**This schema represents a basic Online Course Management System. Answer questions 1 - 5 based on this schema.**

```
CREATE TABLE Instructor (
    InstructorID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL UNIQUE,
    Phone VARCHAR(15),
    Department VARCHAR(50)
);

CREATE TABLE Course (
    CourseID INT AUTO_INCREMENT PRIMARY KEY,
    Title VARCHAR(255) NOT NULL,
    Credits INT NOT NULL,
    InstructorID INT,
    FOREIGN KEY (InstructorID) REFERENCES Instructor(InstructorID)
);

CREATE TABLE Enrollment (
    EnrollmentID INT AUTO_INCREMENT PRIMARY KEY,
    StudentID INT,
    CourseID INT,
    EnrollmentDate DATE NOT NULL,
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
    FOREIGN KEY (CourseID) REFERENCES Course(CourseID)
);

CREATE TABLE Student (
    StudentID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL UNIQUE,
    Phone VARCHAR(15)
);
INSERT INTO Instructor (Name, Email, Phone, Department)
VALUES
('Alice Johnson', 'alice.johnson@example.com', '555-101-2020', 'Computer Science'),
('Bob Smith', 'bob.smith@example.com', '555-202-3030', 'Mathematics'),
('Carol Davis', 'carol.davis@example.com', '555-303-4040', 'Physics');
```

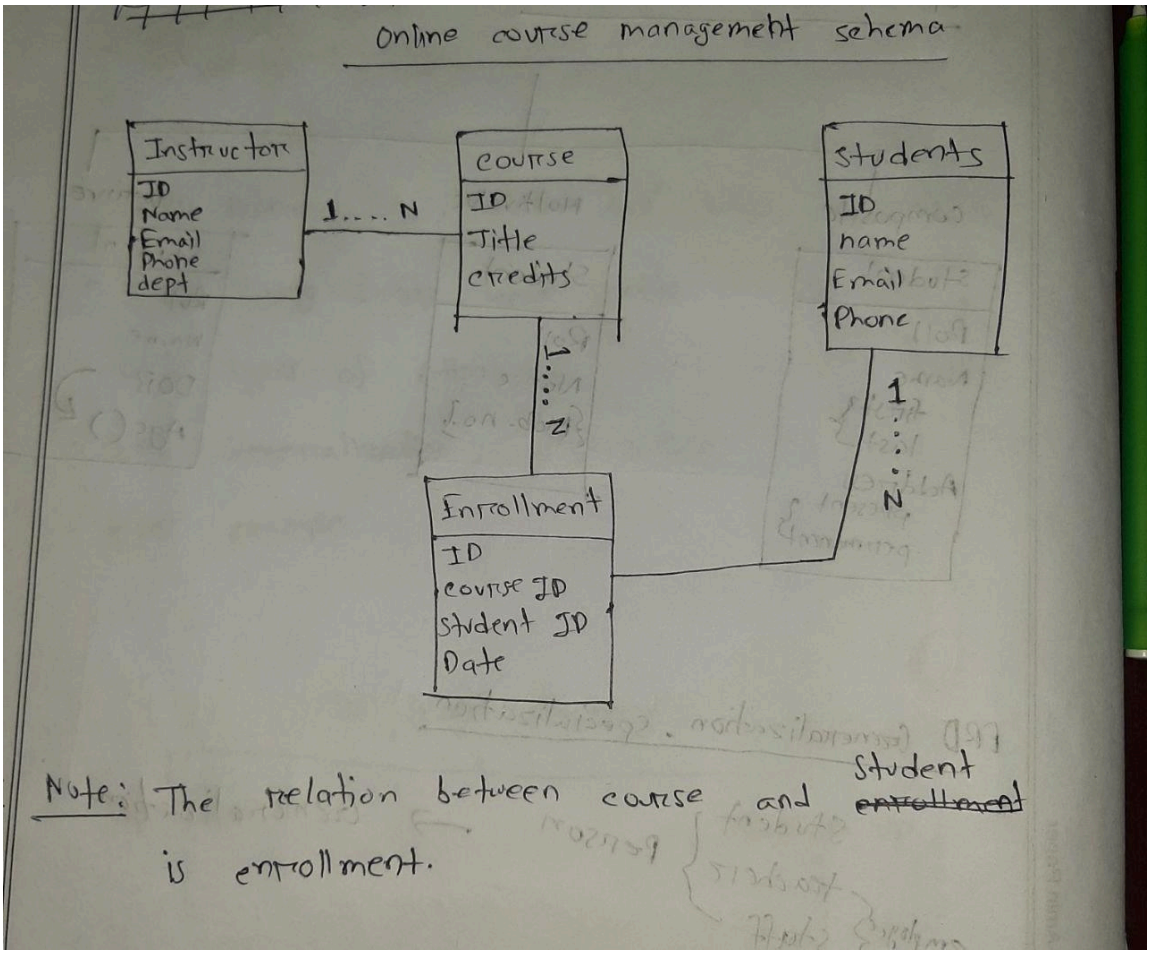
```
INSERT INTO Course (Title, Credits, InstructorID)
VALUES
('Introduction to SQL', 3, 1),
('Data Structures', 4, 2),
('Machine Learning', 5, 1),
('Discrete Mathematics', 3, 2),
('Quantum Physics', 5, 3);

INSERT INTO Student (Name, Email, Phone)
VALUES
('John Doe', 'john.doe@example.com', '123-456-7890'),
('Jane Smith', 'jane.smith@example.com', '234-567-8901'),
('Samuel Brown', 'samuel.brown@example.com', '345-678-9012'),
('Emily White', 'emily.white@example.com', '456-789-0123'),
('Michael Green', 'michael.green@example.com', '567-890-1234');

INSERT INTO Enrollment (StudentID, CourseID, EnrollmentDate)
VALUES
(1, 1, '2024-11-01'),
(2, 2, '2024-11-02'),
(3, 3, '2024-11-03'),
(4, 4, '2024-11-04'),
(5, 5, '2024-11-05');
```

# Answer Script

# (i have inserted the data above)

| Question No. 01 |
| --- |
| Draw an Entity-Relationship (ER) diagram to represent this Online Course Management System schema. |

| Answer No. 01 |
| --- |




| Question No. 02 |
| --- |
| Write an SQL query to insert a new enrollment record for a student (e.g., StudentID 5) into the course with the highest credit hours. |

## Answer No. 02

```
INSERT INTO Enrollment (StudentID, CourseID, EnrollmentDate)
SELECT 5, CourseID, '2024-11-20'
FROM Course
ORDER BY Credits DESC
LIMIT 1;
```

## Question No. 03

Write an SQL UPDATE query to assign a new instructor to a course (e.g., CourseID 3) by updating the InstructorID.

## Answer No.3

```
UPDATE Course
SET InstructorID = 2
WHERE CourseID = 3;
```

## Question No. 04

Write an SQL query to find the names of instructors who teach the most credits (total).

## Answer No. 04

```
SELECT Instructor.Name
FROM Instructor
JOIN Course ON Instructor.InstructorID = Course.InstructorID
GROUP BY Instructor.InstructorID
ORDER BY SUM(Course.Credits) DESC
LIMIT 1;
```

## Question No. 5

Write an SQL query to list all students who are enrolled in more than two courses.

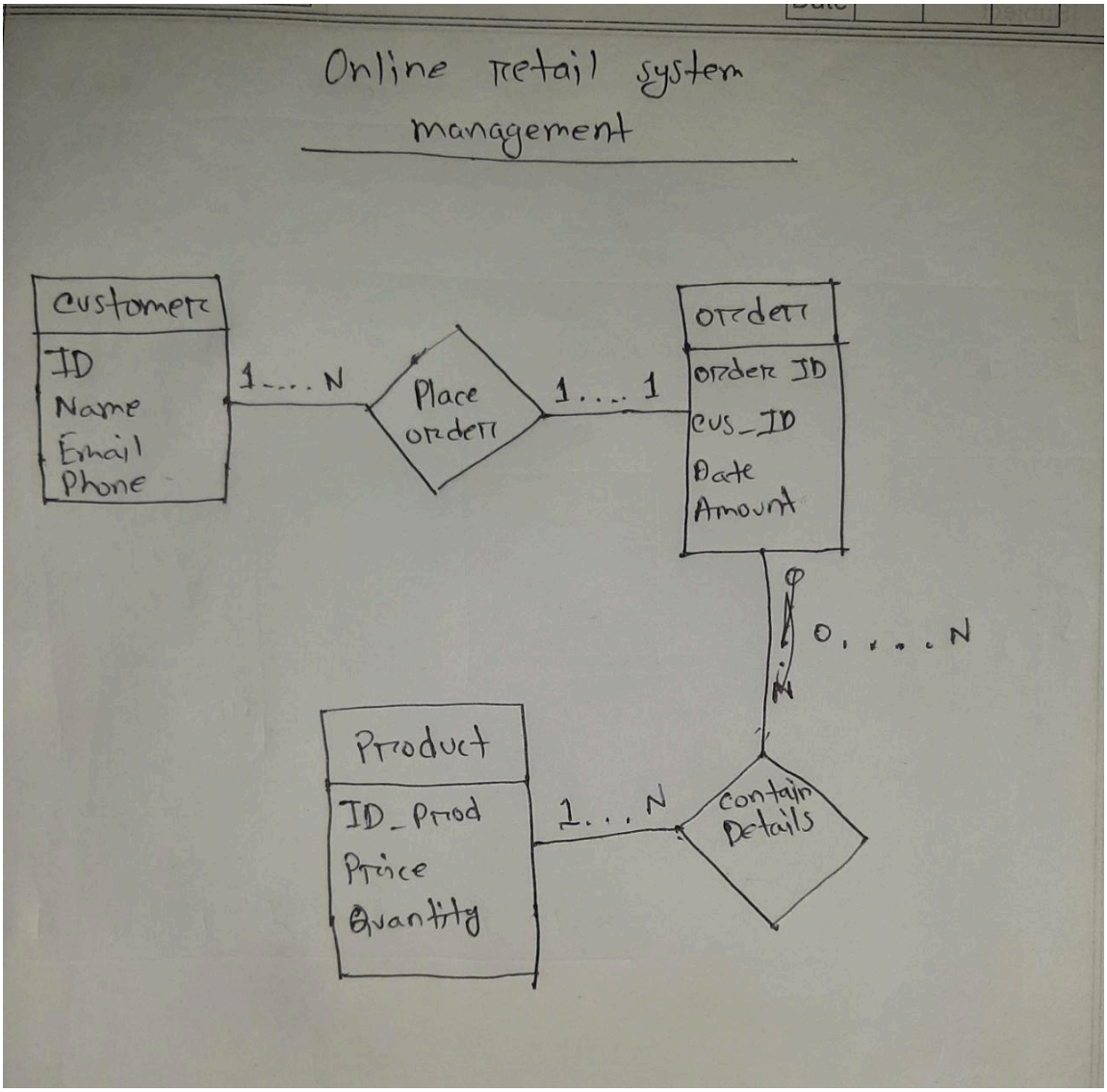
## Answer No. 05

```
SELECT Student.Name
FROM Student
JOIN Enrollment ON Student.StudentID = Enrollment.StudentID
GROUP BY Student.StudentID, Student.Name
HAVING COUNT(Enrollment.CourseID) > 2;
```

## Question No. 06

Design an ER diagram for a simple online retail system that includes entities such as Customers, Products, and Orders. Keep the diagram simple.

## Answer No. 06




| Question No. 07 |
| --- |
| Explain the difference between GROUP BY and ORDER BY in SQL. Provide an example for each to illustrate |
| **Answer No. 07** |
| Differences between group by and order by is given below:<br><br>**<u>Group by:</u>** |

Group by is a function by which we can aggregate the values of a group, we can use it in while using sum(), count(), avg() functions.

Here is an example : suppose we want to find the total student enrolled in a couse ,
SELECT CourseID, COUNT(StudentID) AS TotalStudents
FROM Enrollment
GROUP BY CourseID;.
Here what it does is, it groups the values by courseID, then count each student ,then shows the values as output

**Order by:**

On the other hand order by is used to sort the data we have like in alphabetical or numerical order. We can sort it in ascending or descending order, or also we can sort names of people in ascending or descending order.

Suppose we want to see the names of all student in ascending order. Here is an example :

```
SELECT Name
FROM Student
ORDER BY Name ASC;
```

## Question No. 08

Given a table Instructor with a Salary column, write an SQL query to find the second-highest salary among instructors.

## Answer No. 08

```
SELECT Salary
FROM Instructor
ORDER BY Salary DESC
LIMIT 1 OFFSET 1;
```

## Question No. 09

You have two tables, Instructor and Course. Use ON DELETE CASCADE on Course so that all courses are deleted when an instructor is removed.

## Answer No. 9

```
CREATE TABLE Course (
   CourseID INT AUTO_INCREMENT PRIMARY KEY,
   Title VARCHAR(255) NOT NULL,
   Credits INT NOT NULL,
   InstructorID INT,
   FOREIGN KEY (InstructorID) REFERENCES Instructor(InstructorID)
   ON DELETE CASCADE
);
```

## Question No. 10

Describe the most challenging topic you encountered in this course. Explain why it was challenging and how you overcame it.

## Answer No. 10

Advanced subquery was a bit of a struggle. It was challenging to me because I had a hard time visualizing the connections, but I overcame it by practicing it with active recalling.