

Answer Script

Question No. 01

This is a schema of the Mini Student Library Management System.

Answer question 1 - 5 from this schema.

...

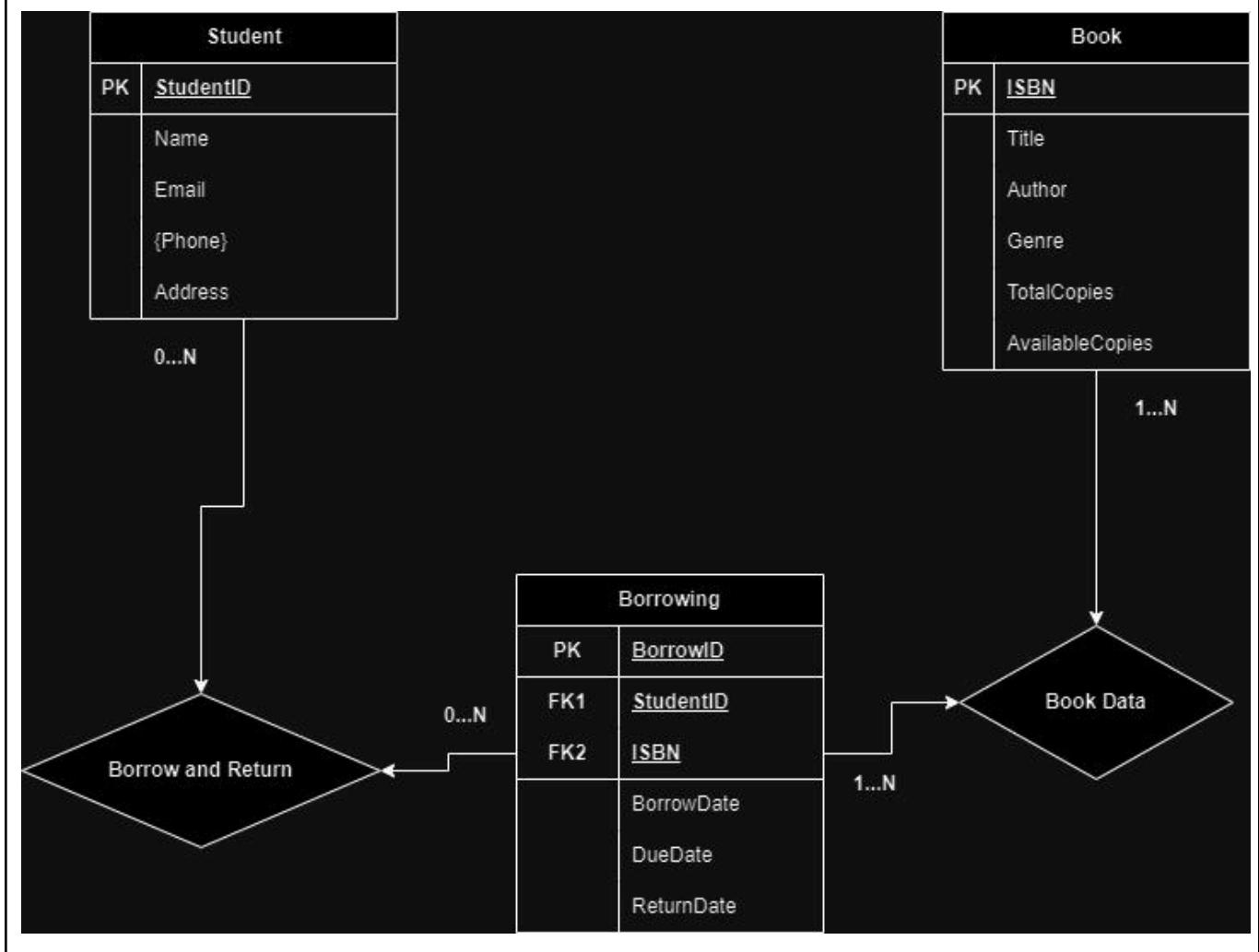
```
CREATE TABLE Student (
    StudentID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Email VARCHAR(255) NOT NULL,
    Phone VARCHAR(15),
    Address TEXT
);
```

```
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
);
```

```
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)
);
```

1) Make an ER Diagram of this Schema

Answer No. 01



Question No. 02

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

Answer No. 02

```
Insert INTO Borrowing (StudentID,ISBN,BorrowDate,DueDate)
values(3,
      (select ISBN from book where AvailableCopies=(select max(AvailableCopies) from book)
      Limit 1),
      current_date(),
      date_add(current_date(),interval 14 DAY)
     );
```

Question No. 03

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

Answer No. 03

```
SET SQL_SAFE_UPDATES = 0;
Update Book
Set AvailableCopies= AvailableCopies-1
where ISBN='9781234567890';
SET SQL_SAFE_UPDATES = 1;
```

Question No. 04

Retrieve the names of students who have borrowed the most books

Answer No. 04

```
Select st.Name as Name, count(b.BorrowId) as Number_of_Books
```

```
from student as st
```

```
join borrowing as b
```

```
on st.StudentID=b.StudentID
```

```
group by st.StudentID
```

```
having count(b.BorrowID)=(Select max(Numbers)
```

```
        from ( select count(BorrowID) as Numbers
```

```
            from borrowing
```

```
            group by StudentID) as t
```

```
);
```

Question No. 05

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

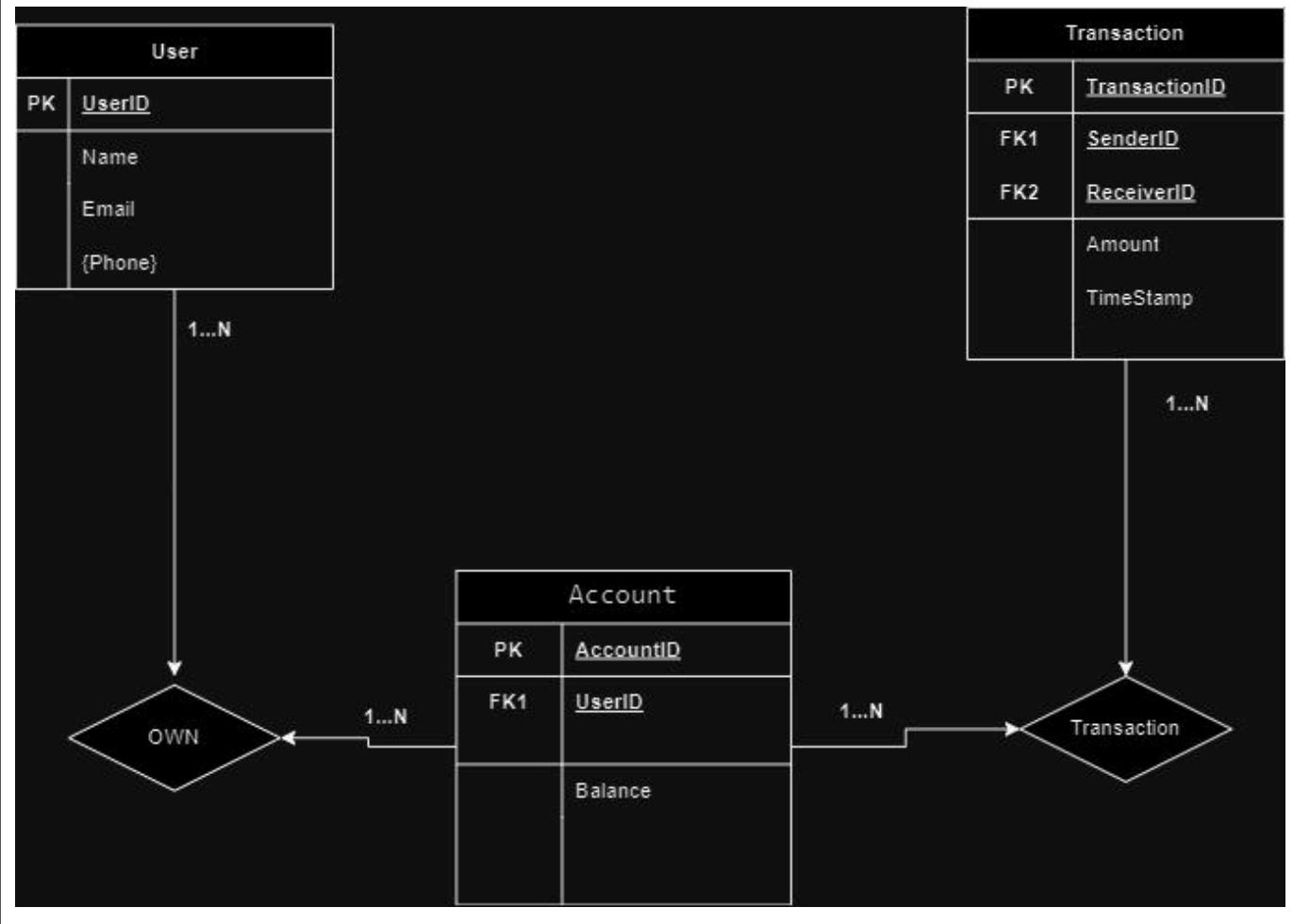
Answer No. 05

```
select *
From Book as ab
join borrowing as cd
on ab.ISBN=cd.ISBN
where (cd.DueDate < current_date() AND cd.ReturnDate IS NULL) OR (cd.ReturnDate >
cd.DueDate);
```

Question No. 06

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 No. 06



Question No. 07

Explain UNION and UNION ALL set operations in MySQL

Answer No. 07

Union : The union operator combines the result sets of two or more select queries into a single result set that includes all the rows that belong to all queries in union. It also removes duplicate rows from the result.

Example:

```
Select column_Name from table1  
Union Select column_name from table2;
```

Union All : The union all operator does the same as the union but it does not remove the duplicate row from the result.

Example:

```
Select column_Name from table1  
Union All Select column_name from table2;
```

Question No. 08

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

Answer No. 08

```
select min(salary)  
from employees  
where salary > (select min(salary) from employees);
```

Question No. 09

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 No. 09

```
create table Department
(
    Department_ID INT primary key,
    Department_Name varchar(20) Not NULL
);
create table Employee
(
    Employee_ID int primary key,
    Name varchar(50) NOT NULL,
    phone varchar(15) NOT null unique,
    Email varchar(50) unique,
    Department_ID INT,
    foreign key (Department_ID) references Department(Department_ID) on DELETE set NULL
);
create Table JobHistory
(
    History_ID int primary key,
    Employee_ID Int NOT NULL,
    Position varchar(20) NOT NULL,
    Join_Date date Not NULL,
    Leave_Date date,
    foreign key (Employee_ID) references Employee(Employee_ID) on Delete cascade
);
```

Question No. 10

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

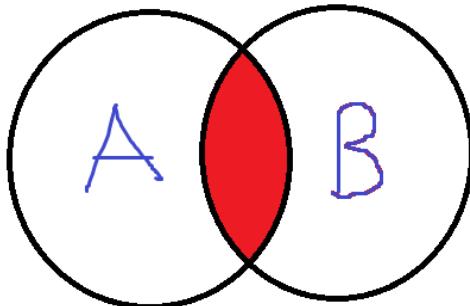
Answer No. 10

I found the Join query topic most interesting because it can combine rows from two or more tables based on a related column between them. It allows us to query data from multiple tables as if they were in a single table, making it very useful for organizing and analyzing data.

Join query are divided in several types like : Left,Right,Inner,Self joins Query.

Example of Inner Join Query:

Inner Joins : The inner join shows data that have been matched or both tables have.



Example:

```
Select emp.employee_id,emp.First_Name,dept.department_Name  
from employees as emp  
inner join departments as dept  
on emp.department_id=dept.department_id;
```