Task 1: Student and Course Enrollment System

Tables: **Student**(sid, sname, ccode, dob, address),**Course**(ccode, cname, fees)

Problem Statements:

Q1) Write a query to add 5 students records into the Student table, including their student ID, name, course code, date of birth, and address.

```
mysql> INSERT INTO Student (sid, sname, ccode, dob, address) VALUES
    -> (1, 'Vinay', 'C101', '2006-06-12', '123 Rtc X road'),
    -> (2, 'Anirudh', 'C102', '2006-06-12', '456 LB Nagar'),
    -> (3, 'Drushya', 'C101', '2003-07-30', '789 BN Reddy'),
    -> (4, 'Sharath', 'C103', '2001-08-05', '135 Victoria St'),
    -> (5, 'Vaishnavi', 'C102', '2000-09-12', '246 Kothapet');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

Q2) Write a query to add 3 courses into the Course table, with course code, name, and fees.

```
mysql> INSERT INTO Course (ccode, cname, fees) VALUES
    -> ('C101', 'Database Management Systems', 3000.00),
    -> ('C102', 'Digital Electronics', 2500.00),
    -> ('C103', 'Java', 2800.00);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Q3) Write a query to modify the address of a student identified by his sid is 2.

```
mysql> UPDATE Student
    -> SET address = 'Updated Address Here'
    -> WHERE sid = 2;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Q4) Write a query to modify the course fees for the course with ccode = 'C101' by increasing it by 4%.

```
mysql> UPDATE Course
    -> SET fees = fees * 1.04
    -> WHERE ccode = 'C101';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Q5) Write a query to remove the record of a student whose sid belongs to 5.

```
mysql> DELETE FROM Student
    -> WHERE sid = 5;
Query OK, 1 row affected (0.01 sec)
```

Q6) Write a query to display all the students enrolled in the course ccode belongs 'C102'.

```
mysql> SELECT * FROM Student
    -> WHERE ccode = 'C102';
+-----+---------+-------+------------+---------------------+
| sid | sname   | ccode | dob        | address             |
+-----+---------+-------+------------+---------------------+
|   2 | Anirudh | C102  | 2006-06-12 | Updated Address Here |
+-----+---------+-------+------------+---------------------+
1 row in set (0.01 sec)
```

Q7) Write a query to display the names and fees of all courses where the fees are greater than 2000.

```
mysql> SELECT cname, fees FROM Course
    -> WHERE fees > 2000;
+----------------------------+---------+
| cname                      | fees    |
+----------------------------+---------+
| Database Management Systems | 3120.00 |
| Digital Electronics        | 2500.00 |
| Java                       | 2800.00 |
+----------------------------+---------+
3 rows in set (0.00 sec)
```

Task 2: University System with Departments

Tables:

Student(sid, sname, ccode, dob, address)**,**

Course(ccode, cname, fees, did)**,**

Department(did, dname, location)

Problem Statements:

Q8) Write a query to add 3 departments into the Department table with department ID, name, and location.

```
mysql> INSERT INTO Department (did, dname, location) VALUES
    -> (1, 'Computer Science', 'Block A'),
    -> (2, 'AIML', 'Block B'),
    -> (3, 'Data Science', 'Block C');
Query OK, 3 rows affected (0.00 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

Q9) Write a query to change the location of the department number 1 to 'New York'.

```
mysql> UPDATE Department
    -> SET location = 'New York'
    -> WHERE did = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Q10) Write a query to remove the department record whose did belong to 3.

```
mysql> DELETE FROM Department
    -> WHERE did = 3;
Query OK, 1 row affected (0.01 sec)
```

Q11) Write a query to find all students and display their names along with the dob who born between January and March

```
mysql> SELECT sname, dob
    -> FROM Student
    -> WHERE MONTH(dob) BETWEEN 1 AND 3;
+---------+------------+
| sname   | dob        |
+---------+------------+
| Vinay   | 2006-01-12 |
| Anirudh | 2006-02-12 |
+---------+------------+
2 rows in set (0.00 sec)
```

Q12) Write a query to display the details of courses taught by departments located in 'California'.

```
mysql> select * from Course
    -> where did=(select did from department
    -> where location="California");
Empty set (0.01 sec)
```

Task 3: University System with Faculty

Tables:

Student(sid, sname, ccode, dob, address)

Course(ccode, cname,did, fees)

Department(did, dname, location)

Faculty(fid, fname, sal, designation, doj, did)

Problem Statements:

Q13) Write a query to add 4 faculty members into the Faculty table with their ID, name, salary, designation, date of joining, and department ID.

```
mysql> INSERT INTO Faculty (fid, fname, sal, designation, doj, did) VALUES
    -> (1, 'Deepa Ganu', 6000, 'Professor', '2020-01-15', 1),
    -> (2, 'Rupa Devi', 5500, 'Assistant Professor', '2019-08-20', 2),
    -> (3, 'Asha', 5000, 'Lecturer', '2021-05-10', 1),
    -> (4, 'Vamshidhar', 6500, 'Associate Professor', '2022-03-22', 2);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql>
mysql> SELECT * FROM Faculty;
+-----+------------+------+--------------------+------------+------+
| fid | fname      | sal  | designation        | doj        | did  |
+-----+------------+------+--------------------+------------+------+
|   1 | Deepa Ganu | 6000 | Professor          | 2020-01-15 |    1 |
|   2 | Rupa Devi  | 5500 | Assistant Professor | 2019-08-20 |    2 |
|   3 | Asha       | 5000 | Lecturer           | 2021-05-10 |    1 |
|   4 | Vamshidhar | 6500 | Associate Professor | 2022-03-22 |    2 |
+-----+------------+------+--------------------+------------+------+
4 rows in set (0.00 sec)
```

Q14) Write a query to remove the faculty member who joined recently.

```
ERROR 1093 (HY000): You can't specify target table 'Faculty' for
mysql> CREATE TEMPORARY TABLE TempRecentDOJ (recent_doj DATE);
Query OK, 0 rows affected (0.01 sec)

mysql>
mysql> INSERT INTO TempRecentDOJ (recent_doj)
    -> SELECT MAX(doj) FROM Faculty;
Query OK, 1 row affected (0.00 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql>
mysql> DELETE FROM Faculty
    -> WHERE doj = (SELECT recent_doj FROM TempRecentDOJ);
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> DROP TEMPORARY TABLE TempRecentDOJ;
Query OK, 0 rows affected (0.00 sec)
```

Q15) Write a query to display all faculty members who have joined before 2022.

```
mysql> SELECT *
    -> FROM Faculty
    -> WHERE doj < '2022-01-01';
+-----+------------+------+--------------------+------------+------+
| fid | fname      | sal  | designation        | doj        | did  |
+-----+------------+------+--------------------+------------+------+
|   1 | Deepa Ganu | 6000 | Professor          | 2020-01-15 |    1 |
|   2 | Rupa Devi  | 5500 | Assistant Professor | 2019-08-20 |    2 |
|   3 | Asha       | 5000 | Lecturer           | 2021-05-10 |    1 |
+-----+------------+------+--------------------+------------+------+
3 rows in set (0.00 sec)
```

Q16) Write a query to display the department wise monthly expenditure on salaries of faculty members

```
mysql> SELECT Department.dname, SUM(Faculty.sal) AS total_salary
    -> FROM Faculty, Department
    -> WHERE Faculty.did = Department.did
    -> GROUP BY Department.dname;
+------------------+--------------+
| dname            | total_salary |
+------------------+--------------+
| Computer Science |        11000 |
| AIML             |         5500 |
+------------------+--------------+
2 rows in set (0.01 sec)
```

Q17) Write a query to display the faculty member(s) with the highest salary in each department.

```
mysql> SELECT f.*
    -> FROM Faculty f
    -> WHERE f.sal = (
    ->     SELECT MAX(sal)
    ->     FROM Faculty
    ->     WHERE did = f.did
    -> );
+-----+------------+------+---------------------+------------+------+
| fid | fname      | sal  | designation         | doj        | did  |
+-----+------------+------+---------------------+------------+------+
|   1 | Deepa Ganu | 6000 | Professor           | 2020-01-15 |    1 |
|   2 | Rupa Devi  | 5500 | Assistant Professor | 2019-08-20 |    2 |
+-----+------------+------+---------------------+------------+------+
2 rows in set (0.00 sec)
```

Q18) Write a query to increment the salary by 12 % of their salary who teaches DBMS.

```
mysql> UPDATE Faculty
    -> SET sal = sal * 1.12
    -> WHERE did IN (
    ->     SELECT did
    ->     FROM Course
    ->     WHERE cname = 'DBMS'
    -> );
Query OK, 0 rows affected (0.01 sec)
Rows matched: 0  Changed: 0  Warnings: 0
```

Task 4: Employee and Department System

Tables:

Employee(eid, ename, salary, doj, comm,did)

Department(did, dname, location)

Problem Statements:

Q19) Write a query to add 5 employees into the Employee table with their employee ID, name, salary, date of joining,did and commission at a time.

```
mysql> INSERT INTO Employee (eid, ename, salary, doj, did, commission) VALUES
    -> (1, 'Vinay', 5000, '2020-01-10', 1, 500),
    -> (2, 'Drushya', 6000, '2019-03-15', 2, 600),
    -> (3, 'Anirudh', 5500, '2021-06-20', 1, 550),
    -> (4, 'Sharath', 7000, '2018-11-30', 2, 700),
    -> (5, 'Vaishnavi', 6500, '2022-02-25', 1, 650);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Employee;
+-----+-----------+--------+------------+------+------------+
| eid | ename     | salary | doj        | did  | commission |
+-----+-----------+--------+------------+------+------------+
|   1 | Vinay     |   5000 | 2020-01-10 |    1 |        500 |
|   2 | Drushya   |   6000 | 2019-03-15 |    2 |        600 |
|   3 | Anirudh   |   5500 | 2021-06-20 |    1 |        550 |
|   4 | Sharath   |   7000 | 2018-11-30 |    2 |        700 |
|   5 | Vaishnavi |   6500 | 2022-02-25 |    1 |        650 |
+-----+-----------+--------+------------+------+------------+
5 rows in set (0.00 sec)
```

Q20) Write a query to give extra 5% commission for the employees whose experience is more than 5 years.

```
mysql> UPDATE Employee
    -> SET commission = commission * 1.05
    -> WHERE YEAR(CURDATE()) - YEAR(doj) > 5;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Q21) Write a query to remove the record of the employee who has been with the company for the longest time (use a subquery to identify the employee with the earliest date of joining).

```
mysql> CREATE TEMPORARY TABLE TempEmployees AS
    -> SELECT eid
    -> FROM Employee
    -> WHERE doj = (SELECT MIN(doj) FROM Employee);
Query OK, 1 row affected (0.00 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql>
mysql> DELETE FROM Employee
    -> WHERE eid IN (SELECT eid FROM TempEmployees);
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> DROP TEMPORARY TABLE TempEmployees;
Query OK, 0 rows affected (0.00 sec)
```

Q22) Write a query to display the name and salary of employees who have a salary greater than 5000.

```
mysql> SELECT ename, salary
    -> FROM Employee
    -> WHERE salary > 5000;
+-----------+--------+
| ename     | salary |
+-----------+--------+
| Drushya   |   6000 |
| Anirudh   |   5500 |
| Vaishnavi |   6500 |
+-----------+--------+
3 rows in set (0.00 sec)
```

Q23) Write a query to display the details of employees who work in departments located in 'San Francisco'.

```
mysql> SELECT *
    -> FROM Employee
    -> WHERE did IN (
    ->     SELECT did
    ->     FROM Department
    ->     WHERE location = 'San Francisco'
    -> );
Empty set (0.00 sec)
```