

## LAB ASSIGNMENT 8 – Due Sunday Noon

### ANSWER THE FOLLOWING QUESTIONS

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

```

/*1.Add an employee using your own name and create a project assignment for your self using existing project id.*/
INSERT INTO employee VALUES (88888, 'Pan', 'Chen', 10000);
INSERT INTO projectAssignment VALUES (88888, 1001, 'slave');
SELECT * FROM EMPLOYEE;
SELECT * FROM projectAssignment;

/*2.Write a scalar function that returns the average salary of the Employees*/
CREATE FUNCTION average_salary()
RETURNS DECIMAL(20,2)
AS BEGIN

```

Results

3	11113	Ben	Yale	50000
4	11114	Callen	Wales	60000
5	11115	Dale	Veller	70000
6	11116	Ethan	Miller	80000
7	11117	Fanny	Niel	90000
8	88888	Pan	Chen	10000

  

employeeID	projectID	projectRole	
5	11114	1001	quality assuran...
6	11114	1002	quality assuran...
7	11115	1001	test engineer
8	11115	1002	manager
9	11116	1002	test engineer
10	11117	1001	Oracle DBA
11	11117	1002	test engineer
12	88888	1001	slave

Query executed successfully. | ist-s-students.syr.edu (12.... | AD\pachen (78) | pachen659 | 00:00:00 | 20 rows

- 2) Write a scalar function that returns the average salary of the Employee

```

/*2.Write a scalar function that returns the average salary of the Employees*/
CREATE FUNCTION average_salary()
RETURNS DECIMAL(20,2)
AS BEGIN
    DECLARE @avg INT;
    SELECT @avg=AVG(employeeSalary)
    FROM employee
    RETURN @avg;
END;
/*call the scalar function*/
SELECT dbo.average_salary() as 'average';

/*3) Write a table-valued function that returns the Projects given an EmployeeID as a

```

Results

	average
1	53750.00

- 3) Write a table-valued function that returns the Projects given an EmployeeID as a parameter and

- Show the function created
- return the results for your own project

```

/*3) Write a table-valued function that returns the Projects given an EmployeeID as a parameter and
a. Show the function created
b. return the results for your own project*/
CREATE FUNCTION projecttable(@EmployeeID INT)
RETURNS TABLE
AS
RETURN
(SELECT t1.EmployeeID, t2.projectID, employeeFName, employeeLName
ProjectDesc, ProjectStartDate, ProjectDuration, ProjectRole FROM PROJECTASSIGNMENT T1
INNER JOIN project t2 ON t1.projectID = t2.projectID
INNER JOIN employee t3 ON t1.employeeID = t3.employeeID
WHERE t1.EmployeeID=@EmployeeID);

/*call the table-valued function*/
SELECT * FROM dbo.projecttable(88888);

```

- 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 NumProjects Int;
SELECT * FROM EMPLOYEE;
SELECT * FROM PROJECTASSIGNMENT;
CREATE PROCEDURE totalprojects
AS
BEGIN
    UPDATE EMPLOYEE
    SET NumProjects = projectcount.count1
    FROM
    (SELECT EMPLOYEEID, count(projectID) 'count1' FROM projectAssignment
    GROUP BY EMPLOYEEID) AS projectcount
    WHERE EMPLOYEE.EMPLOYEEID = projectcount.employeeID
END;
EXEC totalprojects;
DROP PROCEDURE totalprojects

```

/\*5) Create a trigger that can update the num of projects whenever a new project is assi

100 % <

Results Messages

	employeeID	employeeFName	employeeLName	employeeSalary	NumProjects
1	11111	James	Smith	30000	NULL
2	11112	Ada	Zack	40000	NULL
3	11113	Ben	Yale	50000	NULL
4	11114	Callen	Wales	60000	NULL
5	11115	Dale	Veller	70000	NULL
6	11116	Ethan	Miller	80000	NULL
7	11117	Fanny	Niel	90000	NULL
8	88888	Pan	Chen	10000	NULL

```

ALTER TABLE EMPLOYEE ADD NumProjects Int;
SELECT * FROM EMPLOYEE;
SELECT * FROM PROJECTASSIGNMENT;
CREATE PROCEDURE totalprojects
AS
BEGIN
    UPDATE EMPLOYEE
    SET NumProjects = projectcount.count1
    FROM
    (SELECT EMPLOYEEID, count(projectID) 'count1' FROM projectAssignment
    GROUP BY EMPLOYEEID) AS projectcount
    WHERE EMPLOYEE.EMPLOYEEID = projectcount.employeeID
END;
EXEC totalprojects;
DROP PROCEDURE totalprojects
/*5) Create a trigger that can update the num of projects whenever a new pro

```

100 % <

Results Messages

	employeeID	employeeFName	employeeLName	employeeSalary	NumProjects
1	11111	James	Smith	30000	1
2	11112	Ada	Zack	40000	1
3	11113	Ben	Yale	50000	2
4	11114	Callen	Wales	60000	2
5	11115	Dale	Veller	70000	2
6	11116	Ethan	Miller	80000	1
7	11117	Fanny	Niel	90000	2
8	88888	Pan	Chen	10000	1

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

```

/*5) Create a trigger that can update the num of projects whenever a new project is assigned to an employee.
a. Test the trigger with the below insert
INSERT INTO projectAssignment VALUES (11114, 1003, 'quality assurance');
INSERT INTO projectAssignment VALUES (11115, 1003, 'test engineer');
*/
CREATE TRIGGER updateprojectnumtrigger
ON ProjectAssignment
FOR INSERT, UPDATE
AS
IF @@ROWCount>=1
BEGIN
UPDATE EMPLOYEE
SET NumProjects = projectcount.count1
FROM
(SELECT EMPLOYEEID, count(projectID) 'count1' FROM projectAssignment
GROUP BY EMPLOYEEID) AS projectcount
WHERE EMPLOYEE.EMPLOYEEID = projectcount.employeeID
END;
SELECT * FROM EMPLOYEE

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

```

Results					
	employeeID	employeeFName	employeeLName	employeeSalary	NumProjects
1	11111	James	Smith	30000	1
2	11112	Ada	Zack	40000	1
3	11113	Ben	Yale	50000	2
4	11114	Callen	Wales	60000	2
5	11115	Dale	Veller	70000	2
6	11116	Ethan	Miller	80000	1
7	11117	Fanny	Niel	90000	2
8	88888	Pan	Chen	10000	1

a. Test the trigger with the below insert

```

INSERT INTO projectAssignment VALUES (11114, 1003, 'quality
assurance');
INSERT INTO projectAssignment VALUES (11115, 1003, 'test
engineer');

```

```

/*
CREATE TRIGGER updateprojectnumtrigger
ON ProjectAssignment
FOR INSERT, UPDATE
AS
IF @@ROWCount>=1
BEGIN
    UPDATE EMPLOYEE
    SET NumProjects = projectcount.count1
    FROM
    (SELECT EMPLOYEEID, count(projectID) 'count1' FROM projectAssignment
    GROUP BY EMPLOYEEID) AS projectcount
    WHERE EMPLOYEE.EMPLOYEEID = projectcount.employeeID
END;

SELECT * FROM EMPLOYEE

INSERT INTO projectAssignment VALUES (11114, 1003, 'quality assurance');
INSERT INTO projectAssignment VALUES (11115, 1003, 'test engineer');

```

83 %

Results Messages

	employeeID	employeeFName	employeeLName	employeeSalary	NumProjects
1	11111	James	Smith	30000	1
2	11112	Ada	Zack	40000	1
3	11113	Ben	Yale	50000	2
4	11114	Callen	Wales	60000	3
5	11115	Dale	Veller	70000	3
6	11116	Ethan	Miller	80000	1
7	11117	Fanny	Niel	90000	2
8	88888	Pan	Chen	10000	1

## INSTRUCTIONS

- For each question, ensure that the screenshot of the function and procedure scripts and return value is also pasted in the document
- Make sure you paste the before and after screenshots for question 4. That is, the values for 'No of Projects' in Employee before and after execution of the procedure

*Please submit your lab report in one PDF file to BlackBoard. Name your file in this format "IST659-Lab8-Lastname-Firstname.pdf".*

