '107'



**For Smallest:**

exec display3 '101'



1. Write a stored procedure to increase unit price of all products by specified percentage in product table of Northwind database.

Query:

CREATE PROC PRO @up MONEY

AS

BEGIN

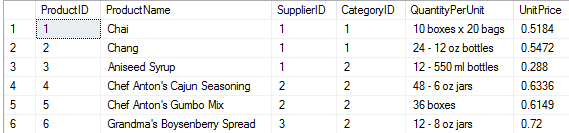
UPDATE Products SET UnitPrice = ((UnitPrice \* (@up/100)) + UnitPrice);

END

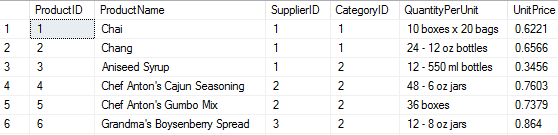
EXEC pro '20'

**Result:**

**Before:**



**After:**



1. Using stored procedures create a calculator that perform a function based on input

**Query:**

alter procedure Calculator @a int, @b int ,@opr char as

select @a as 'Number1',@b as 'Number2', @opr as 'Operator',"Answer" = case @opr

when '+' then @a+@b

when '-' then @a-@b

when '\*' then @a\*@b

when '/' then @a/@b end

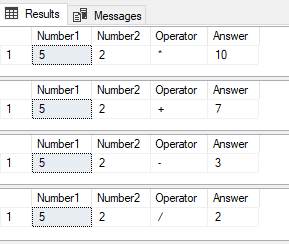
exec Calculator 5,2,'\*'

exec Calculator 5,2,'+'

exec Calculator 5,2,'-'

exec Calculator 5,2,'/'

**Output:**



1. Display Author\_Name, Author\_ID, Title\_Id and Type from Author and TitleAuthor tables using pubs database.

4th Column named **Type** is based on following case statement:

when TitleId starts with **BU** then its Type is **Business**

when TitleId starts with **TC** then its Type is **Traditional Cooking**

when TitleId starts with **PC** then its Type is **Popular Computing**

when TitleId starts with **MC** then its Type is **Modern Cooking**

when TitleId starts with **PS**  then its Type is **Psychology**

**Query:**

create procedure AuthorDetails as select authors.au\_id, au\_fname+' '+au\_lname as AuthorName,title\_id,"Type"=case

when title\_id like 'BU%' then 'Business'

when title\_id like 'TC%' then 'Traditional Cooking'

when title\_id like 'PC%' then 'Popular Computing'

when title\_id like 'MC%' then 'Modern Cooking'

when title\_id like 'PS%' then 'Psychology' end from authors inner join titleauthor on authors.au\_id=titleauthor.au\_id

exec AuthorDetails

**Output:**

