**Case Study Assignment: Library Management System (MySQL Queries)**

**Scenario:**

**A Library Management System maintains data related to books, authors, members, and book transactions.**

**The database contains the following tables:**

**1. Authors (AuthorID, Name, Country)**

**2. Books (BookID, Title, AuthorID, Genre, PublishedYear, Price)**

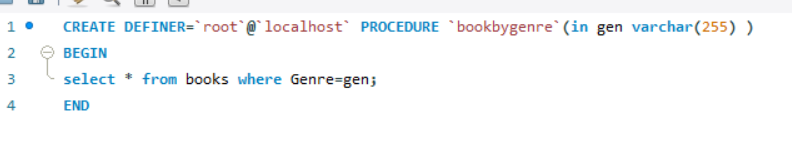
**3. Members (MemberID, Name, Email, JoinDate, MembershipType)**

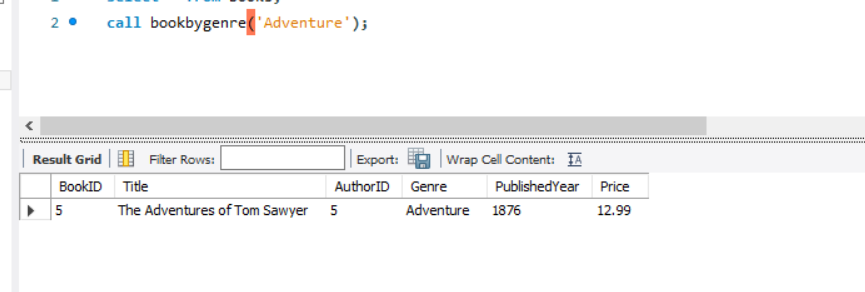
**4. Transactions (TransactionID, BookID, MemberID, IssueDate, ReturnDate, Status)**

**-- Assignment Questions:**

**1. Stored Procedure**

**1. Write a stored procedure to retrieve book details by genre.**

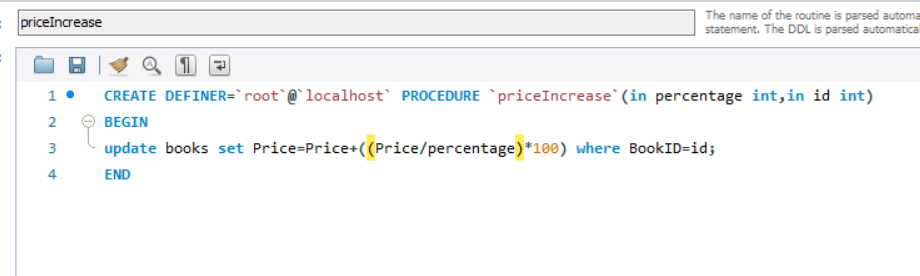
****

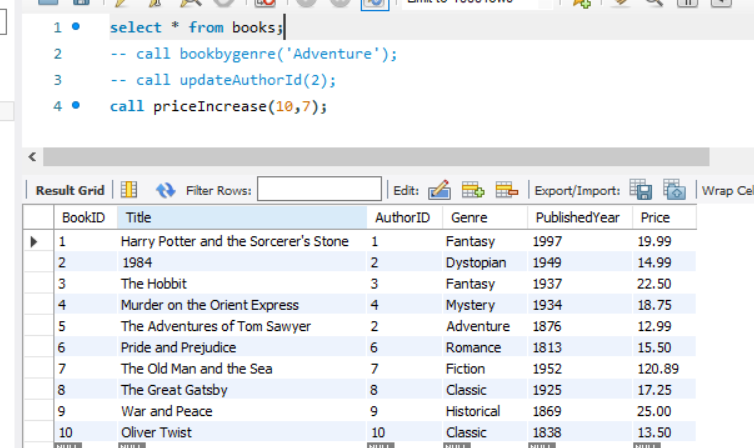
****

**2. Create a stored procedure that updates the price of books based on a given percentage increase.**

**Before:**

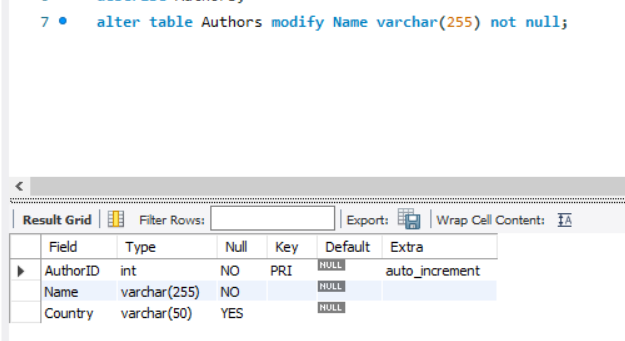
****

****

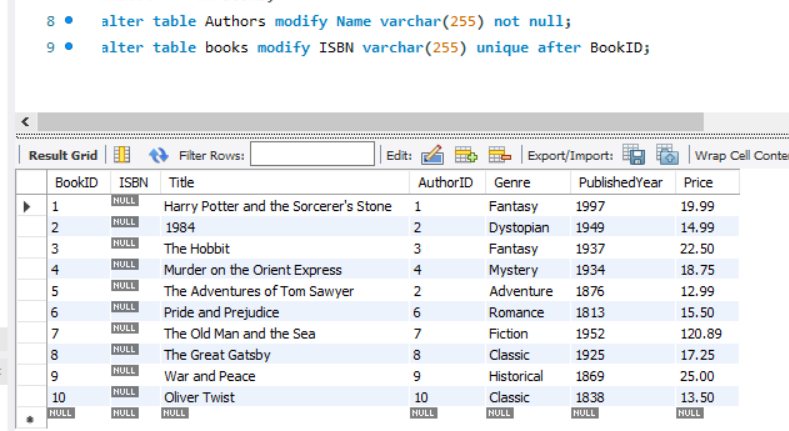
****

**2. DDL (Data Definition Language) Queries**

**3. Write a query to create the Authors table with appropriate constraints.**

****

**4. Modify the Books table to add a new column ISBN with a UNIQUE constraint.**

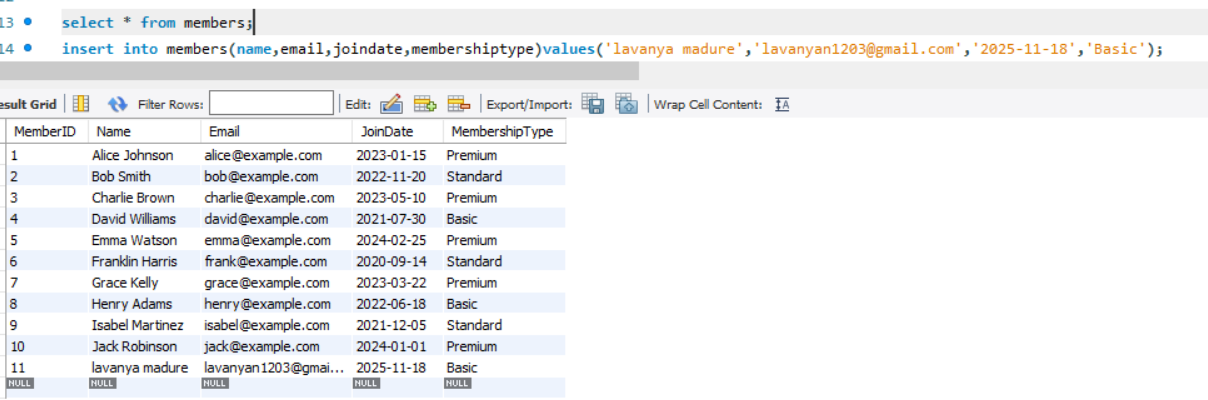
****

**5. Write a query to drop the Transactions table if it exists.**

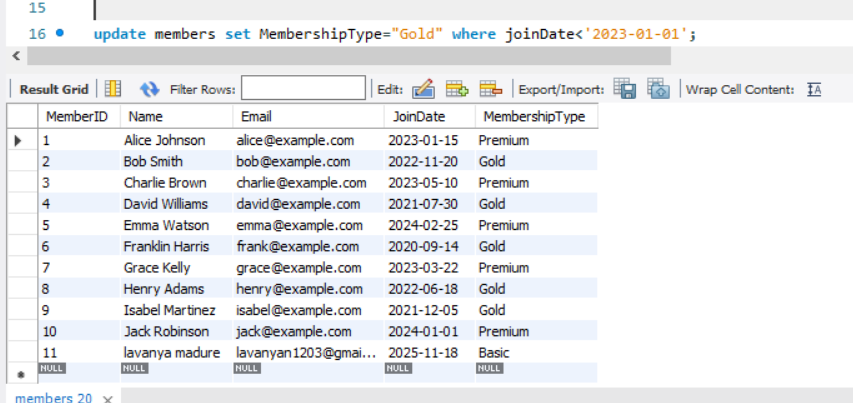
****

**3. DML (Data Manipulation Language) Queries**

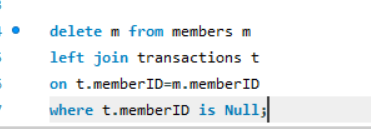
**6. Insert a new record into the Members table.**

****

**7. Update the MembershipType of members who joined before 2023-01-01 to "Gold".**

****

**8. Delete members who have never borrowed a book.**

****

**4. DQL (Data Query Language) Queries**

**9. Retrieve all books written by a specific author using WHERE.**

**select \* from Authors where name='Agatha Christie';**

**10. Find books that belong to either ‘Fiction’ or ‘Mystery’ genres using OR.**

**select \* from books where Genre in ('Fantasy','Mystery');**

**11. Get a list of members who joined after ‘2023-01-01’ but are NOT Premium members.**

**select \* from members where joindate<'2023-01-01' and membershiptype !='Premium';**

**5. Filtering with Operators**

**12. Retrieve books that have a price greater than 500 but less than 1000 using BETWEEN.**

**select \* from members where joindate<'2023-01-01' and membershiptype !='Premium';**

**13. Get a list of books where the title contains ‘History’ using LIKE.**

**select \* from books where title like '%Great%';**

**6. Sorting and Grouping**

**14. Display books sorted by PublishedYear in descending order.**

**select \* from books order by publishedyear desc;**

**15. Count the number of books in each genre using GROUP BY.**

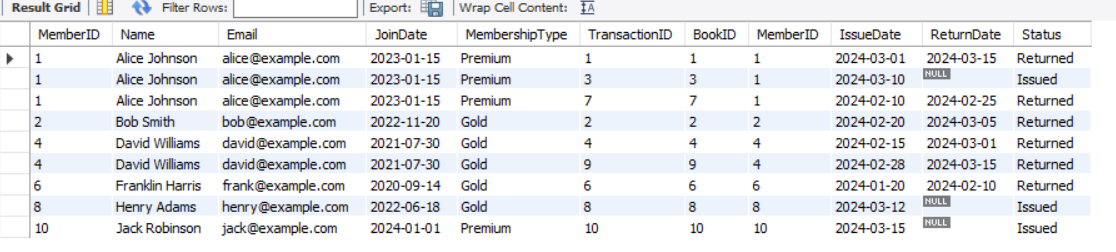
**select genre,count(\*) from books**

**group by genre;**

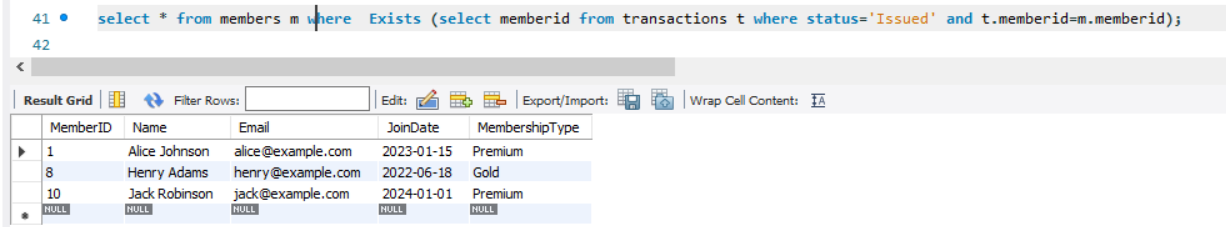
**7. Advanced Queries**

**16. Retrieve members who have borrowed at least one book using EXISTS.**

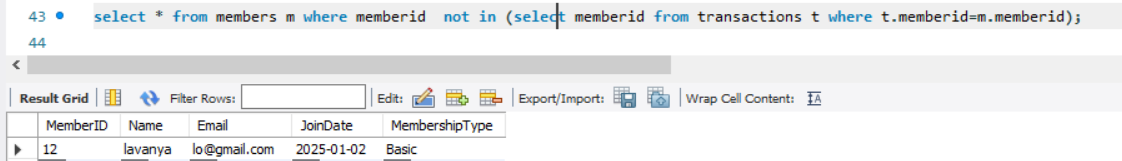
**Before:**

****

**After:**

****

**17. Get books that have never been borrowed using NOT IN with a subquery.**

****

**18. Retrieve all book transactions, displaying member names along with book titles using JOINs.**

**select m.name as Member\_Name , b.title as Book\_Name,b.price from members m**

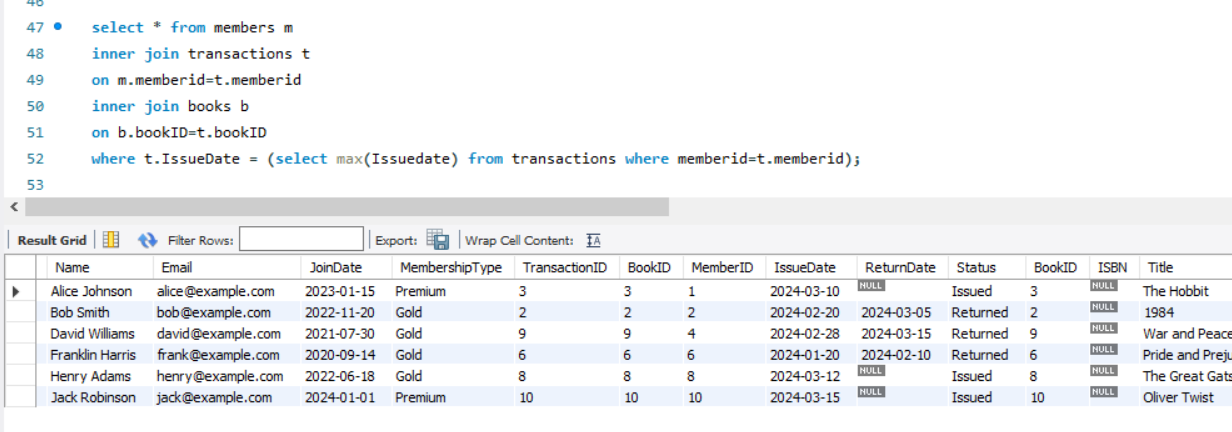
**inner join transactions t**

**on m.memberid=t.memberid**

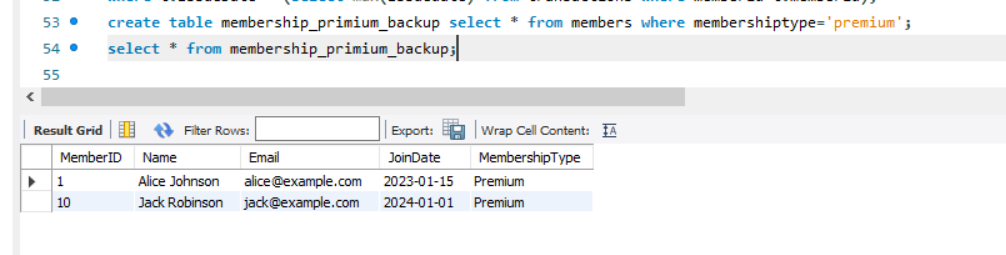
**inner join books b**

**on b.bookID=t.bookID;**

**19. Get the most recent book issued by each member using SUBQUERY.**

****

**20. Copy all members with a Premium membership to a backup table .**

****