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

Signature

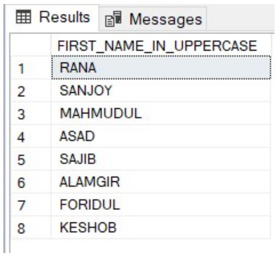
```sql
CREATE DATABASE assignment;
USE assignment;

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:



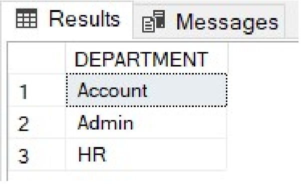
	FIRST_NAME_IN_UPPERCASE
1	RANA
2	SANJOY
3	MAHMUDUL
4	ASAD
5	SAJIB
6	ALAMGIR
7	FORIDUL
8	KESHOB

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

SQL:

```
SELECT DISTINCT DEPARTMENT
    FROM Worker;
```

Output:



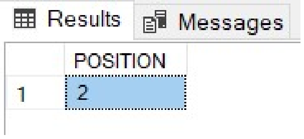
	DEPARTMENT
1	Account
2	Admin
3	HR

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:



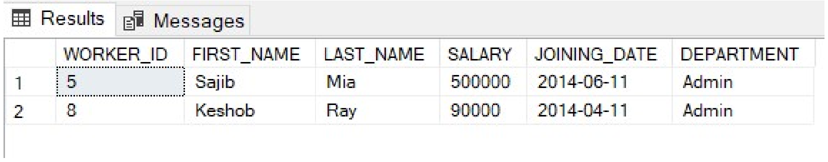
	POSITION
1	2

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:



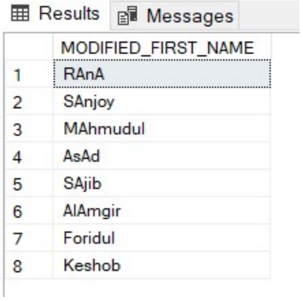
	WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	5	Sajib	Mia	500000	2014-06-11	Admin
2	8	Keshob	Ray	90000	2014-04-11	Admin

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:



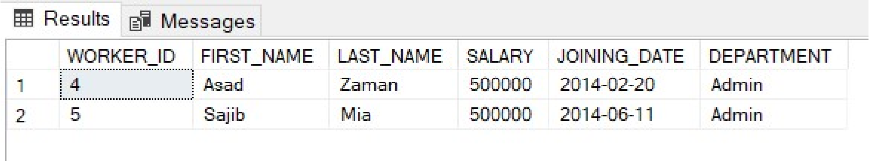
	MODIFIED_FIRST_NAME
1	RAnA
2	SAnjoy
3	MAhmudul
4	AsAd
5	SAjib
6	AlAmgir
7	Foridul
8	Keshob

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:



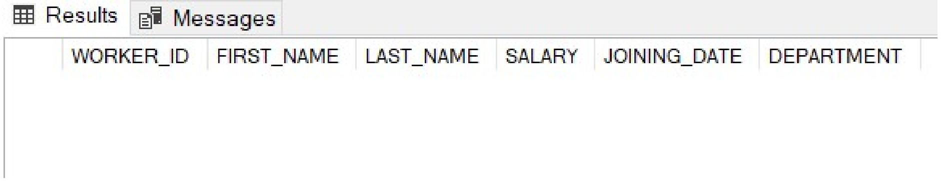
	WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	4	Asad	Zaman	500000	2014-02-20	Admin
2	5	Sajib	Mia	500000	2014-06-11	Admin

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:



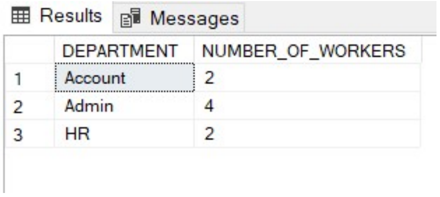
	WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT

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:



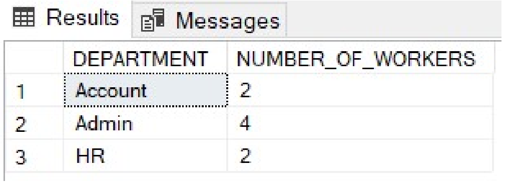
	DEPARTMENT	NUMBER_OF_WORKERS
1	Account	2
2	Admin	4
3	HR	2

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:



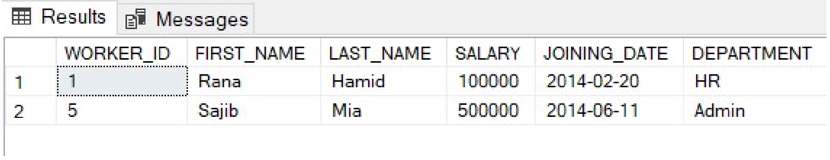
	DEPARTMENT	NUMBER_OF_WORKERS
1	Account	2
2	Admin	4
3	HR	2

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:



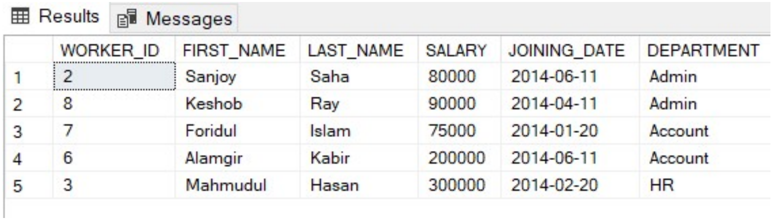
	WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	1	Rana	Hamid	100000	2014-02-20	HR
2	5	Sajib	Mia	500000	2014-06-11	Admin

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:



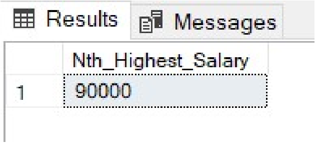
	WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	2	Sanjoy	Saha	80000	2014-06-11	Admin
2	8	Keshob	Ray	90000	2014-04-11	Admin
3	7	Foridul	Islam	75000	2014-01-20	Account
4	6	Alamgir	Kabir	200000	2014-06-11	Account
5	3	Mahmudul	Hasan	300000	2014-02-20	HR

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:



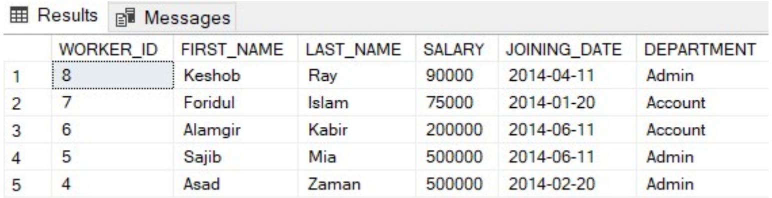
	Nth_Highest_Salary
1	90000

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:



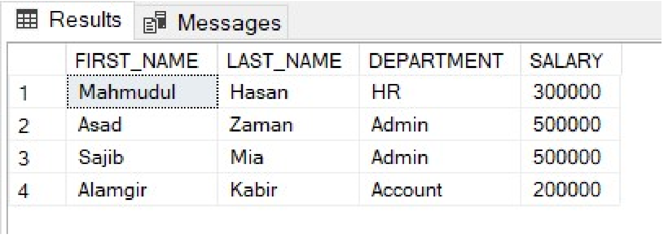
	WORKER_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	8	Keshob	Ray	90000	2014-04-11	Admin
2	7	Foridul	Islam	75000	2014-01-20	Account
3	6	Alamgir	Kabir	200000	2014-06-11	Account
4	5	Sajib	Mia	500000	2014-06-11	Admin
5	4	Asad	Zaman	500000	2014-02-20	Admin

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:



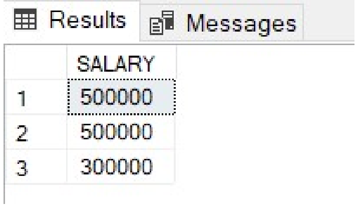
	FIRST_NAME	LAST_NAME	DEPARTMENT	SALARY
1	Mahmudul	Hasan	HR	300000
2	Asad	Zaman	Admin	500000
3	Sajib	Mia	Admin	500000
4	Alamgir	Kabir	Account	200000

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

SQL:

```
SELECT TOP 3 SALARY
FROM Worker
ORDER BY SALARY DESC;
```

Output:



	SALARY
1	500000
2	500000
3	300000

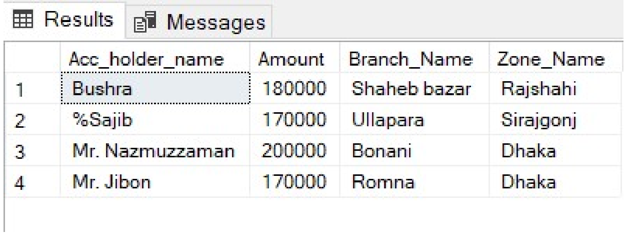
```
CREATE TABLE Account_Detail (
    Account_no INT PRIMARY KEY,
    Acc_holder_name VARCHAR(50),
    Amount INT,
    Branch_Id VARCHAR(10),
    Zone_Id VARCHAR(10)
);

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:



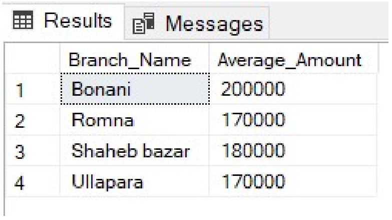
	Acc_holder_name	Amount	Branch_Name	Zone_Name
1	Bushra	180000	Shaheb bazar	Rajshahi
2	%Sajib	170000	Ullapara	Sirajgonj
3	Mr. Nazmuzzaman	200000	Bonani	Dhaka
4	Mr. Jibon	170000	Romna	Dhaka

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:



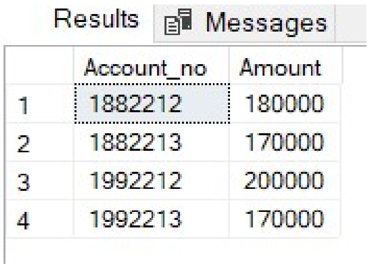
	Branch_Name	Average_Amount
1	Bonani	200000
2	Romna	170000
3	Shaheb bazar	180000
4	Ullapara	170000

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:



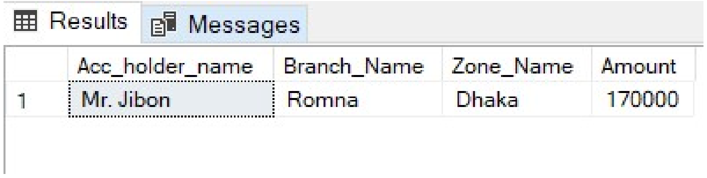
	Account_no	Amount
1	1882212	180000
2	1882213	170000
3	1992212	200000
4	1992213	170000

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:



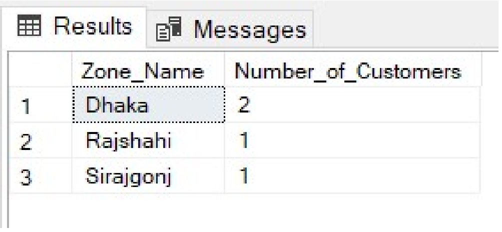
	Acc_holder_name	Branch_Name	Zone_Name	Amount
1	Mr. Jibon	Romna	Dhaka	170000

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:



	Zone_Name	Number_of_Customers
1	Dhaka	2
2	Rajshahi	1
3	Sirajgonj	1