Phase 3_ Schema Definitions and Query Implementations

Table Creation:

Write all SQL statements used to **create** your tables.

```
DROP TABLE Message CASCADE CONSTRAINTS;
DROP TABLE ConversationThread CASCADE CONSTRAINTS;
DROP TABLE Certificate CASCADE CONSTRAINTS;
DROP TABLE Feedback CASCADE CONSTRAINTS;
DROP TABLE Announcement CASCADE CONSTRAINTS;
DROP TABLE Post CASCADE CONSTRAINTS;
DROP TABLE StudyGroupMembership CASCADE CONSTRAINTS;
DROP TABLE StudyGroup CASCADE CONSTRAINTS;
DROP TABLE QuizAttempt CASCADE CONSTRAINTS;
DROP TABLE Submission CASCADE CONSTRAINTS;
DROP TABLE Quiz CASCADE CONSTRAINTS;
DROP TABLE Assignment CASCADE CONSTRAINTS;
DROP TABLE Lecture CASCADE CONSTRAINTS;
DROP TABLE S_Module CASCADE CONSTRAINTS;
DROP TABLE Enrollment CASCADE CONSTRAINTS;
DROP TABLE Course CASCADE CONSTRAINTS;
DROP TABLE Instructor CASCADE CONSTRAINTS;
DROP TABLE EducationalBackground CASCADE CONSTRAINTS;
DROP TABLE S_User CASCADE CONSTRAINTS;

-- Create User Table
CREATE TABLE S_User (
    UserID VARCHAR2(50) PRIMARY KEY,
    Password VARCHAR2(50) NOT NULL,
    EmailAddress VARCHAR2(100) NOT NULL,
    ProfilePicture VARCHAR2(1000)
);
```

```
-- Create EducationalBackground Table
CREATE TABLE EducationalBackground (
    UserID VARCHAR2(50),
    Degree VARCHAR2(100),
    Institution VARCHAR2(100),
    GraduationYear NUMBER(4),
    PRIMARY KEY (UserID, Degree),
    FOREIGN KEY (UserID) REFERENCES S_User(UserID)
);
-- Create Instructor Table
CREATE TABLE Instructor (
    InstructorID VARCHAR2(50) PRIMARY KEY,
    Biography VARCHAR2(2000)
);


-- Create Course Table
CREATE TABLE Course (
    CourseID VARCHAR2(50) PRIMARY KEY,
    Description VARCHAR2(2000) NOT NULL,
    TopicsCovered VARCHAR2(2000),
    InstructorID VARCHAR2(50),
    StartDate DATE NOT NULL,
    EndDate DATE NOT NULL,
    FOREIGN KEY (InstructorID) REFERENCES Instructor(InstructorID)
);


-- Create Enrollment Table
CREATE TABLE Enrollment (
    EnrollmentID VARCHAR2(50) PRIMARY KEY,
    UserID VARCHAR2(50),
    CourseID VARCHAR2(50),
    EnrollmentDate DATE NOT NULL,
    FOREIGN KEY (UserID) REFERENCES S_User(UserID),
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)
);
```

```
-- Create Module Table
CREATE TABLE S_Module (
    ModuleID VARCHAR2(50) PRIMARY KEY,
    CourseID VARCHAR2(50),
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)
);


-- Create Lecture Table
CREATE TABLE Lecture (
    LectureID VARCHAR2(50) PRIMARY KEY,
    ModuleID VARCHAR2(50),
    VideoContent VARCHAR2(1000),
    TextSummary VARCHAR2(1000),
    Duration NUMBER,
    FOREIGN KEY (ModuleID) REFERENCES S_Module(ModuleID)
);


-- Create Assignment Table
CREATE TABLE Assignment (
    AssignmentID VARCHAR2(50) PRIMARY KEY,
    ModuleID VARCHAR2(50),
    SubmissionDeadline DATE,
    MaxMarks NUMBER,
    FOREIGN KEY (ModuleID) REFERENCES S_Module(ModuleID)
);


-- Create Quiz Table
CREATE TABLE Quiz (
    QuizID VARCHAR2(50) PRIMARY KEY,
    ModuleID VARCHAR2(50),
    SubmissionDeadline DATE,
    MaxMarks NUMBER,
    FOREIGN KEY (ModuleID) REFERENCES S_Module(ModuleID)
);


-- Create Submission Table
```

```
CREATE TABLE Submission (
    SubmissionID VARCHAR2(50) PRIMARY KEY,
    UserID VARCHAR2(50),
    AssignmentID VARCHAR2(50),
    MTimestamp DATE NOT NULL,
    FOREIGN KEY (UserID) REFERENCES S_User(UserID),
    FOREIGN KEY (AssignmentID) REFERENCES Assignment(AssignmentID)
);


-- Create QuizAttempt Table
CREATE TABLE QuizAttempt (
    QuizAttemptID VARCHAR2(50) PRIMARY KEY,
    UserID VARCHAR2(50),
    QuizID VARCHAR2(50),
    Score NUMBER,
    FOREIGN KEY (UserID) REFERENCES S_User(UserID),
    FOREIGN KEY (QuizID) REFERENCES Quiz(QuizID)
);


-- Create StudyGroup Table
CREATE TABLE StudyGroup (
    GroupID VARCHAR2(50) PRIMARY KEY
);


-- Create StudyGroupMembership Table
CREATE TABLE StudyGroupMembership (
    GroupID VARCHAR2(50),
    UserID VARCHAR2(50),
    PRIMARY KEY (GroupID, UserID),
    FOREIGN KEY (GroupID) REFERENCES StudyGroup(GroupID),
    FOREIGN KEY (UserID) REFERENCES S_User(UserID)
);


-- Create Post Table
CREATE TABLE Post (
    PostID VARCHAR2(50) PRIMARY KEY,
```

```
    GroupID VARCHAR2(50),

    UserID VARCHAR2(50),

    MTimestamp DATE NOT NULL,

    FOREIGN KEY (GroupID) REFERENCES StudyGroup(GroupID),

    FOREIGN KEY (UserID) REFERENCES S_User(UserID)

);


-- Create Announcement Table

CREATE TABLE Announcement (

    AnnouncementID VARCHAR2(50) PRIMARY KEY,

    CourseID VARCHAR2(50),

    MTimestamp DATE NOT NULL,

    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)

);


-- Create Feedback Table

CREATE TABLE Feedback (

    FeedbackID VARCHAR2(50) PRIMARY KEY,

    UserID VARCHAR2(50),

    CourseID VARCHAR2(50),

    Ratings NUMBER,

    TextComments VARCHAR2(4000),

    FOREIGN KEY (UserID) REFERENCES S_User(UserID),

    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)

);


-- Create Certificate Table

CREATE TABLE Certificate (

    CertificateID VARCHAR2(50) PRIMARY KEY,

    UserID VARCHAR2(50),

    CourseID VARCHAR2(50),

    DateOfIssuance DATE NOT NULL,

    FOREIGN KEY (UserID) REFERENCES S_User(UserID),

    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)

);
```

```
-- Create ConversationThread Table
CREATE TABLE ConversationThread (
    ThreadID VARCHAR2(50) PRIMARY KEY,
    CourseID VARCHAR2(50),
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)
);


-- Create Message Table
CREATE TABLE Message (
    MessageID VARCHAR2(50) PRIMARY KEY,
    SenderID VARCHAR2(50),
    ReceiverID VARCHAR2(50),
    MTimestamp DATE NOT NULL,
    ThreadID VARCHAR2(50),
    FOREIGN KEY (SenderID) REFERENCES S_User(UserID),
    FOREIGN KEY (ReceiverID) REFERENCES S_User(UserID),
    FOREIGN KEY (ThreadID) REFERENCES ConversationThread(ThreadID)
);
```

Database State:

Write all SQL statements used to **insert** values into the database

-- Insert rows into S_User table
INSERT INTO S_User (UserID, Password, EmailAddress, ProfilePicture) VALUES ('user1', 'pass1', 'user1@example.com', 'profile1.jpg');
INSERT INTO S_User (UserID, Password, EmailAddress, ProfilePicture) VALUES ('user2', 'pass2', 'user2@example.com', 'profile2.jpg');
INSERT INTO S_User (UserID, Password, EmailAddress, ProfilePicture) VALUES ('user3', 'pass3', 'user3@example.com', 'profile3.jpg');
INSERT INTO S_User (UserID, Password, EmailAddress, ProfilePicture) VALUES ('user4', 'pass4', 'user4@example.com', 'profile4.jpg');
INSERT INTO S_User (UserID, Password, EmailAddress, ProfilePicture) VALUES ('user5', 'pass5', 'user5@example.com', 'profile5.jpg');
INSERT INTO S_User (UserID, Password, EmailAddress, ProfilePicture) VALUES ('user6', 'pass6', 'user6@example.com', 'profile6.jpg');

-- Insert rows into EducationalBackground table
INSERT INTO EducationalBackground (UserID, Degree, Institution, GraduationYear) VALUES ('user1', 'BSc Computer Science', 'University A', 2020);
INSERT INTO EducationalBackground (UserID, Degree, Institution, GraduationYear) VALUES ('user2', 'BSc Information Technology', 'University B', 2019);
INSERT INTO EducationalBackground (UserID, Degree, Institution, GraduationYear) VALUES ('user3', 'BSc Software Engineering', 'University C', 2021);
INSERT INTO EducationalBackground (UserID, Degree, Institution, GraduationYear) VALUES ('user4', 'MSc Computer Science', 'University A', 2022);
INSERT INTO EducationalBackground (UserID, Degree, Institution, GraduationYear) VALUES ('user5', 'MSc Information Technology', 'University B', 2023);
INSERT INTO EducationalBackground (UserID, Degree, Institution, GraduationYear) VALUES ('user6', 'PhD Computer Science', 'University C', 2024);

-- Insert rows into Instructor table
INSERT INTO Instructor (InstructorID, Biography) VALUES ('inst1', 'Expert in AI and Machine Learning');
INSERT INTO Instructor (InstructorID, Biography) VALUES ('inst2', 'Specialist in Data Science');
INSERT INTO Instructor (InstructorID, Biography) VALUES ('inst3', 'Software Engineering Veteran');
INSERT INTO Instructor (InstructorID, Biography) VALUES ('inst4', 'Cybersecurity Guru');
INSERT INTO Instructor (InstructorID, Biography) VALUES ('inst5', 'Database Systems Expert');
INSERT INTO Instructor (InstructorID, Biography) VALUES ('inst6', 'Web Development Enthusiast');

-- Insert rows into Course table
INSERT INTO Course (CourseID, Description, TopicsCovered, InstructorID, StartDate, EndDate) VALUES ('course1', 'Introduction to AI', 'AI, ML, Neural Networks', 'inst1', TO_DATE('2023-01-01', 'YYYY-MM-DD'), TO_DATE('2023-06-01', 'YYYY-MM-DD'));
INSERT INTO Course (CourseID, Description, TopicsCovered, InstructorID, StartDate, EndDate) VALUES ('course2', 'Data Science Basics', 'Data Analysis, Visualization', 'inst2', TO_DATE('2023-02-01', 'YYYY-MM-DD'), TO_DATE('2023-07-01', 'YYYY-MM-DD'));
INSERT INTO Course (CourseID, Description, TopicsCovered, InstructorID, StartDate, EndDate) VALUES ('course3', 'Advanced Software Engineering', 'Design Patterns, Architecture', 'inst3', TO_DATE('2023-03-01', 'YYYY-MM-DD'), TO_DATE('2023-08-01', 'YYYY-MM-DD'));

INSERT INTO Course (CourseID, Description, TopicsCovered, InstructorID, StartDate, EndDate) VALUES ('course4', 'Cybersecurity Fundamentals', 'Network Security, Encryption', 'inst4', TO_DATE('2023-04-01', 'YYYY-MM-DD'), TO_DATE('2023-09-01', 'YYYY-MM-DD'));
INSERT INTO Course (CourseID, Description, TopicsCovered, InstructorID, StartDate, EndDate) VALUES ('course5', 'Database Systems', 'SQL, NoSQL, Transactions', 'inst5', TO_DATE('2023-05-01', 'YYYY-MM-DD'), TO_DATE('2023-10-01', 'YYYY-MM-DD'));
INSERT INTO Course (CourseID, Description, TopicsCovered, InstructorID, StartDate, EndDate) VALUES ('course6', 'Web Development', 'HTML, CSS, JavaScript', 'inst6', TO_DATE('2023-06-01', 'YYYY-MM-DD'), TO_DATE('2023-11-01', 'YYYY-MM-DD'));

-- Insert rows into Enrollment table
INSERT INTO Enrollment (EnrollmentID, UserID, CourseID, EnrollmentDate) VALUES ('enroll1', 'user1', 'course1', TO_DATE('2023-01-05', 'YYYY-MM-DD'));
INSERT INTO Enrollment (EnrollmentID, UserID, CourseID, EnrollmentDate) VALUES ('enroll2', 'user2', 'course2', TO_DATE('2023-02-05', 'YYYY-MM-DD'));
INSERT INTO Enrollment (EnrollmentID, UserID, CourseID, EnrollmentDate) VALUES ('enroll3', 'user3', 'course3', TO_DATE('2023-03-05', 'YYYY-MM-DD'));
INSERT INTO Enrollment (EnrollmentID, UserID, CourseID, EnrollmentDate) VALUES ('enroll4', 'user4', 'course4', TO_DATE('2023-04-05', 'YYYY-MM-DD'));
INSERT INTO Enrollment (EnrollmentID, UserID, CourseID, EnrollmentDate) VALUES ('enroll5', 'user5', 'course5', TO_DATE('2023-05-05', 'YYYY-MM-DD'));
INSERT INTO Enrollment (EnrollmentID, UserID, CourseID, EnrollmentDate) VALUES ('enroll6', 'user6', 'course6', TO_DATE('2023-06-05', 'YYYY-MM-DD'));

-- Insert rows into S_Module table
INSERT INTO S_Module (ModuleID, CourseID) VALUES ('module1', 'course1');
INSERT INTO S_Module (ModuleID, CourseID) VALUES ('module2', 'course2');
INSERT INTO S_Module (ModuleID, CourseID) VALUES ('module3', 'course3');
INSERT INTO S_Module (ModuleID, CourseID) VALUES ('module4', 'course4');
INSERT INTO S_Module (ModuleID, CourseID) VALUES ('module5', 'course5');
INSERT INTO S_Module (ModuleID, CourseID) VALUES ('module6', 'course6');

-- Insert rows into Lecture table
INSERT INTO Lecture (LectureID, ModuleID, VideoContent, TextSummary, Duration) VALUES ('lecture1', 'module1', 'video1.mp4', 'Summary of lecture 1', 60);
INSERT INTO Lecture (LectureID, ModuleID, VideoContent, TextSummary, Duration) VALUES ('lecture2', 'module2', 'video2.mp4', 'Summary of lecture 2', 45);
INSERT INTO Lecture (LectureID, ModuleID, VideoContent, TextSummary, Duration) VALUES ('lecture3', 'module3', 'video3.mp4', 'Summary of lecture 3', 90);
INSERT INTO Lecture (LectureID, ModuleID, VideoContent, TextSummary, Duration) VALUES ('lecture4', 'module4', 'video4.mp4', 'Summary of lecture 4', 50);
INSERT INTO Lecture (LectureID, ModuleID, VideoContent, TextSummary, Duration) VALUES ('lecture5', 'module5', 'video5.mp4', 'Summary of lecture 5', 30);
INSERT INTO Lecture (LectureID, ModuleID, VideoContent, TextSummary, Duration) VALUES ('lecture6', 'module6', 'video6.mp4', 'Summary of lecture 6', 40);

-- Insert rows into Assignment table
INSERT INTO Assignment (AssignmentID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('assignment1', 'module1', TO_DATE('2023-02-01', 'YYYY-MM-DD'), 100);
INSERT INTO Assignment (AssignmentID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('assignment2', 'module2', TO_DATE('2023-03-01', 'YYYY-MM-DD'), 100);
INSERT INTO Assignment (AssignmentID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('assignment3', 'module3', TO_DATE('2023-04-01', 'YYYY-MM-DD'), 100);

INSERT INTO Assignment (AssignmentID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('assignment4', 'module4', TO_DATE('2023-05-01', 'YYYY-MM-DD'), 100);
INSERT INTO Assignment (AssignmentID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('assignment5', 'module5', TO_DATE('2023-06-01', 'YYYY-MM-DD'), 100);
INSERT INTO Assignment (AssignmentID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('assignment6', 'module6', TO_DATE('2023-07-01', 'YYYY-MM-DD'), 100);

-- Insert rows into Quiz table
INSERT INTO Quiz (QuizID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('quiz1', 'module1', TO_DATE('2023-02-15', 'YYYY-MM-DD'), 50);
INSERT INTO Quiz (QuizID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('quiz2', 'module2', TO_DATE('2023-03-15', 'YYYY-MM-DD'), 50);
INSERT INTO Quiz (QuizID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('quiz3', 'module3', TO_DATE('2023-04-15', 'YYYY-MM-DD'), 50);
INSERT INTO Quiz (QuizID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('quiz4', 'module4', TO_DATE('2023-05-15', 'YYYY-MM-DD'), 50);
INSERT INTO Quiz (QuizID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('quiz5', 'module5', TO_DATE('2023-06-15', 'YYYY-MM-DD'), 50);
INSERT INTO Quiz (QuizID, ModuleID, SubmissionDeadline, MaxMarks) VALUES ('quiz6', 'module6', TO_DATE('2023-07-15', 'YYYY-MM-DD'), 50);

-- Insert rows into Submission table
INSERT INTO Submission (SubmissionID, UserID, AssignmentID, MTimestamp) VALUES ('submission1', 'user1', 'assignment1', TO_DATE('2023-02-01', 'YYYY-MM-DD'));
INSERT INTO Submission (SubmissionID, UserID, AssignmentID, MTimestamp) VALUES ('submission2', 'user2', 'assignment2', TO_DATE('2023-03-01', 'YYYY-MM-DD'));
INSERT INTO Submission (SubmissionID, UserID, AssignmentID, MTimestamp) VALUES ('submission3', 'user3', 'assignment3', TO_DATE('2023-04-01', 'YYYY-MM-DD'));
INSERT INTO Submission (SubmissionID, UserID, AssignmentID, MTimestamp) VALUES ('submission4', 'user4', 'assignment4', TO_DATE('2023-05-01', 'YYYY-MM-DD'));
INSERT INTO Submission (SubmissionID, UserID, AssignmentID, MTimestamp) VALUES ('submission5', 'user5', 'assignment5', TO_DATE('2023-06-01', 'YYYY-MM-DD'));
INSERT INTO Submission (SubmissionID, UserID, AssignmentID, MTimestamp) VALUES ('submission6', 'user6', 'assignment6', TO_DATE('2023-07-01', 'YYYY-MM-DD'));

-- Insert rows into QuizAttempt table
INSERT INTO QuizAttempt (QuizAttemptID, UserID, QuizID, Score) VALUES ('attempt1', 'user1', 'quiz1', 45);
INSERT INTO QuizAttempt (QuizAttemptID, UserID, QuizID, Score) VALUES ('attempt2', 'user2', 'quiz2', 40);
INSERT INTO QuizAttempt (QuizAttemptID, UserID, QuizID, Score) VALUES ('attempt3', 'user3', 'quiz3', 42);
INSERT INTO QuizAttempt (QuizAttemptID, UserID, QuizID, Score) VALUES ('attempt4', 'user4', 'quiz4', 44);
INSERT INTO QuizAttempt (QuizAttemptID, UserID, QuizID, Score) VALUES ('attempt5', 'user5', 'quiz5', 43);
INSERT INTO QuizAttempt (QuizAttemptID, UserID, QuizID, Score) VALUES ('attempt6', 'user6', 'quiz6', 41);

-- Insert rows into StudyGroup table
INSERT INTO StudyGroup (GroupID) VALUES ('group1');
INSERT INTO StudyGroup (GroupID) VALUES ('group2');
INSERT INTO StudyGroup (GroupID) VALUES ('group3');

```
INSERT INTO StudyGroup (GroupID) VALUES ('group4');
INSERT INTO StudyGroup (GroupID) VALUES ('group5');
INSERT INTO StudyGroup (GroupID) VALUES ('group6');

-- Insert rows into StudyGroupMembership table
INSERT INTO StudyGroupMembership (GroupID, UserID) VALUES ('group1', 'user1');
INSERT INTO StudyGroupMembership (GroupID, UserID) VALUES ('group2', 'user2');
INSERT INTO StudyGroupMembership (GroupID, UserID) VALUES ('group3', 'user3');
INSERT INTO StudyGroupMembership (GroupID, UserID) VALUES ('group4', 'user4');
INSERT INTO StudyGroupMembership (GroupID, UserID) VALUES ('group5', 'user5');
INSERT INTO StudyGroupMembership (GroupID, UserID) VALUES ('group6', 'user6');

-- Insert rows into Post table
INSERT INTO Post (PostID, GroupID, UserID, MTimestamp) VALUES ('post1', 'group1', 'user1', TO_DATE('2023-01-01', 'YYYY-MM-DD'));
INSERT INTO Post (PostID, GroupID, UserID, MTimestamp) VALUES ('post2', 'group2', 'user2', TO_DATE('2023-02-01', 'YYYY-MM-DD'));
INSERT INTO Post (PostID, GroupID, UserID, MTimestamp) VALUES ('post3', 'group3', 'user3', TO_DATE('2023-03-01', 'YYYY-MM-DD'));
INSERT INTO Post (PostID, GroupID, UserID, MTimestamp) VALUES ('post4', 'group4', 'user4', TO_DATE('2023-04-01', 'YYYY-MM-DD'));
INSERT INTO Post (PostID, GroupID, UserID, MTimestamp) VALUES ('post5', 'group5', 'user5', TO_DATE('2023-05-01', 'YYYY-MM-DD'));
INSERT INTO Post (PostID, GroupID, UserID, MTimestamp) VALUES ('post6', 'group6', 'user6', TO_DATE('2023-06-01', 'YYYY-MM-DD'));

-- Insert rows into Announcement table
INSERT INTO Announcement (AnnouncementID, CourseID, MTimestamp) VALUES ('ann1', 'course1', TO_DATE('2023-01-01', 'YYYY-MM-DD'));
INSERT INTO Announcement (AnnouncementID, CourseID, MTimestamp) VALUES ('ann2', 'course2', TO_DATE('2023-02-01', 'YYYY-MM-DD'));
INSERT INTO Announcement (AnnouncementID, CourseID, MTimestamp) VALUES ('ann3', 'course3', TO_DATE('2023-03-01', 'YYYY-MM-DD'));
INSERT INTO Announcement (AnnouncementID, CourseID, MTimestamp) VALUES ('ann4', 'course4', TO_DATE('2023-04-01', 'YYYY-MM-DD'));
INSERT INTO Announcement (AnnouncementID, CourseID, MTimestamp) VALUES ('ann5', 'course5', TO_DATE('2023-05-01', 'YYYY-MM-DD'));
INSERT INTO Announcement (AnnouncementID, CourseID, MTimestamp) VALUES ('ann6', 'course6', TO_DATE('2023-06-01', 'YYYY-MM-DD'));

-- Insert rows into Feedback table
INSERT INTO Feedback (FeedbackID, UserID, CourseID, Ratings, TextComments) VALUES ('fb1', 'user1', 'course1', 5, 'Great course!');
INSERT INTO Feedback (FeedbackID, UserID, CourseID, Ratings, TextComments) VALUES ('fb2', 'user2', 'course2', 4, 'Very informative');
INSERT INTO Feedback (FeedbackID, UserID, CourseID, Ratings, TextComments) VALUES ('fb3', 'user3', 'course3', 5, 'Loved it');
INSERT INTO Feedback (FeedbackID, UserID, CourseID, Ratings, TextComments) VALUES ('fb4', 'user4', 'course4', 3, 'Good, but could be better');
INSERT INTO Feedback (FeedbackID, UserID, CourseID, Ratings, TextComments) VALUES ('fb5', 'user5', 'course5', 4, 'Well explained');
INSERT INTO Feedback (FeedbackID, UserID, CourseID, Ratings, TextComments) VALUES ('fb6', 'user6', 'course6', 5, 'Excellent content');
```

```
-- Insert rows into Certificate table
INSERT INTO Certificate (CertificateID, UserID, CourseID, DateOfIssuance) VALUES ('cert1', 'user1', 'course1', TO_DATE('2023-06-01', 'YYYY-MM-DD'));
INSERT INTO Certificate (CertificateID, UserID, CourseID, DateOfIssuance) VALUES ('cert2', 'user2', 'course2', TO_DATE('2023-07-01', 'YYYY-MM-DD'));
INSERT INTO Certificate (CertificateID, UserID, CourseID, DateOfIssuance) VALUES ('cert3', 'user3', 'course3', TO_DATE('2023-08-01', 'YYYY-MM-DD'));
INSERT INTO Certificate (CertificateID, UserID, CourseID, DateOfIssuance) VALUES ('cert4', 'user4', 'course4', TO_DATE('2023-09-01', 'YYYY-MM-DD'));
INSERT INTO Certificate (CertificateID, UserID, CourseID, DateOfIssuance) VALUES ('cert5', 'user5', 'course5', TO_DATE('2023-10-01', 'YYYY-MM-DD'));
INSERT INTO Certificate (CertificateID, UserID, CourseID, DateOfIssuance) VALUES ('cert6', 'user6', 'course6', TO_DATE('2023-11-01', 'YYYY-MM-DD'));

-- Insert rows into ConversationThread table
INSERT INTO ConversationThread (ThreadID, CourseID) VALUES ('thread1', 'course1');
INSERT INTO ConversationThread (ThreadID, CourseID) VALUES ('thread2', 'course2');
INSERT INTO ConversationThread (ThreadID, CourseID) VALUES ('thread3', 'course3');
INSERT INTO ConversationThread (ThreadID, CourseID) VALUES ('thread4', 'course4');
INSERT INTO ConversationThread (ThreadID, CourseID) VALUES ('thread5', 'course5');
INSERT INTO ConversationThread (ThreadID, CourseID) VALUES ('thread6', 'course6');

-- Insert rows into Message table
INSERT INTO Message (MessageID, SenderID, ReceiverID, MTimestamp, ThreadID) VALUES ('msg1', 'user1', 'user2', TO_DATE('2023-01-01', 'YYYY-MM-DD'), 'thread1');
INSERT INTO Message (MessageID, SenderID, ReceiverID, MTimestamp, ThreadID) VALUES ('msg2', 'user2', 'user3', TO_DATE('2023-02-01', 'YYYY-MM-DD'), 'thread2');
INSERT INTO Message (MessageID, SenderID, ReceiverID, MTimestamp, ThreadID) VALUES ('msg3', 'user3', 'user4', TO_DATE('2023-03-01', 'YYYY-MM-DD'), 'thread3');
INSERT INTO Message (MessageID, SenderID, ReceiverID, MTimestamp, ThreadID) VALUES ('msg4', 'user4', 'user5', TO_DATE('2023-04-01', 'YYYY-MM-DD'), 'thread4');
INSERT INTO Message (MessageID, SenderID, ReceiverID, MTimestamp, ThreadID) VALUES ('msg5', 'user5', 'user6', TO_DATE('2023-05-01', 'YYYY-MM-DD'), 'thread5');
INSERT INTO Message (MessageID, SenderID, ReceiverID, MTimestamp, ThreadID) VALUES ('msg6', 'user6', 'user1', TO_DATE('2023-06-01', 'YYYY-MM-DD'), 'thread6');
```

Table Screenshot:

List all your table **instance/data**, after the insertion process take screenshot from each table.

```
6 rows selected.

SQL> select* from Quiz
  2  ;

QUIZID                                             MODULEID                           SUBMISSIO  MAXMARKS
-------------------------------------------------- ----------------------------       ---------  --------
quiz1                                              module1                            15-FEB-23        50
quiz2                                              module2                            15-MAR-23        50
quiz3                                              module3                            15-APR-23        50
quiz4                                              module4                            15-MAY-23        50
quiz5                                              module5                            15-JUN-23        50
quiz6                                              module6                            15-JUL-23        50

6 rows selected.

SQL> select* from Assignment;

ASSIGNMENTID                                       MODULEID                           SUBMISSIO  MAXMARKS
-------------------------------------------------- ----------------------------       ---------  --------
assignment1                                        module1                            01-FEB-23       100
assignment2                                        module2                            01-MAR-23       100
assignment3                                        module3                            01-APR-23       100
assignment4                                        module4                            01-MAY-23       100
assignment5                                        module5                            01-JUN-23       100
assignment6                                        module6                            01-JUL-23       100

6 rows selected.

SQL> select* from Lecture;

LECTUREID                                          MODULEID
-------------------------------------------------- ----------------------------
VIDEOCONTENT
---------------------------------------------------------------------------------------
TEXTSUMMARY
---------------------------------------------------------------------------------------
  DURATION
----------
lecture1                                           module1
video1.mp4
Summary of lecture 1
        60


LECTUREID                                          MODULEID
-------------------------------------------------- ----------------------------
VIDEOCONTENT
---------------------------------------------------------------------------------------
TEXTSUMMARY
```

```
SQL> select * from StudyGroup;

GROUPID
-------------------------------------------------
group1
group2
group3
group4
group5
group6

6 rows selected.

SQL> select* from QuizAttempt;

QUIZATTEMPTID                                      USERID                             QUIZID                                SCORE
-------------------------------------------------- ----------------------------       ---------------------------------     -----
attempt1                                           user1                              quiz1                                    45
attempt2                                           user2                              quiz2                                    40
attempt3                                           user3                              quiz3                                    42
attempt4                                           user4                              quiz4                                    44
attempt5                                           user5                              quiz5                                    43
attempt6                                           user6                              quiz6                                    41

6 rows selected.

SQL> select* from Submission;

SUBMISSIONID                                       USERID                             ASSIGNMENTID                          MTIMESTAM
-------------------------------------------------- ----------------------------       ---------------------------------     ---------
submission1                                        user1                              assignment1                           01-FEB-23
submission2                                        user2                              assignment2                           01-MAR-23
submission3                                        user3                              assignment3                           01-APR-23
submission4                                        user4                              assignment4                           01-MAY-23
submission5                                        user5                              assignment5                           01-JUN-23
submission6                                        user6                              assignment6                           01-JUL-23

6 rows selected.

SQL> select* from Quiz
  2  ;

QUIZID                                             MODULEID                           SUBMISSIO  MAXMARKS
-------------------------------------------------- ----------------------------       ---------  --------
quiz1                                              module1                            15-FEB-23        50
quiz2                                              module2                            15-MAR-23        50
quiz3                                              module3                            15-APR-23        50
quiz4                                              module4                            15-MAY-23        50
quiz5                                              module5                            15-JUN-23        50
quiz6                                              module6                            15-JUL-23        50

6 rows selected.
```

Query Implementation:

Views list:

```
1.
2.  -- 1-View for User Enrollment Details: This view combines
    information f
3.  --rom the S_User, Enrollment, and Course tables to show which users
    are enrolled in which courses.
4.

5.  CREATE VIEW UserEnrollmentDetails AS
6.  SELECT
7.      u.UserID,
8.      u.EmailAddress,
9.      c.CourseID,
10.     c.Description,
11.     e.EnrollmentDate
12. FROM
13.     S_User u
14. JOIN
15.     Enrollment e ON u.UserID = e.UserID
```

```
16. JOIN
17.     Course c ON e.CourseID = c.CourseID;
18.
19.

20. select * from UserEnrollmentDetails;
21.

22. --2- View for Course Instructors: This view combines information
    from the Course and Instructor tables
23. --to show which instructor is teaching which course.
24.
25. CREATE VIEW CourseInstructors AS
26. SELECT
27.     c.CourseID,
28.     c.Description AS CourseDescription,
29.     i.InstructorID,
30.     i.Biography AS InstructorBiography
31. FROM
32.     Course c
33. JOIN
34.     Instructor i ON c.InstructorID = i.InstructorID;
35.

36. select * from CourseInstructors;
```
37. –

38. -

Queries list:

```
1.

2.  --1-Query to find all courses a specific user is enrolled in:
3.  SELECT
4.      c.CourseID,
5.      c.Description
6.  FROM
7.      UserEnrollmentDetails ued
8.  JOIN
9.      Course c ON ued.CourseID = c.CourseID
10. WHERE
11.     ued.UserID = 'user1';
12.

13. --2-Query to find all users enrolled in a specific course:
14.
```

```sql
15. SELECT
16.     u.UserID,
17.     u.EmailAddress
18. FROM
19.     UserEnrollmentDetails ued
20. JOIN
21.     S_User u ON ued.UserID = u.UserID
22. WHERE
23.     ued.CourseID = 'course1';
24.
25. --3-Query to find the instructor of a specific course:
26. SELECT
27.     ci.InstructorID,
28.     ci.InstructorBiography
29. FROM
30.     CourseInstructors ci
31. WHERE
32.     ci.CourseID = 'course1';
33.
34. --4-Query to list all courses along with their instructors:
35.
36. SELECT
37.     ci.CourseID,
38.     ci.CourseDescription,
39.     ci.InstructorID,
40.     ci.InstructorBiography
41. FROM
42.     CourseInstructors ci;
43.
44. --5-Query to find the details of a user's educational background:
45. SELECT
46.     eb.UserID,
47.     eb.Degree,
48.     eb.Institution,
49.     eb.GraduationYear
50. FROM
51.     EducationalBackground eb
52. WHERE
53.     eb.UserID = 'user1';
54. --6-Query to find all users who have submitted assignments for a
    specific module:
55. SELECT
56.     s.UserID,
57.     s.SubmissionID,
```

```
58.    s.MTimestamp
59.FROM
60.    Submission s
61.JOIN
62.    Assignment a ON s.AssignmentID = a.AssignmentID
63.WHERE
64.    a.ModuleID = 'module1';
65.
66.--7-Query to get the highest score in a specific quiz:
67.
68.SELECT
69.    MAX(qa.Score) AS HighestScore
70.FROM
71.    QuizAttempt qa
72.WHERE
73.    qa.QuizID = 'quiz1';
74.
75.--8-Query to list all announcements made for a specific course:
76.SELECT
77.    a.AnnouncementID,
78.    a.MTimestamp
79.FROM
80.    Announcement a
81.WHERE
82.    a.CourseID = 'course1';
83.
```
84. -

85. -

Createsh the result of **two views** and **eight queries** from the above lists.

```
SQL>
SQL> CREATE VIEW CourseInstructors AS
  2  SELECT
  3      c.CourseID,
  4      c.Description AS CourseDescription,
  5      i.InstructorID,
  6      i.Biography AS InstructorBiography
  7  FROM
  8      Course c
  9  JOIN
 10      Instructor i ON c.InstructorID = i.InstructorID;

View created.

SQL>
SQL>
SQL> select * from CourseInstructors;

COURSEID
----------------------------------------
COURSEDESCRIPTION
----------------------------------------------------------------
INSTRUCTORID
----------------------------------------
INSTRUCTORBIOGRAPHY
----------------------------------------------------------------
course1
Introduction to AI
inst1
Expert in AI and Machine Learning


COURSEID
----------------------------------------
COURSEDESCRIPTION
----------------------------------------------------------------
INSTRUCTORID
----------------------------------------
INSTRUCTORBIOGRAPHY
----------------------------------------------------------------
course2
Data Science Basics
inst2
Specialist in Data Science


COURSEID
----------------------------------------
COURSEDESCRIPTION
```

```
COURSEID
----------------------------------------
COURSEDESCRIPTION
----------------------------------------------------------------
INSTRUCTORID
----------------------------------------
INSTRUCTORBIOGRAPHY
----------------------------------------------------------------
course3
Advanced Software Engineering
inst3
Software Engineering Veteran


COURSEID
----------------------------------------
COURSEDESCRIPTION
----------------------------------------------------------------
INSTRUCTORID
----------------------------------------
INSTRUCTORBIOGRAPHY
----------------------------------------------------------------
course4
Cybersecurity Fundamentals
inst4
Cybersecurity Guru


COURSEID
----------------------------------------
COURSEDESCRIPTION
----------------------------------------------------------------
INSTRUCTORID
----------------------------------------
INSTRUCTORBIOGRAPHY
----------------------------------------------------------------
course5
Database Systems
inst5
Database Systems Expert


COURSEID
----------------------------------------
COURSEDESCRIPTION
----------------------------------------------------------------
INSTRUCTORID
----------------------------------------
INSTRUCTORBIOGRAPHY
----------------------------------------------------------------
```

```
SQL> --3-Query to find the instructor of a specific course:
SQL> SELECT
  2      ci.InstructorID,
  3      ci.InstructorBiography
  4  FROM
  5      CourseInstructors ci
  6  WHERE
  7      ci.CourseID = 'course1';

INSTRUCTORID
-----------------------------------------------------
INSTRUCTORBIOGRAPHY
-----------------------------------------------------------------------------
inst1
Expert in AI and Machine Learning


SQL>
SQL>
SQL> --4-Query to list all courses along with their instructors:
SQL>
SQL> SELECT
  2      ci.CourseID,
  3      ci.CourseDescription,
  4      ci.InstructorID,
  5      ci.InstructorBiography
  6  FROM
  7      CourseInstructors ci;

COURSEID
-----------------------------------------------------
COURSEDESCRIPTION
-----------------------------------------------------------------------------
INSTRUCTORID
-----------------------------------------------------
INSTRUCTORBIOGRAPHY
-----------------------------------------------------------------------------
course1
Introduction to AI
inst1
Expert in AI and Machine Learning


COURSEID
-----------------------------------------------------
COURSEDESCRIPTION
-----------------------------------------------------------------------------
INSTRUCTORID
-----------------------------------------------------
INSTRUCTORBIOGRAPHY
```

```
COURSEID
-----------------------------------------------------
COURSEDESCRIPTION
-----------------------------------------------------------------------------
INSTRUCTORID
-----------------------------------------------------
INSTRUCTORBIOGRAPHY
-----------------------------------------------------------------------------
course2
Data Science Basics
inst2
Specialist in Data Science


COURSEID
-----------------------------------------------------
COURSEDESCRIPTION
-----------------------------------------------------------------------------
INSTRUCTORID
-----------------------------------------------------
INSTRUCTORBIOGRAPHY
-----------------------------------------------------------------------------
course3
Advanced Software Engineering
inst3
Software Engineering Veteran


COURSEID
-----------------------------------------------------
COURSEDESCRIPTION
-----------------------------------------------------------------------------
INSTRUCTORID
-----------------------------------------------------
INSTRUCTORBIOGRAPHY
-----------------------------------------------------------------------------
course4
Cybersecurity Fundamentals
inst4
Cybersecurity Guru


COURSEID
-----------------------------------------------------
COURSEDESCRIPTION
-----------------------------------------------------------------------------
INSTRUCTORID
-----------------------------------------------------
INSTRUCTORBIOGRAPHY
-----------------------------------------------------------------------------
```

```
COURSEDESCRIPTION
---------------------------------------------------------------------
INSTRUCTORID
---------------------------------------------------------------------
INSTRUCTORBIOGRAPHY
---------------------------------------------------------------------
course5
Database Systems
inst5
Database Systems Expert


COURSEID
---------------------------------------------------
COURSEDESCRIPTION
---------------------------------------------------------------------
INSTRUCTORID
---------------------------------------------------------------------
INSTRUCTORBIOGRAPHY
---------------------------------------------------------------------
course6
Web Development
inst6
Web Development Enthusiast


6 rows selected.

SQL>
SQL> --5-Query to find the details of a user's educational background:
SQL> SELECT
  2        eb.UserID,
  3        eb.Degree,
  4        eb.Institution,
  5        eb.GraduationYear
  6  FROM
  7      EducationalBackground eb
  8  WHERE
  9      eb.UserID = 'user1';

USERID
---------------------------------------------------
DEGREE
---------------------------------------------------------------------
INSTITUTION
---------------------------------------------------------------------
GRADUATIONYEAR
---------------
user1
```



```
SQL>
SQL> --5-Query to find the details of a user's educational background:
SQL> SELECT
  2        eb.UserID,
  3        eb.Degree,
  4        eb.Institution,
  5        eb.GraduationYear
  6  FROM
  7      EducationalBackground eb
  8  WHERE
  9      eb.UserID = 'user1';

USERID
---------------------------------------------------
DEGREE
---------------------------------------------------------------------
INSTITUTION
---------------------------------------------------------------------
GRADUATIONYEAR
---------------
user1
BSc Computer Science
University A
         2020


SQL> --6-Query to find all users who have submitted assignments for a specific module:
SQL> SELECT
  2        s.UserID,
  3        s.SubmissionID,
  4        s.MTimestamp
  5  FROM
  6      Submission s
  7  JOIN
  8      Assignment a ON s.AssignmentID = a.AssignmentID
  9  WHERE
 10      a.ModuleID = 'module1';

USERID
---------------------------------------------------
SUBMISSIONID                                        MTIMESTAM
--------------------------------------------------- ---------
user1
submission1                                         01-FEB-23


SQL>
SQL> --7-Query to get the highest score in a specific quiz:
SQL>
SQL> SELECT
```