

# **20MCA134-ADVANCED DBMS LAB**

## **RECORD**

Anjali C Abraham

Roll No: 18

RMCA-2020-S2

## **EXPERIMENT NO. 1**

### **AIM**

To study various DDL commands - CREATE

### **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

## OUTPUT

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

CREATE TABLE BRANCH (BNAME VARCHAR(20) PRIMARY KEY,CITY VARCHAR(30) CHECK (CITY IN

('NAGPUR','DELHI','BANGALORE','BOMBAY')) NOT NULL);

```
CREATE TABLE BORROW (LOANNO VARCHAR(8) CHECK (LOANNO LIKE 'L%')  
PRIMARY KEY,CNAME
```

```
  VARCHAR(15) REFERENCES CUSTOMER(CNAME) ,BNAME VARCHAR(20)  
REFERENCES
```

```
  BRANCH(BNAME),AMOUNT FLOAT(8) CHECK (AMOUNT>0) NOT NULL);
```

```
CREATE TABLE DEPOSIT (ACTNO VARCHAR(5) CHECK (ACTNO LIKE 'D%')  
PRIMARY KEY ,CNAME
```

```
  VARCHAR(15) REFERENCES CUSTOMER(CNAME) ,BNAME VARCHAR(20)  
REFERENCES
```

```
  BRANCH(BNAME),AMOUNT FLOAT(8) CHECK (AMOUNT>0) NOT NULL, ADATE  
DATE);
```

```
INSERT INTO CUSTOMER VALUES('ANIL','CALCUTTA');
```

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

```
INSERT INTO CUSTOMER VALUES('MEHUL','BARODA');
```

```
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('KRANTI','BOMBAY');
```

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

```
INSERT INTO BRANCH VALUES('VRCE','NAGPUR');
```

```
INSERT INTO BRANCH VALUES('AJNI','NAGPUR');
```

```
INSERT INTO BRANCH VALUES('KAROLBAGH','DELHI');
```

```
INSERT INTO BRANCH VALUES('CHANDNI','DELHI');
```

```
INSERT INTO BRANCH VALUES('DHARAMPETH','NAGPUR');
```

```
INSERT INTO BRANCH VALUES('MG ROAD','BANGALORE');
```

```
INSERT INTO BRANCH VALUES('ANDHERI','BOMBAY');
```

```
INSERT INTO BRANCH VALUES('NEHRU PALACE','DELHI');
```

```
INSERT INTO BRANCH VALUES('POWAI','BOMBAY');
```

```
INSERT INTO DEPOSIT VALUES('D100','ANIL','VRCE',1000.00,'1-MAR-95');
```

```
INSERT INTO DEPOSIT VALUES('D101','SUNIL','ANJNI',500.00,'4-JAN-96');
```

INSERT INTO DEPOSIT VALUES('D102','MEHUL','KAROLBAGH',3500.00,'17-NOV-95');

INSERT INTO DEPOSIT VALUES('D104','MADHURI','CHANDNI',1200.00,'17-DEC-95');

INSERT INTO DEPOSIT VALUES('D105','PRAMOD','MG ROAD',3000.00,'27-MAR-96');

INSERT INTO DEPOSIT VALUES('D106','SANDIP','ANDHERI',2000.00,'31-MAR-96');

INSERT INTO DEPOSIT VALUES('D107','SHIVANI','VIRAR',1000.00,'5-SEP-95');

INSERT INTO DEPOSIT VALUES('D108','KRANTI','NEHRU PLACE',5000.00,'2-JUL-95');

INSERT INTO DEPOSIT VALUES('D109','MINU','POWAI',7000.00,'10-AUG-95');

INSERT INTO BORROW VALUES('L201','ANIL','VRCE',1000.00);

INSERT INTO BORROW VALUES('L206','MEHUL','AJNI',5000.00);

INSERT INTO BORROW VALUES('L311','SUNIL','DHARAMPETH',3000.00);

INSERT INTO BORROW VALUES('L321','MADHURI','ANDHERI',2000.00);

INSERT INTO BORROW VALUES('L371','PRAMOD','VIRAR',8000.00);

INSERT INTO BORROW VALUES('L481','KRANTI','NEHRU PLACE',3000.00);

## **EXPERIMENT NO. 2**

### **AIM**

To familiarize with selecting data from single table

### **Questions**

1. List all data from table deposite
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 deposite
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 l205
15. Give names of depositors having account at VRCE
16. Give names of all branched 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

### **SQL Commands**

SELECT\*FROM DEPOSIT;

SELECT\*FROM BORROW;

SELECT\*FROM CUSTOMER;

SELECT\*FROM BRANCH;

```

SELECT ACTNO,AMOUNT FROM DEPOSIT;

SELECT CNAME,ACTNO FROM DEPOSIT;

SELECT CNAME FROM CUSTOMER;

SELECT BNAME FROM BRANCH;

SELECT CNAME FROM BORROW;

SELECT CNAME FROM CUSTOMER WHERE CITY='NAGPUR';

SELECT CNAME from DEPOSIT where AMOUNT>4000;

SELECT ADATE FROM DEPOSIT where CNAME='ANIL';

SELECT BNAME from BRANCH where CITY='Bombay';

SELECT * from BORROW;

SELECT CNAME from BORROW where LOANNO='L205';

SELECT CNAME from DEPOSIT WHERE BNAME='VRCE';

SELECT BNAME from BRANCH WHERE CITY='Delhi';

SELECT CNAME from DEPOSIT WHERE ADATE='1996-12-1';

SELECT ACTNO,AMOUNT from DEPOSIT WHERE ADATE BETWEEN '1996-12-1'
AND '1996-05-1';

SELECT CITY from BRANCH WHERE BNAME='KAROLBAGH';

SELECT * from customer join borrow on customer.cname=borrow.cname join deposit on
deposit.cname=borrow.cname WHERE customer.cname='ANIL';

```

## OUTPUT

```

D100|ANIL|VRCE|1000.0|1-MAR-95
D101|SUNIL|ANJNI|500.0|4-JAN-96
D102|MEHUL|KAROLBAGH|3500.0|17-NOV-95
D104|MADHURI|CHANDNI|1200.0|17-DEC-95
D105|PRAMOD|MG ROAD|3000.0|27-MAR-96
D106|SANDIP|ANDHERI|2000.0|31-MAR-96
D107|SHIVANI|VIRAR|1000.0|5-SEP-95
D108|KRANTI|NEHRU PLACE|5000.0|2-JUL-95
D109|MINU|POWAI|7000.0|10-AUG-95
L201|ANIL|VRCE|1000.0
L206|MEHUL|AJNI|5000.0
L311|SUNIL|DHARAMPETH|3000.0
L321|MADHURI|ANDHERI|2000.0
L371|PRAMOD|VIRAR|8000.0
L481|KRANTI|NEHRU PLACE|3000.0
ANIL|CALCUTTA
SUNIL|DELHI
MEHUL|BARODA
MANDAR|PATNA
MADHURI|NAGPUR
PRAMOD|NAGPUR
SANDIP|SURAT
SHIVANI|BOMBAY
KRANTI|BOMBAY
NAREN|BOMBAY
VRCE|NAGPUR
AJNI|NAGPUR

```

AJNI | NAGPUR  
KAROLBAGH | DELHI  
CHANDNI | DELHI  
DHARAMPETH | NAGPUR  
MG ROAD | BANGALORE  
ANDHERI | BOMBAY  
NEHRU PALACE | DELHI  
POWAI | BOMBAY  
D100 | 1000.0  
D101 | 500.0  
D102 | 3500.0  
D104 | 1200.0  
D105 | 3000.0  
D106 | 2000.0  
D107 | 1000.0  
D108 | 5000.0  
D109 | 7000.0  
ANIL | D100  
SUNIL | D101  
MEHUL | D102  
MADHURI | D104  
PRAMOD | D105  
SANDIP | D106  
SHIVANI | D107  
KRANTI | D108  
MINU | D109  
ANIL

Activate Wind

NEHRU PALACE  
POWAI  
VRCE  
ANIL  
MEHUL  
SUNIL  
MADHURI  
PRAMOD  
KRANTI  
MADHURI  
PRAMOD  
KRANTI  
MINU  
1-MAR-95  
L201 | ANIL | VRCE | 1000.0  
L206 | MEHUL | AJNI | 5000.0  
L311 | SUNIL | DHARAMPETH | 3000.0  
L321 | MADHURI | ANDHERI | 2000.0  
L371 | PRAMOD | VIRAR | 8000.0  
L481 | KRANTI | NEHRU PLACE | 3000.0  
ANIL  
DELHI  
ANIL | CALCUTTA | L201 | ANIL | VRCE | 1000.0 | D100 | ANIL | VRCE | 1000.0 | 1-MAR-95



## **EXPERIMENT NO. 3**

### **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

insert 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	

2.Modifying the structure of tables

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

b.Dropping a column from a table: sal

c.Modifying existing column :ename varchar2(15)

d.Renaming the tables: emp to emp1

e.truncating the tables:emp1

f.Destroying tables:emp

3.Create a table stud with sname varchar2(20) primary key , rollno number(10) not null,dob date not null

4.Create a table student as regno number (6), mark number (3) check constraint (mark >=0 and mark

<=100));

In table student add check constraint(length(regno<=4))

5.Create a table cust with(custid number(6) constraint unique, name char(10)

6. Refer the table “stud” in table “ student”

## **OUTPUT**

```
CREATE TABLE emp(EMPNO INT(4) PRIMARY KEY, ENAME CHAR(10), HIREDATE DATE, SALARY INT(5),
```

```
COMMISSION INT(5));
```

```
INSERT INTO emp(EMPNO,ENAME,HIREDATE,SALARY) VALUES
```

```
(101,'RAMESH','1980-01-17',5000),(102,'AJAY','1985-07-05',5000),
```

```
(103,'RAVI','1981-08-12',1500),(104,'Nikesh','1983-03-03',3000),
```

```
(105,'Ravi','1985-07-05',3000);
```

```
ALTER TABLE emp ADD sal int;
```

```
ALTER TABLE emp DROP COLUMN sal;
```

```
ALTER TABLE emp MODIFY ename VARCHAR(15);
```

```
RENAME table emp to emp1;
```

```
TRUNCATE TABLE emp1;
```

```
DROP TABLE emp1;
```

```
CREATE TABLE stud(sname varchar(20) PRIMARY KEY, rollno int NOT NULL, dob date NOT NULL);
```

```
SELECT * FROM stud;
```

```
Create table student (regno int, mark int(3) constraint b check (mark >=0 and mark <=100));
```

```
SELECT * from STUDENT;
```

```
Alter table student add regno check (length(regno)<=4));
```

```
CREATE TABLE cust(custid number(6) UNIQUE, name char(10));
```

OUTPUT:

```
mysql> CREATE TABLE emp(EMPNO NUMBER(4) PRIMARY KEY,
```

```
-> ENAME CHAR(10),
```

```
-> HIREDATE DATE,
```

```
-> SALARY NUMBER(5),
```

```
-> COMMISSION NUMBER(5));
```

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your

MySQL server version for the right syntax to use near 'NUMBER(4) PRIMARY KEY,

ENAME CHAR(10),

HIREDATE ' at line 1

```
mysql> CREATE TABLE emp(EMPNO INT(4) PRIMARY KEY,  
-> ENAME CHAR(10),  
-> HIREDATE DATE,  
-> SALARY NUMBER(5),  
-> COMMISSION NUMBER(5));
```

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your

MySQL server version for the right syntax to use near 'NUMBER(5),  
COMMISSION NUMBER(5))' at line 4

```
mysql> CREATE TABLE emp(EMPNO INT(4) PRIMARY KEY,  
-> ENAME CHAR(10),  
-> HIREDATE DATE,  
-> SALARY INT(5),  
-> COMMISSION NUMBER(5));
```

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your

MySQL server version for the right syntax to use near 'NUMBER(5))' at line 5

```
mysql> CREATE TABLE emp(EMPNO INT(4) PRIMARY KEY,  
-> ENAME CHAR(10),  
-> HIREDATE DATE,  
-> SALARY INT(5),  
-> COMMISSION INT(5));
```

Query OK, 0 rows affected, 3 warnings (0.04 sec)

```
mysql> INSERT INTO emp VALUES(101,'RAMESH','17-01-1980',5000);
```

ERROR 1136 (21S01): Column count doesn't match value count at row 1

```
mysql> INSERT INTO emp VALUES(102,'AJAY','05-07-1985',5000,500);
```

ERROR 1292 (22007): Incorrect date value: '05-07-1985' for column 'HIREDATE' at row 1

```
mysql> INSERT INTO emp VALUES(103,'RAVI','12-08-1981',1500);
```

ERROR 1136 (21S01): Column count doesn't match value count at row 1

```
mysql> INSERT INTO emp VALUES(104,'03-03-1983',3000,700);
```

ERROR 1136 (21S01): Column count doesn't match value count at row 1

```
mysql> INSERT INTO emp VALUES(101,'RAMESH','17-01-1980',5000);
```

ERROR 1136 (21S01): Column count doesn't match value count at row 1

```
mysql> INSERT INTO emp(EMPNO,ENAME,HIREDATE,SALARY)
VALUES(101,'RAMESH','17-01-1980',5000);
```

```
mysql> INSERT INTO emp(EMPNO,ENAME,HIREDATE,SALARY)
VALUES(101,'RAMESH','1980-01-17',5000);
```

Query OK, 1 row affected (0.01 sec)

```
mysql> INSERT INTO emp(EMPNO,ENAME,HIREDATE,SALARY,COMMISSION)
VALUES(102,'AJAY','05-07-1985',5000,500);
```

```
mysql> INSERT INTO emp(EMPNO,ENAME,HIREDATE,SALARY,COMMISSION)
VALUES(102,'AJAY','1985-07-05',5000,500);
```

Query OK, 1 row affected (0.00 sec)

```
mysql> INSERT INTO emp(EMPNO,ENAME,HIREDATE,SALARY)
VALUES(103,'RAVI','1981-08-12',1500);
```

Query OK, 1 row affected (0.01 sec)

```
mysql> INSERT INTO emp(EMPNO,HIREDATE,SALARY,COMMISSION)
VALUES(104,'Nikesh','1983-03-03',3000,700);
```

```
mysql> INSERT INTO emp(EMPNO,ENAME,HIREDATE,SALARY,COMMISSION)
VALUES(104,'Nikesh','1983-03-03',3000,700);
```

Query OK, 1 row affected (0.01 sec)

```
mysql> INSERT INTO emp(EMPNO,ENAME,HIREDATE,SALARY)
VALUES(105,'Ravi','1985-07-05',3000);
```

Query OK, 1 row affected (0.01 sec)

```
mysql> ALTER TABLE emp ADD sal number(7,2);
```

```
mysql> ALTER TABLE emp ADD sal int;
```

Query OK, 0 rows affected (0.03 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> select * from emp
```

```
-> select * from emp;
```

MySQL server version for the right syntax to use near 'select \* from emp' at line 2

```
mysql> select * from emp;
```

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

5 rows in set (0.00 sec)

mysql> ALTER TABLE emp DROP COLUMN sal;

Query OK, 0 rows affected (0.06 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> select \* from emp;

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

5 rows in set (0.00 sec)

mysql> ALTER TABLE emp MODIFY ename VARCHAR(15);

Query OK, 5 rows affected (0.06 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> select \* from emp;

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

5 rows in set (0.00 sec)

mysql> RENAME table emp to emp1;

Query OK, 0 rows affected (0.02 sec)

mysql> select \* from emp1;

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

5 rows in set (0.00 sec)

mysql> TRUNCATE TABLE emp1;

Query OK, 0 rows affected (0.03 sec)

mysql> select \* from emp1;

Empty set (0.00 sec)

mysql> desc emp1;

Field	Type	Null	Key	Default	Extra
EMPNO	int	NO	PRI	NULL	
ename	varchar(15)	YES		NULL	
HIREDATE	date	YES		NULL	
SALARY	int	YES		NULL	
COMMISSION	int	YES		NULL	

5 rows in set (0.01 sec)

mysql> DROP TABLE emp1;

Query OK, 0 rows affected (0.03 sec)

mysql> desc emp1;

ERROR 1146 (42S02): Table 'dbmsassignment.emp1' doesn't exist

mysql>

mysql> CREATE TABLE stud(sname varchar(20) PRIMARY KEY,

-> rollno NOT NULL,

-> dob date NOT NULL);

MySQL server version for the right syntax to use near 'NOT NULL,

dob date NOT NULL)' at line 2

mysql> CREATE TABLE stud(sname varchar(20) PRIMARY KEY,rollno int NOT NULL,  
dob date NOT NULL);

Query OK, 0 rows affected (0.04 sec)

mysql> Create table student (regno int,mark int(3) constraint b check (mark >=0 and  
mark<=100));

Query OK, 0 rows affected, 1 warning (0.03 sec)

mysql> select \* from student;

Empty set (0.02 sec)

mysql> desc student;

Field	Type	Null	Key	Default	Extra
regno	int	YES		NULL	
mark	int	YES		NULL	

2 rows in set (0.01 sec)

mysql> Alter table student add constraint b2 check (length(regno)<=4));

mysql> select \* from student;

Empty set (0.00 sec)

mysql> desc student;

Field	Type	Null	Key	Default	Extra
regno	int	YES		NULL	
mark	int	YES		NULL	

## **EXPERIMENT NO. 4**

### **AIM**

To familiarize with aggregate functions

### **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

### **OUTPUT**

```
mysql> SELECT SUM(AMOUNT) FROM BORROW;
```

SUM(AMOUNT)
22000.00

1 row in set (0.00 sec)

```
mysql> SELECT SUM(AMOUNT) FROM DEPOSIT;
```

SUM(AMOUNT)
28700.00

1 row in set (0.00 sec)



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

MAX(AMOUNT)
NULL

1 row in set (0.00 sec)

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

SUM(AMOUNT)
27700.00

1 row in set (0.00 sec)

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

SUM(D1.AMOUNT)
4200.00

1 row in set (0.00 sec)

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

MAX(D1.AMOUNT)
7000.00

1 row in set (0.00 sec)

mysql> SELECT SUM(AMOUNT) from deposit,BRANCH where city='BOMBAY';

SUM(AMOUNT)
57400.00

1 row in set (0.00 sec)

mysql> SELECT COUNT(DISTINCT(CITY)) FROM BRANCH ;

COUNT(DISTINCT(CITY))
4

1 row in set (0.00 sec)

mysql> SELECT count(city) from CUSTOMER;

count(city)
10

1 row in set (0.00 sec)

mysql> SELECT BNAME , SUM(AMOUNT) FROM DEPOSIT GROUP BY BNAME;

BNAME	SUM(AMOUNT)
VRCE	1000.00
AJNI	5000.00
KAROLBAGH	3500.00
CHANDNI	1200.00
M.G.ROAD	3000.00
ANDHERI	2000.00
VIRAR	1000.00

NEHRU PLACE	5000.00
POWAI	7000.00

9 rows in set (0.00 sec)

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

CITY	SUM(D1.AMOUNT)
CULCUTTA	1000.00
DELHI	5000.00
BARODA	3500.00
NAGPUR	4200.00
SURAT	2000.00
BOMBAY	13000.00

6 rows in set (0.00 sec)

mysql> SELECT BNAME , SUM(AMOUNT) FROM BORROW,CUSTOMER WHERE city  
='NAGPUR' GROUP BY BNAME;

BNAME	SUM(AMOUNT)
VRCE	2000.00
AJNI	10000.00
DHARAMPETH	6000.00
ANDHERI	4000.00
VIRAR	16000.00
NEHRU PLACE	6000.00

6 rows in set (0.00 sec)

mysql> SELECT count(cname) from CUSTOMER;

count(cname)
10

1 row in set (0.00 sec)

mysql> select BName, count(\*) from DEPOSIT, CUSTOMER where deposit.CNAME =  
CUSTOMER.CNAME group by BNAME;

BName	count(*)
VRCE	1
AJNI	1
KAROLBAGH	1
CHANDNI	1
M.G.ROAD	1
ANDHERI	1
VIRAR	1
NEHRU PLACE	1
POWAI	1

9 rows in set (0.00 sec)

```
mysql> select BName, count(*) from DEPOSIT, CUSTOMER where deposit.CNAME = CUSTOMER.CNAME group by BNAME;
```

BName	count(*)
VRCE	1
AJNI	1
KAROLBAGH	1
CHANDNI	1
M.G.ROAD	1
ANDHERI	1
VIRAR	1
NEHRU PLACE	1
POWAI	1

9 rows in set (0.00 sec)

```
mysql> SELECT MAX(AMOUNT) FROM BORROW WHERE BNAME ='VRCE';
```

MAX(AMOUNT)
1000.00

1 row in set (0.00 sec)

```
mysql> SELECT COUNT(DISTINCT (CNAME)) FROM CUSTOMER WHERE CNAME IN ((SELECT CNAME FROM DEPOSIT) INTERSECT (SELECT CNAME FROM BORROW));
```

(No column name)
3

1 row in set (0.00 sec)

```
mysql> SELECT SUM(AMOUNT) FROM BORROW;
```

SUM(AMOUNT)
22000.00

1 row in set (0.00 sec)

```
mysql> SELECT SUM(AMOUNT) FROM DEPOSIT;
```

SUM(AMOUNT)
28700.00

1 row in set (0.00 sec)

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

SUM(AMOUNT)
27700.00

1 row in set (0.00 sec)

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

SUM(D1.AMOUNT)
4200.00

1 row in set (0.00 sec)

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

MAX(D1.AMOUNT)
7000.00

1 row in set (0.00 sec)

```
mysql> SELECT SUM(AMOUNT) from deposit,BRANCH where city='BOMBAY';
```

SUM(AMOUNT)
57400.00

1 row in set (0.00 sec)

```
mysql> SELECT COUNT(DISTINCT(CITY)) FROM BRANCH ;
```

COUNT(DISTINCT(CITY))
4

1 row in set (0.00 sec)

```
mysql> SELECT count(city) from CUSTOMER;
```

count(city)
10

1 row in set (0.00 sec)

```
mysql> SELECT BNAME , SUM(AMOUNT) FROM DEPOSIT GROUP BY BNAME;
```

BNAME	SUM(AMOUNT)
VRCE	1000.00
AJNI	5000.00
KAROLBAGH	3500.00
CHANDNI	1200.00
M.G.ROAD	3000.00
ANDHERI	2000.00
VIRAR	1000.00
NEHRU PLACE	5000.00
POWAI	7000.00

9 rows in set (0.00 sec)

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

CITY	SUM(D1.AMOUNT)
CULCUTTA	1000.00
DELHI	5000.00
BARODA	3500.00
NAGPUR	4200.00
SURAT	2000.00
BOMBAY	13000.00

6 rows in set (0.00 sec)

```
mysql> SELECT BNAME , SUM(AMOUNT) FROM BORROW,CUSTOMER WHERE city
='NAGPUR' GROUP BY BNAME;
```

BNAME	SUM(AMOUNT)
VRCE	2000.00
AJNI	10000.00
DHARAMPETH	6000.00
ANDHERI	4000.00
VIRAR	16000.00
NEHRU PLACE	6000.00

6 rows in set (0.00 sec)

```
mysql> SELECT count(cname) from CUSTOMER;
```

count(cname)
10

1 row in set (0.00 sec)

```
mysql> select BName, count(*) from DEPOSIT, CUSTOMER where deposit.CNAME =
CUSTOMER.CNAME group by BNAME;
```

BName	count(*)
VRCE	1
AJNI	1
KAROLBAGH	1
CHANDNI	1
M.G.ROAD	1
ANDHERI	1
VIRAR	1
NEHRU PLACE	1
POWAI	1

9 rows in set (0.00 sec)

```
mysql> select BName, count(*) from DEPOSIT, CUSTOMER where deposit.CNAME =
CUSTOMER.CNAME group by BNAME;
```

BName	count(*)
VRCE	1
AJNI	1
KAROLBAGH	1
CHANDNI	1
M.G.ROAD	1
ANDHERI	1
VIRAR	1
NEHRU PLACE	1
POWAI	1

9 rows in set (0.00 sec)

```
mysql> SELECT MAX(AMOUNT) FROM BORROW WHERE BNAME ='VRCE';
```

MAX(AMOUNT)
1000.00

1 row in set (0.00 sec)

```
mysql> SELECT COUNT(DISTINCT (CNAME)) FROM CUSTOMER WHERE CNAME
IN ((SELECT CNAME FROM DEPOSIT) INTERSECT (SELECT CNAME FROM
BORROW));
```

(No column name)
3

1 row in set (0.00 sec)

## **EXPERIMENT NO. 5**

### **AIM**

To familiarize with set operations

### **Questions**

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

### **OUTPUT**

```
mysql> SELECT CNAME FROM DEPOSIT MINUS (SELECT CNAME FROM BORROW);
```

Empty set (0.00 sec)

```
mysql> SELECT CNAME FROM DEPOSIT UNION (SELECT CNAME FROM BORROW);
```

CNAME
ANIL
SUNIL
MEHUL
MADHURI
PRAMOD
SANDIP
SHIVANI
KRANTI
NAREN

9 rows in set (0.01 sec)

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

CNAME
SUNIL

1 row in set (0.00 sec)

```
mysql> 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');
```

CNAME
MADHURI

1 row in set (0.01 sec)

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

CNAME
PRAMOD
MADHURI

2 rows in set (0.00 sec)

```
mysql> 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');
```

Empty set (0.00 sec)

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

CITY
NAGPUR
NAGPUR

2 rows in set (0.00 sec)

```
mysql> SELECT DISTINCT D1.CNAME FROM deposit D1, borrow B1 WHERE D1.AMOUNT>1000 AND B1.AMOUNT<10000;
```

CNAME
SUNIL
MEHUL
MADHURI
PRAMOD
SANDIP
KRANTI
NAREN

7 rows in set (0.00 sec)



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

CITY
NAGPUR

1 row in set (0.00 sec)

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

Empty set, 10 warnings (0.00 sec)

```
mysql> SELECT B1.CITY FROM BRANCH B1 WHERE B1.BNAME IN (SELECT  
D1.BNAME FROM DEPOSIT D1 WHERE D1.CNAME IN ('ANIL','SUNIL'));
```

CITY
NAGPUR
NAGPUR

2 rows in set (0.00 sec)

```
mysql> 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');
```

CNAME	AMOUNT	CITY
ANIL	1000.00	CULCUTTA

1 row in set (0.00 sec)

## **EXPERIMENT NO. 6**

### **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

### **OUTPUT**

```
mysql> 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;
```

Empty set (0.00 sec)

```
mysql> SELECT distinct(customer.CNAME), BRANCH.CITY FROM BRANCH, customer
WHERE BRANCH.city = customer.city;
```

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

6 rows in set (0.00 sec)

```
mysql> SELECT C1.CNAME FROM CUSTOMER C1,DEPOSIT D1,BORROW B1 WHERE
C1.CITY='NAGPUR' AND C1.CNAME=D1.CNAME AND D1.CNAME = B1.CNAME;
```

CNAME
-------

MADHURI
PRAMOD

2 rows in set (0.00 sec)

```
mysql> 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;
```

CNAME	AMOUNT	CNAME	AMOUNT
SUNIL	3000.00	SUNIL	5000.00
MEHUL	5000.00	MEHUL	3500.00
PRAMOD	8000.00	PRAMOD	3000.00
KRANTI	3000.00	KRANTI	5000.00

4 rows in set (0.00 sec)

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

CNAME
SUNIL

1 row in set (0.01 sec)

```
mysql> SELECT BR1.CNAME,BR1.AMOUNT FROM BORROW BR1 WHERE
BR1.AMOUNT > ALL (SELECT BR2.AMOUNT FROM BORROW BR2 WHERE
BR2.CNAME = 'PRAMOD');
```

Empty set (0.00 sec)

mysql>

```
mysql> 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'));
```

CNAME
MADHURI
PRAMOD

2 rows in set (0.00 sec)

```
mysql> 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;
```

Empty set (0.00 sec)

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

CITY
NAGPUR
NAGPUR

2 rows in set (0.00 sec)

```
mysql> SELECT C1.CNAME, C1.CITY FROM CUSTOMER C1 WHERE C1.CNAME =  
'ANIL' OR C1.CNAME = 'SUNIL';
```

CNAME	CITY
ANIL	CULCUTTA
SUNIL	DELHI

2 rows in set (0.00 sec)

## **EXPERIMENT NO. 7**

### **AIM**

To familiarize with Group by and Having clause

### **Questions**

- 1.List the branches having sum of deposit more than 5000.
- 2.List the branches having sum of deposit more than 500 and located in city BOMBAY
- 3.List the names of customers having deposited in the branches where the average deposit is more than 5000.
- 4.List the names of customers having maximum deposit
- 5.List the name of branch having highest number of depositors?
- 6.Count the number of depositors living in NAGPUR.
- 7.Give names of customers in VRCE branch having more deposite than any other customer in same branch
- 8.Give the names of branch where number of depositors is more than 5
- 9.Give the names of cities in which the maximum number of branches are located
- 10.Count the number of customers living in the city where branch is located

### **OUTPUT**

```
mysql> 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;
```

BNAME
POWAI

1 row in set (0.06 sec)

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

BNAME
POWAI

1 row in set (0.00 sec)

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

CNAME
NAREN

1 row in set (0.00 sec)

mysql> SELECT MAX(AMOUNT),CNAME FROM deposit;

CNAME	AMOUNT
ANIL	7000.00

1 row in set (0.00 sec)

mysql> 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);

BNAME
CHANDNI
KAROLBAGH
M.G.ROAD
VRCE

4 row in set (0.10 sec)

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

count(deposit.cname)
18

1 row in set (0.00 sec)

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

CNAME
ANIL

1 row in set (0.00 sec)

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

Empty set (0.00 sec)

mysql> 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;

Empty set (0.00 sec)

mysql> 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);

count(b1.bname)
6

1 row in set (0.00 sec)

## **EXPERIMENT NO. 8**

### **AIM**

To have familiarize with trigger functions

### **Question**

Create a Trigger for employe 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 `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 `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>

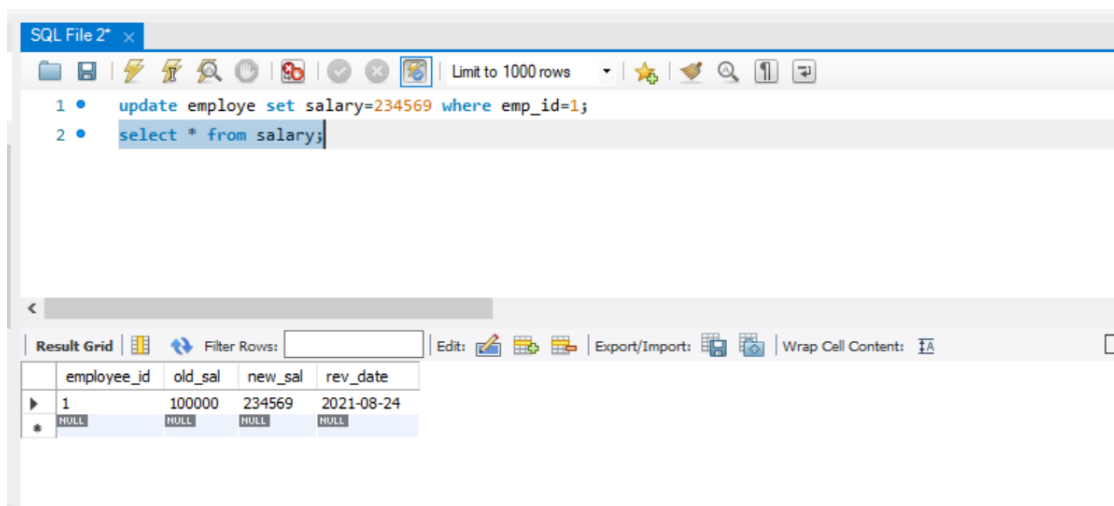
CREATE DEFINER=`root`@`localhost` TRIGGER
`employee_db`.`employee_AFTER_UPDATE` AFTER UPDATE ON `employee`
FOR EACH ROW
BEGIN
  if(new.salary != old.salary)
  then
    INSERT INTO salary (employee_id,old_sal,new_sal,rev_date) values
    (new.emp_id,old.salary,new.salary,sysdate());
  END if;
END

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 commands: `1 • update employee set salary=234569 where emp_id=1;` and `2 • select * from salary;`. Below the editor, the "Result Grid" is displayed, showing the output of the second command. The grid has four columns: `employee_id`, `old_sal`, `new_sal`, and `rev_date`. The first row shows the result for `emp_id=1`, with `old_sal` as 100000, `new_sal` as 234569, and `rev_date` as 2021-08-24. A second row with all NULL values is also visible.

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



## **AIM**

To have familiarize with trigger functions

## **Question**

Create a Trigger for employee table it will update another table  
personal\_updatations 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 `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 `personal_updatations` (
```

```

`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

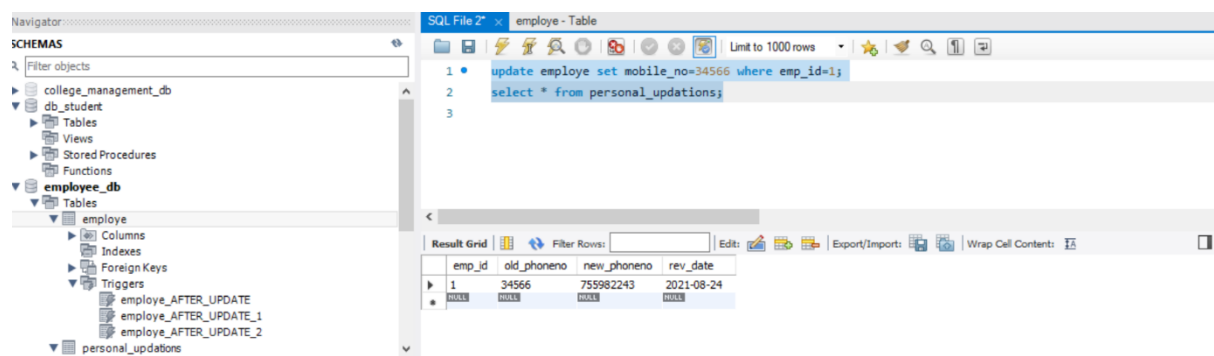
sql>

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 pane shows the SQL File Editor with the following commands:

```

1 • update employee set mobile_no=34566 where emp_id=1;
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 is a header row with the word 'NULL' in each column.

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

## **AIM**

To have familiarize with trigger functions

## **Question**

Create a Trigger for employee 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 `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 `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

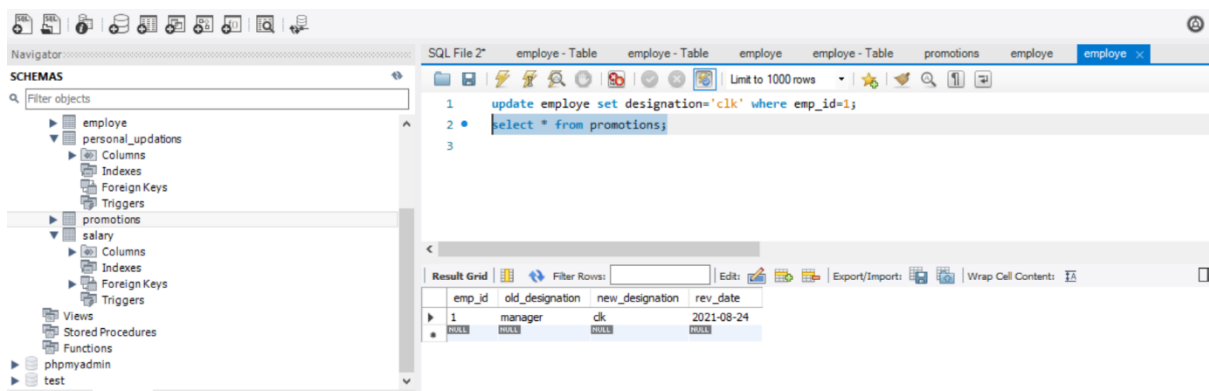
sql>

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

select * from promotions;

```

## OUTPUT



The screenshot shows a MySQL IDE interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure, including tables like 'employee', 'promotions', and 'salary'. The main window displays the SQL File 2\* editor with the following commands:

```

1 update employee set designation='clk' where emp_id=1;
2 select * from promotions;
3

```

Below the editor, the 'Result Grid' shows the output of the second command, displaying the contents of the 'promotions' table:

emp_id	old_designation	new_designation	rev_date
1	manager	clk	2021-08-24

## **EXPERIMENT NO. 9**

### **How to Download & Install MongoDB on Windows**

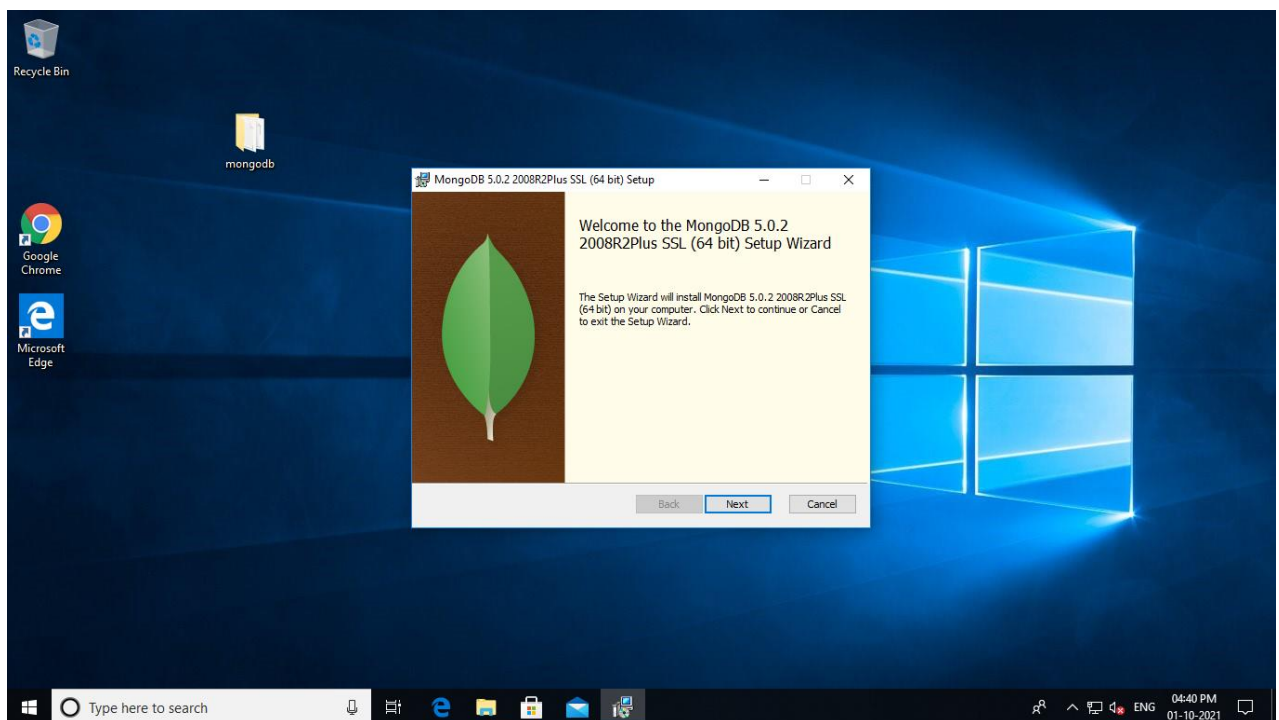
The installers for MongoDB are available in both the 32-bit and 64-bit format. The 32-bit installers are good for development and test environments. But for production environments you should use the 64-bit installers. Otherwise, you can be limited to the amount of data that can be stored within MongoDB.

### **Download & Install MongoDB on Windows**

The following steps can be used to install MongoDB on Windows 10:

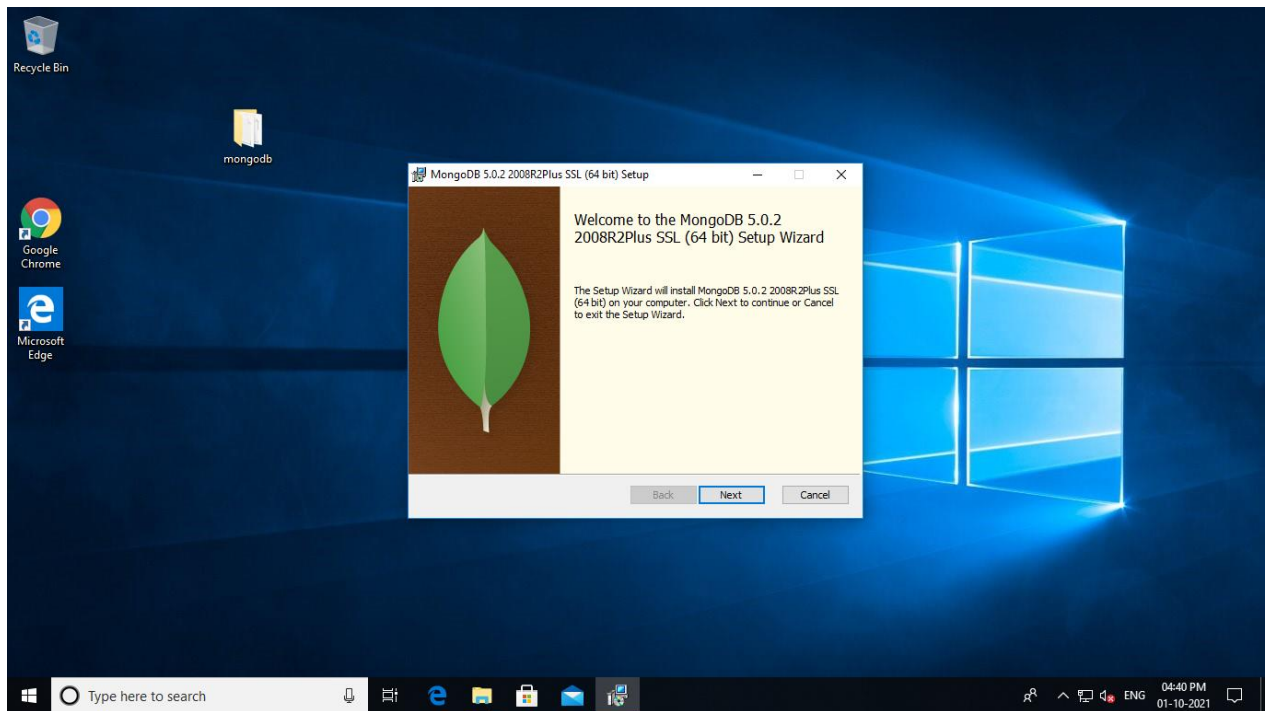
#### **Step 1:**

After downloading MongoDB, open the msi file and click next.



#### **Step 2:**

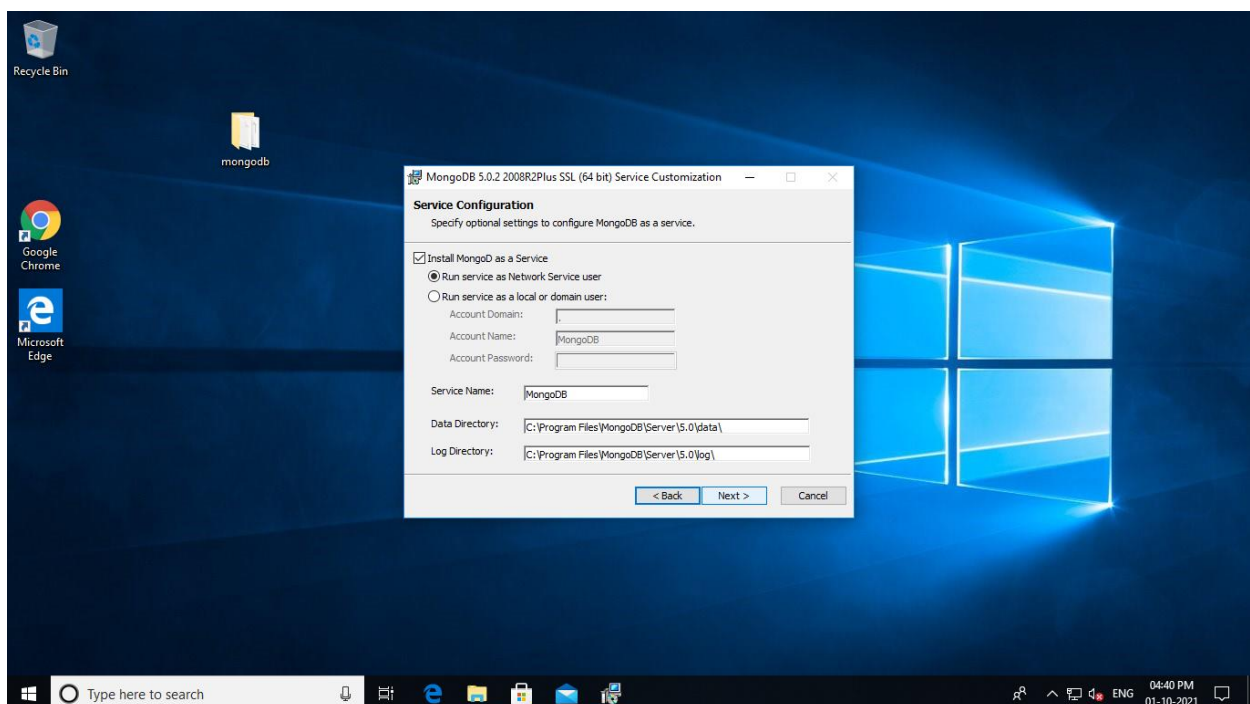
- a. Accept the End-User License Agreement
- b. Click Next



### Step 3:

Click on the "complete" button to install all of the components. The custom option can be used to install selective components or if you want to change the location of the installation.

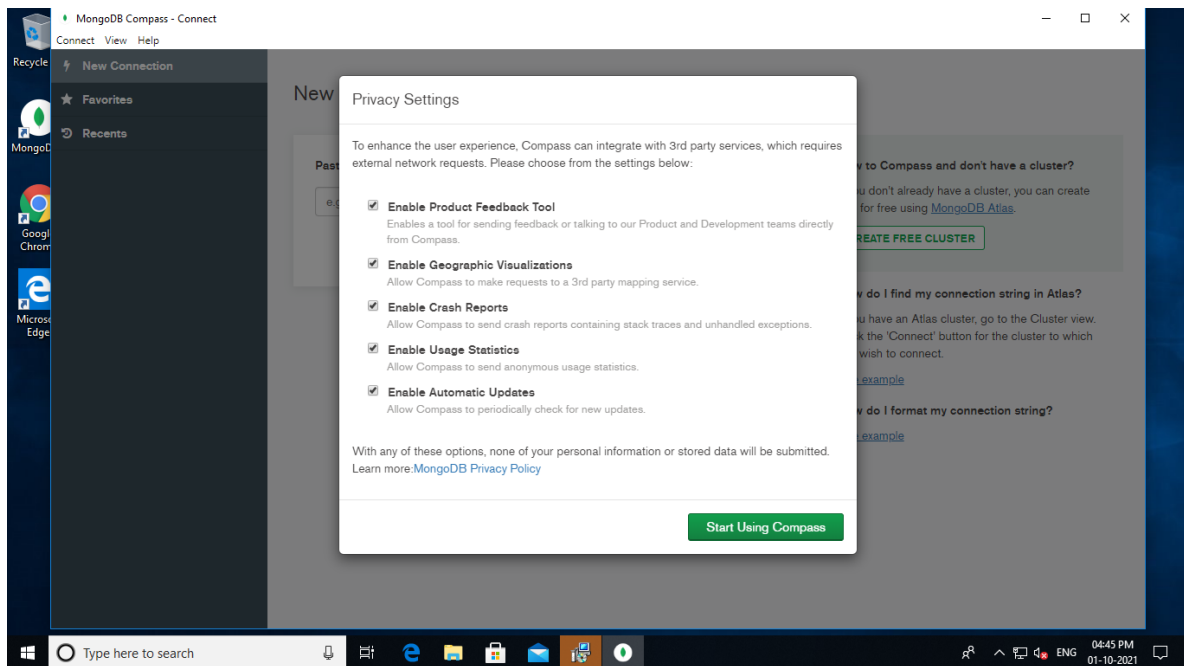
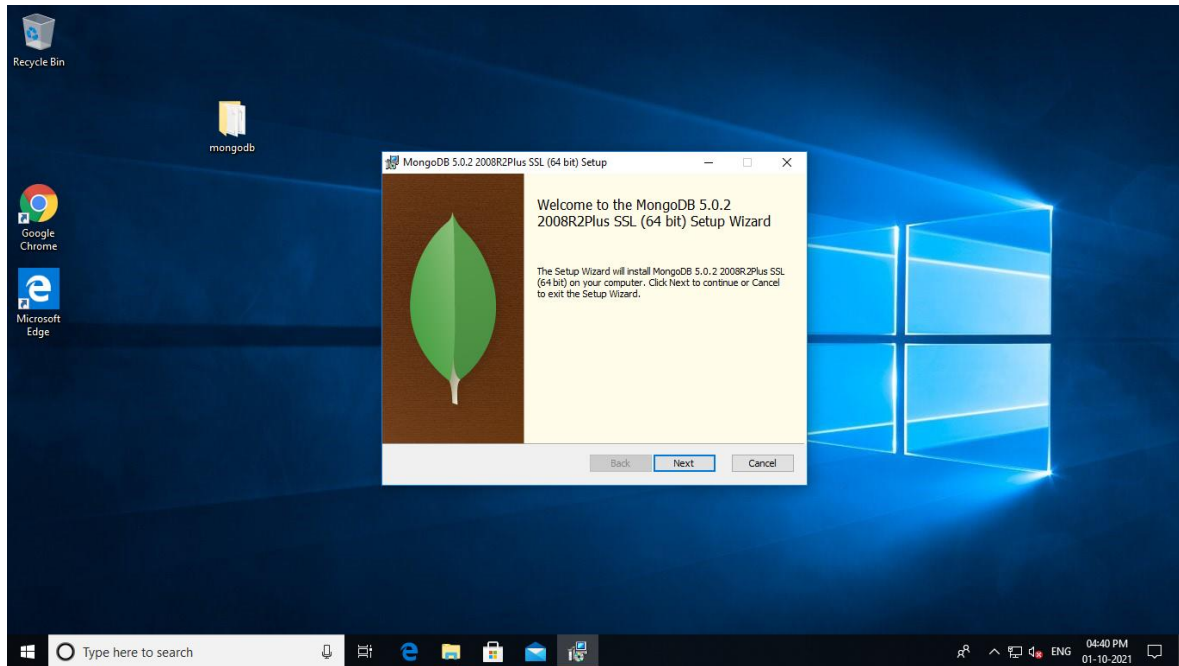
1. Select "Run service as Network Service user". Make a note of the data directory, we'll need this later.
2. Click Next

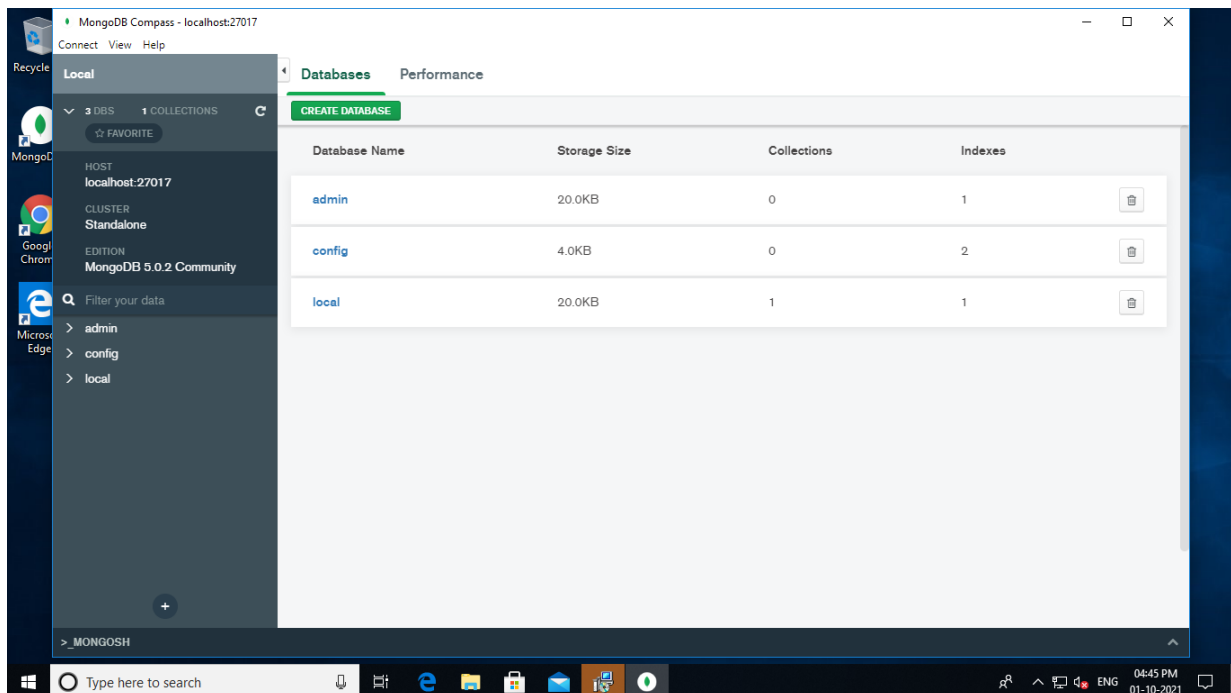
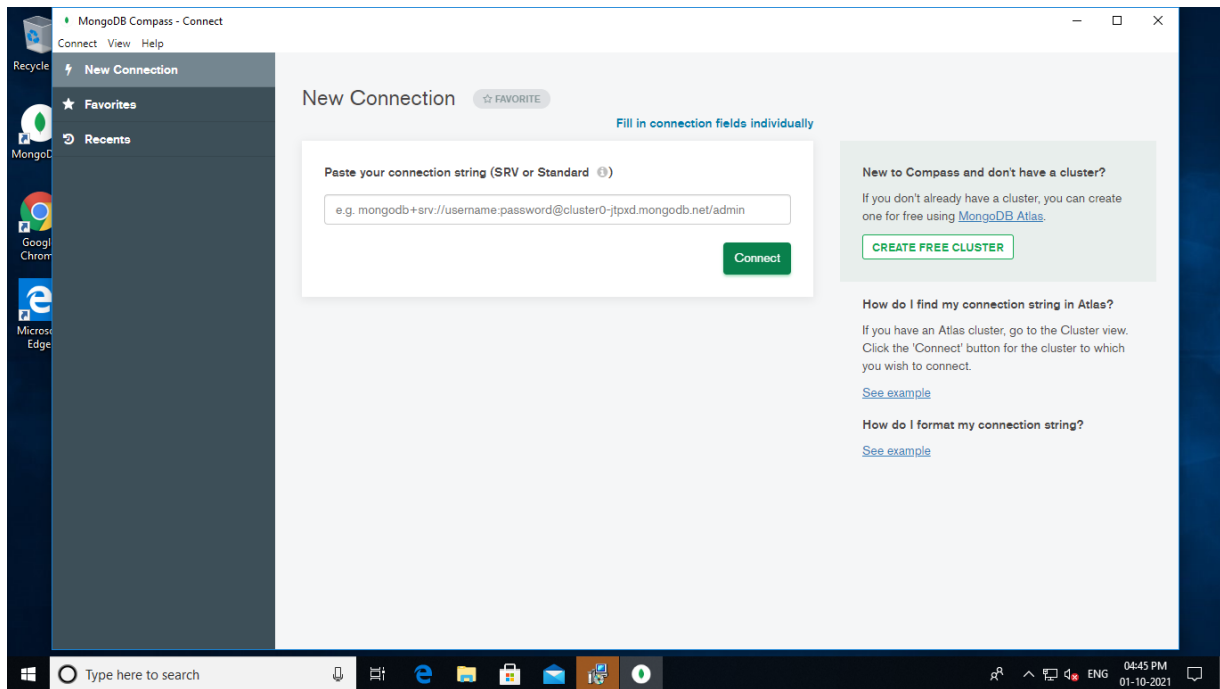


## Step 4:

Click on the Install button to start the installation.

Installation begins. Click next once completed. Click on the Finish button to complete the installation.










## EXPERIMENT NO. 10

### Install PHP MongoDB Driver

Download correct mongodb driver for PHP, based on your PHP version, Architecture and whether Thread Safety is enabled. You can get that information from the [phpinfo](#) page.

Do the following steps to install and configure MongoDB driver on Windows XAMPP Server.

1. Download the latest stable version of the PHP MongoDB driver from following URL <https://pecl.php.net/package/mongodb>.

Available Releases			
Version	State	Release Date	Downloads
<a href="#">1.3.2</a>	stable	2017-10-30	<a href="#">mongodb-1.3.2.tgz</a> (904.0kB) 
<a href="#">1.3.1</a>	stable	2017-10-16	<a href="#">mongodb-1.3.1.tgz</a> (904.0kB) 
<a href="#">1.3.0</a>	stable	2017-09-19	<a href="#">mongodb-1.3.0.tgz</a> (906.1kB) 

2. Extract the archive File.
3. Copy the `php_mongodb.dll` file from the extracted folder to the PHP extension directory. this is usually the "C:\xampp\php\ext" folder in XAMPP Server.
4. Open the `php.ini` file inside your PHP installation (C:\xampp\php) and add the following line:  
**`extension=php_mongodb.dll`**
5. Save the file and close it. Restart the Apache web server.

### Test MongoDB Connection From PHP Script

Let's write a very simple PHP program that creates a connection to the MongoDB server and dump the connection status.

Add following PHP code to your php script and access from the Web browser.

The above PHP example will output something similar to:

```
<?php
```

```
$connection = new MongoDB\Driver\Manager("mongodb://localhost:27017");
```

```
var_dump($connection);
```

```
?>
```

Result

```
object(MongoDB\Driver\Manager)#1 (2) { ["uri"]=> string(25)
"mongodb://localhost:27017" ["cluster"]=> array(1) { [0]=> array(11) {
["host"]=> string(9) "localhost" ["port"]=> int(27017) ["type"]=> int(0)
["is_primary"]=> bool(false) ["is_secondary"]=> bool(false) ["is_arbiter"]=>
bool(false) ["is_hidden"]=> bool(false) ["is_passive"]=> bool(false) ["tags"]=>
array(0) { } ["last_is_master"]=> array(0) { } ["round_trip_time"]=> int(-1) } } }
```

## **EXPERIMENT NO. 11**

### **Program to connect to mongodb and insert data**

#### **Connection**

```
<?php
```

```
require 'vendor/autoload.php';

$m = new MongoClient("mongodb://localhost:27017");

// connect to mongodb

echo "Connection to database successfully";

// select a database

$db = $m->test12;

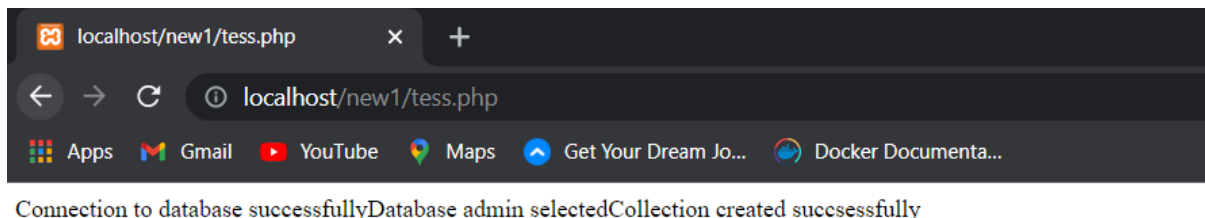
echo "Database admin selected";

$collection = $db->createCollection("sample");

echo "Collection created successfully";

?>
```

#### **Result**



#### **Insertion**

```
<?php
```

```
// connect to mongodb

require 'vendor/autoload.php';

$m = new MongoClient("mongodb://localhost:27017");

echo "Connection to database successfully";
```

```
echo " ";
```

```
$db = $m->test12;
```

```
echo "Database mydb selected";
```

```
$collection = $db->newtest;
```

```
echo "Collection selected successfully";
```

```
echo " ";
```

```
$document1 = array(
```

```
    "title" => "class",
```

```
    "description" => "students",
```

```
    "likes" => 1000,
```

```
    "url" => "http://www.apple.com/mongodb/",
```

```
    "by" => "apple"
```

```
);
```

```
$document2 = array(
```

```
    "title" => "cs",
```

```
    "description" => "stu",
```

```
    "likes" => 9000,
```

```
    "url" => "http://www.apple.com/mongodb/",
```

```
    "by" => "apple"
```

```
);
```

```
$collection->insertMany([$document1,$document2]);
```

```
echo "Document inserted successfully";
```

```
?>
```

## Result

### In Browser



### In MongoCompass



## **EXPERIMENT NO. 12**

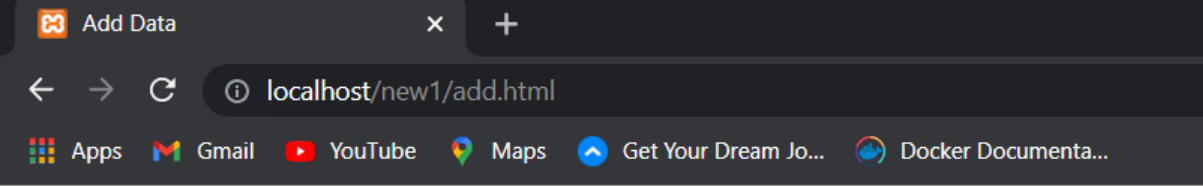
Perform CRUD operations in MongoDB in a registration form

### **Registration form**

```
<html>
<head>
  <title>Add Data</title>
</head>
<body>
  <a href="index2.php">Home</a>
  <br/><br/>
  <form action="add.php" method="post" name="form1">
    <table width="25%" border="0">
      <tr>
        <td>Name</td>
        <td><input type="text" name="name"></td>
      </tr>
      <tr>
        <td>Age</td>
        <td><input type="text" name="age"></td>
      </tr>
      <tr>
        <td>Email</td>
        <td><input type="text" name="email"></td>
      </tr>
      <tr>
        <td></td>
        <td><input type="submit" name="Submit" value="Add"></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

## Result

### In Browser



The screenshot shows a web browser window with a single tab titled 'Add Data'. The address bar displays 'localhost/new1/add.html'. Below the address bar, there are several bookmark icons: 'Apps', 'Gmail', 'YouTube', 'Maps', 'Get Your Dream Jo...', and 'Docker Documenta...'. The main content area of the browser shows a form with the following elements:

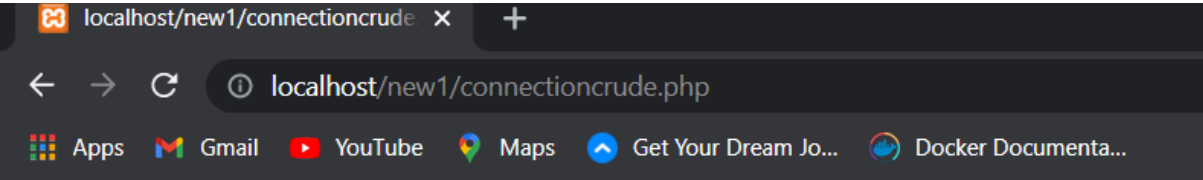
- A link labeled 'Home' in purple text.
- Three input fields labeled 'Name', 'Age', and 'Email' stacked vertically.
- An 'Add' button located below the 'Email' field.

## C-Create Connection

```
//connectioncrude.php
<?php
require 'vendor/autoload.php';
$connection = (new MongoDB\Client);
echo "$connection";
$db=$connection->user2;
echo "$db";
$collection=$db->details;
?>
```

## Result

### In Browser



The screenshot shows a web browser window with a single tab titled 'localhost/new1/connectioncrude'. The address bar displays 'localhost/new1/connectioncrude.php'. Below the address bar, there are several bookmark icons: 'Apps', 'Gmail', 'YouTube', 'Maps', 'Get Your Dream Jo...', and 'Docker Documenta...'. The main content area of the browser displays the output of the PHP script:

```
mongodb://127.0.0.1/user2
```

## In MongoDB Compass



### Insert Values

```
//add.php
```

```
<html>
```

```
<head>
```

```
    <title>Add Data</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
include_once("connectioncrude.php");
```

```
if(isset($_POST['Submit'])) {
```

```
    $user = array (
```

```
        'name' => $_POST['name'],
```

```
        'age' => $_POST['age'],
```



```

        'email' => $_POST['email']

    );

    $db->users->insertOne($user);

    echo "<font color='green'>Data added successfully.";
    echo "<br/><a href='index2.php'>View Result</a>";

}

?>

</body>

</html>

```

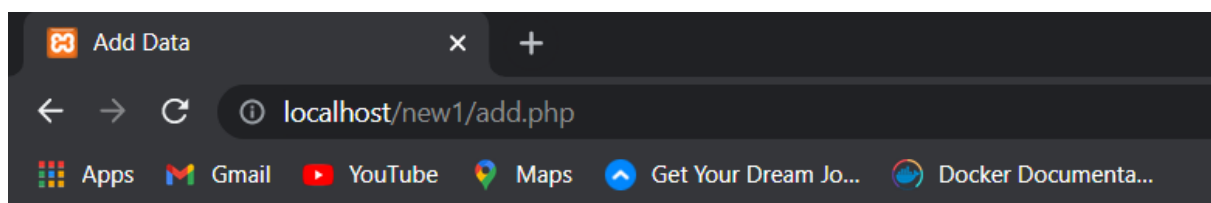
## Result

### In Browser



[Home](#)

Name	<input type="text" value="Ammu"/>
Age	<input type="text" value="15"/>
Email	<input type="text" value="ammu@gmail.com"/>
	<input type="button" value="Add"/>



[mongodb://127.0.0.1/user2Data added successfully.](#)  
[View Result](#)

## In Mongodb Compass

The screenshot shows the MongoDB Compass interface for the 'user2.users' collection. The top bar indicates 1 document and 1 index. The main area displays a table of 5 documents with columns: \_id, name, age, and email. Each document has a corresponding action menu (edit, delete, etc.) on the right.

	_id ObjectId	name String	age String	email String
1	615b17e1e8220002b00004b	"Ammu"	"15"	"ammu@gmail.com"
2	615b199be8220002b000050	"Ajin"	"19"	"ajin@gmail.com"
3	615b19b5e8220002b000051	"Meenu"	"78"	"meenu@gmail.com"
4	615b19dae8220002b000052	"Athira"	"18"	"athira@gmail.com"
5	615b19fce8220002b000053	"Sini"	"20"	"sini@gmail.com"

## R-Read

```
//index2.php
```

```
<?php
```

```
include_once("connectioncrude.php");
```

```
$result = $db->users->find();
```

```
?>
```

```
<html>
```

```
<head>
```

```
    <title>Homepage</title>
```

```
</head>
```

```
<body>
```

```
<a href="add.html">Add New Data</a><br/><br/>
```

```
    <table width='80%' border=0>
```

```
        <tr bgcolor='#CCCCCC'>
```

```
            <td>Name</td>
```

```

        <td>Age</td>

        <td>Email</td>

        <td>Update</td>

    </tr>

<?php
foreach ($result as $res) {
    echo "<tr>";
    echo "<td>".$res['name'].</td>";
    echo "<td>".$res['age'].</td>";
    echo "<td>".$res['email'].</td>";

    echo "<td><a href=\"edit.php?id=$res[_id]\">Edit</a> | <a
href=\"delete.php?id=$res[_id]\" onClick=\"return confirm('Are you sure you
want to delete?')\">Delete</a></td>";

    }
?>

</table>

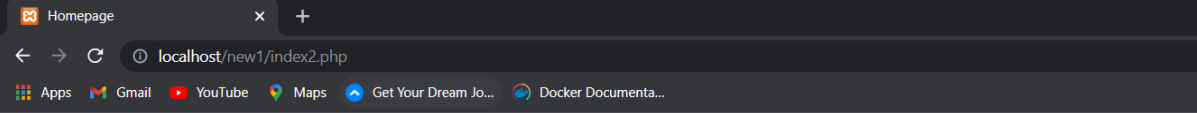
</body>

</html>

```

## Result

### In Browser



mongodb://127.0.0.1/user2 [Add New Data](#)

Name	Age	Email	Update
Ammu	15	ammu@gmail.com	<a href="#">Edit</a>   <a href="#">Delete</a>
Ajin	19	ajin@gmail.com	<a href="#">Edit</a>   <a href="#">Delete</a>
Meenu	78	meenu@gmail.com	<a href="#">Edit</a>   <a href="#">Delete</a>
Athira	18	athira@gmail.com	<a href="#">Edit</a>   <a href="#">Delete</a>
Sini	20	sini@gmail.com	<a href="#">Edit</a>   <a href="#">Delete</a>

## U-Update

```
//edit.php
```

```
<?php
```

```
include_once("connectioncrude.php");
```

```
if(isset($_POST['update']))
```

```
{
```

```
    $id = $_POST['id'];
```

```
    $name1 = $_POST['name'];
```

```
    $age1 = $_POST['age'];
```

```
    $email1 = $_POST['email'];
```

```
        $db->users->updateOne(
```

```
            ['_id' => new MongoDB\BSON\ObjectID($id)],
```

```
            ['$set' => ['name' => $name1, 'age' => $age1, 'email' =>
```

```
$email1,]]);
```

```
        header("Location: index2.php");
```

```
}
```

```
?>
```

```
<?php
```

```
$id = $_GET['id'];
```

```
$result = $db->users->findOne(['_id' => new  
MongoDB\BSON\ObjectID($id)]);
```

```
$name = $result['name'];
```

```
$age = $result['age'];
```

```
$email = $result['email'];
```

```
?>
```

```
<html>
```

```
<head>
```

```
    <title>Edit Data</title>
```

```
</head>
```

```
<body>
```

```
    <a href="index2.php">Home</a>
```

```
    <br/><br/>
```

```
    <form name="form1" method="post" action="edit.php">
```

```

<table border="0">
    <tr>
        <td>Name</td>
        <td><input type="text" name="name"
value="<?php echo $name;?>"></td>
    </tr>
    <tr>
        <td>Age</td>
        <td><input type="text" name="age"
value="<?php echo $age;?>"></td>
    </tr>
    <tr>
        <td>Email</td>
        <td><input type="text" name="email"
value="<?php echo $email;?>"></td>
    </tr>
    <tr>
        <td><input type="hidden" name="id"
value=<?php echo $_GET['id'];?>></td>
        <td><input type="submit" name="update"
value="Update"></td>
    </tr>
</table>
</form>
</body>
</html>

```

## Result

### In Browser



Name	Age	Email	Update
Ammu	15	ammu@gmail.com	<a href="#">Edit</a>   <a href="#">Delete</a>
Ajin	19	ajin@gmail.com	<a href="#">Edit</a>   <a href="#">Delete</a>
Meenu	78	meenu@gmail.com	<a href="#">Edit</a>   <a href="#">Delete</a>
Athira	18	athira@gmail.com	<a href="#">Edit</a>   <a href="#">Delete</a>
Sini	20	sini@gmail.com	<a href="#">Edit</a>   <a href="#">Delete</a>

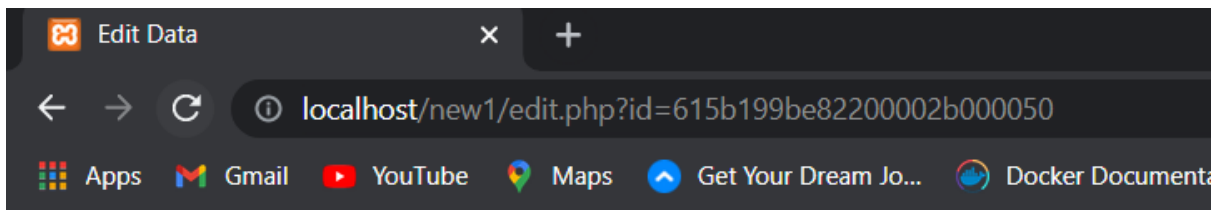


mongodb://127.0.0.1/user2 [Home](#)

Name

Age

Email



mongodb://127.0.0.1/user2 [Home](#)

Name

Age

Email



mongodb://127.0.0.1/user2 [Add New Data](#)

Name	Age	Email	Update
Ammu	15	ammu@gmail.com	<a href="#">Edit</a> / <a href="#">Delete</a>
Manu	30	manu@gmail.com	<a href="#">Edit</a> / <a href="#">Delete</a>
Meenu	78	meenu@gmail.com	<a href="#">Edit</a> / <a href="#">Delete</a>
Athira	18	athira@gmail.com	<a href="#">Edit</a> / <a href="#">Delete</a>
Sini	20	sini@gmail.com	<a href="#">Edit</a> / <a href="#">Delete</a>

## In MongoDB Compass

The screenshot shows the MongoDB Compass interface for the `user2.users` collection. The top bar indicates 1 document and 1 index. The main table displays 5 documents with fields: `_id` (ObjectID), `name` (String), `age` (String), and `email` (String). The documents are as follows:

	_id ObjectID	name String	age String	email String
1	615b17e1e8220002b00004b	"Ammu"	"15"	"ammu@gmail.com"
2	615b199be8220002b000050	"Manu"	"30"	"manu@gmail.com"
3	615b19b5e8220002b000051	"Meenu"	"78"	"meenu@gmail.com"
4	615b19dae8220002b000052	"Athira"	"18"	"athira@gmail.com"
5	615b19fce8220002b000053	"Sini"	"20"	"sini@gmail.com"

## D-Delete

//delete.php

<?php

```
include("connectioncrude.php");
```

```
$id = $_GET['id'];
```

```
echo $id;
```

```
$db->users->deleteOne(['_id' => new MongoDB\BSON\ObjectID($id)]);
```

```
header("Location:index2.php");
```

```
?>
```

## Result

### In Browser

The screenshot shows a web browser displaying a table of user data. The table has columns: Name, Age, and Email. The data is as follows:

| Name   | Age | Email            |
|--------|-----|------------------|
| Ammu   | 15  | ammu@gmail.com   |
| Manu   | 30  | manu@gmail.com   |
| Meenu  | 78  | meenu@gmail.com  |
| Athira | 18  | athira@gmail.com |
| Sini   | 20  | sini@gmail.com   |

A confirmation dialog box is overlaid on the table, asking "Are you sure you want to delete?". The dialog has "OK" and "Cancel" buttons. Below the table, there are links for "Edit" and "Delete" for each row.

Homepage x +

localhost/new1/index2.php

Apps Gmail YouTube Maps Get Your Dream Jo... Docker Documenta...

mongodb://127.0.0.1/user2 [Add New Data](#)

| Name  | Age | Email           | Update  |
|-------|-----|-----------------|---|
| Ammu  | 15  | ammu@gmail.com  | <a href="#">Edit</a>   <a href="#">Delete</a> |
| Manu  | 30  | manu@gmail.com  | <a href="#">Edit</a>   <a href="#">Delete</a> |
| Meenu | 78  | meenu@gmail.com | <a href="#">Edit</a>   <a href="#">Delete</a> |
| Sini  | 20  | sini@gmail.com  | <a href="#">Edit</a>   <a href="#">Delete</a> |

## In Mongodb Compass

user2.users Documents

user2.users DOCUMENTS 1 TOTAL SIZE 75B AVG. SIZE 75B INDEXES 1 TOTAL SIZE 24.0KB AVG. SIZE 24.0KB

Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER** { field: 'value' } **OPTIONS** **FIND** **RESET** **REFRESH**

**ADD DATA** **VIEW** **REFRESH**

Displaying documents 1 - 4 of 4

|   | _id ObjectId             | name String | age String | email String      |
|---|--------------------------|-------------|------------|-------------------|
| 1 | 615b17e1e82200002b00004b | "Ammu"      | "15"       | "ammu@gmail.com"  |
| 2 | 615b199be82200002b000050 | "Manu"      | "30"       | "manu@gmail.com"  |
| 3 | 615b19b5e82200002b000051 | "Meenu"     | "78"       | "meenu@gmail.com" |
| 4 | 615b19fce82200002b000053 | "Sini"      | "20"       | "sini@gmail.com"  |