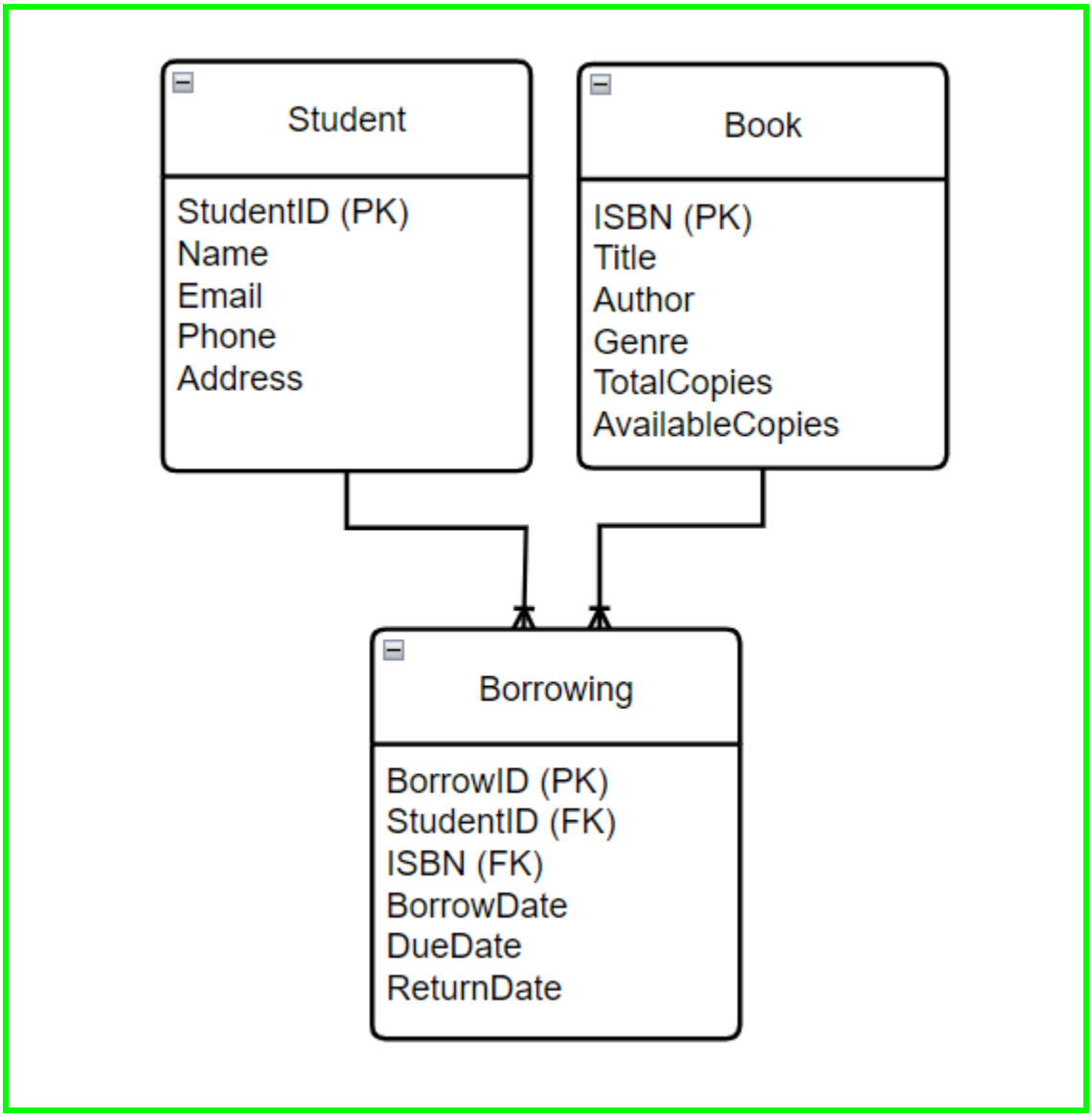
Final Exam

-- QS 1: Make an ER Diagram of this Schema.



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

```
insert into Borrowing (BorrowId, StudentID, ISBN, BorrowDate,
DueDate, ReturnDate) values (
        101, -- BorrowId
         3, -- StudentID
        (select ISBN from book order by AvailableCopies desc limit 1),
-- ISBN
         curdate(), -- BorrowDate
        date_add(curdate(), interval 1 day), -- DueDate
        date_add(curdate(), interval 10 day) -- ReturnDate
);
```

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

```
set SQL_SAFE_UPDATES = off;
update book set AvailableCopies = AvailableCopies - 1
where ISBN = '9781234567890';
```

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

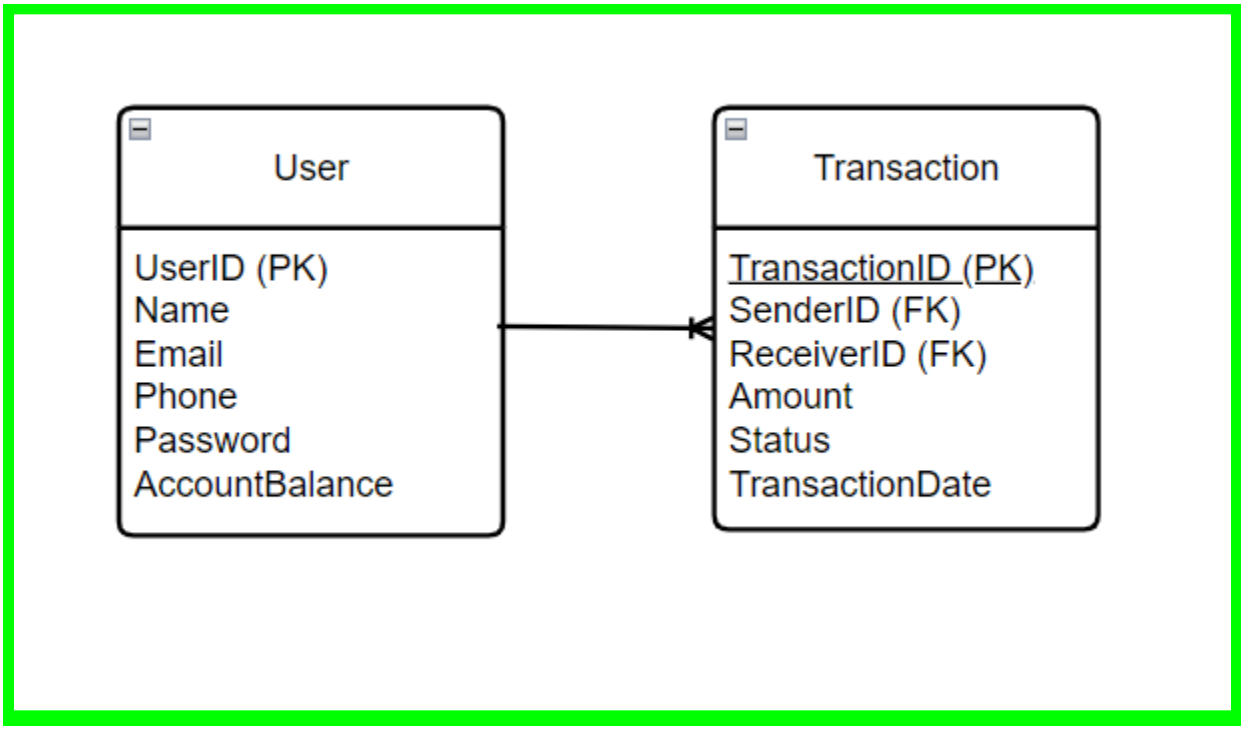
```
with MaxCount as (select count(*) from borrowing  group by
studentid  order by count(*) desc limit 1)

select Student.name from Student join Borrowing
on Student.StudentId = Borrowing.StudentId
group by Borrowing.StudentId having count(*) = (select * from
MaxCount);
```

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

```
select * from borrowing where ReturnDate is null and DueDate <
CURDATE();
```

-- QS 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).



-- QS 7: Explain UNION and UNION ALL set operations in MySQL.

UNION works like a set. The values returned by UNION are **unique and sorted**. Duplicate values are removed.
Example of **UNION**: 1, 2, 3, 4, 5, 10, 15...

UNION ALL includes duplicate values and Faster than UNION.
Example of **UNION ALL**: 1, 71, 10, 2, 30, 4, 50, 5, 15, 10, 10, 15, 15...

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

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

-- QS 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.)

```
create table Department (
   DepartmentID int primary key,
   DepartmentName VARCHAR(15)
);
CREATE TABLE Employee (
   EmployeeID int primary key,
   Name VARCHAR(100) not null,
   DepartmentID int,
   foreign key (DepartmentID) references
Department(DepartmentID) on delete cascade
);
create table JobHistory (
   JobHistoryID int primary key,
   EmployeeID int,
   JobTitle VARCHAR(20),
   StartDate date,
   EndDate date,
   foreign key (EmployeeID) references Employee(EmployeeID) on
delete set null
);
```

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

Most Interesting Topic: **Joins in SQL**

In SQL course, the topic I found most interesting was Joins. Joins allow us to combine data from multiple tables based on a related column, enabling us to perform complex queries and analyze relationships in the data.