```
CREATE TABLE Student (
    StudentID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL,
    Phone VARCHAR(15),
    Address TEXT
);
```

StudentID	Name	Email	Phone	Address

```
CREATE TABLE Book (
    ISBN VARCHAR(13) PRIMARY KEY,
    Title VARCHAR(255) NOT NULL,
    Author VARCHAR(255) NOT NULL,
    Genre VARCHAR(50),
    TotalCopies INT NOT NULL,
    AvailableCopies INT NOT NULL
);
```

ISBN	Title	Author	Genre	TotalCopies	AvailableCopies

```
CREATE TABLE Borrowing (
    BorrowID INT AUTO_INCREMENT PRIMARY KEY,
    StudentID INT,
    ISBN VARCHAR(13),
    BorrowDate DATE NOT NULL,
    DueDate DATE NOT NULL,
    ReturnDate DATE,
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
    FOREIGN KEY (ISBN) REFERENCES Book(ISBN)
);
```

BorrowID	StudentID	ISBN	BorrowDate	DueDate	ReturnDate

Answer::

StudentID	Name	Email	Phone	Address

ISBN	Title	Author	Genre	TotalCopies	AvailableCopies

BorrowID	StudentID	ISBN	BorrowDate	DueDate	ReturnDate

- ☐ Entities: Student, Book, Borrowing
- ☐ Attributes: As defined in each CREATE TABLE statement.
- ☐ Relationships: Borrowing links Student and Book.

This diagram would show Student and Book as entities connected by Borrowing, which is a many-to-many relationship since a student can borrow many books and a book can be borrowed by many students. The Borrowing entity would have BorrowDate, DueDate, and ReturnDate as attributes, with foreign keys from both Student and Book.

2.Insert a new borrowing record for a student (e.g., StudentID 3) for a book with the most available copies.

Answer ::

```
INSERT INTO Borrowing (StudentID, ISBN, BorrowDate, DueDate)
VALUES (3, '1234567890123', CURDATE(), DATE_ADD(CURDATE(), INTERVAL 14 DAY));
```

3. Using Update Query, decrease the available copies of a book (e.g., ISBN '9781234567890') by 1 when a student borrows it.

Answer ::

```
UPDATE Book
SET AvailableCopies = AvailableCopies - 1
WHERE ISBN = '9781234567890';
```

4.Retrieve the names of students who have borrowed the most books.

Answer ::

```
SELECT s.Name
FROM Student s
JOIN (SELECT StudentID, COUNT(*) AS BooksBorrowed
    FROM Borrowing
    GROUP BY StudentID
    ORDER BY BooksBorrowed DESC
```

```
LIMIT 1) AS bb ON s.StudentID = bb.StudentID;
```

5.Retrieve the books that are overdue (i.e., the return date is before the current date).

Answer ::

```
SELECT b.Title
FROM Book b
JOIN Borrowing br ON b.ISBN = br.ISBN
WHERE br.DueDate < CURDATE() AND (br.ReturnDate IS NULL OR br.ReturnDate >
br.DueDate);
```

6.You want to make a mobile banking platform for sending and receiving money from your friends. Make an ERD of this system. (Keep it simple)

Answer ::

For a simple mobile banking system for sending and receiving money:

Entities: User, Transaction
Attributes: User(ID, Name, Email), Transaction(TransID, SenderID, ReceiverID, Amount, TransDate)
Relationships: Each Transaction involves two Users (a sender and a receiver), indicating a many-to-many relationship but represented with two one-to-many relationships because each transaction is a separate entity.

7.Explain UNION and UNION ALL set operations in MySQL

Answer ::

UNION: only keeps unique records.
UNION ALL: keeps all records, including duplicates.

8. There is a table named Employee. In that table there is a field named Salary. Determine the second lowest salary.

Answer ::

```
SELECT MIN(Salary) AS SecondLowestSalary
FROM Employee
WHERE Salary > (SELECT MIN(Salary) FROM Employee);
```

9.There are tables named Employee, Job History, Department.

a. Use ON DELETE CASCADE on Job History for deleting Employee
b. Use ON DELETE SET NULL on Employee for deleting Department

Answer ::

a. ON DELETE CASCADE on Job History for Employee: When an employee is deleted, automatically delete their job history records.
b. ON DELETE SET NULL on Employee for Department: When a department is deleted, set the department field in the employee records to NULL.

10. In this course, which topic you found most interesting. Explain the topic in short and why you found it most interesting?

Answer ::

MYSQL Database,table create, ER diagram , query etc