

**ADVANCED DBMS LAB****Experiment No.: 1****Aim**

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

**QUESTION**

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<br>1980 | 5000 |     |
| 102 | Ajay   | 05-Jul<br>1985 | 5000 | 500 |
| 103 | Ravi   | 12-Aug<br>1981 | 1500 |     |
| 104 | Nikesh | 03-Mar<br>1983 | 3000 | 700 |
| 105 | Ravi   | 05-jul<br>1985 | 3000 |     |

2.Modifying the structure of tables

- Add new columns: sal number(7,2)
- Dropping a column from a table: sal
- Modifying existing column :ename varchar2(15)
- Renaming the tables: emp to emp1
- truncating the tables:emp1
- 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))

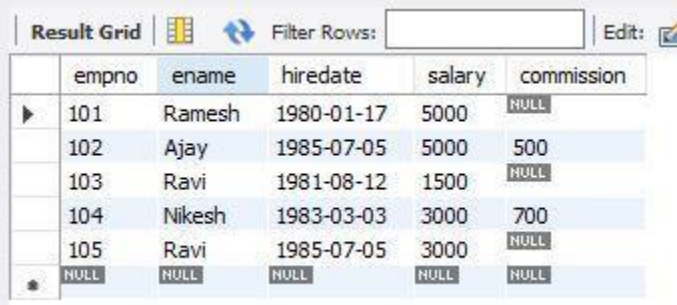
5.Create a table cust with(custid number(6) constraint unique, name char(10) 6. Refer the table “stud” in table “ student”

**PROCEDURE AND OUTPUT SCREENSHOT****1. EMP table (Create and Insert)**

```
CREATE TABLE emp(
  empno int(5),
  ename varchar(10),
  hiredate date,
  salary bigint,
  commission bigint,
  PRIMARY KEY (empno)
);
```

```
INSERT INTO emp VALUES
(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);
```

```
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       |
| ✱ | NULL  | NULL   | NULL       | NULL   | NULL       |

**2. Modifying the structure of tables****a. Add new columns: sal number(7,2)**

```
>alter table emp add sal numeric(7,2);
```



|   | 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**

>alter table emp drop sal;

|   | 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       |

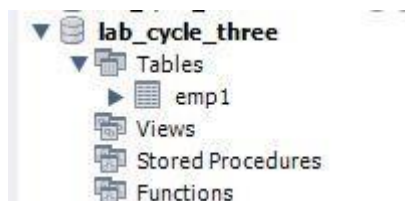
**c. Modifying existing column :ename varchar2(15)**

>alter table emp modify column ename varchar(15);

| Column     | Type        | Default Value | Nullable | Character Set | Collation        | Privileges                         | Extra | Comments |
|------------|-------------|---------------|----------|---------------|------------------|------------------------------------|-------|----------|
| empno      | int         |               | NO       |               |                  | select, insert, update, references |       |          |
| ename      | varchar(15) |               | YES      | utf8mb4       | utf8mb4_0900_... | select, insert, update, references |       |          |
| hiredate   | date        |               | YES      |               |                  | select, insert, update, references |       |          |
| salary     | bigint      |               | YES      |               |                  | select, insert, update, references |       |          |
| commission | bigint      |               | YES      |               |                  | select, insert, update, references |       |          |

**d. Renaming the tables: emp to emp1**

>alter table emp rename emp1;

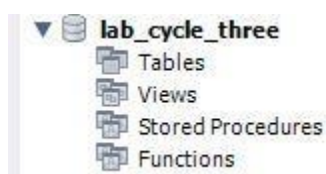
**e. truncating the tables: emp1**

>truncate table emp1;

|   | empno | ename | hiredate | salary | commission |
|---|-------|-------|----------|--------|------------|
| ✱ | NULL  | NULL  | NULL     | NULL   | NULL       |

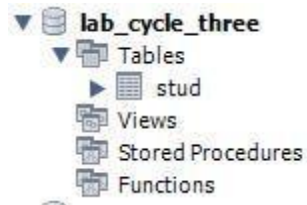
**f. Destroying tables: emp**

>drop table emp1;



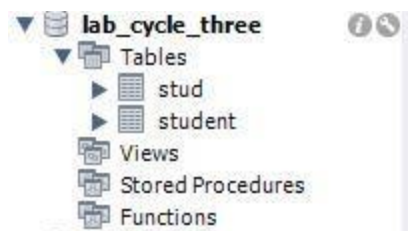
### 3 .STUD table

```
CREATE TABLE stud (  
    sname varchar(20),  
    rollno numeric(10) not null,  
    dob date not null,  
    PRIMARY KEY (sname)  
);
```



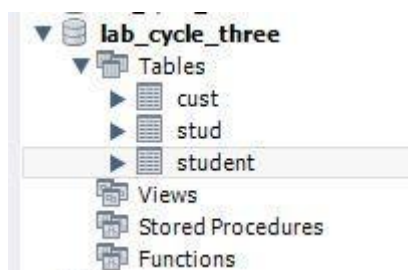
### 4.STUDENT table

```
CREATE TABLE student (  
    regno numeric(6),  
    mark numeric(3),  
    check(mark >= 0 and mark <= 100)  
);
```



### 5. CUST Table

```
CREATE TABLE cust (  
    custid numeric(6) unique,  
    name varchar(10)  
);
```



**ADVANCED DBMS LAB****Experiment No.: 2****Aim**

To study various DML commands – select, insert, delete, update

**Name: SHAMJAD MAZOOD NAZER**

**Roll No: 36**

**Batch: B**

**Date: 25-03-2022**

**QUESTION**

Create the following Tables and Insert values.

**Table 1: DEPOSIT**

ACTNO VARCHAR (5) PRIMARY KEY, FIRST LETTER MUST START WITH 'D'  
CNAME VARCHAR (20) FOREIGN KEY REFERENCES CUSTOMER  
BNAME VARCHAR (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 VARCHAR (15) PRIMARY KEY  
CITY VARCHAR (20) NOT NULL

**Table 4: BORROW**

LOANNO VARCHAR (28) PRIMARY KEY / FIRST LETTER MUST START WITH 'L'  
CNAME VARCHAR (15) FOREIGN KEY REFERENCES CUSTOMER  
BNAME VARCHAR2(20) FOREIGN KEY REFERENCES BRANCH  
AMOUNT NUMBER(8,2) NOT NULL, CANNOT BE 0

**PROCEDURE & OUTPUT****1. Customer table (Create and Insert)**

```
CREATE TABLE CUSTOMER(  
CNAME varchar(15) primary key,  
CITY varchar(20) NOT NULL  
);  
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 CUSTOMER values('VRCE','NAGPUR');  
INSERT INTO CUSTOMER values('AJNI','NAGPUR');
```

**2. Branch (Create and Insert)**

```
CREATE TABLE BRANCH(BNAME VARCHAR(20) PRIMARY KEY,CITY  
VARCHAR(30)CHECK(CITY IN('NAGPUR','DELHI','BANGALORE','BOMBAY'))NOT  
NULL);  
  
INSERT INTO BRANCH VALUES('KAROLBAGH','DELHI');  
INSERT INTO BRANCH VALUES('CHANDINI','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');
```

**3. Borrow (Create and Insert)**

```
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);
```

```
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);
```




**4. Deposit (Create and Insert)**

```
CREATE TABLE DEPOSITE(ACTNO VARCHAR(20)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 DEPOSITE VALUES('D1456','PRAVAV','DELHI',32000,'1978-06-24');
INSERT INTO DEPOSITE VALUES('D1478','HARSHAL','BANGALORE',905000,'1996-05-
24');
INSERT INTO DEPOSITE VALUES('D1492','THARUN','BOMBAY',123000,'1999-03-12');
INSERT INTO DEPOSITE VALUES('D1123','DEVIKA','NAGPUR',82000,'2000-01-09');
INSERT INTO DEPOSITE VALUES('D1543','KIRAN','DELHI',89000,'1980-12-02');
INSERT INTO DEPOSITE VALUES('D1864','MANASI','BANGALORE',23400,'1995-09-12');
```

**1. List all data from table deposit.**

```
SELECT * FROM deposit;
```

| Result Grid  |       |         |         |              |            |
|--|-------|---------|---------|--------------|------------|
| Filter Rows: <input type="text"/>  |       |         |         |              |            |
| Edit:    |       |         |         |              |            |
|  | ACTNO | AMOUNT  | CNAME   | BNAME        | ADATE      |
| ▶  | D100  | 1000.00 | ANIL    | VRCE         | 1995-03-01 |
|  | D101  | 500.00  | SUNIL   | AJNI         | 1996-01-04 |
|  | D102  | 3500.00 | MEHUL   | KAROLBAGH    | 1995-11-17 |
|  | D104  | 1200.00 | MADHURI | DHARAMPETH   | 1995-12-17 |
|  | D105  | 3000.00 | PRAMOD  | MG ROAD      | 1996-03-27 |
|  | D106  | 2000.00 | SANDIP  | ANDHERI      | 1996-03-31 |
|  | D107  | 1000.00 | SHIVANI | ANDHERI      | 1995-09-05 |
|  | D108  | 5000.00 | KRANTI  | NEHRU PALACE | 1995-07-02 |
|  | D109  | 7000.00 | NAREN   | POWAI        | 1995-08-10 |
| *  | NULL  | NULL    | NULL    | NULL         | NULL       |

**2. List all data from borrow.**

```
SELECT * FROM borrow;
```

| Result Grid |        |              |              |         |
|-------------|--------|--------------|--------------|---------|
|             |        | Filter Rows: |              |         |
|             | LOANNO | CNAME        | BNAME        | AMOUNT  |
| ▶           | L201   | ANIL         | VRCE         | 1000.00 |
|             | L206   | MEHUL        | AJNI         | 5000.00 |
|             | L311   | SUNIL        | AJNI         | 3000.00 |
|             | L321   | MADHURI      | ANDHERI      | 2000.00 |
|             | L371   | PRAMOD       | MG ROAD      | 8000.00 |
|             | L481   | KRANTI       | NEHRU PALACE | 3000.00 |
| *           | NULL   | NULL         | NULL         | NULL    |

BORROW 16 ×

**3. List all data from customer.**

```
SELECT * FROM customer;
```

| Result Grid |         |              |
|-------------|---------|--------------|
|             |         | Filter Rows: |
|             | CNAME   | CITY         |
| ▶           | ANIL    | CALCUTTA     |
|             | KRANTI  | BOMBAY       |
|             | MADHURI | NAGPUR       |
|             | MANDAR  | PATNA        |
|             | MEHUL   | BARODA       |
|             | NAREN   | BOMBAY       |
|             | PRAMOD  | NAGPUR       |
|             | SANDIP  | SURAT        |
|             | SHIVANI | BOMBAY       |
|             | SUNIL   | DELHI        |
| *           | NULL    | NULL         |

CUSTOMER 13 ×

**4. List all data from branch.**

```
SELECT * FROM branch;
```

| Result Grid |              |              |
|-------------|--------------|--------------|
|             |              | Filter Rows: |
|             | BNAME        | city         |
| ▶           | AJNI         | NAGPUR       |
|             | ANDHERI      | BOMBAY       |
|             | CHANDINI     | DELHI        |
|             | DHARAMPETH   | NAGPUR       |
|             | KAROLBAGH    | DELHI        |
|             | MG ROAD      | BANGALORE    |
|             | NEHRU PALACE | DELHI        |
|             | POWAI        | BOMBAY       |
|             | VRCE         | NAGPUR       |
| *           | NULL         | NULL         |



**5. Give account no and amount of deposit.**

SELECT ACTNO, AMOUNT FROM deposit;

|   | ACTNO | AMOUNT  |
|---|-------|---------|
| ▶ | D100  | 1000.00 |
|   | D101  | 500.00  |
|   | D102  | 3500.00 |
|   | D104  | 1200.00 |
|   | D105  | 3000.00 |
|   | D106  | 2000.00 |
|   | D107  | 1000.00 |
|   | D108  | 5000.00 |
|   | D109  | 7000.00 |
| * | NULL  | NULL    |

**6. Give customer name and account no of depositors.**

SELECT CNAME, ACTNO FROM deposit;

|   | CNAME   | ACTNO |
|---|---------|-------|
| ▶ | ANIL    | D100  |
|   | KRANTI  | D108  |
|   | MADHURI | D104  |
|   | MEHUL   | D102  |
|   | NAREN   | D109  |
|   | PRAMOD  | D105  |
|   | SANDIP  | D106  |
|   | SHIVANI | D107  |
|   | SUNIL   | D101  |
| * | NULL    | NULL  |

**7. Give name of customers.**

SELECT CNAME FROM customer;

|   | CNAME   |
|---|---------|
| ▶ | ANIL    |
|   | KRANTI  |
|   | MADHURI |
|   | MANDAR  |
|   | MEHUL   |
|   | NAREN   |
|   | PRAMOD  |
|   | SANDIP  |
|   | SHIVANI |
|   | SUNIL   |
| * | NULL    |

**8. Give name of branches.**

```
SELECT BNAME FROM branch;
```

|   | BNAME        |
|---|--------------|
| ▶ | AJNI         |
|   | ANDHERI      |
|   | CHANDINI     |
|   | DHARAMPETH   |
|   | KAROLBAGH    |
|   | MG ROAD      |
|   | NEHRU PALACE |
|   | POWAI        |
|   | VRCE         |
| * | HULL         |

**9. Give name of borrows.**

```
SELECT CNAME FROM borrow;
```

|   | CNAME   |
|---|---------|
| ▶ | ANIL    |
|   | KRANTI  |
|   | MADHURI |
|   | MEHUL   |
|   | PRAMOD  |
|   | SUNIL   |

**10. Give names of customer living in city Nagpur.**

```
SELECT CNAME FROM CUSTOMER WHERE CITY = 'NAGPUR';
```

|   | CNAME   |
|---|---------|
| ▶ | MADHURI |
|   | PRAMOD  |
| * | HULL    |

**11. Give names of depositors having amount greater than 4000.**

```
SELECT CNAME FROM deposit WHERE amount > 4000;
```

|   | CNAME  |
|---|--------|
| ▶ | KRANTI |
|   | NAREN  |

**12. Give account date of Anil.**

```
SELECT ADATE FROM deposit WHERE CNAME = 'ANIL';
```

| CNAME | ADATE      |
|-------|------------|
| ANIL  | 1995-03-01 |

**13. Give name of all branches located in Bombay.**

```
SELECT BNAME FROM branch WHERE CITY = 'BOMBAY';
```

| BNAME   |
|---------|
| ANDHERI |
| POWAI   |
| HULL    |

**14. Give name of borrower having loan number L205.**

```
SELECT CNAME FROM borrow WHERE LOANNO='L205';
```

| CNAME |
|-------|
| MEHUL |

**15. Give names of depositors having account at VRCE.**

```
SELECT CNAME FROM deposit WHERE BNAME = 'VRCE';
```

| CNAME |
|-------|
| ANIL  |

**16. Give names of all branched located in city Delhi.**

```
SELECT BNAME FROM branch WHERE city = 'DELHI';
```

| BNAME        |
|--------------|
| CHANDINI     |
| KAROLBAGH    |
| NEHRU PALACE |
| HULL         |

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

```
SELECT CNAME FROM deposit WHERE ADATE = '1996-12-01';
```

| CNAME |
|-------|
| SUNIL |

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

SELECT ACTNO, AMOUNT FROM deposit WHERE ADATE BETWEEN '1996-12-01' AND '1996-05-01';

|   | ACTNO | AMOUNT |
|---|-------|--------|
| ▶ | D101  | 500.00 |
| ✱ | NULL  | NULL   |

**19. Give name of the city where branch KAROLBAGH is located.**

SELECT city FROM branch WHERE BNAME = 'KAROLBAGH';

|   | CITY  |
|---|-------|
| ▶ | DELHI |

**20. Give details of customer ANIL.**

SELECT \* FROM customer JOIN borrow ON customer.CNAME = borrow.CNAME JOIN deposit ON deposit.CNAME = borrow.CNAME WHERE customer.CNAME = 'ANIL';

|   | CNAME | CITY     | LOANNO | CNAME | BNAME | AMOUNT  | ACTNO | AMOUNT  | CNAME | BNAME | ADATE      |
|---|-------|----------|--------|-------|-------|---------|-------|---------|-------|-------|------------|
| ▶ | ANIL  | CALCUTTA | L201   | ANIL  | VRCE  | 1000.00 | D100  | 1000.00 | ANIL  | VRCE  | 1995-03-01 |

**ADVANCED DBMS LAB****Experiment No.: 3****Aim**

To familiarize with set operations.

**Commands**

1. List all the customers who are depositors but not borrowers.

```
SELECT c_name FROM deposit WHERE c_name NOT IN (SELECT c_name
FROM borrow);
```

| Result Grid |         |
|-------------|---------|
|             | c_name  |
| ▶           | SANDIP  |
|             | SHIVANI |
|             | MINU    |

2. List all the customers who are both depositors and borrowers

```
SELECT c_name FROM deposit UNION (SELECT c_name FROM borrow);
```

| Result Grid |         |
|-------------|---------|
|             | c_name  |
| ▶           | ANIL    |
|             | SUNIL   |
|             | MEHUL   |
|             | MADHURI |
|             | PRAMOD  |
|             | SANDIP  |
|             | SHIVANI |
|             | KRANTI  |
|             | MINU    |

3. List all the depositors having deposit in all the branches where Sunil is having Account

```
SELECT D1.c_name FROM deposit D1 WHERE D1.b_name IN (SELECT
D2.b_name FROM deposit D2 WHERE D2.c_name = 'SUNIL');
```

| Result Grid |        |
|-------------|--------|
|             | c_name |
| ▶           | SUNIL  |

**4. List all the customers living in city NAGPUR and having branch city BOMBAY or DELHI**

```
SELECT C1.c_name FROM customer C1,deposit D1, branch B1 WHERE C1.city = 'NAGPUR' AND C1.c_name = D1.c_name AND D1.b_name = B1.b_name AND B1.city IN ('BOMBAY','DELHI');
```

| Result Grid |         |
|-------------|---------|
|             | c_name  |
| ▶           | MADHURI |

**4. List all the depositors living in city NAGPUR**

```
SELECT DISTINCT(customer.c_name) from customer,deposit WHERE city='NAGPUR';
```

| Result Grid |         |
|-------------|---------|
|             | c_name  |
| ▶           | PRAMOD  |
|             | MADHURI |

**5. List all the depositors living in the city NAGPUR and having branch in city BOMBAY**

```
SELECT C1.c_name FROM customer C1,deposit D1, branch B1 WHERE C1.city = 'NAGPUR' AND C1.c_name = D1.c_name AND D1.b_name = B1.b_name AND B1.city IN ('BOMBAY');
```

| Result Grid |        |
|-------------|--------|
|             | c_name |

**6. List the branch cities of Anil and Sunil**

```
SELECT B1.city FROM deposit D1, branch B1 WHERE D1.b_name = B1.b_name AND D1.c_name IN ('SUNIL','ANIL');
```

| Result Grid |        |
|-------------|--------|
|             | city   |
| ▶           | NAGPUR |

**8. List the customers having deposit greater than 1000 and loan less than 10000.**

```
SELECT DISTINCT D1.c_name FROM deposit D1, borrow B1 WHERE
D1.amount>1000 AND B1.amount<10000;
```

| Result Grid |         |
|-------------|---------|
|             | c_name  |
| ▶           | MEHUL   |
|             | MADHURI |
|             | PRAMOD  |
|             | SANDIP  |
|             | KRANTI  |
|             | MINU    |

**9. List the cities of depositors having branch VRCE.**

```
SELECT B1.city FROM deposit D1, branch B1 WHERE D1.BNAME=B1.b_name
AND B1.b_name='VRCE';
```

| Result Grid |        |
|-------------|--------|
|             | city   |
| ▶           | NAGPUR |

**10. List the depositors having amount less than 1000 and living in the same city as Anil**

```
SELECT D1.c_name FROM deposit D1, customer C1 , customer C2 WHERE
C1.CITY = C2.CITY AND C2.c_name = 'ANIL' AND C1.c_name = D1.c_name
AND D1.amount < 1000;
```

| CNAME |
|-------|
| SUNIL |
| SUNIL |
| SUNIL |
| SUNIL |
| SUNIL |
| SUNIL |
| SUNIL |
| SUNIL |
| SUNIL |
| SUNIL |

### 11. List all the cities where branches of Anil and Sunil are locate

```
SELECT B1.city FROM branch B1 WHERE B1.b_name IN (SELECT D1.b_name
FROM deposit D1 WHERE D1.c_name IN ('ANIL','SUNIL'));
```

| city   |
|--------|
| NAGPUR |

### 12. List the amount for the depositors living in the city where Anil is living

```
SELECT DISTINCT(D1.c_name),D1.amount ,C1.city FROM deposit D1,
customer C1, branch B1 WHERE D1.c_name=C1.c_name AND C1.city
IN(SELECT C2.city FROM customer C2 WHERE C2.c_name='ANIL');
```

| c_name | amount | city     |
|--------|--------|----------|
| ANIL   | 1000   | CALCUTTA |



**ADVANCED DBMS LAB****Experiment No.: 4****AIM**

To familiarize with join or cartesian product.

**QUESTION**

1. Give name of customers having living city BOMBAY and branch city NAGPUR.

```
SELECT D1.c_name, D1.b_name, C1.c_name, C1.city, B1.city, B1.b_name FROM
DEPOSIT D1, CUSTOMER C1, BRANCH B1 WHERE C1.city = 'BOMBAY'
AND B1.city = 'NAGPUR' AND D1.c_name = C1.c_name AND D1.b_name =
B1.b_name;
```

Result Grid

Filter Rows:

Export:

|  |        |        |        |      |      |        |
|--|--------|--------|--------|------|------|--------|
|  | c_name | b_name | c_name | city | city | b_name |
|--|--------|--------|--------|------|------|--------|

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

```
SELECT distinct(customer.c_name), BRANCH.city FROM BRANCH, customer
WHERE BRANCH.city = customer.city;
```

|             |         |        |
|-------------|---------|--------|
| Result Grid |         | Filter |
|             | c_name  | city   |
| ▶           | KRANTI  | BOMBAY |
|             | MADHURI | NAGPUR |
|             | NAREN   | BOMBAY |
|             | PRAMOD  | NAGPUR |
|             | SHIVANI | BOMBAY |
|             | SUNIL   | DELHI  |



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

```
SELECT C1.c_name FROM CUSTOMER C1,DEPOSIT D1,BORROW B1
WHERE C1.city='NAGPUR' AND C1.c_name=D1.c_name AND D1.c_name =
B1.c_name;
```

| Result Grid |         |
|-------------|---------|
|             | c_name  |
| ▶           | MADHURI |
|             | PRAMOD  |

4. Give names of borrowers having deposit amount greater than 1000 and loan amount greater than 2000.

```
SELECT BR1.c_name, BR1.amount, D1.c_name, D1.amount FROM BORROW
BR1,DEPOSIT D1 WHERE D1.c_name = BR1.c_name AND D1.amount > 1000
AND BR1.amount > 2000;
```

|             |        |   |   |              |  |
|-------------|--------|---|---|--------------|--|
| Result Grid |        |  |  | Filter Rows: |  |
|             | c_name | amount  | c_name  | amount       |  |
| ▶           | MEHUL  | 5000  | MEHUL   | 3500         |  |
|             | PRAMOD | 8000  | PRAMOD  | 3000         |  |
|             | KRANTI | 3000  | KRANTI  | 5000         |  |



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

```
SELECT D1.c_name FROM DEPOSIT D1 WHERE D1.b_name IN (SELECT
D2.b_name FROM DEPOSIT D2 WHERE D2.c_name = 'SUNIL');
```

| Result Grid |        |
|-------------|--------|
|             | c_name |
| ▶           | SUNIL  |

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

```
SELECT BR1.c_name,BR1.amount FROM BORROW BR1 WHERE BR1.amount
> ALL (SELECT BR2.amount FROM BORROW BR2 WHERE BR2.c_name =
'PRAMOD');
```

|             |        |   |   |
|-------------|--------|---|---|
| Result Grid |        |  |  |
|             | c_name | amount  |   |

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

```
SELECT C.c_name FROM CUSTOMER C WHERE C.city IN (SELECT B.city
FROM BRANCH B WHERE B.b_name IN (SELECT D.b_name FROM DEPOSIT
D WHERE D.c_name='SUNIL'));
```

| Result Grid |        |
|-------------|--------|
|             | c_name |
|             | NULL   |

8. Give branch city and living city of Pramod.

```
SELECT B1.city , C1.city FROM BRANCH B1,CUSTOMER C1, DEPOSIT D1
WHERE C1.c_name = 'PRAMOD' AND C1.c_name = D1.c_name AND
D1.b_name = B1.b_name;
```

| Result Grid |           |        |
|-------------|-----------|--------|
|             | city      | city   |
|             | BANGALORE | NAGPUR |

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

```
SELECT B1.city FROM DEPOSIT D1, BRANCH B1 WHERE D1.b_name =
B1.b_name AND D1.c_name IN ('SUNIL','ANIL');
```

| Result Grid |        |
|-------------|--------|
|             | city   |
|             | NAGPUR |

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

```
SELECT C1.c_name, C1.city FROM CUSTOMER C1 WHERE C1.c_name =
'ANIL' OR C1.c_name = 'SUNIL';
```

| Result Grid |        |          |
|-------------|--------|----------|
|             | c_name | city     |
|             | ANIL   | CALCUTTA |
|             | SUNIL  | DELHI    |
|             | NULL   | NULL     |

**ADVANCED DBMS LAB****Experiment No.: 5****AIM**

To familiarize with Group by and Having clause.

Name: SHAMJAD MAZOOD NAZER

Roll No: 36

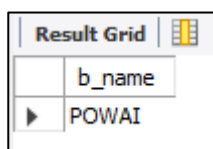
Batch: B

Date: 06-05-2022

**COMMANDS AND OUTPUT**

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

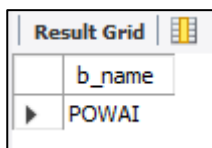
```
SELECT D.b_name FROM DEPOSIT D, BRANCH B WHERE  
D.b_name=B.b_name AND B.city='BOMBAY' GROUP BY D.b_name HAVING  
SUM(D.amount)>5000;
```



| b_name |
|--------|
| POWAI  |

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

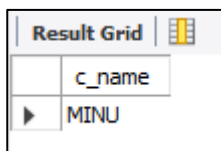
```
SELECT D.b_name FROM DEPOSIT D, BRANCH B WHERE  
D.b_name=B.b_name GROUP BY D.b_name HAVING SUM(D.amount)>5000;
```



| b_name |
|--------|
| POWAI  |

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

```
SELECT c_name from deposit where amount=(select AVG(amount) from  
DEPOSIT GROUP BY b_name having AVG(amount)>5000)
```



| c_name |
|--------|
| MINU   |

**4. List the names of customers having maximum deposit.**

```
SELECT MAX(amount),c_name FROM deposit;
```

| Result Grid |             |        | Filter |
|-------------|-------------|--------|--------|
|             | MAX(amount) | c_name |        |
| ▶           | 7000        | ANIL   |        |

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

```
SELECT D1.b_name FROM DEPOSIT D1 GROUP BY D1.b_name HAVING
COUNT(D1.c_name) >= ALL (SELECT COUNT(D2.c_name) FROM DEPOSIT
D2 GROUP BY D2.b_name);
```

| Result Grid |             | Filter |
|-------------|-------------|--------|
|             | b_name      |        |
| ▶           | VRCE        |        |
|             | ANJNI       |        |
|             | KAROLBAGH   |        |
|             | CHANDNI     |        |
|             | MG ROAD     |        |
|             | ANDHERI     |        |
|             | VIRAR       |        |
|             | NEHRU PLACE |        |
|             | POWAI       |        |

**6. Count the number of depositors living in NAGPUR.**

```
SELECT count(deposit.c_name)from deposit,CUSTOMER where
CUSTOMER.city='nagpur';
```

| Result Grid |                       | Filter |
|-------------|-----------------------|--------|
|             | count(deposit.c_name) |        |
| ▶           | 18                    |        |

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

```
SELECT c_name from deposit where b_name='VRCE' and amount=(select
max(amount) from deposit where b_name='VRCE');
```

| Result Grid |        |
|-------------|--------|
|             | c_name |
| ▶           | ANIL   |

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

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

| Result Grid |        |
|-------------|--------|
|             | b_name |

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

```
SELECT C.c_name ,count(B.b_name) from CUSTOMER C inner join Branch B on
C.c_name=B.b_name group by C.c_name order by count(B.b_name) DESC;
```

| Result Grid |                 |
|-------------|-----------------|
|             | c_name          |
|             | count(B.b_name) |

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

```
SELECT count(b1.b_name) From deposit d1 , borrow b1 , customer c1 Where
c1.c_name=d1.c_name and d1.c_name=b1.c_name and c1.city in (select city from
customer);
```

| Result Grid |                  |
|-------------|------------------|
|             | count(b1.b_name) |
| ▶           | 6                |

**ADVANCED DBMS LAB****Experiment No.: 6****Aim**

Implementation of triggers

**Name: SHAMJAD MAZOOD NAZER**

**Roll No: 36**

**Batch: B**

**Date: 10-05-2022**

**QUESTION**

1. Create a student table with fields id, name, subject1, subject2, subject3 and total, percentage. For each entry of row, update total marks and percentage using triggers in SQL.
2. Create a Trigger for student table that will update another table shows the name, total marks and percentage.

**Procedure**

```
CREATE TABLE STUDENT(SID INT PRIMARY KEY auto_increment NOT
NULL,STD_NAME varchar(20), PHYSICS INT,CHEMISTRY INT, TOTAL
INT);
```

```
create trigger total_t
before
insert
on student
for each row
set new.total=new.physics+new.chemistry ;
insert into student values(1,"stebin",3,4,0);
select * from student;
desc student;
```

```
CREATE TABLE MARKS(MARKID INT PRIMARY KEY
auto_increment,NAME VARCHAR(20),TOTAL_MARKS INT);
CREATE TRIGGER MARK_TRIGGER
AFTER
INSERT
ON STUDENT
FOR EACH ROW
INSERT INTO MARKS(NAME,TOTAL_MARKS)
VALUES(new.std_NAME,new.TOTAL);
```

```

INSERT INTO STUDENT(sID,std_NAME,physics,chemistry)
values(1058,'RONIKA',35,42),(1059,'VIJAY',44,38); INSERT INTO
STUDENT(STD_NAME,PHYSICS,CHEMISTRY)VALUES('VINEETH',33,7);
SELECT *FROM MARKS;
DROP TABLE MARKS;

```

### Output Screenshot

| Result Grid |      |          |         |           |       |
|-------------|------|----------|---------|-----------|-------|
|             | SID  | STD_NAME | PHYSICS | CHEMISTRY | TOTAL |
|             | 1000 | RONIKA   | 35      | 42        | 77    |
|             | 1002 | RONIKA   | 35      | 42        | 77    |
|             | 1003 | VIJAY    | 44      | 38        | 82    |
|             | 1005 | VIJAY    | 44      | 38        | 82    |
|             | 1058 | RONIKA   | 35      | 42        | 77    |
|             | 1059 | VIJAY    | 44      | 38        | 82    |
|             | 1060 | VINEETH  | 33      | 7         | 40    |
| *           | NULL | NULL     | NULL    | NULL      | NULL  |

|   | MARKID | NAME    | TOTAL_MARKS |
|---|--------|---------|-------------|
| ▶ | 2      | RONIKA  | 77          |
|   | 3      | VIJAY   | 82          |
|   | 4      | RONIKA  | 77          |
|   | 5      | VIJAY   | 82          |
|   | 6      | VINEETH | 40          |
| * | NULL   | NULL    | NULL        |



## **ADVANCED DBMS LAB**

### **Experiment No.: 7**

#### **Aim**

Installation of mongo db on windows

**Name: SHAMJAD MAZOOD NAZER**

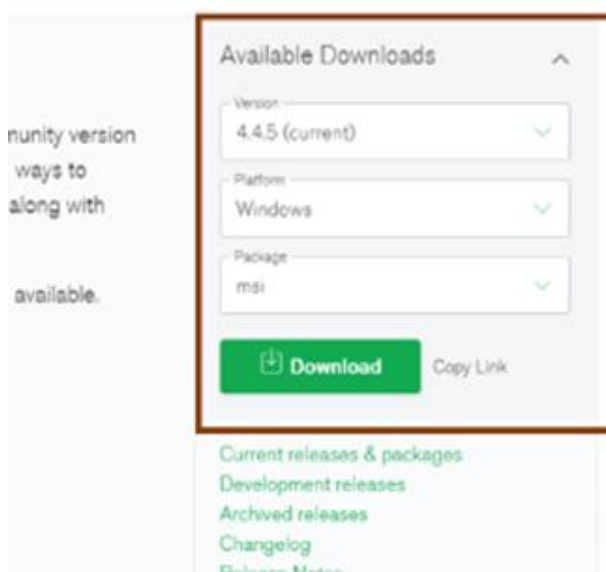
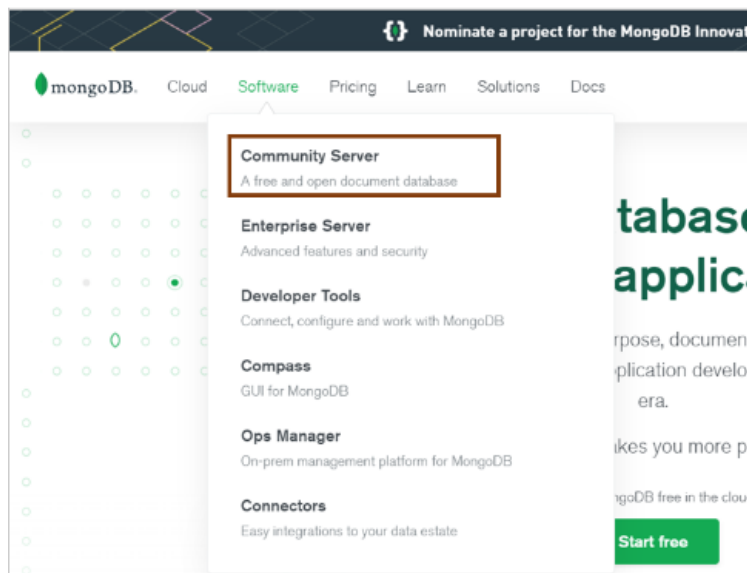
**Roll No: 36**

**Batch: B**

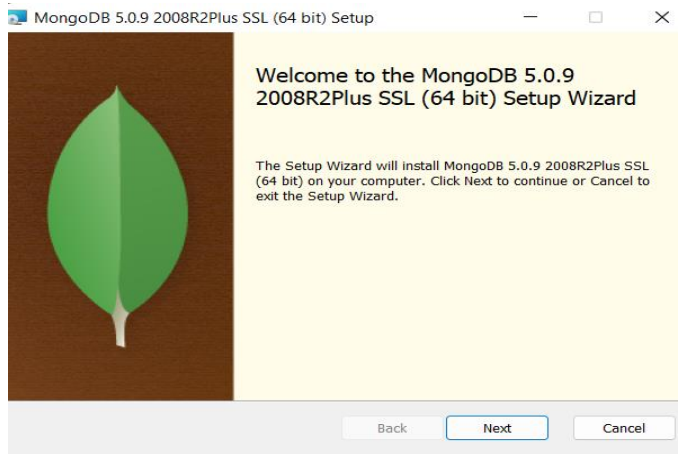
**Date: 24-05-2022**

#### **Procedure**

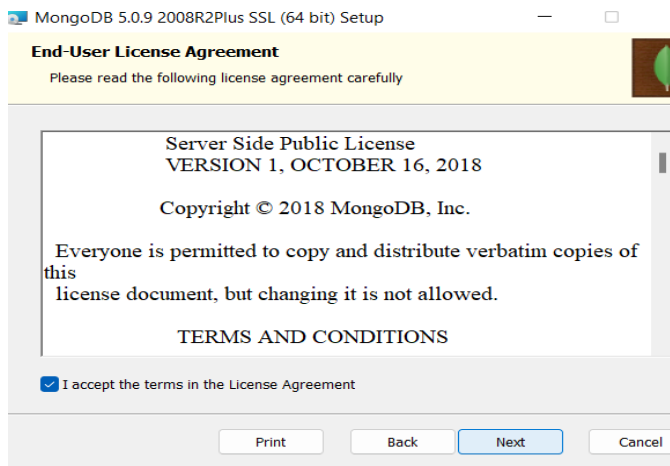
Step 1: download the community server version of MongoDB



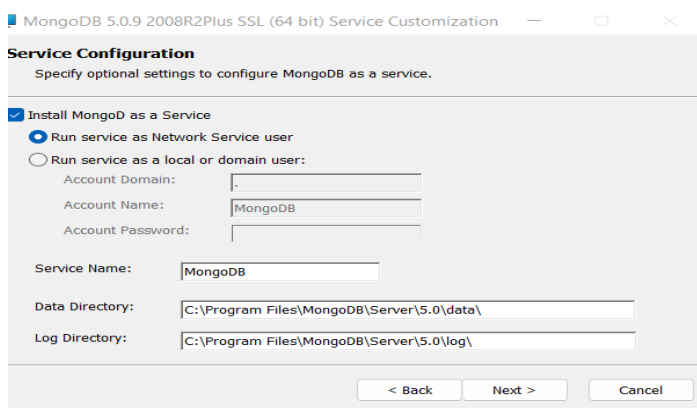
## Step 2: install the software on your pc



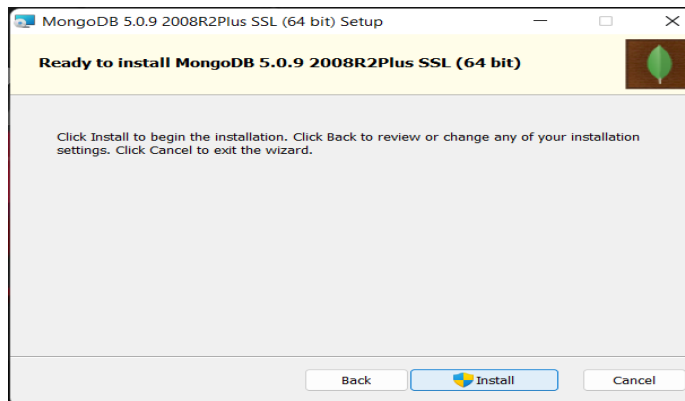
## Step 3: accept the term and conditions



## Step 4: choose service configuration



## Step 5: make necessary changes and install



## Step 6: verify the installation by typing mongo on cmd

```
Command Prompt - mongo
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\VORTEX>mongo
MongoDB shell version v5.0.9
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("4e8b1077-2f0e-406b-85e6-e9524c012cbf") }
MongoDB server version: 5.0.9

=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed
in an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====

The server generated these startup warnings when booting:
2022-06-14T19:37:23.056+05:30: Access control is not enabled for the database. Read and write access could
be performed without authentication. This warning can be silenced by passing --accessControlDefaults=enabled to
the mongod process.
-----
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
-----
> _
```

**ADVANCED DBMS LAB****Experiment No.: 8****Aim**

Designing Databases using NoSQL: MongoDB

**Name: SHAMJAD MAZOOD NAZER**

**Roll No: 36**

**Batch: B**

**Date: 03-06-2022**

**Procedure & Output: -**

- To show Database

```
> show dbs
admin      0.000GB
config     0.000GB
form124    0.000GB
local      0.000GB
vijay      0.000GB
>
```

- To create new Database

```
> use vector
switched to db vector
>
```

- To create collection and show it

```
> db.createCollection("names")
{ "ok" : 1 }
> show collections
names
```

**ADVANCED DBMS LAB****Experiment No.: 9****AIM**

Build sample collections/documents to perform query operations in MongoDB.

**COMMANDS AND OUTPUT****01. Mongo.exe**

```
C:\Program Files\MongoDB\Server\5.0\bin>mongo.exe
MongoDB shell version v5.0.8
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("994c6d89-1190-4e2b-a9c3-06e8eb976ef5") }
MongoDB server version: 5.0.8
=====
```

**02. Create a database.**

➤ use <database\_name>

```
> use rmcab36
switched to db rmcab36
```

**03. Check the database list.**

➤ show dbs

```
--
> show dbs
ajce      0.000GB
admin     0.000GB
config    0.000GB
local     0.000GB
mcadb     0.000GB
rmca2023a 0.000GB
rmcab36   0.000GB
```

**Name: SHAMJAD MAZOOD NAZER**

**Roll No: 36**

**Batch: B**

**Date: 03-06-2022**

**04. Create a collection.**➤ `db.createCollection(name)`

```
> show collections
MCU
```

```
> db.createCollection("DCU")
{ "ok" : 1 }
```

**05. Insert a value into a collection.**➤ `db.collection_name.insert({ document })`

```
> db.MCU.insert({name:"Captain America: The First Avenger"})
WriteResult({ "nInserted" : 1 })
```

**06. Insert many values into a collection with single KEY-VALUE.**➤ `db.collection_name.insertMany({ documents })`

```
> db.MCU.insertMany([ {name:"Iron Man"}, {name:"Thor"}, {name:"Hulk"}, {name:"The Avengers"} ])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("6299af75f2c21322199838ab"),
    ObjectId("6299af75f2c21322199838ac"),
    ObjectId("6299af75f2c21322199838ad"),
    ObjectId("6299af75f2c21322199838ae")
  ]
}
```

**07. Insert many values into a collection.**➔ `db.collection_name.insertMany([ {key1:-:value1}, {key2:value2} ])`

```
> db.DCU.insertMany( [ { _id:100, movie:"The Dark Knight", year:"2008"}, { _id:101, movie:"Man of Steel", year:"2013"}, { _id:102, movie:"Batman v Superman : Dawn of Justice", year:"2016"}, { _id:103, movie:"Wonder Woman", year:"2017"} ] )
{ "acknowledged" : true, "insertedIds" : [ 100, 101, 102, 103 ] }
```

**08.Display the contents in a collection.**

- db.collection\_name.find()

```
> db.MCU.find({})
{ "_id" : ObjectId("6299af1bf2c21322199838aa"), "name" : "Captain America: The First Avenger" }
{ "_id" : ObjectId("6299af75f2c21322199838ab"), "name" : "Iron Man" }
{ "_id" : ObjectId("6299af75f2c21322199838ac"), "name" : "Thor" }
{ "_id" : ObjectId("6299af75f2c21322199838ad"), "name" : "Hulk" }
{ "_id" : ObjectId("6299af75f2c21322199838ae"), "name" : "The Avengers" }
```

**09.Display the contents in a collection in a formatted way.**

- db.collection\_name.find()

```
> db.MCU.find({})
{ "_id" : ObjectId("6299af1bf2c21322199838aa"), "name" : "Captain America: The First Avenger" }
{ "_id" : ObjectId("6299af75f2c21322199838ab"), "name" : "Iron Man" }
{ "_id" : ObjectId("6299af75f2c21322199838ac"), "name" : "Thor" }
{ "_id" : ObjectId("6299af75f2c21322199838ad"), "name" : "Hulk" }
{ "_id" : ObjectId("6299af75f2c21322199838ae"), "name" : "The Avengers" }
```

**10.Update a collection.**

- Db.collection\_name.updateOne({ Selection\_criteria }, { \$set:{ UpdatedKey:UpdatedValue } })

```
> db.DCU.updateOne({_id:101 },[{$set : {"_year":2014}}])
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.DCU.find()
{ "_id" : 100, "movie" : "The Dark Knight", "year" : "2008" }
{ "_id" : 101, "movie" : "Man of Steel", "year" : "2013", "_year" : 2014 }
{ "_id" : 102, "movie" : "Batman v Superman : Dawn of Justice", "year" : "2016" }
{ "_id" : 103, "movie" : "Wonder Woman", "year" : "2017" }
```

**11.Update Many in a collection.**

- db.collection\_name.updateMany({ Selection\_criteria }, { \$set:{ UpdatedKey:UpdatedValue } })

```
> db.DCU.updateMany({movie:"Dawn of Justice"},{$set: {year:"2021"}})
{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }
> db.DCU.find({})
{ "_id" : 100, "movie" : "The Dark Knight", "year" : "2008" }
{ "_id" : 101, "movie" : "Man of Steel", "year" : "2013", "_year" : 2014 }
{ "_id" : 102, "movie" : "Batman v Superman : Dawn of Justice", "year" : "2016" }
{ "_id" : 103, "movie" : "Wonder Woman", "year" : "2017" }
{ "_id" : 104, "movie" : "Dawn of Justice", "year" : "2021" }
{ "_id" : 105, "movie" : "Dawn of Justice", "year" : "2021" }
```



## 12.Delete from a collection.

➤ db.collection\_name.deleteOne/deleteMany({Deletion\_criteria})

```
> db.DCU.deleteMany({year:"2013"})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.DCU.find({})
{ "_id" : 100, "movie" : "The Dark Knight", "year" : "2008" }
{ "_id" : 102, "movie" : "Batman v Superman : Dawn of Justice", "year" : "2016" }
{ "_id" : 103, "movie" : "Wonder Woman", "year" : "2017" }
{ "_id" : 105, "movie" : "Dawn of Justice", "year" : "2021" }
>
```



**ADVANCED DBMS LAB****Experiment No.: 10****Aim**

PHP form data to MongoDB

**Name: SHAMJAD MAZOOD NAZER**

**Roll No: 36**

**Batch: B**

**Date: 06-06-2022**

**QUESTION**

CREATE AN PHP FROM AND STORE THE DATA IN THE MONGODB DATABASE

**PROCEDURE**

INDEX.HTML

```
<html>
<head>
  <title>Document</title>
</head>
<body>
  <h2>insert to mongo</h2>
  <form action="insert.php" method="post">
    <input type="text" name="name" placeholder="name">
    <input type="number" name="rollno" placeholder="rollno">
    <input type="password" name="password" placeholder="password">
    <input type="text" name="firstname" placeholder="firstname">
    <input type="submit" name="submit">
  </form>
</body>
</html>
```

INSERTION.PHP

```
<?php
$mongo = new MongoDB\Driver\Manager("mongodb://localhost:27017");
if(isset($_POST["submit"])){
  $name=$_POST["name"];
  $first_name=$_POST["firstname"];
  $rollno=$_POST["rollno"];
  $passwd=$_POST["password"];
```

```

$writer=new MongoDB\Driver\Bulkwrite;
$writer-
>insert(["name"=>$name,"rollno"=>$rollno,"passwd"=>$passwd,"firstname"=>$first_name]);
$mongo->executeBulkWrite('form124.insertion',$writer);
header("Location:success.html");
die();
}
?>

```

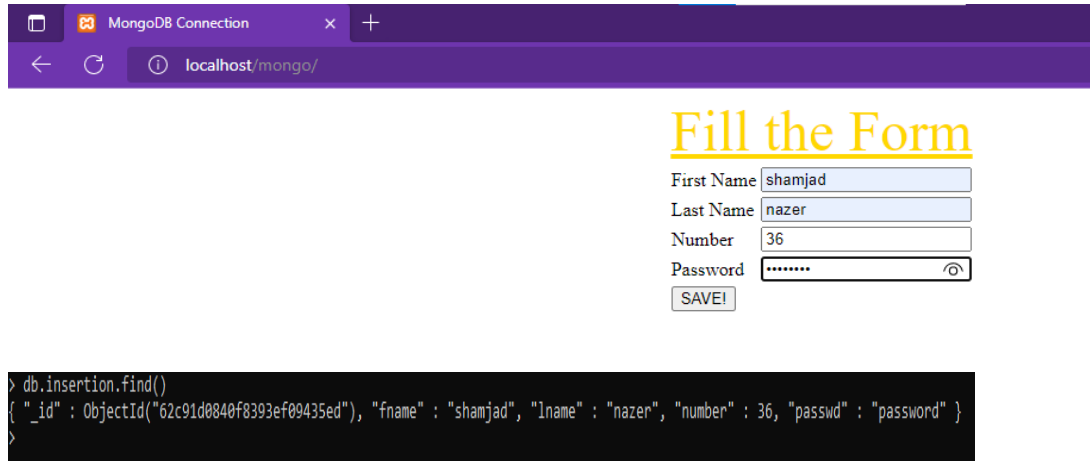
## SUCCESS.HTML

```

<html>
<head> <title>Document</title>
</head>
<body>
  <h2>successfully created</h2>
</body>
</html>

```

## OUTPUT



The screenshot shows a web browser window with a single tab titled 'MongoDB Connection'. The address bar shows 'localhost/mongo/'. The page content includes a heading 'Fill the Form' in yellow, followed by a form with four input fields: 'First Name' (containing 'shamjad'), 'Last Name' (containing 'nazer'), 'Number' (containing '36'), and 'Password' (containing masked characters '\*\*\*\*\*'). A 'SAVE!' button is located below the password field. Below the form is a terminal window with a black background and white text. The terminal shows the command 'db.insertion.find()' and its output: '{ "\_id" : ObjectId("62c91d0840f8393ef09435ed"), "fname" : "shamjad", "lname" : "nazer", "number" : 36, "passwd" : "password" }'.