

## **EXPERIMENT NO. 1**

### **AIM**

To study various DDL commands – CREATE, ALTER, DROP, TRUNGATE, RENAME

### **Question**

Create the following Tables

#### **Table 1: DEPOSIT**

ACTNO VARCHAR2(5) PRIMARY KEY, FIRST LETTER MUST START WITH 'D'

CNAME VARCHAR2(15) FOREIGN KEY REFERENCES CUSTOMER

BNAME VARCHAR2(20) FOREIGN KEY REFERENCES BRANCH

AMOUNT NUMBER (8,2) NOT NULL, CANNOT BE 0

ADATE DATE

#### **Table 2: BRANCH**

BNAME VARCHAR2(20) PRIMARY KEY

CITY VARCHAR2(30) NOT NULL , any one of NAGPUR, DELHI, BANGALORE, BOMBAY

#### **Table 3: CUSTOMER**

CNAME VARCHAR2(15) PRIMARY KEY

CITY VARCHAR2(20) NOT NULL ,

#### **Table 4: BORROW**

LOANNO VARCHAR2(8) PRIMARY KEY / FIRST LETTER MUST START WITH 'L'

CNAME VARCHAR2(15) FOREIGN KEY REFERENCES CUSTOMER

BNAME VARCHAR2(20) FOREIGN KEY REFERENCES BRANCH

AMOUNT NUMBER(8,2) NOT NULL, CANNOT BE 0

### **INSERTION OF VALUES**

#### **1. Inserting values to Branch**

VRCE	NAGPUR
AJNI	NAGPUR
KAROLBAGH	DELHI
CHANDNI	DELHI
DHARAMPETH	NAGPUR
MG ROAD	BANGALORE

ANDHERI	BOMBAY
NEHRU PALACE	DELHI
POWAI	BOMBAY

## 2. Inserting values into Customer table

ANIL	CALCUTTA
SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MADHURI	NAGPUR
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
KRANTI	BOMBAY
NAREN	BOMBAY

## 3. Inserting values into Deposit table

Actno	Cname	Bname	Amount	Adate
D100	ANIL	VRCE	1000.00	1-MAR-95
D101	SUNIL	ANJNI	500.00	4-JAN-96
D102	MEHUL	KAROLBAGH	3500.00	17-NOV-95
D104	MADHURI	CHANDNI	1200.00	17-DEC-95
D105	PRAMOD	MG ROAD	3000.00	27-MAR-96
D106	SANDIP	ANDHERI	2000.00	31-MAR-96
D107	SHIVANI	VIRAR	1000.00	5-SEP-95
D108	KRANTI	NEHRU PLACE	5000.00	2-JUL-95
D109	MINU	POWAI	7000.00	10-AUG-95

## 4. Inserting values into borrow table

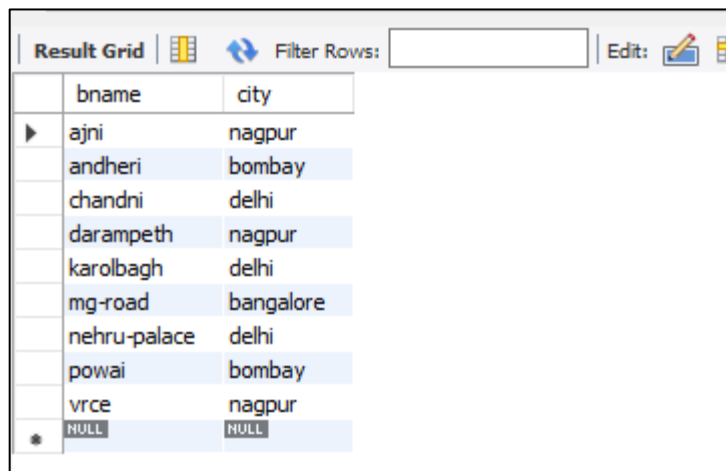
L201	ANIL	VRCE	1000.00
L206	MEHUL	AJNI	5000.00
L311	SUNIL	DHARAMPETH	3000.00
L321	MADHURI	ANDHERI	2000.00
L371	PRAMOD	VIRAR	8000.00
L481	KRANTI	NEHRU PLACE	3000.00

## Procedure

CREATE DATABASE BANK;

USE BANK;

```
CREATE TABLE BRANCH (BNAME VARCHAR(20) NOT NULL PRIMARY KEY,CITY  
VARCHAR (30) NOT NULL);  
  
INSERT INTO BRANCH (BNAME, CITY)  
  
VALUES (AJNI, NAGPUR),  
  
(ANDHERI, BOMBAY),  
  
(CHANDNI, DELHI),  
  
(DHARAMPETH, NAGPUR),  
  
(KAROLBAGH, DELHI),  
  
(MG ROAD, BANGALORE),  
  
(NEHRU PALACE, DELHI),  
  
(POWAI, BOMBAY),  
  
(VRCE, NAGPUR);
```



bname	city
ajni	nagpur
andheri	bombay
chandni	delhi
darampeth	nagpur
karolbagh	delhi
mg-road	bangalore
nehru-palace	delhi
powai	bombay
vrce	nagpur
NULL	NULL

```
CREATE TABLE CUSTOMER(CNAME VARCHAR(15) PRIMARY KEY ,CITY VARCHAR  
(20) NOT NULL);
```

```
INSERT INTO CUSTOMER VALUES ('ANIL' ,'CALCUTTA');  
  
INSERT INTO CUSTOMER VALUES ('KRANTI' ,'BOMBAY');  
  
INSERT INTO CUSTOMER VALUES ('MEHUL' ,'BARODA');
```

```

INSERT INTO CUSTOMER VALUES ('NAREN' ,'BOMBAY');

INSERT INTO CUSTOMER VALUES ('MANDAR' ,'PATNA');

INSERT INTO CUSTOMER VALUES ('MADHURI' ,'NAGPUR');

INSERT INTO CUSTOMER VALUES ('PRAMOD' ,'NAGPUR');

INSERT INTO CUSTOMER VALUES ('SANDIP' ,'SURAT');

INSERT INTO CUSTOMER VALUES ('SHIVANI' ,'BOMBAY');

INSERT INTO CUSTOMER VALUES ('SUNIL' ,'DELHI');

```

cname	city
anil	calicut
kranti	bombay
madhuri	nagpur
mandar	patna
mehul	baroda
naren	bombay
paramod	nagpur
sandhip	surat
shivani	bombay
sunil	delhi
NULL	NULL

deposit 35   branch 36   customer 37 x

```

CREATE TABLE DEPOSIT (ACTNO VARCHAR(5) CHECK (ACTNO LIKE 'D%') PRIMARY
KEY ,CNAME VARCHAR (15) ,FOREIGN KEY (CNAME) REFERENCES CUSTOMER
(CNAME),BNAME VARCHAR(30), FOREIGN KEY (BNAME) REFERENCES
BRANCH(BNAME),AMOUNT INT NOT NULL, ADATE DATE);

INSERT INTO `DEPOSIT` (`ACTNO`,`CNAME`,`BNAME`,`AMOUNT`,`ADATE`)
VALUES

('D100','ANIL','VRCE',1000,'1995-03-01'),

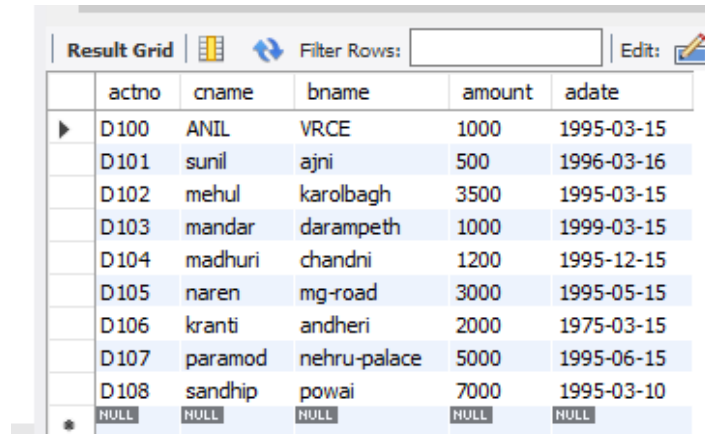
('D101','SUNIL','AJNI',500,'1996-01-04'),

('D102','MEHUL','KAROLBAGH',3500,'1995-11-17'),

('D104','MADHURI','CHANDNI',1200,'1995-12-17'),

```

( 'D105', 'PRAMOD', 'MG ROAD', 3000, '1996-03-27'),  
 ( 'D106', 'SANDIP ', 'ANDHERI', 2000, '1996-03-31'),  
 ( 'D107', 'SHIVANI', 'VRCE', 1000, '1995-09-05'),  
 ( 'D108', 'KRANTI', 'NEHRU PALACE', 5000, '1995-07-02'),  
 ( 'D109', 'NAREN', 'POWAI', 7000, '1995-08-10');



	actno	cname	bname	amount	adate
▶	D100	ANIL	VRCE	1000	1995-03-15
	D101	sunil	ajni	500	1996-03-16
	D102	mehul	karolbagh	3500	1995-03-15
	D103	mandar	darampeth	1000	1999-03-15
	D104	madhuri	chandni	1200	1995-12-15
	D105	naren	mg-road	3000	1995-05-15
	D106	kranti	andheri	2000	1975-03-15
	D107	paramod	nehru-palace	5000	1995-06-15
	D108	sandhip	powai	7000	1995-03-10
*	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE BORROW (LOANNO VARCHAR (15) CHECK (LOANNO
LIKE'L%')PRIMARY KEY ,CNAME VARCHAR (15) ,FOREIGN KEY (CNAME)
REFERENCES CUSTOMER (CNAME),BNAME VARCHAR(30), FOREIGN KEY (BNAME)
REFERENCES BRANCH(BNAME),AMOUNT INT NOT NULL);

INSERT INTO `BORROW` (`LOAN_NO`, `CNAME`, `BNAME`, `AMOUNT`) VALUES
('L201', 'ANIL', 'VRCE', 1000),
('L206', 'MEHUL', 'AJNI', 5000),
('L311', 'SUNIL', 'DHARAMPETH', 3000),
('L321', 'MADHURI', 'ANDHERI', 2000),
('L371', 'PRAMOD', 'VRCE', 8000),
('L481', 'KRANTI', 'NEHRU PALACE', 3000);
```

Result Grid				
		Filter Rows:		Edit:
	LOANNO	CNAME	BNAME	AMOUNT
▶	L201	ANIL	VRCE	1000
	L206	MEHUL	AJNI	5000
	L311	SUNIL	DHARAMPETH	3000
	L321	MADHURI	ANDHERI	2000
	L371	PRAMOD	VIRAR	8000
	L481	KRANTI	NEHRU PLACE	3000
★	NULL	NULL	NULL	NULL

### **a)Questions**

1. List all data from table deposit
2. List all data from borrow
3. List all data from customer
4. List all data from branch
5. Give account no and amount of deposit
6. Give customer name and account no of depositors
7. Give name of customers
8. Give name of branches
9. Give name of borrows
10. Give names of customer living in city Nagpur
11. Give names of depositors having amount greater than 4000
12. Give account date of Anil
13. Give name of all branches located in Bombay
14. Give name of borrower having loan number 1205
15. Give names of depositors having account at VRCE
16. Give names of all branches located in city Delhi
17. Give name of the customers who opened account date '1-12-96'
18. Give account no and deposit amount of customers having account opened between dates '1-12-96' and '1-5-96'
19. Give name of the city where branch KAROLBAGH is located
20. Give details of customer ANIL

### **Procedure**

## 1 List all data from table deposit-

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

```

40 • insert into deposit values ('D109', 'MANDAR', 'POWAI', 7000.00, '1995-08-10');
41 • SELECT * FROM deposit;
42 • insert into borrow values('L201', 'ANIL', 'VRICE', 1000.00);
43 • insert into borrow values('L206', 'MEHUL', 'AGNI', 5000.00);
44 • insert into borrow values('L311', 'SUNIL', 'DHARAMPETH', 3000.00);
45 • insert into borrow values('L321', 'MADHURI', 'ANDHERI', 2000.00);
46 • insert into borrow values('L371', 'PRAMOD', 'VRICE', 8000.00);
47 • insert into borrow values('L481', 'KRANTI', 'NEHRU PALACE', 3000.00);
48 • SELECT * FROM deposit;

```

The output grid shows the data from the deposit table:

actno	cname	bname	amount	adate
D100	ANIL	VRICE	1000	1995-03-01
D101	SUNIL	AGNI	500	1996-01-04
D102	MEHUL	KAROLBAGH	3500	1995-11-17
D104	MADHURI	CHANDNI	1200	1995-12-17
D105	PRAMOD	MG ROAD	3000	1996-03-27

The Action Output pane shows the execution results of the queries:

#	Time	Action	Message	Duration / Fetch
52	04:55:11	insert into borrow values('L206','MEHUL','AGNI',5000.00)	1 row(s) affected	0.031 sec
53	04:55:11	insert into borrow values('L311','SUNIL','DHARAMPETH',3000.00)	1 row(s) affected	0.032 sec
54	04:55:11	insert into borrow values('L321','MADHURI','ANDHERI',2000.00)	1 row(s) affected	0.031 sec
55	04:55:11	insert into borrow values('L371','PRAMOD','VRICE',8000.00)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('bank'.bor...	0.000 sec
56	04:56:06	insert into borrow values('L371','PRAMOD','VRICE',8000.00)	1 row(s) affected	0.047 sec
57	04:59:49	SELECT * FROM deposit LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec

## 2 List all data from borrow-

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

```

41 • SELECT * FROM deposit;
42 • insert into borrow values('L201', 'ANIL', 'VRICE', 1000.00);
43 • insert into borrow values('L206', 'MEHUL', 'AGNI', 5000.00);
44 • insert into borrow values('L311', 'SUNIL', 'DHARAMPETH', 3000.00);
45 • insert into borrow values('L321', 'MADHURI', 'ANDHERI', 2000.00);
46 • insert into borrow values('L371', 'PRAMOD', 'VRICE', 8000.00);
47 • insert into borrow values('L481', 'KRANTI', 'NEHRU PALACE', 3000.00);
48 • SELECT * FROM deposit;
49 • SELECT * FROM borrow;

```

The output grid shows the data from the borrow table:

loanno	cname	bname	amount
L201	ANIL	VRICE	1000
L206	MEHUL	AGNI	5000
L311	SUNIL	DHARAMPETH	3000
L321	MADHURI	ANDHERI	2000
L371	PRAMOD	VRICE	8000

The Action Output pane shows the execution results of the queries:

#	Time	Action	Message	Duration / Fetch
53	04:55:11	insert into borrow values('L311','SUNIL','DHARAMPETH',3000.00)	1 row(s) affected	0.032 sec
54	04:55:11	insert into borrow values('L321','MADHURI','ANDHERI',2000.00)	1 row(s) affected	0.031 sec
55	04:55:11	insert into borrow values('L371','PRAMOD','VRICE',8000.00)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('bank'.bor...	0.000 sec
56	04:56:06	insert into borrow values('L371','PRAMOD','VRICE',8000.00)	1 row(s) affected	0.047 sec
57	04:59:49	SELECT * FROM deposit LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
58	05:02:20	SELECT * FROM borrow LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

## 3 List all data from customer

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Query 1

```

42 insert into borrow values('L201', 'ANIL', 'VRICE',1000.00);
43 insert into borrow values('L206', 'MEHUL', 'AGNI', 5000.00);
44 insert into borrow values('L311','SUNIL', 'DHARAMPETH', 3000.00);
45 insert into borrow values('L321', 'MADHURI', 'ANDHERI', 2000.00);
46 insert into borrow values('L371', 'PRAMOD', 'VRICE', 8000.00);
47 insert into borrow values('L481', 'KRANTI', 'NEHRU PALACE', 3000.00);
48 SELECT * FROM deposit;
49 SELECT * FROM borrow;
50 SELECT * FROM customer;

```

Result Grid

crname	city
ANIL	CALCUTTA
KRANTI	BOMBAY
MADHURI	NAGPUR
MANDAR	PATNA
MEHUL	BARODA

customer 8

Output

Action Output

#	Time	Action	Message	Duration / Fetch
54	04:55:11	insert into borrow values('L321','MADHURI','ANDHERI',2000.00)	1 row(s) affected	0.031 sec
55	04:55:11	insert into borrow values('L371','PRAMOD','VRICE',8000.00)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('bank', 'bor...	0.000 sec
56	04:56:06	insert into borrow values('L371','PRAMOD','VRICE',8000.00)	1 row(s) affected	0.047 sec
57	04:59:49	SELECT * FROM deposit LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
58	05:02:20	SELECT * FROM borrow LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
59	05:04:18	SELECT * FROM customer LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

#### 4 List all data from branch

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Query 1

```

44 insert into borrow values('L311','SUNIL', 'DHARAMPETH', 3000.00);
45 insert into borrow values('L321', 'MADHURI', 'ANDHERI', 2000.00);
46 insert into borrow values('L371', 'PRAMOD', 'VRICE', 8000.00);
47 insert into borrow values('L481', 'KRANTI', 'NEHRU PALACE', 3000.00);
48 SELECT * FROM deposit;
49 SELECT * FROM borrow;
50 SELECT * FROM customer;
51 SELECT * FROM branch;
52

```

Result Grid

bname	city
AGNI	NAGPUR
ANDHERI	BOMBAY
CHANDNI	DELHI
DHARAMPETH	NAGPUR
KAROLBAGH	DELHI

branch 9

Output

Action Output

#	Time	Action	Message	Duration / Fetch
55	04:55:11	insert into borrow values('L371','PRAMOD','VRICE',8000.00)	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('bank', 'bor...	0.000 sec
56	04:56:06	insert into borrow values('L371','PRAMOD','VRICE',8000.00)	1 row(s) affected	0.047 sec
57	04:59:49	SELECT * FROM deposit LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
58	05:02:20	SELECT * FROM borrow LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
59	05:04:18	SELECT * FROM customer LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
60	05:05:34	SELECT * FROM branch LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec

#### 5 Give account no and amount of deposit-



The screenshot shows the MySQL Workbench interface with a query window titled 'Query 1'. The query contains several SQL statements: five INSERT statements into the 'borrow' table, three SELECT statements from 'deposit', 'borrow', and 'customer' tables, and one SELECT statement from 'deposit' table. The 'Result Grid' shows the output of the last query, displaying columns 'actno' and 'amount' with five rows of data. The 'Output' pane shows the execution log with messages for each statement.

actno	amount
D100	1000
D101	500
D102	3500
D104	1200
D105	3000

## 6 Give customer name and account no of depositors

The screenshot shows the MySQL Workbench interface with a query window titled 'Query 1'. The query contains several SQL statements: three INSERT statements into the 'borrow' table, three SELECT statements from 'deposit', 'borrow', and 'customer' tables, and one SELECT statement from 'deposit' table. The 'Result Grid' shows the output of the last query, displaying columns 'actno' and 'cname' with five rows of data. The 'Output' pane shows the execution log with messages for each statement.

actno	cname
D100	ANIL
D108	KRANTI
D104	MADHURI
D109	MANDAR
D102	MEHUL

Give name of customers-

The screenshot shows the MySQL Workbench interface. The left sidebar contains the 'MANAGEMENT' and 'INSTANCE' sections. The main query editor displays a series of SQL statements. The 'Result Grid' shows the output of the last query, which is a list of customer names. The 'Action Output' pane at the bottom shows the execution progress of the queries.

**Query 1:**

```

46 • insert into borrow values('L371', 'PRAVOD', 'VRICE', 8000.00);
47 • insert into borrow values('L481', 'KRANTI', 'NEHRU PALACE', 3000.00);
48 • SELECT * FROM deposit;
49 • SELECT * FROM borrow;
50 • SELECT * FROM customer;
51 • SELECT actno,amount FROM deposit;
52 • SELECT actno,amount FROM deposit;
53
54 • select cname from customer;

```

**Result Grid:**

cname
ANIL
KRANTI
MADHURI
MANDAR
MEHUL

**Action Output:**

#	Time	Action	Message	Duration / Fetch
58	05:02:20	SELECT * FROM borrow LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
59	05:04:18	SELECT * FROM customer LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
60	05:05:34	SELECT * FROM branch LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
61	05:08:22	SELECT actno,amount FROM deposit LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
62	05:11:28	SELECT actno,cname FROM deposit LIMIT 0, 1000	9 row(s) returned	0.000 sec / 0.000 sec
63	05:13:50	select cname from customer LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

## 8 Give name of branches-

The screenshot shows the MySQL Workbench interface. The left sidebar contains the 'MANAGEMENT' and 'INSTANCE' sections. The main query editor displays a series of SQL statements. The 'Result Grid' shows the output of the last query, which is a list of branch names. The 'Action Output' pane at the bottom shows the execution progress of the queries.

**Query 1:**

```

51 • SELECT actno,amount FROM deposit;
52 • SELECT actno,amount FROM deposit;
53
54 • select cname from customer;
55 • select bnames from branch;
56 • use bank;
57 • select bname from branch;
58

```

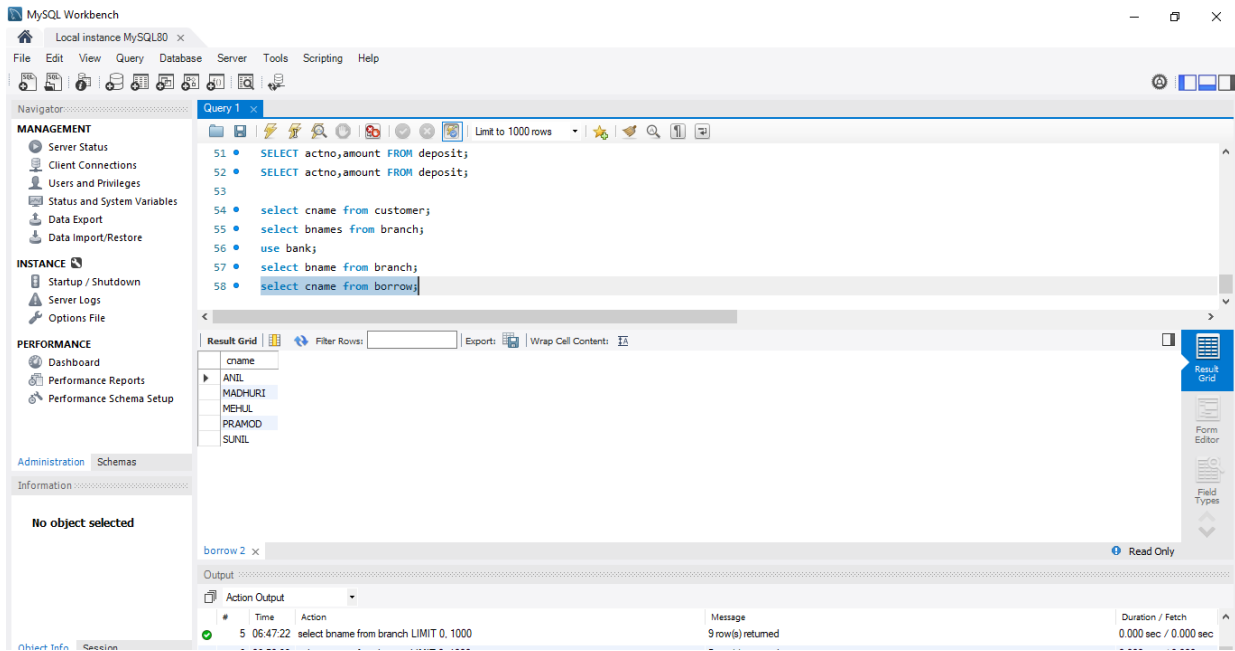
**Result Grid:**

bname
AGNI
ANDHERI
CHANDNI
DHARAMPETH
KAROLBAGH
MG ROAD
NEHRU PALACE
POWAI
VRICE

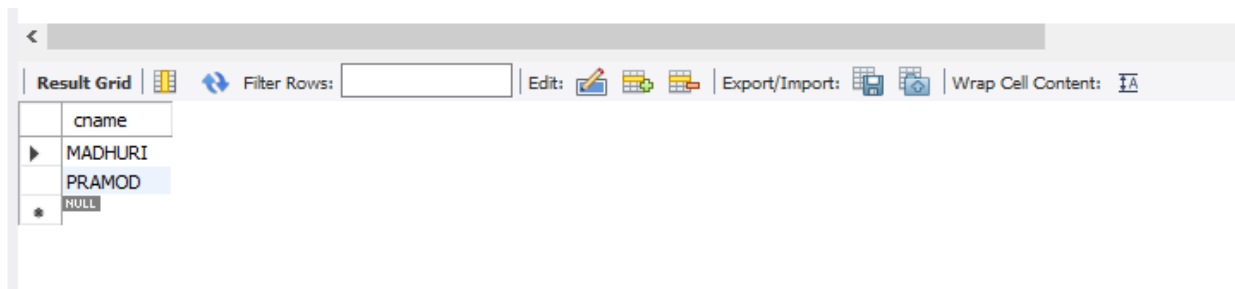
**Action Output:**

#	Time	Action	Message	Duration / Fetch
1	06:45:57	select cnames from branch LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking ...	0.000 sec
2	06:46:15	select bnames from branch LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking ...	0.000 sec

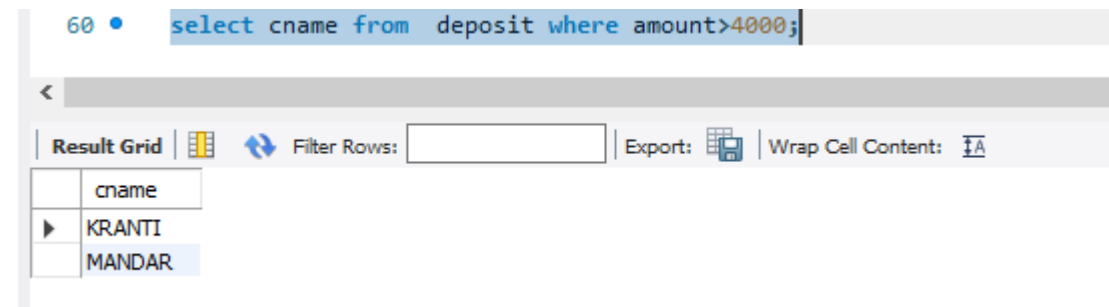
## 9 Give name of borrows-



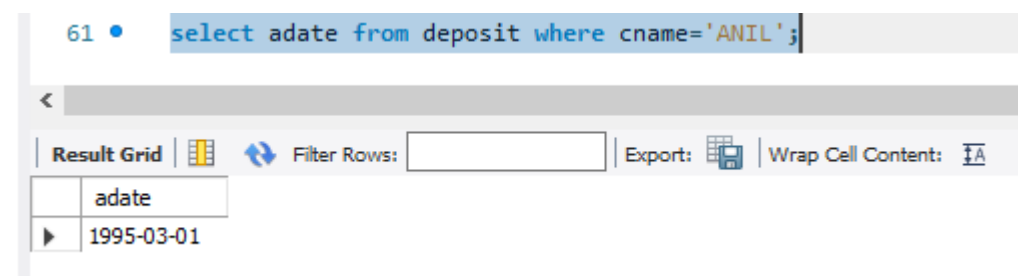
10-Give names of customer living in city Nagpur-



11 - Give names of depositors having amount greater than 4000-



12- Give account date of Anil



13- Give name of all branches located in Bombay

```
62 • select bname from branch where city='BOMBAY';
```

Result Grid

bname
ANDHERI
POWAI
NULL

14- Give name of borrower having loan number 1205

```
63 • select bname from borrow where loanno='L206';
```

Result Grid

bname
AGNI

15- Give names of depositors having account at VRCE

```
64 • select cname from deposit where bname='VRICE';
```

Result Grid

cname
ANIL
SHIVANI

16- Give names of all branches located in city Delhi

```
65 • select bname from branch where city='DELHI';
```

Result Grid

bname
CHANDNI
KAROLBAGH
NEHRU PALACE
NULL

17- Give name of the customers who opened account date '1-12-96'

66 • `select cname from deposit where adate='1996-01-04';`

Result Grid

	cname
▶	SUNIL

18- Give account no and deposit amount of customers having account opened between dates '1-12-96' and '1-5-96'

67 • `select actno,amount from deposit where adate between '1996-12-01' and '1996-05-01';`

Result Grid

	actno	amount
*	NULL	NULL

19- Give name of the city where branch KAROLBAGH is located-

68 • `select city from branch where bname='KAROLBAGH';`

Result Grid

	city
▶	DELHI

20- Give details of customer ANIL-

69 • `select * from deposit where cname='ANIL';`

Result Grid

	actno	cname	bname	amount	adate
▶	D 100	ANIL	VRICE	1000	1995-03-01
*	NULL	NULL	NULL	NULL	NULL

## **b)Questions**

- 1.List total loan
- 2.List total deposit
- 3.List total loan taken from KAROLBAGH branch
- 4.List total deposit of customers having account date later than 1-Jan-96
- 5.List total deposit of customers living in city NAGPUR
- 6.List maximum deposit of customer living in Bombay
- 7.List total deposit of customer having branch in BOMBAY
- 8.Count total number of branch cities

- 9.Count total number of customers cities
- 10.Give branch names and branch wise deposit
- 11.Give city wise name and branch wise deposit
- 12.Give the branch wise loan of customer living in NAGPUR
- 13.Count total number of customers
- 14.Count total number of depositors branch wise
- 15.Count total number of depositors branch wise
- 16.Give maximum loan from branch VRCE
17. Give the number of customers who are depositors as well as borrowers

## Procedure

1. SELECT SUM(AMOUNT) FROM BORROW;

Result Grid		Filter Rows:
	SUM(AMOUNT)	
▶	22000	

Result 1 x

2. SELECT SUM(AMOUNT) FROM DEPOSIT;

Result Grid		Filter Rows:
	SUM(AMOUNT)	
▶	24200	

Result 2 x


3. SELECT MAX(AMOUNT) FROM BORROW WHERE BNAME ='KAROLBAGH';


Result Grid		Filter Rows:
	MAX(AMOUNT)	
▶	NULL	

4. SELECT SUM(AMOUNT) from deposit where adate>'1995-03-01';

Result Grid		Filter Rows:
	SUM(AMOUNT)	
▶	23200	


5. SELECT SUM(D1.AMOUNT) FROM DEPOSIT D1 , CUSTOMER C1 WHERE C1.CITY = 'NAGPUR' AND C1.CNAME = D1.CNAME;

Result Grid			Filter Rows: <input type="text"/>
	SUM(D1.AMOUNT)		
▶	4200		

Result 5 × 


6. SELECT MAX(D1.AMOUNT) FROM DEPOSIT D1 , CUSTOMER C1 WHERE C1.CITY = 'Bombay' AND C1.CNAME = D1.CNAME;

Result Grid			Filter Rows: <input type="text"/>
	MAX(D1.AMOUNT)		
▶	1000		

Result 6 × 


7. SELECT SUM(AMOUNT) from deposit,BRANCH where city='BOMBAY';

Result Grid			Filter Rows: <input type="text"/>
	SUM(AMOUNT)		
▶	48400		

Result 7 × 

8. SELECT COUNT(DISTINCT(CITY)) FROM BRANCH ;

Result Grid			Filter Rows: <input type="text"/>
	COUNT(DISTINCT(CITY))		
▶	4		

Result 8 × 

9. SELECT count(city) from CUSTOMER;

Result Grid			Filter Rows:	<input type="text"/>
	count(city)			
▶	10			

Result 9 ×

10. SELECT BNAME , SUM(AMOUNT) FROM DEPOSIT GROUP BY BNAME;

Result Grid			Filter Rows:	<input type="text"/>	Export:
	BNAME	SUM(AMOUNT)			
▶	VRCE	1000			
	ANJNI	500			
	KAROLBAGH	3500			
	CHANDNI	1200			
	MG ROAD	3000			
	ANDHERI	2000			
	VIRAR	1000			
	NEHRU PLACE	5000			
	POWAI	7000			

Result 10 ×

11. SELECT C1.CITY , SUM(D1.AMOUNT) FROM CUSTOMER C1 , DEPOSIT D1 WHERE D1.CNAME = C1.CNAME GROUP BY C1.CITY;

Result Grid			Filter Rows:	<input type="text"/>	Export:
	CITY	SUM(D1.AMOUNT)			
▶	CALCUTTA	1000			
	DELHI	500			
	BARODA	3500			
	NAGPUR	4200			
	SURAT	2000			
	BOMBAY	1000			

Result 11 ×

12. SELECT BNAME , SUM(AMOUNT) FROM BORROW



Result Grid			Filter Rows:	Export:
	BNAME	SUM(AMOUNT)		
▶	VRCE	2000		
	AJNI	10000		
	DHARAMPETH	6000		
	ANDHERI	4000		
	VIRAR	16000		
	NEHRU PLACE	6000		

Result 12 ×

13. SELECT BORROW,CUSTOMER WHERE city ='NAGPUR' GROUP BY BNAME;

Result Grid			Filter Rows:
	count(cname)		
▶	10		

Result 13 ×

14. SELECT count(cname) from CUSTOMER;

Result Grid			Filter Rows:	Export:
	BNAME	count(*)		
▶	VRCE	1		
	ANJNI	1		
	KAROLBAGH	1		
	CHANDNI	1		
	MG ROAD	1		
	ANDHERI	1		
	VIRAR	1		

Result 14 ×

15. SELECT BNAME, count(\*) from DEPOSIT, CUSTOMER where deposit.CNAME  
CUSTOMER.CNAME group by BNAME;

Result Grid			Filter Rows:	Export:
	BNAME	count(*)		
▶	VRCE	1		
	ANJNI	1		
	KAROLBAGH	1		
	CHANDNI	1		
	MG ROAD	1		
	ANDHERI	1		
	VIRAR	1		

Result 15 ×

16. SELECT BNAME, count(\*) from DEPOSIT, CUSTOMER where deposit.CNAME = CUSTOMER.CNAME group by BNAME;

Result Grid		Filter Rows:
	MAX(AMOUNT)	
▶	1000	

Result 16 ×

17. select count(customer.CNAME) from customer where customer.CNAME IN (select deposit.cname from deposit) and customer.CNAME IN (select borrow.cname from borrow);

Result Grid		Filter Rows:
	count(customer.CNAME)	
▶	5	

Result 17 ×

## **EXPERIMENT NO. 2**

### **AIM**

To familiarize DDL Commands- ALTER,DROP,TRUNCATE,RENAME

### **Questions**

1.Create a table emp with attributes empno number(4)as primary key, ename char(10),hiredate, salary,commission and insert the following 5 rows of data

101	Ramesh	17-Jan 1980	5000	
102	Ajay	05-Jul 1985	5000	500
103	Ravi	12-Aug 1981	1500	
104	Nikesh	03-Mar 1983	3000	700
105	Ravi	05-jul 1985	3000	

### **ANSWERS**

```
CREATE TABLE EMP(empno INT PRIMARY KEY,ename VARCHAR(10),hiredate
DATE,salaryINT,commission INT);
```

```
INSERT INTO emp(empno,ename,hiredate,salary) VALUES(101,"Ramesh","1980-01-17",5000);
```

```
INSERT INTO emp(empno,ename,hiredate,salary,commission)
VALUES(102,"Ajay","1985-07-05",5000,500);
```

```
INSERT INTO emp(empno,ename,hiredate,salary) VALUES(103,"Ravi","1981-08-
12",1500);INSERT INTO emp(empno,ename,hiredate,salary,commission)
VALUES(104,"Nikesh","1983-03-03",3000,700);
```

```
INSERT INTO emp(empno,ename,hiredate,salary) VALUES(105,"Ravi","1985-07-05",3000);
```

16 • `SELECT * FROM EMP;`  
17

Result Grid

	empno	ename	hiredate	salary	commission
▶	101	Ramesh	1980-01-17	5000	NULL
	102	Ajay	1985-07-05	5000	500
	103	Ravi	1981-08-12	1500	NULL
	104	Nikesh	1983-03-03	3000	700
	105	Ravi	1985-07-05	3000	NULL
✱	NULL	NULL	NULL	NULL	NULL

## 2. Add check constraint (Salary <=5000)

`ALTER TABLE emp ADD CHECK(salary<=5000);`

✖	13	09:59:09	ALTER TABLE emp ADD CHECK(salary>=5000)	Error Code: 4025. CONSTRAINT 'CONSTRAINT_1' failed for table
✔	14	10:01:48	ALTER TABLE emp ADD CHECK(salary<=5000)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

## 3. Add Check Constraint (length(empno)<=3)

`ALTER TABLE emp ADD CHECK(length(empno)<=3);`

19 • `ALTER TABLE emp ADD CHECK(length(empno)<=3);`  
20

Output

#	Time	Action	Message
✔	11	09:50:54	SELECT * FROM EMP LIMIT 0, 1000 5 row(s) returned
✖	12	09:58:55	ALTER TABLE EMP ADD CHECK(salary>=5000) Error Code: 4025. CONSTRAINT 'CONSTRAINT_1' failed for table
✖	13	09:59:09	ALTER TABLE emp ADD CHECK(salary>=5000) Error Code: 4025. CONSTRAINT 'CONSTRAINT_1' failed for table
✔	14	10:01:48	ALTER TABLE emp ADD CHECK(salary<=5000) 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
✔	15	10:08:26	ALTER TABLE emp ADD CHECK(length(empno)<=3) 5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0

## 4. Modifying the structure of tables

### a. Add new columns: sal number(7,2)

```

20 • ALTER TABLE emp ADD COLUMN SAL FLOAT(7,2);
21 • SELECT * FROM emp;
22

```

Result Grid

empno	ename	hiredate	salary	commission	SAL
101	Ramesh	1980-01-17	5000	NULL	NULL
102	Ajay	1985-07-05	5000	500	NULL
103	Ravi	1981-08-12	1500	NULL	NULL
104	Nikesh	1983-03-03	3000	700	NULL
105	Ravi	1985-07-05	3000	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL

b. Dropping a column from a table: sal

```

22 • ALTER TABLE emp DROP COLUMN SAL;
23 • SELECT * FROM emp;
24

```

Result Grid

empno	ename	hiredate	salary	commission
101	Ramesh	1980-01-17	5000	NULL
102	Ajay	1985-07-05	5000	500
103	Ravi	1981-08-12	1500	NULL
104	Nikesh	1983-03-03	3000	700
105	Ravi	1985-07-05	3000	NULL
NULL	NULL	NULL	NULL	NULL

c. Modifying existing column :ename varchar2(15)

```

24 • ALTER TABLE emp MODIFY COLUMN ename VARCHAR(15);
25

```

Output

#	Time	Action	Message
✓ 17	10:13:19	SELECT * FROM emp LIMIT 0, 1000	5 row(s) returned
✓ 18	10:16:46	ALTER TABLE emp DROP COLUMN SAL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 19	10:16:56	SELECT * FROM emp LIMIT 0, 1000	5 row(s) returned
✓ 20	10:26:40	ALTER TABLE emp MODIFY COLUMN ename VARCHAR(15)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0

d. Renaming the tables: emp to emp1

```

25 • ALTER TABLE emp RENAME TO emp1;
26

```

Output

#	Time	Action	Message
✓ 18	10:16:46	ALTER TABLE emp DROP COLUMN SAL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 19	10:16:56	SELECT * FROM emp LIMIT 0, 1000	5 row(s) returned
✓ 20	10:26:40	ALTER TABLE emp MODIFY COLUMN ename VARCHAR(15)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 21	10:29:31	ALTER TABLE emp RENAME TO emp1	0 row(s) affected

e.truncating the tables:emp1

```
26 • TRUNCATE TABLE emp1;  
27 • SELECT * FROM emp1;  
28
```

<

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	empno	ename	hiredate	salary	commission
*	NULL	NULL	NULL	NULL	NULL

f.Destroying tables:emp

```
28 • DROP TABLE emp1;  
29
```

<

Output :

Action Output

#	Time	Action	Message
✓ 22	10:31:33	TRUNCATE TABLE emp1	0 row(s) affected
✗ 23	10:31:43	SELECT * FROM emp LIMIT 0, 1000	Error Code: 1146. Table 'bank.emp' doesn't exist
✓ 24	10:31:50	SELECT * FROM emp1 LIMIT 0, 1000	0 row(s) returned
✓ 25	10:32:51	DROP TABLE emp1	0 row(s) affected

## Experiment No.: 3

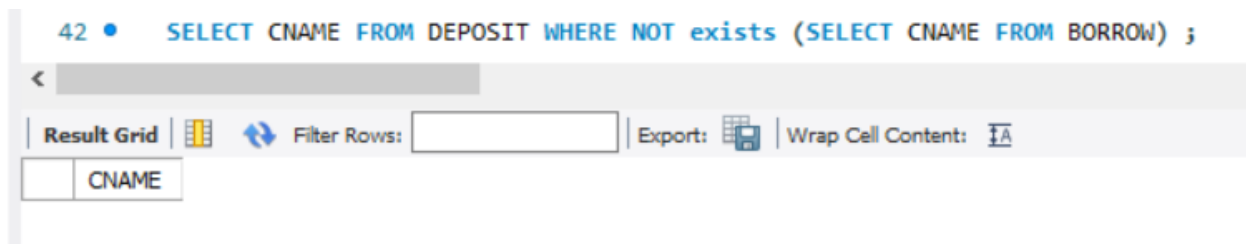
Aim To familiarize with set operations

### Question-

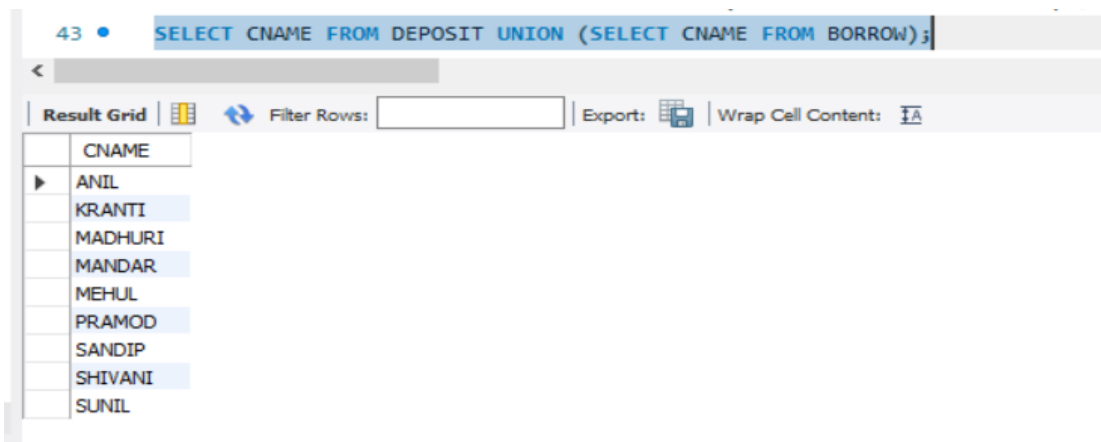
1. List all the customers who are depositors but not borrowers.
2. List all the customers who are both depositors and borrowers
3. List all the depositors having deposit in all the branches where Sunil is having Account
4. List all the customers living in city NAGPUR and having branch city BOMBAY or DELHI
5. List all the depositors living in city NAGPUR
6. List all the depositors living in the city NAGPUR and having branch in city BOMBAY
7. List the branch cities of Anil and Sunil
8. List the customers having deposit greater than 1000 and loan less than 10000.
9. List the cities of depositors having branch VRCE.
10. List the depositors having amount less than 1000 and living in the same city as Anil
11. List all the cities where branches of Anil and Sunil are locate
12. List the amount for the depositors living in the city where Anil is living

### PROCEDURE –

1. SELECT CNAME FROM DEPOSIT WHERE NOT exists (SELECT CNAME FROM BORROW) ;



2. SELECT CNAME FROM DEPOSIT UNION (SELECT CNAME FROM BORROW)



3. SELECT D1.CNAME FROM DEPOSIT D1 WHERE D1.BNAME IN (SELECT D2.BNAME FROM DEPOSIT D2 WHERE D2.CNAME = 'SUNIL' );

```
44 • SELECT D1.CNAME FROM DEPOSIT D1 WHERE D1.BNAME IN (SELECT D2.BNAME
45 FROM DEPOSIT D2 WHERE D2.CNAME = 'SUNIL' );
```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

CNAME
SUNIL

4. SELECT C1.CNAME FROM CUSTOMER C1,DEPOSIT D1, BRANCH B1 WHERE C1.CITY = 'NAGPUR' AND C1.CNAME = D1.CNAME AND D1.BNAME = B1.BNAME AND B1.CITY IN ('BOMBAY','DELHI');

```
46 • SELECT C1.CNAME FROM CUSTOMER C1,DEPOSIT D1, BRANCH B1 WHERE C1.CITY = 'NAGPUR'
```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

CNAME
MADHURI

5. SELECT Distinct(CUSTOMER.CNAME) from CUSTOMER,DEPOSIT WHERE City='NAGPUR';

```
48 • SELECT Distinct(CUSTOMER.CNAME) from CUSTOMER,DEPOSIT WHERE City='NAGPUR';
```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

CNAME
PRAMOD
MADHURI

6. SELECT C1.CNAME FROM CUSTOMER C1,DEPOSIT D1, BRANCH B1 WHERE C1.CITY = 'NAGPUR' AND C1.CNAME = D1.CNAME AND D1.BNAME = B1.BNAME AND B1.CITY IN ('BOMBAY');

```
49 • 'NAGPUR' AND C1.CNAME = D1.CNAME AND D1.BNAME = B1.BNAME AND B1.CITY IN ('BOM'
```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

CNAME
-------

7. SELECT B1.CITY FROM DEPOSIT D1, BRANCH B1 WHERE D1.BNAME = B1.BNAME AND D1.CNAME IN ('SUNIL','ANIL');



50 • DEPOSIT D1, BRANCH B1 WHERE D1.BNAME = B1.BNAME AND D1.CNAME IN ('SUNIL', 'ANIL')

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	CITY
▶	NAGPUR
	NAGPUR

8. SELECT DISTINCT D1.CNAME FROM deposit D1, borrow B1 WHERE D1.AMOUNT>1000 AND B1.AMOUNT

51 • T DISTINCT D1.CNAME FROM deposit D1, borrow B1 WHERE D1.AMOUNT>1000 AND B1.AMOU

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	CNAME
▶	MEHUL
	MADHURI
	PRAMOD
	SANDIP
	KRANTI
	MANDAR

9. SELECT B1.CITY FROM deposit D1, branch B1 WHERE D1.BNAME=B1.BNAME AND B1.BNAME='VRCE';

52 • SELECT B1.CITY FROM deposit D1, branch B1 WHERE D1.BNAME=B1.BNAME AND B1.BNAME

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	CITY
--	------

10. SELECT D1.CNAME FROM deposit D1, customer C1 WHERE AMOUNT<1000 AND C1.CITY=(C1.CNAME='ANIL');

52 • D1.CNAME FROM deposit D1, customer C1 WHERE AMOUNT<1000 AND C1.CITY=(C1.CNAME




Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	CNAME
▶	SUNIL
	SUNIL
	SUNIL
	SUNIL
	SUNIL
	SUNIL
	SUNIL
	SUNIL
	SUNIL

11. ELECT B1.CITY FROM BRANCH B1 WHERE B1.BNAME IN (SELECT D1.BNAME FROM DEPOSIT D1 WHERE D1.CNAME IN ('ANIL','SUNIL'));

52 • B1.BNAME IN (SELECT D1.BNAME FROM DEPOSIT D1 WHERE D1.CNAME IN ('ANIL','SUN

<




Result Grid  Filter Rows:  Export:  Wrap Cell Content: 

	CITY
▶	NAGPUR
	NAGPUR

12. SELECT DISTINCT(D1.CNAME),D1.AMOUNT ,C1.CITY FROM deposit D1, CUSTOMER C1, BRANCH B1 WHERE D1.CNAME=C1.CNAME AND C1.CITY IN(SELECT C2.CITY FROM customer C2 WHERE C2.CNAME='ANIL');

50 • SELECT DISTINCT(D1.CNAME),D1.AMOUNT ,C1.CITY FROM deposit D1, CUSTOMER

<

Result Grid  Filter Rows:  Export:  Wrap Cell Content: 

	CNAME	AMOUNT	CITY
▶	ANIL	1000	CALCUTTA

## EXPERIMENT NO. 4

### Aim-

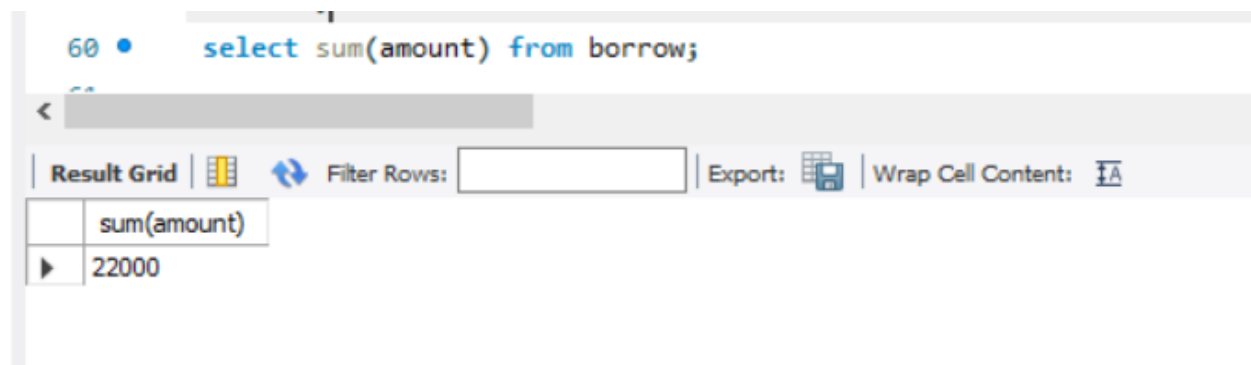
To familiarize with aggregate functions

### Questions-

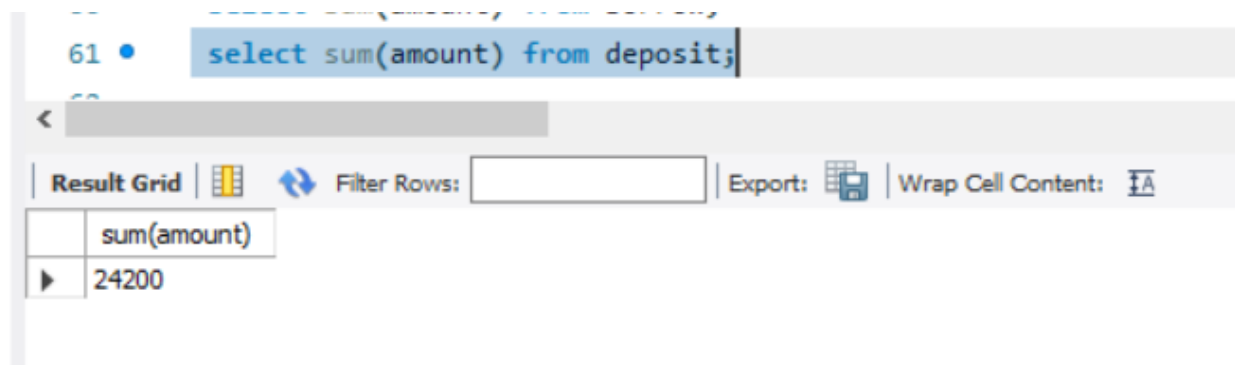
1. List total loan mysql>

```
SELECT SUM(AMOUNT)
```

```
FROM BORROW;
```

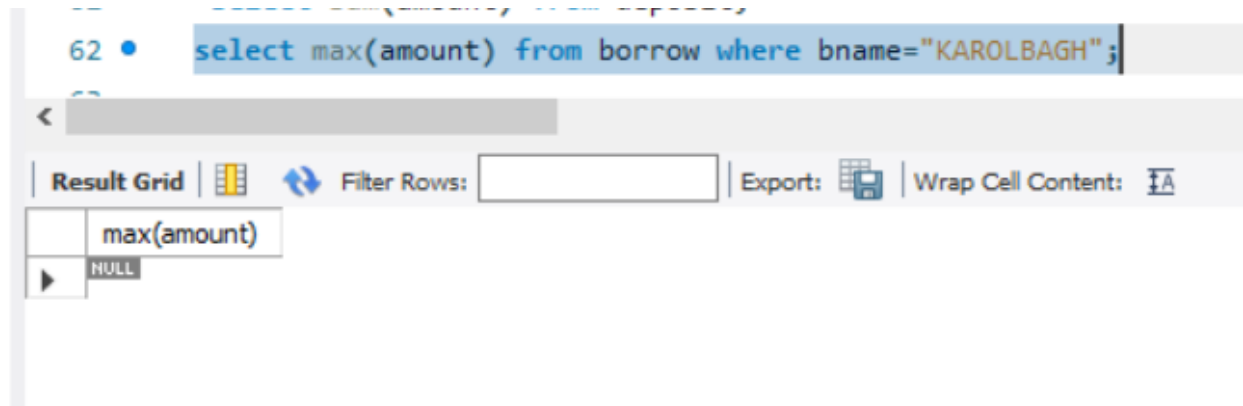


2. List total deposit



3. List total loan taken from KAROLBAGH branch




```
SELECT MAX(AMOUNT) FROM BORROW WHERE BNAME ='KAROLBAGH';
```



4. List total deposit of customers having account date later than 1-Jan-96

```
62 • select sum(amount) from deposit where adate>'1995-03-01';
```

<




Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	sum(amount)
▶	23200

5. List total deposit of customers living in city NAGPUR

```
63 • select sum(d1.amount) from deposit d1, customer c1 where c1.city="NAGPUR" and
```

<




Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	sum(d1.amount)
▶	4200

6. List maximum deposit of customer living in Bombay

```
64 • select max(d1.amount) from deposit d1, customer c1 where c1.city="BOMBAY"
```

<




Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	max(d1.amount)
▶	5000

7. List total deposit of customer having branch in BOMBAY

```
65 • select sum(amount) from deposit, branch where city="BOMBAY";
```

<

Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	sum(amount)
▶	48400

8. Count total number of branch cities

```
66 • select count(distinct(city)) from branch;
```

Result Grid | | Filter Rows:  | Export: | Wrap Cell Content:

	count(distinct(city))
▶	4

9.Count total number of customers cities

```
67 • select count(city) from customer;
```

Result Grid | | Filter Rows:  | Export: | Wrap Cell Content:

	count(city)
▶	10

10.Give branch names and branch wise deposit

```
68 • select bname,sum(amount) from deposit group by bname;
```




Result Grid | | Filter Rows:  | Export: | Wrap Cell Content:

	bname	sum(amount)
▶	AGNI	500
	ANDHERI	2000
	CHANDNI	1200
	KAROLBAGH	3500
	MG ROAD	3000
	NEHRU PALACE	5000
	POWAI	7000
	VRICE	2000

11.Give city wise name and branch wise deposit

69 • `select c1.city,sum(d1.amount) from customer c1,deposit d1 where`

< `70`




Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	city	sum(d1.amount)
▶	CALCUTTA	1000
	DELHI	500
	BARODA	3500
	NAGPUR	4200
	SURAT	2000
	BOMBAY	6000
	PATNA	7000

12. Give the branch wise loan of customer living in NAGPUR

70 • `select bname,sum(amount) from borrow,customer where city="NAGPUR" group`

< `71`




Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	bname	sum(amount)
▶	VRICE	18000
	AGNI	10000
	DHARAMPETH	6000
	ANDHERI	4000
	NEHRU PALACE	6000

13. Count total number of customers

71 • `select count(cname) from customer;`

< `72`





Result Grid |  Filter Rows:  | Export:  | Wrap Cell Content: 

	count(cname)
▶	10

14. Count total number of depositors branch wise

72 • `select bname,count(*) from deposit,customer where deposit.cname=c`

<





Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 

	bname	count(*)
▶	VRICE	2
	AGNI	1
	KAROLBAGH	1
	CHANDNI	1
	MG ROAD	1
	ANDHERI	1
	NEHRU PALACE	1
	POWAI	1

15.Count total number of depositors branch wise

72 • `select bname,count(*) from deposit,customer where deposit.cname=c`

<





Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 

	bname	count(*)
▶	VRICE	2
	AGNI	1
	KAROLBAGH	1
	CHANDNI	1
	MG ROAD	1
	ANDHERI	1
	NEHRU PALACE	1
	POWAI	1

16.Give maximum loan from branch VRCE

73 • `select max(amount) from borrow where bname="VRCE";`

<

Result Grid   Filter Rows:  | Export:  | Wrap Cell Content: 

	max(amount)
▶	NULL

17.Give the number of customers who are depositors as well as borrowers

## **EXPERIMENT NO. 5**

### **AIM**

To familiarize with join or cartesian product

### **Questions**

1. Give name of customers having living city

BOMBAY and branch city NAGPUR

2. Give names of customers having the same living city as their branch city

3. Give names of customers who are borrowers as well as depositors and having city

NAGPUR.

4. Give names of borrowers having deposit amount greater than 1000 and loan amount greater

than 2000.

5. Give names of depositors having the same branch as the branch of Sunil

6. Give names of borrowers having loan amount greater than the loan amount of Pramod

7. Give the name of the customer living in the city where branch of depositor Sunil is located.

8. Give branch city and living city of Pramod

9. Give branch city of Sunil and branch city of Anil



10. Give the living city of Anil and the living city of Sunil

### **Procedure**

1. SELECT D1.CNAME,D1.BNAME,C1.CNAME,C1.CITY,B1.CITY,B1.BNAME  
FROM DEPOSIT D1,CUSTOMER C1,BRANCH B1 WHERE C1.CITY = 'BOMBAY' AND  
B1.CITY = 'NAGPUR' AND D1.CNAME = C1.CNAME AND D1.BNAME = B1.BNAME;

	CNAME	BNAME	CNAME	CITY	CITY	BNAME

2. SELECT distinct(customer.CNAME), BRANCH.CITY FROM BRANCH, customer  
WHERE BRANCH.city = customer.city;
3. SELECT C1.CNAME FROM CUSTOMER C1,DEPOSIT D1,BORROW B1 WHERE  
C1.CITY='NAGPUR' AND C1.CNAME=D1.CNAME AND D1.CNAME = B1.CNAME;
4. SELECT BR1.CNAME, BR1.AMOUNT, D1.CNAME, D1.AMOUNT FROM  
BORROW BR1,DEPOSIT D1 WHERE D1.CNAME = BR1.CNAME AND D1.AMOUNT >  
1000 AND BR1.AMOUNT > 2000;
5. SELECT D1.CNAME FROM DEPOSIT D1 WHERE D1.BNAME IN (SELECT  
D2.BNAME FROM DEPOSIT D2 WHERE D2.CNAME = 'SUNIL');
6. SELECT BR1.CNAME,BR1.AMOUNT FROM BORROW BR1 WHERE  
BR1.AMOUNT > ALL (SELECT BR2.AMOUNT FROM BORROW BR2 WHERE  
BR2.CNAME = 'PRAMOD');
7. SELECT C.CNAME FROM CUSTOMER C WHERE C.CITY IN (SELECT B.CITY  
FROM BRANCH B WHERE B.BNAME IN (SELECT D.BNAME FROM DEPOSIT D  
WHERE D.CNAME='SUNIL'));
8. SELECT B1.CITY , C1.CITY FROM BRANCH B1,CUSTOMER C1, DEPOSIT D1  
WHERE C1.CNAME = 'PRAMOD' AND C1.CNAME = D1.CNAME AND D1.BNAME =  
B1.BNAME;
9. SELECT B1.CITY FROM DEPOSIT D1, BRANCH B1 WHERE D1.BNAME =  
B1.BNAME AND D1.CNAME IN ('SUNIL' , 'ANIL');
10. SELECT C1.CNAME, C1.CITY FROM CUSTOMER C1 WHERE C1.CNAME =  
'ANIL' OR C1.CNAME = 'SUNIL'

**Output Screenshot**

	CNAME	BNAME	CNAME	CITY	CITY	BNAME
--	-------	-------	-------	------	------	-------

	CNAME	CITY
▶	KRANTI	BOMBAY
	MADHURI	NAGPUR
	NAREN	BOMBAY
	PRAMOD	NAGPUR
	SHIVANI	BOMBAY
	SUNIL	DELHI

	CNAME
▶	MADHURI
	PRAMOD

	CNAME	AMOUNT	CNAME	AMOUNT
▶	MEHUL	5000	MEHUL	3500
	PRAMOD	8000	PRAMOD	3000
	KRANTI	3000	KRANTI	5000

	CNAME
▶	SUNIL

	CNAME
*	NULL

	CNAME	AMOUNT
--	-------	--------

	CITY	CITY
▶	BANGALORE	NAGPUR

	CITY
▶	NAGPUR

	CNAME	CITY
▶	ANIL	CALCUTTA
	SUNIL	DELHI
*	NULL	NULL

## EXPERIMENT NO. 6

### AIM

To familiarize with Group by and Having clause

### Questions

1. List the branches having sum of deposit more than 5000.

```
SELECT D.BNAME FROM DEPOSIT D, BRANCH B WHERE D.BNAME=B.BNAME AND
B.CITY='BOMBAY' GROUP BY D.BNAME HAVING SUM(D.AMOUNT)>5000;
```

75 • SELECT D.BNAME FROM DEPOSIT D, BRANCH B WHERE D.BNAME=B.BNAME AND B.CITY='BOMBAY'

< 75

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	BNAME
▶	POWAI

2. List the branches having sum of deposit more than 500 and located in city BOMBAY

```
SELECT D.BNAME FROM DEPOSIT D, BRANCH B ,Customer C WHERE
D.BNAME=B.BNAME and C.city="Bombay" GROUP BY D.BNAME HAVING
SUM(D.AMOUNT)>500 ;
```

76 • SELECT D.BNAME FROM DEPOSIT D, BRANCH B ,Customer C WHERE D.BNAME=B.

< 76

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	BNAME
▶	VRICE
	AGNI
	KAROLBAGH
	CHANDNI
	MG ROAD
	ANDHERI
	NEHRU PALACE
	POWAI

3. List the names of customers having deposited in the branches where the average deposit is more than 5000.

```
select CNAME from deposit where AMOUNT=(select AVG(Amount) from DEPOSIT
GROUP BY BNAME having AVG(Amount)>5000);
```

77 • `select CNAME from deposit where AMOUNT=(select AVG(Amount) from DEPOSIT GROUP BY BNAME having AVG(Amount)>5000);`

78

<

Result Grid | | Filter Rows:  | Export: | Wrap Cell Content:

	CNAME
▶	MANDAR

4. List the names of customers having maximum deposit

```
SELECT MAX(AMOUNT),CNAME FROM deposit;
```

89 • `SELECT d.bname FROM deposit d, branch b WHERE`  
 90 • `select cname from deposit where amount=(select`  
 91 • `SELECT MAX(amount),cname FROM deposit;`  
 92 • `SELECT d1.bname FROM deposit d1 GROUP BY d1.b`

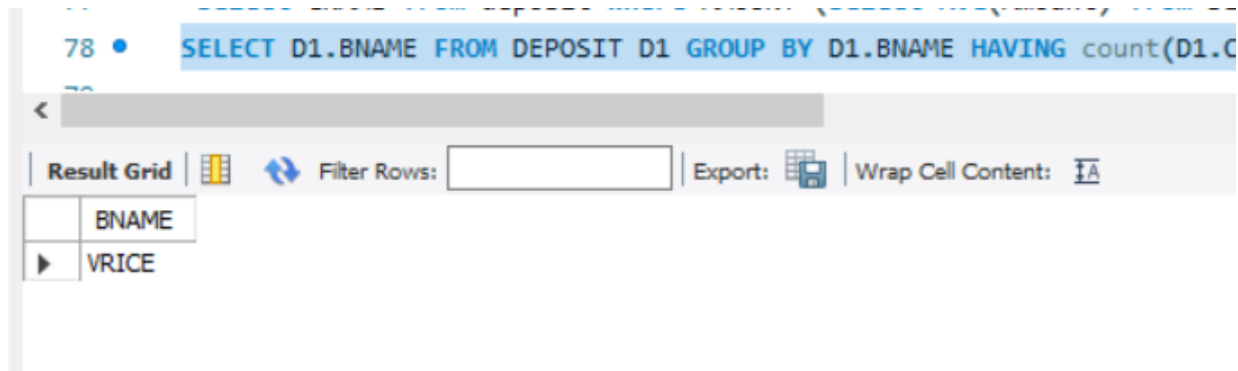
93 •

Result Grid | | Filter Rows:  | Export: | Wr

	MAX(amount)	cname
▶	12200	ANIL

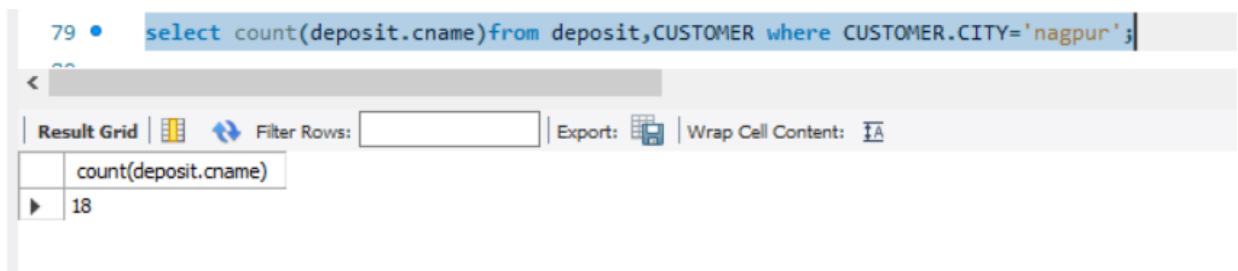
5. List the name of branch having highest number of depositors?

```
SELECT D1.BNAME FROM DEPOSIT D1 GROUP BY
D1.BNAME HAVING count(D1.CNAME) >= ALL (SELECT
count(D2.CNAME) FROM DEPOSIT D2 GROUP BY
D2.BNAME);
```



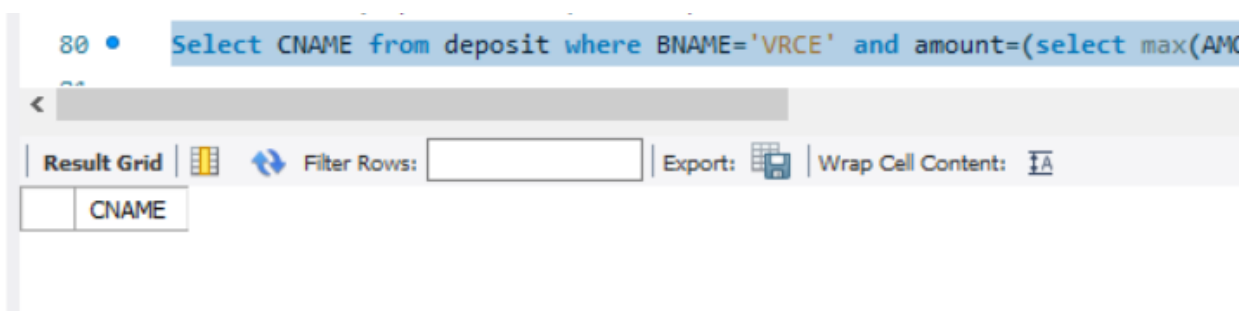
6.Count the number of depositors living in NAGPUR.

`select count(deposit.cname)from deposit,CUSTOMER where  
CUSTOMER.CITY='nagpur';`



7.Give names of customers in VRCE branch having more deposit than any other customer in same branch

`Select CNAME from deposit where BNAME='VRCE' and amount=(select max(AMOUNT) from  
deposit where BNAME='VRCE');`



8.Give the names of branch where number of depositors is more than 5

`SELECT BNAME from deposit GROUP BY BNAME HAVING COUNT(BNAME)>5;`

```
81 • SELECT BNAME from deposit GROUP BY BNAME HAVING COUNT(BNAME)>5;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
BNAME				

9. Give the names of cities in which the maximum number of branches are located

select C.CNAME ,count(B.BNAME) from CUSTOMER C inner join Branch B on

C.CNAME=B.BNAME group by C.Cname order by count(B.BName) DESC;

```
82 • (B.BNAME) from CUSTOMER C inner join Branch B on C.CNAME=B.BNAME g
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
CNAME	count(B.BNAME)			

10. Count the number of customers living in the city where branch is located

select count(b1.bname) From deposit d1 , borrow b1 , customer c1 Where

c1.cname=d1.cname and d1.cname=b1.cname and c1.city in (select city from customer);

```
83 • rrow b1 , customer c1 Where c1.cname=d1.cname and d1.cname=b1.cnam
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
count(b1.bname)				
6				

<b><u>EXPERIMENT NO.:7</u></b>
--------------------------------

**AIM**

To have familiarize with trigger functions

**Question**

Create a Trigger for employee table it will update another table salary while updating values

**OBJECTIVE**

To develop and execute a Trigger for After update/Delete/Insert operations on a table

**PROCEDURE**

step 1: start

step 2: initialize the trigger.

step 3: On update the trigger has to be executed.

step 4: execute the trigger procedure after updation

step 5: carryout the operation on the table to check for trigger execution.

step 6: stop

**PROGRAM**

Sql>

```
CREATE TABLE `employee` (
  `emp_id` int(11) NOT NULL,
  `emp_name` varchar(45) DEFAULT NULL,
  `dob` date DEFAULT NULL,
  `address` varchar(45) DEFAULT NULL,
  `designation` varchar(45) DEFAULT NULL,
  `mobile_no` int(11) DEFAULT NULL,
  `dept_no` int(11) DEFAULT NULL,
  `salary` int(11) DEFAULT NULL,
  PRIMARY KEY (`emp_id`)
);
```

Sql>

```

CREATE TABLE `salary` (
  `employee_id` int(11) NOT NULL,
  `old_sal` int(11) DEFAULT NULL,
  `new_sal` int(11) DEFAULT NULL,
  `rev_date` date DEFAULT NULL,
  PRIMARY KEY (`employee_id`)
);

Sql>

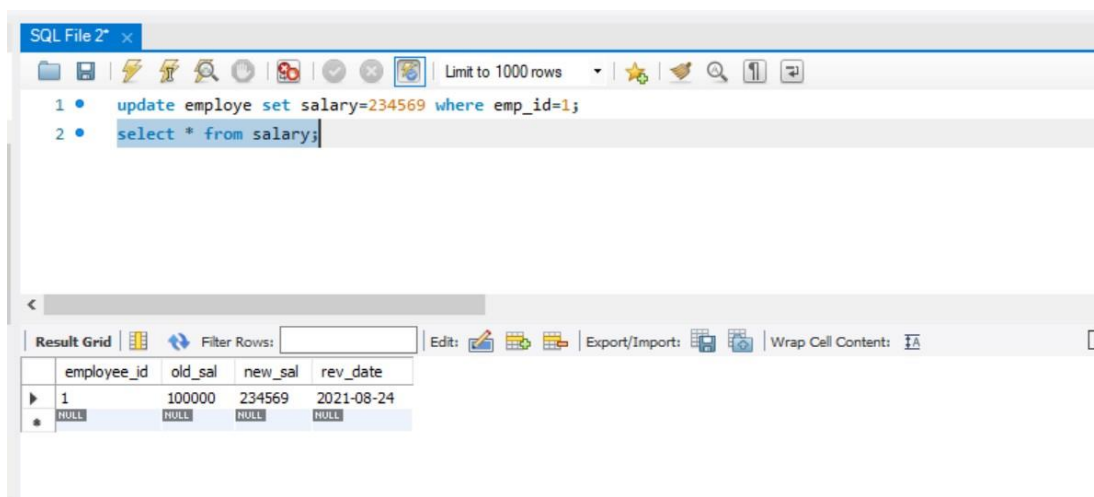
CREATE DEFINER=`root`@`localhost` TRIGGER
`employee_db`.`employee_AFTER_UPDATE` AFTER UPDATE ON `employee`
FOR EACH ROW
BEGIN
  if(new.salary != old.salary)
  then
    INSERT INTO salary (employee_id,old_sal,new_sal,rev_date) values
    (new.emp_id,old.salary,new.salary,sysdate());
  END if;
END

Sql>

update employee set salary=234569 where emp_id=1;
select * from salary;

```

## OUTPUT



The screenshot shows a SQL IDE window titled "SQL File 2". The SQL editor contains two queries:   
 1. `update employee set salary=234569 where emp_id=1;`   
 2. `select * from salary;`   
 The "Result Grid" at the bottom displays the output of the second query. It shows a table with four columns: `employee_id`, `old_sal`, `new_sal`, and `rev_date`. The first row contains the values 1, 100000, 234569, and 2021-08-24. A second row is partially visible with NULL values.

employee_id	old_sal	new_sal	rev_date
1	100000	234569	2021-08-24
NULL	NULL	NULL	NULL



## **EXPERIMENT NO. 8**

### **AIM**

To have familiarize with trigger functions

### **Question**

Create a Trigger for employe table it will update another table  
personal\_updatons while updating values

### **OBJECTIVE**

To develop and execute a Trigger for Before and After update/Delete/Insert  
operations on a table

### **PROCEDURE**

step 1: start

step 2: initialize the trigger.

step 3: On update the trigger has to be executed.

step 4: execute the trigger procedure after updation

step 5: carryout the operation on the table to check for trigger execution.

step 6: stop

### **PROGRAM**

sql>

```
CREATE TABLE `employe` (
  `emp_id` int(11) NOT NULL,
  `emp_name` varchar(45) DEFAULT NULL,
  `dob` date DEFAULT NULL,
  `address` varchar(45) DEFAULT NULL,
  `designation` varchar(45) DEFAULT NULL,
  `mobile_no` int(11) DEFAULT NULL,
  `dept_no` int(11) DEFAULT NULL,
  `salary` int(11) DEFAULT NULL,
  PRIMARY KEY (`emp_id`)
);
```

Sql>

```
CREATE TABLE `personal_updatons` (
```

```

`emp_id` int(11) NOT NULL,
`old_phoneno` int(11) DEFAULT NULL,
`new_phoneno` int(11) DEFAULT NULL,
`rev_date` date DEFAULT NULL,
PRIMARY KEY (`emp_id`)
);

Sql>

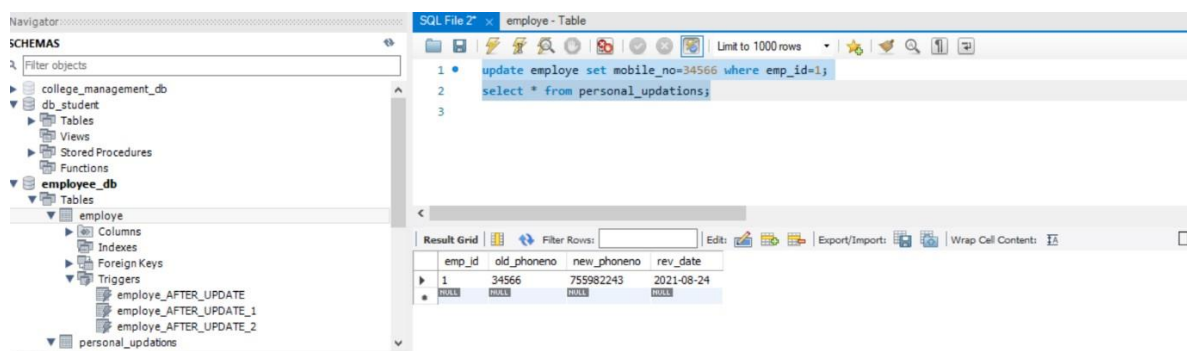
CREATE DEFINER=`root`@`localhost` TRIGGER
`employee_AFTER_UPDATE_1` AFTER UPDATE ON `employee` FOR EACH
ROW BEGIN
if(new.mobile_no != old.mobile_no)
then
INSERT INTO personal_updatons
(emp_id,old_phoneno,new_phoneno,rev_date) values
(new.emp_id,new.mobile_no,old.mobile_no,sysdate());
END if;
END

sql>

update employee set mobile_no=34566 where emp_id=4 ;
select * from personal_updatons;

```

## OUTPUT



The screenshot shows a SQL IDE interface. On the left, the 'SCHEMAS' pane displays a tree view of databases, including 'college\_management\_db', 'db\_student', and 'employee\_db'. The 'employee\_db' database is selected, and its tables are listed. The main window shows the SQL Editor with the following commands:

```

1 update employee set mobile_no=34566 where emp_id=4;
2 select * from personal_updatons;
3

```

Below the editor, the 'Result Grid' displays the output of the second command. It shows a table with four columns: 'emp\_id', 'old\_phoneno', 'new\_phoneno', and 'rev\_date'. The first row contains the values 1, 34566, 755982243, and 2021-08-24. The second row contains NULL, NULL, NULL, and NULL.

emp_id	old_phoneno	new_phoneno	rev_date
1	34566	755982243	2021-08-24
NULL	NULL	NULL	NULL

## **EXPERIMENT NO. 9**

### **AIM**

To have familiarize with trigger functions

### **Question**

Create a Trigger for employe table it will update another table promotions while updating values

### **OBJECTIVE**

To develop and execute a Trigger for Before and After update/Delete/Insert operations on a table

### **PROCEDURE**

step 1: start

step 2: initialize the trigger.

step 3: On update the trigger has to be executed.

step 4: execute the trigger procedure after updation

step 5: carryout the operation on the table to check for trigger execution.

step 6: stop

### **PROGRAM**

sql>

```
CREATE TABLE `employe` (
  `emp_id` int(11) NOT NULL,
  `emp_name` varchar(45) DEFAULT NULL,
  `dob` date DEFAULT NULL,
  `address` varchar(45) DEFAULT NULL,
  `designation` varchar(45) DEFAULT NULL,
  `mobile_no` int(11) DEFAULT NULL,
  `dept_no` int(11) DEFAULT NULL,
  `salary` int(11) DEFAULT NULL,
  PRIMARY KEY (`emp_id`)
);
```

Sql>

```
CREATE TABLE `promotions` (
```

```

`emp_id` int(11) NOT NULL,
`old_designation` varchar(11) DEFAULT NULL,
`new_designation` varchar(11) DEFAULT NULL,
`rev_date` date DEFAULT NULL,
PRIMARY KEY (`emp_id`)
);

sql>

CREATE DEFINER=`root`@`localhost` TRIGGER
`employee_AFTER_UPDATE_2` AFTER UPDATE ON `employee` FOR EACH
ROW BEGIN
if(new.designation != old.designation)
then
INSERT INTO promotions (emp_id,old_designation,new_designation,rev_date)
values (new.emp_id,new.designation,old.designation,sysdate());
END if;
END

sql>

update employee set designation='clk' where emp_id=1;

select * from promotions;

```

## OUTPUT

