WEEK 6 ASSESMENT

1. Create two tables, employees and sales. Get a list of all employees who did not make any sales.

> CREATE TABLE employees (id INT, name VARCHAR(255), PRIMARY KEY (id));

CREATE TABLE sales ( employee\_id INT, amount INT, FOREIGN KEY (employee\_id) REFERENCES employees(id));

INSERT INTO employees (id, name) VALUES (1, 'John');

INSERT INTO employees (id, name) VALUES (2, 'Mary');

INSERT INTO employees (id, name) VALUES (3, 'Peter');

INSERT INTO employees (id, name) VALUES (4, 'Jane');

INSERT INTO sales ( employee\_id, amount) VALUES (1, 100);

INSERT INTO sales ( employee\_id, amount) VALUES (2, 200);

INSERT INTO sales ( employee\_id, amount) VALUES ( 3, 300);

INSERT INTO sales ( employee\_id, amount) VALUES ( 4, 0);

SELECT \* FROM employees WHERE id NOT IN (SELECT employee\_id FROM sales WHERE amount > 0);

3.Write one procedure that can insert or update the employee (avoid using if statement to check the statement e.g., if (statement ==’Insert))

> UPDATE employees SET name = 'Joy' WHERE id = 1;

5.Write an SQL query to fetch only odd rows from the table (create dummy data to use)

> SELECT \* FROM employees WHERE id % 2 = 1;

2. Assuming you have Customers table; with columns CustomerID, CustomerName, ContactName, Address, City, PostalCode and Country. Write a query to list the number of customers in each country; only include countries with more than 3 customers , use ORDER BY too.

SELECT Country, COUNT(\*) AS NumberOfCustomers FROM Customers GROUP BY Country HAVING COUNT(\*) > 3 ORDER BY COUNT(\*) DESC;

6.Write a function that can calculate age given a certain date of birth

>

CREATE FUNCTION dbo.Age(@DOB DATETIME)

RETURNS INT

AS

BEGIN

DECLARE @TODAY DATETIME

DECLARE @AGE INT

SET @TODAY = GETDATE()

SET @AGE = DATEDIFF(YEAR, @DOB, @TODAY)

RETURN @AGE

END

4.Write an SQL query to fetch duplicate records from EmployeeDetails (without considering the primary key – EmpId)(create dummy data to use)

CREATE TABLE EmployeeDetails (EmpId INT, EmpName VARCHAR(50), EmpAddress VARCHAR(50), EmpSalary INT);

INSERT INTO EmployeeDetails VALUES (1, 'John', 'USA', 1000);

INSERT INTO EmployeeDetails VALUES (2, 'John', 'USA', 1000);

INSERT INTO EmployeeDetails VALUES (3, 'John', 'USA', 1000);

INSERT INTO EmployeeDetails VALUES (4, 'John', 'USA', 1000);

INSERT INTO EmployeeDetails VALUES (4, 'JoY', 'USA', 1000);

SELECT EmpName, EmpAddress, EmpSalary

FROM EmployeeDetails

GROUP BY EmpName, EmpAddress, EmpSalary

HAVING COUNT(\*) > 1;