WEEK – 1

1. University Database Schema

```
CREATE TABLE Students (
   StudentID VARCHAR2(20),
   StudentName VARCHAR2(50),
   Major VARCHAR2(50)
);

CREATE TABLE Courses (
   CourseID VARCHAR2(20),
   CourseName VARCHAR2(50),
   Credits NUMBER
);

CREATE TABLE Enrollments (
   StudentID VARCHAR2(20),
   CourseID VARCHAR2(20),
   EnrollmentDate DATE
);

CREATE TABLE Instructors (
   InstructorID NUMBER,
   InstructorName VARCHAR2(50),
   Phone NUMBER
);

CREATE TABLE Course_Instructors (
   CourseID VARCHAR2(20),
   InstructorID NUMBER
);
```

WEEK – 2

2.1 DML Commands

```
CREATE TABLE students (
   rollno VARCHAR2(30),
   name VARCHAR2(30)
);

INSERT INTO students VALUES ('24B11CS234','Bala');
INSERT INTO students VALUES ('24B11CS381','Kiran');

SELECT * FROM students;
SELECT name FROM students;

DELETE FROM students WHERE rollno='24B11CS381';

UPDATE students SET name='Bala Raju' WHERE rollno='24B11CS234';
```

2.2 Aggregate Functions

```
SELECT AVG(amount) FROM company;
SELECT SUM(amount) FROM company;
SELECT MAX(amount) FROM company;
SELECT MIN(amount) FROM company;
SELECT COUNT(*) FROM company;

SELECT companyn, SUM(amount) FROM company GROUP BY companyn;
SELECT companyn, MIN(amount) FROM company GROUP BY companyn;
SELECT companyn, MAX(amount) FROM company GROUP BY companyn;
SELECT companyn, COUNT(*) FROM company GROUP BY companyn;

SELECT companyn, COUNT(*) FROM company GROUP BY companyn HAVING COUNT(*) > 1;
SELECT companyn, SUM(amount) FROM company GROUP BY companyn HAVING SUM(amount) >
10000;
```

WEEK – 3

3.1 Joins

```
SELECT * FROM tb1 INNER JOIN tb2 ON tb1.rno = tb2.rno;

SELECT * FROM tb1 LEFT OUTER JOIN tb2 ON tb1.rno = tb2.rno;

SELECT * FROM tb1 RIGHT OUTER JOIN tb2 ON tb1.rno = tb2.rno;

SELECT * FROM tb1 NATURAL JOIN tb2;

SELECT * FROM tb1 CROSS JOIN tb2;

SELECT t1.rno, t2.name FROM tb1 t1, tb1 t2 WHERE t1.rno = t2.rno;

SELECT * FROM tb1, tb2 WHERE tb1.rno = tb2.rno;
```

3.2 Set Operations

```
SELECT s.sname FROM sailors s,reserves r,boats b
WHERE s.sid=r.sid AND b.bid=r.bid AND b.bcolor='Red'
UNION
SELECT s.sname FROM sailors s,reserves r,boats b
WHERE s.sid=r.sid AND b.bid=r.bid AND b.bcolor='Green';


SELECT s.sname FROM sailors s,reserves r,boats b
WHERE s.sid=r.sid AND b.bid=r.bid AND b.bcolor='Red'
INTERSECT
SELECT s.sname FROM sailors s,reserves r,boats b
WHERE s.sid=r.sid AND b.bid=r.bid AND b.bcolor='Green';


SELECT s.sname FROM sailors s,reserves r,boats b
WHERE s.sid=r.sid AND b.bid=r.bid AND b.bcolor='Red'
MINUS
SELECT s.sname FROM sailors s,reserves r,boats b
WHERE s.sid=r.sid AND b.bid=r.bid AND b.bcolor='Green';
```

3.3 Nested & Correlated Subqueries

```
SELECT s.sname FROM sailors s
WHERE s.sid IN (SELECT r.sid FROM reserves r WHERE r.bid=103);
```

```
SELECT s.sname FROM sailors s
WHERE s.sid NOT IN (SELECT r.sid FROM reserves r WHERE r.bid=103);

SELECT sid FROM sailors WHERE rating >= ALL (SELECT rating FROM sailors);

SELECT s.sid FROM sailors s
WHERE s.rating > ANY (SELECT rating FROM sailors WHERE sname='Andy');

SELECT s.sname FROM sailors s
WHERE EXISTS (SELECT * FROM reserves r WHERE s.sid=r.sid AND r.bid=103);

SELECT s.sname FROM sailors s
WHERE NOT EXISTS (SELECT * FROM reserves r WHERE s.sid=r.sid AND r.bid=103);
```

3.4 Views

```
CREATE VIEW myview AS SELECT rollno,name FROM st1;
SELECT * FROM myview;

INSERT INTO myview VALUES (506,'prathisha');
DELETE FROM myview WHERE rollno=506;

CREATE OR REPLACE VIEW myview AS SELECT * FROM st1;
CREATE VIEW myview1 AS SELECT * FROM st1 WITH READ ONLY;
CREATE VIEW myview2 AS SELECT * FROM st1 WHERE marks<101 WITH CHECK OPTION;

DROP VIEW myview1;
```

WEEK – 4

4.1 Full Normalized Schema + Data

```
CREATE TABLE Students (StudentID INT PRIMARY KEY, StudentName VARCHAR(100), Major
VARCHAR(50));
```

CREATE TABLE Courses (CourseID INT PRIMARY KEY, CourseName VARCHAR(100), Credits INT);

CREATE TABLE Instructors (InstructorID INT PRIMARY KEY, InstructorName VARCHAR(100), Phone VARCHAR(15));

CREATE TABLE Course_Instructors (CourseID INT, InstructorID INT, PRIMARY KEY(CourseID,InstructorID));

CREATE TABLE Enrollments (EnrollmentID INT PRIMARY KEY, StudentID INT, CourseID INT, EnrollmentDate DATE);

INSERT INTO Students VALUES (1,'Asha Patel','CSE'),(2,'Ravi Kumar','ECE'),(3,'Maya Singh','CSE'),(4,'John Doe','Mathematics');

INSERT INTO Courses VALUES (101,'Database Systems',4),(102,'Operating Systems',3),(103,'Computer Networks',3),(104,'Calculus I',4);

INSERT INTO Instructors VALUES (11,'Dr. Mehta','9876543210'),(12,'Prof. Sharma','9123456780'),(13,'Dr. Rao','9000001111');

INSERT INTO Course_Instructors VALUES (101,11),(102,12),(103,11),(104,13);

INSERT INTO Enrollments VALUES (1001,1,101,'2025-07-01'),(1002,1,102,'2025-07-02'),(1003,2,101,'2025-07-03'),(1004,3,103,'2025-07-04');

4.2 DCL & TCL

CREATE USER student_user IDENTIFIED BY student123;

GRANT CONNECT, RESOURCE TO student_user;

GRANT SELECT, INSERT, UPDATE ON department TO student_user;

REVOKE INSERT ON department FROM student_user;

SAVEPOINT sp1;

ROLLBACK TO sp1;

COMMIT;

WEEK – 5 Indexes

CREATE TABLE StudentsF (

StudentID INT PRIMARY KEY,

FirstName VARCHAR2(50),

LastName VARCHAR2(50),

```
    EnrollmentDate DATE
);

CREATE INDEX idx_LastName ON StudentsF(LastName);

INSERT INTO StudentsF VALUES (101,'John','Doe','15-AUG-2025');
INSERT INTO StudentsF VALUES (102,'Jane','Smith','16-AUG-2025');
INSERT INTO StudentsF VALUES (103,'Ravi','Kumar','25-OCT-2025');

SELECT * FROM StudentsF WHERE StudentID = 102;
SELECT * FROM StudentsF WHERE LastName = 'Doe';

DELETE FROM StudentsF WHERE StudentID = 103;
DELETE FROM StudentsF WHERE LastName = 'Smith';

DROP INDEX idx_LastName;
```