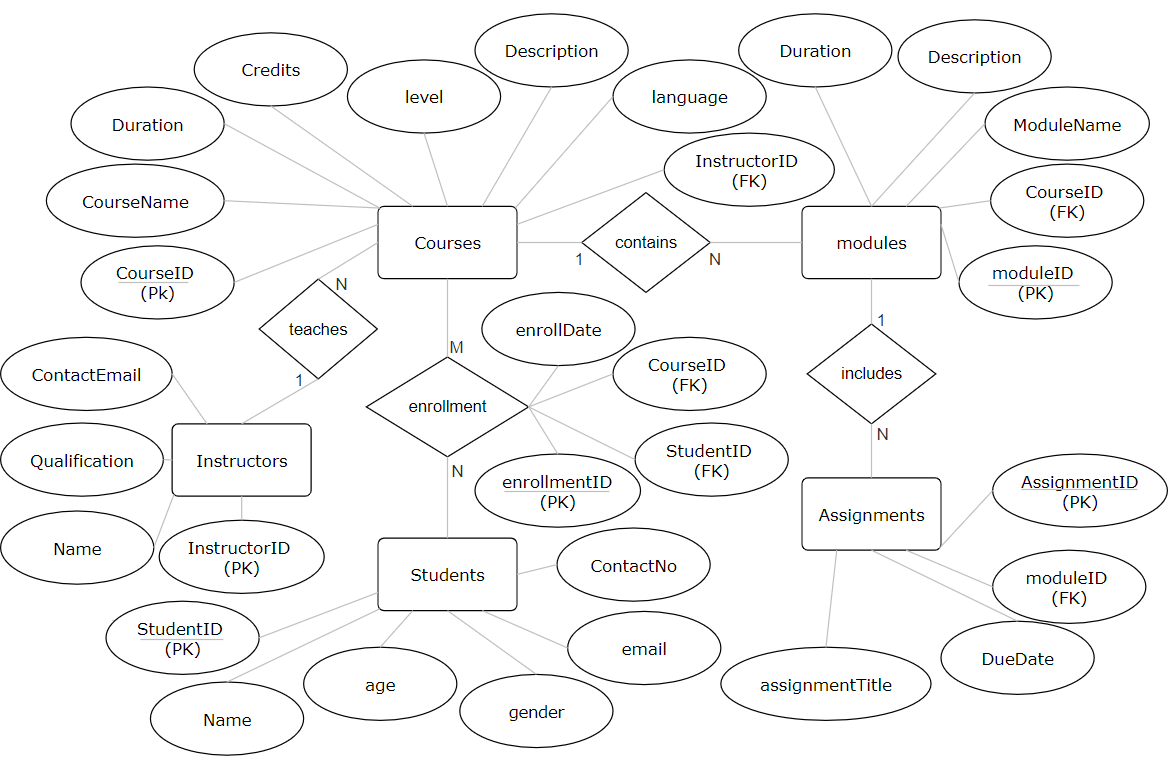
**DBMS CIA 3 (PART a)**

**Register Number : 2341316  
Class: BSc 3DM**

**Entity-Relationship (ER) Diagram for 'NPTEL'**



Entities for the NPTEL Domain:

1. Courses: Represents all courses offered by NPTEL.
2. Instructors: Represents instructors who teach NPTEL courses.
3. Students: Represents students enrolled in NPTEL courses.
4. Enrollment: Represents the enrollment relationship between students and courses.
5. Modules: Represents the course modules.
6. Assignments: Represents assignments for each course module.

**Project Description:**

The project revolves around the design and implementation of a DBMS for managing NPTEL-like online courses, including entities such as instructors, students, courses, enrollments, assignments, and modules. The project illustrates the use of various MySQL commands, focusing on Data Definition Language (DDL), Data Manipulation Language (DML), Transaction Control Language (TCL), Views (VDL), and joins to manipulate and query the data efficiently. It also includes relational algebra operations to manage the database effectively.

**Project Breakdown:**

1. DDL Commands (Data Definition Language)

* CREATE DATABASE: Creates the NPTEL database.
  + Example Code:

CREATE DATABASE NPTEL;

* + Output: Creates a new database named NPTEL.
* CREATE: Used to create new tables and databases.
  + Example Code:

CREATE TABLE Students ( StudentID INT PRIMARY KEY AUTO\_INCREMENT, Name VARCHAR(100), Age INT, Gender VARCHAR(10), Email VARCHAR(100) UNIQUE );

* + Output: Creates a Students table with the specified structure.
* ALTER: Modifies an existing table.
* DROP: Deletes a table or database.

2. DML Commands (Data Manipulation Language)

* INSERT: Adds new data into the table.
  + Example Code:

INSERT INTO Students (Name, Age, Gender, Email) VALUES ('Alice', 22, 'Female', 'alice@example.com');

* + Output: Inserts a student record.
* SELECT: Retrieves data from one or more tables
  + Example Code:

SELECT \* FROM Students;

* + Output: Displays all records from the Students table.
* **UPDATE**: Modifies existing data in a table.
  + Example Code:

UPDATE Students SET Age = 26 WHERE Name = 'David';

* + Output: Updates the age of the student named David.
* **DELETE**: Removes records from a table.
  + Example Code:

DELETE FROM Students WHERE Age > 24;

* + Output: Deletes students older than 24 years.

3. TCL Commands (Transaction Control Language)

* START TRANSACTION: Begins a transaction.
  + Example Code:

START TRANSACTION;

* + Output: Begins a transaction session to execute multiple commands.
* SAVEPOINT: Sets a savepoint within a transaction.
  + Example Code:

SAVEPOINT sp1;

* COMMIT: Saves all changes made in a transaction.
  + Example Code:

COMMIT;

* ROLLBACK: Reverts the changes made in a transaction.
  + Example Code:

START TRANSACTION; DELETE FROM Students WHERE StudentID = 3; ROLLBACK;

* + Output: The deletion is reversed, and the record remains intact.

4. VDL Commands (View Definition Language)

* CREATE VIEW: Creates a virtual table (view) based on the result of a query.
  + Example Code:

CREATE VIEW CourseDetails AS SELECT Courses.CourseName, Instructors.Name AS InstructorName FROM Courses JOIN Instructors ON Courses.InstructorID = Instructors.InstructorID;

* + Output: Creates a view that shows course names with their respective instructors.
* UPDATE VIEW: Modifies data in a view.
  + Example Code:

UPDATE AdvancedCourses SET Credits = 6 WHERE CourseName = 'Machine Learning';

* DELETE (from a view): Tries to delete data via a view.
  + Example Code:

DELETE FROM Courses WHERE CourseID IN ( SELECT CourseID FROM AdvancedCourses WHERE Credits = 6 );

* + Output:

5. Types of Joins

* INNER JOIN: Combines rows from two or more tables based on a related column.
  + Example Code:

SELECT Students.Name, Courses.CourseName FROM Students INNER JOIN Enrollment ON Students.StudentID = Enrollment.StudentID INNER JOIN Courses ON Enrollment.CourseID = Courses.CourseID;

* + Output: Lists students and their enrolled courses.
* **LEFT JOIN**: Returns all records from the left table and matched records from the right table.
  + Example Code:

SELECT Courses.CourseName, Students.Name AS StudentName FROM Courses LEFT JOIN Enrollment ON Courses.CourseID = Enrollment.CourseID LEFT JOIN Students ON Enrollment.StudentID = Students.StudentID;

* + Output: Displays all courses and students enrolled (if any).
* **RIGHT JOIN**: Returns all records from the right table and matched records from the left table.
* **FULL OUTER JOIN**: Returns all records when there is a match in either left or right table (simulated using UNION).
  + Example Code:

(SELECT Students.Name, Courses.CourseName FROM Students LEFT JOIN Enrollment ON Students.StudentID = Enrollment.StudentID LEFT JOIN Courses ON Enrollment.CourseID = Courses.CourseID) UNION (SELECT Students.Name, Courses.CourseName FROM Students RIGHT JOIN Enrollment ON Students.StudentID = Enrollment.StudentID RIGHT JOIN Courses ON Enrollment.CourseID = Courses.CourseID);

6. Relational Algebra Operations

* Projection: Retrieves specific columns from the table.
  + Example Code:

SELECT Name, Age FROM Students;

* Selection: Retrieves rows that meet specific conditions.
* Cartesian Product: Combines all rows from two tables.
* Example Code:

SELECT \* FROM Courses, Instructors;

* Division: Retrieves records that meet all the specified conditions.
* Example Code:

SELECT StudentID FROM Enrollment GROUP BY StudentID HAVING COUNT(DISTINCT CourseID) = (SELECT COUNT(CourseID) FROM Courses);

7. Set Operations and Subqueries

* Nested Queries: Queries inside queries.
  + Example Code:

SELECT CourseName FROM Courses WHERE InstructorID = ( SELECT InstructorID FROM Instructors WHERE Name = 'John Doe' );

* + Output: Lists courses taught by John Doe.
* Set Difference: Finds records in one table that don’t match another table (without MINUS).
* Example Code:

SELECT CourseID FROM Courses WHERE CourseID NOT IN (SELECT CourseID FROM Enrollment);

* ANY/ALL: Compares values against any/all results from a subquery.
* Example Code:

SELECT CourseName FROM Courses WHERE Duration > ANY (SELECT Duration FROM Modules);

8. Aggregate Functions and Grouping

* GROUP BY: Groups data by a specific column.
  + Example:

SELECT CourseID, COUNT(\*) FROM Modules GROUP BY CourseID;

* + Output: Counts the number of modules per course.
* HAVING: Filters results after grouping
* Example Code:

SELECT CourseID FROM Modules GROUP BY CourseID HAVING COUNT(ModuleID) > 1;

**Full MySQL executed code:**

Enter password: \*\*\*\*\*\*\*\*\*\*

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 15

Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> -- Step 1: Create the NPTEL Database

mysql> CREATE DATABASE NPTEL;

Query OK, 1 row affected (0.01 sec)

mysql> USE NPTEL;

Database changed

mysql>

mysql> -- Step 2: Create Tables (Instructors, Courses, Students, Enrollment, Modules, Assignments)

mysql> CREATE TABLE Instructors (

-> InstructorID INT PRIMARY KEY AUTO\_INCREMENT,

-> Name VARCHAR(100) NOT NULL,

-> Qualification VARCHAR(100),

-> ContactEmail VARCHAR(100) UNIQUE

-> );

Query OK, 0 rows affected (0.04 sec)

mysql>

mysql> CREATE TABLE Courses (

-> CourseID INT PRIMARY KEY AUTO\_INCREMENT,

-> CourseName VARCHAR(100) NOT NULL,

-> Duration INT,

-> Credits INT,

-> Level VARCHAR(50),

-> Description TEXT,

-> Language VARCHAR(50),

-> InstructorID INT,

-> FOREIGN KEY (InstructorID) REFERENCES Instructors(InstructorID)

-> );

Query OK, 0 rows affected (0.04 sec)

mysql>

mysql> CREATE TABLE Students (

-> StudentID INT PRIMARY KEY AUTO\_INCREMENT,

-> Name VARCHAR(100) NOT NULL,

-> Age INT,

-> Gender VARCHAR(10),

-> Email VARCHAR(100) UNIQUE,

-> ContactNo VARCHAR(15)

-> );

Query OK, 0 rows affected (0.04 sec)

mysql>

mysql> CREATE TABLE Enrollment (

-> EnrollmentID INT PRIMARY KEY AUTO\_INCREMENT,

-> StudentID INT,

-> CourseID INT,

-> EnrollDate DATE,

-> FOREIGN KEY (StudentID) REFERENCES Students(StudentID),

-> FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)

-> );

Query OK, 0 rows affected (0.04 sec)

mysql>

mysql> CREATE TABLE Modules (

-> ModuleID INT PRIMARY KEY AUTO\_INCREMENT,

-> CourseID INT,

-> ModuleName VARCHAR(100),

-> Description TEXT,

-> Duration INT,

-> FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)

-> );

Query OK, 0 rows affected (0.04 sec)

mysql>

mysql> CREATE TABLE Assignments (

-> AssignmentID INT PRIMARY KEY AUTO\_INCREMENT,

-> ModuleID INT,

-> AssignmentTitle VARCHAR(100),

-> DueDate DATE,

-> FOREIGN KEY (ModuleID) REFERENCES Modules(ModuleID)

-> );

Query OK, 0 rows affected (0.04 sec)

mysql>

mysql> -- Verify Table Creation

mysql> SHOW TABLES;

+-----------------+

| Tables\_in\_nptel |

+-----------------+

| assignments |

| courses |

| enrollment |

| instructors |

| modules |

| students |

+-----------------+

6 rows in set (0.01 sec)

mysql>

mysql> -- Step 3: Insert Sample Data

mysql> INSERT INTO Instructors (Name, Qualification, ContactEmail) VALUES

-> ('John Doe', 'PhD in Computer Science', 'john.doe@example.com'),

-> ('Jane Smith', 'MSc in Data Science', 'jane.smith@example.com');

Query OK, 2 rows affected (0.03 sec)

Records: 2 Duplicates: 0 Warnings: 0

mysql>

mysql> -- Verify Data Insertion

mysql> SELECT \* FROM Instructors;

+--------------+------------+-------------------------+------------------------+

| InstructorID | Name | Qualification | ContactEmail |

+--------------+------------+-------------------------+------------------------+

| 1 | John Doe | PhD in Computer Science | john.doe@example.com |

| 2 | Jane Smith | MSc in Data Science | jane.smith@example.com |

+--------------+------------+-------------------------+------------------------+

2 rows in set (0.00 sec)

mysql>

mysql> INSERT INTO Courses (CourseName, Duration, Credits, Level, Description, Language, InstructorID) VALUES

-> ('Data Structures', 12, 4, 'Intermediate', 'Introduction to Data Structures', 'English', 1),

-> ('Machine Learning', 16, 5, 'Advanced', 'Comprehensive course on Machine Learning', 'English', 2),

-> ('Deep Learning', 20, 6, 'Advanced', 'In-depth course on Deep Learning techniques', 'English', 2);

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql>

mysql> -- Verify Data Insertion

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 5 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql>

mysql> INSERT INTO Students (Name, Age, Gender, Email, ContactNo) VALUES

-> ('Alice', 22, 'Female', 'alice@example.com', '1234567890'),

-> ('Bob', 24, 'Male', 'bob@example.com', '0987654321'),

-> ('Charlie', 23, 'Male', 'charlie@example.com', '1122334455');

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql>

mysql> -- Verify Data Insertion

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

+-----------+---------+------+--------+---------------------+------------+

3 rows in set (0.00 sec)

mysql>

mysql> INSERT INTO Enrollment (StudentID, CourseID, EnrollDate) VALUES

-> (1, 1, '2024-08-01'),

-> (2, 2, '2024-08-05'),

-> (3, 3, '2024-08-10');

Query OK, 3 rows affected (0.00 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql>

mysql> -- Verify Data Insertion

mysql> SELECT \* FROM Enrollment;

+--------------+-----------+----------+------------+

| EnrollmentID | StudentID | CourseID | EnrollDate |

+--------------+-----------+----------+------------+

| 1 | 1 | 1 | 2024-08-01 |

| 2 | 2 | 2 | 2024-08-05 |

| 3 | 3 | 3 | 2024-08-10 |

+--------------+-----------+----------+------------+

3 rows in set (0.00 sec)

mysql>

mysql> INSERT INTO Modules (CourseID, ModuleName, Description, Duration) VALUES

-> (1, 'Introduction to Arrays', 'Basic concepts of arrays', 3),

-> (1, 'Linked Lists', 'Concepts of linked lists', 4),

-> (2, 'Supervised Learning', 'Introduction to supervised learning algorithms', 4),

-> (3, 'Neural Networks', 'Introduction to neural network architectures', 5);

Query OK, 4 rows affected (0.00 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql>

mysql> -- Verify Data Insertion

mysql> SELECT \* FROM Modules;

+----------+----------+------------------------+------------------------------------------------+----------+

| ModuleID | CourseID | ModuleName | Description | Duration |

+----------+----------+------------------------+------------------------------------------------+----------+

| 1 | 1 | Introduction to Arrays | Basic concepts of arrays | 3 |

| 2 | 1 | Linked Lists | Concepts of linked lists | 4 |

| 3 | 2 | Supervised Learning | Introduction to supervised learning algorithms | 4 |

| 4 | 3 | Neural Networks | Introduction to neural network architectures | 5 |

+----------+----------+------------------------+------------------------------------------------+----------+

4 rows in set (0.00 sec)

mysql>

mysql> INSERT INTO Assignments (ModuleID, AssignmentTitle, DueDate) VALUES

-> (1, 'Array Assignment 1', '2024-09-01'),

-> (2, 'Linked List Assignment', '2024-09-10'),

-> (3, 'Supervised Learning Quiz', '2024-09-15'),

-> (4, 'Neural Network Project', '2024-10-01');

Query OK, 4 rows affected (0.00 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql>

mysql> -- Verify Data Insertion

mysql> SELECT \* FROM Assignments;

+--------------+----------+--------------------------+------------+

| AssignmentID | ModuleID | AssignmentTitle | DueDate |

+--------------+----------+--------------------------+------------+

| 1 | 1 | Array Assignment 1 | 2024-09-01 |

| 2 | 2 | Linked List Assignment | 2024-09-10 |

| 3 | 3 | Supervised Learning Quiz | 2024-09-15 |

| 4 | 4 | Neural Network Project | 2024-10-01 |

+--------------+----------+--------------------------+------------+

4 rows in set (0.00 sec)

mysql>

mysql> -- Step 4: Demonstrate Joins

mysql> -- Inner Join: List students enrolled in each course

mysql> SELECT Students.Name AS StudentName, Courses.CourseName

-> FROM Students

-> INNER JOIN Enrollment ON Students.StudentID = Enrollment.StudentID

-> INNER JOIN Courses ON Enrollment.CourseID = Courses.CourseID;

+-------------+------------------+

| StudentName | CourseName |

+-------------+------------------+

| Alice | Data Structures |

| Bob | Machine Learning |

| Charlie | Deep Learning |

+-------------+------------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Verify Join

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

+-----------+---------+------+--------+---------------------+------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Enrollment;

+--------------+-----------+----------+------------+

| EnrollmentID | StudentID | CourseID | EnrollDate |

+--------------+-----------+----------+------------+

| 1 | 1 | 1 | 2024-08-01 |

| 2 | 2 | 2 | 2024-08-05 |

| 3 | 3 | 3 | 2024-08-10 |

+--------------+-----------+----------+------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 5 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Left Join: List all courses and their enrolled students (if any)

mysql> SELECT Courses.CourseName, Students.Name AS StudentName

-> FROM Courses

-> LEFT JOIN Enrollment ON Courses.CourseID = Enrollment.CourseID

-> LEFT JOIN Students ON Enrollment.StudentID = Students.StudentID;

+------------------+-------------+

| CourseName | StudentName |

+------------------+-------------+

| Data Structures | Alice |

| Machine Learning | Bob |

| Deep Learning | Charlie |

+------------------+-------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Verify Join

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 5 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Enrollment;

+--------------+-----------+----------+------------+

| EnrollmentID | StudentID | CourseID | EnrollDate |

+--------------+-----------+----------+------------+

| 1 | 1 | 1 | 2024-08-01 |

| 2 | 2 | 2 | 2024-08-05 |

| 3 | 3 | 3 | 2024-08-10 |

+--------------+-----------+----------+------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

+-----------+---------+------+--------+---------------------+------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Right Join: List all students and their enrolled courses (if any)

mysql> SELECT Students.Name AS StudentName, Courses.CourseName

-> FROM Students

-> RIGHT JOIN Enrollment ON Students.StudentID = Enrollment.StudentID

-> RIGHT JOIN Courses ON Enrollment.CourseID = Courses.CourseID;

+-------------+------------------+

| StudentName | CourseName |

+-------------+------------------+

| Alice | Data Structures |

| Bob | Machine Learning |

| Charlie | Deep Learning |

+-------------+------------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Verify Join

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

+-----------+---------+------+--------+---------------------+------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Enrollment;

+--------------+-----------+----------+------------+

| EnrollmentID | StudentID | CourseID | EnrollDate |

+--------------+-----------+----------+------------+

| 1 | 1 | 1 | 2024-08-01 |

| 2 | 2 | 2 | 2024-08-05 |

| 3 | 3 | 3 | 2024-08-10 |

+--------------+-----------+----------+------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 5 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Full Outer Join (Using UNION since MySQL doesn't support FULL OUTER JOIN directly)

mysql> (SELECT Students.Name AS StudentName, Courses.CourseName

-> FROM Students

-> LEFT JOIN Enrollment ON Students.StudentID = Enrollment.StudentID

-> LEFT JOIN Courses ON Enrollment.CourseID = Courses.CourseID)

-> UNION

-> (SELECT Students.Name AS StudentName, Courses.CourseName

-> FROM Students

-> RIGHT JOIN Enrollment ON Students.StudentID = Enrollment.StudentID

-> RIGHT JOIN Courses ON Enrollment.CourseID = Courses.CourseID);

+-------------+------------------+

| StudentName | CourseName |

+-------------+------------------+

| Alice | Data Structures |

| Bob | Machine Learning |

| Charlie | Deep Learning |

+-------------+------------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Verify Join

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

+-----------+---------+------+--------+---------------------+------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Enrollment;

+--------------+-----------+----------+------------+

| EnrollmentID | StudentID | CourseID | EnrollDate |

+--------------+-----------+----------+------------+

| 1 | 1 | 1 | 2024-08-01 |

| 2 | 2 | 2 | 2024-08-05 |

| 3 | 3 | 3 | 2024-08-10 |

+--------------+-----------+----------+------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 5 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Step 5: Set Operations and Nested Queries

mysql> -- Set Difference without MINUS Operator

mysql> SELECT CourseID FROM Courses

-> WHERE CourseID NOT IN (

-> SELECT CourseID FROM Enrollment

-> );

Empty set (0.00 sec)

mysql>

mysql> -- Verify Set Difference

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 5 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Enrollment;

+--------------+-----------+----------+------------+

| EnrollmentID | StudentID | CourseID | EnrollDate |

+--------------+-----------+----------+------------+

| 1 | 1 | 1 | 2024-08-01 |

| 2 | 2 | 2 | 2024-08-05 |

| 3 | 3 | 3 | 2024-08-10 |

+--------------+-----------+----------+------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Nested Query: Find courses taught by 'John Doe'

mysql> SELECT CourseName

-> FROM Courses

-> WHERE InstructorID = (

-> SELECT InstructorID FROM Instructors WHERE Name = 'John Doe'

-> );

+-----------------+

| CourseName |

+-----------------+

| Data Structures |

+-----------------+

1 row in set (0.00 sec)

mysql>

mysql> -- Verify Nested Query

mysql> SELECT \* FROM Instructors;

+--------------+------------+-------------------------+------------------------+

| InstructorID | Name | Qualification | ContactEmail |

+--------------+------------+-------------------------+------------------------+

| 1 | John Doe | PhD in Computer Science | john.doe@example.com |

| 2 | Jane Smith | MSc in Data Science | jane.smith@example.com |

+--------------+------------+-------------------------+------------------------+

2 rows in set (0.00 sec)

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 5 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Nested Query with more than 2 tables: Find students enrolled in courses taught by 'Jane Smith'

mysql> SELECT Students.Name AS StudentName

-> FROM Students

-> JOIN Enrollment ON Students.StudentID = Enrollment.StudentID

-> JOIN Courses ON Enrollment.CourseID = Courses.CourseID

-> WHERE Courses.InstructorID = (

-> SELECT InstructorID FROM Instructors WHERE Name = 'Jane Smith'

-> );

+-------------+

| StudentName |

+-------------+

| Bob |

| Charlie |

+-------------+

2 rows in set (0.00 sec)

mysql>

mysql> -- Verify Nested Query

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

+-----------+---------+------+--------+---------------------+------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Enrollment;

+--------------+-----------+----------+------------+

| EnrollmentID | StudentID | CourseID | EnrollDate |

+--------------+-----------+----------+------------+

| 1 | 1 | 1 | 2024-08-01 |

| 2 | 2 | 2 | 2024-08-05 |

| 3 | 3 | 3 | 2024-08-10 |

+--------------+-----------+----------+------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 5 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Instructors;

+--------------+------------+-------------------------+------------------------+

| InstructorID | Name | Qualification | ContactEmail |

+--------------+------------+-------------------------+------------------------+

| 1 | John Doe | PhD in Computer Science | john.doe@example.com |

| 2 | Jane Smith | MSc in Data Science | jane.smith@example.com |

+--------------+------------+-------------------------+------------------------+

2 rows in set (0.00 sec)

mysql>

mysql> -- Subquery using relational operators and group functions: Find courses with more than 1 module

mysql> SELECT CourseName

-> FROM Courses

-> WHERE CourseID IN (

-> SELECT CourseID FROM Modules

-> GROUP BY CourseID

-> HAVING COUNT(ModuleID) > 1

-> );

+-----------------+

| CourseName |

+-----------------+

| Data Structures |

+-----------------+

1 row in set (0.00 sec)

mysql>

mysql> -- Verify Subquery

mysql> SELECT \* FROM Modules;

+----------+----------+------------------------+------------------------------------------------+----------+

| ModuleID | CourseID | ModuleName | Description | Duration |

+----------+----------+------------------------+------------------------------------------------+----------+

| 1 | 1 | Introduction to Arrays | Basic concepts of arrays | 3 |

| 2 | 1 | Linked Lists | Concepts of linked lists | 4 |

| 3 | 2 | Supervised Learning | Introduction to supervised learning algorithms | 4 |

| 4 | 3 | Neural Networks | Introduction to neural network architectures | 5 |

+----------+----------+------------------------+------------------------------------------------+----------+

4 rows in set (0.00 sec)

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 5 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Use of ANY and ALL with membership operators

mysql> -- Find courses where the duration is greater than the duration of any module

mysql> SELECT CourseName

-> FROM Courses

-> WHERE Duration > ANY (

-> SELECT Duration FROM Modules

-> );

+------------------+

| CourseName |

+------------------+

| Data Structures |

| Machine Learning |

| Deep Learning |

+------------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Verify Query

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 5 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Modules;

+----------+----------+------------------------+------------------------------------------------+----------+

| ModuleID | CourseID | ModuleName | Description | Duration |

+----------+----------+------------------------+------------------------------------------------+----------+

| 1 | 1 | Introduction to Arrays | Basic concepts of arrays | 3 |

| 2 | 1 | Linked Lists | Concepts of linked lists | 4 |

| 3 | 2 | Supervised Learning | Introduction to supervised learning algorithms | 4 |

| 4 | 3 | Neural Networks | Introduction to neural network architectures | 5 |

+----------+----------+------------------------+------------------------------------------------+----------+

4 rows in set (0.00 sec)

mysql>

mysql> -- Step 6: Transaction Control Commands

mysql> START TRANSACTION;

Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO Students (Name, Age, Gender, Email) VALUES ('David', 25, 'Male', 'david@example.com');

Query OK, 1 row affected (0.00 sec)

mysql>

mysql> -- Verify Insertion

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

| 4 | David | 25 | Male | david@example.com | NULL |

+-----------+---------+------+--------+---------------------+------------+

4 rows in set (0.00 sec)

mysql>

mysql> SAVEPOINT sp1;

Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE Students SET Age = 26 WHERE Name = 'David';

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql>

mysql> -- Verify Update

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

| 4 | David | 26 | Male | david@example.com | NULL |

+-----------+---------+------+--------+---------------------+------------+

4 rows in set (0.00 sec)

mysql>

mysql> ROLLBACK TO sp1;

Query OK, 0 rows affected (0.00 sec)

mysql>

mysql> -- Verify Rollback

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

| 4 | David | 25 | Male | david@example.com | NULL |

+-----------+---------+------+--------+---------------------+------------+

4 rows in set (0.00 sec)

mysql>

mysql> COMMIT;

Query OK, 0 rows affected (0.00 sec)

mysql>

mysql> -- Verify Commit

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

| 4 | David | 25 | Male | david@example.com | NULL |

+-----------+---------+------+--------+---------------------+------------+

4 rows in set (0.00 sec)

mysql>

mysql> -- Step 7: Views

mysql> -- Create a view of all courses and their instructors

mysql> CREATE VIEW CourseDetails AS

-> SELECT Courses.CourseName, Instructors.Name AS InstructorName

-> FROM Courses

-> JOIN Instructors ON Courses.InstructorID = Instructors.InstructorID;

Query OK, 0 rows affected (0.01 sec)

mysql>

mysql> -- Verify View Creation

mysql> SELECT \* FROM CourseDetails;

+------------------+----------------+

| CourseName | InstructorName |

+------------------+----------------+

| Data Structures | John Doe |

| Machine Learning | Jane Smith |

| Deep Learning | Jane Smith |

+------------------+----------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Create a view with selected fields and conditions

mysql> CREATE VIEW AdvancedCourses AS

-> SELECT CourseName, Credits, Level

-> FROM Courses

-> WHERE Level = 'Advanced';

Query OK, 0 rows affected (0.01 sec)

mysql>

mysql> -- Verify View Creation

mysql> SELECT \* FROM AdvancedCourses;

+------------------+---------+----------+

| CourseName | Credits | Level |

+------------------+---------+----------+

| Machine Learning | 5 | Advanced |

| Deep Learning | 6 | Advanced |

+------------------+---------+----------+

2 rows in set (0.00 sec)

mysql>

mysql> -- Create a view using nested queries

mysql> CREATE VIEW EnrollmentDetails AS

-> SELECT Students.Name AS StudentName, Courses.CourseName, Enrollment.EnrollDate

-> FROM Students

-> JOIN Enrollment ON Students.StudentID = Enrollment.StudentID

-> JOIN Courses ON Enrollment.CourseID = Courses.CourseID;

Query OK, 0 rows affected (0.01 sec)

mysql>

mysql> -- Verify View Creation

mysql> SELECT \* FROM EnrollmentDetails;

+-------------+------------------+------------+

| StudentName | CourseName | EnrollDate |

+-------------+------------------+------------+

| Alice | Data Structures | 2024-08-01 |

| Bob | Machine Learning | 2024-08-05 |

| Charlie | Deep Learning | 2024-08-10 |

+-------------+------------------+------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Perform equi-join and make the result a view

mysql> CREATE VIEW StudentCourseView AS

-> SELECT Students.StudentID, Students.Name, Courses.CourseID, Courses.CourseName

-> FROM Students

-> JOIN Enrollment ON Students.StudentID = Enrollment.StudentID

-> JOIN Courses ON Enrollment.CourseID = Courses.CourseID;

Query OK, 0 rows affected (0.01 sec)

mysql>

mysql> -- Verify View Creation

mysql> SELECT \* FROM StudentCourseView;

+-----------+---------+----------+------------------+

| StudentID | Name | CourseID | CourseName |

+-----------+---------+----------+------------------+

| 1 | Alice | 1 | Data Structures |

| 2 | Bob | 2 | Machine Learning |

| 3 | Charlie | 3 | Deep Learning |

+-----------+---------+----------+------------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Update View Example

mysql> UPDATE AdvancedCourses SET Credits = 6 WHERE CourseName = 'Machine Learning';

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql>

mysql> -- Verify Update

mysql> SELECT \* FROM AdvancedCourses;

+------------------+---------+----------+

| CourseName | Credits | Level |

+------------------+---------+----------+

| Machine Learning | 6 | Advanced |

| Deep Learning | 6 | Advanced |

+------------------+---------+----------+

2 rows in set (0.00 sec)

mysql>

mysql> -- Delete Records through View

mysql> DELETE FROM Courses WHERE CourseID IN (

-> SELECT CourseID FROM AdvancedCourses WHERE Credits = 6

-> );

ERROR 1443 (HY000): The definition of table 'AdvancedCourses' prevents operation DELETE on table 'Courses'.

mysql>

mysql> -- Verify Deletion

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 6 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM AdvancedCourses;

+------------------+---------+----------+

| CourseName | Credits | Level |

+------------------+---------+----------+

| Machine Learning | 6 | Advanced |

| Deep Learning | 6 | Advanced |

+------------------+---------+----------+

2 rows in set (0.00 sec)

mysql>

mysql> -- Insert into View

mysql> INSERT INTO CourseDetails (CourseName, InstructorName)

-> VALUES ('Artificial Intelligence', 'John Doe');

ERROR 1393 (HY000): Can not modify more than one base table through a join view 'nptel.coursedetails'

mysql>

mysql> -- Verify Insertion

mysql> SELECT \* FROM CourseDetails;

+------------------+----------------+

| CourseName | InstructorName |

+------------------+----------------+

| Data Structures | John Doe |

| Machine Learning | Jane Smith |

| Deep Learning | Jane Smith |

+------------------+----------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Courses;

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 6 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Update the view using if-else-like condition

mysql> UPDATE StudentCourseView

-> SET CourseName = 'Updated Course Name'

-> WHERE StudentID = 1;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql>

mysql> -- Verify Update

mysql> SELECT \* FROM StudentCourseView;

+-----------+---------+----------+---------------------+

| StudentID | Name | CourseID | CourseName |

+-----------+---------+----------+---------------------+

| 1 | Alice | 1 | Updated Course Name |

| 2 | Bob | 2 | Machine Learning |

| 3 | Charlie | 3 | Deep Learning |

+-----------+---------+----------+---------------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Step 8: Relational Algebra Operations

mysql> -- Cartesian Product Example

mysql> SELECT \* FROM Courses, Instructors;

+----------+---------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+--------------+------------+-------------------------+------------------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID | InstructorID | Name | Qualification | ContactEmail |

+----------+---------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+--------------+------------+-------------------------+------------------------+

| 1 | Updated Course Name | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 | 2 | Jane Smith | MSc in Data Science | jane.smith@example.com |

| 1 | Updated Course Name | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 | 1 | John Doe | PhD in Computer Science | john.doe@example.com |

| 2 | Machine Learning | 16 | 6 | Advanced | Comprehensive course on Machine Learning | English | 2 | 2 | Jane Smith | MSc in Data Science | jane.smith@example.com |

| 2 | Machine Learning | 16 | 6 | Advanced | Comprehensive course on Machine Learning | English | 2 | 1 | John Doe | PhD in Computer Science | john.doe@example.com |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 | 2 | Jane Smith | MSc in Data Science | jane.smith@example.com |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 | 1 | John Doe | PhD in Computer Science | john.doe@example.com |

+----------+---------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+--------------+------------+-------------------------+------------------------+

6 rows in set (0.00 sec)

mysql>

mysql> -- Verify Cartesian Product

mysql> SELECT \* FROM Courses;

+----------+---------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+---------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Updated Course Name | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 6 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+---------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Instructors;

+--------------+------------+-------------------------+------------------------+

| InstructorID | Name | Qualification | ContactEmail |

+--------------+------------+-------------------------+------------------------+

| 1 | John Doe | PhD in Computer Science | john.doe@example.com |

| 2 | Jane Smith | MSc in Data Science | jane.smith@example.com |

+--------------+------------+-------------------------+------------------------+

2 rows in set (0.00 sec)

mysql>

mysql> -- Division Example (Finding students enrolled in all courses)

mysql> SELECT StudentID

-> FROM Enrollment

-> GROUP BY StudentID

-> HAVING COUNT(DISTINCT CourseID) = (SELECT COUNT(CourseID) FROM Courses);

Empty set (0.00 sec)

mysql>

mysql> -- Verify Division

mysql> SELECT \* FROM Enrollment;

+--------------+-----------+----------+------------+

| EnrollmentID | StudentID | CourseID | EnrollDate |

+--------------+-----------+----------+------------+

| 1 | 1 | 1 | 2024-08-01 |

| 2 | 2 | 2 | 2024-08-05 |

| 3 | 3 | 3 | 2024-08-10 |

+--------------+-----------+----------+------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Courses;

+----------+---------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description | Language | InstructorID |

+----------+---------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Updated Course Name | 12 | 4 | Intermediate | Introduction to Data Structures | English | 1 |

| 2 | Machine Learning | 16 | 6 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+---------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Rename Example

mysql> SELECT Name AS InstructorName FROM Instructors;

+----------------+

| InstructorName |

+----------------+

| John Doe |

| Jane Smith |

+----------------+

2 rows in set (0.00 sec)

mysql>

mysql> -- Verify Rename

mysql> SELECT \* FROM Instructors;

+--------------+------------+-------------------------+------------------------+

| InstructorID | Name | Qualification | ContactEmail |

+--------------+------------+-------------------------+------------------------+

| 1 | John Doe | PhD in Computer Science | john.doe@example.com |

| 2 | Jane Smith | MSc in Data Science | jane.smith@example.com |

+--------------+------------+-------------------------+------------------------+

2 rows in set (0.00 sec)

mysql>

mysql> -- Step 9: VDL Commands

mysql> -- Create a view using an original table with all fields

mysql> CREATE VIEW FullStudentDetails AS

-> SELECT \* FROM Students;

Query OK, 0 rows affected (0.01 sec)

mysql>

mysql> -- Verify View Creation

mysql> SELECT \* FROM FullStudentDetails;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

| 4 | David | 25 | Male | david@example.com | NULL |

+-----------+---------+------+--------+---------------------+------------+

4 rows in set (0.00 sec)

mysql>

mysql> -- Create a view from the master table with selected fields only satisfying certain conditions

mysql> CREATE VIEW AdultStudents AS

-> SELECT StudentID, Name, Age

-> FROM Students

-> WHERE Age >= 18;

Query OK, 0 rows affected (0.01 sec)

mysql>

mysql> -- Verify View Creation

mysql> SELECT \* FROM AdultStudents;

+-----------+---------+------+

| StudentID | Name | Age |

+-----------+---------+------+

| 1 | Alice | 22 |

| 2 | Bob | 24 |

| 3 | Charlie | 23 |

| 4 | David | 25 |

+-----------+---------+------+

4 rows in set (0.00 sec)

mysql>

mysql> -- Create a view using nested queries (at least 2 different tables)

mysql> CREATE VIEW ModuleAssignmentDetails AS

-> SELECT Modules.ModuleName, Assignments.AssignmentTitle, Assignments.DueDate

-> FROM Modules

-> JOIN Assignments ON Modules.ModuleID = Assignments.ModuleID;

Query OK, 0 rows affected (0.01 sec)

mysql>

mysql> -- Verify View Creation

mysql> SELECT \* FROM ModuleAssignmentDetails;

+------------------------+--------------------------+------------+

| ModuleName | AssignmentTitle | DueDate |

+------------------------+--------------------------+------------+

| Introduction to Arrays | Array Assignment 1 | 2024-09-01 |

| Linked Lists | Linked List Assignment | 2024-09-10 |

| Supervised Learning | Supervised Learning Quiz | 2024-09-15 |

| Neural Networks | Neural Network Project | 2024-10-01 |

+------------------------+--------------------------+------------+

4 rows in set (0.00 sec)

mysql>

mysql> -- Perform equi-join or full outer join on any 2 tables and make the result of the join operation as a view

mysql> CREATE VIEW InstructorCourseView AS

-> SELECT Instructors.Name AS InstructorName, Courses.CourseName

-> FROM Instructors

-> JOIN Courses ON Instructors.InstructorID = Courses.InstructorID;

Query OK, 0 rows affected (0.01 sec)

mysql>

mysql> -- Verify View Creation

mysql> SELECT \* FROM InstructorCourseView;

+----------------+---------------------+

| InstructorName | CourseName |

+----------------+---------------------+

| John Doe | Updated Course Name |

| Jane Smith | Machine Learning |

| Jane Smith | Deep Learning |

+----------------+---------------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Update a view where only 1 row is updated and verify the updated result both in master table as well as the view

mysql> UPDATE InstructorCourseView SET CourseName = 'Advanced Data Structures'

-> WHERE InstructorName = 'John Doe';

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql>

mysql> -- Verify Update

mysql> SELECT \* FROM InstructorCourseView;

+----------------+--------------------------+

| InstructorName | CourseName |

+----------------+--------------------------+

| John Doe | Advanced Data Structures |

| Jane Smith | Machine Learning |

| Jane Smith | Deep Learning |

+----------------+--------------------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM Courses;

+----------+--------------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| CourseID | CourseName | Duration | Credits | Level | Description

| Language | InstructorID |

+----------+--------------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

| 1 | Advanced Data Structures | 12 | 4 | Intermediate | Introduction to Data Structures

| English | 1 |

| 2 | Machine Learning | 16 | 6 | Advanced | Comprehensive course on Machine Learning | English | 2 |

| 3 | Deep Learning | 20 | 6 | Advanced | In-depth course on Deep Learning techniques | English | 2 |

+----------+--------------------------+----------+---------+--------------+---------------------------------------------+----------+--------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Delete multiple records from the master table and verify the content both in master table as well as the view

mysql> DELETE FROM Students WHERE Age > 24;

Query OK, 1 row affected (0.00 sec)

mysql>

mysql> -- Verify Deletion

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

+-----------+---------+------+--------+---------------------+------------+

3 rows in set (0.00 sec)

mysql> SELECT \* FROM FullStudentDetails;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

+-----------+---------+------+--------+---------------------+------------+

3 rows in set (0.00 sec)

mysql>

mysql> -- Insert at least 3 records in a view which has selected columns; verify the content of both the view and the master table

mysql> INSERT INTO AdultStudents (StudentID, Name, Age)

-> VALUES (4, 'Eve', 28), (5, 'Frank', 30), (6, 'Grace', 26);

Query OK, 3 rows affected (0.00 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql>

mysql> -- Verify Insertion

mysql> SELECT \* FROM AdultStudents;

+-----------+---------+------+

| StudentID | Name | Age |

+-----------+---------+------+

| 1 | Alice | 22 |

| 2 | Bob | 24 |

| 3 | Charlie | 23 |

| 4 | Eve | 28 |

| 5 | Frank | 30 |

| 6 | Grace | 26 |

+-----------+---------+------+

6 rows in set (0.00 sec)

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

| 4 | Eve | 28 | NULL | NULL | NULL |

| 5 | Frank | 30 | NULL | NULL | NULL |

| 6 | Grace | 26 | NULL | NULL | NULL |

+-----------+---------+------+--------+---------------------+------------+

6 rows in set (0.00 sec)

mysql>

mysql> -- Update the view using if-else-like condition

mysql> UPDATE AdultStudents

-> SET Age = CASE WHEN Age < 25 THEN 25 ELSE Age END

-> WHERE StudentID = 4;

Query OK, 0 rows affected (0.00 sec)

Rows matched: 1 Changed: 0 Warnings: 0

mysql>

mysql> -- Verify Update

mysql> SELECT \* FROM AdultStudents;

+-----------+---------+------+

| StudentID | Name | Age |

+-----------+---------+------+

| 1 | Alice | 22 |

| 2 | Bob | 24 |

| 3 | Charlie | 23 |

| 4 | Eve | 28 |

| 5 | Frank | 30 |

| 6 | Grace | 26 |

+-----------+---------+------+

6 rows in set (0.00 sec)

mysql> SELECT \* FROM Students;

+-----------+---------+------+--------+---------------------+------------+

| StudentID | Name | Age | Gender | Email | ContactNo |

+-----------+---------+------+--------+---------------------+------------+

| 1 | Alice | 22 | Female | alice@example.com | 1234567890 |

| 2 | Bob | 24 | Male | bob@example.com | 0987654321 |

| 3 | Charlie | 23 | Male | charlie@example.com | 1122334455 |

| 4 | Eve | 28 | NULL | NULL | NULL |

| 5 | Frank | 30 | NULL | NULL | NULL |

| 6 | Grace | 26 | NULL | NULL | NULL |

+-----------+---------+------+--------+---------------------+------------+

6 rows in set (0.00 sec)

**GitHub link:**