

# MYSQL CODING CHALLENGE

DAY:02/06/2025

SUBMITTED BY: YUVASHRI. S

## QUESTIONS

1.Consider below EMPLOYEE table as the source data

CREATE TABLE

```
EMPLOYEE ( EMPLOYEE_ID INT, NAME VARCHAR2(20), SALARY DECIMAL(8,2) );  
  
INSERT INTO EMPLOYEE(EMPLOYEE_ID,NAME,SALARY) VALUES(100,'Jennifer',4400);  
INSERT INTO EMPLOYEE(EMPLOYEE_ID,NAME,SALARY) VALUES(100,'Jennifer',4400);  
INSERT INTO EMPLOYEE(EMPLOYEE_ID,NAME,SALARY) VALUES(101,'Michael',13000);  
INSERT INTO EMPLOYEE(EMPLOYEE_ID,NAME,SALARY) VALUES(101,'Michael',13000);  
INSERT INTO EMPLOYEE(EMPLOYEE_ID,NAME,SALARY) VALUES(101,'Michael',13000);  
INSERT INTO EMPLOYEE(EMPLOYEE_ID,NAME,SALARY) VALUES(102,'Pat',6000);  
INSERT INTO EMPLOYEE(EMPLOYEE_ID,NAME,SALARY) VALUES(102,'Pat',6000);  
INSERT INTO EMPLOYEE(EMPLOYEE_ID,NAME,SALARY) VALUES(103,'Den',11000);
```

i) How to delete DUPLICATE records from a table using a SQL Query Using ROWID and Correlated subquery?

The screenshot displays the MySQL Workbench interface. The SQL Editor contains the following query:

```
/* 1. A) How to delete DUPLICATE records from a table using a SQL Query Using ROWID and  
Correlated subquery?*/  
  
ALTER TABLE EMPLOYEE ADD COLUMN id INT AUTO_INCREMENT PRIMARY KEY;  
  
DELETE e1 FROM EMPLOYEE e1  
INNER JOIN EMPLOYEE e2  
WHERE  
e1.EMPLOYEE_ID = e2.EMPLOYEE_ID  
AND e1.NAME = e2.NAME  
AND e1.SALARY = e2.SALARY  
AND e1.id > e2.id;
```

The Output tab shows the execution results:

#	Time	Action	Message	Duration / Fetch
124	12:03:09	INSERT INTO EMPLOYEE(EMPLOYEE_ID, NAME, SALARY) VALUES (101, 'Michael', 13000);	1 row(s) affected	0.000 sec
125	12:03:09	INSERT INTO EMPLOYEE(EMPLOYEE_ID, NAME, SALARY) VALUES (101, 'Michael', 13000);	1 row(s) affected	0.000 sec
126	12:03:09	INSERT INTO EMPLOYEE(EMPLOYEE_ID, NAME, SALARY) VALUES (102, 'Pat', 6000);	1 row(s) affected	0.000 sec
127	12:03:09	INSERT INTO EMPLOYEE(EMPLOYEE_ID, NAME, SALARY) VALUES (102, 'Pat', 6000);	1 row(s) affected	0.000 sec
128	12:03:09	INSERT INTO EMPLOYEE(EMPLOYEE_ID, NAME, SALARY) VALUES (103, 'Den', 11000);	1 row(s) affected	0.016 sec
129	12:03:17	DELETE e1 FROM EMPLOYEE e1 INNER JOIN EMPLOYEE e2 WHERE e1.EMPLOYEE_ID = e2.EMPLOYEE_ID AND e1.NAME = e2.NAME AND e1.SALARY = e2.SALARY AND e1.id > e2.id;	8 row(s) affected	0.000 sec

## 1.ii) How to find the employee with second MAX Salary using a SQL query? Without using SQLAnalytic Functions and Using SQL Analytic Functions.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
/* 1.B) How to find the employee with second MAX Salary using a SQL query? Without using SQL  
Analytic Functions and Using SQL Analytic Functions.*/  
  
SELECT EMPLOYEE_ID, NAME, SALARY  
FROM EMPLOYEE  
WHERE SALARY = (  
    SELECT MAX(SALARY)  
    FROM EMPLOYEE  
    WHERE SALARY < (SELECT MAX(SALARY) FROM EMPLOYEE)  
);
```

The result grid shows one row:

EMPLOYEE_ID	NAME	SALARY
103	Den	11000.00

The output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
129	12:03:17	DELETE e1 FROM EMPLOYEE e1 INNER JOIN EMPLOYEE e2 WHERE e1.EMPLOYEE_ID = e2.EMPLOYEE_ID	8 row(s) affected	0.000 sec
130	12:03:43	SELECT EMPLOYEE_ID, NAME, SALARY FROM EMPLOYEE WHERE SALARY = ( SELECT MAX(SALARY) FROM EMPLOYEE WHERE SALARY < (SELECT MAX(SALARY) FROM EMPLOYEE)	1 row(s) returned	0.000 sec / 0.000 sec

## 1.iii) How to find the employee with third MAX Salary using a SQL query without using Analytic Functions?

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
/* 1.C)How to find the employee with third MAX Salary using a SQL query without using AnalyticFunctions?*/  
  
SELECT EMPLOYEE_ID, NAME, SALARY FROM EMPLOYEE  
WHERE SALARY = (  
    SELECT MAX(SALARY)  
    FROM EMPLOYEE  
    WHERE SALARY < (  
        SELECT MAX(SALARY) FROM EMPLOYEE WHERE SALARY < (  
            SELECT MAX(SALARY) FROM EMPLOYEE  
        )  
    )  
);
```

The result grid shows one row:

EMPLOYEE_ID	NAME	SALARY
102	Pat	6000.00

The output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
130	12:03:43	SELECT EMPLOYEE_ID, NAME, SALARY FROM EMPLOYEE WHERE SALARY = ( SELECT MAX(SALARY) FROM EMPLOYEE WHERE SALARY < (SELECT MAX(SALARY) FROM EMPLOYEE WHERE SALARY < (SELECT MAX(SALARY) FROM EMPLOYEE)	1 row(s) returned	0.000 sec / 0.000 sec
131	12:04:37	SELECT EMPLOYEE_ID, NAME, SALARY FROM EMPLOYEE WHERE SALARY = ( SELECT MAX(SALARY) FROM EMPLOYEE WHERE SALARY < (SELECT MAX(SALARY) FROM EMPLOYEE WHERE SALARY < (SELECT MAX(SALARY) FROM EMPLOYEE)	1 row(s) returned	0.000 sec / 0.000 sec

**2. CREATE TABLE Departments( Department\_ID number, Department\_Name varchar(50) );**

**INSERT INTO DEPARTMENTS VALUES('10','Administration');**

**INSERT INTO DEPARTMENTS VALUES('20','Marketing');**

**INSERT INTO DEPARTMENTS VALUES('30','Purchasing');**

**INSERT INTO DEPARTMENTS VALUES('40','Human Resources');**

**INSERT INTO DEPARTMENTS VALUES('50','Shipping');**

**INSERT INTO DEPARTMENTS VALUES('60','IT');**

**INSERT INTO DEPARTMENTS VALUES('70','Public Relations');**

**INSERT INTO DEPARTMENTS VALUES('80','Sales');**

**i)How to read TOP 5 records from a table using SQL query?**

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following queries:

```
67 • INSERT INTO DEPARTMENTS VALUES('60','IT');
68 • INSERT INTO DEPARTMENTS VALUES('70','Public Relations');
69 • INSERT INTO DEPARTMENTS VALUES('80','Sales');
70
71 /* 2. A) How to read TOP 5 records from a table*/
72 • SELECT * FROM Departments
73   ORDER BY Department_ID ASC
74   LIMIT 5;
75
76 /* 2. B) How to read LAST 5 records from a table using a SQL query?*/
77
78 • SELECT * FROM Departments
```

The Result Grid shows the top 5 records from the Departments table:

Department_ID	Department_Name
10	Administration
20	Marketing
30	Purchasing
40	Human Resources
50	Shipping

The Output pane shows the execution of the query, indicating that 5 rows were returned.

## 2.ii) How to read LAST 5 records from a table using a SQL query?

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
73 ORDER BY Department_ID ASC
74 LIMIT 5;
75
76 /* 2.8) How to read LAST 5 records from a table using a SQL query?*/
77
78 SELECT * FROM Departments
79 ORDER BY Department_ID DESC
80 LIMIT 5;
81
82 /* Q. 3*/
83 /* 3.A) JOINS */
84
```

The result grid shows the following data:

Department_ID	Department_Name
80	Sales
70	Public Relations
60	IT
50	Shipping
40	Human Resources

The output pane shows the execution of the query:

#	Time	Action	Message	Duration / Fetch
132	12:05:36	SELECT * FROM Departments ORDER BY Department_ID ASC LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec
133	12:07:52	SELECT * FROM Departments ORDER BY Department_ID DESC LIMIT 5	5 row(s) returned	0.000 sec / 0.000 sec

## 3.What is the result of NORMAL JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, and FULL OUTER JOIN between tables A & B?

### i)NORMAL JOIN

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
94
95 INSERT INTO Table_B (COL1, COL2, COL3, COL4) VALUES (1), (2), (3), (4);
96
97 /**NORMAL JOIN*/
98 SELECT A.COL AS A_COL, B.COL AS B_COL
99 FROM Table_A A
100 INNER JOIN Table_B B ON A.COL = B.COL;
101
102 /**LEFT OUTER JOIN*/
103
104 SELECT A.COL AS A_COL, B.COL AS B_COL
105 FROM Table_A A
```

The result grid shows the following data:

A_COL	B_COL
1	1
1	1
0	0

The output pane shows the execution of the query:

#	Time	Action	Message	Duration / Fetch
138	12:08:54	SELECT A.COL AS A_COL, B.COL AS B_COL FROM Table_A A INNER JOIN Table_B B ON A.COL = B.COL	3 row(s) returned	0.000 sec / 0.000 sec
139	12:08:59	SELECT A.COL AS A_COL, B.COL AS B_COL FROM Table_A A INNER JOIN Table_B B ON A.COL = B.COL	3 row(s) returned	0.000 sec / 0.000 sec

## ii) LEFT OUTER JOIN

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
97  /**NORMAL JOIN*/  
98  s; Execute the selected portion of the script or everything, if there is no selection  
99  FROM Table_A A  
100  INNER JOIN Table_B B ON A.COL = B.COL;  
101  
102  /**LEFT OUTER JOIN*/  
103  
104  SELECT A.COL AS A_COL, B.COL AS B_COL  
105  FROM Table_A A  
106  LEFT JOIN Table_B B ON A.COL = B.COL;  
107  
108  /**RIGHT OUTER JOIN */
```

The Result Grid shows the output of the query:

A_COL	B_COL
1	1
1	1
0	0
NULL	NULL

The Output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
139	12:08:59	SELECT A.COL AS A_COL, B.COL AS B_COL FROM Table_A A INNER JOIN Table_B B ON A.COL = B.COL...	3 row(s) returned	0.000 sec / 0.000 sec
140	12:09:21	SELECT A.COL AS A_COL, B.COL AS B_COL FROM Table_A A LEFT JOIN Table_B B ON A.COL = B.COL...	4 row(s) returned	0.000 sec / 0.000 sec

## iii) RIGHT OUTER JOIN

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
106  LEFT JOIN Table_B B ON A.COL = B.COL;  
107  Execute the selected portion of the script or everything, if there is no selection  
108  /**RIGHT OUTER JOIN */  
109  SELECT A.COL AS A_COL, B.COL AS B_COL  
110  FROM Table_A A  
111  RIGHT JOIN Table_B B ON A.COL = B.COL;  
112  
113  /** FULL OUTER JOIN*/  
114  SELECT A.COL AS A_COL, B.COL AS B_COL  
115  FROM Table_A A  
116  LEFT JOIN Table_B B ON A.COL = B.COL  
117  UNION
```

The Result Grid shows the output of the query:

A_COL	B_COL
1	1
1	1
0	0
NULL	NULL

The Output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
140	12:09:21	SELECT A.COL AS A_COL, B.COL AS B_COL FROM Table_A A LEFT JOIN Table_B B ON A.COL = B.COL...	4 row(s) returned	0.000 sec / 0.000 sec
141	12:09:49	SELECT A.COL AS A_COL, B.COL AS B_COL FROM Table_A A RIGHT JOIN Table_B B ON A.COL = B.COL...	5 row(s) returned	0.000 sec / 0.000 sec

## IV) FULL OUTER JOIN

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
/* 4 A) Write a SQL query for a report that provides the following information for each person in the Person table, regardless if there is an address for each of those people:
FirstName, LastName, City, State*/

SELECT
    p.FirstName,
    p.LastName,
    a.City,
    a.State
FROM Person p
LEFT JOIN Address a ON p.PersonId = a.PersonId

/*Q.5*/
```

The Results grid displays the following data:

FirstName	LastName	City	State
John	Doe	New York	NY
Jane	Smith	Los Angeles	CA
Mike	Brown	Chennai	Tamilnadu
Lisa	White		
Rose	Jacky		

The bottom status bar indicates the query was completed at 12:11 on 02-06-2025.

4. Write a SQL query for a report that provides the following information for each person in the Person table, regardless if there is an address for each of those people:  
**FirstName, LastName, City, State**

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
/* 4 A) Write a SQL query for a report that provides the following information for each person in the Person table, regardless if there is an address for each of those people:
FirstName, LastName, City, State*/

SELECT
    p.FirstName,
    p.LastName,
    a.City,
    a.State
FROM Person p
LEFT JOIN Address a ON p.PersonId = a.PersonId

/*Q.5*/
```

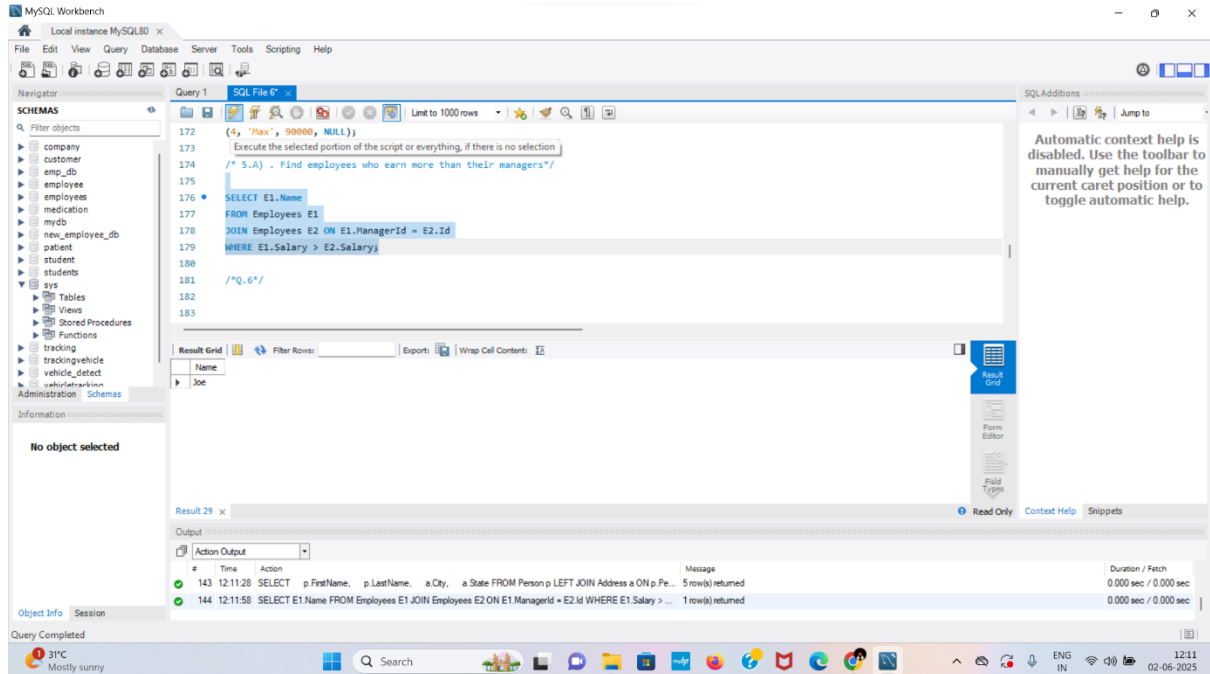
The Results grid displays the following data:

FirstName	LastName	City	State
John	Doe	New York	NY
Jane	Smith	Los Angeles	CA
Mike	Brown	Chennai	Tamilnadu
Lisa	White		
Rose	Jacky		

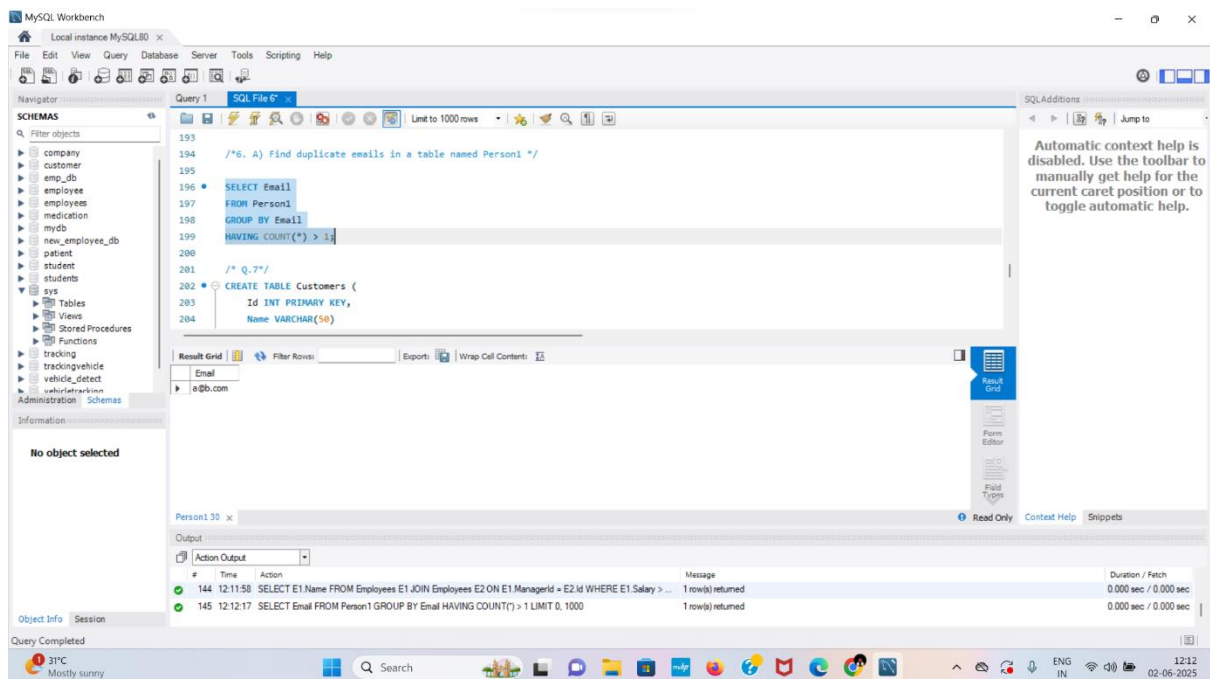
The bottom status bar indicates the query was completed at 12:11 on 02-06-2025.



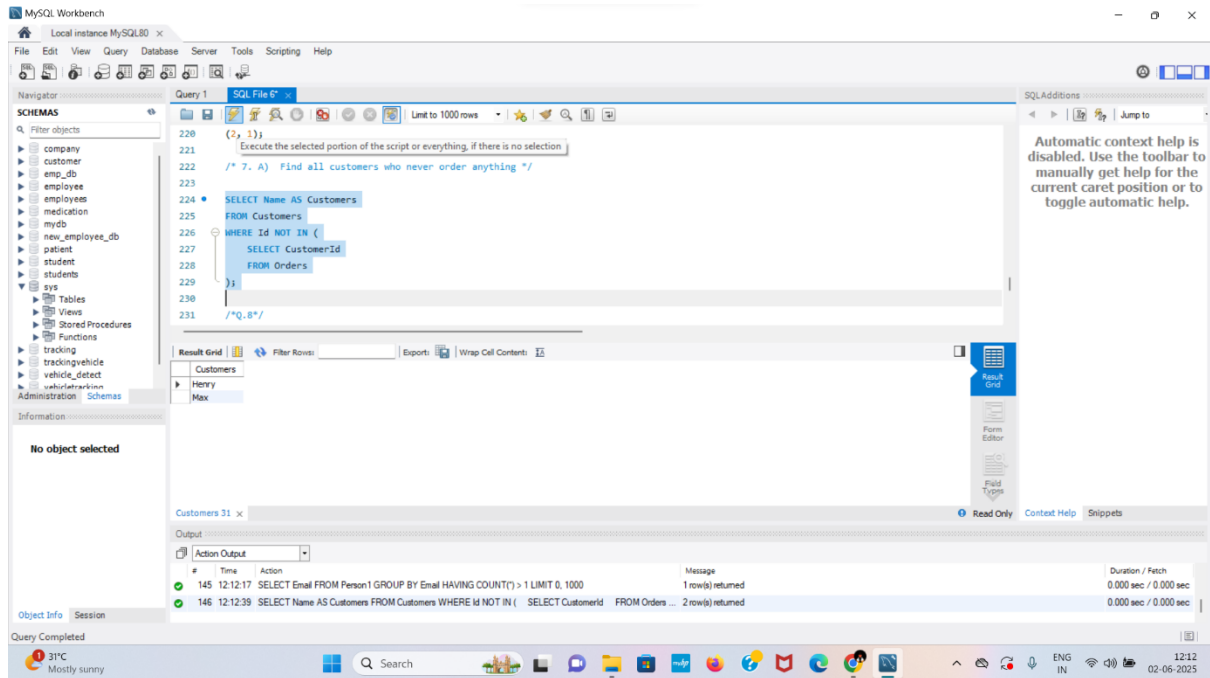
5. write a SQL query that finds out employees who earn more than their managers. For the above table, Joe is the only employee who earns more than his manager.



6. Write a SQL query to find all duplicate emails in a table named Person.



**7. Suppose that a website contains two tables, the Customers table and the Orders table. Write a SQL query to find all customers who never order anything.**



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```

220 (2, 1);
221 Execute the selected portion of the script or everything, if there is no selection
222 /* 7. A) Find all customers who never order anything */
223
224 SELECT Name AS Customers
225 FROM Customers
226 WHERE Id NOT IN (
227     SELECT CustomerId
228     FROM Orders
229 );
230
231 /*Q.6*/
  
```

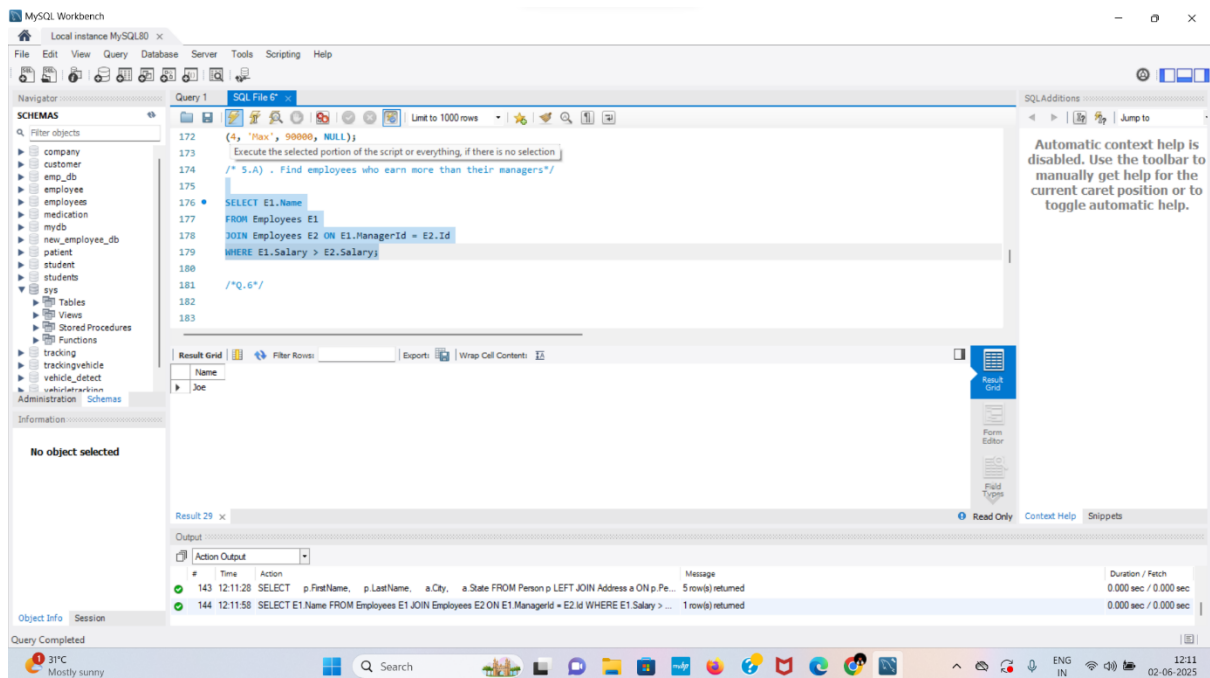
The Results window shows the output of the query:

Customers
Henry
Max

The Output window shows the execution log:

#	Time	Action	Message	Duration / Fetch
145	12:12:17	SELECT Email FROM Person1 GROUP BY Email HAVING COUNT(*) > 1 LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
146	12:12:39	SELECT Name AS Customers FROM Customers WHERE Id NOT IN ( SELECT CustomerId FROM Orders ...	2 row(s) returned	0.000 sec / 0.000 sec

**8. Given the Employee table, write a SQL query that finds out managers with at least 5 direct report. For the above table, your SQL query should return:**



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```

172 (4, 'Max', 90000, NULL);
173 Execute the selected portion of the script or everything, if there is no selection
174 /* 5.A) . Find employees who earn more than their managers*/
175
176 SELECT E1.Name
177 FROM Employees E1
178 JOIN Employees E2 ON E1.ManagerId = E2.Id
179 WHERE E1.Salary > E2.Salary;
180
181 /*Q.6*/
182
183
  
```

The Results window shows the output of the query:

Name
Joe

The Output window shows the execution log:

#	Time	Action	Message	Duration / Fetch
143	12:11:28	SELECT p.FirstName, p.LastName, a.City, a.State FROM Person p LEFT JOIN Address a ON p.Pe...	5 row(s) returned	0.000 sec / 0.000 sec
144	12:11:58	SELECT E1.Name FROM Employees E1 JOIN Employees E2 ON E1.ManagerId = E2.Id WHERE E1.Salary > ...	1 row(s) returned	0.000 sec / 0.000 sec