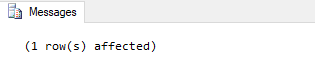
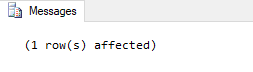
**Lab 7 Exercise**

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

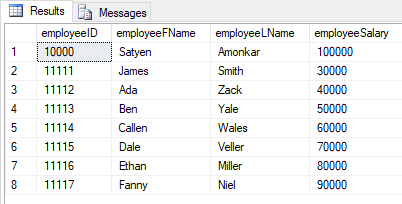
INSERT INTO employee VALUES (10000, 'Satyen', 'Amonkar', 100000);

`

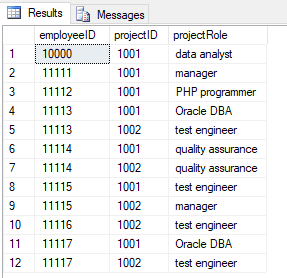
INSERT INTO projectAssignment VALUES (10000, 1001, 'data analyst');



SELECT \* FROM employee;



SELECT \* FROM projectAssignment



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

CREATE FUNCTION avg\_sal()

RETURNS DECIMAL(10,1)

AS

BEGIN

DECLARE @ret DECIMAL(10,1);

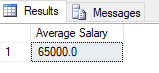
SELECT @ret= AVG(e.employeeSalary) FROM employee e

RETURN @ret;

END;



SELECT dbo.avg\_sal() AS 'Average Salary';



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

CREATE FUNCTION projects(@EmployeeID INT)

RETURNS TABLE

AS

RETURN(

SELECT e.employeeID,e.employeeFName,e.employeeLName,p.\*

FROM employee e inner join projectAssignment pr

ON e.employeeID = pr.employeeID

INNER JOIN project p

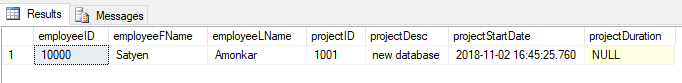
ON pr.projectID = p.projectID

WHERE e.employeeID = @EmployeeID

)



SELECT \* FROM projects(10000);



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)

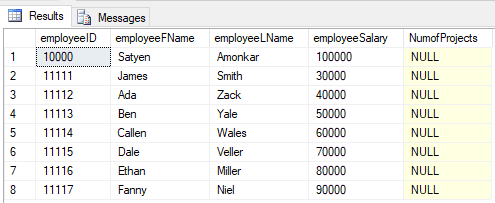
ALTER TABLE employee

ADD "NumofProjects" INT;



**Query before executing Procedure**

select \* from employee;



CREATE PROCEDURE projectUpdate

AS

BEGIN

Update employee

SET NumofProjects = temp.numOfprj

FROM

(SELECT e.employeeID, e.employeeFName, COUNT(pr.projectID) 'numOfprj'

FROM employee e INNER JOIN projectAssignment pr

ON e.employeeID = pr.employeeID

GROUP BY e.employeeID, e.employeeFName

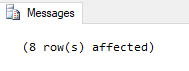
) AS temp

WHERE employee.employeeID = temp.employeeID

END;

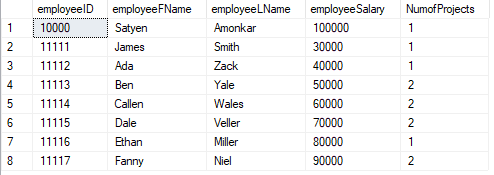


EXECUTE projectUpdate



**Query after executing Procedure**

SELECT \* FROM employee;



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

CREATE TRIGGER projectassignmentTrigger

ON projectAssignment

FOR INSERT

AS

IF @@ROWCOUNT >= 1

BEGIN

UPDATE employee

SET NumofProjects = temp.numOfprj

FROM

(SELECT e.employeeID, e.employeeFName, COUNT(pr.projectID) 'numOfprj'

FROM employee e INNER JOIN projectAssignment pr

ON e.employeeID = pr.employeeID

GROUP BY e.employeeID, e.employeeFName

) AS temp

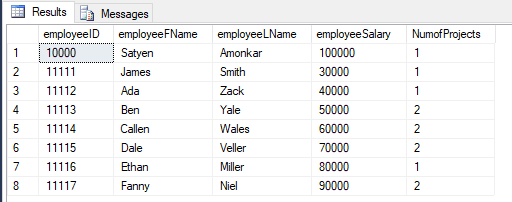
WHERE employee.employeeID = temp.employeeID

END;

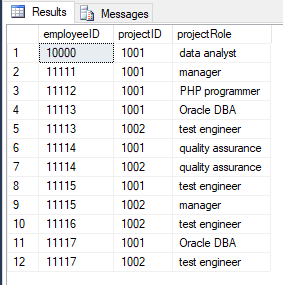


**Query before inserting**

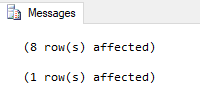
SELECT \* FROM employee;



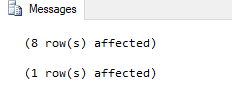
SELECT \* FROM projectAssignment;



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

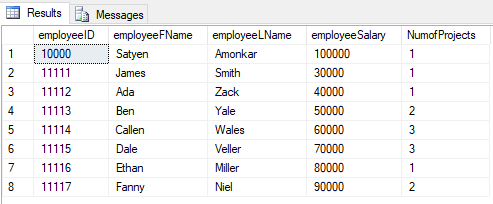


INSERT INTO projectAssignment VALUES (11114, 1003,'quality assurance');



**Query after inserting**

SELECT \* FROM employee e;



SELECT \* FROM projectAssignment pr;

