**University Course Management System**

**3. Practice SQL Data Definition Language (DDL) Commands**

**3.1 Table creation and alteration**

1. Create all 5 tables based on the schema provided.
2. Add a column Dept\_Head (varchar2(50)) to Departments.
3. Change size of Experience\_Years in Professors to NUMBER(3).
4. Rename Student\_Count to Total\_Students in Courses.
5. Drop and recreate the Enrollments table.

**4. Practice SQL Data Manipulation Language (DML) Commands**

**4.1 Insertion, Deletion, Update**

1. Insert the sample data into all five tables.
2. Insert a new department: (‘D04’, ‘Civil Engg’, ‘Block C’, 5).
3. Create a new table high\_achievers containing students who scored more than 85 in any course.
4. Create a backup table Courses\_Backup with all data from Courses and Professors\_Backup with all the data from professors.
5. Add a new course ‘CIV101’, ‘Structural Analysis’, under D04, taught by P1001, with 3 credits and 0 students.
6. Update professor of ‘CIV101’ to P1004 and savepoint SP1.
7. Change credits of ‘CSE201’ to 4 and set savepoint SP2.
8. Delete all courses from Courses\_Backup that have less than 4 credits.
9. Delete all professors from Professors\_Backup with less than 10 years experience.
10. Rollback to SP1 and rename Courses\_Backup to Course\_Master.

**4.2 Data Retrieval (SELECT)**

1. List all department names.
2. Display all data from the Professors table.
3. List student names and DOBs.
4. List course names and credits.
5. Get courses offered by the ‘Computer Science’ department.
6. List professors whose name starts with ‘Dr’.
7. List courses with credits more than 3.
8. Display all courses with “Theory” in their name.
9. List students born after Jan 1, 2003.
10. Find all professors with 10+ years experience.
11. List all courses with more than 1 student.
12. Display all distinct semesters from Enrollments.
13. Show information of students with ID S0001, S0002.
14. Show all courses **not** in the ‘Mechanical Engg’ department.

**4.3 SQL Functions**

1. Use numeric functions like ROUND, MOD, POWER on dummy values.
2. Use string functions like LENGTH, SUBSTR, INSTR, UPPER, LOWER on names in Professors and Students.
3. Use conversion functions on DOB.
4. Count total number of students.
5. Find max and min marks in Enrollments as max\_marks, min\_marks.
6. Count number of students with marks over 75.

**4.4 Date Functions**

1. List student names and their day of birth.
2. Format DOBs in ‘DD-Month-YYYY’ format.
3. Show DOBs in ‘DD-MM-YY’ format.
4. Add 100 days to all DOBs.
5. List students born in May.
6. List students born between 2002 and 2003.

**4.5 Set Operators**

Create Top\_Courses table:

CREATE TABLE Top\_Courses (

Course\_Name VARCHAR2(100),

Dept\_Name VARCHAR2(50)

);

Insert some course-department pairs.

1. Show unique course names from both Courses and Top\_Courses.
2. Show common courses between Courses and Top\_Courses.
3. Show top courses that are not in current courses.
4. Show union all of both.

**4.6 Sorting**

1. Sort professors by descending experience.

**4.7 Group By, Having**

1. Number of students per department (only those >1).
2. Departments with avg classroom count >5.
3. Courses taught by professors with more than 1 course.

**4.8 Subqueries**

1. Find names of students enrolled in any course.
2. Find department of student S0001.
3. Departments where student count > 1.
4. List courses taught by professors with >10 years experience.
5. List students in departments with at least 1 professor.
6. Find the professors with exp less than the average experience of all the professors.

**4.9 NOT EXISTS**

1. Departments where every professor has >5 years experience.
2. Departments with no students.

**4.10 Correlated Subqueries**

1. Students with marks above average.
2. Students with DOB later than any student from 'Electrical Engg'.

**4.11 JOINING Tables**

1. List student names along with their department name.
2. Courses and number of students enrolled.
3. List course name and professor name.
4. Departments with more than one professor.
5. List students enrolled in courses with more than 3 credits.

**5. PL/SQL Section**

**5.1 Basic Blocks**

1. Accept a course ID and show number of students enrolled.
2. If department 'Biotech' doesn't exist, insert it with ‘D05’.
3. Calculate total students in ‘Computer Science’.

**5.2 Cursors**

1. Insert a new student and show row count.
2. Update marks and show affected row count.
3. List students with marks above 80 (Explicit Cursor).
4. List all courses taught by P1001 (Explicit Cursor).

**5.3 Procedures**

1. Procedure: Accept student ID and return number of enrolled courses.
2. Procedure: Insert new course if department and professor exist.
3. Procedure: Update course credits and log to new table.

**5.4 Functions**

1. Function to return age of student.
2. Function to count courses offered by a department.
3. Function to check if a professor teaches more than 2 courses.

**5.5 Triggers**

1. Prevent deletion of courses with more than 2 credits.
2. Log updates to Marks in Enrollments\_Log.
3. Prevent duplicate enrollments in a course.
4. Prevent update / delete being done on Sunday / Saturday.

**5.6 Views**

1. View: Students and number of courses they are enrolled in.
2. View: Courses and average marks of enrolled students (only for courses with >1 student).