योगः कर्मसु कौशलम्
IN PURSUIT OF PERFECTION

VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES - TECHNICAL CAMPUS
Grade A++ Accredited Institution by NAAC
NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;
Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE
An ISO 9001:2015 Certified Institution

SCHOOL OF ENGINEERING & TECHNOLOGY

B. Tech Programme: AI-ML (A) (4th Semester)

Course Title: Database Management Systems Lab

Course Code: AIML-254

Submitted To:

Dr. Deepika Bhatia

Submitted By:

Name: Kunsh Sabharwal

Enrolment No: 01117711623

VIPS
योगः कर्मसु कौशलम्
IN PURSUIT OF PERFECTION
VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES - TECHNICAL CAMPUS
Grade A++ Accredited Institution by NAAC
NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;
Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE
An ISO 9001:2015 Certified Institution
SCHOOL OF ENGINEERING & TECHNOLOGY

VISION OF INSTITUTE

To be an educational institute that empowers the field of engineering to build a sustainable future by providing quality education with innovative practices that supports people, planet and profit.

MISSION OF INSTITUTE

To groom the future engineers by providing value-based education and awakening students' curiosity, nurturing creativity and building capabilities to enable them to make significant contributions to the world.

KUNSH SABHARWAL

VIPS
योगः कर्मसु कौशलम्
IN PURSUIT OF PERFECTION

VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES - TECHNICAL CAMPUS
Grade A++ Accredited Institution by NAAC
NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;
Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE
An ISO 9001:2015 Certified Institution
SCHOOL OF ENGINEERING & TECHNOLOGY

INDEX

S. No.	Experiment	Date	Marks			Remarks	Updated Marks	Faculty Signature
			Laboratory Assessment (15 Marks)	Class Participation (5 Marks)	Viva (5 Marks)			
1.	Study and practice various database management systems like MySQL/Oracle/ PostgreSQL/SQL Server and others.							
2.	Implement simple queries of DDL and DML.							
3.	Implement basic queries to Create, Insert, Update, Delete and Select Statements for two different scenarios (For instance: Bank, College etc.)							
4.	Implement queries including various functions- mathematical, string, date etc.							

VIPS
योगः कर्मसु कौशलम्
IN PURSUIT OF PERFECTION
VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES - TECHNICAL CAMPUS
Grade A++ Accredited Institution by NAAC
NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;
Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE
An ISO 9001:2015 Certified Institution
SCHOOL OF ENGINEERING & TECHNOLOGY

5.	Implement queries including Sorting, Grouping and Subqueries- like any, all, exists, not exists.							
6.	Implement queries including various Set operations (Union, Intersection, Except etc.).							
7.	Implement various JOIN operations- (Inner, Outer).							
8.	Write a PL/SQL program using FOR loop to insert ten rows into a database table.							
9.	Given the table EMPLOYEE (Emp No, Name, Salary, Designation, DeptID), write a cursor to select the five highest-paid employees from the table.							

10.	Illustrate how you can embed PL/SQL in a high-level host language such as C/Java And demonstrates how a banking debit transaction migh t be done.							

VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES - TECHNICAL CAMPUS
Grade A++ Accredited Institution by NAAC
NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;
Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE
An ISO 9001:2015 Certified Institution
SCHOOL OF ENGINEERING & TECHNOLOGY

Sr. No.	EXPERIMENT TITLE (Beyond Curriculum)	Date	Faculty Signature
1	Write the steps to install and implement NOSQL databases-MongoDB		
2	Study and implement basic commands of MongoDB		
3	Implement any one real-time project using MySQL/MongoDB such as Library Database Management System etc..		

SECTION 1

Database Management System Lab

GGSIPU

EXPERIMENT 1

Problem statement: Study and practice various database management systems like MySQL/Oracle/PostgreSQL/SQL Server and others.

Theory:

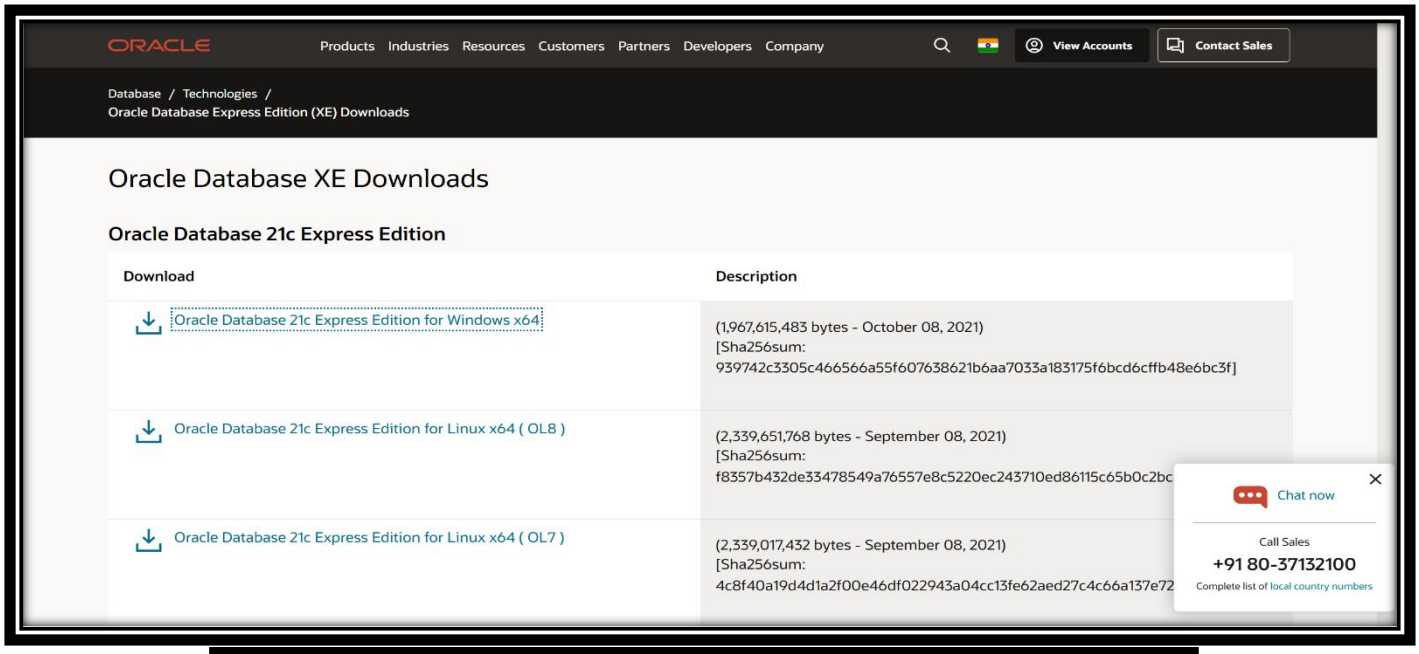

ORACLE®

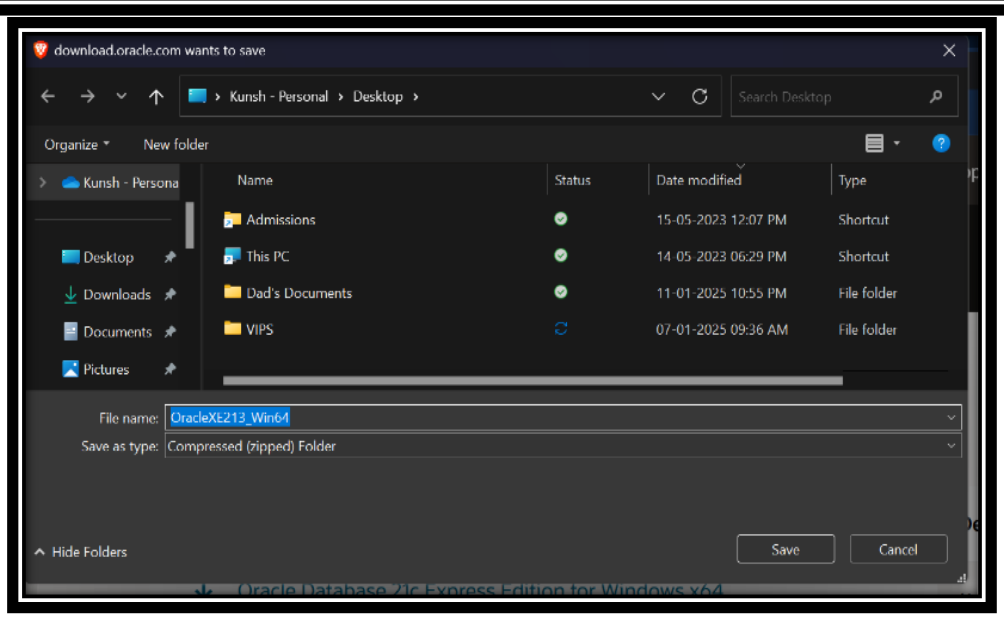

PostgreSQL

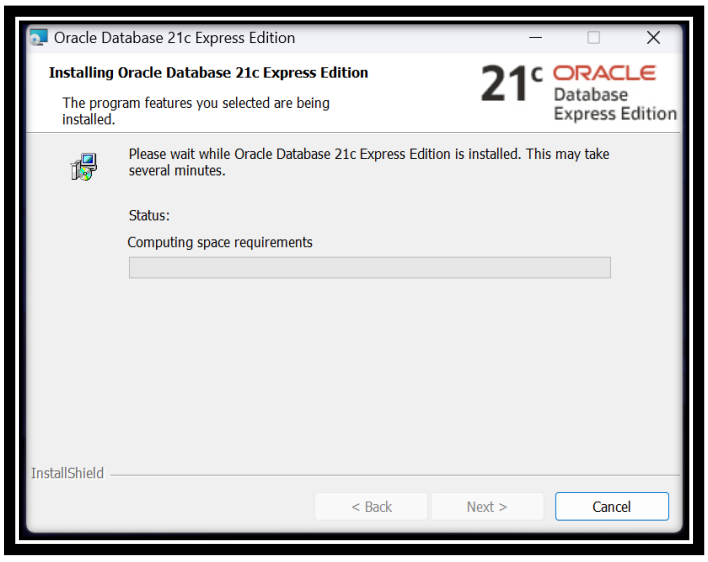

Microsoft®
SQL Server®

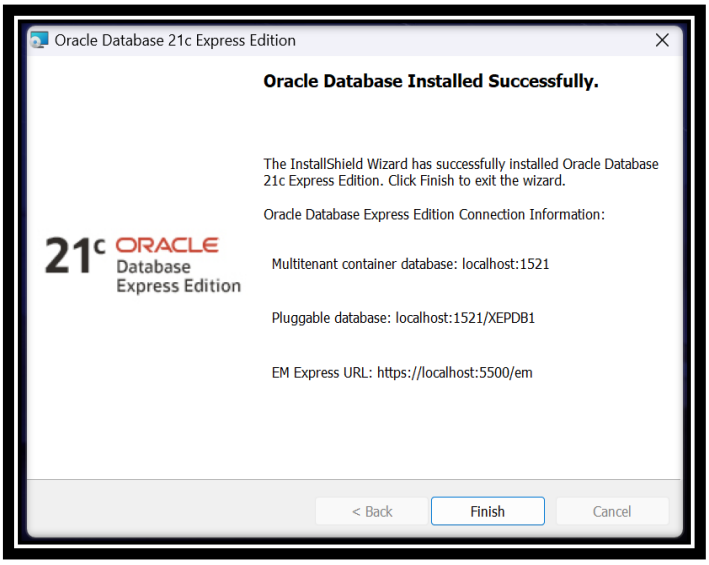

SQL

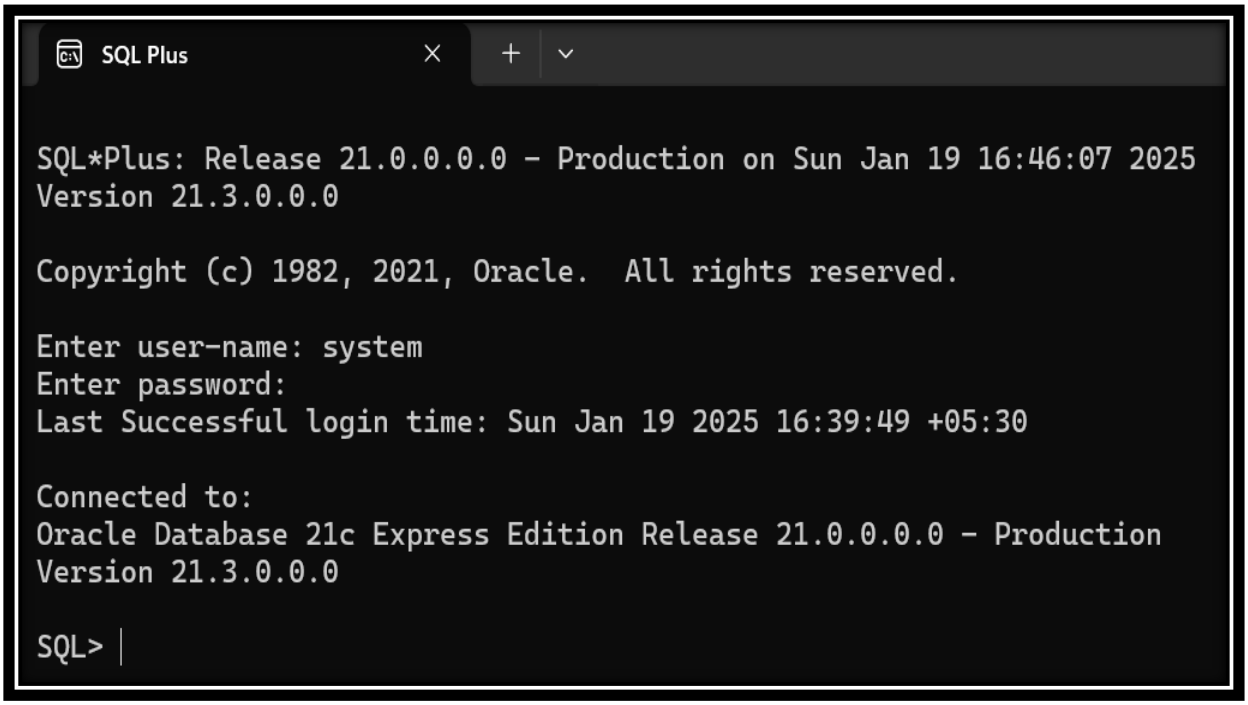
Installation screenshots of SQL:









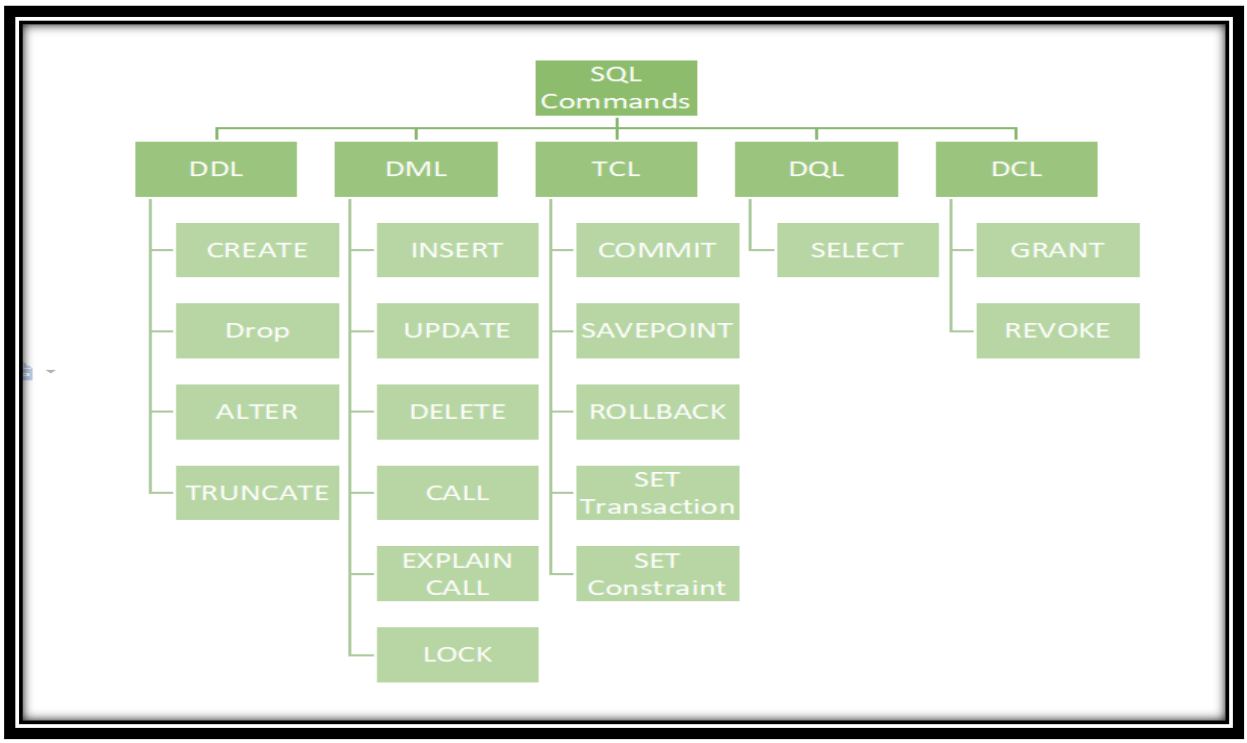


Learning Outcome:

EXPERIMENT 2

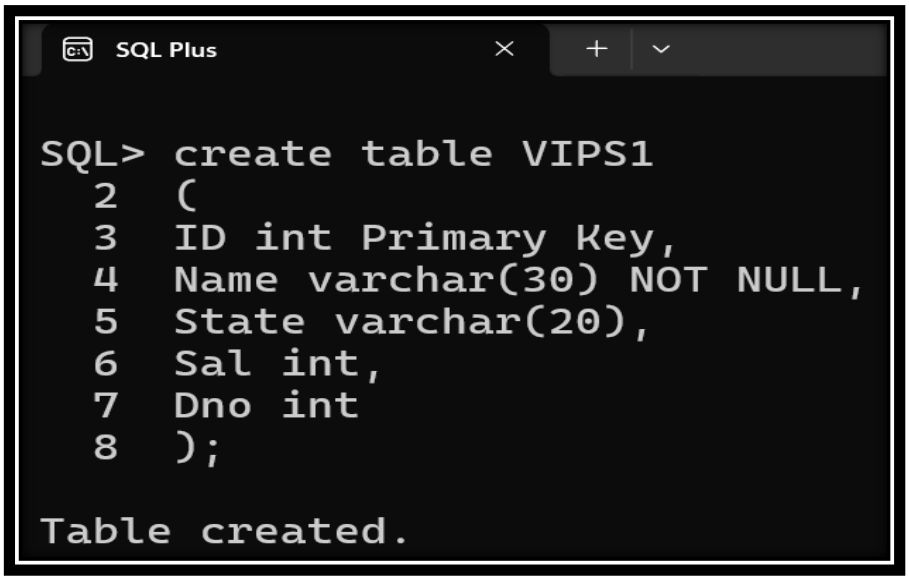
Problem statement: Implement simple queries of DDL and DML.

Theory:

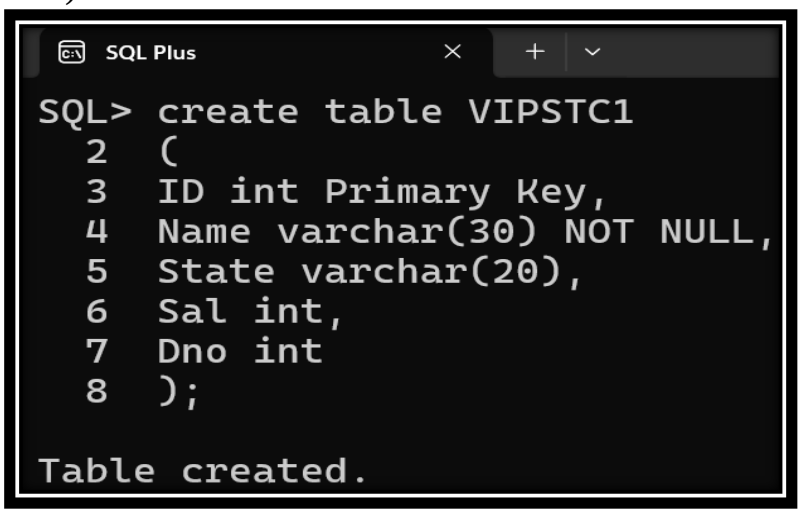


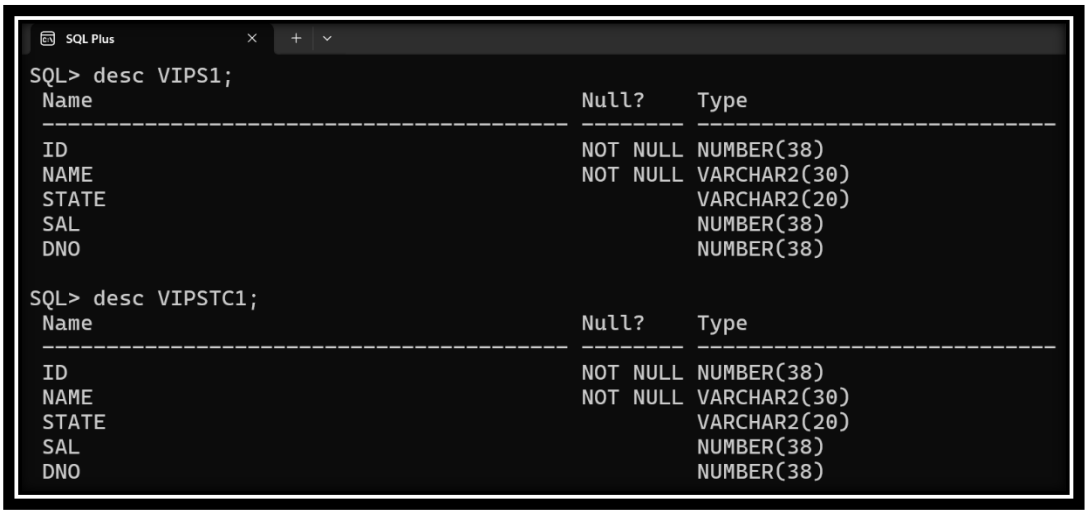
SQL Commands: - (DDL Commands)

1) Create a table VIPS1 (ID, NAME, STATE, SAL, Dno). (Add Constraints)

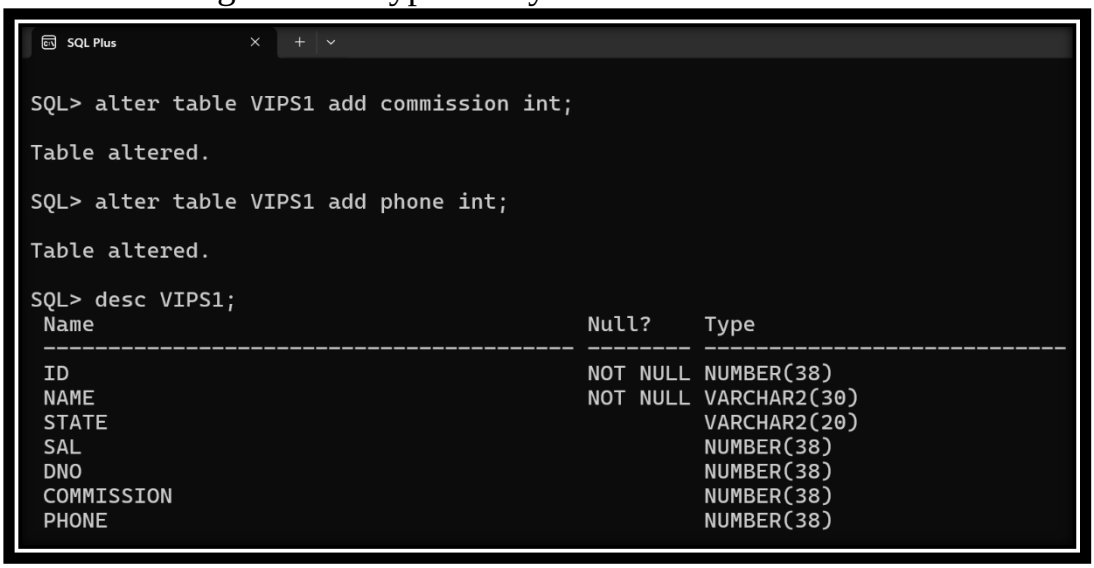


2) Create a table VIPSTC1(ID, NAME, STATE, SAL, Dno). (Add Constraints)





3) Add column Commission and Phone to table.
   a. Drop Column Phone
   b. Change the datatype of any of the column.



```
SQL Plus
SQL> alter table VIPS1 drop column phone;

Table altered.
```

```
SQL Plus
SQL> alter table VIPS1 modify commission varchar(8);

Table altered.

SQL> desc VIPS1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER(38)
 NAME                                      NOT NULL VARCHAR2(30)
 STATE                                              VARCHAR2(20)
 SAL                                                NUMBER(38)
 DNO                                                NUMBER(38)
 COMMISSION                                         VARCHAR2(8)
```

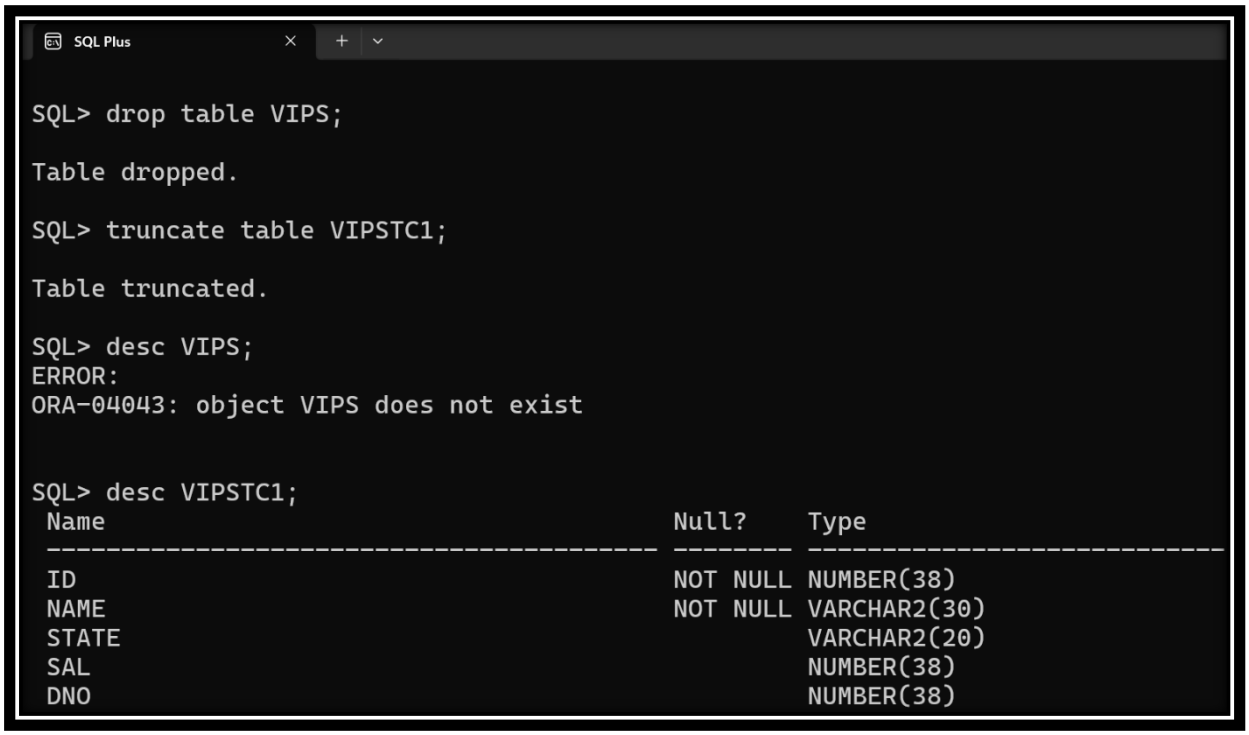
4) Rename one of the columns of the table.

```
SQL Plus
SQL> alter table VIPS1 rename column SAL to SALARY;

Table altered.

SQL> desc VIPS1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER(38)
 NAME                                      NOT NULL VARCHAR2(30)
 STATE                                              VARCHAR2(20)
 SALARY                                             NUMBER(38)
 DNO                                                NUMBER(38)
 COMMISSION                                         VARCHAR2(8)
```

5) Rename the table name.

```
SQL Plus
SQL> alter table VIPS1 rename to VIPS;

Table altered.

SQL> desc VIPS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID                                        NOT NULL NUMBER(38)
 NAME                                      NOT NULL VARCHAR2(30)
 STATE                                              VARCHAR2(20)
 SALARY                                             NUMBER(38)
 DNO                                                NUMBER(38)
 COMMISSION                                         VARCHAR2(8)
```

6) Drop the table VIPS1 and Truncate VIPSTC1 and show the difference.



DML Commands: -

1) Insert Values in both tables. (at least 7 rows)

SQL Plus

```
SQL> insert into vipstc1 values (1,'Kunsh','Delhi',500000,1);

1 row created.

SQL> insert into vipstc1 values (2,'Ram','Delhi',400000,1);

1 row created.

SQL> insert into vipstc1 values (3,'Aman','Maharashtra',400000,2);

1 row created.

SQL> insert into vipstc1 values (4,'Amit','Maharashtra',350000,1);

1 row created.

SQL> insert into vipstc1 values (5,'Deepak','Gujarat',350000,2);

1 row created.

SQL> insert into vipstc1 values (6,'Suman','Gujarat',250000,1);

1 row created.

SQL> insert into vipstc1 values (7,'Harpreet','Punjab',250000,2);

1 row created.

SQL> insert into vipstc1 values (8,'Nisha','Delhi',25000,1);

1 row created.

SQL> insert into vipstc1 values (9,'Rahul','Delhi',100000,2);

1 row created.

SQL> insert into vipstc1 values (10,'Raju','Bihar',150000,1);

1 row created.
```

2) Show the entire tables.

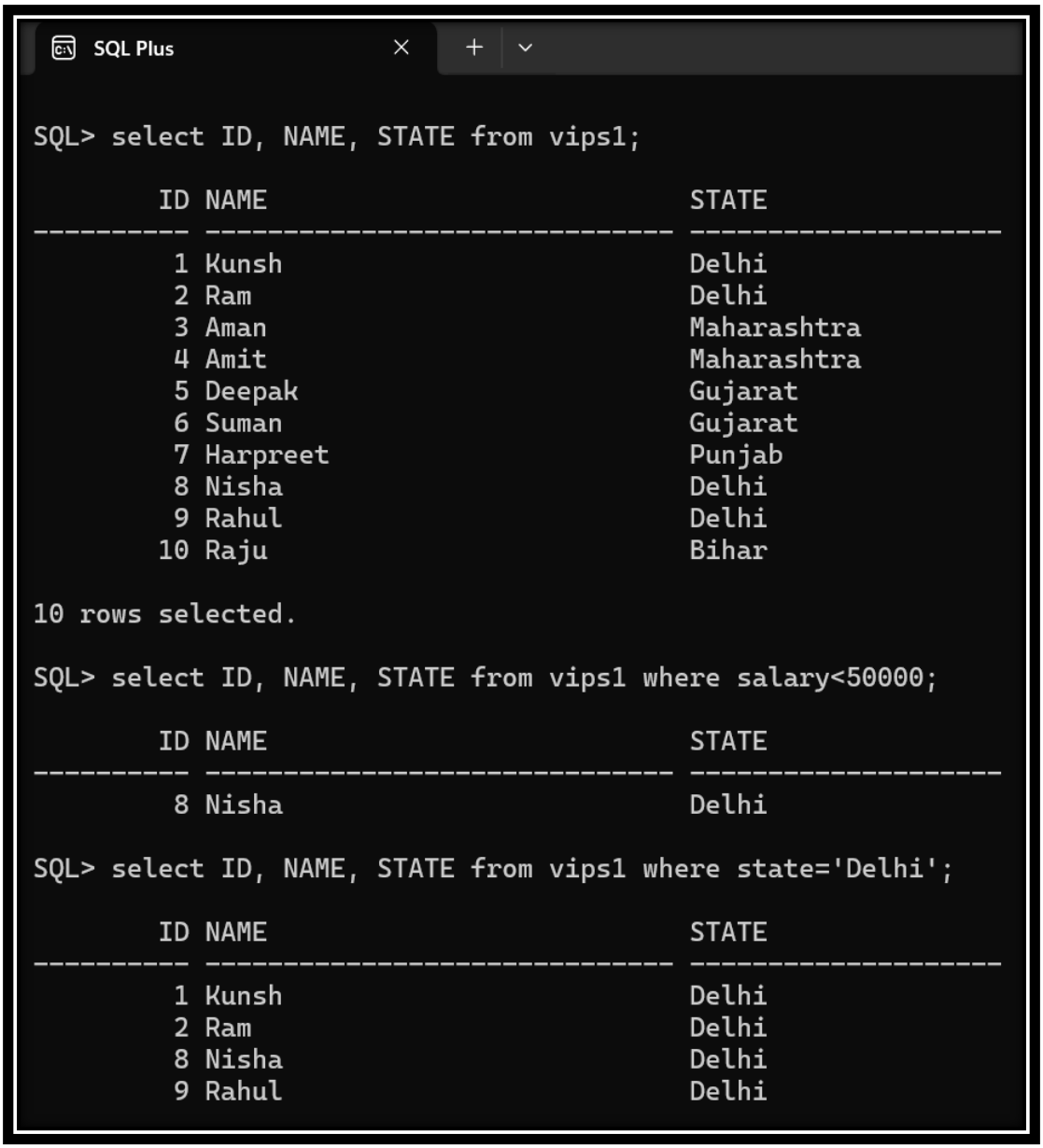
SQL Plus

```
SQL> select * from vips1;

        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         1 Kunsh                          Delhi                500000
         1 CEO

         2 Ram                            Delhi                400000
         1 EVP

         3 Aman                           Maharashtra          400000
         2 EVP


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         4 Amit                           Maharashtra          350000
         1 VP

         5 Deepak                         Gujarat              350000
         2 VP

         6 Suman                          Gujarat              250000
         1 AVP


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         7 Harpreet                       Punjab               250000
         2 AVP

         8 Nisha                          Delhi                 25000
         1 Nurse

         9 Rahul                          Delhi                100000
         2 SM


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
        10 Raju                           Bihar                150000
         1 SM


10 rows selected.
```

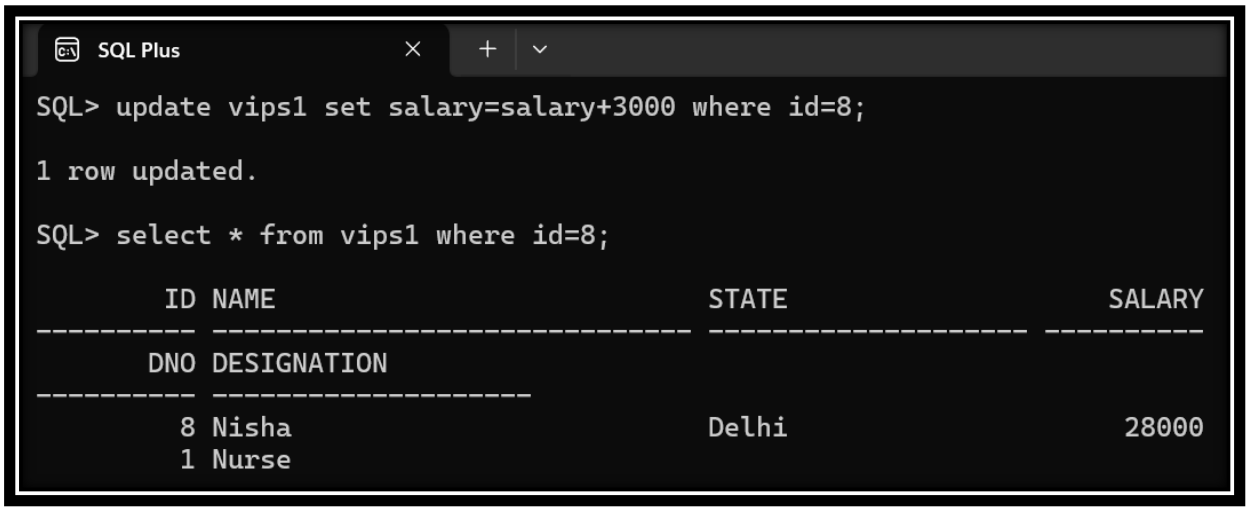
SQL Plus

```
SQL> select * from vipstc1;

        ID NAME                           STATE                    SAL
---------- ------------------------------ -------------------- ----------
       DNO
----------
         1 Kunsh                          Delhi                 500000
         1

         2 Ram                            Delhi                 400000
         1

         3 Aman                           Maharashtra           400000
         2


        ID NAME                           STATE                    SAL
---------- ------------------------------ -------------------- ----------
       DNO
----------
         4 Amit                           Maharashtra           350000
         1

         5 Deepak                         Gujarat               350000
         2

         6 Suman                          Gujarat               250000
         1


        ID NAME                           STATE                    SAL
---------- ------------------------------ -------------------- ----------
       DNO
----------
         7 Harpreet                       Punjab                250000
         2

         8 Nisha                          Delhi                  25000
         1

         9 Rahul                          Delhi                 100000
         2


        ID NAME                           STATE                    SAL
---------- ------------------------------ -------------------- ----------
       DNO
----------
        10 Raju                           Bihar                 150000
         1


10 rows selected.
```

3) Select only 3 columns from the table (With and without where clause)



4) Update the salary of any one of the employees by 3000.

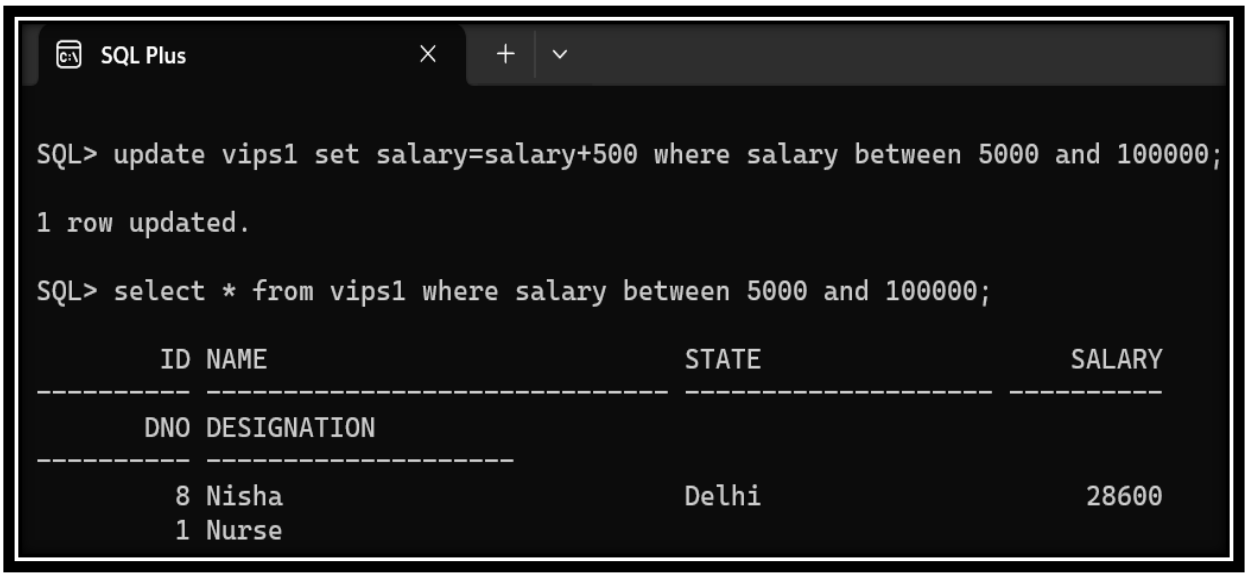


5) Update the salary of each employee by 100.

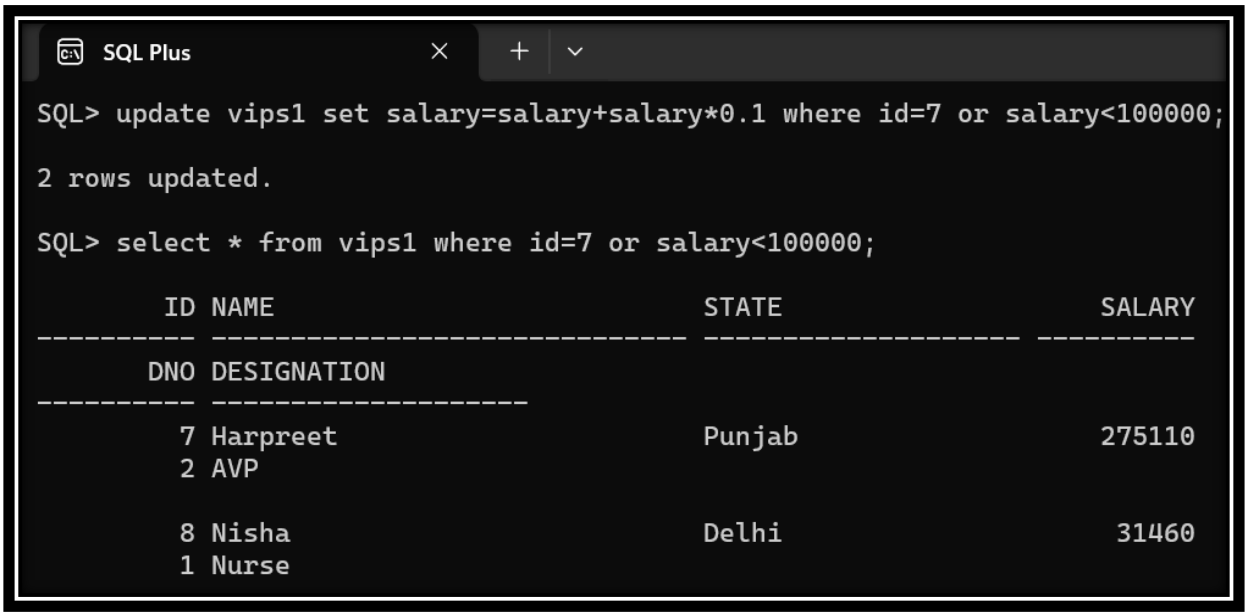
SQL Plus

```
SQL> update vips1 set salary=salary+100;

10 rows updated.

SQL> select * from vips1;

        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         1 Kunsh                          Delhi                500100
         1 CEO

         2 Ram                            Delhi                400100
         1 EVP

         3 Aman                           Maharashtra          400100
         2 EVP


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         4 Amit                           Maharashtra          350100
         1 VP

         5 Deepak                         Gujarat              350100
         2 VP

         6 Suman                          Gujarat              250100
         1 AVP


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         7 Harpreet                       Punjab               250100
         2 AVP

         8 Nisha                          Delhi                 28100
         1 Nurse

         9 Rahul                          Delhi                100100
         2 SM


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
        10 Raju                           Bihar                150100
         1 SM


10 rows selected.
```

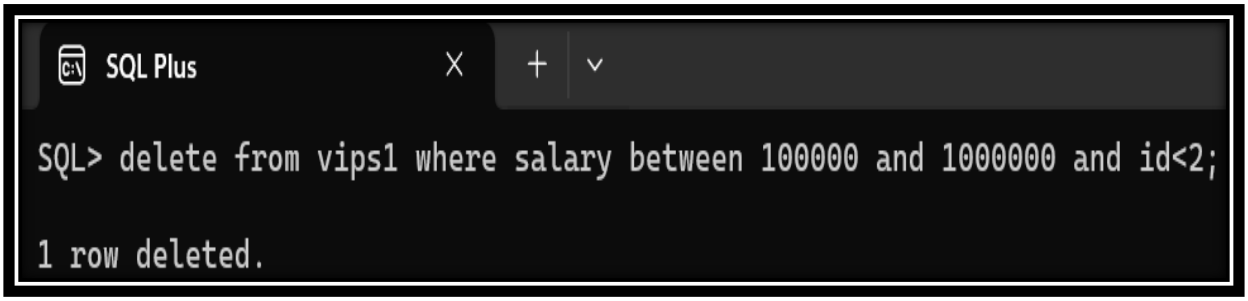
6) Update the salary of the employee by 500 with salary between 5000 and 1,00,000.



7) Update the salary of the employees by 10% whose ID is 7 or salary is less than 1,00,000.



8) Delete the employee with salary between 1,00,000 and 1,00,00,000 and ID less than 2.



SQL Plus

```
SQL> select * from vips1;

        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         2 Ram                            Delhi                400100
         1 EVP

         3 Aman                           Maharashtra          400100
         2 EVP

         4 Amit                           Maharashtra          350100
         1 VP


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         5 Deepak                         Gujarat              350100
         2 VP

         6 Suman                          Gujarat              250100
         1 AVP

         7 Harpreet                       Punjab               275110
         2 AVP


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         8 Nisha                          Delhi                 31460
         1 Nurse

         9 Rahul                          Delhi                100100
         2 SM

        10 Raju                           Bihar                150100
         1 SM


9 rows selected.
```

9) Delete the employee whose designation is Nurse.

SQL Plus

```
SQL> delete from vips1 where designation='Nurse';

1 row deleted.

SQL> select * from vips1;

        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         2 Ram                            Delhi                400100
         1 EVP

         3 Aman                           Maharashtra          400100
         2 EVP

         4 Amit                           Maharashtra          350100
         1 VP


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         5 Deepak                         Gujarat              350100
         2 VP

         6 Suman                          Gujarat              250100
         1 AVP

         7 Harpreet                       Punjab               275110
         2 AVP


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         9 Rahul                          Delhi                100100
         2 SM

        10 Raju                           Bihar                150100
         1 SM


8 rows selected.
```

10) Delete the employee with no salary.

```
SQL Plus

SQL> update vips1 set salary=NULL where id=5;

1 row updated.
```

```
SQL Plus

SQL> delete from vips1 where salary is NULL;

1 row deleted.

SQL> select * from vips1;

        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         2 Ram                            Delhi                400100
         1 EVP

         3 Aman                           Maharashtra          400100
         2 EVP

         4 Amit                           Maharashtra          350100
         1 VP


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
         6 Suman                          Gujarat              250100
         1 AVP

         7 Harpreet                       Punjab               275110
         2 AVP

         9 Rahul                          Delhi                100100
         2 SM


        ID NAME                           STATE                SALARY
---------- ------------------------------ -------------------- ----------
       DNO DESIGNATION
---------- --------------------
        10 Raju                           Bihar                150100
         1 SM


7 rows selected.
```

Learning Outcome:

EXPERIMENT 3

Problem statement: Implement basic queries to Create, Insert, Update, Delete and Select Statements for two different scenarios (For instance: Bank, College etc.)

Theory:

SQL Commands: - (For Bank Scenario)

1) Create the 5 tables given below: -
    a. Acc (acc_no, name, phone)
    b. Trans (*acc_no*, C_D, amount)
    c. Loan (*acc_no*, interest, time, type)
    d. FD (*acc_no*, amount, interest, time)
    e. Locker (*acc_no*, l_no)

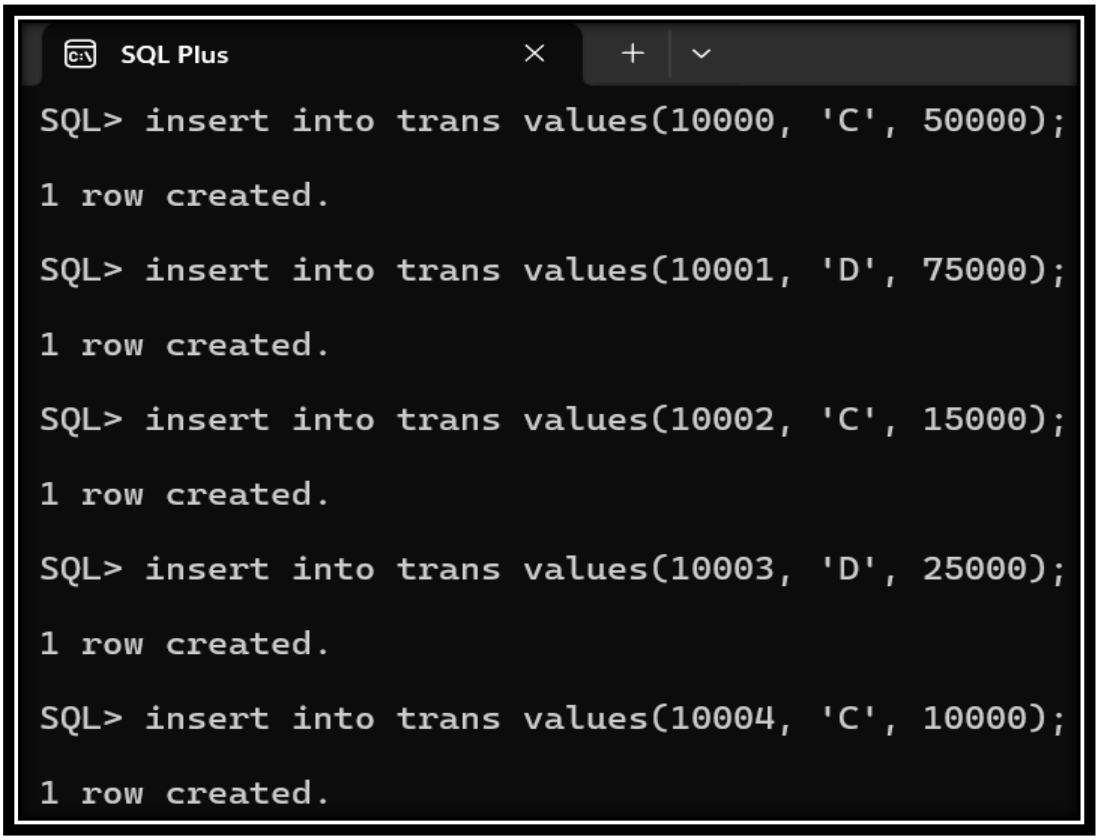
SQL Plus

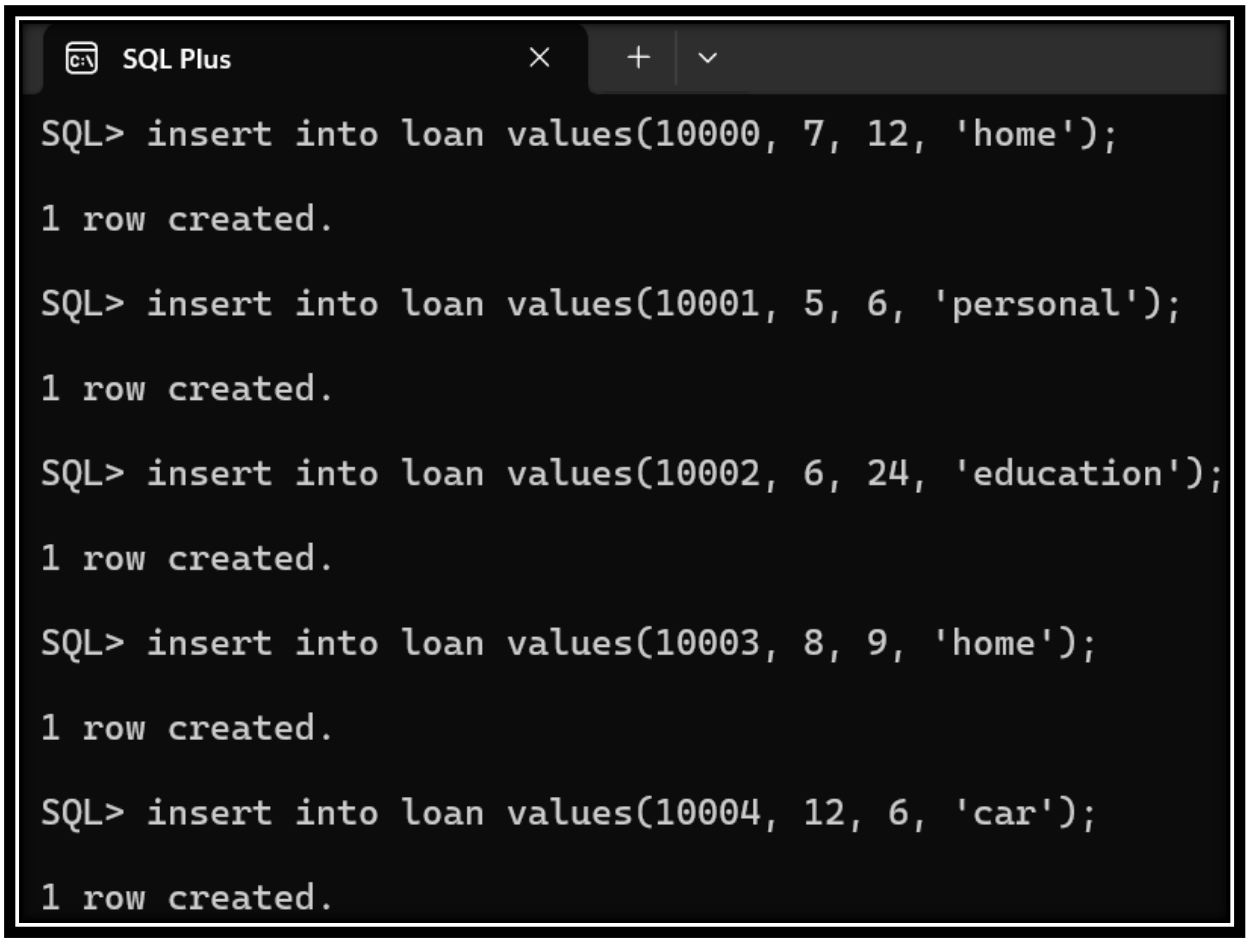
```
SQL> create table acc
  2  (
  3  acc_no int Primary Key,
  4  name varchar(10),
  5  phone int
  6  );

Table created.

SQL> create table trans
  2  (
  3  acc_no int,foreign key(acc_no) references acc(acc_no),
  4  c_d varchar(3),
  5  amount int
  6  );

Table created.

SQL>
SQL> create table loan
  2  (
  3  acc_no int,foreign key(acc_no) references acc(acc_no),
  4  interest int,
  5  time_month int,
  6  type varchar(10)
  7  );

Table created.

SQL> create table fd
  2  (
  3  acc_no int,foreign key(acc_no) references acc(acc_no),
  4  amount int,
  5  interest int,
  6  time_month int
  7  );

Table created.

SQL> create table locker
  2  (
  3  acc_no int, foreign key(acc_no) references acc(acc_no),
  4  l_no int
  5  );

Table created.
```

```
SQL Plus

SQL> desc acc;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACC_NO                                    NOT NULL NUMBER(38)
 NAME                                               VARCHAR2(10)
 PHONE                                              NUMBER(38)

SQL> desc trans;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACC_NO                                             NUMBER(38)
 C_D                                                VARCHAR2(3)
 AMOUNT                                             NUMBER(38)

SQL> desc loan;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACC_NO                                             NUMBER(38)
 INTEREST                                           NUMBER(38)
 TIME_MONTH                                         NUMBER(38)
 TYPE                                               VARCHAR2(10)

SQL> desc fd;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACC_NO                                             NUMBER(38)
 AMOUNT                                             NUMBER(38)
 INTEREST                                           NUMBER(38)
 TIME_MONTH                                         NUMBER(38)

SQL> desc locker;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACC_NO                                             NUMBER(38)
 L_NO                                               NUMBER(38)
```

2) Insert values in each table and show contents.

```
SQL Plus

SQL> insert into acc values(10000, 'Sumit', 9999999999);

1 row created.

SQL> insert into acc values(10001, 'Raman', 8888888888);

1 row created.

SQL> insert into acc values(10002, 'Sita', 7777766666);

1 row created.

SQL> insert into acc values(10003, 'Kunsh', 5555555555);

1 row created.

SQL> insert into acc values(10004, 'Ketan',4567891230);

1 row created.
```


SQL Plus
SQL> insert into trans values(10000, 'C', 50000);
1 row created.
SQL> insert into trans values(10001, 'D', 75000);
1 row created.
SQL> insert into trans values(10002, 'C', 15000);
1 row created.
SQL> insert into trans values(10003, 'D', 25000);
1 row created.
SQL> insert into trans values(10004, 'C', 10000);
1 row created.


SQL Plus
SQL> insert into loan values(10000, 7, 12, 'home');
1 row created.
SQL> insert into loan values(10001, 5, 6, 'personal');
1 row created.
SQL> insert into loan values(10002, 6, 24, 'education');
1 row created.
SQL> insert into loan values(10003, 8, 9, 'home');
1 row created.
SQL> insert into loan values(10004, 12, 6, 'car');
1 row created.

SQL Plus

```
SQL> insert into fd values(10000, 12000, 6, 12);

1 row created.

SQL> insert into fd values(10001, 24000, 4, 6);

1 row created.

SQL> insert into fd values(10002, 30000, 5, 9);

1 row created.

SQL> insert into fd values(10003, 60000, 8, 18);

1 row created.

SQL> insert into fd values(10004, 50000, 6, 12);

1 row created.
```

SQL Plus

```
SQL> insert into locker values(10000, 100);

1 row created.

SQL> insert into locker values(10001, 101);

1 row created.

SQL> insert into locker values(10002, 102);

1 row created.

SQL> insert into locker values(10003, 103);

1 row created.

SQL> insert into locker values(10004, 104);

1 row created.
```

```
SQL> select * from acc;

    ACC_NO NAME            PHONE
---------- ---------- ----------
     10000 Sumit      9999999999
     10001 Raman      8888888888
     10002 Sita       7777766666
     10003 Kunsh      5555555555
     10004 Ketan      4567891230
```

```
SQL> select * from trans;

    ACC_NO C_D      AMOUNT
---------- --- ----------
     10000 C          50000
     10001 D          75000
     10002 C          15000
     10003 D          25000
     10004 C          10000
```

```
SQL> select * from loan;

    ACC_NO   INTEREST TIME_MONTH TYPE
---------- ---------- ---------- ----------
     10000          7         12 home
     10001          5          6 personal
     10002          6         24 education
     10003          8          9 home
     10004         12          6 car
```

```
SQL> select * from fd;

    ACC_NO     AMOUNT   INTEREST TIME_MONTH
---------- ---------- ---------- ----------
     10000      12000          6         12
     10001      24000          4          6
     10002      30000          5          9
     10003      60000          8         18
     10004      50000          6         12
```

```
SQL> select * from locker;

    ACC_NO       L_NO
---------- ----------
     10000        100
     10001        101
     10002        102
     10003        103
     10004        104
```

3) Add a column loan_no in loan table and update values for this column.

```
SQL> alter table loan add loan_no int;

Table altered.

SQL> desc loan;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACC_NO                                             NUMBER(38)
 INTEREST                                           NUMBER(38)
 TIME_MONTH                                         NUMBER(38)
 TYPE                                               VARCHAR2(10)
 LOAN_NO                                            NUMBER(38)
```

```
SQL> update loan set loan_no=1 where acc_no=10000;

1 row updated.

SQL> update loan set loan_no=2 where acc_no=10001;

1 row updated.

SQL> update loan set loan_no=3 where acc_no=10002;

1 row updated.

SQL> update loan set loan_no=4 where acc_no=10003;

1 row updated.

SQL> update loan set loan_no=5 where acc_no=10004;

1 row updated.

SQL> select * from loan;

    ACC_NO   INTEREST TIME_MONTH TYPE          LOAN_NO
---------- ---------- ---------- ---------- ----------
     10000          7         12 home                1
     10001          5          6 personal            2
     10002          6         24 education           3
     10003          8          9 home                4
     10004         12          6 car                 5
```

4) Insert a row in account table without phone number.

```
SQL> insert into acc values(10005,'Raju',null);

1 row created.

SQL> select * from acc;

    ACC_NO NAME           PHONE
---------- ---------- ----------
     10000 Sumit      9999999999
     10001 Raman      8888888888
     10002 Sita       7777766666
     10003 Kunsh      5555555555
     10004 Ketan      4567891230
     10005 Raju

6 rows selected.
```

5) Delete account with no phone number.

```
SQL> delete from acc where phone is null;

1 row deleted.

SQL> select * from acc;

    ACC_NO NAME            PHONE
---------- ---------- ----------
     10000 Sumit      9999999999
     10001 Raman      8888888888
     10002 Sita       7777766666
     10003 Kunsh      5555555555
     10004 Ketan      4567891230
```

SQL Commands: - (For College Scenario)

1) Create the below mentioned tables with proper constraints: -
   a. Faculty (<u>fid</u>, name, *dno*, sal, address, phone, dob, exp)
   b. Dept (<u>dno</u>, dname, budget)
   c. Student (<u>sid</u>, name, *dno*, phone, dob, address, sem)
   d. Society (<u>sc_id</u>, name)
   e. Std_soc (*sid*, *sc_id*)
   f. Course (<u>cid</u>, cname)
   g. Fac_course (*fid*, *cid*)

SQL Plus

```
SQL> create table Dept
  2  (
  3  dno int Primary Key,
  4  dname varchar(10),
  5  budget int
  6  );

Table created.

SQL> create table Faculty
  2  (
  3  fid int Primary Key,
  4  name varchar(10),
  5  dno int,foreign key(dno) references Dept(dno),
  6  sal int,
  7  address varchar(15),
  8  phone int,
  9  dob date,
 10  exp int
 11  );

Table created.
```

SQL Plus

```
SQL> create table student
  2  (
  3  sid int Primary Key,
  4  name varchar(10),
  5  dno int,foreign key(dno) references Dept(dno),
  6  phone int,
  7  dob date,
  8  address varchar(15),
  9  sem int
 10  );

Table created.

SQL> create table society
  2  (
  3  sc_id int Primary Key,
  4  name varchar(10)
  5  );

Table created.

SQL> create table std_soc
  2  (
  3  sid int,foreign key(sid) references Student(sid),
  4  sc_id int,foreign key(sc_id) references Society(sc_id)
  5  );

Table created.

SQL> create table course
  2  (
  3  cid int Primary Key,
  4  cname varchar(10)
  5  );

Table created.

SQL> create table Fac_course
  2  (
  3  fid int,foreign key(fid) references Faculty(fid),
  4  cid int,foreign key(cid) references Course(cid)
  5  );

Table created.
```

SQL Plus

```
SQL> desc dept;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DNO                                       NOT NULL NUMBER(38)
 DNAME                                              VARCHAR2(10)
 BUDGET                                             NUMBER(38)

SQL> desc faculty;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 FID                                       NOT NULL NUMBER(38)
 NAME                                               VARCHAR2(10)
 DNO                                                NUMBER(38)
 SAL                                                NUMBER(38)
 ADDRESS                                            VARCHAR2(15)
 PHONE                                              NUMBER(38)
 DOB                                                DATE
 EXP                                                NUMBER(38)

SQL> desc student;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL NUMBER(38)
 NAME                                               VARCHAR2(10)
 DNO                                                NUMBER(38)
 PHONE                                              NUMBER(38)
 DOB                                                DATE
 ADDRESS                                            VARCHAR2(15)
 SEM                                                NUMBER(38)

SQL> desc society;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SC_ID                                     NOT NULL NUMBER(38)
 NAME                                               VARCHAR2(10)

SQL> desc std_soc;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                                NUMBER(38)
 SC_ID                                              NUMBER(38)

SQL> desc course;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CID                                       NOT NULL NUMBER(38)
 CNAME                                              VARCHAR2(10)

SQL> desc fac_course;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 FID                                                NUMBER(38)
 CID                                                NUMBER(38)
```

2) Insert rows in each table (at least 5 records).

```
SQL> insert into dept values(1, 'English', 500000);

1 row created.

SQL> insert into dept values(2, 'CSE', 1000000);

1 row created.

SQL> insert into dept values(3, 'AIML', 800000);

1 row created.

SQL> insert into dept values(4, 'IIOT', 600000);

1 row created.

SQL> insert into dept values(5, 'Maths', 800000);

1 row created.

SQL> select * from dept;

       DNO DNAME           BUDGET
---------- ---------- ----------
         1 English         500000
         2 CSE            1000000
         3 AIML            800000
         4 IIOT            600000
         5 Maths           800000
```

SQL Plus

```
SQL> insert into faculty values(1, 'Aditi', 1, 40000, 'Rohini', 9999999999, '4-July-1995', 5);

1 row created.

SQL> insert into faculty values(2, 'Akshay', 2, 50000, 'Noida', 8888888888, '13-June-1994', 6);

1 row created.

SQL> insert into faculty values(3, 'Aman', 3, 60000, 'Dwarka', 7777777777, '2-October-1995', 5);

1 row created.

SQL> insert into faculty values(4, 'Salman', 4, 45000, 'Janakpuri', 6666666666, '1-April-1994', 6);

1 row created.

SQL> insert into faculty values(5, 'Poonam', 5, 50000, 'Rohini', 5555555555, '5-July-1996', 4);

1 row created.

SQL> select * from faculty;

       FID NAME             DNO        SAL ADDRESS             PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP
----------
         1 Aditi              1      40000 Rohini          9999999999 04-JUL-95
         5

         2 Akshay             2      50000 Noida           8888888888 13-JUN-94
         6

         3 Aman               3      60000 Dwarka          7777777777 02-OCT-95
         5


       FID NAME             DNO        SAL ADDRESS             PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP
----------
         4 Salman             4      45000 Janakpuri       6666666666 01-APR-94
         6

         5 Poonam             5      50000 Rohini          5555555555 05-JUL-96
         4
```

```
SQL> insert into student values(1, 'Kunsh', 3, 9990159801, '5-July-2005', 'Subhash Nagar', 4);

1 row created.

SQL> insert into student values(2, 'Raju', 1, 9990159352, '15-August-2004', 'Rohini', 6);

1 row created.

SQL> insert into student values(3, 'Geeta', 2, 9990159278, '26-January-2005', 'Pitampura', 6);

1 row created.

SQL> insert into student values(4, 'Akash', 5, 9990159972, '24-March-2004', 'Dwarka', 4);

1 row created.

SQL> insert into student values(5, 'Rachit', 4, 9990159292, '13-April-2006', 'Noida', 2);

1 row created.

SQL> select * from student;

       SID NAME              DNO      PHONE DOB       ADDRESS               SEM
---------- ---------- ---------- ---------- --------- --------------- ----------
         1 Kunsh               3 9990159801 05-JUL-05 Subhash Nagar            4
         2 Raju                1 9990159352 15-AUG-04 Rohini                   6
         3 Geeta               2 9990159278 26-JAN-05 Pitampura                6
         4 Akash               5 9990159972 24-MAR-04 Dwarka                   4
         5 Rachit              4 9990159292 13-APR-06 Noida                    2
```

```
SQL> insert into society values(1, 'Art');

1 row created.

SQL> insert into society values(2, 'Singing');

1 row created.

SQL> insert into society values(3, 'Dancing');

1 row created.

SQL> insert into society values(4, 'Debating');

1 row created.

SQL> insert into society values(5, 'Gaming');

1 row created.

SQL> select * from society;

     SC_ID NAME
---------- ----------
         1 Art
         2 Singing
         3 Dancing
         4 Debating
         5 Gaming
```

```
SQL> insert into std_soc values(1,1);

1 row created.

SQL> insert into std_soc values(2,2);

1 row created.

SQL> insert into std_soc values(3,3);

1 row created.

SQL> insert into std_soc values(4,4);

1 row created.

SQL> insert into std_soc values(5,5);

1 row created.

SQL> select * from std_soc;

       SID       SC_ID
---------- ----------
         1          1
         2          2
         3          3
         4          4
         5          5
```

```
SQL> insert into course values(1, 'B Tech');

1 row created.

SQL> insert into course values(2, 'BJMC');

1 row created.

SQL> insert into course values(3, 'BBA');

1 row created.

SQL> insert into course values(4, 'BALLB');

1 row created.

SQL> insert into course values(5, 'B Com');

1 row created.

SQL> select * from course;

       CID CNAME
---------- ----------
         1 B Tech
         2 BJMC
         3 BBA
         4 BALLB
         5 B Com
```

```
SQL> insert into fac_course values(1,1);

1 row created.

SQL> insert into fac_course values(2,2);

1 row created.

SQL> insert into fac_course values(3,3);

1 row created.

SQL> insert into fac_course values(4,4);

1 row created.

SQL> insert into fac_course values(5,5);

1 row created.

SQL> select * from fac_course;

       FID        CID
---------- ----------
         1          1
         2          2
         3          3
         4          4
         5          5
```

3) Add a column no_of_awards to faculty table and update the values in the column

```
SQL> alter table faculty add no_of_awards int;

Table altered.

SQL> update faculty set no_of_awards=5 where fid=1;

1 row updated.

SQL> update faculty set no_of_awards=3 where fid=2;

1 row updated.

SQL> update faculty set no_of_awards=4 where fid=3;

1 row updated.

SQL> update faculty set no_of_awards=6 where fid=4;

1 row updated.

SQL> update faculty set no_of_awards=2 where fid=5;

1 row updated.
```

```
SQL> select * from faculty;

       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP NO_OF_AWARDS
---------- ------------
         1 Aditi                1      40000 Rohini          9999999999 04-JUL-95
         5             5

         2 Akshay               2      50000 Noida           8888888888 13-JUN-94
         6             3

         3 Aman                 3      60000 Dwarka          7777777777 02-OCT-95
         5             4


       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP NO_OF_AWARDS
---------- ------------
         4 Salman               4      45000 Janakpuri       6666666666 01-APR-94
         6             6

         5 Poonam               5      50000 Rohini          5555555555 05-JUL-96
         4             2
```

4) Retrieve the faculties with exp greater than 10 years and sal<60000.

```
SQL> insert into faculty values(6, 'Puneet', 5, 40000, 'Rohini', 44444444444, '18-September-1986', 14, 8);

1 row created.

SQL> insert into faculty values(7, 'Mona', 4, 55000, 'Pitampura', 3333333333, '21-December-1987', 13, 7);

1 row created.

SQL> select * from faculty where exp>10 and sal<60000;

       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP NO_OF_AWARDS
---------- ------------
         6 Puneet               5      40000 Rohini          44444444444 18-SEP-86
        14             8

         7 Mona                 4      55000 Pitampura       3333333333 21-DEC-87
        13             7
```

5) Retrieve the students enrolled in dno 2 or sem 4.

```
SQL> select * from student where dno=2 or sem=4;

       SID NAME              DNO      PHONE DOB       ADDRESS                SEM
---------- ---------- ---------- ---------- --------- --------------- ----------
         1 Kunsh                3 9990159801 05-JUL-05 Subhash Nagar            4
         3 Geeta                2 9990159278 26-JAN-05 Pitampura                6
         4 Akash                5 9990159972 24-MAR-04 Dwarka                   4
```

6) Increase the faculties salary whose dno is 1.

```
SQL> update faculty set sal=sal+10000 where dno=1;

1 row updated.

SQL> select * from faculty where dno=1;

       FID NAME             DNO        SAL ADDRESS             PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP NO_OF_AWARDS
---------- ------------
         1 Aditi               1      50000 Rohini          9999999999 04-JUL-95
         5            5
```

7) Increase the budget of AIML Department by 10000

```
SQL> update dept set budget=budget+10000 where dname='AIML';

1 row updated.

SQL> select * from dept where dname='AIML';

       DNO DNAME          BUDGET
---------- ---------- ----------
         3 AIML           810000
```

8) Update the phone no. of student id 21.

```
SQL> insert into student values(21, 'Rohan', 4, 9990100000, '15-May-2006', 'Moti Nagar', 2);

1 row created.

SQL> select * from student where sid=21;

       SID NAME             DNO      PHONE DOB       ADDRESS              SEM
---------- ---------- ---------- ---------- --------- --------------- ----------
        21 Rohan               4 9990100000 15-MAY-06 Moti Nagar              2

SQL> update student set phone=9990159555 where sid=21;

1 row updated.

SQL> select * from student where sid=21;

       SID NAME             DNO      PHONE DOB       ADDRESS              SEM
---------- ---------- ---------- ---------- --------- --------------- ----------
        21 Rohan               4 9990159555 15-MAY-06 Moti Nagar              2
```

9) Delete the society CODE (see if it allows (provided there are students enrolled))

```
SQL> update society set name='CODE' where sc_id=3;

1 row updated.

SQL> select * from society;

     SC_ID NAME
---------- ----------
         1 Art
         2 Singing
         3 CODE
         4 Debating
         5 Gaming

SQL> delete from society where name='CODE';
delete from society where name='CODE'
*
ERROR at line 1:
ORA-02292: integrity constraint (SYSTEM.SYS_C008336) violated - child record
found
```

10) Delete the student with no phone no.

```
SQL> insert into student values(15, 'Rahul', 4, null, '20-August-2005', 'Moti Bagh', 4);

1 row created.

SQL> select * from student where sid=15;

       SID NAME               DNO      PHONE DOB       ADDRESS                SEM
---------- ---------- ---------- ---------- --------- --------------- ----------
        15 Rahul                4            20-AUG-05 Moti Bagh                4

SQL> delete from student where phone is null;

1 row deleted.

SQL> select * from student;

       SID NAME               DNO      PHONE DOB       ADDRESS                SEM
---------- ---------- ---------- ---------- --------- --------------- ----------
         1 Kunsh                3 9990159801 05-JUL-05 Subhash Nagar            4
         2 Raju                 1 9990159352 15-AUG-04 Rohini                   6
         3 Geeta                2 9990159278 26-JAN-05 Pitampura                6
         4 Akash                5 9990159972 24-MAR-04 Dwarka                   4
         5 Rachit               4 9990159292 13-APR-06 Noida                    2
        21 Rohan                4 9990159555 15-MAY-06 Moti Nagar               2

6 rows selected.
```

Learning Outcome:

EXPERIMENT 4

Problem statement: Implement queries including various functions- mathematical, string, date etc.

Theory:

KUNSH SABHARWAL

SQL Commands:

1) Mathematical Functions:

a. SQRT()
b. SIN() / COS()/ TAN()
c. ABS()
d. CEIL()
e. FLOOR()
f. ROUND()

SQL Plus

```
SQL> SELECT SQRT(100) "Square root" FROM DUAL;

Square root
-----------
         10

SQL> SELECT SIN(30 * 3.14159265359/180) "Sine of 30 degrees" FROM DUAL;

Sine of 30 degrees
------------------
                .5

SQL> SELECT COS(60 * 3.14159265359/180) "Cos of 60 degrees" FROM DUAL;

Cos of 60 degrees
-----------------
               .5

SQL> SELECT TAN(45 * 3.14159265359/180) "Tan of 45 degrees" FROM DUAL;

Tan of 45 degrees
-----------------
                1

SQL> SELECT ABS(-28) "Absolute" FROM DUAL;

  Absolute
----------
        28

SQL> SELECT CEIL(9.4) from dual;

 CEIL(9.4)
----------
        10

SQL> SELECT floor(9.4) from dual;

FLOOR(9.4)
----------
         9

SQL> SELECT ROUND(15.193,1) "Round" FROM DUAL;

     Round
----------
      15.2

SQL> SELECT ROUND(15.193,2) "Round" FROM DUAL;

     Round
----------
     15.19
```

2) String Functions:

a. ASCII()
b. CHR()
c. LENGTH()
d. CONCAT()
e. LOWER() / UPPER()
f. SUBSTR()

SQL Plus

```
SQL> SELECT ASCII('A') "ASCII 'A'" FROM DUAL;

 ASCII 'A'
----------
        65

SQL> SELECT ASCII('a') "ASCII 'a'" FROM DUAL;

 ASCII 'a'
----------
        97

SQL> SELECT CHR(98) FROM DUAL;

C
-
b

SQL> SELECT CHR(66) FROM DUAL;

C
-
B

SQL> SELECT LENGTH('KUNSH') "Length in characters" FROM DUAL;

Length in characters
--------------------
                   5

SQL> SELECT LENGTH('   KUNSH   ') "Length in characters" FROM DUAL;

Length in characters
--------------------
                  11

SQL> select concat('Kunsh','AIML - A') from dual;

CONCAT('KUNSH
-------------
KunshAIML - A

SQL> select lower('KUNSH') from dual;

LOWER
-----
kunsh

SQL> select upper('kunsh') from dual;

UPPER
-----
KUNSH

SQL> select substr('ABCDEF',3,4) from dual;

SUBS
----
CDEF
```

3) Date Functions:

a. CURRENT_DATE()
b. CURRENT_TIMESTAMP()
c. MONTHS_BETWEEN()
d. SYSDATE()
e. LAST_DAY()
f. NEXT_DAY()

SQL Plus

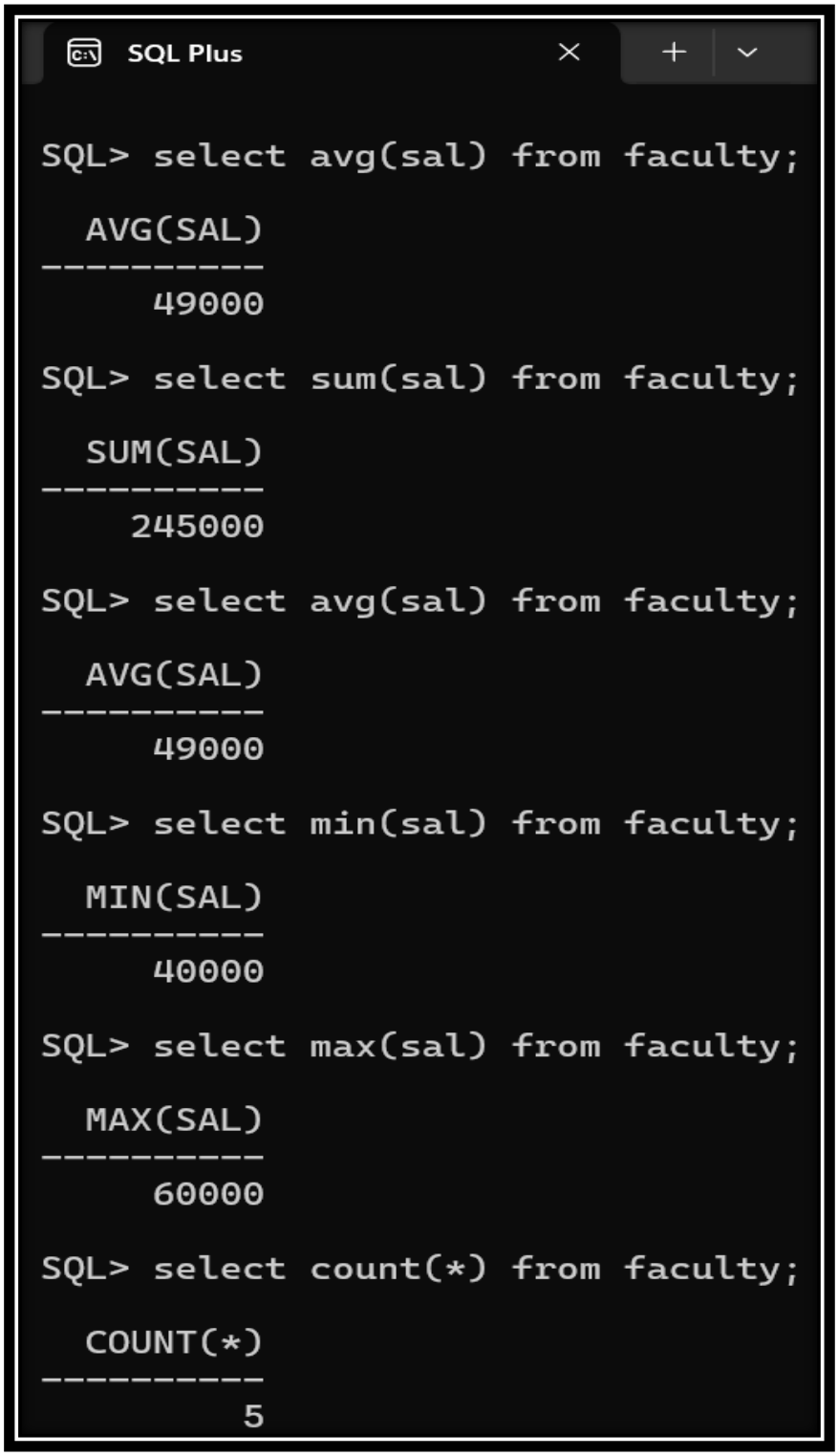
```
SQL> SELECT CURRENT_DATE FROM DUAL;

CURRENT_DATE
-------------------
10-FEB-2025 00:27:06

SQL> SELECT CURRENT_TIMESTAMP FROM DUAL;

CURRENT_TIMESTAMP
---------------------------------------------------------------------------
10-FEB-25 12.27.08.727000 AM +05:30

SQL> SELECT MONTHS_BETWEEN(TO_DATE('02-01-2025','MM-DD-YYYY'),(TO_DATE('01-01-2025','MM-DD-YYYY'))) FROM DUAL;

MONTHS_BETWEEN(TO_DATE('02-01-2025','MM-DD-YYYY'),(TO_DATE('01-01-2025','MM-DD-Y
---------------------------------------------------------------------------
                                                                          1

SQL> SELECT SYSDATE FROM DUAL;

SYSDATE
-------------------
10-FEB-2025 00:27:17

SQL> SELECT LAST_DAY(SYSDATE) FROM DUAL;

LAST_DAY(SYSDATE)
-------------------
28-FEB-2025 00:27:48

SQL> SELECT NEXT_DAY('10-FEB-2025','FRIDAY') "NEXT DAY" FROM DUAL;

NEXT DAY
-------------------
14-FEB-2025 00:00:00
```

4) Aggregate Functions:

a. SUM()
b. AVG()
c. MIN()
d. MAX()
e. COUNT(*)

SQL Plus

```
SQL> select * from faculty;

       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP NO_OF_AWARDS
---------- ------------
         1 Aditi              1      40000 Rohini          9999999999 04-JUL-95
         5

         2 Akshay             2      50000 Noida           8888888888 13-JUN-94
         6

         3 Aman               3      60000 Dwarka          7777777777 02-OCT-95
         5


       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP NO_OF_AWARDS
---------- ------------
         4 Salman             4      45000 Janakpuri       6666666666 01-APR-94
         6

         5 Poonam             5      50000 Rohini          5555555555 05-JUL-96
         4


SQL> desc faculty;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 FID                                       NOT NULL NUMBER(38)
 NAME                                               VARCHAR2(10)
 DNO                                                NUMBER(38)
 SAL                                                NUMBER(38)
 ADDRESS                                            VARCHAR2(15)
 PHONE                                              NUMBER(38)
 DOB                                                DATE
 EXP                                                NUMBER(38)
 NO_OF_AWARDS                                       NUMBER(38)
```



Learning Outcome:

EXPERIMENT 5

Problem statement: Implement queries including Sorting, Grouping and Subqueries- like any, all, exists, not exists.

Theory:

SQL Commands:

1) Create the below mentioned tables with proper constraints: -
   a. Faculty (<u>fid</u>, name, *dno*, sal, address, phone, dob, exp)
   b. Dept (<u>dno</u>, dname, budget)

```
SQL Plus

SQL> create table Dept
  2  (
  3  dno int Primary Key,
  4  dname varchar(10),
  5  budget int
  6  );

Table created.

SQL> create table Faculty
  2  (
  3  fid int Primary Key,
  4  name varchar(10),
  5  dno int,foreign key(dno) references Dept(dno),
  6  sal int,
  7  address varchar(15),
  8  phone int,
  9  dob date,
 10  exp int
 11  );

Table created.
```

```
SQL Plus

SQL> insert into dept values(&1, '&2', &3);
Enter value for 1: 10
Enter value for 2: AIML
Enter value for 3: 5000
old   1: insert into dept values(&1, '&2', &3)
new   1: insert into dept values(10, 'AIML', 5000)

1 row created.

SQL> insert into dept values(&1, '&2', &3);
Enter value for 1: 20
Enter value for 2: AIDS
Enter value for 3: 7000
old   1: insert into dept values(&1, '&2', &3)
new   1: insert into dept values(20, 'AIDS', 7000)

1 row created.
```

SQL Plus

```
SQL> insert into dept values(&1, '&2', &3);
Enter value for 1: 30
Enter value for 2: IIOT
Enter value for 3: 8000
old   1: insert into dept values(&1, '&2', &3)
new   1: insert into dept values(30, 'IIOT', 8000)

1 row created.

SQL> insert into dept values(&1, '&2', &3);
Enter value for 1: 40
Enter value for 2: CSE
Enter value for 3: 9000
old   1: insert into dept values(&1, '&2', &3)
new   1: insert into dept values(40, 'CSE', 9000)

1 row created.

SQL> insert into dept values(&1, '&2', &3);
Enter value for 1: 50
Enter value for 2: IT
Enter value for 3: 12000
old   1: insert into dept values(&1, '&2', &3)
new   1: insert into dept values(50 , 'IT', 12000)

1 row created.
```

SQL Plus

```
SQL> insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8);
Enter value for 1: 1
Enter value for 2: Puneet
Enter value for 3: 10
Enter value for 4: 5000
Enter value for 5: Paschim Vihar
Enter value for 6: 98123
Enter value for 7: 01-March-85
Enter value for 8: 14
old   1: insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8)
new   1: insert into faculty values(1,'Puneet',10,5000,'Paschim Vihar',98123,'01-March-85',14)

1 row created.
```

SQL Plus

```
SQL> insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8);
Enter value for 1: 2
Enter value for 2: Ashish
Enter value for 3: 10
Enter value for 4: 5000
Enter value for 5: Tilak Nagar
Enter value for 6: 98124
Enter value for 7: 07-November-85
Enter value for 8: 13
old   1: insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8)
new   1: insert into faculty values(2,'Ashish',10,5000,'Tilak Nagar',98124,'07-November-85',13)

1 row created.

SQL> insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8);
Enter value for 1: 3
Enter value for 2: Ishleen
Enter value for 3: 20
Enter value for 4: 6000
Enter value for 5: Rajouri
Enter value for 6: 98174
Enter value for 7: 08-December-85
Enter value for 8: 5
old   1: insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8)
new   1: insert into faculty values(3,'Ishleen',20,6000,'Rajouri',98174,'08-December-85',5)

1 row created.

SQL> insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8);
Enter value for 1: 4
Enter value for 2: Sonakshi
Enter value for 3: 30
Enter value for 4: 2000
Enter value for 5: Pitampura
Enter value for 6: NULL
Enter value for 7: 23-June-85
Enter value for 8: 8
old   1: insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8)
new   1: insert into faculty values(4,'Sonakshi',30,2000,'Pitampura',NULL,'23-June-85',8)

1 row created.
```

SQL Plus

```
SQL> insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8);
Enter value for 1: 5
Enter value for 2: Suman
Enter value for 3: 20
Enter value for 4: 1500
Enter value for 5: ITO
Enter value for 6: 98764
Enter value for 7: 19-April-85
Enter value for 8: 19
old   1: insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8)
new   1: insert into faculty values(5,'Suman',20,1500,'ITO',98764,'19-April-85',19)

1 row created.

SQL> insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8);
Enter value for 1: 6
Enter value for 2: Gopal
Enter value for 3: 40
Enter value for 4: 9000
Enter value for 5: Vikaspuri
Enter value for 6: 98960
Enter value for 7: 10-March-85
Enter value for 8: 12
old   1: insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8)
new   1: insert into faculty values(6,'Gopal',40,9000,'Vikaspuri',98960,'10-March-85',12)

1 row created.

SQL> insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8);
Enter value for 1: 7
Enter value for 2: Archa
Enter value for 3: 40
Enter value for 4: 6000
Enter value for 5: Hari Nagar
Enter value for 6: NULL
Enter value for 7: 07-March-85
Enter value for 8: 9
old   1: insert into faculty values(&1,'&2',&3,&4,'&5',&6,'&7',&8)
new   1: insert into faculty values(7,'Archa',40,6000,'Hari Nagar',NULL,'07-March-85',9)

1 row created.
```

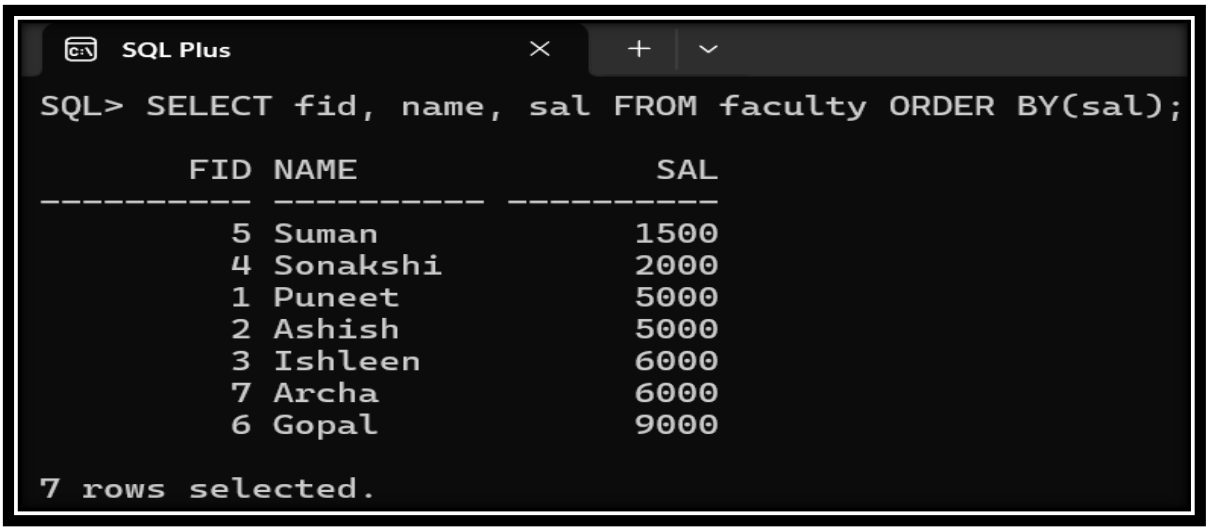
SQL Plus

```
SQL> select * from dept;

       DNO DNAME          BUDGET
---------- ---------- ----------
        10 AIML             5000
        20 AIDS             7000
        30 IIOT             8000
        40 CSE              9000
        50 IT              12000
```

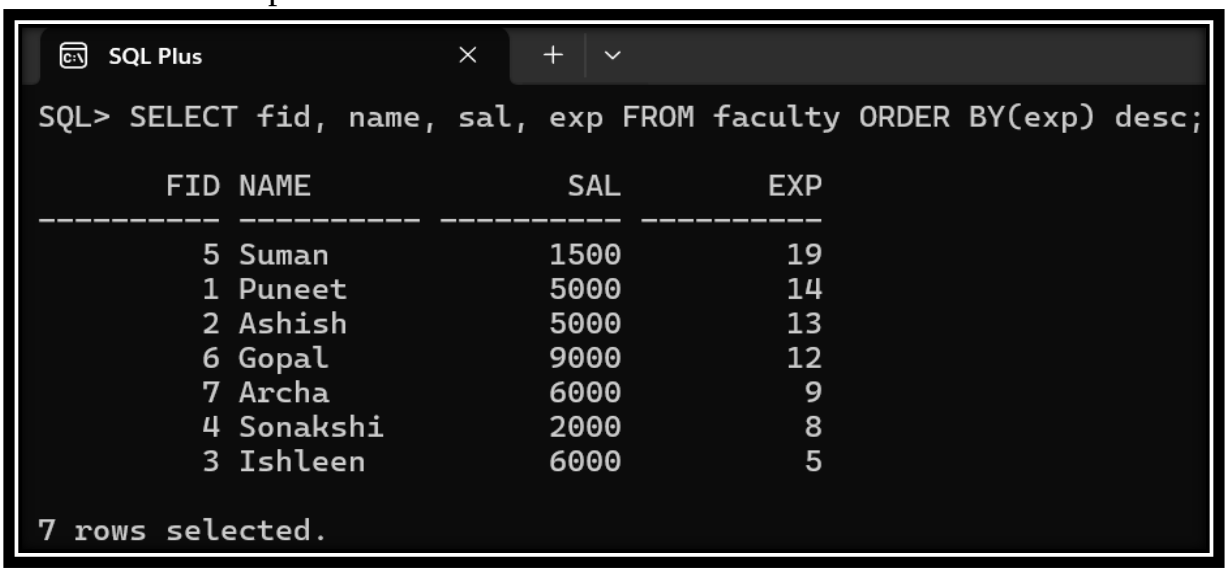
SQL Plus

```
SQL> select * from faculty;

       FID NAME            DNO        SAL ADDRESS            PHONE DOB
---------- ---------- ---------- ---------- -------------- ---------- ---------
       EXP
----------
         1 Puneet             10       5000 Paschim Vihar          98123 01-MAR-85
        14

         2 Ashish             10       5000 Tilak Nagar            98124 07-NOV-85
        13

         3 Ishleen            20       6000 Rajouri                98174 08-DEC-85
         5


       FID NAME            DNO        SAL ADDRESS            PHONE DOB
---------- ---------- ---------- ---------- -------------- ---------- ---------
       EXP
----------
         4 Sonakshi           30       2000 Pitampura                 23-JUN-85
         8

         5 Suman              20       1500 ITO                    98764 19-APR-85
        19

         6 Gopal              40       9000 Vikaspuri              98960 10-MAR-85
        12


       FID NAME            DNO        SAL ADDRESS            PHONE DOB
---------- ---------- ---------- ---------- -------------- ---------- ---------
       EXP
----------
         7 Archa              40       6000 Hari Nagar                07-MAR-85
         9


7 rows selected.
```

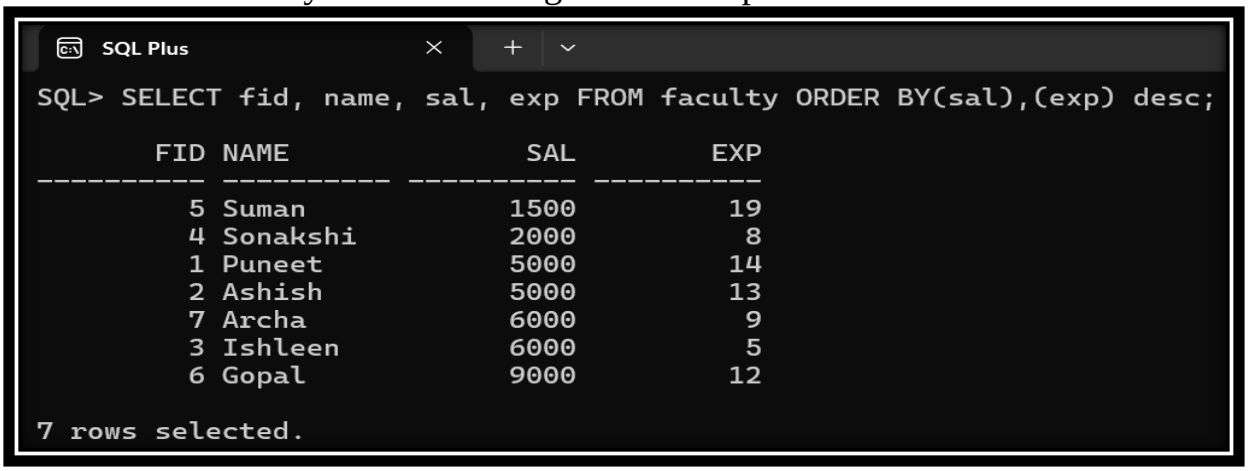
2) Retrieve the ID, Name and Salary of Faculty in increasing order of Salary.



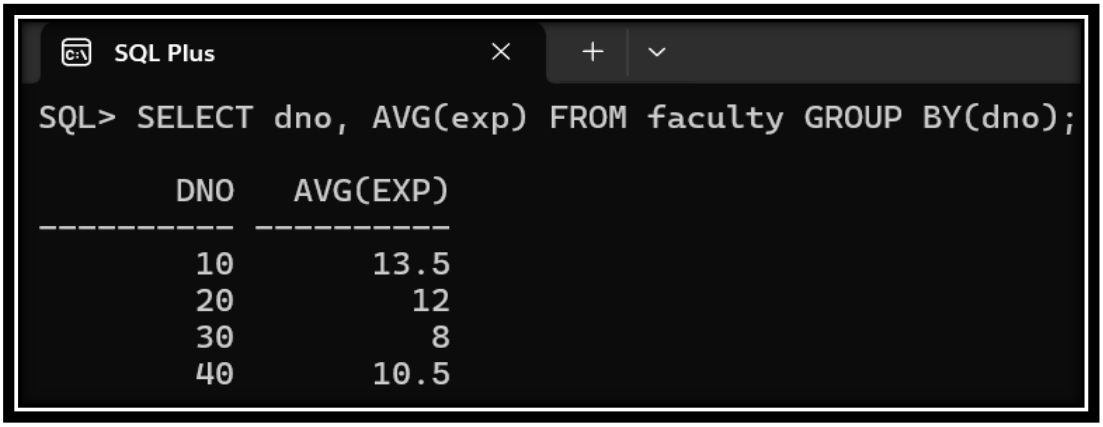
3) Retrieve the ID, Name, Salary and Experience of Faculty in decreasing order of Experience.



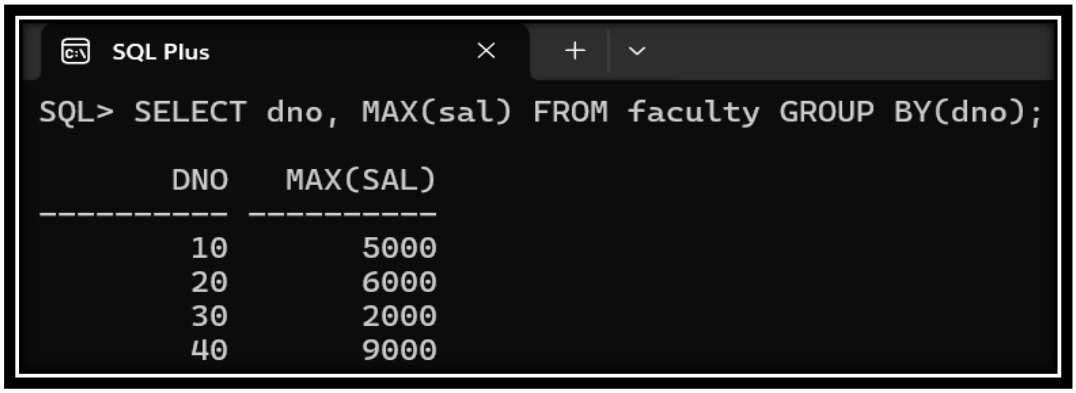
4) Retrieve the ID, Name, Salary and Experience of Faculty in increasing order of Salary and Decreasing order of Experience.



5) Show the average experience department wise

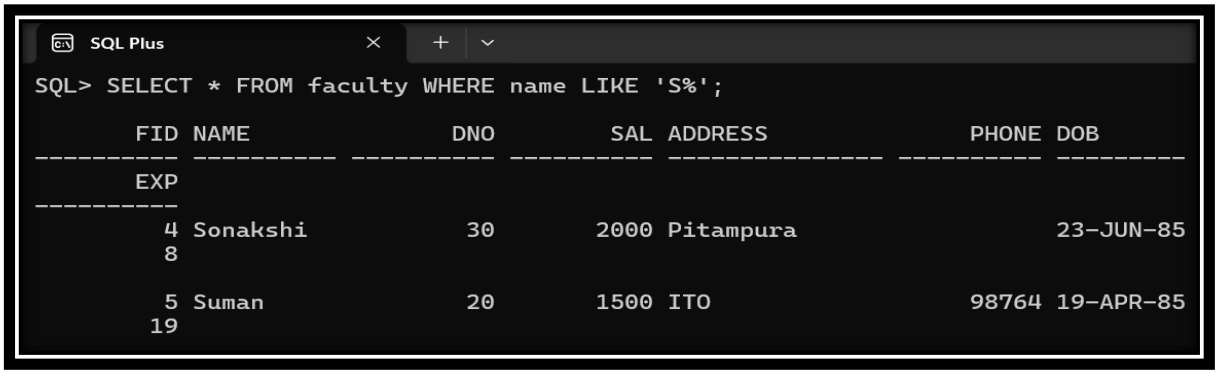


6) Retrieve the maximum salary department wise.



7) Retrieve the average salary department wise having average salary greater than 5000.

8) Retrieve faculties details where name (table should have names accordingly): -
   a) Starting with 'S'



b) Ending with 'n'

```
SQL> SELECT * FROM faculty WHERE name LIKE '%n';

       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP
----------
         3 Ishleen            20       6000 Rajouri             98174 08-DEC-85
         5

         5 Suman              20       1500 ITO                 98764 19-APR-85
        19
```

c) Starting with 'a' and ending with 'h'

```
SQL> SELECT * FROM faculty WHERE name LIKE 'A%h';

       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP
----------
         2 Ashish             10       5000 Tilak Nagar         98124 07-NOV-85
        13
```

d) Must contain 'a'

```
SQL> SELECT * FROM faculty WHERE name LIKE '%a%';

       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP
----------
         4 Sonakshi           30       2000 Pitampura                23-JUN-85
         8

         5 Suman              20       1500 ITO                 98764 19-APR-85
        19

         6 Gopal              40       9000 Vikaspuri           98960 10-MAR-85
        12


       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP
----------
         7 Archa              40       6000 Hari Nagar               07-MAR-85
         9
```

e) Should have 5 letters only starting with 'g'

```
SQL> SELECT * FROM faculty WHERE name LIKE 'G____';

       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP
----------
         6 Gopal              40       9000 Vikaspuri           98960 10-MAR-85
        12
```

9) Show the use of ANY/ALL clauses.

```
SQL> SELECT * FROM dept WHERE dno = ANY (SELECT dno FROM faculty WHERE dno<30);

       DNO DNAME          BUDGET
---------- ---------- ----------
        10 AIML             5000
        20 AIDS             7000

SQL> SELECT * FROM faculty WHERE dno = ANY (SELECT dno FROM dept WHERE dno>20);

       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP
----------
         4 Sonakshi           30       2000 Pitampura                  23-JUN-85
         8

         6 Gopal              40       9000 Vikaspuri            98960 10-MAR-85
        12

         7 Archa              40       6000 Hari Nagar                 07-MAR-85
         9
```

```
SQL> SELECT * FROM dept WHERE dno > ALL (SELECT dno FROM faculty WHERE dno<30);

       DNO DNAME          BUDGET
---------- ---------- ----------
        30 IIOT             8000
        40 CSE              9000
        50 IT              12000

SQL> SELECT * FROM faculty WHERE dno < ALL (SELECT dno FROM dept WHERE dno>20);

       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP
----------
         5 Suman              20       1500 ITO                  98764 19-APR-85
        19

         3 Ishleen            20       6000 Rajouri              98174 08-DEC-85
         5

         2 Ashish             10       5000 Tilak Nagar          98124 07-NOV-85
        13


       FID NAME              DNO        SAL ADDRESS              PHONE DOB
---------- ---------- ---------- ---------- --------------- ---------- ---------
       EXP
----------
         1 Puneet             10       5000 Paschim Vihar        98123 01-MAR-85
        14
```

10) Show the use of EXISTS/NOT EXISTS clauses

```
SQL> SELECT fid, name, dno FROM faculty WHERE EXISTS(SELECT dno FROM dept WHERE dno>40);

       FID NAME              DNO
---------- ---------- ----------
         1 Puneet             10
         2 Ashish             10
         3 Ishleen            20
         4 Sonakshi           30
         5 Suman              20
         6 Gopal              40
         7 Archa              40

7 rows selected.

SQL> SELECT fid, name, dno FROM faculty WHERE EXISTS(SELECT dno FROM dept WHERE dno>50);

no rows selected
```

```
SQL> SELECT * FROM dept WHERE NOT EXISTS(SELECT dno FROM faculty WHERE dno<40);

no rows selected

SQL> SELECT * FROM dept WHERE NOT EXISTS(SELECT dno FROM faculty WHERE dno>40);

       DNO DNAME          BUDGET
---------- ---------- ----------
        10 AIML             5000
        20 AIDS             7000
        30 IIOT             8000
        40 CSE              9000
        50 IT              12000
```

11) Find the names of Faculty who work in department AIML (using subquery)

```
SQL> SELECT name FROM faculty WHERE dno IN (SELECT dno FROM dept where dname='AIML');

NAME
----------
Puneet
Ashish
```

Learning Outcome:

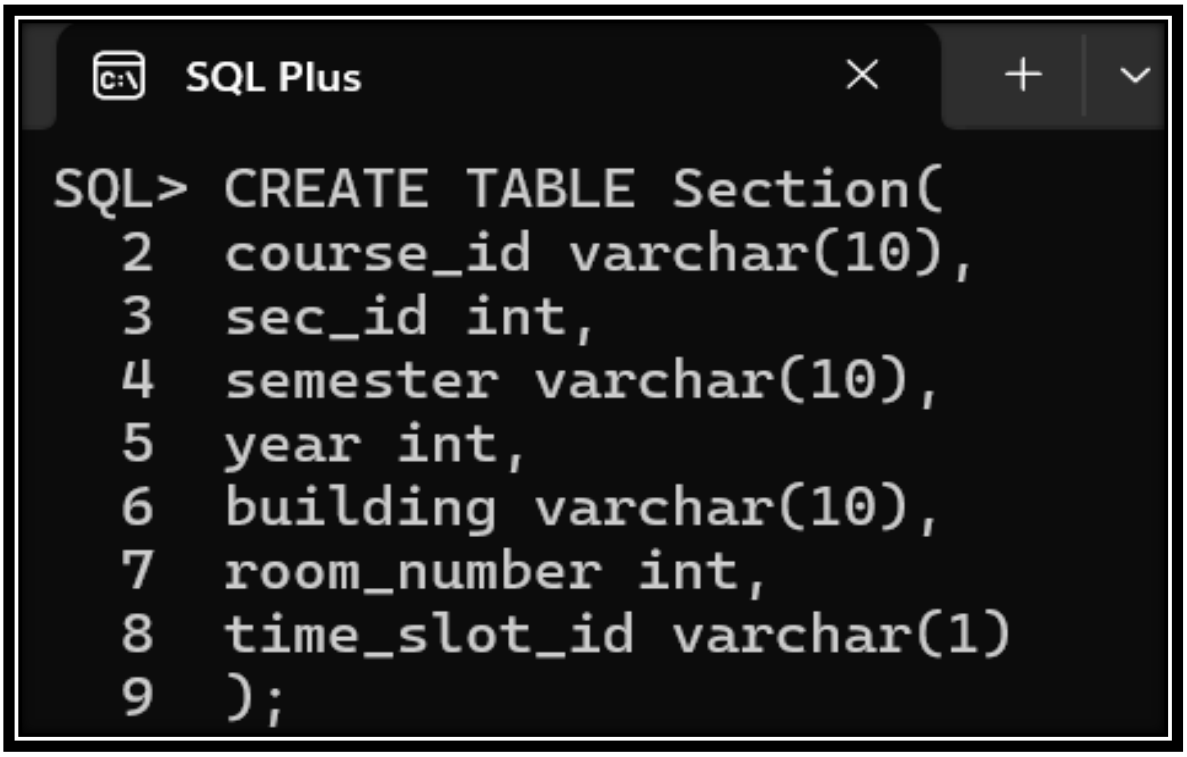
EXPERIMENT 6

Problem statement: Implement queries including various Set operations (Union, Intersection, Except etc.).

Theory:

SQL Commands:

1) Create a table named section with the following columns and add appropriate values to it:
    a. Section (course_id, sec_id, semester, year, building, room_number, time_slot_id)




```
SQL> desc section;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COURSE_ID                                          VARCHAR2(10)
 SEC_ID                                             NUMBER(38)
 SEMESTER                                           VARCHAR2(10)
 YEAR                                               NUMBER(38)
 BUILDING                                           VARCHAR2(10)
 ROOM_NUMBER                                        NUMBER(38)
 TIME_SLOT_ID                                       VARCHAR2(1)
```

KUNSH SABHARWAL

SQL Plus

```
SQL> INSERT INTO section VALUES('BIO-101', 1, 'Summer', 2017, 'Painter', 514, 'B');

1 row created.

SQL> INSERT INTO section VALUES('BIO-301', 1, 'Summer', 2018, 'Painter', 514, 'A');

1 row created.

SQL> INSERT INTO section VALUES('CS-101', 1, 'Fall', 2017, 'Packard', 101, 'H');

1 row created.

SQL> INSERT INTO section VALUES('CS-101', 1, 'Spring', 2018, 'Packard', 101, 'F');

1 row created.

SQL> INSERT INTO section VALUES('CS-190', 1, 'Spring', 2017, 'Taylor', 3128, 'E');

1 row created.

SQL> INSERT INTO section VALUES('CS-190', 2, 'Spring', 2017, 'Taylor', 3128, 'A');

1 row created.

SQL> INSERT INTO section VALUES('CS-315', 1, 'Spring', 2018, 'Watson', 120, 'D');

1 row created.

SQL> INSERT INTO section VALUES('CS-319', 1, 'Spring', 2018, 'Watson', 100, 'B');

1 row created.

SQL> INSERT INTO section VALUES('CS-319', 2, 'Spring', 2018, 'Taylor', 3128, 'C');

1 row created.

SQL> INSERT INTO section VALUES('CS-347', 1, 'Fall', 2017, 'Taylor', 3128, 'A');

1 row created.

SQL> INSERT INTO section VALUES('EE-181', 1, 'Spring', 2017, 'Taylor', 3128, 'C');

1 row created.

SQL> INSERT INTO section VALUES('FIN-201', 1, 'Spring', 2018, 'Packard', 101, 'B');

1 row created.

SQL> INSERT INTO section VALUES('HIS-351', 1, 'Spring', 2018, 'Painter', 514, 'C');

1 row created.

SQL> INSERT INTO section VALUES('MU-199', 1, 'Spring', 2018, 'Packard', 101, 'D');

1 row created.

SQL> INSERT INTO section VALUES('PHY-101', 1, 'Fall', 2017, 'Watson', 100, 'A');

1 row created.
```

SQL Plus

```
SQL> select * from section;

COURSE_ID      SEC_ID SEMESTER         YEAR BUILDING   ROOM_NUMBER T
---------- ---------- ---------- ---------- ---------- ----------- -
BIO-101             1 Summer           2017 Painter            514 B
BIO-301             1 Summer           2018 Painter            514 A
CS-101              1 Fall             2017 Packard            101 H
CS-101              1 Spring           2018 Packard            101 F
CS-190              1 Spring           2017 Taylor            3128 E
CS-190              2 Spring           2017 Taylor            3128 A
CS-315              1 Spring           2018 Watson             120 D
CS-319              1 Spring           2018 Watson             100 B
CS-319              2 Spring           2018 Taylor            3128 C
CS-347              1 Fall             2017 Taylor            3128 A
EE-181              1 Spring           2017 Taylor            3128 C

COURSE_ID      SEC_ID SEMESTER         YEAR BUILDING   ROOM_NUMBER T
---------- ---------- ---------- ---------- ---------- ----------- -
FIN-201             1 Spring           2018 Packard            101 B
HIS-351             1 Spring           2018 Painter            514 C
MU-199              1 Spring           2018 Packard            101 D
PHY-101             1 Fall             2017 Watson             100 A

15 rows selected.
```

2) Find courses that ran in Fall 2017.

```
SQL> SELECT course_id FROM section WHERE semester='Fall' AND year=2017;

COURSE_ID
----------
CS-101
CS-347
PHY-101
```

3) Find the courses that ran in Spring 2018.

```
SQL> SELECT course_id FROM section WHERE semester='Spring' AND year=2018;

COURSE_ID
----------
CS-101
CS-315
CS-319
CS-319
FIN-201
HIS-351
MU-199

7 rows selected.
```

4) Find courses that ran in Fall 2017 or in Spring 2018.

```
SQL> SELECT course_id FROM section WHERE semester='Fall' AND year=2017 UNION SELECT course_id FROM section WHERE semester='Spring' AND year=2018;

COURSE_ID
----------
CS-101
CS-347
PHY-101
CS-315
CS-319
FIN-201
HIS-351
MU-199

8 rows selected.
```

5) Find courses that ran in Fall 2017 and in Spring 2018.

```
SQL> SELECT course_id FROM section WHERE semester='Fall' AND year=2017 INTERSECT SELECT course_id FROM section WHERE semester='Spring' AND year=2018;

COURSE_ID
----------
CS-101
```

6) Find courses that ran in Fall 2017 but not in Spring 2018.

```
SQL> SELECT course_id FROM section WHERE semester='Fall' AND year=2017 MINUS SELECT course_id FROM section WHERE semester='Spring' AND year=2018;

COURSE_ID
----------
CS-347
PHY-101
```

7) Find courses that ran in Spring 2018 but not in Fall 2017.

```
SQL> SELECT course_id FROM section WHERE semester='Spring' AND year=2018 MINUS SELECT course_id FROM section WHERE semester='Fall' AND year=2017;

COURSE_ID
----------
CS-315
CS-319
FIN-201
HIS-351
MU-199
```

8) Create 2 different tables with same number of columns and make use of Union, Union all, Intersect and Except.
   a. Building (id, name)
   b. Construction (id, name)

```
SQL> CREATE TABLE building(
  2      id int,
  3      name varchar(10)
  4      );

Table created.

SQL> CREATE TABLE construction(
  2      id int,
  3      name varchar(10)
  4      );

Table created.
```

```
SQL> INSERT INTO building VALUES(1,'Emporio');

1 row created.

SQL> INSERT INTO building VALUES(2,'Unity-one');

1 row created.

SQL> INSERT INTO building VALUES(3,'Pacific');

1 row created.

SQL> INSERT INTO building VALUES(4,'Promenade');

1 row created.

SQL> INSERT INTO building VALUES(5,'D-Mall');

1 row created.

SQL> select * from building;

        ID NAME
---------- ----------
         1 Emporio
         2 Unity-one
         3 Pacific
         4 Promenade
         5 D-Mall
```

```
SQL> INSERT INTO construction VALUES(2,'Unity-one');

1 row created.

SQL> INSERT INTO construction VALUES(5,'D-Mall');

1 row created.

SQL> INSERT INTO construction VALUES(6,'DLF');

1 row created.

SQL> INSERT INTO construction VALUES(7,'Mohan');

1 row created.

SQL> INSERT INTO construction VALUES(8,'DLF');

1 row created.

SQL> select * from construction;

        ID NAME
---------- ----------
         2 Unity-one
         5 D-Mall
         6 DLF
         7 Mohan
         8 DLF
```

```
SQL> SELECT * FROM building UNION SELECT * FROM construction;

        ID NAME
---------- ----------
         1 Emporio
         2 Unity-one
         3 Pacific
         4 Promenade
         5 D-Mall
         6 DLF
         7 Mohan
         8 DLF

8 rows selected.
```

```
SQL> SELECT * FROM building UNION ALL SELECT * FROM construction;

        ID NAME
---------- ----------
         1 Emporio
         2 Unity-one
         3 Pacific
         4 Promenade
         5 D-Mall
         2 Unity-one
         5 D-Mall
         6 DLF
         7 Mohan
         8 DLF

10 rows selected.
```

```
SQL> SELECT * FROM building INTERSECT SELECT * FROM construction;

        ID NAME
---------- ----------
         2 Unity-one
         5 D-Mall
```

```
SQL> SELECT * FROM building MINUS SELECT * FROM construction;

        ID NAME
---------- ----------
         1 Emporio
         3 Pacific
         4 Promenade
```

Learning Outcome:

EXPERIMENT 7

Problem statement: Implement various JOIN operations- (Inner, Outer).

Theory:

SQL Commands:

1) Create the following tables
    a. Employee – (empid, empname, *deptid)*
    b. Department – (*deptid*, deptname)

```
SQL> CREATE TABLE Employee(
  2          empid int,
  3     empname varchar(10),
  4     deptid int PRIMARY KEY
  5  );

Table created.

SQL> CREATE TABLE Department(
  2     deptid int, FOREIGN KEY (deptid) REFERENCES Employee(deptid),
  3     deptname varchar(10)
  4  );

Table created.
```

2) Insert values in the above tables.

```
SQL> INSERT INTO Employee VALUES(101, 'Ram', 1);

1 row created.

SQL> INSERT INTO Employee VALUES(102, 'Raju', 2);

1 row created.

SQL> INSERT INTO Employee VALUES(103, 'Shyam', 3);

1 row created.

SQL> INSERT INTO Employee VALUES(104, 'Krishna', 4);

1 row created.

SQL> INSERT INTO Employee VALUES(105, 'Rajat', 5);

1 row created.
```

```
SQL> INSERT INTO Department VALUES(1, 'AIML');

1 row created.

SQL> INSERT INTO Department VALUES(2, 'AIDS');

1 row created.

SQL> INSERT INTO Department VALUES(3, 'CSE');

1 row created.

SQL> INSERT INTO Department VALUES(4, 'VLSI');

1 row created.

SQL> INSERT INTO Department VALUES(5, 'IIOT');

1 row created.
```

```
SQL> select * from employee;

     EMPID EMPNAME        DEPTID
---------- ---------- ----------
       101 Ram                 1
       102 Raju                2
       103 Shyam               3
       104 Krishna             4
       105 Rajat               5
```

```
SQL> select * from department;

    DEPTID DEPTNAME
---------- ----------
         1 AIML
         2 AIDS
         3 CSE
         4 VLSI
         5 IIOT
```

3) Implement INNER JOIN on the tables.

```
SQL> SELECT employee.empname,department.deptname FROM employee INNER JOIN department ON employee.deptid=department.deptid;

EMPNAME    DEPTNAME
---------- ----------
Ram        AIML
Raju       AIDS
Shyam      CSE
Krishna    VLSI
Rajat      IIOT

SQL> SELECT e.empname,d.deptname FROM employee e INNER JOIN department d ON e.deptid=d.deptid;

EMPNAME    DEPTNAME
---------- ----------
Ram        AIML
Raju       AIDS
Shyam      CSE
Krishna    VLSI
Rajat      IIOT
```

4) Implement LEFT JOIN (OUTER JOIN) on the tables.

```
SQL> INSERT INTO Employee (empname, deptid) VALUES('Sakshi', 6);

1 row created.

SQL> INSERT INTO Employee (empname, deptid) VALUES('Rakesh', 7);

1 row created.

SQL> INSERT INTO Employee (empname, deptid) VALUES('Suraj', 8);

1 row created.

SQL> select * from employee;

     EMPID EMPNAME        DEPTID
---------- ---------- ----------
       101 Ram                 1
       102 Raju                2
       103 Shyam               3
       104 Krishna             4
       105 Rajat               5
           Sakshi              6
           Rakesh              7
           Suraj               8
```

```
SQL> INSERT INTO Department (deptid) VALUES(6);

1 row created.

SQL> INSERT INTO Department (deptid) VALUES(7);

1 row created.
```

```
SQL> select * from department;

    DEPTID DEPTNAME
---------- ----------
         1 AIML
         2 AIDS
         3 CSE
         4 VLSI
         5 IIOT
         6
         7
```

```
SQL> SELECT e.empid,e.empname,d.deptname,d.deptid FROM employee e LEFT JOIN department d ON e.deptid=d.deptid;

    EMPID EMPNAME    DEPTNAME      DEPTID
--------- ---------- ---------- ---------
      101 Ram        AIML            1
      102 Raju       AIDS            2
      103 Shyam      CSE             3
      104 Krishna    VLSI            4
      105 Rajat      IIOT            5
          Sakshi                     6
          Rakesh                     7
          Suraj

8 rows selected.
```

5) Implement RIGHT JOIN (OUTER JOIN) on the tables.

```
SQL> SELECT e.empid,e.empname,d.deptname,d.deptid FROM employee e RIGHT JOIN department d ON e.deptid=d.deptid;

    EMPID EMPNAME    DEPTNAME      DEPTID
--------- ---------- ---------- ---------
      101 Ram        AIML            1
      102 Raju       AIDS            2
      103 Shyam      CSE             3
      104 Krishna    VLSI            4
      105 Rajat      IIOT            5
          Sakshi                     6
          Rakesh                     7

7 rows selected.
```

6) Implement FULL JOIN (OUTER JOIN) on the tables.

```
SQL> SELECT e.empid,e.empname,d.deptname,d.deptid FROM employee e FULL JOIN department d ON e.deptid=d.deptid;

     EMPID EMPNAME    DEPTNAME       DEPTID
---------- ---------- ---------- ----------
       101 Ram        AIML                1
       102 Raju       AIDS                2
       103 Shyam      CSE                 3
       104 Krishna    VLSI                4
       105 Rajat      IIOT                5
           Sakshi                         6
           Rakesh                         7
           Suraj

8 rows selected.
```

7) Implement THETA and EQUI Join (INNER JOIN).

```
SQL> CREATE TABLE car(
  2  CarModel varchar(10),
  3  CarPrice int
  4  );

Table created.

SQL> CREATE TABLE boat(
  2  BoatModel varchar(10),
  3  BoatPrice int
  4  );

Table created.
```

```
SQL> SELECT * FROM car;

CARMODEL      CARPRICE
---------- ----------
CarA            20000
CarB            30000
CarC            50000

SQL> SELECT * FROM boat;

BOATMODEL   BOATPRICE
---------- ----------
Boat1           10000
Boat2           40000
Boat3           60000
Boat4           30000
```

```
SQL> INSERT INTO car values('CarA',20000);

1 row created.

SQL> INSERT INTO car values('CarB',30000);

1 row created.

SQL> INSERT INTO car values('CarC',50000);

1 row created.
```

```
SQL> INSERT INTO boat values('Boat1',10000);

1 row created.

SQL> INSERT INTO boat values('Boat2',40000);

1 row created.

SQL> INSERT INTO boat values('Boat3',60000);

1 row created.

SQL> INSERT INTO boat values('Boat4',30000);

1 row created.
```

```
SQL> SELECT * FROM car, boat WHERE CarPrice > BoatPrice;

CARMODEL     CARPRICE BOATMODEL   BOATPRICE
---------- ---------- ---------- ----------
CarA            20000 Boat1           10000
CarB            30000 Boat1           10000
CarC            50000 Boat1           10000
CarC            50000 Boat2           40000
CarC            50000 Boat4           30000

SQL> SELECT * FROM car, boat WHERE CarPrice = BoatPrice;

CARMODEL     CARPRICE BOATMODEL   BOATPRICE
---------- ---------- ---------- ----------
CarB            30000 Boat4           30000
```

Learning Outcome:

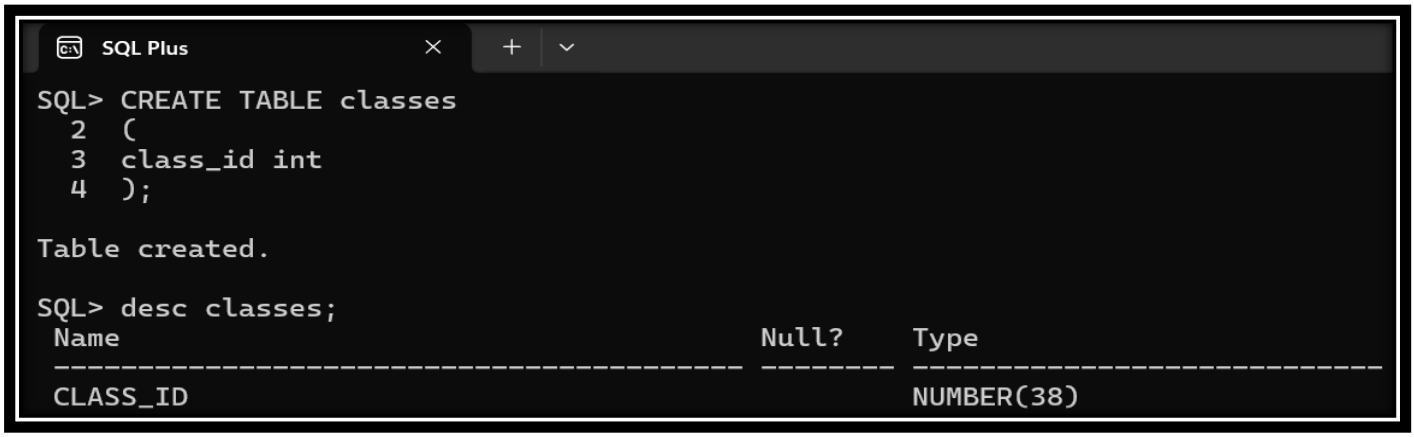
EXPERIMENT 8

Problem statement: Write a PL/SQL program using FOR loop to insert ten rows into a database table.

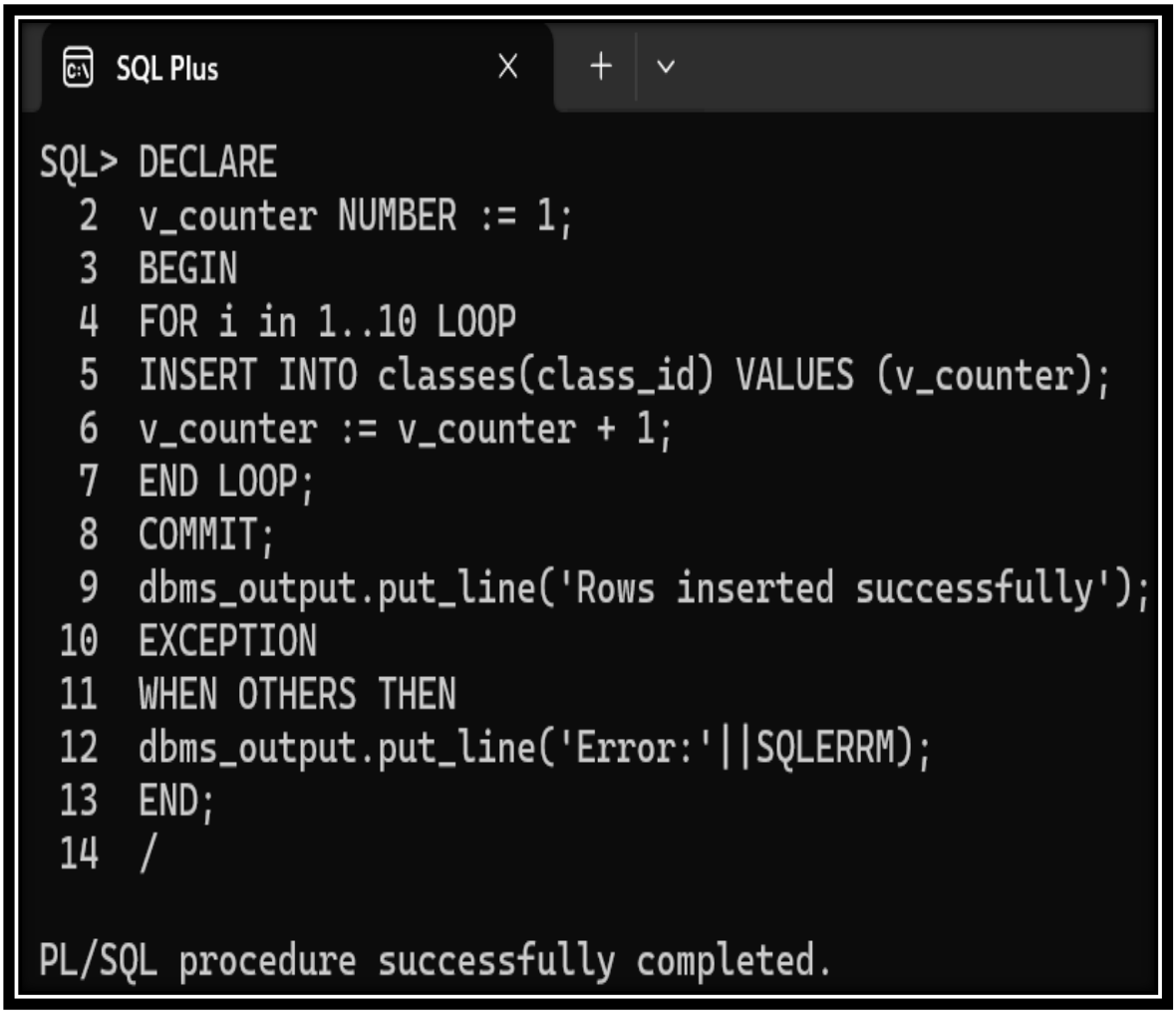
Theory:

SQL Commands:

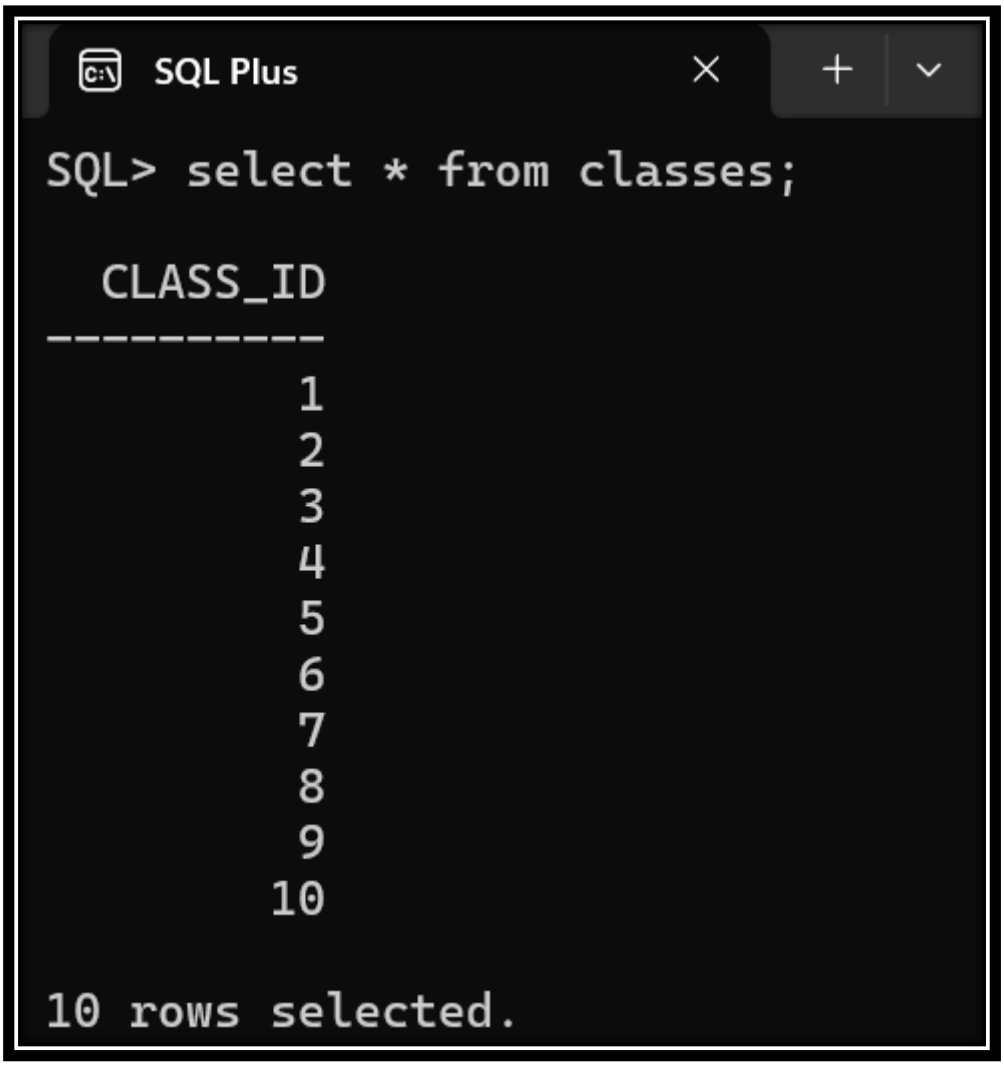
1) Create table classes with only one column (class_id int) to display 10 classes.



2) Create the PL/SQL command to DECLARE a variable v_counter starting from 1 and in BEGIN make a FOR LOOP to run from 1 – 10, incrementing value of v_counter each time. Add these values to the table each time the for loop iterates.



3) Print the full table.



Learning Outcome:

EXPERIMENT 9

Problem statement: Given the table EMPLOYEE (Emp No, Name, Salary, Designation, DeptID), write a cursor to select the five highest-paid employees from the table.

Theory:

SQL Commands:

1) Create table employee with columns (empno int, name varchar (10), salary int, designation varchar (10), deptid int).

```
SQL> CREATE TABLE employee(
  2  empno int,
  3  name varchar(10),
  4  salary int,
  5  designation varchar(15),
  6  deptid int);

Table created.
```

```
SQL> desc employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPNO                                              NUMBER(38)
 NAME                                               VARCHAR2(10)
 SALARY                                             NUMBER(38)
 DESIGNATION                                        VARCHAR2(15)
 DEPTID                                             NUMBER(38)
```

2) Insert appropriate values into the above table with employee number, names, salary, designation, and department ID.

SQL Plus

```
SQL> INSERT INTO employee values(1, 'Ram', 1000, 'Clerk', 101);

1 row created.

SQL> INSERT INTO employee values(2, 'Mohan', 2000, 'AVP', 102);

1 row created.

SQL> INSERT INTO employee values(3, 'Raju', 3000, 'VP', 103);

1 row created.

SQL> INSERT INTO employee values(4, 'Rajesh', 4000, 'EVP', 104);

1 row created.

SQL> INSERT INTO employee values(5, 'Rakesh', 5000, 'TL', 105);

1 row created.

SQL> INSERT INTO employee values(6, 'Raman', 6000, 'Manager', 106);

1 row created.
```

3) Create the required cursor to select only the 5 highest paid employees from the table.

SQL Plus

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2  CURSOR c1 is SELECT empno, name, salary, designation, deptid FROM employee ORDER BY(salary) desc FETCH FIRST 5 ROWS ONLY;
  3  counter int;
  4  e_no int;
  5  e_name varchar(10);
  6  sal int;
  7  des varchar(15);
  8  did int;
  9  BEGIN
 10  OPEN c1;
 11  LOOP
 12  FETCH c1 into e_no, e_name, sal, des, did;
 13  EXIT WHEN c1%notfound;
 14  dbms_output.put_line(e_no || ' ' || e_name || ' ' || sal || ' ' || des || ' ' || did);
 15  END LOOP;
 16  CLOSE c1;
 17  END;
 18  /
6 Raman 6000 Manager 106
5 Rakesh 5000 TL 105
4 Rajesh 4000 EVP 104
3 Raju 3000 VP 103
2 Mohan 2000 AVP 102

PL/SQL procedure successfully completed.
```

Learning Outcome:

EXPERIMENT 10

Problem statement: Illustrate how you can embed PL/SQL in a high-level host language such as C/Java And demonstrates how a banking debit transaction might be done.

Theory:

SQL Commands: -

(a) Create a table in SQL Plus and add values to it and display the table.

SQL Plus

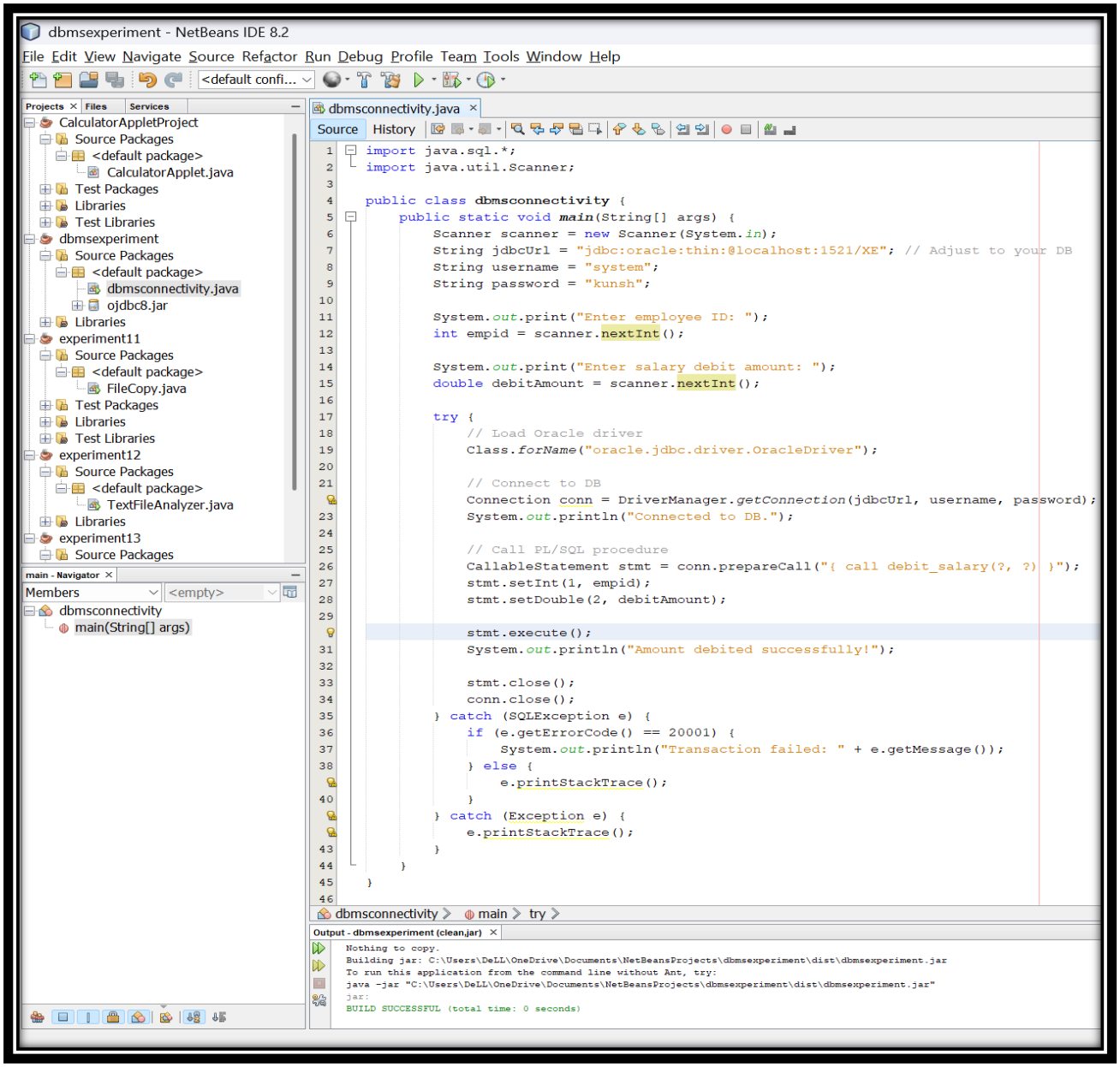
```
SQL> CREATE TABLE employee (
  2  empid int PRIMARY KEY,
  3  empname varchar(10),
  4  salary int);

Table created.

SQL> DESC employee;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPID                                     NOT NULL NUMBER(38)
 EMPNAME                                            VARCHAR2(10)
 SALARY                                             NUMBER(38)

SQL> INSERT INTO employee VALUES (1, 'Ram', 10000);

1 row created.

SQL> INSERT INTO employee VALUES (2, 'Shyam', 5000);

1 row created.

SQL> INSERT INTO employee VALUES (3, 'Kunsh', 4000);

1 row created.

SQL> INSERT INTO employee VALUES (4, 'Rakesh', 7000);

1 row created.

SQL> INSERT INTO employee VALUES (5, 'Rajeev', 3000);

1 row created.

SQL> SELECT * FROM employee;

     EMPID EMPNAME        SALARY
---------- ---------- ----------
         1 Ram             10000
         2 Shyam            5000
         3 Kunsh            4000
         4 Rakesh           7000
         5 Rajeev           3000
```

(b) Create a procedure to implement when the connectivity is done via java.

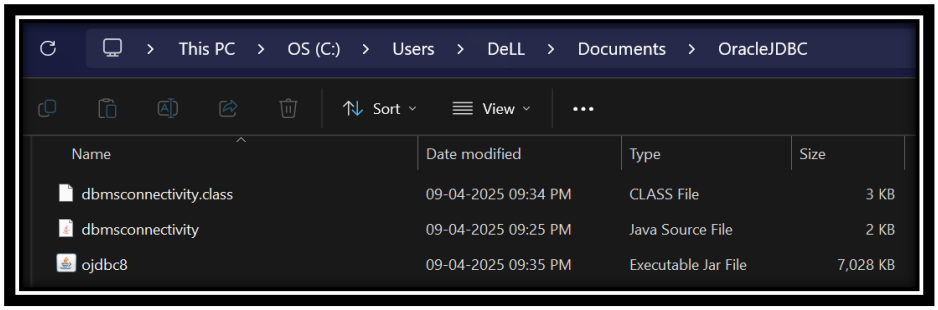
SQL Plus

```
SQL> SET SERVEROUTPUT ON;
SQL> CREATE OR REPLACE PROCEDURE debit_salary (
  2  p_empid IN NUMBER,
  3  p_amount IN NUMBER
  4  )
  5  AS e_salary NUMBER;
  6  BEGIN
  7  SELECT salary INTO e_salary FROM employee WHERE empid = p_empid FOR UPDATE;
  8  IF e_salary < p_amount THEN
  9  RAISE_APPLICATION_ERROR(-20001, 'Insufficient Salary.');
 10  ELSE
 11  UPDATE employee
 12  SET salary = salary - p_amount
 13  WHERE empid = p_empid;
 14  END IF;
 15  END;
 16  /

Procedure created.
```

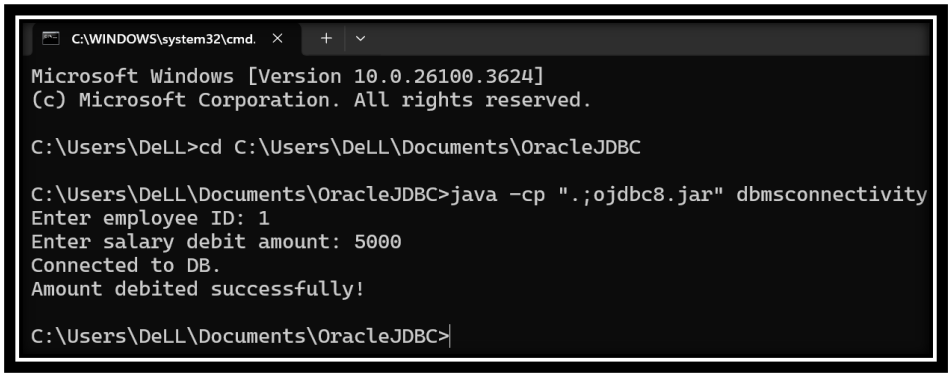
(c) Write the connectivity code in any IDE for Java.



(d) Open command prompt and navigate to the folder containing the .class file, .java file and ojdbc.jar file for the connectivity.



(e) Print the table again in SQL to confirm the debit of salary.



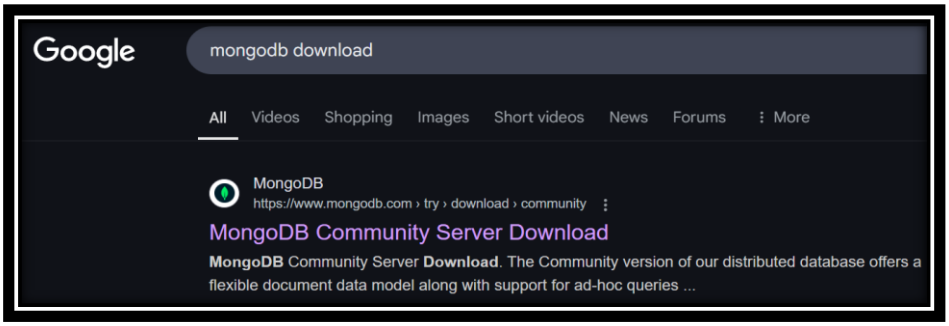
Learning Outcome:

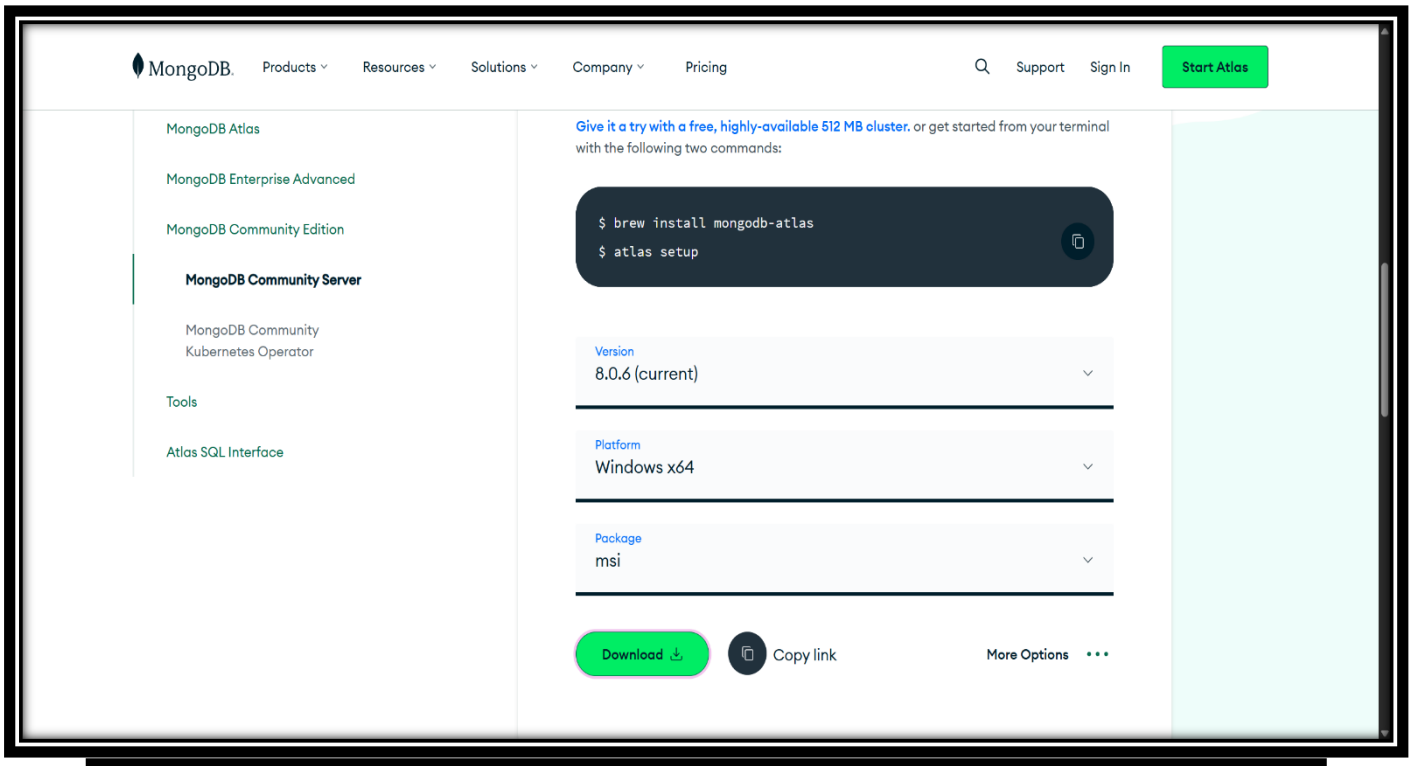
Beyond Curriculum
EXPERIMENT 1

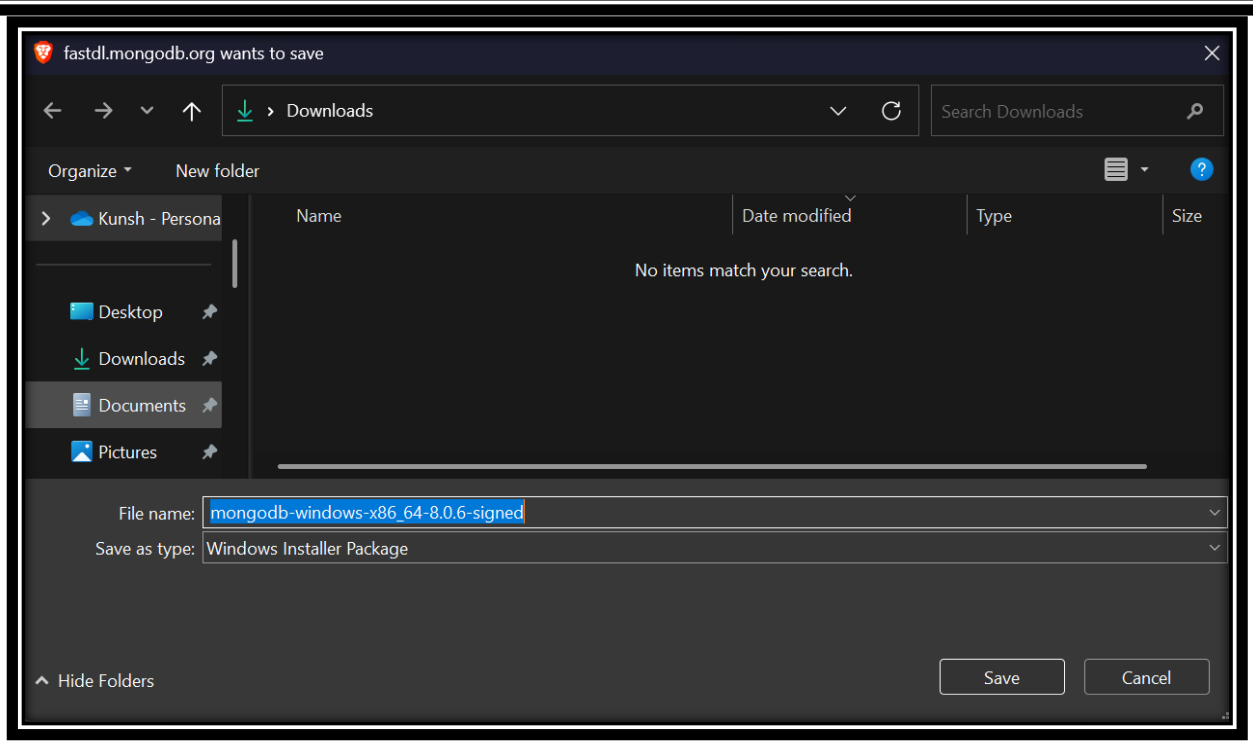
Problem statement: Write the steps to install and implement NOSQL databases-MongoDB**.**

Theory:

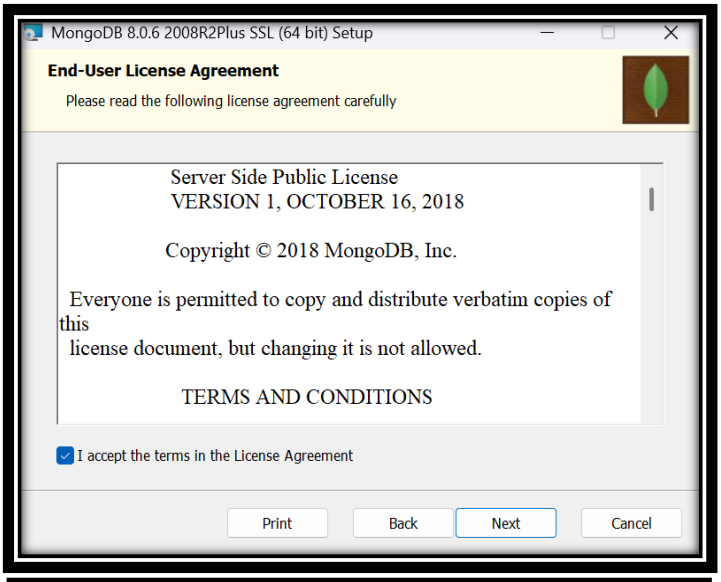
Installation Screenshots:

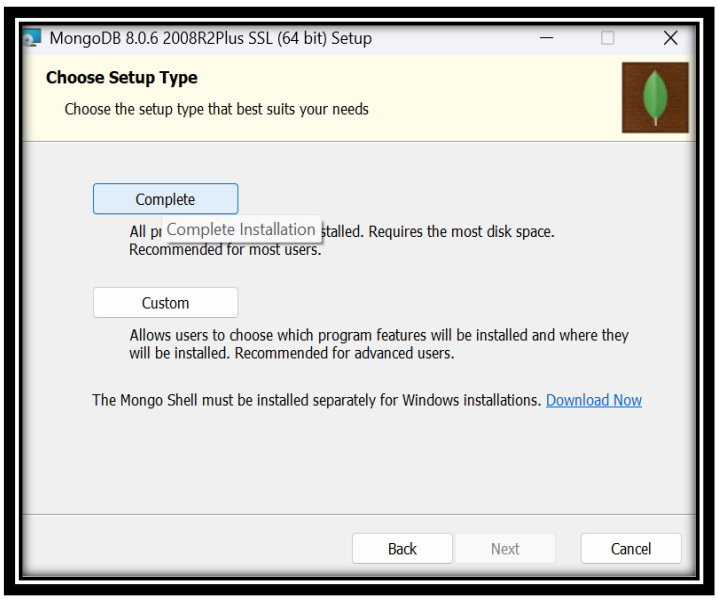


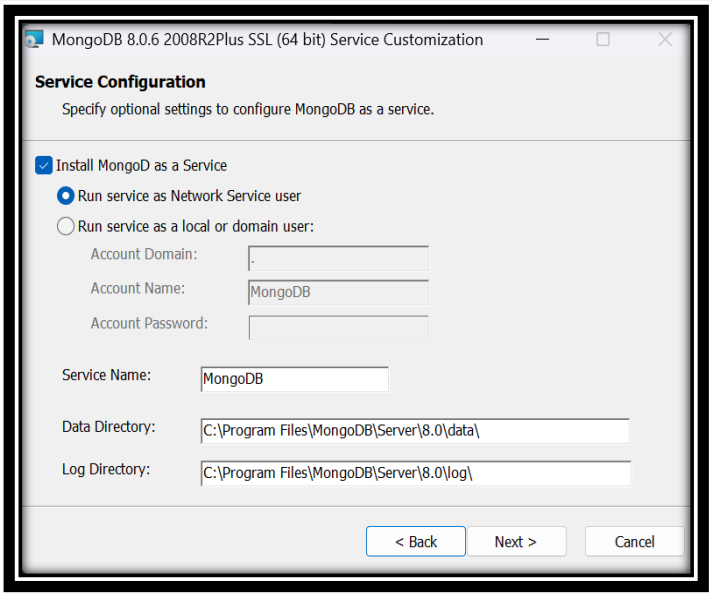


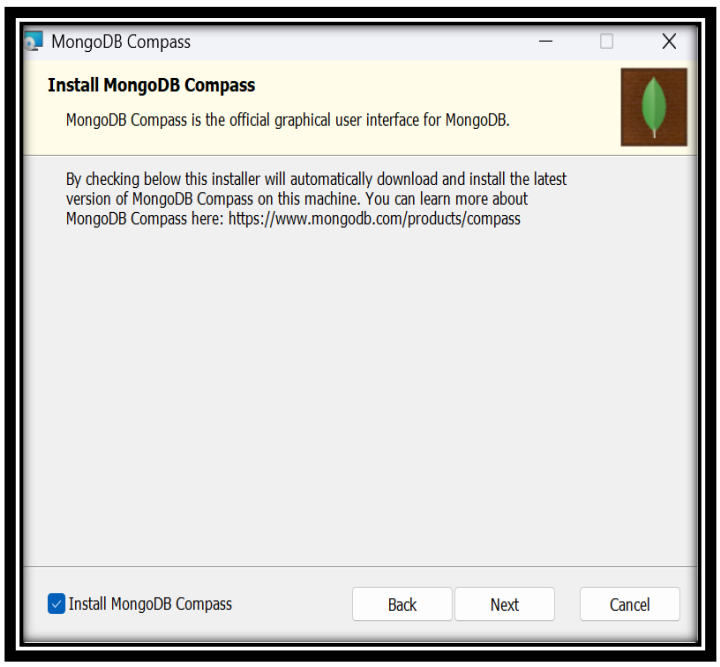


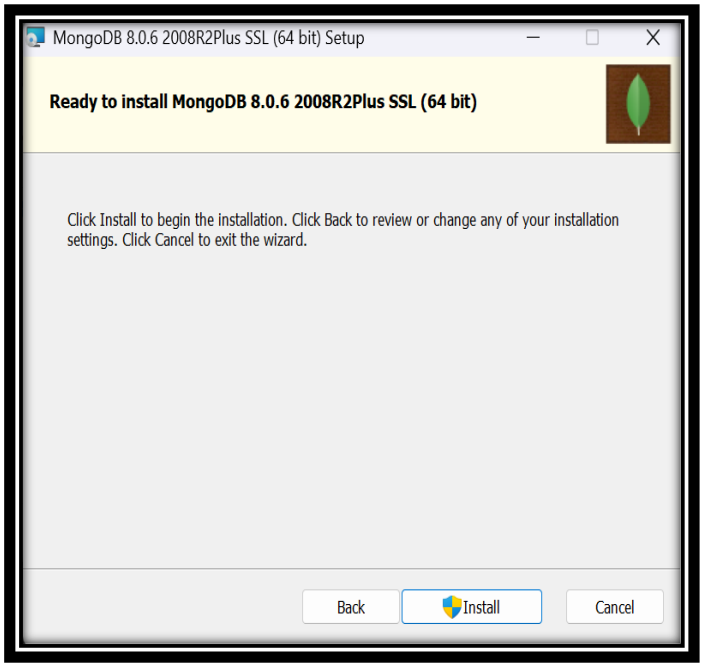
KUNSH SABHARWAL



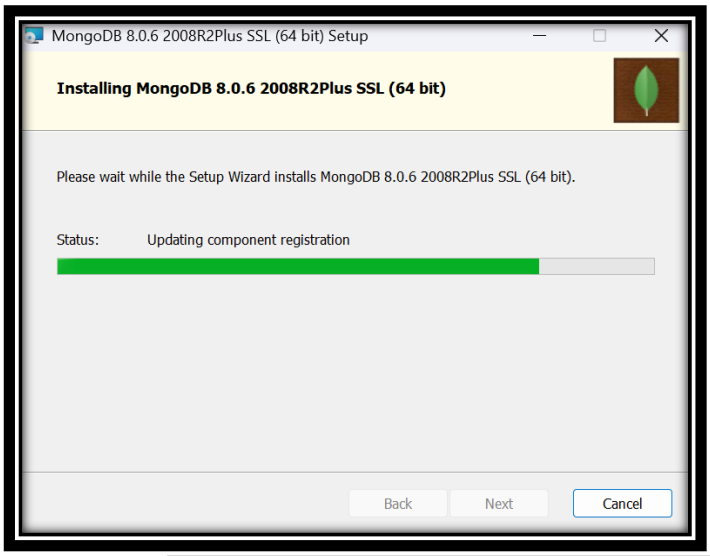


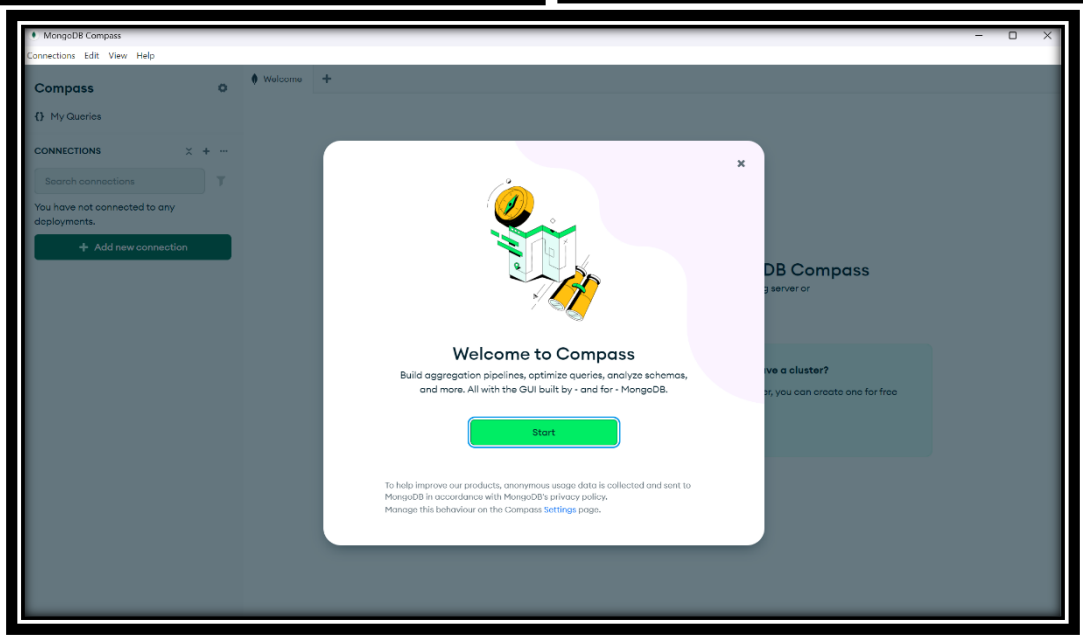


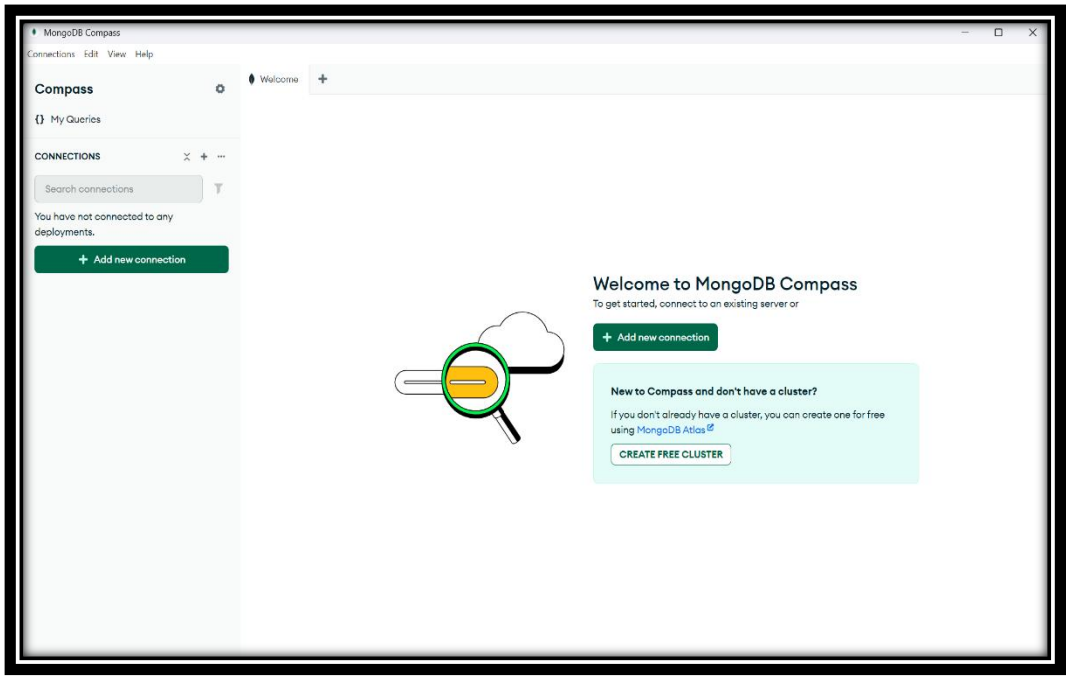


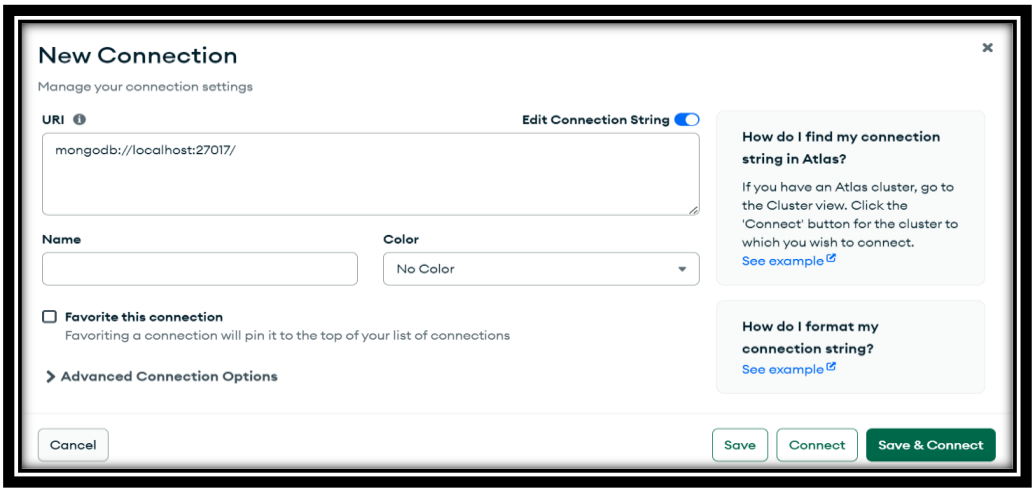


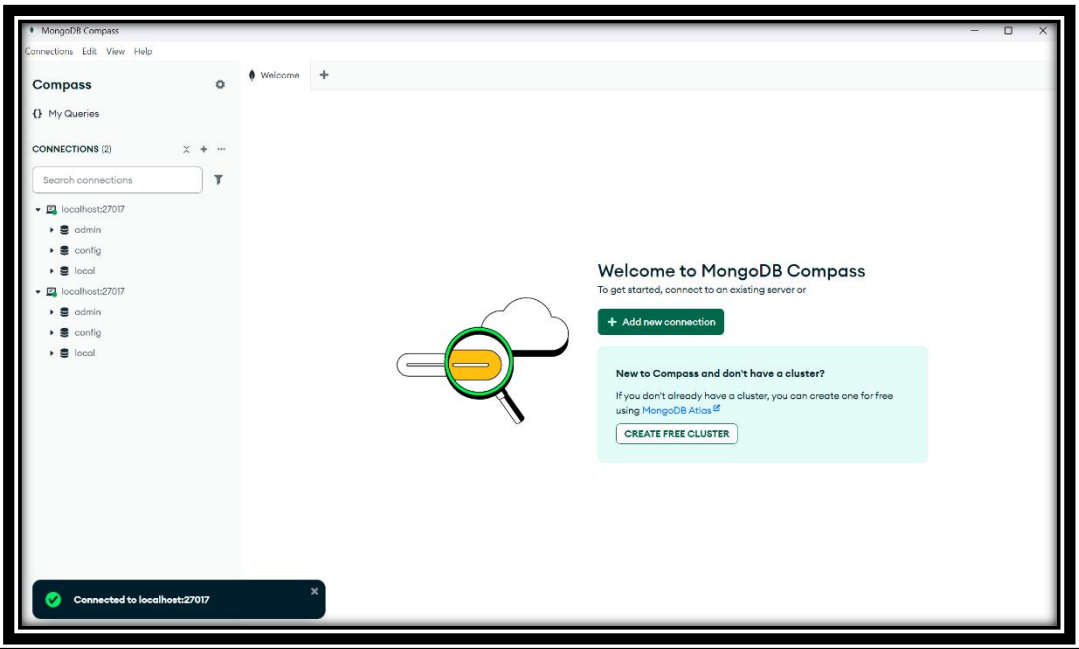
KUNSH SABHARWAL

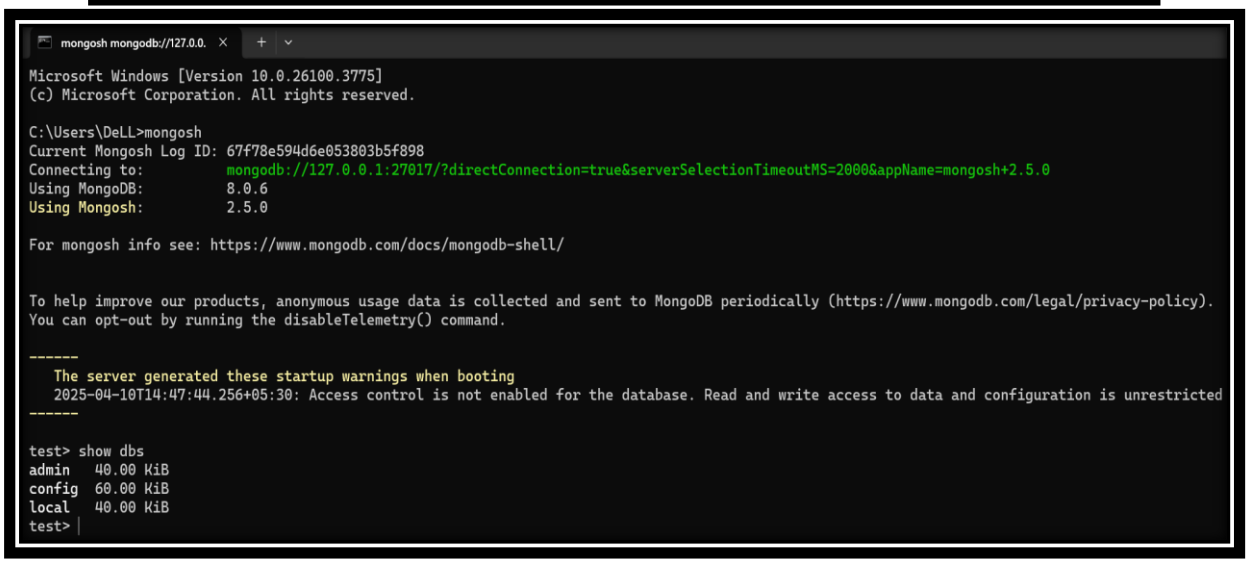












Learning Outcome:

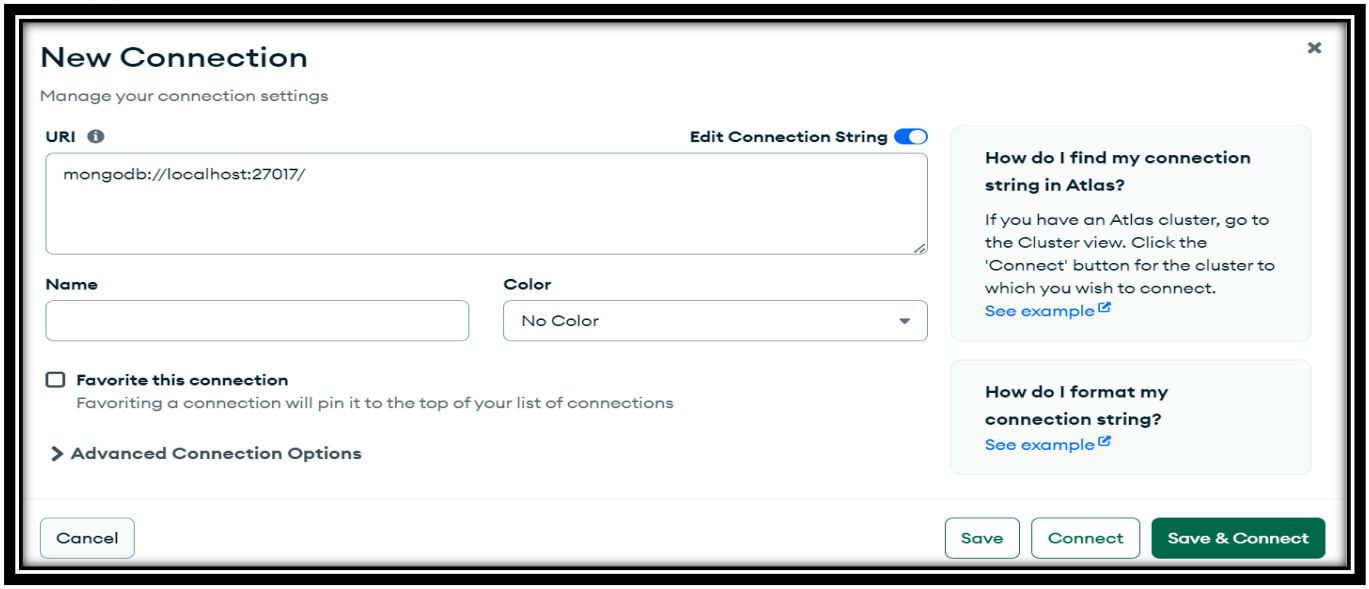
Beyond Curriculum
EXPERIMENT 2

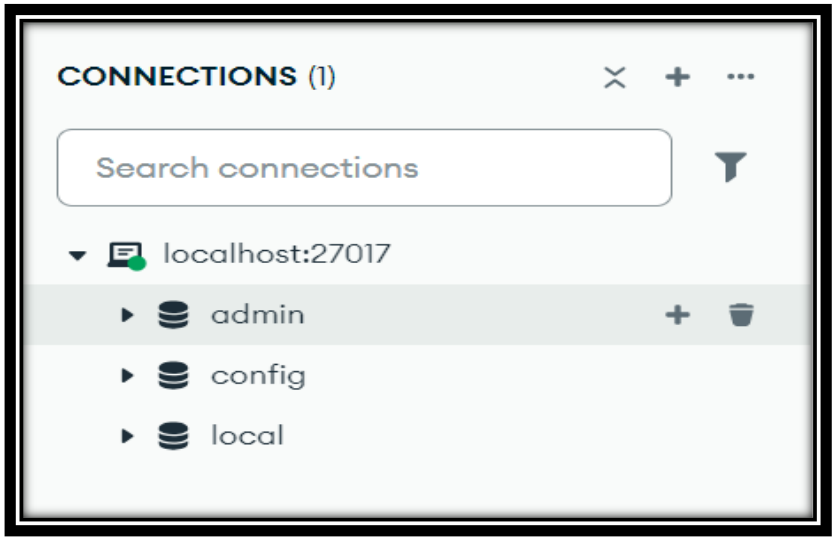
Problem statement: Study and implement basic commands of MongoDB

Theory:

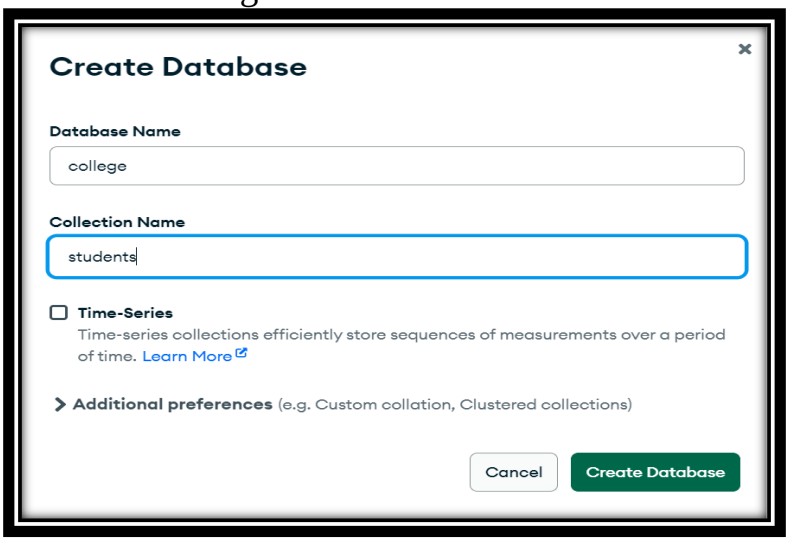
MongoDB Compass: -

(a) Connect to the runtime: -localhost:27017.

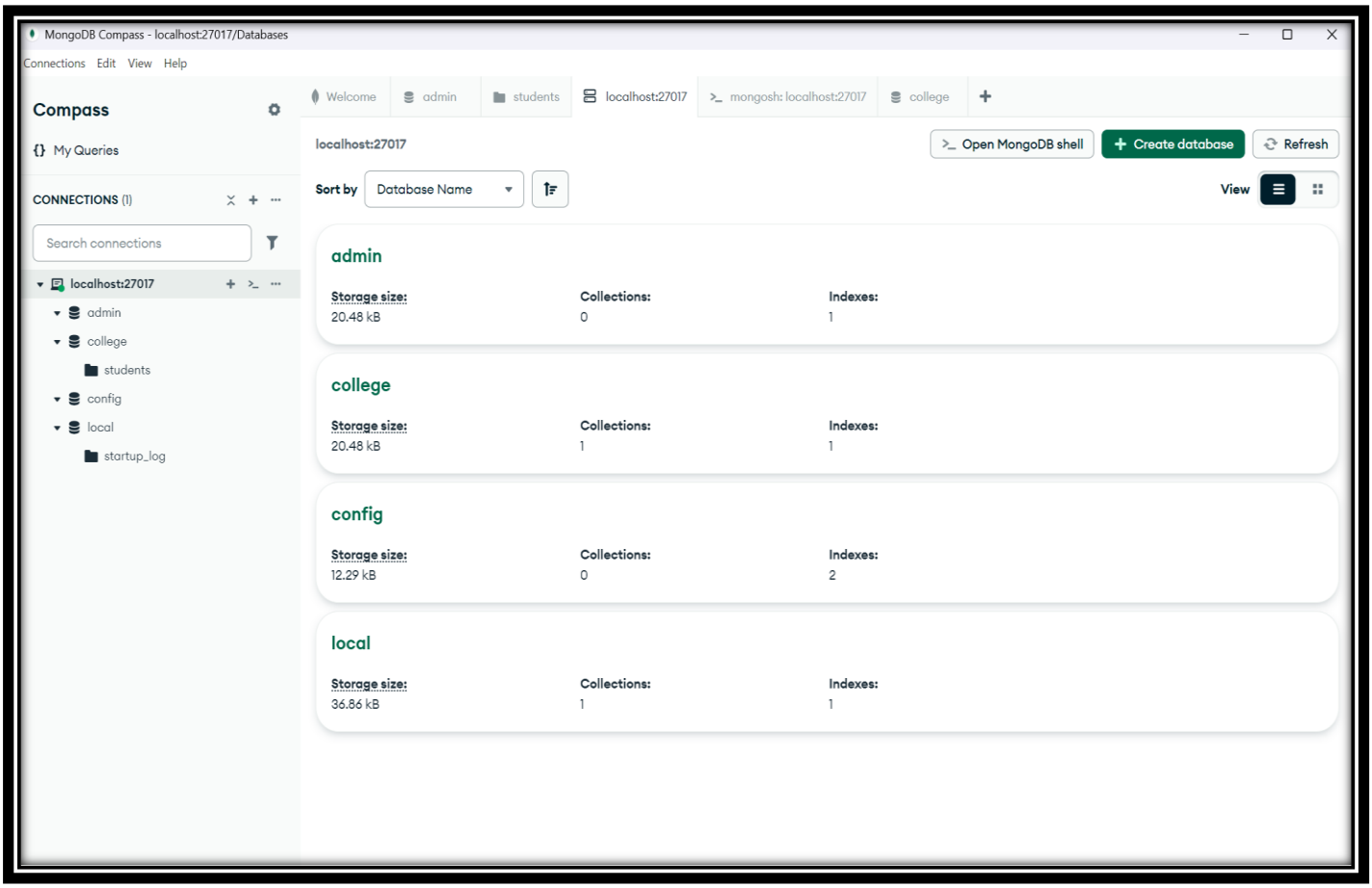




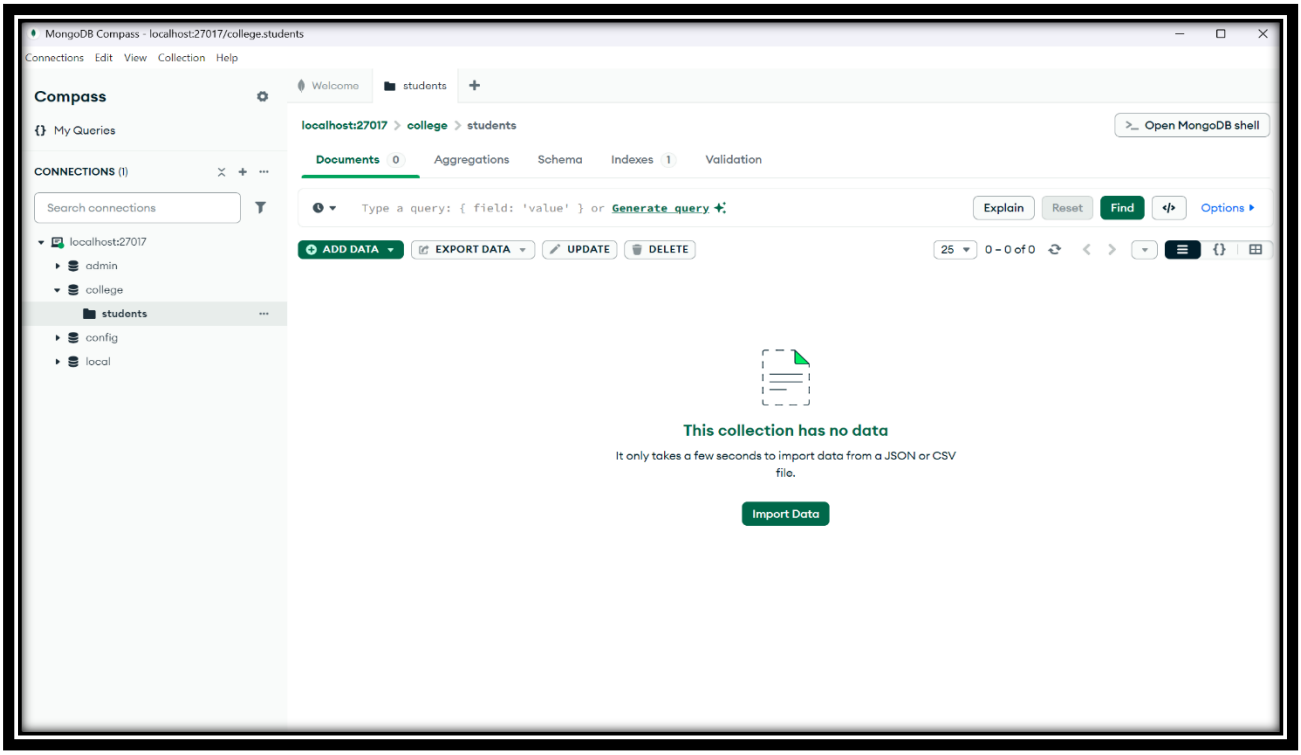
(b) We see three existing databases named – admin, config and local. Create a new database with name college and in it a collection named students.



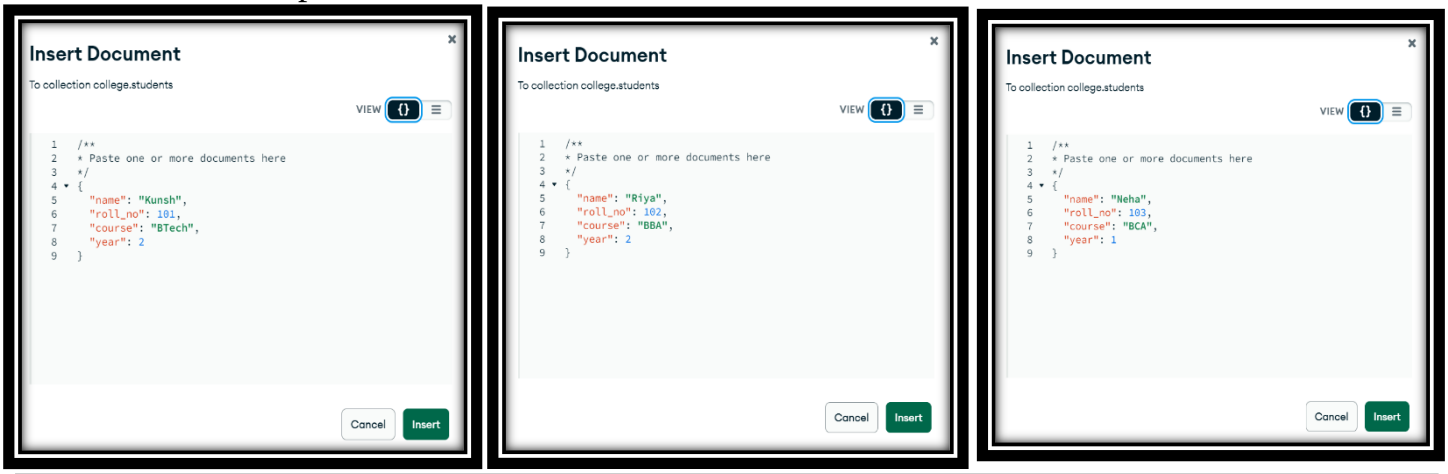
(c) View all the databases in compass's GUI by clicking on the connections name i.e. localhost:27017

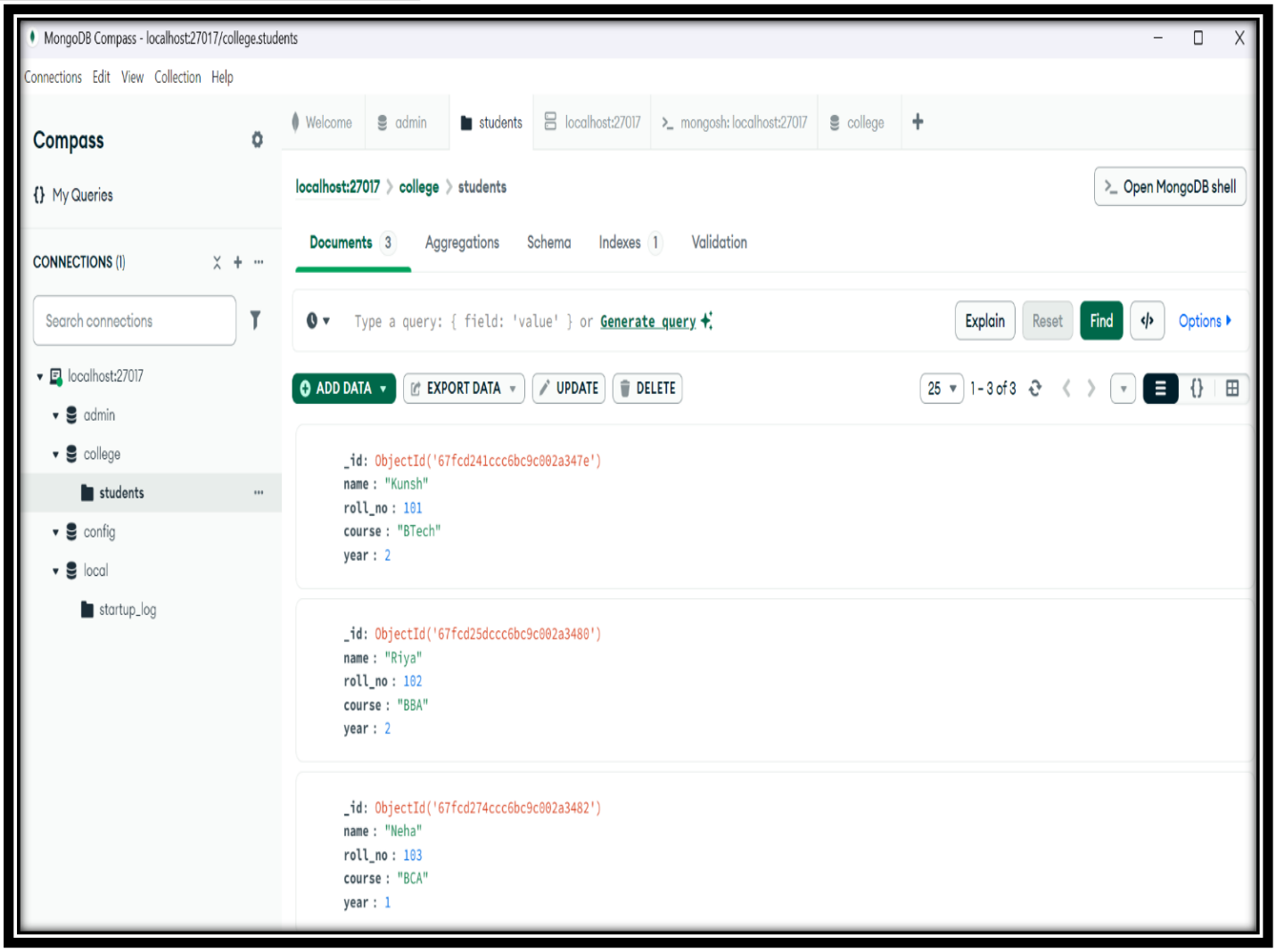


(d) On visiting the college database and the collection student we enter the documents tab which is empty by default.



(e) Insert values in the documents tab by clicking on add data → insert document and keep changing the value you want to insert there and click insert after each update to add that value.





(f) After inserting the values, we can click on the button in the top right of the MongoDB Compass GUI – >_ Open MongoDB shell. After clicking, a navy-blue colour screen appears where we may type and execute some basic queries related to the data that we just inserted.

Commands in MongoDB shell: -

```
>_MONGOSH
> show dbs;
< admin      40.00 KiB
  college    72.00 KiB
  config    108.00 KiB
  local      72.00 KiB
> use college;
< switched to db college
> show collections;
< students
```

```
>_MONGOSH
> db["students"].find()
< {
    _id: ObjectId('67fcd241ccc6bc9c002a347e'),
    name: 'Kunsh',
    roll_no: 101,
    course: 'BTech',
    year: 2
  }
  {
    _id: ObjectId('67fcd25dccc6bc9c002a3480'),
    name: 'Riya',
    roll_no: 102,
    course: 'BBA',
    year: 2
  }
  {
    _id: ObjectId('67fcd274ccc6bc9c002a3482'),
    name: 'Neha',
    roll_no: 103,
    course: 'BCA',
    year: 1
  }
```

```
>_MONGOSH
> db["students"].find().pretty();
< {
    _id: ObjectId('67fcd241ccc6bc9c002a347e'),
    name: 'Kunsh',
    roll_no: 101,
    course: 'BTech',
    year: 2
  }
  {
    _id: ObjectId('67fcd25dccc6bc9c002a3480'),
    name: 'Riya',
    roll_no: 102,
    course: 'BBA',
    year: 2
  }
  {
    _id: ObjectId('67fcd274ccc6bc9c002a3482'),
    name: 'Neha',
    roll_no: 103,
    course: 'BCA',
    year: 1
  }
```

KUNSH SABHARWAL

```
>_MONGOSH
> db.students.insertOne({
    "name": "Amit",
    "roll_no": 104,
    "course": "MBA",
    "year": 1
  });
< {
    acknowledged: true,
    insertedId: ObjectId('67fcd3eec60422e72b0fdedc')
  }
> db.students.updateOne(
    { "name": "Kunsh" },
    { $set: { "course": "MTech" } }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.students.deleteOne({ "name": "Neha" });
< {
    acknowledged: true,
    deletedCount: 1
  }
> db.students.countDocuments();
< 3
```

```
>_MONGOSH
> db.students.find({ "year": 2 }).pretty();
< {
    _id: ObjectId('67fcd241ccc6bc9c002a347e'),
    name: 'Kunsh',
    roll_no: 101,
    course: 'MTech',
    year: 2
  }
  {
    _id: ObjectId('67fcd25dccc6bc9c002a3480'),
    name: 'Riya',
    roll_no: 102,
    course: 'BBA',
    year: 2
  }
```

```
>_MONGOSH
> db.students.find().sort({ "roll_no": 1 }).pretty();
< {
    _id: ObjectId('67fcd241ccc6bc9c002a347e'),
    name: 'Kunsh',
    roll_no: 101,
    course: 'MTech',
    year: 2
  }
  {
    _id: ObjectId('67fcd25dccc6bc9c002a3480'),
    name: 'Riya',
    roll_no: 102,
    course: 'BBA',
    year: 2
  }
  {
    _id: ObjectId('67fcd3eec60422e72b0fdedc'),
    name: 'Amit',
    roll_no: 104,
    course: 'MBA',
    year: 1
  }
```

```
> db.students.find();
< {
    _id: ObjectId('67fcd241ccc6bc9c002a347e'),
    name: 'Kunsh',
    roll_no: 101,
    course: 'MTech',
    year: 2
  }
  {
    _id: ObjectId('67fcd25dccc6bc9c002a3480'),
    name: 'Riya',
    roll_no: 102,
    course: 'BBA',
    year: 2
  }
  {
    _id: ObjectId('67fcd3eec60422e72b0fdedc'),
    name: 'Amit',
    roll_no: 104,
    course: 'MBA',
    year: 1
  }
college>
```

Learning Outcome: -