

# DBMS LAB 3

V Venkataraman 106118106

## Q1

```
use mysampldb;
```

```
DROP TABLE EMP;
```

```
CREATE TABLE EMP(
```

```
    EMPNO INT NOT NULL,
```

```
    EFNAME VARCHAR(100) NOT NULL,
```

```
    ELNAME VARCHAR(100) NOT NULL,
```

```
    JOB VARCHAR(100),
```

```
    DEPTNO INT,
```

```
    ECITY VARCHAR(100),
```

```
    SAL FLOAT,
```

```
    MANAGERNAME VARCHAR(100),
```

```
    MANAGERNO INT,
```

```
    PRIMARY KEY(EMPNO),
```

```
    CHECK (SAL>0)
```

```
);
```

```
INSERT INTO EMP VALUES(1, 'Suresh', 'Raina', 'SDE', 1,  
'Chennai', 100000.25, 'Jack', 101);
```

```
INSERT INTO EMP VALUES(2, 'Yuvraj', 'Singh', NULL, 2,  
'Mumbai', 2000000.25, 'MS', 4);
```

```
INSERT INTO EMP VALUES(3, 'Virender', 'Sewhag', NULL,  
3, 'Delhi', 300000.25, 'Jack', 101);
```

```
INSERT INTO EMP VALUES(4, 'MS', 'Dhoni', 'Sales', 1,  
'Chennai', 40000000.25, NULL, NULL);
```

```
INSERT INTO EMP VALUES(5, 'Sachin', 'Tendulkar',  
'Research', 2, 'Mumbai', 5000000.25, 'Mike', 103);
```

```

INSERT INTO EMP VALUES(6, 'Virat', 'Kohli', 'Research', 3,
'Bangalore', 400000.25, 'Chris', 104);
INSERT INTO EMP VALUES(7, 'Zaheer', 'Khan', 'Electrical',
1, 'Kolkata', 300000.25, 'Virat', 6);
INSERT INTO EMP VALUES(8, 'Ashish', 'Nehra', 'SDE', 2,
'Chennai', 300000.25, 'Chris', 105);
INSERT INTO EMP VALUES(9, 'Ravichander', 'Ashwin',
'Sales', 3, 'Chennai', 2000000.25, 'Jack', 101);
INSERT INTO EMP VALUES(10, 'Harbajan', 'Singh', NULL,
1, 'Delhi', 100000000.25, NULL, NULL);
select * from EMP;

```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with 'mysampdb' selected. The main window shows the 'Result Grid' for the 'EMP' table. The table contains 10 rows of employee data. Below the table, the 'Output' panel shows the execution of a query: 'select \* from EMP LIMIT 0, 1000', which returned 10 rows.

EMPNO	EFNAME	ELNAME	JOB	DEPTNO	ECITY	SAL	MANAGERNAME	MANAGERNO
1	Suresh	Raina	SDE	1	Chennai	100000	Jack	101
2	Yuvraj	Singh	MS	2	Mumbai	2000000	MS	4
3	Virender	Sewhag	MS	3	Delhi	300000	Jack	101
4	MS	Dhoni	Sales	1	Chennai	40000000		
5	Sachin	Tendulkar	Research	2	Mumbai	5000000	Mike	103
6	Virat	Kohli	Research	3	Bangalore	400000	Chris	104
7	Zaheer	Khan	Electrical	1	Kolkata	300000	Virat	6
8	Ashish	Nehra	SDE	2	Chennai	300000	Chris	105
9	Ravichander	Ashwin	Sales	3	Chennai	2000000	Jack	101
10	Harbajan	Singh	MS	1	Delhi	100000000		

```

SELECT * FROM EMP ORDER BY
CASE
    WHEN JOB IS NULL THEN DEPTNO
    ELSE JOB
END
;

```

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mysampkdb

Tables

books

emp1

issue

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

No object selected

Lab3cmds1 Lab3cmds2

Limit to 1000 rows

25 • INSERT INTO EMP VALUES(8, 'Ashish', 'Nehra', 'SDE', 2, 'Chennai', 300000.25, 'Chris', 1

26 • INSERT INTO EMP VALUES(9, 'Ravichander', 'Ashwin', 'Sales', 3, 'Chennai', 200000.25, 'J

27 • INSERT INTO EMP VALUES(10, 'Harbajan', 'Singh', NULL, 1, 'Delhi', 1000000.25, NULL, 1

28 • select \* from EMP;

Result Grid

EMPNO	EFNAME	ELNAME	JOB	DEPTNO	ECITY	SAL	MANAGERNAME	MANAGERNO
10	Harbajan	Singh		1	Delhi	1000000.00		
2	Yuvraj	Singh		2	Mumbai	2000000.00	MS	4
3	Virender	Sewhag		3	Delhi	300000.00	Jack	101
7	Zaheer	Khan	Electrical	1	Kolkata	300000.00	Virat	6
5	Sachin	Tendulkar	Research	2	Mumbai	500000.00	Mike	103
6	Virat	Kohli	Research	3	Bangalore	400000.00	Chris	104
4	MS	Dhoni	Sales	1	Chennai	4000000.00		
9	Ravichander	Ashwin	Sales	3	Chennai	200000.00	Jack	101
1	Suresh	Raina	SDE	1	Chennai	100000.00	Jack	101
8	Ashish	Nehra	SDE	2	Chennai	300000.00	Chris	105

Output

Action Output

#	Time	Action	Message	Duration / Fetch
56	18:25:32	select * from EMP LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
57	18:26:26	SELECT * FROM EMP ORDER BY CASE WHEN JOB IS NULL THEN DEPTNO ELSE...	10 row(s) returned	0.047 sec / 0.000 sec

Object Info Session

06:26 PM 14-09-2020

SELECT COUNT(\*) FROM EMP;

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mysampkdb

Tables

books

emp1

issue

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

No object selected

Lab3cmds1 Lab3cmds2

Limit to 1000 rows

36

37

38 • SELECT COUNT(\*) FROM EMP;

39

Result Grid

COUNT(*)
10

Output

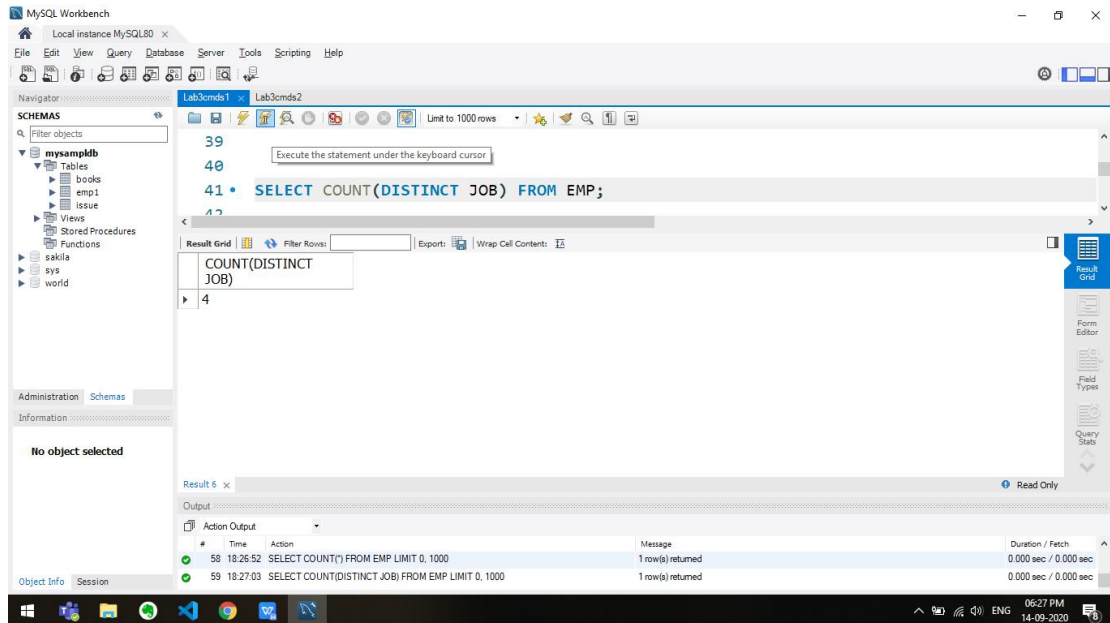
Action Output

#	Time	Action	Message	Duration / Fetch
57	18:26:26	SELECT * FROM EMP ORDER BY CASE WHEN JOB IS NULL THEN DEPTNO ELSE...	10 row(s) returned	0.047 sec / 0.000 sec
58	18:26:52	SELECT COUNT(*) FROM EMP LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

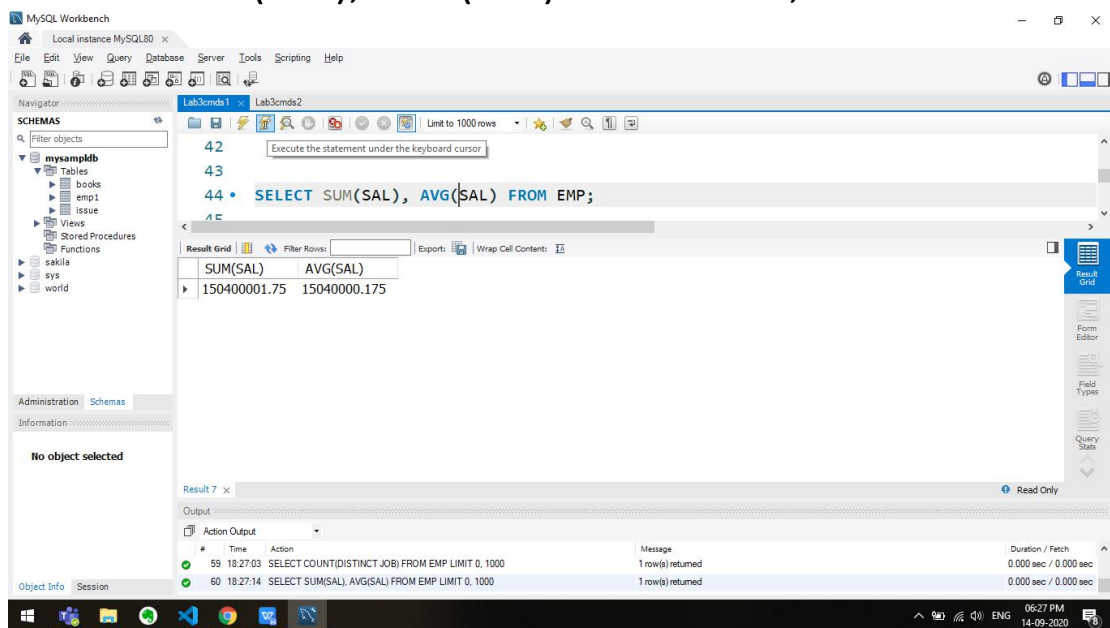
Object Info Session

06:26 PM 14-09-2020

SELECT COUNT(DISTINCT JOB) FROM EMP;



SELECT SUM(SAL), AVG(SAL) FROM EMP;



SELECT MAX(SAL) AS max\_salary, MIN(SAL) AS min\_salary FROM EMP;

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mysampdb

Tables

books

emp1

issue

Views

Stored Procedures

Functions

skila

sys

world

Administration Schemas

Information

No object selected

Lab3cmds1 Lab3cmds2

Limit to 1000 rows

45

46

47 • `SELECT MAX(SAL) AS max_salary, MIN(SAL) AS min_salary FROM EMP;`

Result Grid

Filter Rows:

Export: Wrap Cell Contents

	max_salary	min_salary
▶	100000000	100000

Output

Action Output

#	Time	Action	Message	Duration / Fetch
60	18:27:14	SELECT SUM(SAL), AVG(SAL) FROM EMP LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
61	18:27:24	SELECT MAX(SAL) AS max_salary, MIN(SAL) AS min_salary FROM EMP LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

06:27 PM 14-09-2020

`SELECT MAX(SAL) - MIN(SAL) AS difference FROM EMP;`

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mysampdb

Tables

books

emp1

issue

Views

Stored Procedures

Functions

skila

sys

world

Administration Schemas

Information

No object selected

Lab3cmds1 Lab3cmds2

Limit to 1000 rows

48

49

50 • `SELECT MAX(SAL) - MIN(SAL) AS difference FROM EMP;`

Result Grid

Filter Rows:

Export: Wrap Cell Contents

	difference
▶	99899999.75

Output

Action Output

#	Time	Action	Message	Duration / Fetch
61	18:27:24	SELECT MAX(SAL) AS max_salary, MIN(SAL) AS min_salary FROM EMP LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
62	18:27:37	SELECT MAX(SAL) - MIN(SAL) AS difference FROM EMP LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

06:27 PM 14-09-2020

`SELECT SUM(SAL), MIN(SAL), MAX(SAL), AVG(SAL) FROM EMP WHERE JOB = 'Research';`

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mysampdb

Tables

books

emp1

issue

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

No object selected

Lab3cmds1 Lab3cmds2

Limit to 1000 rows

51 Execute the statement under the keyboard cursor

52

53 • `SELECT SUM(SAL), MIN(SAL), MAX(SAL), AVG(SAL) FROM EMP WHERE JOB = 'Research';`

54

Result Grid

SUM(SAL)	MIN(SAL)	MAX(SAL)	AVG(SAL)
5400000.25	400000	5000000	2700000.125

Output

Action Output

#	Time	Action	Message	Duration / Fetch
62	18:27:37	SELECT MAX(SAL) - MIN(SAL) AS difference FROM EMP LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
63	18:27:45	SELECT SUM(SAL), MIN(SAL), MAX(SAL), AVG(SAL) FROM EMP WHERE JOB = 'Research...	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

06:27 PM 14-09-2020

`SELECT COUNT(DISTINCT SAL) FROM EMP;`

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

mysampdb

Tables

books

emp1

issue

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

No object selected

Lab3cmds1 Lab3cmds2

Limit to 1000 rows

54 Execute the statement under the keyboard cursor

55

56 • `SELECT COUNT(DISTINCT SAL) FROM EMP;`

57

Result Grid

COUNT(DISTINCT SAL)
7

Output

Action Output

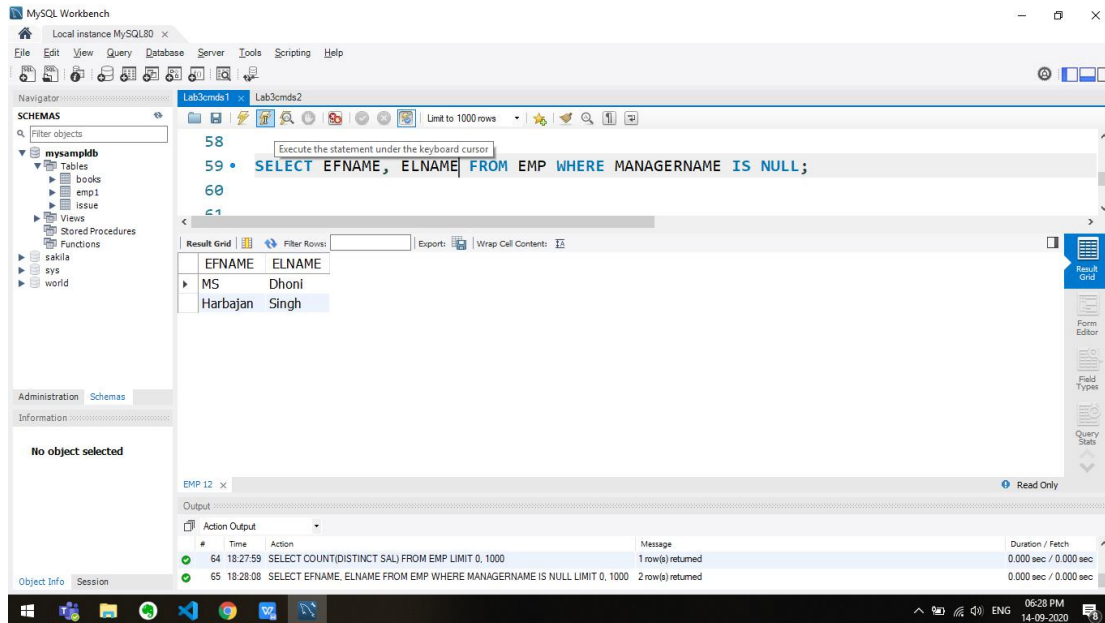
#	Time	Action	Message	Duration / Fetch
63	18:27:45	SELECT SUM(SAL), MIN(SAL), MAX(SAL), AVG(SAL) FROM EMP WHERE JOB = 'Research...	1 row(s) returned	0.000 sec / 0.000 sec
64	18:27:59	SELECT COUNT(DISTINCT SAL) FROM EMP LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

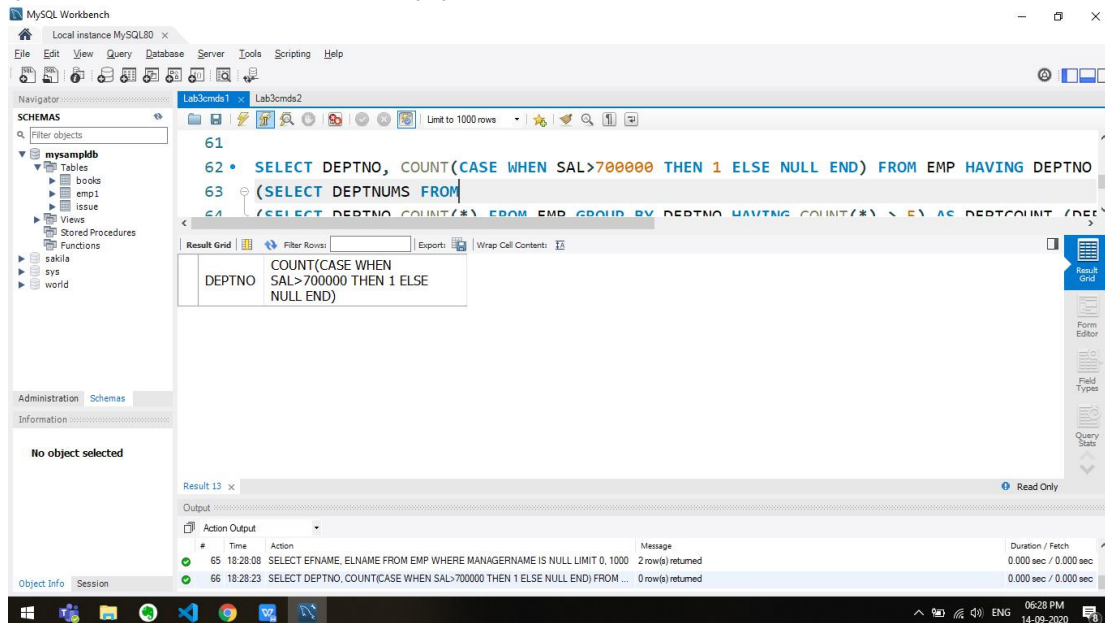
06:28 PM 14-09-2020

`SELECT EFNAME, ELNAME FROM EMP WHERE MANAGERNAME IS NULL;`



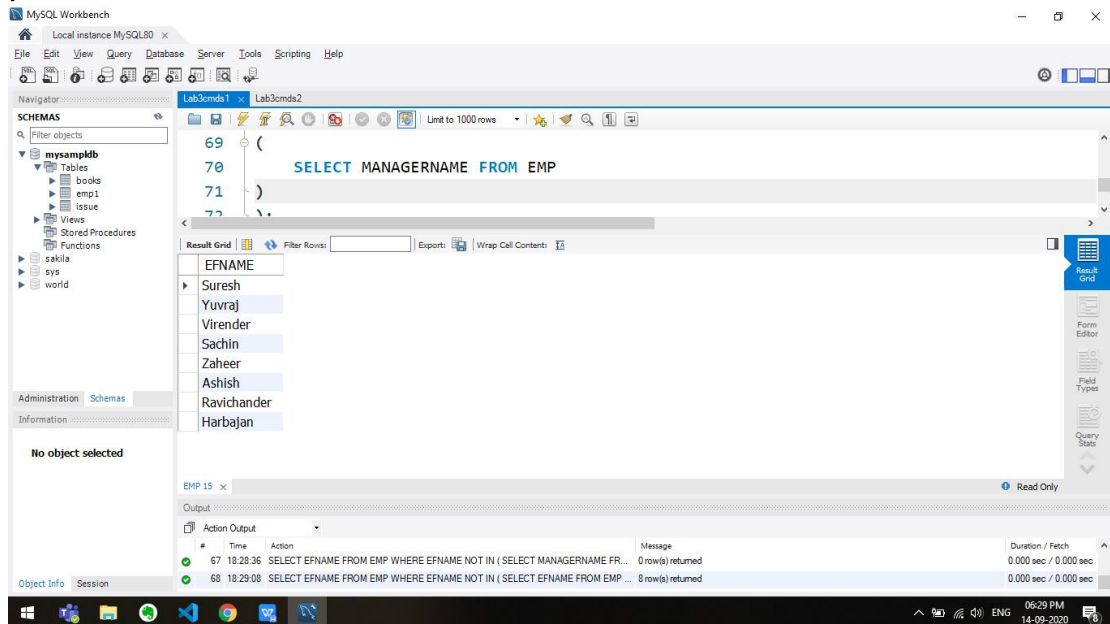


SELECT DEPTNO, COUNT(CASE WHEN SAL>700000 THEN 1 ELSE NULL END) FROM EMP HAVING DEPTNO IN (SELECT DEPTNUMS FROM (SELECT DEPTNO,COUNT(\*) FROM EMP GROUP BY DEPTNO HAVING COUNT(\*) > 5) AS DEPTCOUNT (DEPTNUMS, COUNT) );

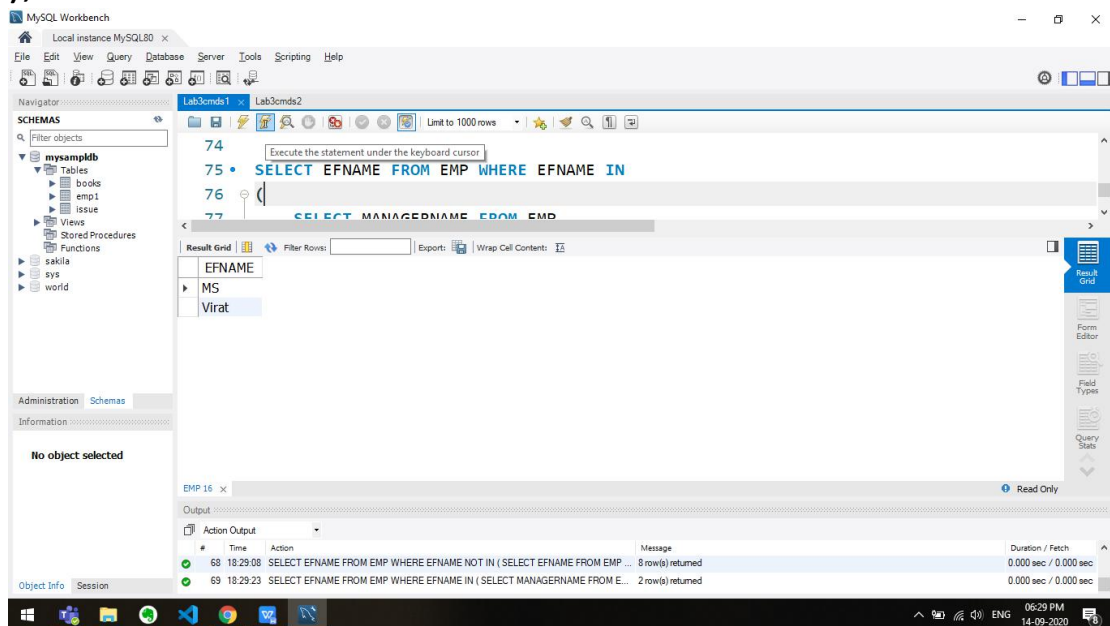


SELECT EFNAME FROM EMP WHERE EFNAME NOT IN

```
(
SELECT MANAGERNAME FROM EMP
);
```



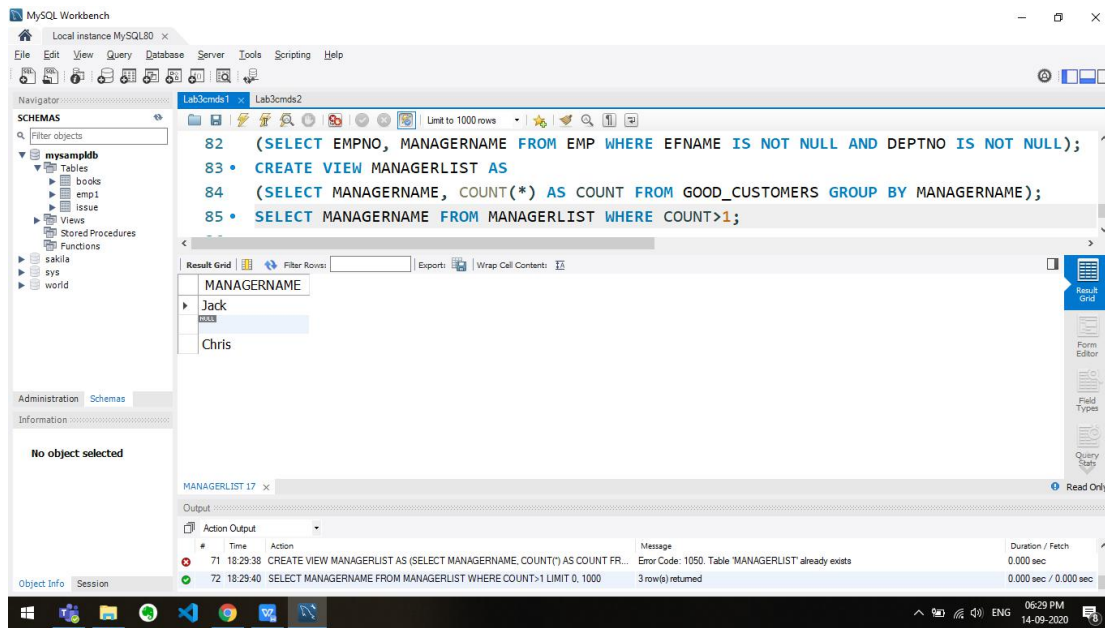
```
SELECT EFNAME FROM EMP WHERE EFNAME IN
(
SELECT MANAGERNAME FROM EMP
);
```



CREATE VIEW GOOD\_CUSTOMERS AS



```
(SELECT EMPNO, MANAGERNAME FROM EMP WHERE
EFNAME IS NOT NULL AND DEPTNO IS NOT NULL);
CREATE VIEW MANAGERLIST AS
(SELECT MANAGERNAME, COUNT(*) AS COUNT FROM
GOOD_CUSTOMERS GROUP BY MANAGERNAME);
SELECT MANAGERNAME FROM MANAGERLIST WHERE
COUNT>1;
```



Q2

```
use mysampldb;
DROP TABLE Borrower;
DROP TABLE Depositor;
DROP TABLE Customer;
DROP TABLE Bank;
```

```
CREATE TABLE Borrower(
    customer_name VARCHAR(100),
    bank_no INT,
    loan_no INT,
    Amount FLOAT
);
```

```
CREATE TABLE Depositor(  
    customer_name VARCHAR(100),  
    bank_no INT,  
    account_no INT,  
    Amount FLOAT  
);
```

```
CREATE TABLE Customer(  
    customer_name VARCHAR(100),  
    street VARCHAR(100),  
    customer_city VARCHAR(100),  
    PRIMARY KEY(customer_name)  
);
```

```
CREATE TABLE Bank(  
    bankno INT,  
    Bankname VARCHAR(100),  
    city VARCHAR(100),  
    street VARCHAR(100),  
    PRIMARY KEY(bankno)  
);
```

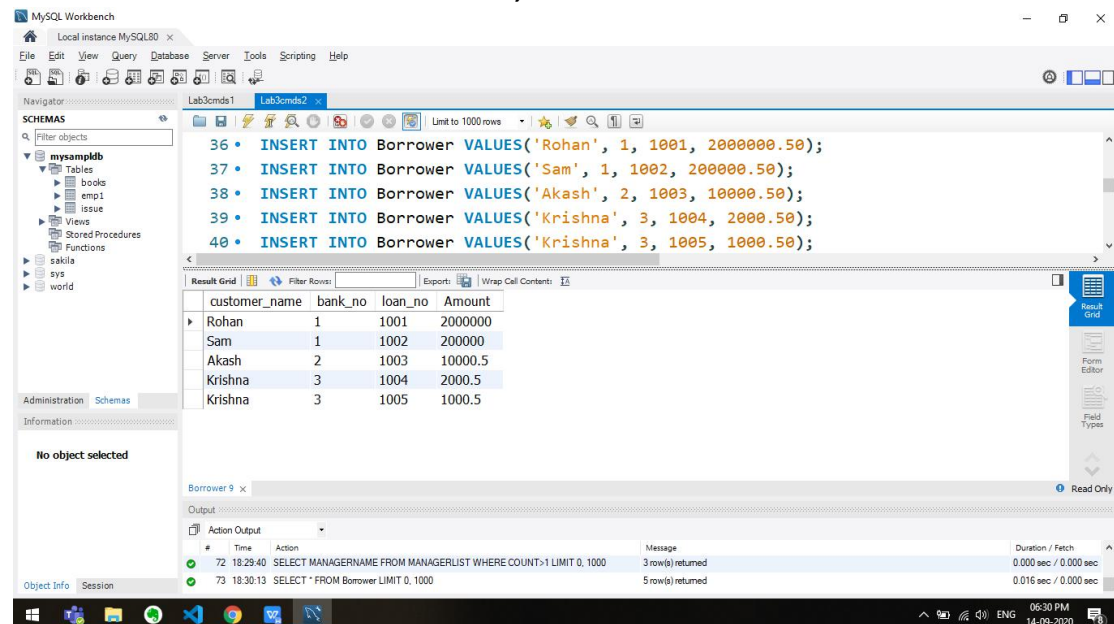
```
INSERT INTO Borrower VALUES('Rohan', 1, 1001,  
2000000.50);
```

```
INSERT INTO Borrower VALUES('Sam', 1, 1002,  
200000.50);
```

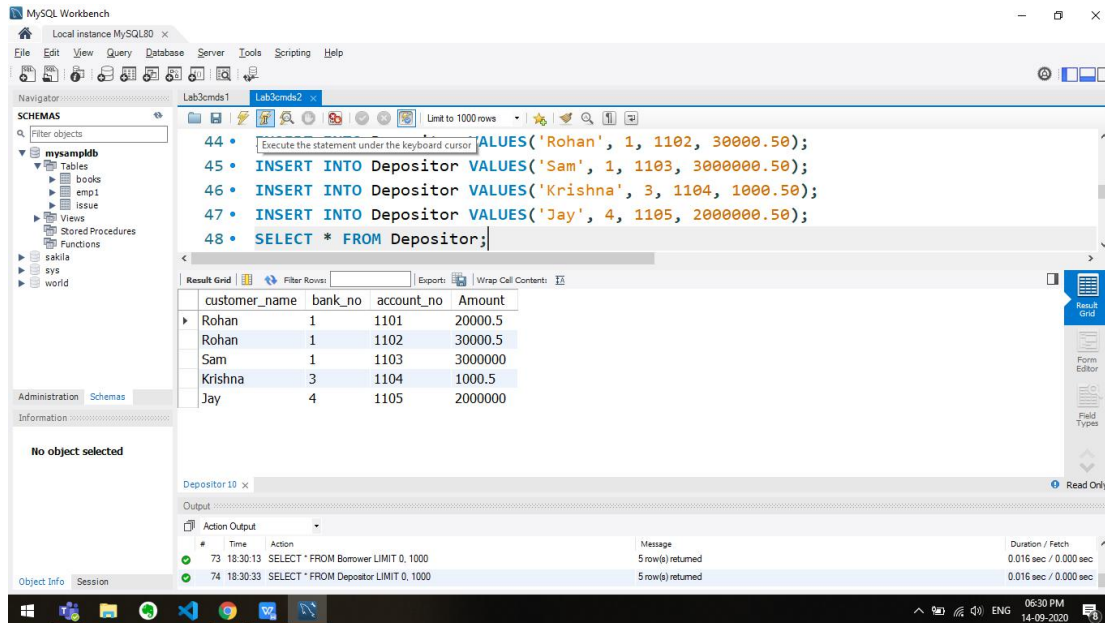
```
INSERT INTO Borrower VALUES('Akash', 2, 1003,  
10000.50);
```

```
INSERT INTO Borrower VALUES('Krishna', 3, 1004,  
2000.50);
```

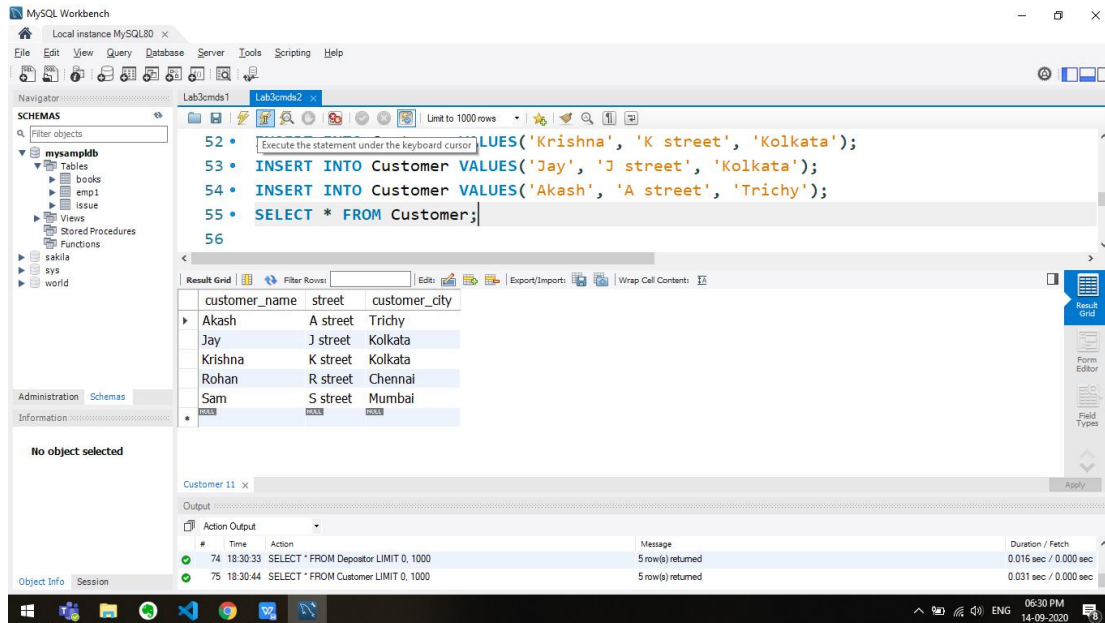
```
INSERT INTO Borrower VALUES('Krishna', 3, 1005,  
1000.50);  
SELECT * FROM Borrower;
```



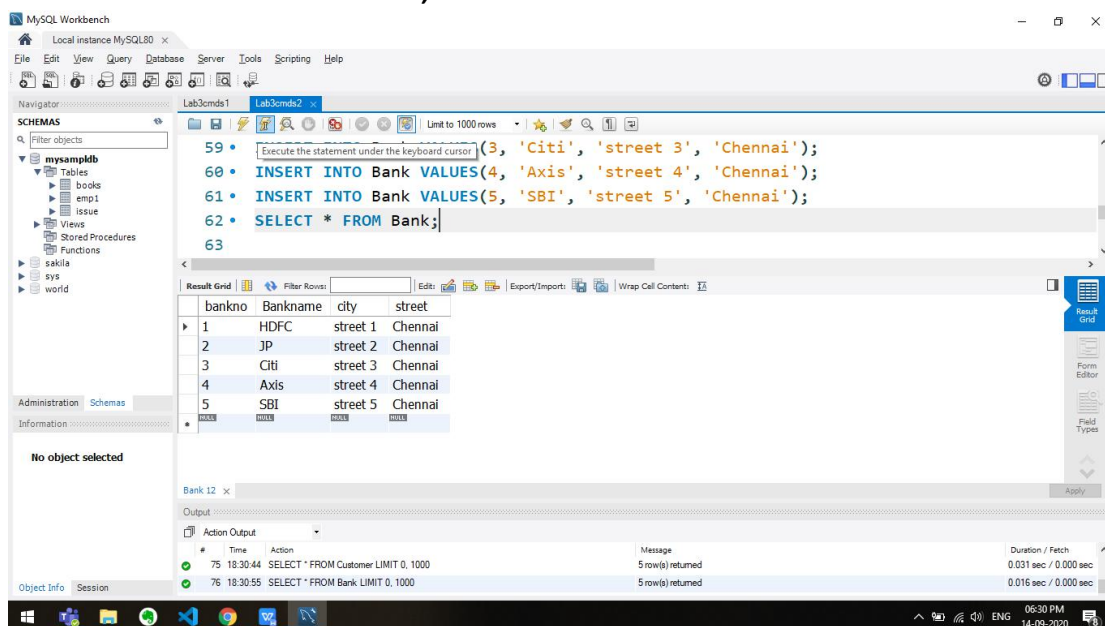
```
INSERT INTO Depositor VALUES('Rohan', 1, 1101,  
20000.50);  
INSERT INTO Depositor VALUES('Rohan', 1, 1102,  
30000.50);  
INSERT INTO Depositor VALUES('Sam', 1, 1103,  
3000000.50);  
INSERT INTO Depositor VALUES('Krishna', 3, 1104,  
1000.50);  
INSERT INTO Depositor VALUES('Jay', 4, 1105,  
2000000.50);  
SELECT * FROM Depositor;
```



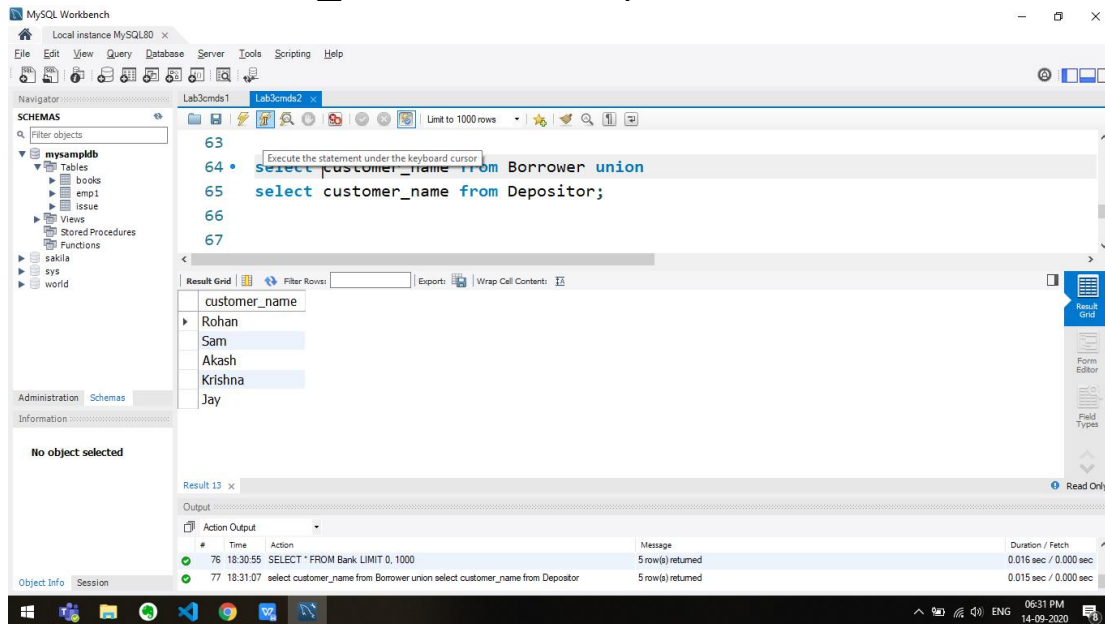
INSERT INTO Customer VALUES('Rohan', 'R street',  
'Chennai');  
INSERT INTO Customer VALUES('Sam', 'S street',  
'Mumbai');  
INSERT INTO Customer VALUES('Krishna', 'K street',  
'Kolkata');  
INSERT INTO Customer VALUES('Jay', 'J street',  
'Kolkata');  
INSERT INTO Customer VALUES('Akash', 'A street',  
'Trichy');  
SELECT \* FROM Customer;



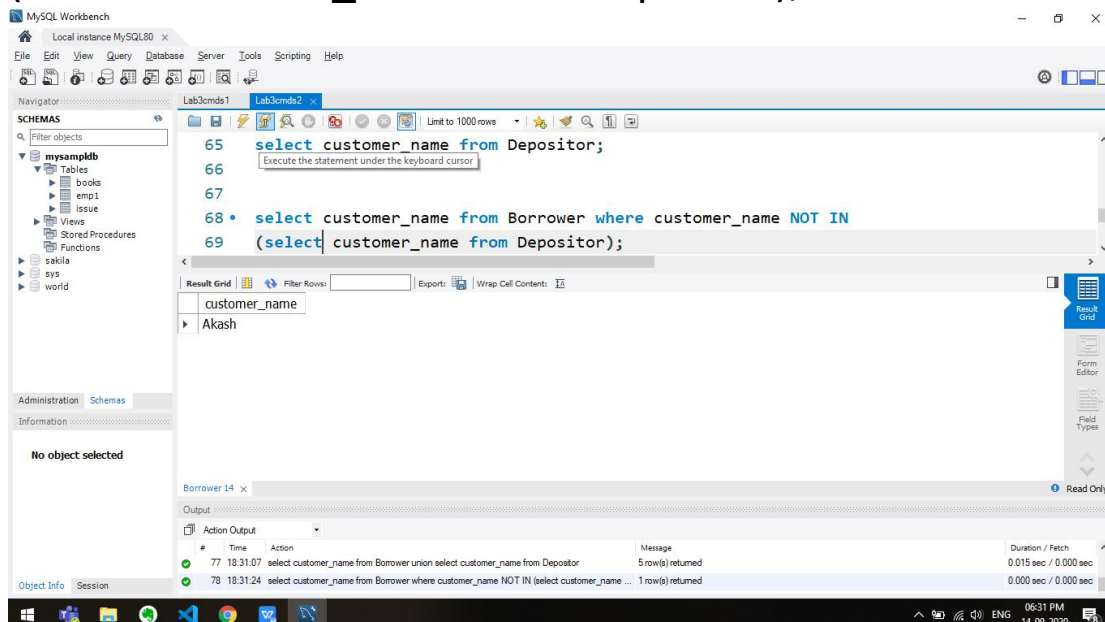
INSERT INTO Bank VALUES(1, 'HDFC', 'street 1', 'Chennai');  
 INSERT INTO Bank VALUES(2, 'JP', 'street 2', 'Chennai');  
 INSERT INTO Bank VALUES(3, 'Citi', 'street 3', 'Chennai');  
 INSERT INTO Bank VALUES(4, 'Axis', 'street 4', 'Chennai');  
 INSERT INTO Bank VALUES(5, 'SBI', 'street 5', 'Chennai');  
 SELECT \* FROM Bank;



select customer\_name from Borrower union  
select customer\_name from Depositor;

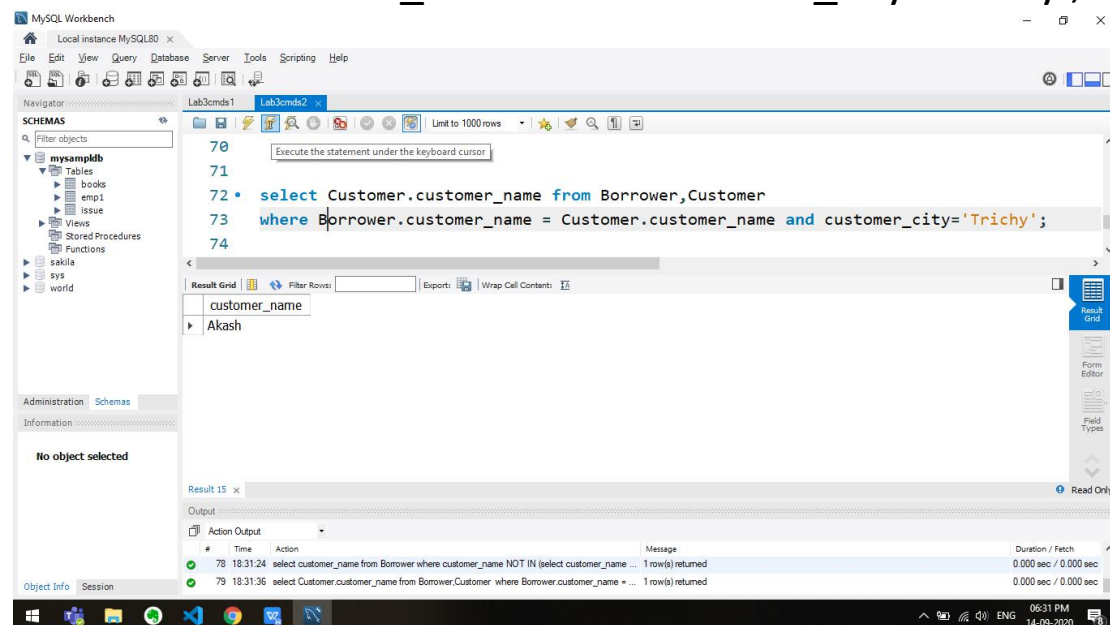


select customer\_name from Borrower where  
customer\_name NOT IN  
(select customer\_name from Depositor);

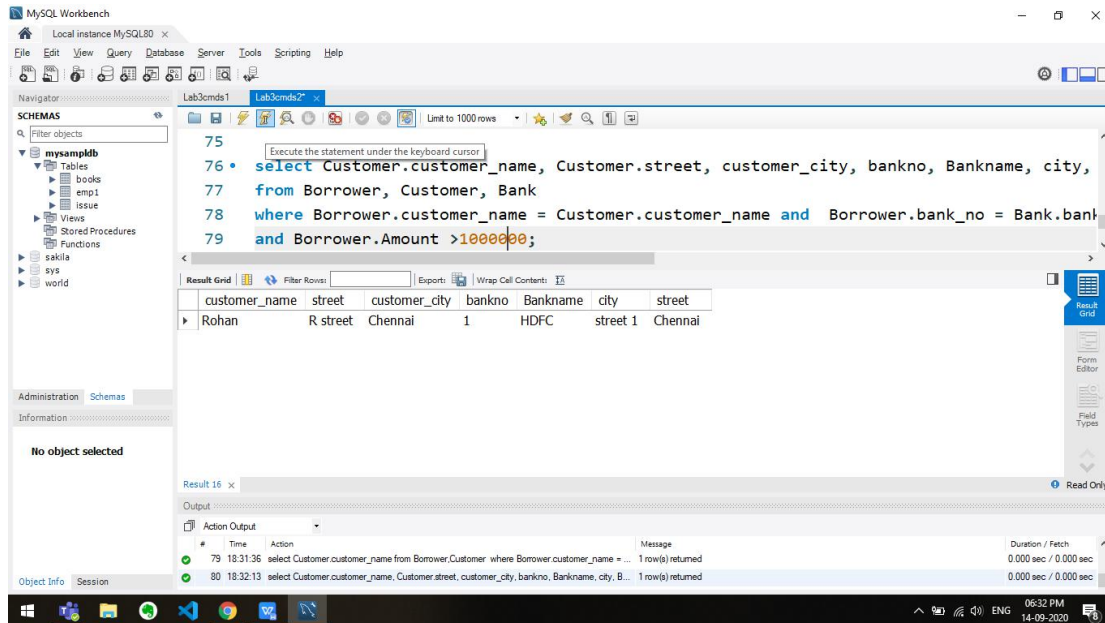




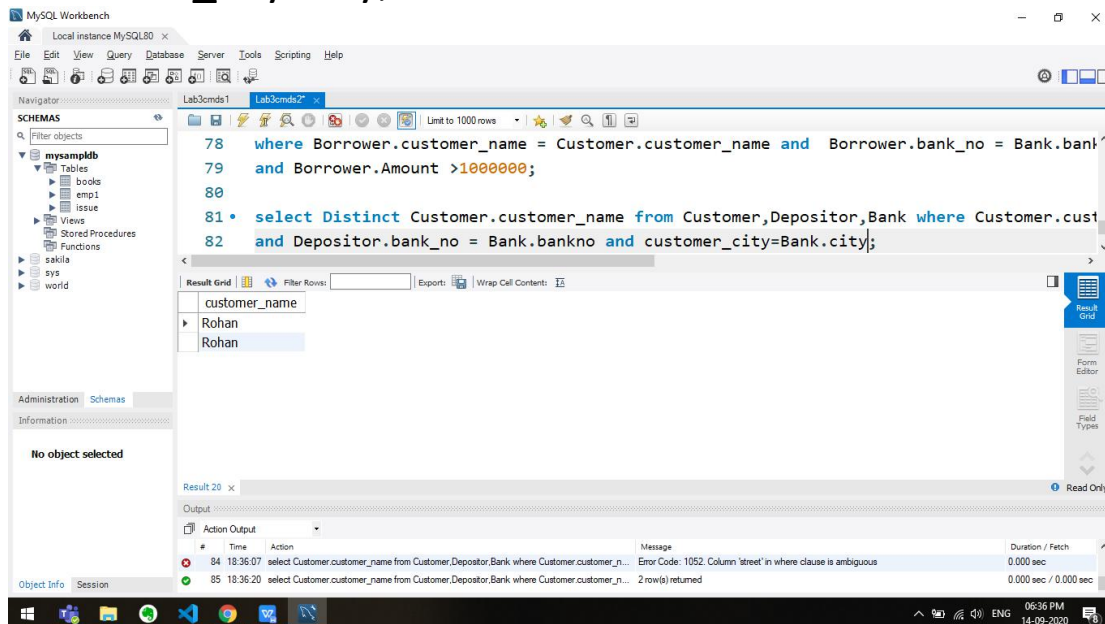
```
select Customer.customer_name from
Borrower, Customer
where Borrower.customer_name =
Customer.customer_name and customer_city='Trichy';
```



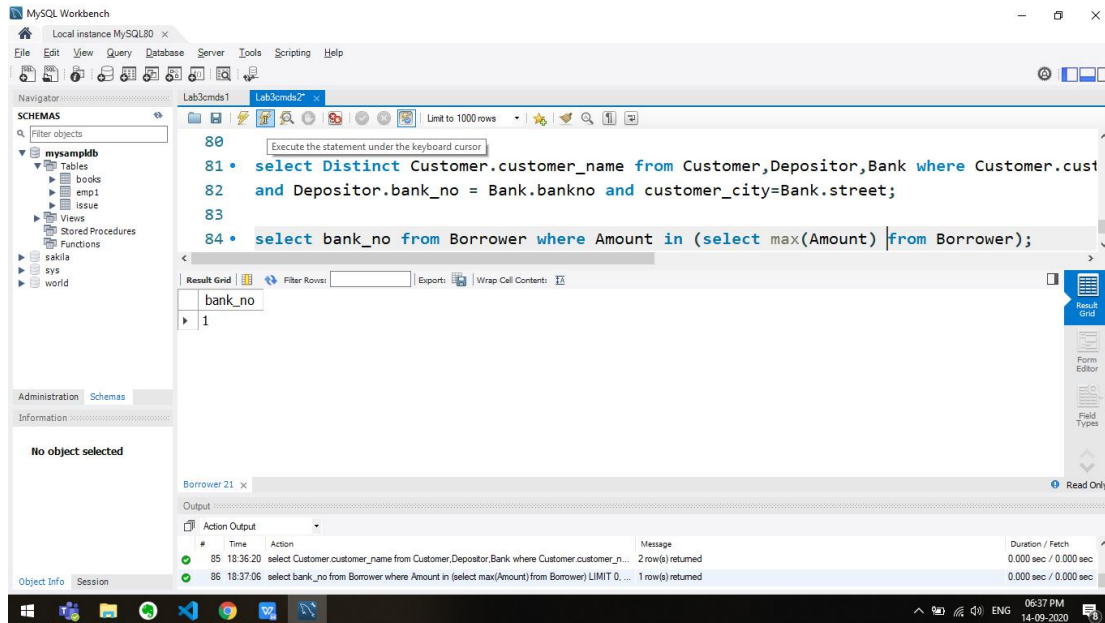
```
select Customer.customer_name, Customer.street,
customer_city, bankno, Bankname, city, Bank.street
from Borrower, Customer, Bank
where Borrower.customer_name =
Customer.customer_name and Borrower.bank_no =
Bank.bankno
and Borrower.Amount >1000000;
```



select Distinct Customer.customer\_name from  
Customer, Depositor, Bank where  
Customer.customer\_name = Depositor.customer\_name  
and Depositor.bank\_no = Bank.bankno and  
customer\_city=city;



select bank\_no from Borrower where Amount in (select  
max(Amount) from Borrower);



DELETE Customer, Depositor, Borrower FROM Customer,  
Depositor, Borrower

WHERE Customer.customer\_name =  
Borrower.customer\_name OR  
Customer.customer\_name = Depositor.customer\_name;  
SELECT \* FROM CUSTOMER;

