Week 7 ASSESMENT

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

CREATE TABLE employees2 (

id INT PRIMARY KEY, name VARCHAR(50),

);

INSERT INTO employees2 VALUES

(1, 'John' ), (2, 'Jane'), (3, 'Bob' ), (4, 'Mary' ), (5, 'Mike' ), (6, 'May' )

CREATE TABLE sales2 (

id INT PRIMARY KEY,employee\_id INT,amount DECIMAL(10,2),

date DATE

);

INSERT INTO sales2 VALUES

(1, 1, 1000.00, '2022-01-01'),(2, 1, 500.00, '2022-01-05'),(3, 4, 750.00, '2022-02-01'),

(4, 5, 100.00, '2022-02-15'),(6, 34, 0.00, '2022-02-15');

SELECT employees2.name

FROM employees2

LEFT JOIN sales2 ON employees2.id = sales2.employee\_id

WHERE sales2.amount IS NULL OR sales2.amount = 0;

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

CREATE TABLE test (id INT PRIMARY KEY,name VARCHAR(50),age INT

);

INSERT INTO test VALUES

(1, 'John' , 20),(2, 'Jane', 21),(3, 'Bob' , 22),(4, 'Mary' , 23),(5, 'Mike' , 24),

(6, 'May' , 25);

SELECT \* FROM test 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(CustomerID) AS NumCustomers

FROM Customers

GROUP BY Country

HAVING COUNT(CustomerID) > 3

ORDER BY NumCustomers DESC;

4.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

SELECT dbo.Age('2000-01-01')

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

SELECT EmpName, EmpAddress, EmpSalary

FROM EmployeeDetails

GROUP BY EmpName, EmpAddress, EmpSalary

HAVING COUNT(\*) > 1;

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

CREATE PROCEDURE InsertOrUpdateEmployee

  @EmpId INT,

  @EmpName VARCHAR(50),

  @EmpEmail VARCHAR(50),

  @EmpPhone VARCHAR(20),

  @EmpAddress VARCHAR(100)

AS

BEGIN

  MERGE INTO Employee AS target

  USING (VALUES (@EmpId, @EmpName, @EmpEmail, @EmpPhone, @EmpAddress))

    AS source (EmpId, EmpName, EmpEmail, EmpPhone, EmpAddress)

    ON target.EmpId = source.EmpId

  WHEN MATCHED THEN

    UPDATE SET

      EmpName = source.EmpName,

      EmpEmail = source.EmpEmail,

      EmpPhone = source.EmpPhone,

      EmpAddress = source.EmpAddress

  WHEN NOT MATCHED THEN

    INSERT (EmpId, EmpName, EmpEmail, EmpPhone, EmpAddress)

    VALUES (source.EmpId, source.EmpName, source.EmpEmail, source.EmpPhone, source.EmpAddress);

END;