**IST 659 LAB Assignment NO -7**

1. Add an employee using your own name and create a project assignment for yourself using existing project id.

**Query:**

/\*1) Add an employee using your own name \*/

INSERT INTO employee VALUES (11118, 'Rahuk', 'Rathod', 100000);

/\* updating the employeeFName from Rahuk to Rahul \*/

update employee

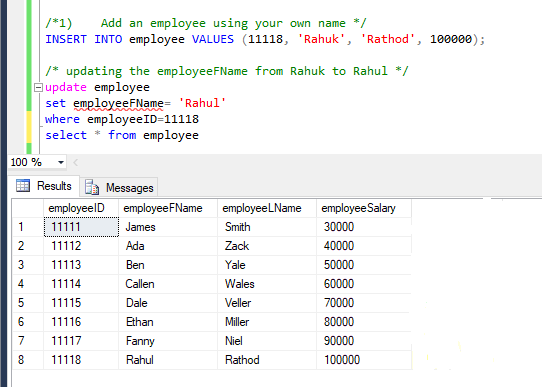
set employeeFName= 'Rahul'

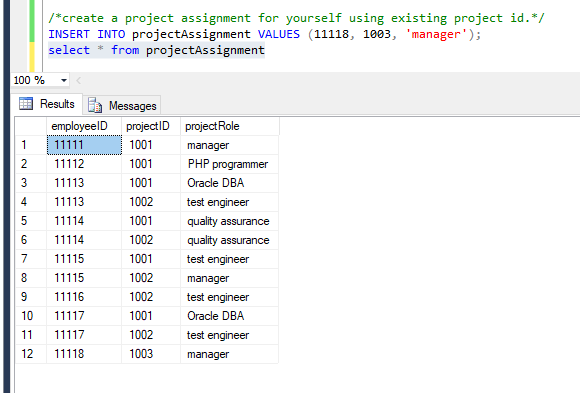
where employeeID=11118

/\*create a project assignment for yourself using existing project id.\*/

INSERT INTO projectAssignment VALUES (11118, 1003, 'manager');

**Result:**





1. Write a scalar function that returns the average salary of the Employees

**Query:**

/\* scalar function that returns the average salary of the Employees\*/

create function averageSalary(@i int)

returns decimal(10,1)

as

begin

declare @ret INT;

select @ret= avg(employeeSalary)

from employee

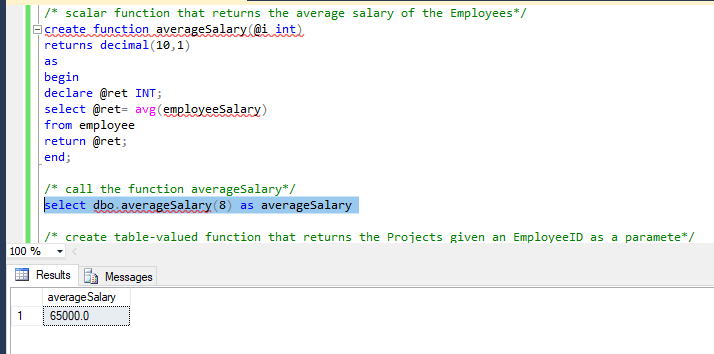
return @ret;

end;

/\* call the function averageSalary\*/

select dbo.averageSalary(8) as averageSalary

**Result:**

****

1. Write a table-valued function that returns the Projects given an EmployeeID as a parameter and
   1. Show the function created
   2. return the results for your own project

**Query:**

/\* create table-valued function that returns the Projects given an EmployeeID as a paramete\*/

create function employeeProject(@employeeID int)

returns table

as

return

(

select pa.projectId as employeeProject

from projectAssignment pa

Inner Join employee e on pa.employeeID = e.employeeID

Inner Join project p on pa.projectID = p.projectID

where e.employeeID = @employeeID

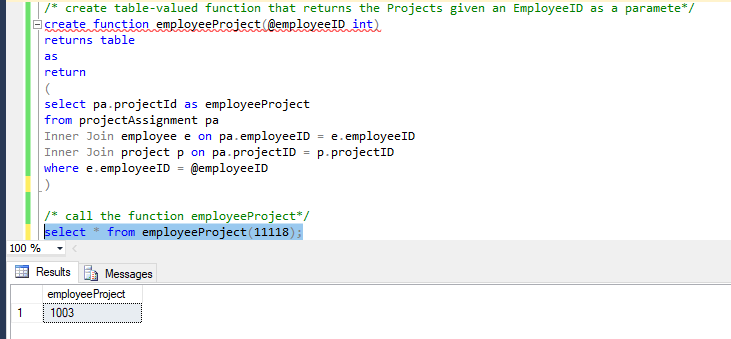
)

drop function dbo.employeeProject

/\* call the function employeeProject\*/

select \* from employeeProject(11118);

**Result:**

****

1. Alter the Employee table to add a new column called ‘Num of Projects’ which can be **INTEGER** data type. Write a procedure that updates employee table with the total projects assigned to each employee (to the newly created column)

**Query:**

/\*Alter the Employee table to add a new column called ‘Num of Projects’ \*/

alter table employee add NumOfProjects int;

select \* from employee

CREATE PROCEDURE TOTAL\_PROJ

AS

BEGIN

UPDATE EMPLOYEE

SET NumOfProjects = projCount.total\_count

FROM

(

SELECT employeeID,count(projectID) AS total\_count

FROM projectAssignment

GROUP BY employeeID

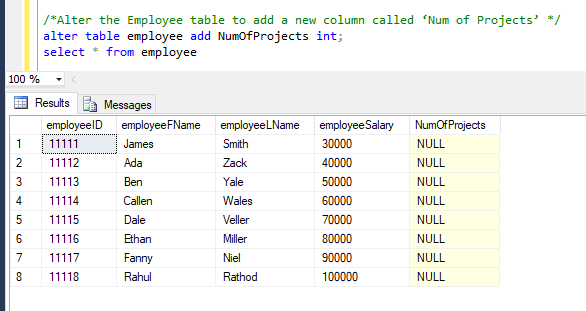
) projCount

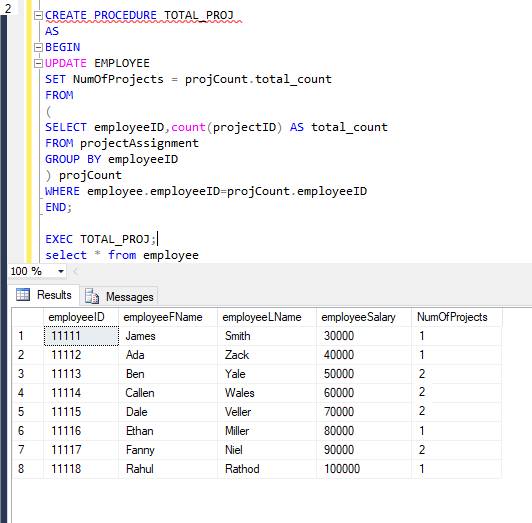
WHERE employee.employeeID=projCount.employeeID

END;

EXEC TOTAL\_PROJ;

**Result:**

****

****

1. Create a trigger that can update the num of projects whenever a new project is assigned to an employee.

Test the trigger with the below insert

* 1. INSERT INTO projectAssignment VALUES (11114, 1003, ‘quality assurance');
  2. INSERT INTO projectAssignment VALUES (11115, 1003,'test engineer');

**Query:**

CREATE TRIGGER NumofProjectsTrigger

ON projectAssignment

FOR INSERT,UPDATE

AS

IF @@ROWCOUNT >= 1

BEGIN

UPDATE EMPLOYEE

SET NumOfProjects = projCount.total\_count

FROM

(

SELECT employeeID,count(projectID) AS total\_count

FROM projectAssignment

GROUP BY employeeID

) projCount

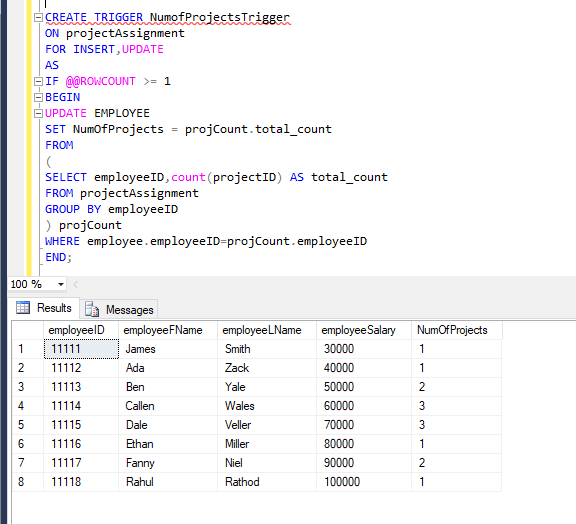
WHERE employee.employeeID=projCount.employeeID

END;

INSERT INTO projectAssignment VALUES (11114, 1003, 'quality assurance');

INSERT INTO projectAssignment VALUES (11115, 1003,'test engineer');

**Result:**

****