3. Practice SQL Data Definition Language (DDL) Commands

3.1. Table creation and alteration

1. Create all 5 tables based on the schema provided.

Table Name: Departments

Column Name	Data Type	Constraints
Dept_ID	VARCHAR2(4)	PRIMARY KEY, starts with 'D'
Dept_Name	VARCHAR2(50)	NOT NULL, UNIQUE
Building	VARCHAR2(30)	
Number_of_Classrooms	NUMBER(3)	CHECK(Number_of_Classrooms >= 0)

CODE:

```
CREATE TABLE Departments (
Dept_ID VARCHAR2(4) PRIMARY KEY CHECK (Dept_ID LIKE 'D%'),
Dept_Name VARCHAR2(50) NOT NULL UNIQUE,
Building VARCHAR2(30),
Number_of_Classrooms NUMBER(3) CHECK (Number_of_Classrooms >= 0) );
```

OUTPUT:

```
SQL> CREATE TABLE Departments (
  2  Dept_ID VARCHAR2(4) PRIMARY KEY CHECK (Dept_ID LIKE 'D%'),
  3  Dept_Name VARCHAR2(50) NOT NULL UNIQUE,
  4  Building VARCHAR2(30),
  5  Number_of_Classrooms NUMBER(3) CHECK (Number_of_Classrooms >= 0)
  6  );

Table created.
```

Table Name: Professors

Column Name	Data Type	Constraints
Prof_ID	VARCHAR2(5)	PRIMARY KEY, starts with 'P'
Prof_Name	VARCHAR2(50)	NOT NULL
Dept_ID	VARCHAR2(4)	FOREIGN KEY REFERENCES Departments(Dept_ID)
Experience_Years	NUMBER(2)	CHECK (Experience_Years >= 0)

CODE:

```
CREATE TABLE Professors
(Prof_ID VARCHAR2(5) PRIMARY KEY CHECK(Prof_ID LIKE 'P%'),
Prof_Name VARCHAR2(50) NOT NULL ,
Dept_ID VARCHAR2(4) REFERENCES Departments(Dept_ID),
Experience_Years NUMBER(2) CHECK(Experience_Years>=0) );
```

OUTPUT:

```
SQL> CREATE TABLE Professors
  2  (Prof_ID VARCHAR2(5) PRIMARY KEY CHECK(Prof_ID LIKE 'P%'),
  3  Prof_Name VARCHAR2(50) NOT NULL ,
  4  Dept_ID VARCHAR2(4) REFERENCES Departments(Dept_ID),
  5  Experience_Years NUMBER(2) CHECK(Experience_Years>=0) );

Table created.
```

Table Name: Courses

Column Name	Data Type	Constraints
Course_ID	VARCHAR2(6)	PRIMARY KEY
Course_Name	VARCHAR2(100)	NOT NULL
Dept_ID	VARCHAR2(4)	FOREIGN KEY REFERENCES Departments(Dept_ID)
Prof_ID	VARCHAR2(5)	FOREIGN KEY REFERENCES Professors(Prof_ID)
Credits	NUMBER(1)	CHECK (Credits BETWEEN 1 AND 5)
Student_Count	NUMBER(4)	CHECK (Student_Count >= 0)

CODE:

```
CREATE TABLE Courses
(Course_ID VARCHAR2(6) PRIMARY KEY ,
Course_Name VARCHAR2(100) NOT NULL ,
Dept_ID VARCHAR2(4) REFERENCES Departments(Dept_ID),
Prof_ID VARCHAR2(5) REFERENCES Professors(Prof_ID),
```

Credits NUMBER(1) CHECK(Credits BETWEEN 1 AND 5),

Student_Count NUMBER(4) CHECK(Student_Count>=0));

OUTPUT:

```
SQL> CREATE TABLE Courses
  2  (Course_ID VARCHAR2(6) PRIMARY KEY ,
  3  Course_Name VARCHAR2(100) NOT NULL ,
  4  Dept_ID VARCHAR2(4) REFERENCES Departments(Dept_ID),
  5  Prof_ID VARCHAR2(5) REFERENCES Professors(Prof_ID),
  6  Credits NUMBER(1) CHECK(Credits BETWEEN 1 AND 5),
  7  Student_Count NUMBER(4) CHECK(Student_Count>=0) );

Table created.
```

Table Name: Students

Column Name	**Data Type**	**Constraints**
Student_ID	VARCHAR2(6)	PRIMARY KEY, starts with 'S'
Student_Name	VARCHAR2(50)	NOT NULL
Dept_ID	VARCHAR2(4)	FOREIGN KEY REFERENCES Departments(Dept_ID)
DOB	DATE	

CODE:

CREATE TABLE Students

(Student_ID VARCHAR2(6) PRIMARY KEY CHECK(Student_ID LIKE 'S%'),

Student_Name VARCHAR2(50) NOT NULL ,

Dept_ID VARCHAR2(4) REFERENCES Departments(Dept_ID),

DOB DATE);

OUTPUT:

```
SQL> CREATE TABLE Students
  2  (Student_ID VARCHAR2(6) PRIMARY KEY CHECK(Student_ID LIKE 'S%'),
  3  Student_Name VARCHAR2(50) NOT NULL ,
  4  Dept_ID VARCHAR2(4) REFERENCES Departments(Dept_ID),
  5  DOB DATE );

Table created.
```

Table Name: Enrollments

Column Name	Data Type	Constraints
Student_ID	VARCHAR2(6)	FOREIGN KEY REFERENCES Students(Student_ID)
Course_ID	VARCHAR2(6)	FOREIGN KEY REFERENCES Courses(Course_ID)
Semester	VARCHAR2(6)	e.g., 'Sem1', 'Sem2'
Marks	NUMBER(5,2)	CHECK (MARKS >= 0 AND MARKS < 100)
PRIMARY KEY	(Student_ID, Course_ID)	

CODE:

```
CREATE TABLE Enrollments
(Student_ID VARCHAR2(6) REFERENCES Students(Student_ID),
Course_ID VARCHAR2(6)  REFERENCES Courses(Course_ID),
Semester VARCHAR2(6) CHECK(Semester LIKE 'Sem%'),
Marks NUMBER(5,2) CHECK(Marks >= 0 AND Marks < 100),
PRIMARY KEY (Student_ID,Course_ID) );
```

OUTPUT:

```
SQL> CREATE TABLE Enrollments
  2  (Student_ID VARCHAR2(6) REFERENCES Students(Student_ID),
  3  Course_ID VARCHAR2(6)  REFERENCES Courses(Course_ID),
  4  Semester VARCHAR2(6) CHECK(Semester LIKE 'Sem%'),
  5  Marks NUMBER(5,2) CHECK(Marks >= 0 AND Marks < 100),
  6  PRIMARY KEY (Student_ID,Course_ID) );

Table created.
```

2. Add a column Dept_Head (varchar2(50)) to Departments.

CODE:

```
ALTER TABLE Departments
ADD Dept_Head VARCHAR2(50);
```

OUTPUT:

```
SQL> ALTER TABLE Departments
  2  ADD Dept_Head VARCHAR2(50);

Table altered.
```

```
SQL> DESC DEPARTMENTS;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPT_ID                                   NOT NULL VARCHAR2(4)
 DEPT_NAME                                 NOT NULL VARCHAR2(50)
 BUILDING                                           VARCHAR2(30)
 NUMBER_OF_CLASSROOMS                               NUMBER(3)
 DEPT_HEAD                                          VARCHAR2(50)
```

3. Change size of Experience_Years in Professors to NUMBER(3).

CODE:

ALTER TABLE Professors

MODIFY(Experience_Years NUMBER(3));

OUTPUT:

```
SQL> ALTER TABLE Professors
  2  MODIFY(Experience_Years NUMBER(3) );

Table altered.
```

```
SQL> DESC PROFESSORS
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PROF_ID                                   NOT NULL VARCHAR2(5)
 PROF_NAME                                 NOT NULL VARCHAR2(50)
 DEPT_ID                                            VARCHAR2(4)
 EXPERIENCE_YEARS                                   NUMBER(3)
```

4. Rename Student_Count to Total_Students in Courses.

CODE:

ALTER TABLE Courses

RENAME COLUMN Student_Count TO Total_Students;

OUTPUT:

```
SQL> ALTER TABLE Courses
  2  RENAME COLUMN Student_Count TO Total_Students;

Table altered.
```

```
SQL> DESC Courses;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COURSE_ID                                 NOT NULL VARCHAR2(6)
 COURSE_NAME                               NOT NULL VARCHAR2(100)
 DEPT_ID                                            VARCHAR2(4)
 PROF_ID                                            VARCHAR2(5)
 CREDITS                                            NUMBER(1)
 TOTAL_STUDENTS                                     NUMBER(4)
```

5. Drop and recreate the Enrollments table.

CODE:

```
DROP TABLE Enrollments;

CREATE TABLE Enrollments

(Student_ID VARCHAR2(6) REFERENCES Students(Student_ID),

Course_ID VARCHAR2(6)  REFERENCES Courses(Course_ID),

Semester VARCHAR2(6) CHECK(Semester LIKE 'Sem%'),

Marks NUMBER(5,2) CHECK(Marks >= 0 AND Marks < 100),

PRIMARY KEY (Student_ID,Course_ID) );
```

OUTPUT:

```
SQL> DROP TABLE Enrollments;

Table dropped.

SQL> CREATE TABLE Enrollments
  2  (Student_ID VARCHAR2(6) REFERENCES Students(Student_ID),
  3  Course_ID VARCHAR2(6)  REFERENCES Courses(Course_ID),
  4  Semester VARCHAR2(6) CHECK(Semester LIKE 'Sem%'),
  5  Marks NUMBER(5,2) CHECK(Marks >= 0 AND Marks < 100),
  6  PRIMARY KEY (Student_ID,Course_ID) );

Table created.
```

4. Practice SQL Data Manipulation Language (DML) Commands

4.1 Insertion, Deletion, Update

1. Insert the sample data into all five tables.

Departments:

Dept_ID	Dept_Name	Building	Number_of_Classrooms
D01	Computer Science	Tech Block	10
D02	Electrical Engg.	Power House	8
D03	Mechanical Engg.	Mech Block	6

CODE:

```
INSERT INTO Departments VALUES('D01','Computer Science','Tech Block',10,NULL);

INSERT INTO Departments VALUES('D02', 'Electrical Engg.','Power House',8,NULL);

INSERT INTO Departments VALUES('D03', 'Mechanical Engg.','Mech Block',6,NULL);
```

OUTPUT:

```
SQL> SELECT * FROM Departments;

DEPT_I DEPT_NAME                    BUILDING            NUMBER_OF_CLASSROOMS DEPT_HEAD
------ ------------------------------ -------------------- -------------------- -------------
D01    Computer Science             Tech Block                         10
D02    Electrical Engg.             Power House                         8
D03    Mechanical Engg.             Mech Block                          6
```

Professors:

Prof_ID	Prof_Name	Dept_ID	Experience_Years
P1001	Dr. Meera Nair	D01	12
P1002	Dr. Arjun Rao	D02	9
P1003	Dr. Kavita Singh	D01	7
P1004	Dr. Raj Malhotra	D03	15

CODE:

```
INSERT INTO Professors VALUES('P1001', 'Dr. Meera Nair','D01',12);

INSERT INTO Professors VALUES('P1002', 'Dr. Arjun Rao','D01',9);

INSERT INTO Professors VALUES('P1003', 'Dr. Kavita Singh','D01',7);
```

INSERT INTO Professors VALUES('P1004', 'Dr. Raj Malhotra','D01',15);

OUTPUT:

```
SQL> SELECT * FROM Professors;

PROF_ PROF_NAME                                           DEPT_I EXPERIENCE_YEARS
----- ------------------------------------------------- ------ ----------------
P1001 Dr. Meera Nair                                    D01                  12
P1002 Dr. Arjun Rao                                     D01                   9
P1003 Dr. Kavita Singh                                  D01                   7
P1004 Dr. Raj Malhotra                                  D01                  15
```

Courses:

Course_ID	Course_Name	Dept_ID	Prof_ID	Credits	Student_Count
CSE101	Data Structures	D01	P1001	4	2
CSE201	Operating Systems	D01	P1003	3	1
EEE101	Circuit Theory	D02	P1002	4	1
ME101	Thermodynamics	D03	P1004	3	1

CODE:

INSERT INTO Courses VALUES('CSE101','Data Structures','D01', 'P1001',4,2);

INSERT INTO Courses VALUES('CSE201','Operating Systems','D01', 'P1003',3,1);

INSERT INTO Courses VALUES('EEE101','Circuit Theory','D02', 'P1002',4,1);

INSERT INTO Courses VALUES('ME101','Thermodynamics','D03', 'P1004',3,1);

OUTPUT:

```
SQL> SELECT * FROM Courses;

COURSE_ COURSE_NAME                  DEPT_ PROF_I CREDITS TOTAL_STUDENTS
------- ---------------------------- ----- ------ ------- --------------
CSE101  Data Structures              D01   P1001        4              2
CSE201  Operating Systems            D01   P1003        3              1
EEE101  Circuit Theory               D02   P1002        4              1
ME101   Thermodynamics               D03   P1004        3              1
```

Students:

Student_ID	Student_Name	Dept_ID	DOB
S0001	Anjali Sharma	D01	2003-05-14

S0002	Ravi Kumar	D02	2002-11-20
S0003	Nisha Verma	D03	2003-02-02
S0004	Aman Sheikh	D01	2002-07-25

CODE:

```
INSERT INTO Students VALUES('S0001','Anjali Sharma','D01',TO_DATE('2003-05-14',
'YYYY-MM-DD'));

INSERT INTO Students VALUES('S0002',' Ravi Kumar','D02',TO_DATE('2002-11-20',
'YYYY-MM-DD'));

INSERT INTO Students VALUES('S0003',' Nisha Verma','D03',TO_DATE('2003-02-02',
'YYYY-MM-DD'));

INSERT INTO Students VALUES('S0004','Aman Sheikh','D01',TO_DATE('2002-07-25',
'YYYY-MM-DD'));
```

OUTPUT:

```
SQL> SELECT * FROM Students;

STUDEN STUDENT_NAME                                       DEPT_ DOB
------ -------------------------------------------------- ----- ---------
S0001  Anjali Sharma                                      D01   14-MAY-03
S0002   Ravi Kumar                                        D02   20-NOV-02
S0003   Nisha Verma                                       D03   02-FEB-03
S0004  Aman Sheikh                                        D01   25-JUL-02
```

Enrollments:

Student_ID	**Course_ID**	**Semester**	**Marks**
S0001	CSE101	Sem1	88.0
S0001	CSE201	Sem2	76.5
S0002	EEE101	Sem1	81.0
S0003	ME101	Sem1	93.0
S0004	CSE101	Sem1	68.5

CODE:

```
INSERT INTO Enrollments VALUES('S0001','CSE101','Sem1',88.0);

INSERT INTO Enrollments VALUES('S0001','CSE201','Sem2',76.5);

INSERT INTO Enrollments VALUES('S0002','EEE101','Sem1',81.0);
```

INSERT INTO Enrollments VALUES('S0003','ME101','Sem1',93.0);

INSERT INTO Enrollments VALUES('S0004','CSE101','Sem1',68.5);

OUTPUT:

```
SQL> SELECT * FROM Enrollments;

STUDEN COURSE_ SEMEST      MARKS
------ ------- ------ ----------
S0001  CSE101  Sem1           88
S0001  CSE201  Sem2         76.5
S0002  EEE101  Sem1           81
S0003  ME101   Sem1           93
S0004  CSE101  Sem1         68.5
```

2. Insert a new department: (‘D04’, ‘Civil Engg’, ‘Block C’, 5).

CODE:

INSERT INTO Departments VALUES ('D04', 'Civil Engg', 'Block C', 5,NULL);

OUTPUT:

```
SQL> SELECT * FROM Departments;

DEPT_ DEPT_NAME                BUILDING        NUMBER_OF_CLASSROOMS DEPT_HEAD
----- ------------------------ --------------- -------------------- -----------
D01   Computer Science         Tech Block                       10
D02   Electrical Engg.         Power House                       8
D03   Mechanical Engg.         Mech Block                        6
D04   Civil Engg               Block C                           5
```

3. Create a new table high_achievers containing students who scored more than 85 in any course.

CODE:

CREATE TABLE high_achievers AS

SELECT Student_Name FROM Students

WHERE Student_ID IN

(SELECT Student_ID FROM Enrollments

WHERE marks>85);

OUTPUT:

```
SQL> CREATE TABLE high_achievers AS
  2  SELECT Student_Name FROM Students
  3  WHERE Student_ID IN
  4  (SELECT Student_ID FROM Enrollments
  5  WHERE marks>85);

Table created.
```

4. Create a backup table Courses_Backup with all data from Courses and Professors_Backup with all the data from professors.

CODE:

CREATE TABLE Courses_Backup AS SELECT * FROM Courses;

CREATE TABLE Professors_Backup AS SELECT * FROM Professors;

OUTPUT:

```
SQL> CREATE TABLE Courses_Backup AS SELECT * FROM Courses;

Table created.

SQL> SELECT * FROM Courses_Backup;

COURSE_I COURSE_NAME                    DEPT_ PROF_I CREDITS TOTAL_STUDENTS
-------- ------------------------------ ----- ------ ------- --------------
CSE101   Data Structures                D01   P1001        4              2
CSE201   Operating Systems              D01   P1003        3              1
EEE101   Circuit Theory                D02   P1002        4              1
ME101    Thermodynamics                 D03   P1004        3              1
```

```
SQL> CREATE TABLE Professors_Backup AS SELECT * FROM Professors;

Table created.

SQL> SELECT * FROM Professors_Backup;

PROF_I PROF_NAME                                          DEPT_ EXPERIENCE_YEARS
------ -------------------------------------------------- ----- ----------------
P1001  Dr. Meera Nair                                     D01                 12
P1002  Dr. Arjun Rao                                      D01                  9
P1003  Dr. Kavita Singh                                   D01                  7
P1004  Dr. Raj Malhotra                                   D01                 15
```

5. Add a new course ‘CIV101’, ‘Structural Analysis’, under D04, taught by P1001, with 3 credits and 0 students.

CODE:

INSERT INTO Courses VALUES('CIV101','Structural Analysis','D04', 'P1001',3,0);

OUTPUT:

```
SQL> SELECT * FROM Courses;

COURSE_I COURSE_NAME                    DEPT_ PROF_I CREDITS TOTAL_STUDENTS
-------- ------------------------------ ----- ------ ------- --------------
CIV101   Structural Analysis            D04   P1001        3              0
CSE101   Data Structures                D01   P1001        4              2
CSE201   Operating Systems              D01   P1003        3              1
EEE101   Circuit Theory                 D02   P1002        4              1
ME101    Thermodynamics                 D03   P1004        3              1
```

6. Update professor of 'CIV101' to P1004 and savepoint SP1.

CODE:

```
UPDATE Courses SET Prof_ID = 'P1004' WHERE Course_ID = 'CIV101';

SAVEPOINT SP1;
```

OUTPUT:

```
SQL> UPDATE Courses SET Prof_ID = 'P1004' WHERE Course_ID = 'CIV101';

1 row updated.

SQL> SELECT * FROM Courses;

COURSE_I COURSE_NAME                    DEPT_ PROF_I CREDITS TOTAL_STUDENTS
-------- ------------------------------ ----- ------ ------- --------------
CIV101   Structural Analysis            D04   P1004        3              0
CSE101   Data Structures                D01   P1001        4              2
CSE201   Operating Systems              D01   P1003        3              1
EEE101   Circuit Theory                 D02   P1002        4              1
ME101    Thermodynamics                 D03   P1004        3              1

SQL> SAVEPOINT SP1;

Savepoint created.
```

7. Change credits of 'CSE201' to 4 and set savepoint SP2.

CODE:

```
UPDATE Courses SET Credits = 4 WHERE Course_ID = 'CSE201';

SAVEPOINT SP2;
```

OUTPUT:

```
SQL> UPDATE Courses SET Credits = 4 WHERE Course_ID = 'CSE201';

1 row updated.

SQL> SELECT * FROM Courses;

COURSE_I COURSE_NAME                    DEPT_ PROF_I CREDITS TOTAL_STUDENTS
-------- ------------------------------ ----- ------ ------- --------------
CIV101   Structural Analysis            D04   P1004        3              0
CSE101   Data Structures                D01   P1001        4              2
CSE201   Operating Systems              D01   P1003        4              1
EEE101   Circuit Theory                 D02   P1002        4              1
ME101    Thermodynamics                 D03   P1004        3              1

SQL> SAVEPOINT SP2;

Savepoint created.
```

8. Delete all courses from Courses_Backup that have less than 4 credits.

CODE:

DELETE FROM Courses_Backup WHERE Credits < 4;

OUTPUT:

```
SQL> DELETE FROM Courses_Backup WHERE Credits < 4;

2 rows deleted.
```

9. Delete all professors from Professors_Backup with less than 10 years experience.

CODE:

DELETE FROM Professors_Backup WHERE Experience_Years < 10;

OUTPUT:

```
SQL> DELETE FROM Professors_Backup WHERE Experience_Years < 10;

2 rows deleted.
```

10. Rollback to SP1 and rename Courses_Backup to Course_Master.

CODE:

ROLLBACK TO SP1;

ALTER TABLE Course_Backup RENAME TO Course_Master;

OUTPUT:

```
SQL> ROLLBACK TO SP1;

Rollback complete.
```

```
SQL> ALTER TABLE Course_Backup RENAME TO Course_Master;

Table altered.
```

4.2 Data Retrieval (SELECT)

1. List all department names.

CODE:

SELECT Dept_Name FROM Departments;

OUTPUT:

```
SQL> SELECT Dept_Name FROM Departments;

DEPT_NAME
--------------------------------------------------
Civil Engg
Computer Science
Electrical Engg.
Mechanical Engg.
```

2. Display all data from the Professors table.

CODE:

SELECT * FROM Professors;

OUTPUT:

```
SQL> SELECT * FROM Professors;

PROF_I PROF_NAME                                      DEPT_ EXPERIENCE_YEARS
------ ---------------------------------------------- ----- ----------------
P1001  Dr. Meera Nair                                 D01                 12
P1002  Dr. Arjun Rao                                  D01                  9
P1003  Dr. Kavita Singh                               D01                  7
P1004  Dr. Raj Malhotra                               D01                 15
```

3. List student names and DOBs.

CODE:

SELECT Student_Name, DOB FROM Students;

OUTPUT:

```
SQL> SELECT Student_Name, DOB FROM Students;

STUDENT_NAME                                       DOB
-------------------------------------------------- ---------
Anjali Sharma                                      14-MAY-03
 Ravi Kumar                                        20-NOV-02
 Nisha Verma                                       02-FEB-03
Aman Sheikh                                        25-JUL-02
```

4. List course names and credits.

CODE:

SELECT Course_Name, Credits FROM Courses;

OUTPUT:

```
SQL> SELECT Course_Name, Credits FROM Courses;

COURSE_NAME                    CREDITS
------------------------------ -------
Structural Analysis                  3
Data Structures                      4
Operating Systems                    4
Circuit Theory                       4
Thermodynamics                       3
```

5. Get courses offered by the 'Computer Science' department.

CODE:

SELECT * FROM Courses WHERE Dept_ID IN

(SELECT Dept_ID FROM Departments WHERE Dept_Name = 'Computer Science');

OUTPUT:

```
SQL> SELECT *  FROM Courses WHERE Dept_ID IN
  2  (SELECT Dept_ID FROM Departments WHERE Dept_Name = 'Computer Science');

COURSE COURSE_NAME              DEPT_ PROF_ID    CREDITS TOTAL_STUDENTS
------ ------------------------ ----- ------- ---------- --------------
CSE101 Data Structures          D01   P1001            4              2
CSE201 Operating Systems        D01   P1003            4              1
```

6. List professors whose name starts with ‘Dr’.

CODE:

SELECT * FROM Professors WHERE Prof_Name LIKE 'Dr%';

OUTPUT:

```
SQL> SELECT * FROM Professors WHERE Prof_Name LIKE 'Dr%';

PROF_I PROF_NAME                                          DEPT_ EXPERIENCE_YEARS
------ -------------------------------------------------- ----- ----------------
P1001  Dr. Meera Nair                                     D01                 12
P1002  Dr. Arjun Rao                                      D01                  9
P1003  Dr. Kavita Singh                                   D01                  7
P1004  Dr. Raj Malhotra                                   D01                 15
```

7. List courses with credits more than 3.

CODE:

SELECT * FROM Courses WHERE Credits > 3;

OUTPUT:

```
SQL> SELECT * FROM Courses WHERE Credits > 3;

COURSE_I COURSE_NAME                 DEPT_ PROF_I CREDITS TOTAL_STUDENTS
-------- --------------------------- ----- ------ ------- --------------
CSE101   Data Structures             D01   P1001        4              2
CSE201   Operating Systems           D01   P1003        4              1
EEE101   Circuit Theory             D02   P1002        4              1
```

8. Display all courses with “Theory” in their name.

CODE:

SELECT * FROM Courses WHERE Course_Name LIKE '%Theory%';

OUTPUT:

```
SQL> SELECT * FROM Courses WHERE Course_Name LIKE '%Theory%';

COURSE_I COURSE_NAME                 DEPT_ PROF_I CREDITS TOTAL_STUDENTS
-------- --------------------------- ----- ------ ------- --------------
EEE101   Circuit Theory              D02   P1002        4              1
```

9. List students born after Jan 1, 2003.

CODE:

SELECT * FROM Students WHERE DOB > TO_DATE('01-01-2003', 'DD-MM-YYYY');

OUTPUT:

```
SQL> SELECT * FROM Students WHERE DOB > TO_DATE('01-01-2003', 'DD-MM-YYYY');

STUDEN STUDENT_NAME                                       DEPT_ DOB
------ -------------------------------------------------- ----- ---------
S0001  Anjali Sharma                                      D01   14-MAY-03
S0003   Nisha Verma                                       D03   02-FEB-03
```

10. Find all professors with 10+ years experience.

CODE:

SELECT * FROM Professors WHERE Experience_Years >= 10;

OUTPUT:

```
SQL> SELECT * FROM Professors WHERE Experience_Years >= 10;

PROF_I PROF_NAME                                          DEPT_ EXPERIENCE_YEARS
------ -------------------------------------------------- ----- ----------------
P1001  Dr. Meera Nair                                     D01                 12
P1004  Dr. Raj Malhotra                                   D01                 15
```

11. List all courses with more than 1 student.

CODE:

SELECT * FROM Courses WHERE Total_Students > 1;

OUTPUT:

```
SQL> SELECT * FROM Courses WHERE Total_Students > 1;

COURSE_I COURSE_NAME                    DEPT_ PROF_I CREDITS TOTAL_STUDENTS
-------- ------------------------------ ----- ------ ------- --------------
CSE101   Data Structures                D01   P1001        4              2
```

12. Display all distinct semesters from Enrollments.

CODE:

SELECT DISTINCT Semester FROM Enrollments;

OUTPUT:

```
SQL> SELECT DISTINCT Semester FROM Enrollments;

SEMEST
------
Sem1
Sem2
```

13. Show information of students with ID S0001, S0002.

CODE:

SELECT * FROM Students WHERE Student_ID IN ('S0001', 'S0002');

OUTPUT:

```
SQL> SELECT * FROM Students WHERE Student_ID IN ('S0001', 'S0002');

STUDEN STUDENT_NAME                                       DEPT_ DOB
------ -------------------------------------------------- ----- ---------
S0001  Anjali Sharma                                      D01   14-MAY-03
S0002   Ravi Kumar                                        D02   20-NOV-02
```

14. Show all courses not in the 'Mechanical Engg' department.

CODE:

SELECT * FROM Courses WHERE Dept_ID NOT IN

(SELECT Dept_ID FROM Departments WHERE Dept_Name = 'Mechanical Engg.');

OUTPUT:

```
SQL> SELECT *  FROM Courses WHERE Dept_ID NOT IN
  2  (SELECT Dept_ID FROM Departments WHERE Dept_Name = 'Mechanical Engg.');

COURSE COURSE_NAME              DEPT_ PROF_ID    CREDITS TOTAL_STUDENTS
------ ------------------------ ----- ------- ---------- --------------
CSE101 Data Structures          D01   P1001            4              2
CSE201 Operating Systems        D01   P1003            4              1
CIV101 Structural Analysis      D04   P1004            3              0
EEE101 Circuit Theory           D02   P1002            4              1
```