** Varendra University**

*Department of*

*Computer Science & Engineering*

**Course Code:** CSE 313

**Course Title: Database Management System**

**Date of Submission:** 22/11/24

**Assignment - 2**

|  |  |  |
| --- | --- | --- |
| **Submitted By . . .**  **Name: Md. Sajidul Islam**  **ID:** 201311124  **Semester:** 6th  **Section:** C |  | **Submitted To . . .**  **Md. Toufikul Islam**  **Lecturer**  **Dept. of CSE, VU** |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature

CREATE DATABASE assignment;

USE assignment;

CREATE TABLE Worker (

WORKER\_ID INT PRIMARY KEY,

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

SALARY INT,

JOINING\_DATE DATE,

DEPARTMENT VARCHAR(50)

);

CREATE TABLE Bonus (

WORKER\_REF\_ID INT,

BONUS\_DATE DATE,

BONUS\_AMOUNT INT,

FOREIGN KEY (WORKER\_REF\_ID) REFERENCES Worker(WORKER\_ID)

);

CREATE TABLE Title (

WORKER\_REF\_ID INT,

WORKER\_TITLE VARCHAR(50),

AFFECTED\_FROM DATE,

FOREIGN KEY (WORKER\_REF\_ID) REFERENCES Worker(WORKER\_ID)

);

INSERT INTO Worker (WORKER\_ID, FIRST\_NAME, LAST\_NAME, SALARY, JOINING\_DATE, DEPARTMENT)

VALUES

(1, 'Rana', 'Hamid', 100000, '2014-02-20', 'HR'),

(2, 'Sanjoy', 'Saha', 80000, '2014-06-11', 'Admin'),

(3, 'Mahmudul', 'Hasan', 300000, '2014-02-20', 'HR'),

(4, 'Asad', 'Zaman', 500000, '2014-02-20', 'Admin'),

(5, 'Sajib', 'Mia', 500000, '2014-06-11', 'Admin'),

(6, 'Alamgir', 'Kabir', 200000, '2014-06-11', 'Account'),

(7, 'Foridul', 'Islam', 75000, '2014-01-20', 'Account'),

(8, 'Keshob', 'Ray', 90000, '2014-04-11', 'Admin');

INSERT INTO Bonus (WORKER\_REF\_ID, BONUS\_DATE, BONUS\_AMOUNT)

VALUES

(1, '2019-02-20', 5000),

(2, '2019-06-11', 3000),

(3, '2019-02-20', 4000),

(4, '2019-02-20', 4500),

(5, '2019-06-11', NULL),

(6, '2019-06-12', 3500);

INSERT INTO Title (WORKER\_REF\_ID, WORKER\_TITLE, AFFECTED\_FROM)

VALUES

(1, 'Manager', '2019-02-20'),

(2, 'Executive', '2019-06-11'),

(8, 'Executive', '2019-06-11'),

(5, 'Manager', '2019-06-11'),

(4, 'Asst. Manager', '2019-06-11'),

(7, 'Executive', '2019-06-11'),

(6, 'Lead', '2019-06-11'),

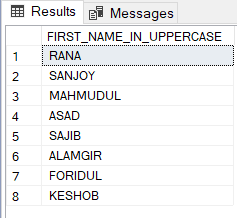
(3, 'Lead', '2019-06-11');

1. Write an sql query to fetch "FIRST\_NAME" from Worker table in upper case

SQL:

SELECT UPPER(FIRST\_NAME) AS FIRST\_NAME\_IN\_UPPERCASE

FROM Worker;

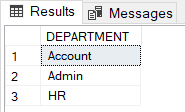
Output:

2. Write an SQL query to fetch unique values of DEPERTMENT from Worker table.

SQL:

SELECT DISTINCT DEPARTMENT

FROM Worker;

Output:

3. Write an SQL query to find the position of the alphabet('a') in the first name column 'Sanjoy' from Worker table.

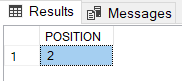
SQL:

SELECT CHARINDEX('a', FIRST\_NAME) AS POSITION

FROM Worker

WHERE FIRST\_NAME = 'Sanjoy';

Output:



4. Write an SQL query to print details of the workers from Workers table whose FIRST\_NAME ends with 'b' and contains at least three alphabet.

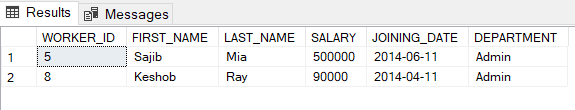
SQL:

SELECT \*

FROM Worker

WHERE FIRST\_NAME LIKE '%b' AND LEN(FIRST\_NAME) >= 3;

Output:

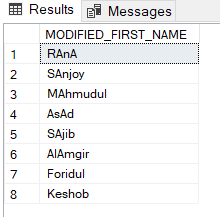


5. Write an SQL query to print the FIRST\_NAME from Worker table after replacing 'a' with 'A’ .

SQL:

SELECT REPLACE(FIRST\_NAME, 'a', 'A') AS MODIFIED\_FIRST\_NAME

FROM Worker;

Output:

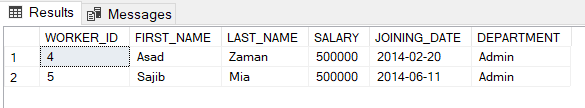
6. Write an SQL query to print details for Workers with the first name as “Asad” and “Sajib” from Worker table.

SQL:

SELECT \*

FROM Worker

WHERE FIRST\_NAME IN ('Asad', 'Sajib');

Output:

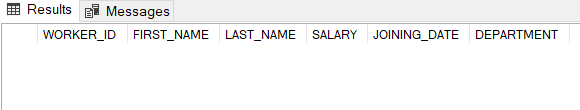
7. Write an SQL query to print details of the Workers who have joined 6 months ago.

SQL:

SELECT \*

FROM Worker

WHERE JOINING\_DATE BETWEEN DATEADD(MONTH, -6, GETDATE()) AND GETDATE();

Output:

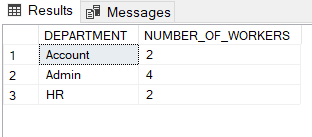
8. Write an SQL query to show all departments along with the number of people in there.

SQL:

SELECT DEPARTMENT, COUNT(\*) AS NUMBER\_OF\_WORKERS

FROM Worker

GROUP BY DEPARTMENT;

Output:

9. Write an SQL query to fetch the departments that have less than five people in it.

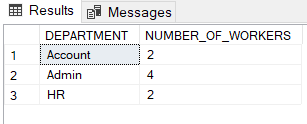
SQL:

SELECT DEPARTMENT, COUNT(\*) AS NUMBER\_OF\_WORKERS

FROM Worker

GROUP BY DEPARTMENT

HAVING COUNT(\*) < 5;

Output:

10. Write an SQL query to print details of the Workers who are also Managers.

SQL:

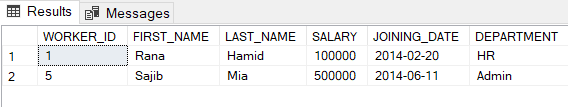
SELECT W.\*

FROM Worker W

JOIN Title T

ON W.WORKER\_ID = T.WORKER\_REF\_ID

WHERE T.WORKER\_TITLE = 'Manager';

Output:

11. List all the employees except ‘Manager’ & ‘Asst. Manager’.

SQL:

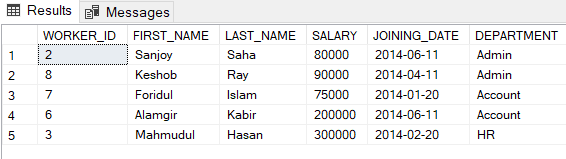
SELECT W.\*

FROM Worker W

JOIN Title T

ON W.WORKER\_ID = T.WORKER\_REF\_ID

WHERE T.WORKER\_TITLE NOT IN ('Manager', 'Asst. Manager');

Output:

12. Write an SQL query to determine the nth (say n=5) highest salary from a table

SQL:

SELECT MIN(SALARY) AS Nth\_Highest\_Salary

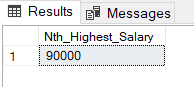
FROM (

SELECT DISTINCT TOP 5 SALARY

FROM Worker

ORDER BY SALARY DESC

) AS TopSalaries;

Output:

13. Write an SQL query to fetch the last five records from a table.

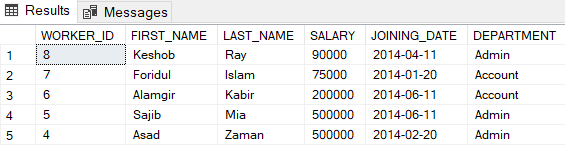
SQL:

SELECT \*

FROM Worker

ORDER BY WORKER\_ID DESC

OFFSET 0 ROWS FETCH NEXT 5 ROWS ONLY;

Output:

14. Write an SQL query to print the name of employees having the highest salary in each department.

SQL:

SELECT FIRST\_NAME, LAST\_NAME, DEPARTMENT, SALARY

FROM Worker W

WHERE SALARY = (

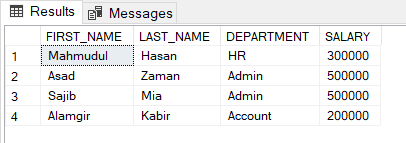
SELECT MAX(SALARY)

FROM Worker

WHERE DEPARTMENT = W.DEPARTMENT

);

Output:



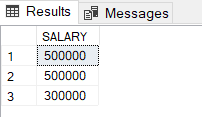
15. Write an SQL query to fetch three max salaries from table.

SQL:

SELECT TOP 3 SALARY

FROM Worker

ORDER BY SALARY DESC;

Output:

CREATE TABLE Account\_Detail (

Account\_no INT PRIMARY KEY,

Acc\_holder\_name VARCHAR(50),

Amount INT,

Branch\_Id VARCHAR(10),

Zone\_Id VARCHAR(10)

);

INSERT INTO Account\_Detail (Account\_no, Acc\_holder\_name, Amount, Branch\_Id, Zone\_Id)

VALUES

(1992212, 'Mr. Nazmuzzaman', 200000, 'B-101', 'Z-803'),

(1992213, 'Mr. Jibon', 170000, 'B-102', 'Z-803'),

(1882212, 'Bushra', 180000, 'B-103', 'Z-802'),

(1882213, '%Sajib', 170000, 'B-104', 'Z-801');

CREATE TABLE Branch (

Br\_Id VARCHAR(10) PRIMARY KEY,

Branch\_Name VARCHAR(50)

);

INSERT INTO Branch (Br\_Id, Branch\_Name)

VALUES

('B-101', 'Bonani'),

('B-102', 'Romna'),

('B-103', 'Shaheb bazar'),

('B-104', 'Ullapara');

CREATE TABLE Zone (

Zone\_Id VARCHAR(10) PRIMARY KEY,

Name VARCHAR(50)

);

INSERT INTO Zone (Zone\_Id, Name)

VALUES

('Z-801', 'Sirajgonj'),

('Z-802', 'Rajshahi'),

('Z-803', 'Dhaka'),

('Z-804', 'Chittagong');

1.Create a simple stored procedure “SPdetails” to find Acc\_holder\_name, Amount, Branch\_Name and Zone\_Name.

SQL:

CREATE PROCEDURE SPdetails

AS

BEGIN

SELECT

a.Acc\_holder\_name,

a.Amount,

b.Branch\_Name,

z.Name AS Zone\_Name

FROM

Account\_Detail a

INNER JOIN

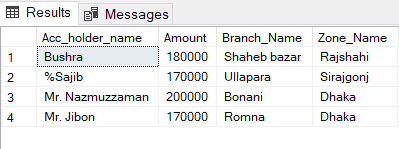
Branch b ON a.Branch\_Id = b.Br\_Id

INNER JOIN

Zone z ON a.Zone\_Id = z.Zone\_Id;

END;

EXEC SPdetails;

Output:

2. Create a simple stored procedure “SPaverage” to Branch \_name and Amount of Branch.

SQL:

CREATE PROCEDURE SPaverage

AS

BEGIN

SELECT

b.Branch\_Name,

AVG(a.Amount) AS Average\_Amount

FROM

Account\_Detail a

INNER JOIN

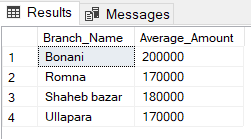
Branch b ON a.Branch\_Id = b.Br\_Id

GROUP BY

b.Branch\_Name;

END;

EXEC SPaverage;

Output:

3. Create a simple stored procedure “SPbalance” to find Amount of each Account\_no.

SQL:

CREATE PROCEDURE SPbalance

AS

BEGIN

SELECT

Account\_no,

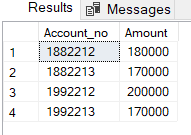
Amount

FROM

Account\_Detail;

END;

EXEC SPbalance;

Output:

4. Create a simple stored procedure “SPamount” to Find all account holders name with their branch name and zone name whose name has substring ‘Mr.’ and Amount Less than Maximum Amount.

SQL:

CREATE PROCEDURE SPamount

AS

BEGIN

-- Find the maximum amount

DECLARE @MaxAmount INT;

SELECT @MaxAmount = MAX(Amount) FROM Account\_Detail;

-- Fetch account holders matching the criteria

SELECT

a.Acc\_holder\_name,

b.Branch\_Name,

z.Name AS Zone\_Name,

a.Amount

FROM

Account\_Detail a

INNER JOIN

Branch b ON a.Branch\_Id = b.Br\_Id

INNER JOIN

Zone z ON a.Zone\_Id = z.Zone\_Id

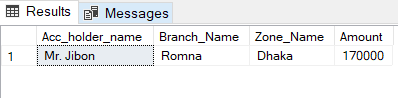
WHERE

a.Acc\_holder\_name LIKE '%Mr.%' -- Name contains 'Mr.'

AND a.Amount < @MaxAmount; -- Amount is less than the maximum

END;

EXEC SPamount;

Output:

5. Create a simple stored procedure “SPdetailsInfo” to find Zone\_name, number of customer of each Zone.

SQL:

CREATE PROCEDURE SPdetailsInfo

AS

BEGIN

SELECT

z.Name AS Zone\_Name,

COUNT(a.Account\_no) AS Number\_of\_Customers

FROM

Account\_Detail a

INNER JOIN

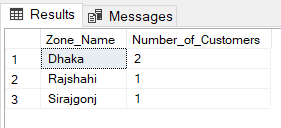
Zone z ON a.Zone\_Id = z.Zone\_Id

GROUP BY

z.Name;

END;

EXEC SPdetailsInfo;

Output: