

MOUNT ZION COLLEGE OF ENGINEERING
Kadammanitta – Pathanamthitta Kerala 689649

Affiliated to APJ Abdul Kalam Technological University)



**20MCA134 ADVANCED DBMS LAB
LABORATORY RECORD
FIRSTYEAR**

Submitted by

PJ SREEDEEP(MZC21MCA-2021)

*Submitted in partial fulfillment of the
requirement for the Award of the Degree of*

**MASTER OF COMPUTER
APPLICATIONS (2021-2023)**

Department of Computer Applications

MOUNT ZION COLLEGE OF ENGINEERING
Kadammanitta – Pathanamthitta Kerala 689649

Affiliated to APJ Abdul Kalam Technological University



CERTIFICATE

Certified that this is a bonafide record of practical work done in Advanced DBMS Lab (20MCA134) Laboratory by P J SREEDEEP Reg No MZC21MCA-2021 of Mount Zion College of Engineering, Kadammanitta – Pathanamthitta during the academic year 2021- 2023

Head of the department

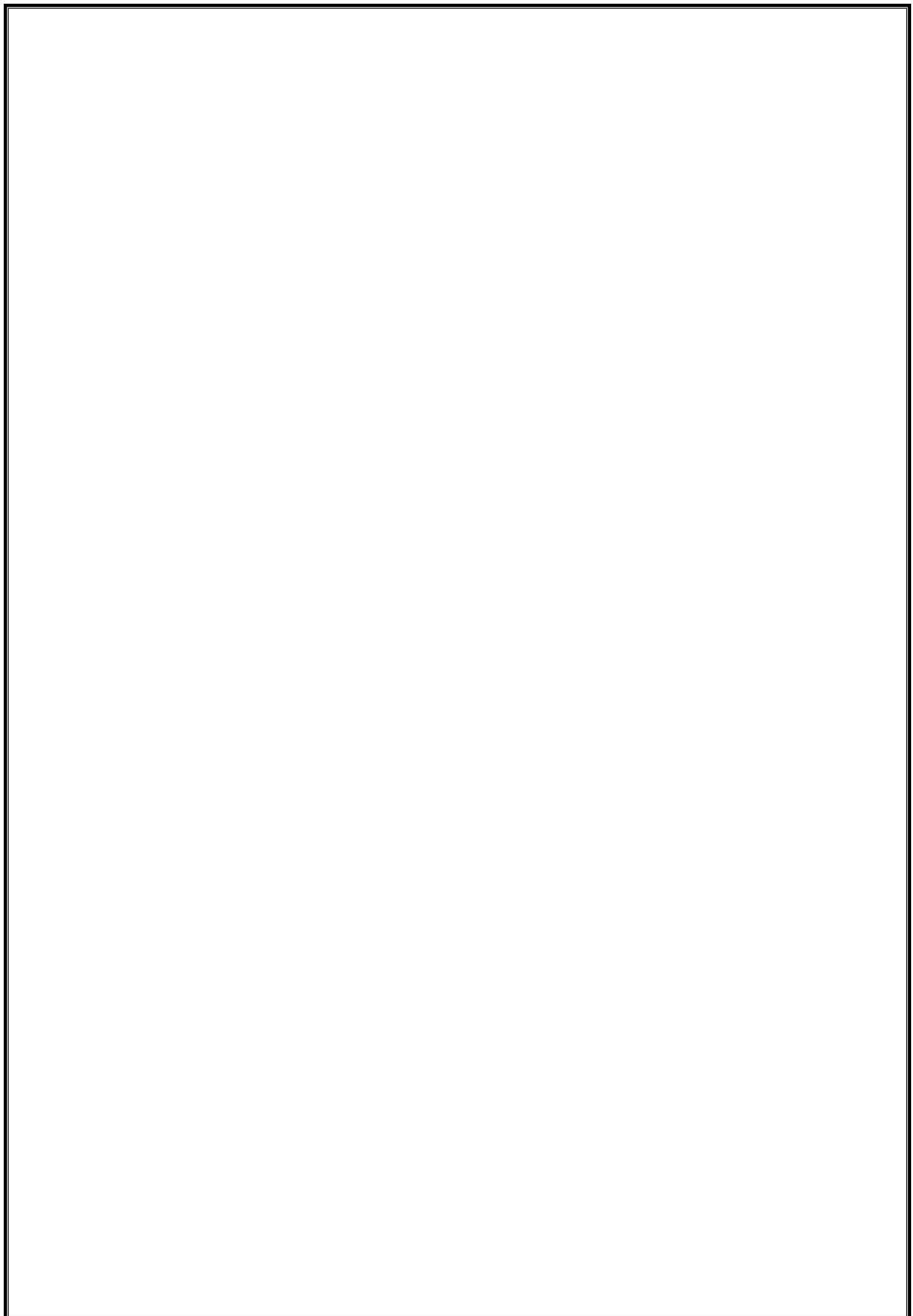
Staff member in-charge

Submitted to the University Examination held on

External Examiner

INDEX

SL.No.	List Of Experiment	Page No.
1	To study various DDL Commands	2
2	To familiarize with selecting data from single table	8
3	To familiarize DDL Commands- ALTER, DROP, TRUNCATE, RENAME	12
4	To familiarize with aggregate functions	20
5	To familiarize with set operations	34
6	To familiarize with join or Cartesian product	39
7	To familiarize with GroupBy and Having clause	43
8	To check whether the given number is prime or not using PL/SQL	47
9	To check whether the given number is palindrome or not using PL/SQL	48
10	To check as certain NO_DATA_FOUND exception using PL/SQL	49
11	To ascertain divide by zero exception using PL/SQL	51
12	Calculate for fixed deposit among using cursors	52
13	Calculate electricity bill using cursors	54
14	Student mark list using cursor	56
15	To perform Banking Operation using Procedures	61
16	Payroll Application using procedures	64
17	Check whether the number is Armstrong or not using functions	67
18	To have familiarize with trigger functions from table Product and Sale item	75
19	To have familiarize with trigger functions. Trigger for department table it will update another table employee	78
20	How to install MongoDB on Windows	81
21	CRUD operations in NoSQL	85
22	Create database, collections using MongoDB Query	89



Introduction

- SQL is Structured Query Language. SQL contains different data types those are:

1. char(size)
2. varchar(size)
3. varchar2(size)
4. date
5. number(p,s) //**P-PRECISION S-SCALE**//
6. number(size)
7. raw(size)
8. raw/longraw(size)

Different types of commands in SQL:

- | | | |
|--------------------|-----|---------------------------------------|
| 1. DDL commands | : - | To create database objects. |
| 2. DML commands | : - | To manipulate data of a database |
| objects | | |
| 3. DQL command | : - | To retrieve the data from a database. |
| 4. DCL/DTL command | : - | To control the data of a database. |

REPORT ON LABORATORY WORK

Name: P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM 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 NULLCANNOT BE 0.

Query:

- 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(CITYIN ('NAGPUR','DELHI','BANGALORE','BOMBAY')) NOT NULL);
- CREATE TABLE BORROW (LOANNO VARCHAR(8) CHECK (LOANNO LIKE'L%') PRIMARYKEY,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%') PRIMARYKEY,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 DEPOSITVALUES('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 INTODEPOSITVALUES('D108','KRANTI','NEHRUPLACE',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('L321','MADHURI','ANDHERI',2000.00);
 - INSERT INTO BORROW VALUES('L371','PRAMOD','VIRAR',8000.00);
 - INSERT INTO BORROW VALUES('L481','S','NEHRU PLACE',3000.00);

INSERTION OF VALUES

1. Inserting values to Branch:

BNAME	CITY
VRCE	NAGPUR
AJNI	NAGPUR
KAROLBAGH	DELHI
CHANDNI	DELHI
DHARAMPETH	NAGPUR
MG ROAD	BANGALORE
ANDHERI	BOMBAY
NEHRU PALACE	DELHI
POWAI	BOMBAY

2. Inserting values to Customer table:

CNAME	CITY
ANIL	CALCUTTA
SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MADHURI	NAGPUR

PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
KRANTI	BOMBAY
NAREN	BOMBAY

3. Inserting values to 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 to borrow table:

LOANNO	CNAME	BNAME	AMOUNT
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

REPORT ON LABORATORY WORK

Name: P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO. 2

AIM

To familiarize with selecting data from single table.

QUESTIONS

1. List all data from table deposit.
2. List all data from borrow.
3. List all data from customer.
4. List all data from branch.
5. Give account no and amount of deposit.
6. Give customer name and account no of depositors.
7. Give name of customers.
8. Give name of branches.
9. Give name of borrows.
10. Give names of customer living in city Nagpur.
11. Give names of depositors having amount greater than 4000.
12. Give account date of Anil.
13. Give name of all branches located in Bombay.
14. Give name of borrower having loan number 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

1. SELECT*FROM DEPOSIT;
2. SELECT*FROM BORROW;
3. SELECT*FROM CUSTOMER;
4. SELECT*FROM BRANCH;
5. SELECT ACTNO,AMOUNT FROM DEPOSIT;
6. SELECT CNAME,ACTNO FROM DEPOSIT;
7. SELECT CNAME FROM CUSTOMER;
8. SELECT BNAME FROM BRANCH;
9. SELECT CNAME FROM BORROW;
10. SELECT CNAME FROM CUSTOMER WHERE CITY='NAGPUR';
11. SELECT CNAME from DEPOSIT where AMOUNT>4000;
12. SELECT ADATE FROM DEPOSIT where CNAME='ANIL';
13. SELECT BNAME from BRANCH where CITY='Bombay';
14. SELECT * from BORROW;
15. SELECT CNAME from BORROW where LOANNO='L205';
16. SELECT CNAME from DEPOSIT WHERE BNAME='VRCE';
17. SELECT BNAME from BRANCH WHERE CITY='Delhi';
18. SELECT CNAME from DEPOSIT WHERE ADATE='1996-12-1';
19. SELECT ACTNO,AMOUNT from DEPOSIT WHERE ADATE BETWEEN '1996-12-1' AND '1996-05-1';
20. SELECT CITY from BRANCH WHERE BNAME='KAROLBAGH';
21. 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|ANJANI|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|VIRAK|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
NEHRU PLACE
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|VIRAK|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
```

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|VIRAK|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

REPORT ON LABORATORY WORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM 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),hire date, salary, commission and insert 5 rows of data.

101	Ramesh	17-Jan -980	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: salnumber(7,2).
 - b. Dropping a column from a table: sal.
 - c. Modifying existing column :ename varchar2(15).

- d. Renaming the tables: emp to emp.
 - e. truncating the tables:emp1.
 - f. Destroying tables:emp.
3. Create a table stud with sname varchar2(20) primary key, rollnonumber(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)).
 5. In table student add check constraint(length(regno<=4)).
 6. Create a table custwith(custid number(6) constraint unique, name char(10)).
 7. Refer the table “stud” in table “ student”.

► SQL Commands

1. 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);
```
2.
 - a) ALTER TABLE emp ADD sal int;
 - b) ALTER TABLE emp DROP COLUMN sal;
 - c) ALTER TABLE emp MODIFY enameVARCHAR(15);
 - d) RENAME table emp to emp1;
 - e) TRUNCATE TABLE emp1;
 - f) DROP TABLE emp1;
3. CREATE TABLE stud(sname varchar(20) PRIMARY KEY, rollno int NOT

NULL, dob date NOT NULL);

SELECT * FROM stud;

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

SELECT * from student;

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

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

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 salnumber(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	SALARY	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	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	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	

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM 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(CIT Y))

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)

REPORT ON LABORATORY WORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM 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 located
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)

REPORT ON LABORATORY WORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM 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)

REPORT ON LABORATORY WORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM 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)

REPORT ON LABORATORY WORK

Name:0 J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PL/SQL

PROGRAM NO. 8

AIM:

To check whether the given no is prime or not using PL/S.

PROGRAM

```
declare
n number(10);i number(10); mnumber(10);
fnumber(10):=0;
begin
n:=&n;
m:=n;
for i in 2..m/2 loop
if(n mod i=0)then
f:=f+1;
endif; end loop;
if(f=0)then
dbms_output.put_line('the given number is prime');
else
dbms_output.put_line('the given number is not prime');
endif;
end;
```

OUTPUT:

Enter value for n: 5

The given number is prime

PL/SQL procedure successfully completed.

REPORT ON LABORATORY WORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO. 9

AIM

To check whether the given no is palindrome or not using PL/SQL.

Program

```
declare
n number(10);
mnumber(10);
s number(10);
r number(10);
begin
n:=&n;
m:=n;
s:=0;
while(n>0) loop
r:=n mod 10; s:=(s*10)+r; n:=floor(n/10);
end loop; if(m=s)then
dbms_output.put_line('The given number is a palindrome');
else
dbms_output.put_line('The given number is not a palindrome');
endif;
end;
```

OUTPUT:

Enter value for n: 121
The given number is a palindrome

PL/SQL procedure successfully completed.

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

EXCEPTION

PROGRAM NO. 10

AIM

To check ascertain NO_DATA_FOUND exception using

PL/SQL.

COMMANDS

```
SQL>create table emp5(empno number(3),ename varchar(10),deptno  
number(3));
```

Table created.

```
SQL>insert into emp5 values(:empno,:ename,:deptno);  
SQL>select*from emp5;
```

EMPNO	ENAME	DEPTNO
101	aaa	1
102	bbb	2

PROGRAM:

```
declare
veno emp5.empno%type:=:empno;
vename emp5.ename%type:=:ename; vdeptno emp5.deptno%type;
begin
>select empno,ename,deptno into veno,vename,vdeptno from emp5 where
empno=veno and ename=vename; dbms_output.put_line('deptno is'||vdeptno);
exception
when no_data_found then
dbms_output.put_line('no records found');
end;
```

OUTPUT:

```
Enter value for eno: 101
Enter value for ename:aaadeptno is1
PL/SQLprocedure successfullycompleted.
Enter value for eno: 103
Enter value for ename:sss
no records found
PL/SQLprocedure successfullycompleted.
```

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO. 11

AIM

To ascertain divide by zero exception using PL/SQL

PROGRAM:

```
declare
anumber(3);
b number(3);
begin
a:=a;
b:=b;
a:=a/b;
dbms_output.put_line('Result is'||a);
exception
whenzero_dividethen
dbms_output.put_line('dividebyzeroerror');
end;
```

OUTPUT:

Enter value for a:5 Enter value for b:0

dividebyzeroerror

PL/SQL procedure successfully complete

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

CURSORS

PROGRAM NO. 12

AIM

Calculating interest for fixed deposit amount using cursors.

QUESTION

To Calculate Interest for Fixed Deposit Amount Using Cursors.

TABLE:

SQL>create table fixed(accno number(5),years number(4),amount number(6),interest number(3)); Table created.

SQL>insert into fixed(accno,years,amount) values(&accno,&years,&amount);
SQL> select * from fixed;

ACCNO	YEARS	AMOUNT	INTEREST
101	3	1000	
102	2	500	
103	6	5000	

PROGRAM:

```
declare
cursorcurselect*fromfixed;
begin
foriincur loop
ifi.years<=2then
updatefixedsetinterest=i.amount*0.09*(1/12)
whereaccno=i.accno; elsifi.years>=2 and i.years<=5 then
updatefixedsetinterest=i.amount*0.11*(1/12)

whereaccno=i.accno;
else
updatefixedsetinterest=i.amount*0.15*(1/12)where
accno=i.accno;
endif; end loop;
end;
```

OUTPUT:

PL/SQL procedure successfully completed. SQL> select * from fixed;

ACCNO	YEARS	AMOUNT	INTEREST
101	3	1000	9
102	2	500	4
103	6	5000	62

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO. 13

AIM

Calculating electricity bill using cursors.

QUESTION:

To Calculate Electricity Bill Using Cursors.

TABLE:

```
SQL>create table eb_cal(ebno number(5),name varchar(15),units  
number(4),charges number(3)); Table created.
```

```
SQL>insert into eb_cal(ebno,name,units)values(:ebno,:name,:units);
```

```
SQL> select * from eb_cal;
```

EBNO	NAME	UNITS	CHARGES
1	aaa	150	
2	bbb	80	
3	ccc	300	

PROGRAM:

```
declare
cursorcurisselect*fromeb_cal;
begin
foriincur loop
ifi.units<=100then
updateeb_calsetcharges=i.units*1where
ebno=i.ebno;
elsifi.units>100andi.units<=200then
updateeb_calsetcharges=i.units*2where
ebno=i.ebno;
else
updateeb_calsetcharges=i.units*3where
ebno=i.ebno;
endif; end loop;
end;
```

OUTPUT:

PL/SQLproceduresuccessfullycompleted.

SQL>select*from eb_cal;

EBNO	NAME	UNITS	CHARGES
1	aaa	150	450
2	bbb	80	80
3	ccc	300	600

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO. 14

AIM

Student Mark List Using Cursor.

QUESTION

To Prepare Students Mark List Using Cursors

TABLE:

```
SQL>create table stu11(regno number(6) primary key, mark1 number(3),mark2  
number(3),mark3 number(3),mark4 number(3),mark5 number(3),tot number(3),avg  
number(3,2));
```

Table created.

```
SQL>insert into stu11(regno,mark1,mark2,mark3,mark4,mark5)  
values(:regno,:mark1,:mark2,:mark3,:mark4,:mark5);
```

PROGRAM:

```
declare
```

```
cursor is select * from stu11; x stu11.regno%type;
```

```
a stu11.mark1%type; b stu11.mark2%type; f stu11.mark3%type; d  
stu11.mark4%type; e stu11.mark5%type; t stu11.tot%type;
```

```
vstu11.avg%type;
```

```
begin openc; loop
```

```
fetch c into x,a,b,f,d,e,t,v; exit when c% not found;
```

```
dbms_output.put_line('Rollno:'||x); dbms_output.put_line('mark1:'||a);
dbms_output.put_line('mark2:'||b); dbms_output.put_line('mark3:'||f);
dbms_output.put_line('mark4:'||d); dbms_output.put_line('mark5:'||e);
dbms_output.put_line('update the tot'); dbms_output.put_line('update the avg');
```

update stu11 set

```
tot=(mark1+mark2+mark3+mark4+mark5)where regno=x; update stu11 set
avg=(tot/5) where regno=x;
```

endloop; close c;

end;

OUTPUT:

roll:101

m1:50

m2:60

m3:70

m4:80

m5:90

update the tot update the avg roll:102

m1:30

m2:40

m3:50

m4:60

m5:70

update the tot update the avg roll:103

m1:45

m2:55

m3:65

m4:75

m5:85

update the tot update the avg

roll:104

m1:55

m2:65

m3:75

m4:85

m5:95

update the tot update the avg

PL/SQL procedure successfully completed. SQL> select * from stu11;

ROLL	M1	M2	M3	M4	M5	TOT	AVG
101	50	60	70	80	90	350	70
102	30	40	50	60	70	250	50
103	45	55	65	75	85	325	65
104	55	65	75	85	95	375	75

COPYING A TABLE USING CURSORS

AIM

Copying of tables using Cursors.

TABLE:

```
SQL>create table stu11(regno number(6) primary key, mark1 number(3),mark2  
number(3),mark3 number(3),mark4 number(3),mark5 number(3),tot number(3),avg  
number(3,2));
```

Table created.

```
SQL>insert into stu11(regno,mark1,mark2,mark3,mark4,mark5)  
values(:regno,:mark1,:mark2,:mark3,:mark4,:mark5);
```

```
SQL>select* fromstu11;
```

ROLL	M1	M2	M3	M4	M5	TOT	AVG
101	50	60	70	80	90	350	70
102	30	40	50	60	70	250	50
103	45	55	65	75	85	325	65
104	55	65	75	85	95	375	75

```
SQL>create table st11(regno number(6) primary key, mark1 number(3),mark2  
number(3),mark3 number(3),mark4 number(3),mark5 number(3),tot number(3),avg  
number(3,2));
```

Table created.

PROGRAM:

```
declare
cursorcurisselect*fromstu11;
begin
for iincur
loop
insert into st11 values(i.regno,i.mark1,i.mark2,i.mark3,i.mark4,i.mark5, i.tot,i.avg);
endloop;
end;
```

OUTPUT:

PL/SQL procedure successfully completed. SQL> select* from st11;

ROLL	M1	M2	M3	M4	M5	TOT	AVG
101	50	60	70	80	90	350	70
102	30	40	50	60	70	250	50
103	45	55	65	75	85	325	65
104	55	65	75	85	95	375	75

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO. 15

FUNCTIONS

ALGORITHM:

1. Define a function along with the arguments with its appropriate data types.
2. Select command, re level, max level from the table and copy the same values to the new variables which have been declared.
3. Add qty hand and relevel is less than max level and if a gain max value to arguments.
4. If it is not less, add the values of qty hand and re level and assign into arguments.
5. Close the cursor and stop the execution.
6. Close the if condition.
7. Stop the execution.

BANKING APPLICATIONUSING PROCEDURES

AIM

To Perform Banking Operations Using Procedures.

TABLE:

```
SQL>create table acc(ac_no number(5), name varchar(20), balance number(6));
```

Table created.

```
SQL>insert into acc values(:ac_no,:name,:balance);
```

```
SQL> select * from acc;
```

AC_NO	NAME	BALANCE
101	aaa	10000
102	bbb	6000
103	ccc	15000
104	ddd	5500

PROCEDURE DEFINITION:

```
Create or replace procedure with draw(ac_no1in number,amount1innumber)is  
begin  
update accse tbalance=balance-amount1wherer_no=ac_no1;  
end;  
create or replace procedure depositt (ac_no1 innumber,amount1 innumber)is  
begin  
update accset balance=balance+amount1whereac_no=ac_no1;  
end;
```

PROGRAM:

```
declare  
choice number; ac_no1 number(5); amount number(5);  
begin  
ac_no1:= :ac_no1; choice:= :choice; amount:= :amount; if choice=1 then  
depositt(ac_no1,amount);  
else
```

```
with draww(ac_no1, amount); end if;  
end;
```

OUTPUT:

Enter value for ac_no1:103

Enter value for choice: 1 Enter value for amount:2000

PL/SQL procedure successfully completed. Enter value for ac_no1: 103

Enter value for choice: 2 Enter value for amount:1000

PL/SQL procedure successfully completed.

SQL>select*from acc;

AC_NO	NAME	BALANCE
101	aaa	10000
102	bbb	6000
103	ccc	16000
104	ddd	5500

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO.16

AIM

Pay Roll Application Using Procedures.

QUESTION:

To carry out pay roll application using procedures.

TABLE:

SQL> create table pay(eno number(10),ename varchar(10),hra number(10),da number(10)); Table created.

SQL>insert into pay values(:eno,:ename,:hra,:da);

SQL>select*from pay;

ENO	ENAME	HRA	DA
1	Kalidass	1200	1000
2	Ravi	2000	1000
3	David	5000	500
4	Dravid	10000	2000
5	Dass	30000	5000

SQL> alter table pay add netamt number(10); Table altered.

SQL>select* from pay;

ENO	ENAME	HRA	DA	NETAMT
1	Kalidass	1200	1000	
2	Ravi	2000	1000	
3	David	5000	500	
4	Dravid	10000	2000	
5	Dass	30000	5000	

PROCEDURE DEFINITION:

```
create or replace procedure net(eno1 in number)is  
begin  
update pay set netamt=hra+da where eno=eno1;  
end;
```

PROGRAM:

```
declare  
choice number; eno1 number(5);  
begin  
eno1:= :eno1; choice:=:choice; if choice=1 then  
net(eno1);  
else  
dbms_output.put_line('PLEASEENTERCHOICE=1');  
endif;  
end;
```

OUTPUT

Enter value for eno1: 1 Enter value for choice:1

PL/SQL procedure successfully completed.

SQL>select*from pay;

ENO	ENAME	HRA	DA	NETAMT
1	Kalidass	1200	1000	2200
2	Ravi	2000	1000	
3	David	5000	500	
4	Dravid	10000	2000	
5	Dass	30000	5000	

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO. 17

AIM

Check whether a number is armstrong or not using functions.

QUESTION

To Check whether a number is Armstrong or not using functions.

FUNCTION BODY:

```
create or replace function armstrong(n number) return number is r number(10);
a number(10); bnumber(10); cnumber(10);
begin
b:=0;
c:=n; while(c>0) loop
r:=c mod 10; b:=b+(r*r*r); c:=floor(c/10); end loop;return b;
end armstrong;
```

PROGRAM:

```
declare
n number(10); mnumber(10); o number(10);
begin
```

```
n:=n;  
o:=n; m:=armstrong(n); if(m=o) then  
dbms_output.put_line('Given no is Armstrong number');  
else  
dbms_output.put_line('Given no is not an armstrong number');  
endif;  
end;
```

OUTPUT

Enter value forn:153

Given no is Armstrong number

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO. 18

AIM

To have familiarize with trigger functions

QUESTION

Create two tables named:

Product(PdtId, Pname, Proce, Qtyinstock),

and

Saleitem(saleId, PdtId, Qty).

Create a trigger called update Available Quantity that updates the quantity in stock in the product table, for every product sold. The trigger should be executed after each insert operation on the Sale item table: for the product with given PdtId(the one inserted into Sale item),update the available quantity in Product table to be the old quantity minus the sold quantity.

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

```
>CREATE TABLE PRODUCT(PDTID VARCHAR(50) NOT NULL PRIMARY KEY,PNAME  
VARCHAR(100),PRICE INT,QTYINSTOCK INT);  
  
>CREATE TABLE SALEITEM(SALEID INT NOT NULL,PDTID VARCHAR(50) NOT NULL,QTY  
INT);  
  
>INSERT INTO PRODUCT VALUES (:PDTID,:PNAME,:PRICE,:QTYINSTOCK);  
  
>SELECT * FROM PRODUCT;
```

Results Explain Describe Saved SQL History

PDTID	PNAME	PRICE	QTYINSTOCK
101	ANU	40	12
102	JSW	121	16
103	STEEL	1245	36

3 rows returned in 0.00 seconds [CSV Export](#)

```
>INSERT INTO SALEITEM VALUES(:SALEID,:PDTID,:QTY);  
  
>SELECT * FROM SALEITEM;
```

Results Explain Describe Saved SQL History

SALEID	PDTID	QTY
201	103	10
201	103	10
201	103	5
202	103	6

4 rows returned in 0.00 seconds [CSV Export](#)

```
>CREATE OR REPLACE TRIGGER UPDATEAVAILABLEQUANTITY  
AFTER INSERT ON SALEITEM  
FOR EACH ROW  
BEGIN  
UPDATE PRODUCT SET QTYINSTOCK=QTYINSTOCK-:NEW.QTY WHERE PDTID=:NEW.PDTID;  
END
```

AFTER EXECUTION OF TRIGGER

>SELECT * FROM PRODUCT;

Results Explain Describe Saved SQL History

PDTID	PNAME	PRICE	QTYINSTOCK
101	ANU	40	12
102	JSW	121	16
103	STEEL	1245	30

3 rows returned in 0.00 seconds [CSV Export](#)

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO. 19

AIM

To have familiarize with trigger functions.

QUESTION

Create a Trigger for dept1 table it will update another table empp3 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

```
>create table dept1(dept_idnumber(5),dept_name varchar(20)not null,mgr_id number(30),loc_id number(20));  
>insert into dept1 values(:dept_id,:dept_name,:mgr_id,:loc_id);  
>select * from dept1;
```

Results Explain Describe Saved SQL History

DEPT_ID	DEPT_NAME	MGR_ID	LOC_ID
100	sales	12	11
105	accounting	12	14
106	accounting	12	15
107	sales	12	16
108	accounting	12	17
102	accounting	12	16
213	sales	12	17
213	sales	12	17
101	accounting	12	14

9 rows returned in 0.00 seconds [CSV Export](#)

```
>create table empp3(empid number(5)primary key not null,first_name varchar(30),last_name varchar(30),job_id varchar(20),manager_id number(5),emp salary number,dept_id number(5));
```

```
>select * from empp3;
```

Results Explain Describe Saved SQL History

EMPID	FIRST_NAME	LAST_NAME	JOB_ID	MANAGER_ID	EMPSALARY	DEPT_ID
101	Ratheesh	Rajeev	1000	11	10000	210
102	Shyam	Murali	1001	12	70000	211
103	Abraham	Thomas	1002	12	90000	212
104	bino	sebasten	1003	12	50000	213
105	NEETHU	GANESH	1004	12	84000	214

5 rows returned in 0.00 seconds [CSV Export](#)

```
create or replace trigger trig1
```

```
after insert on dept1 for each row
```

```
begin
```

```
update empp3 set empsalary=60000 where dept_id=213;
```

```
end;
```

Results Explain Describe Saved SQL History

Trigger created.

0.00 seconds

insert into dept1 values(:deptid,:deptname,:mgrid,:locid);

select * from empp3;

Results Explain Describe Saved SQL History

EMPID	FIRST_NAME	LAST_NAME	JOB_ID	MANAGER_ID	EMPSALARY	DEPT_ID
101	Ratheesh	Rajeev	1000	11	10000	210
102	Shyam	Murali	1001	12	70000	211
103	Abraham	Thomas	1002	12	90000	212
104	bino	sebasten	1003	12	60000	213
105	NEETHU	GANESH	1004	12	84000	214

5 rows returned in 0.00 seconds

[CSV Export](#)

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO. 20

► How to Download & Install MongoDB on Windows

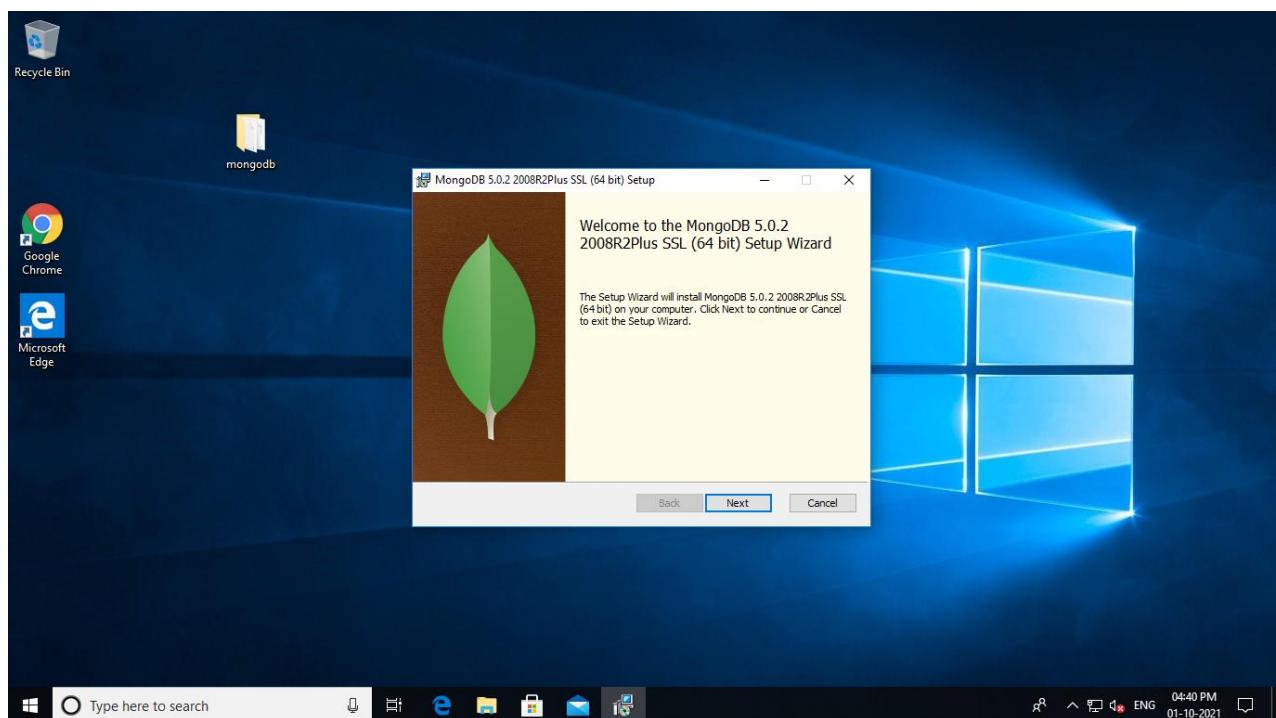
The installers for MongoDB are available in both the 32-bit and 64-bit format. The 32-bit installers are good for development and test environments. But for production environments you should use the 64-bit installers. Otherwise, you can be limited to the amount of data that can be stored within MongoDB.

► Download & Install MongoDB on Windows

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

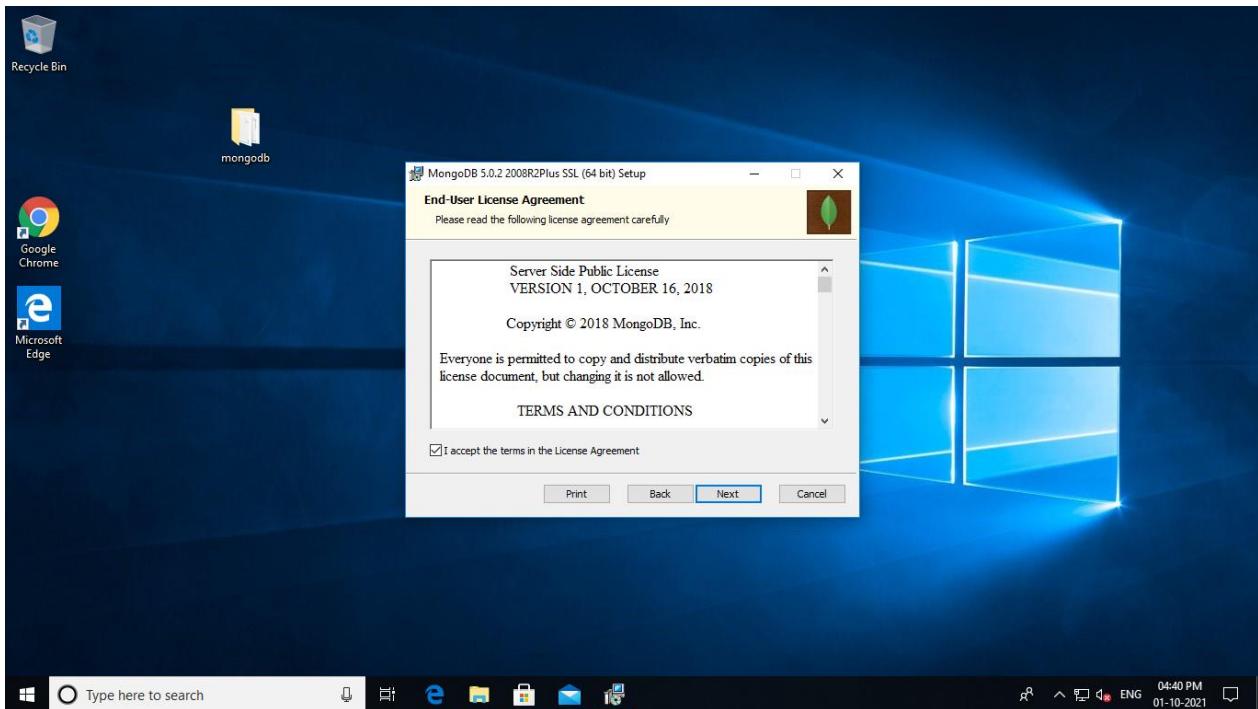
Step 1:

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



Step 2:

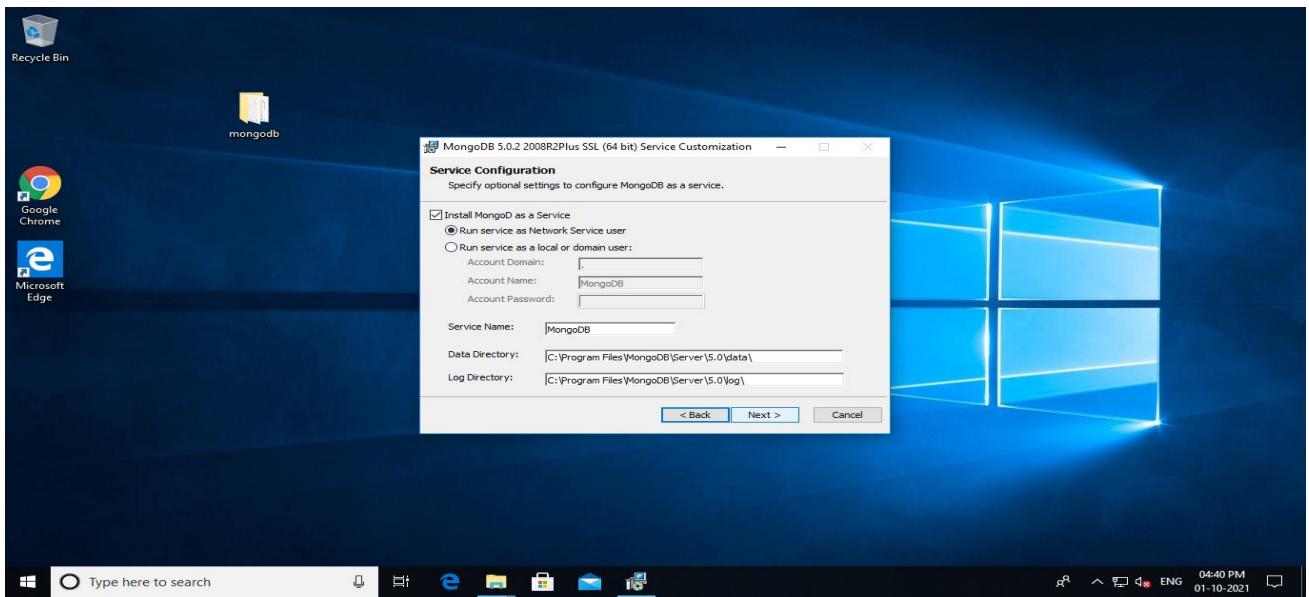
1. Accept the End-User License Agreement
2. Click Next



Step 3:

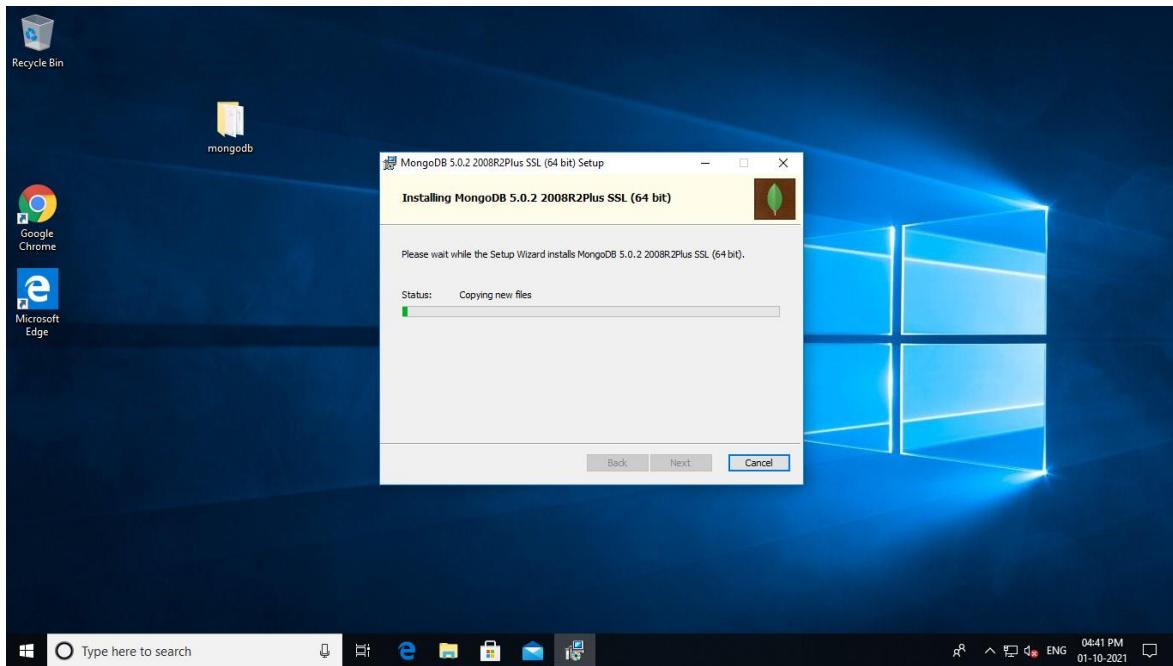
Click on the "complete" button to install all of the components. The custom option can be used to install selective components or if you want to change the location of the installation.

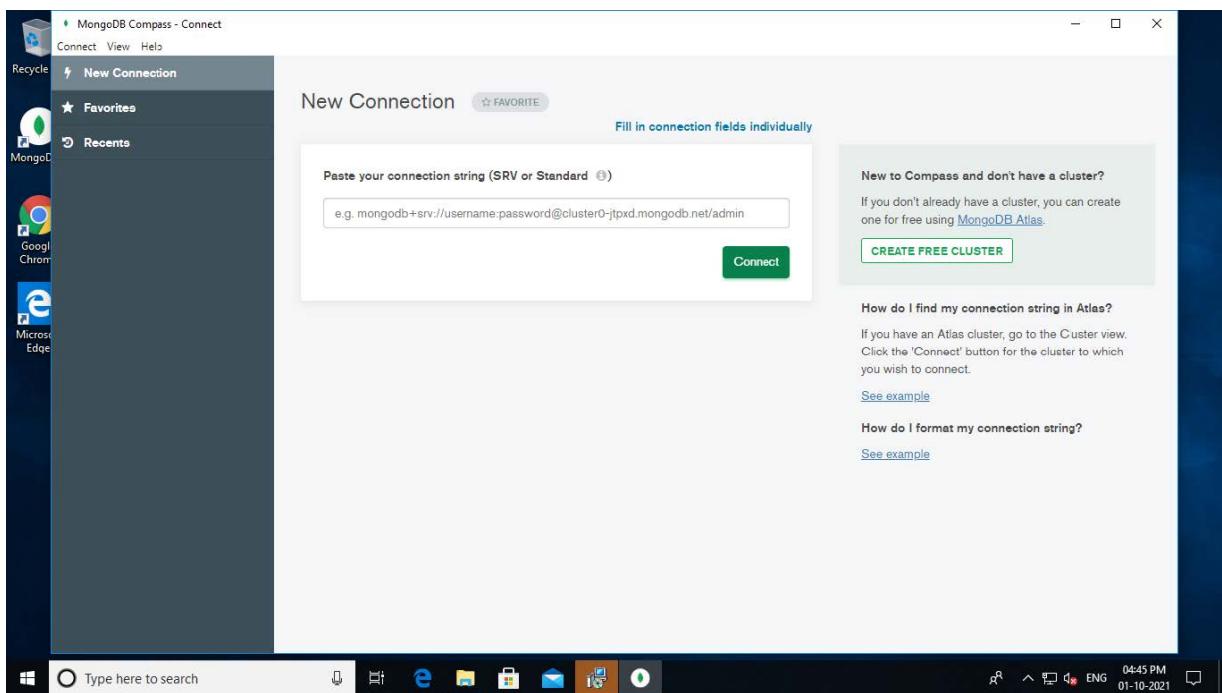
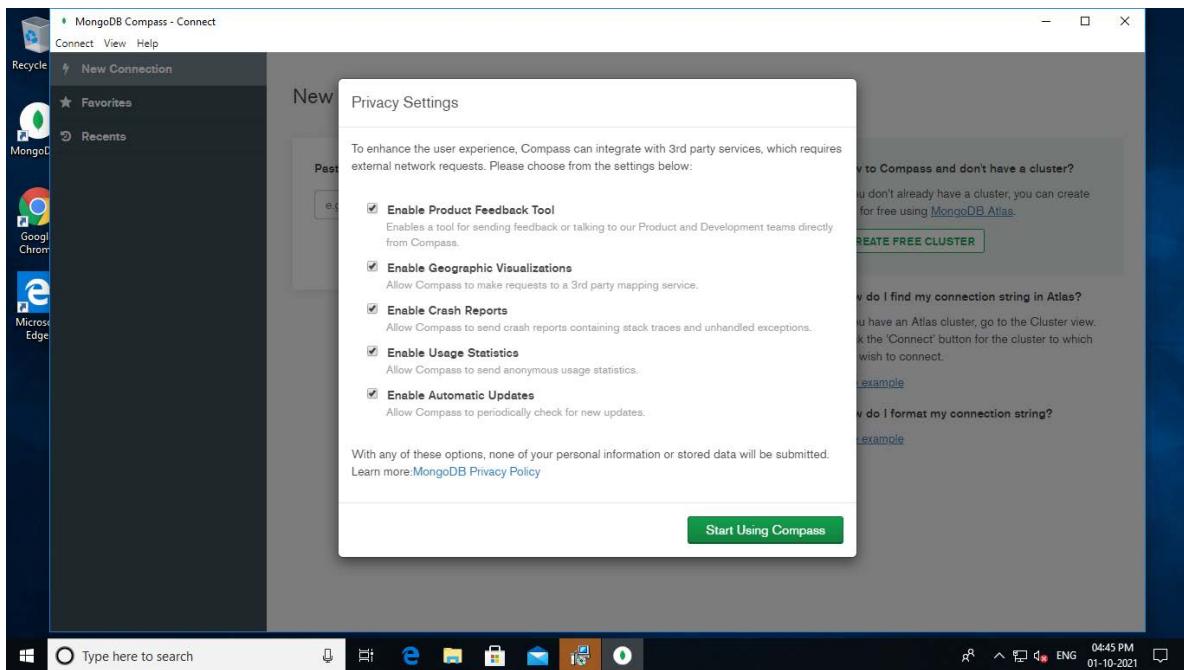
1. Select “Run service as Network Service user”. Make a note of the data directory, we’ll need this later.
2. Click Next



Step 4:

1. Click on the Install button to start the installation.
2. Installation begins. Click next once completed. Click on the
3. Finish button to complete the installation.





MongoDB Compass - localhost:27017

Connect View Help

Local

3 DBS 1 COLLECTIONS

FAVORITE

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 5.0.2 Community

Filter your data

admin

config

local

Databases Performance

CREATE DATABASE

Database Name	Storage Size	Collections	Indexes
admin	20.0KB	0	1
config	4.0KB	0	2
local	20.0KB	1	1

_MONGOSH

Type here to search

04:45 PM 01-10-2021

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM NO. 21

► CRUD Operations in NoSql.

1. Create a database school and create a collection class.

```
on Command Prompt - mongo
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>mongo
MongoDB shell version v4.4.9
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("41fb551d-62c7-4934-96f1-1a6f63ccae3") }
MongoDB server version: 4.4.9
---
The server generated these startup warnings when booting:
2021-10-13T12:17:04.559+05:30: Access control is not enabled for the database. Read and write access
2021-10-13T12:17:04.560+05:30: This server is bound to localhost. Remote systems will be unable to c
d serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start i
---
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()
---

> show dbs
Test      0.000GB
admin     0.000GB
config    0.000GB
food      0.000GB
local     0.000GB
student   0.000GB
> use school
switched to db school
> db.class
school.class
>
```

2. Insert values into class.

```
> db.class.insert([{"id":"5","name":"sree","mark":"60"}]);  
BulkWriteResult({  
    "writeErrors" : [ ],  
    "writeConcernErrors" : [ ],  
    "nInserted" : 1,  
    "nUpserted" : 0,  
    "nMatched" : 0,  
    "nModified" : 0,  
    "nRemoved" : 0,  
    "upserted" : [ ]  
})  
>
```

```
> db.class.insertMany([{"id":"1","name":"anu","mark":"50"}, {"id":"2","name":"appu","mark":"52"},  
... {"id":"3","name":"manu","mark":"45"}, {"id":"4","name":"arya","mark":"55"}]);  
{  
    "acknowledged" : true,  
    "insertedIds" : [  
        ObjectId("61668eef1712115868448dd9"),  
        ObjectId("61668eef1712115868448dda"),  
        ObjectId("61668eef1712115868448ddb"),  
        ObjectId("61668eef1712115868448ddc")  
    ]  
}
```

3. Display the whole data in the collection 'class'.

```
> db.class.find();  
{ "_id" : ObjectId("61668eef1712115868448dd9"), "id" : "1", "name" : "anu", "mark" : "50" }  
{ "_id" : ObjectId("61668eef1712115868448dda"), "id" : "2", "name" : "appu", "mark" : "52" }  
{ "_id" : ObjectId("61668eef1712115868448ddb"), "id" : "3", "name" : "manu", "mark" : "45" }  
{ "_id" : ObjectId("61668eef1712115868448ddc"), "id" : "4", "name" : "arya", "mark" : "55" }  
{ "_id" : ObjectId("61668f511712115868448ddd"), "id" : "5", "name" : "sree", "mark" : "60" }  
>
```

4. Display the details of the student with id=4.

```
> db.class.find({"id":"4"});  
{ "_id" : ObjectId("61668eef1712115868448ddc"), "id" : "4", "name" : "arya", "mark" : "55" }  
>
```

5. Display the details of the students whose mark is greater than 50.

```
> db.class.find({"mark":{$gt:"50"}});  
{ "_id" : ObjectId("61668eef1712115868448dda"), "id" : "2", "name" : "appu", "mark" : "52" }  
{ "_id" : ObjectId("61668eef1712115868448ddc"), "id" : "4", "name" : "arya", "mark" : "55" }  
{ "_id" : ObjectId("61668f511712115868448ddd"), "id" : "5", "name" : "sree", "mark" : "60" }  
>
```

6. Display the details of the students whose mark is greater than or equal to 55.

```
> db.class.find({"mark":{$gte:"55"}});  
{ "_id" : ObjectId("61668eef1712115868448ddc"), "id" : "4", "name" : "arya", "mark" : "55" }  
{ "_id" : ObjectId("61668f511712115868448ddd"), "id" : "5", "name" : "sree", "mark" : "60" }  
>
```

7. Display the details of the students whose mark is less than 50.

```
> db.class.find({"mark": {"$lt": "50"}});
{ "_id" : ObjectId("61668eef1712115868448ddb"), "id" : "3", "name" : "manu", "mark" : "45" }
>
```

8. Display the details of the students whose mark is less than or equal to 50.

```
> db.class.find({"mark": {"$lte": "50"}});
{ "_id" : ObjectId("61668eef1712115868448dd9"), "id" : "1", "name" : "anu", "mark" : "50" }
{ "_id" : ObjectId("61668eef1712115868448ddb"), "id" : "3", "name" : "manu", "mark" : "45" }
>
```

9. Display the details of the student whose mark is equal to 60.

```
> db.class.find({"mark": {"$eq": "60"}});
{ "_id" : ObjectId("61668f511712115868448ddd"), "id" : "5", "name" : "sree", "mark" : "60" }
>
```

10. Find out the total number of entries to the class.

```
> db.class.count();
5
>
```

11. Update the name of the student Anu to Anupama.

```
> db.class.update({"id": "1"}, {"$set": {"name": "Anupama"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.class.find({"id": "1"});
{ "_id" : ObjectId("61668eef1712115868448dd9"), "id" : "1", "name" : "Anupama", "mark" : "50" }
>
```

```
> db.class.update(
... {
...   "id": "5"
... },
... {
...   "$set":
...   {
...     "name": "sreerag", "mark": "57"
...   })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
> db.class.find({"id": "5"});
{ "_id" : ObjectId("61668f511712115868448ddd"), "id" : "5", "name" : "sreerag", "mark" : "57" }
>
```

12. Remove the student whose id equal to2.

```
> db.class.remove({"id":"2"});
WriteResult({ "nRemoved" : 1 })
>
```

13. Drop the collectionclass.

```
> db.class.drop();
true
>
```

REPORT ON LABORATORYWORK

Name:P J SREEDEEP	Roll No:20	Name of lab: Advanced DBMS Lab	Period:
Class: S2, MCA	Date:	Name of lab work: practical	Batch:2021-2023

PROGRAM. 22

AIM

Create a database named empDb and then Create a collection named employees. Write a MongoDB Query to perform CRUD operations , view and search operations as a menu driven application using python.

PROGRAM

```
import pymongo
connection=pymongo.MongoClient("mongodb://localhost:27017")
mydb=connection["empDb"]
myemployee=mydb["employees"]
while True:
    print("\n Menu \n")
    print("1. Add an Employee \n")
    print("2. View all Employees \n")
    print("3. Search an Employee \n")
    print("4. Delete an Employee \n")
    print("5. Update an Employee \n")
    print("6. Employee name starts with \n")
    print("7. Exit \n")
```

```
option=int (input("Select an Option"))

if option==1:

print("Add option is Selected")

empCode=input("Enter the employee Code")

empName=input("Enter the employee Name")

empDesignation=input("Enter the employee Designation")

data={"empCode":empCode,"empName":empName,"empDesignation":empDesignation}

print(data)

myemployee.insert_one(data)

print("Data inserted sucessfully")



elif option==2:

print("View all Employees Option Selected")

result=myemployee.find()

#print(result)

for i in result:

    print (i)

elif option==3:

print("Search option is selected")

empCode=input("Enter the employee code")

data={"empCode":empCode}

result=myemployee.find(data,{"_id":0})

for i in result:

    print (i)
```

```

elif option==4:
    print("Delete option is selected")
    empCode=input("Enter the employee code to Delete")
    data={"empCode":empCode}
    myemployee.delete_one(data)
    print("data deleted successfully")

elif option==5:
    print("Update option is selected")
    empCode=input("Enter an employee Code to be updated")
    empName=input("Enter a name to be updated")
    empDesignation=input("Enter a Designation to be updated")
    setData={"empCode":empCode}

    newData={"$set":{"empName":empName,"empDesignation":empDesignation}}
    myemployee.update_one(setData,newData)
    print("Data updated successfully")

elif option==6:
    ip=input("Enter the employee name starting letter to be searched")
    condition={"empName": {"$gte":ip} }

    result=myemployee.find(condition,{"_id":0,"empDesignation":1,"empName":1}).sort("empName",-1)

    #print(result)
    for i in result:
        print (i)

else:

```

```
break
```

OUTPUT

Menu

1. Add an Employee
2. View all Employees
3. Search an Employee
4. Delete an Employee
5. Update an Employee
6. Employee name starts with
7. Exit

1) Select an Option1

Add option is Selected

Enter the employee Code100

Enter the employee NameGawri

Enter the employee DesignationCEO

```
{'empCode': '100', 'empName': 'Gawri', 'empDesignation': 'CEO'}
```

Data inserted sucessfully

2) Select an Option2

View all Employees Option Selected

```
{'_id': ObjectId('62ce675117aab0bddd09440'), 'empCode': '003', 'empName': 'sheri', 'empDesignation': 'Manager'}
```

```
{'_id': ObjectId('62ce67a4505c8ab5fe901b3c'), 'empCode': '004', 'empName': 'jes', 'empDesignation': 'Marketing mg'}
```

```
{'_id': ObjectId('62cfa36d73737e73f8fd4f94'), 'empCode': '100', 'empName': 'Gawri', 'empDesignation': 'CEO'}
```

3) Select an Option3

Search option is selected

Enter the employee code100

```
{'empCode': '100', 'empName': 'Gawri', 'empDesignation': 'CEO'}
```

4) Select an Option4

Delete option is selected

Enter the employee code to Delete100

data deleted successfully

5) Select an Option5

Update option is selected

Enter an employee Code to be updated004

Enter a name to be updatedJeswin

Enter a Designation to be updatedEngineer

Data updated successfully

Select an Option2

View all Employees Option Selected

```
{'_id': ObjectId('62ce675117aab0bdddd09440'), 'empCode': '003', 'empName': 'sheri', 'empDesignation': 'Manager'}
```

```
{'_id': ObjectId('62ce67a4505c8ab5fe901b3c'), 'empCode': '004', 'empName': 'Jeswin', 'empDesignation': 'Engineer'}
```

6) Select an Option6

Enter the employee code to be searched002

{'empName': 'sher1', 'empDesignation': 'Manager'}

{'empName': 'prasanth', 'empDesignation': 'director'}

{'empName': 'Jeswin', 'empDesignation': 'Engineer'}