Lab Assignment Report

Only for course Teacher						
		Needs Improvement	Developing	Sufficient	Above Average	Total Mark
Allocate mark & Percentage		25%	50%	75%	100%	10
Clarity	2					
Content Quality	4					
Spelling & Grammar	2					
Organization and Formatting	2					
					Total obtained mark	
Comments						

Semester: Fall 2024

Student Name: Afzal Hossain Raju

Student ID: 0242310005341031

Batch: 40 **Section: D**

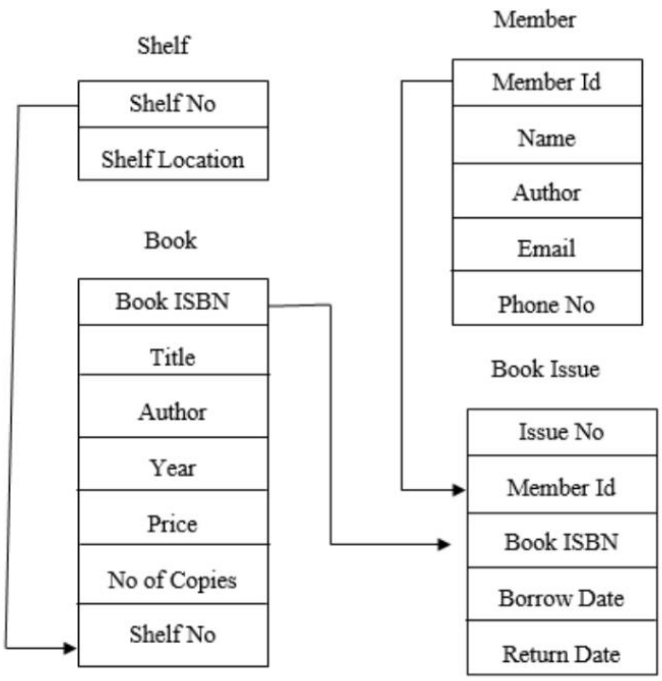
Course Code: SE 224 **Course Name: Database Management System Lab**

Course Teacher Name: Tapushe Rabaya Toma

Designation: Assistant Professor

Submission Date: 08/12/2024

Table diagram:



There are four tables including the given database,

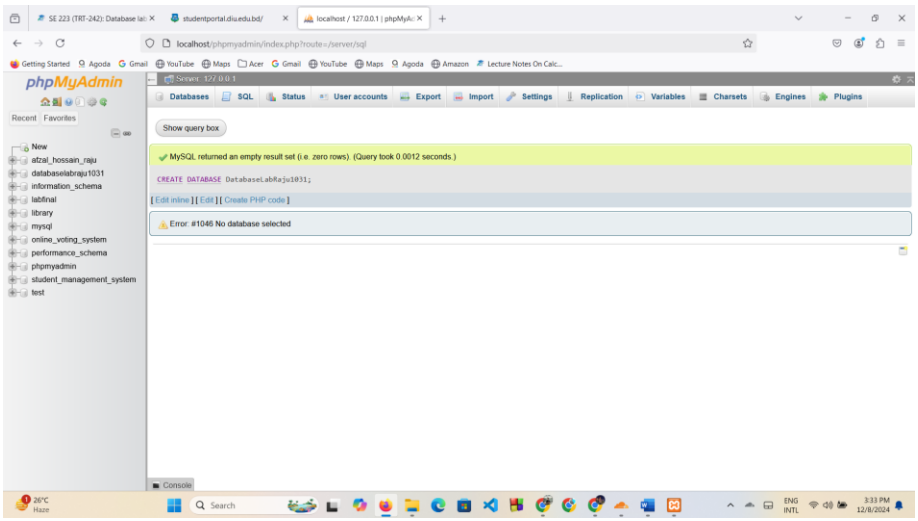
1. Shelf (Shelf No, Shelf Location)

2. Book (Book ISBN, Title, Author, Year, Price, No of Copies, Shelf No)

3. Member (Member Id, Name, Author, Email, Phone No)

4. Book Issue (Issue No, Member Id, Book ISBN, Borrow Date, Return Date)

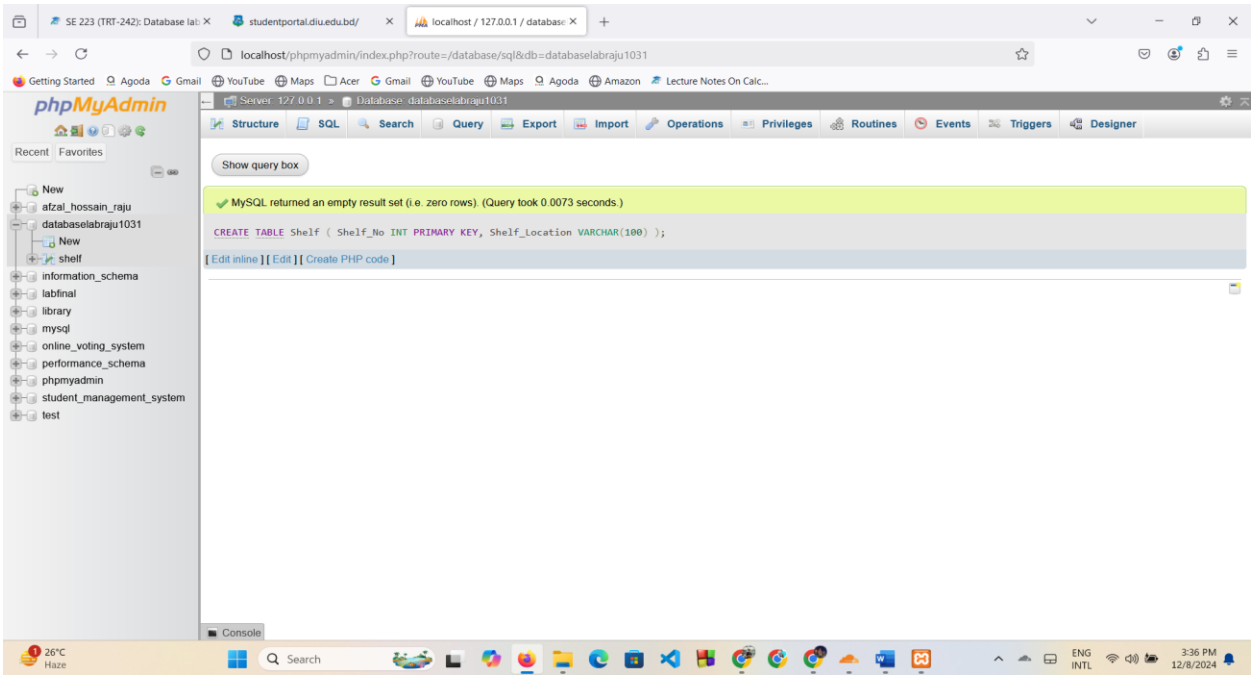
Create database as databaselabraju1031 In SQL:



1. Create all tables with proper "Primary key" and "Foreign Keys":

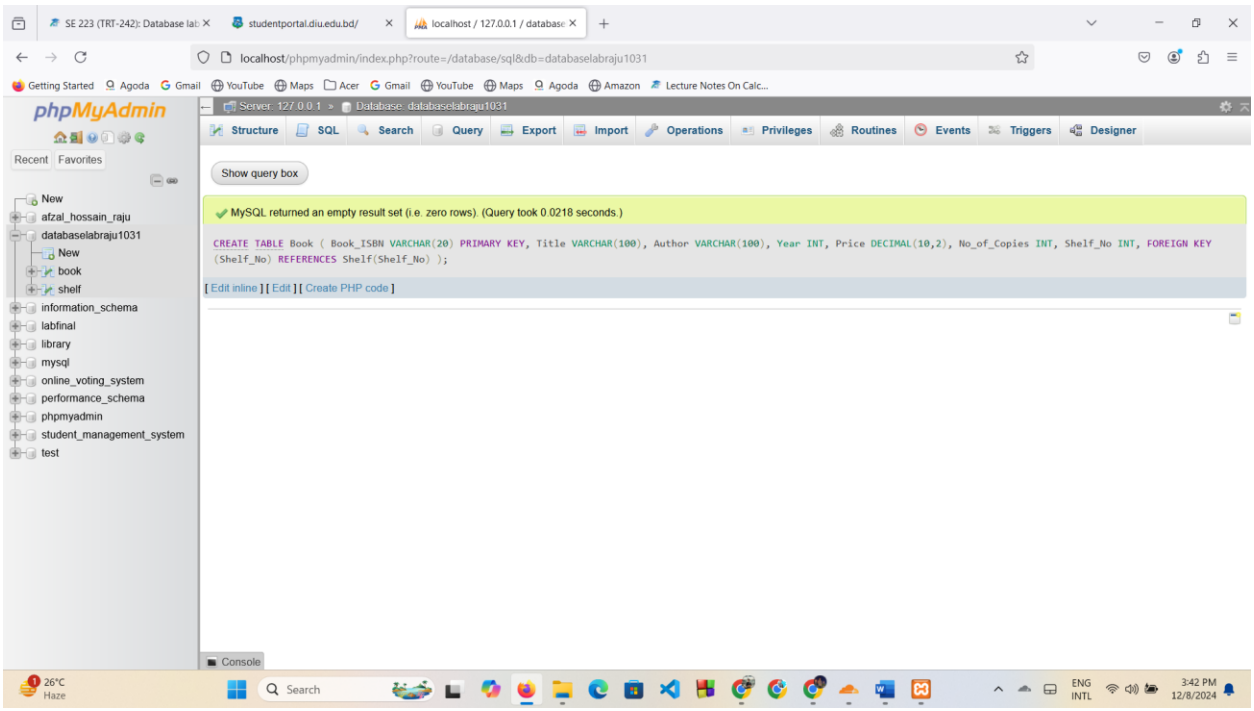
SQL:

```
CREATE TABLE Shelf (
Shelf_No INT PRIMARY KEY,
Shelf_Location VARCHAR(100)
 );
```



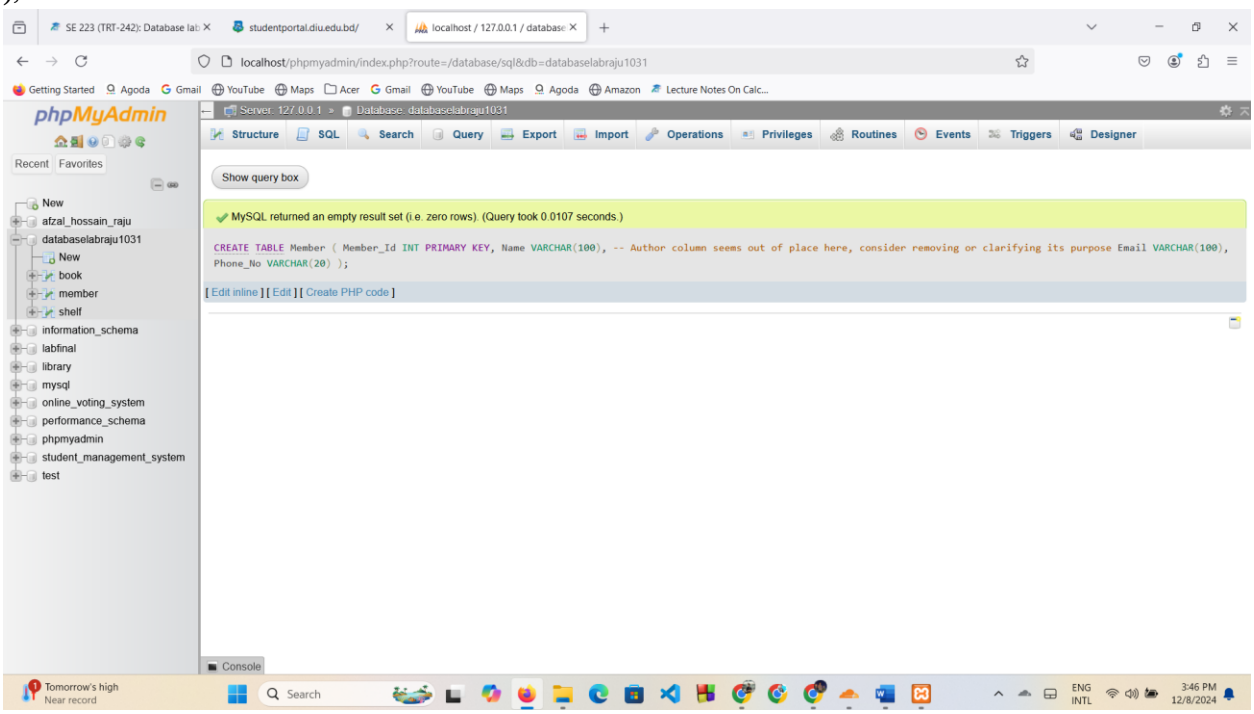
SQL:

```
CREATE TABLE Book (
 Book_ISBN VARCHAR(20) PRIMARY KEY,
 Title VARCHAR(100),
 Author VARCHAR(100),
 Year INT,
 Price DECIMAL(10,2),
 No_of_Copies INT,
 Shelf_No INT,
 FOREIGN KEY (Shelf_No) REFERENCES Shelf(Shelf_No));
```



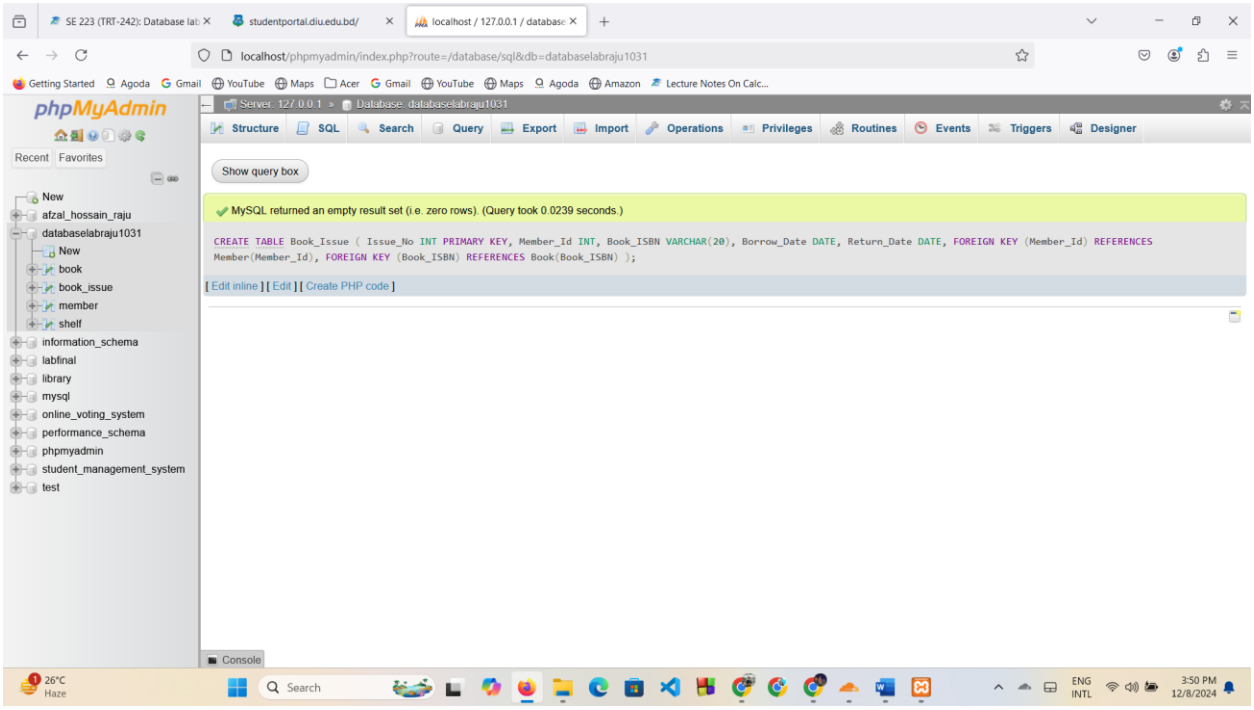
SQL:

```
CREATE TABLE Member (
Member_Id INT PRIMARY KEY,
Name VARCHAR(100),
Email VARCHAR(100),
Phone_No VARCHAR(20)
);
```



SQL:

```
CREATE TABLE Book_Issue (
Issue_No INT PRIMARY KEY,
Member_Id INT,
Book_ISBN VARCHAR(20),
Borrow_Date DATE,
Return_Date DATE,
FOREIGN KEY (Member_Id) REFERENCES Member(Member_Id),
FOREIGN KEY (Book_ISBN) REFERENCES Book(Book_ISBN)
);
```

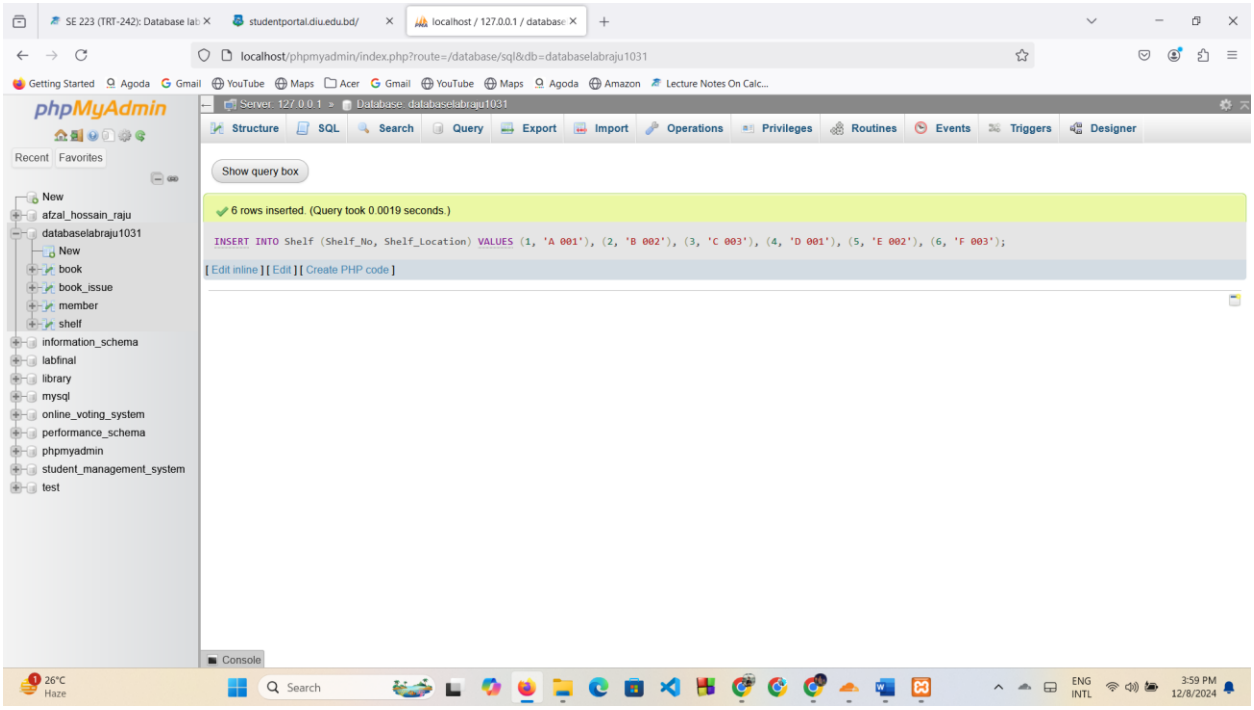


2. Insert a minimum of 6 Tuples defining SQL for each table:

Insert into table shelf:

SQL:

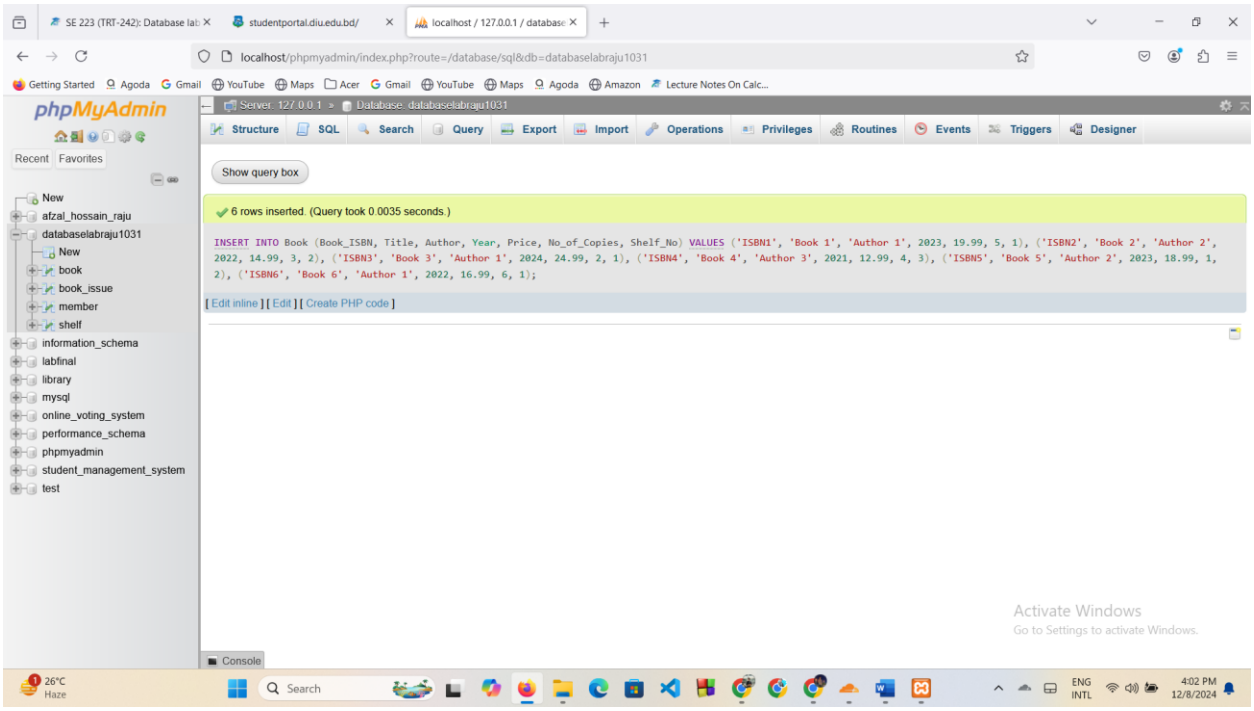
INSERT INTO Shelf (Shelf_No, Shelf_Location) VALUES
(1, 'A 001'),
(2, 'B 002'),
(3, 'C 003')
(4, 'D 001'),
(5, 'E 002'),
(6, 'F 003');



Insert into Book table:

SQL:

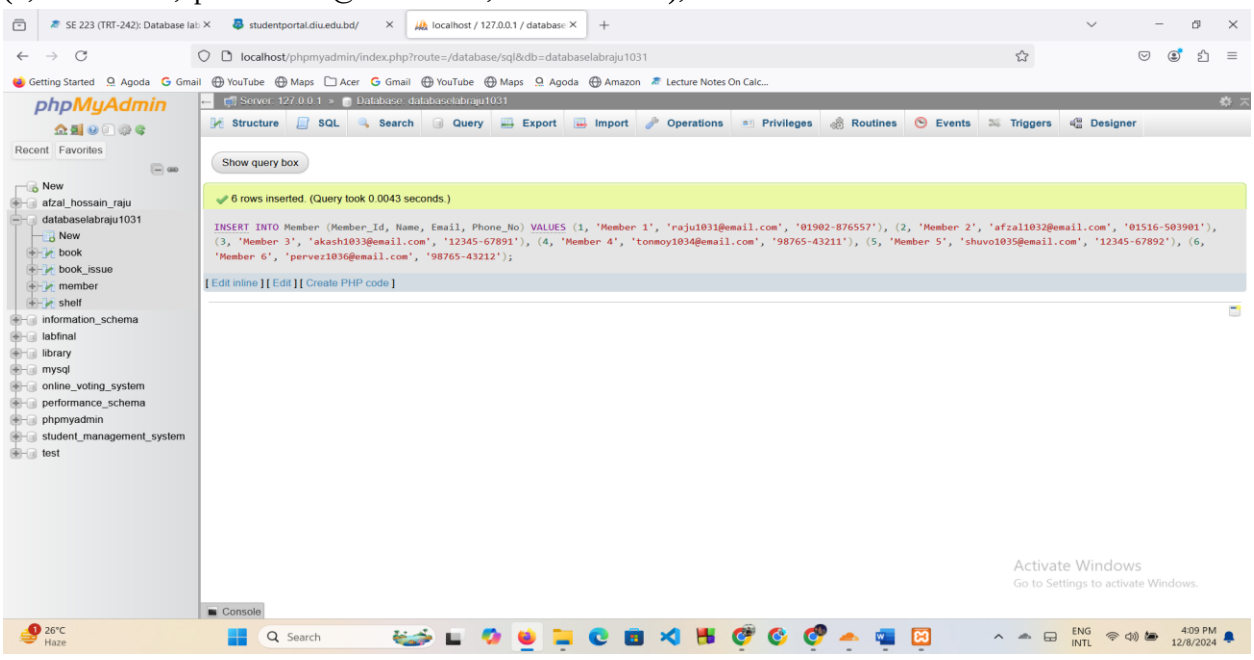
INSERT INTO Book (Book_ISBN, Title, Author, Year, Price, No_of_Copies, Shelf_No) VALUES
('ISBN1', 'Book 1', 'Author 1', 2023, 19.99, 5, 1),
('ISBN2', 'Book 2', 'Author 2', 2022, 14.99, 3, 2),
('ISBN3', 'Book 3', 'Author 1', 2024, 24.99, 2, 1),
('ISBN4', 'Book 4', 'Author 3', 2021, 12.99, 4, 3),
('ISBN5', 'Book 5', 'Author 2', 2023, 18.99, 1, 2),
('ISBN6', 'Book 6', 'Author 1', 2022, 16.99, 6, 1);



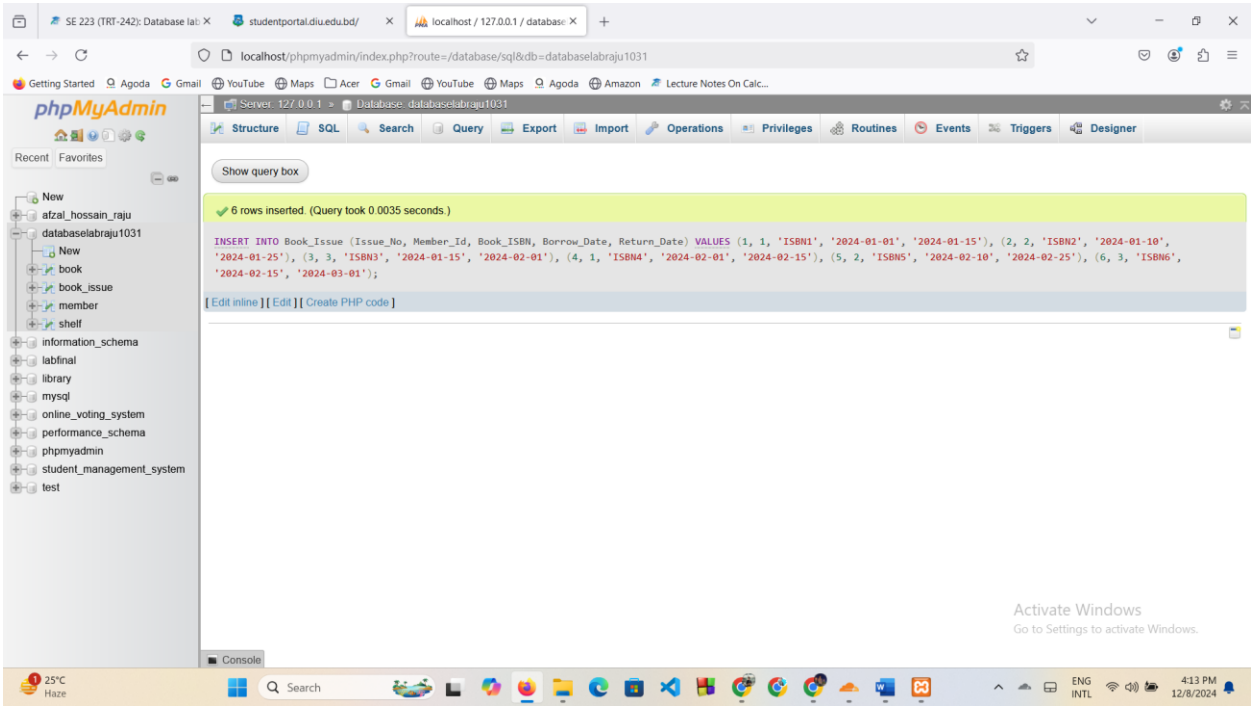
Insert into Member table:

SQL:

INSERT INTO Member (Member_Id, Name, Email, Phone_No) VALUES
(1, 'Member 1', 'raju1031@email.com', '01902-876557'),
(2, 'Member 2', 'afzal1032@email.com', '01516-503901'),
(3, 'Member 3', 'akash1033@email.com', '12345-67891'),
(4, 'Member 4', 'tonmoy1034@email.com', '98765-43211'),
(5, 'Member 5', 'shuvo1035@email.com', '12345-67892'),
(6, 'Member 6', 'pervez1036@email.com', '98765-43212');



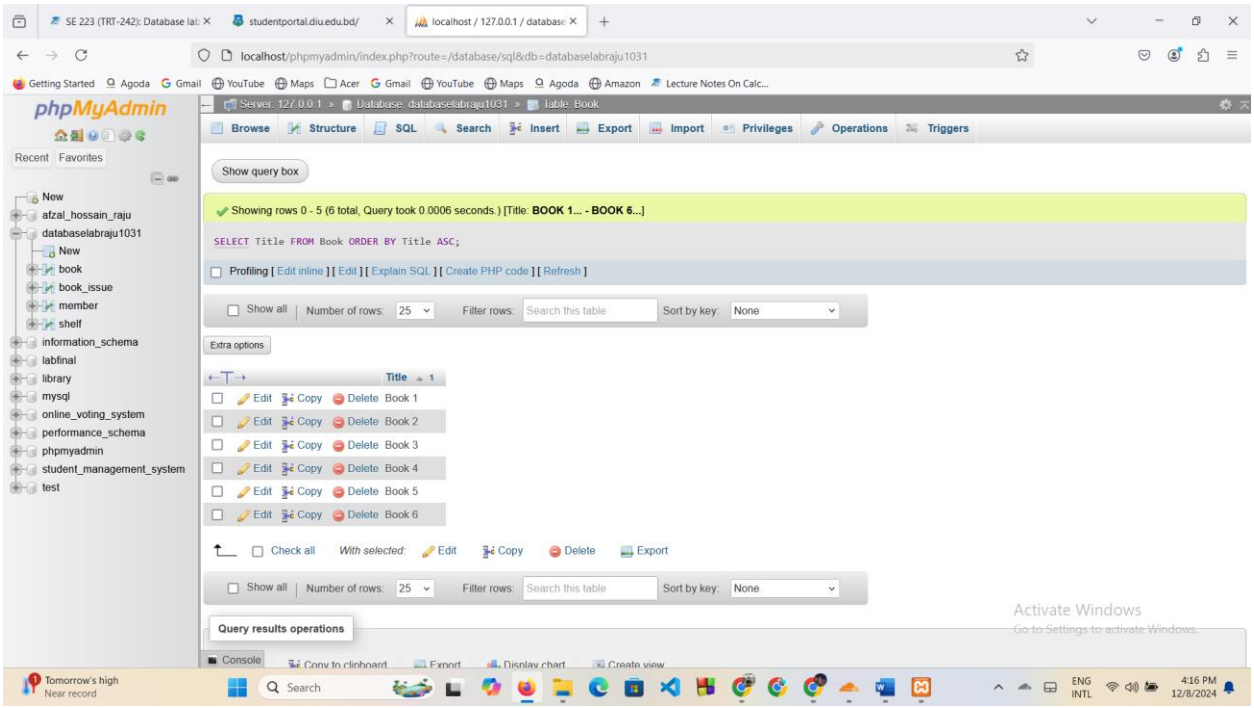
Insert into Book_Issue table:
SQL:
INSERT INTO Book_Issue (Issue_No, Member_Id, Book_ISBN, Borrow_Date, Return_Date) VALUES
(1, 1, 'ISBN1', '2024-01-01', '2024-01-15'),
(2, 2, 'ISBN2', '2024-01-10', '2024-01-25'),
(3, 3, 'ISBN3', '2024-01-15', '2024-02-01'),
(4, 1, 'ISBN4', '2024-02-01', '2024-02-15'),
(5, 2, 'ISBN5', '2024-02-10', '2024-02-25'),
(6, 3, 'ISBN6', '2024-02-15', '2024-03-01');



3. **Select all the books with title in ascending order:**

SQL:

SELECT Title FROM Book ORDER BY Title ASC;



4. Select the book's Title Borrowed by the Member whose Id is 1:

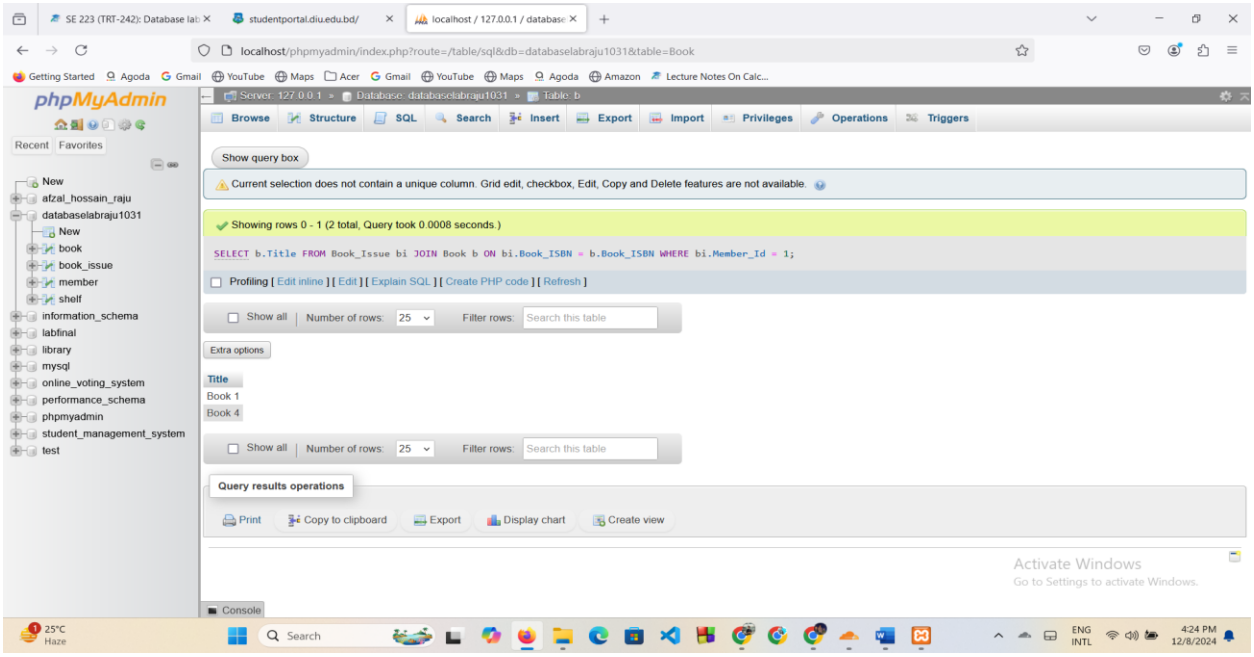
SQL:

SELECT b.Title

FROM Book_Issue bi

JOIN Book b ON bi.Book_ISBN = b.Book_ISBN

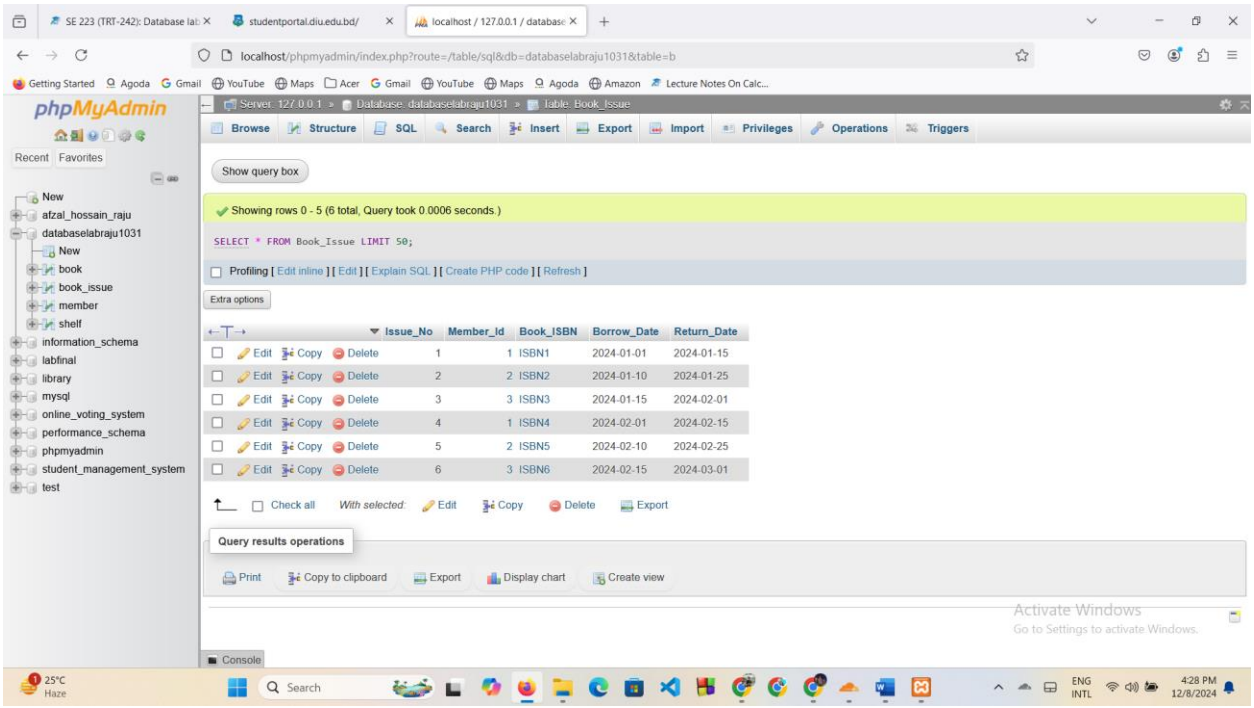
WHERE bi.Member_Id = 1;



5. Select the top 50 Book Issue:

SQL:

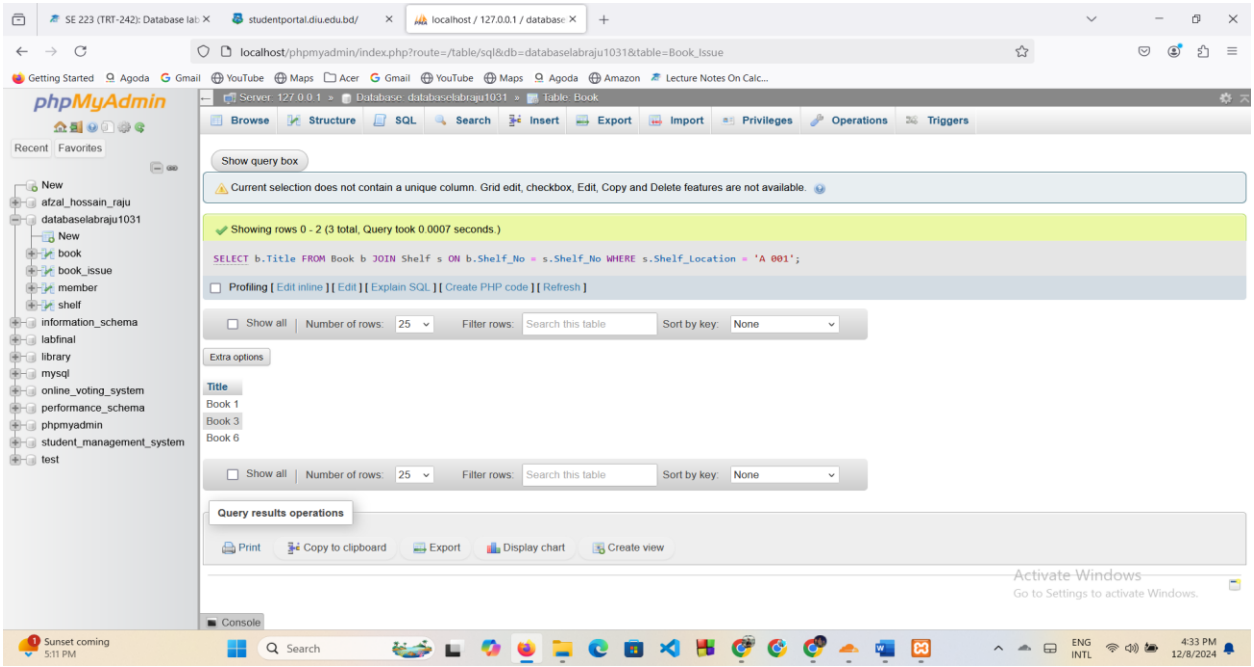
SELECT * FROM Book_Issue LIMIT 50;



6. Select the Books of Shelf Location 'A 001':

SQL:

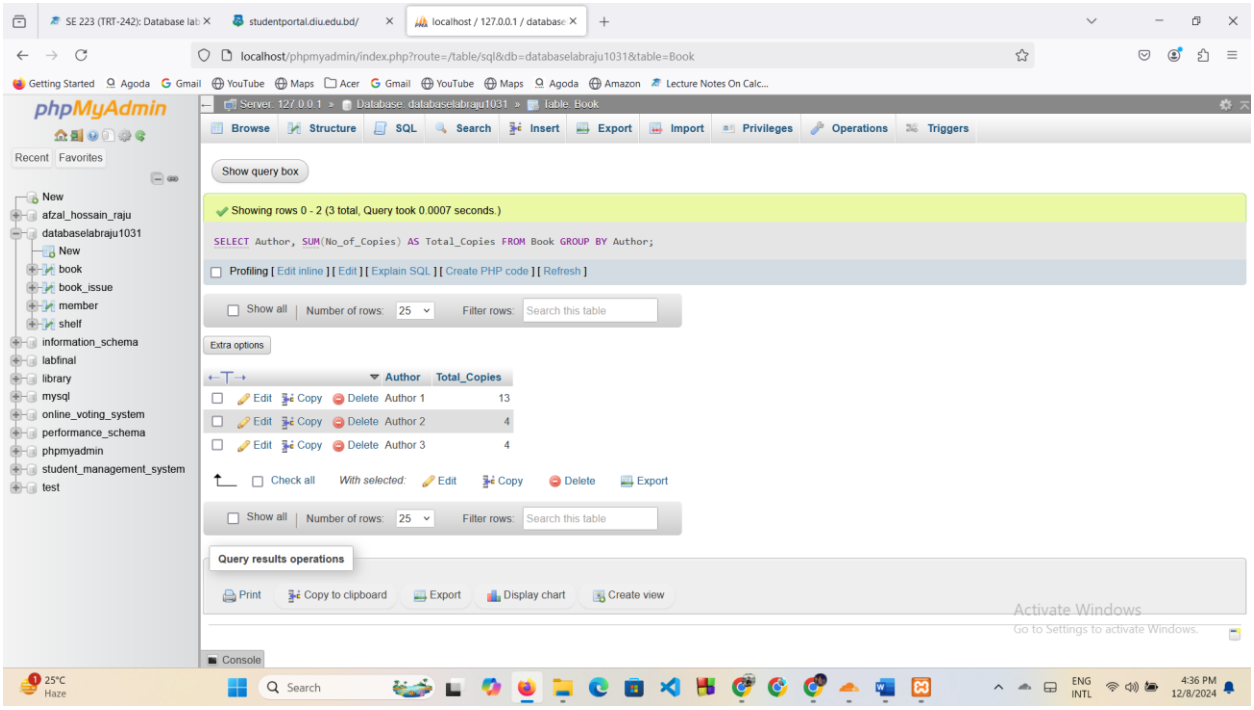
SELECT b.Title
FROM Book b
JOIN Shelf s ON b.Shelf_No = s.Shelf_No
WHERE s.Shelf_Location = 'A 001';



7. Select the total NoOfCopies of Books by each Author:

SQL:

```
SELECT Author, SUM(No_of_Copies) AS Total_Copies
FROM Book GROUP BY Author;
```

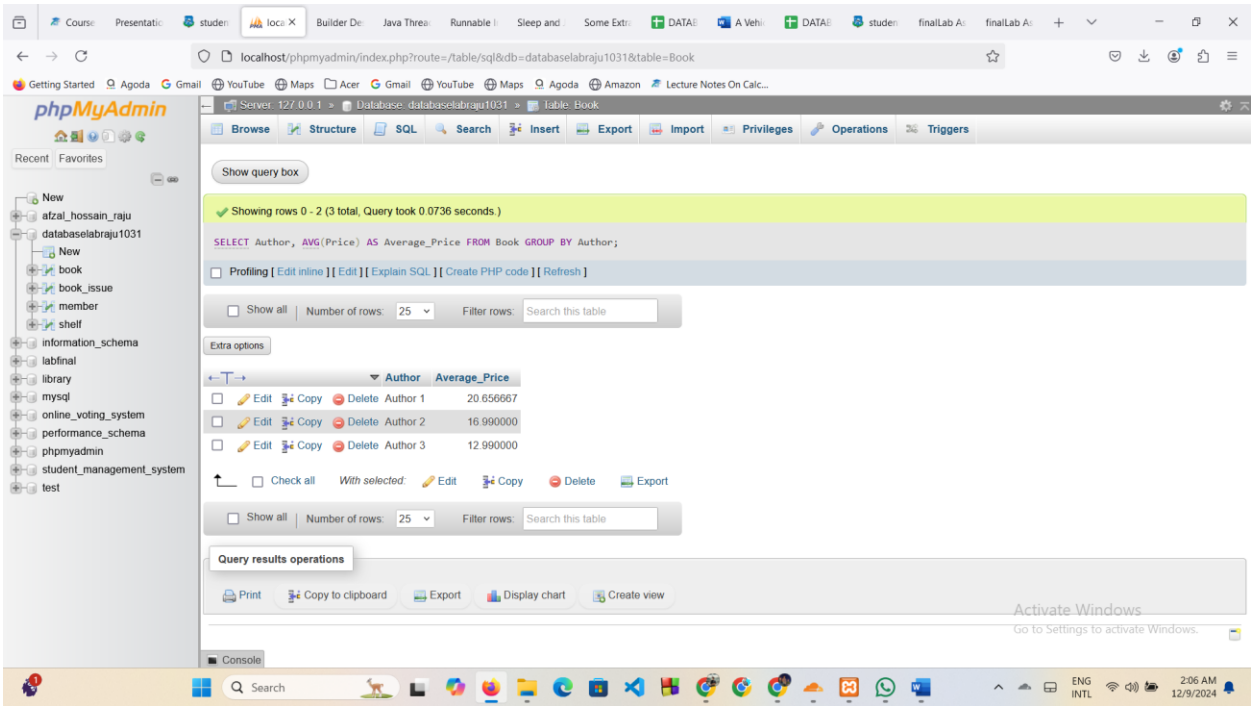


8. Select the average price of Books of each author

SQL:SELECT Author, AVG(Price) AS Average_Price

FROM Book

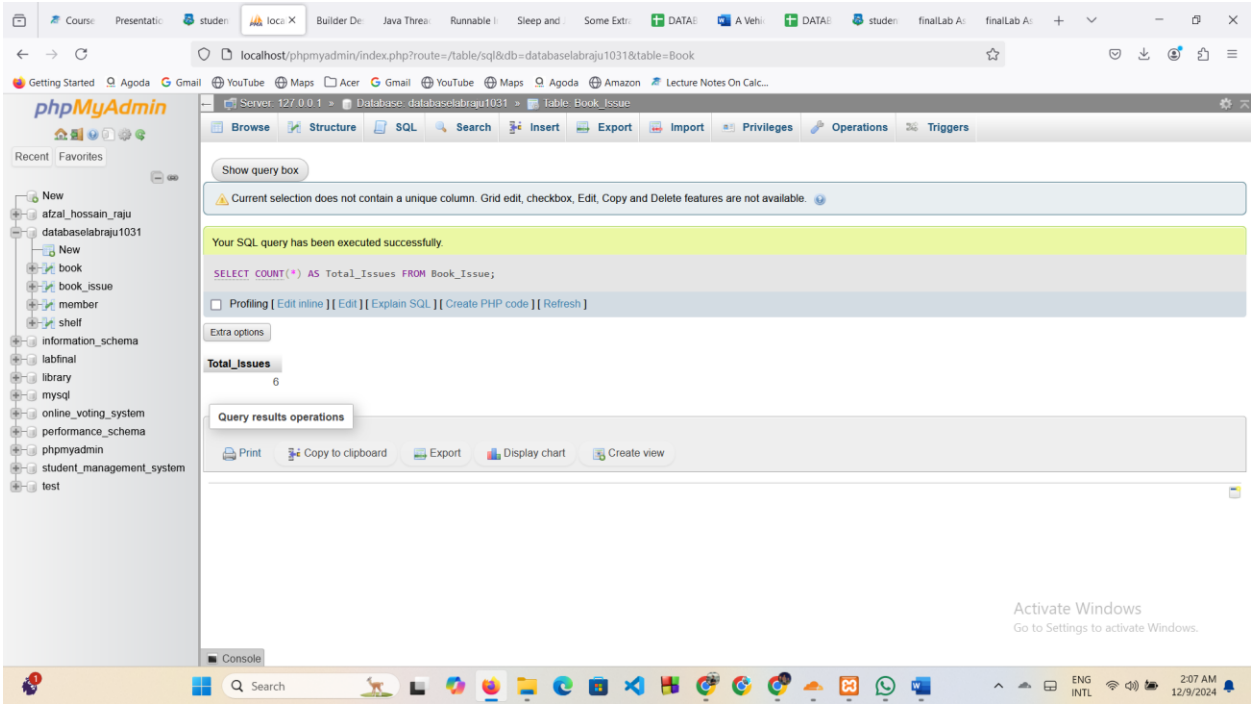
GROUP BY Author;



9. Find how many issues are there in the database

SQL:SELECT COUNT(*) AS Total_Issues

FROM Book_Issue;

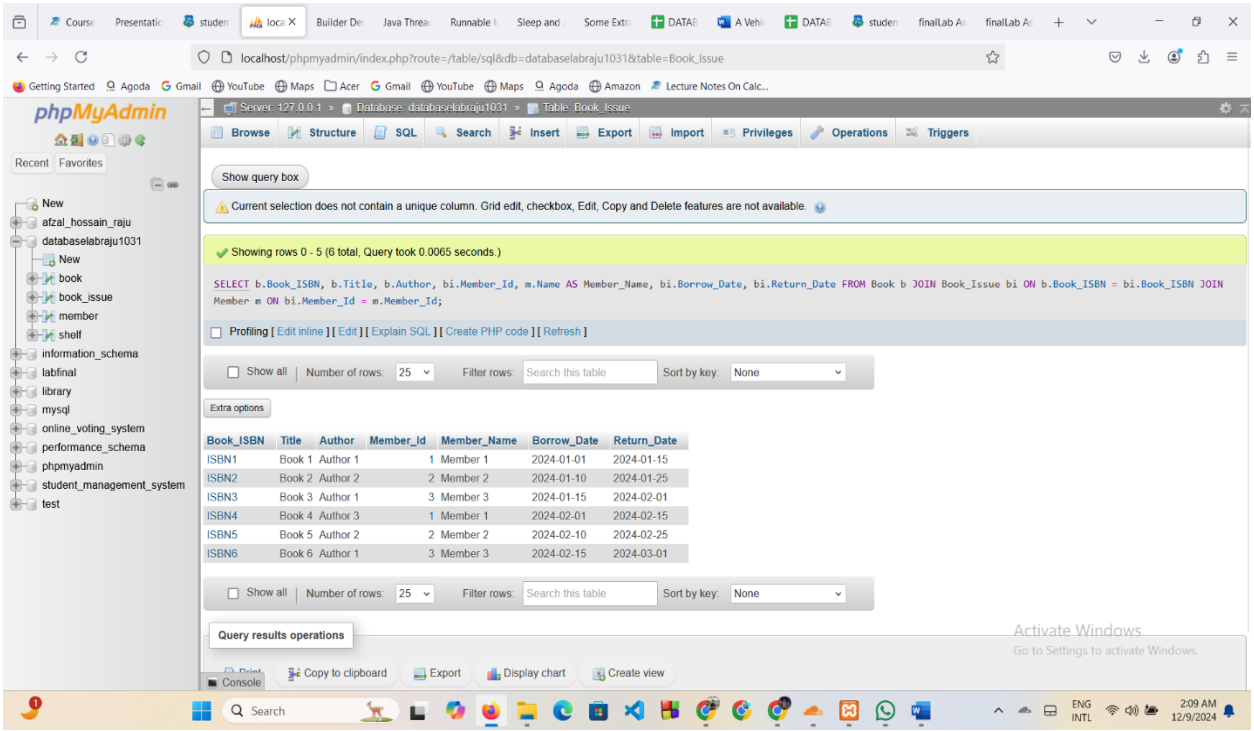


10. Find Book ISBN, Title, Author, Member Id, Member Name, Borrow Date and Return Date

SQL:SELECT b.Book_ISBN, b.Title, b.Author, bi.Member_Id, m.Name AS Member_Name, bi.Borrow_Date, bi.Return_Date FROM Book b

JOIN Book_Issue bi ON b.Book_ISBN = bi.Book_ISBN

JOIN Member m ON bi.Member_Id = m.Member_Id;



11. Select Book ISBN, Title, Borrow Date and Return Date of a member (Parameterized with member Id)

SQL:

```
SELECT
b.Book_ISBN,
b.Title,
bi.Borrow_Date,
bi.Return_Date
FROM Book b
JOIN Book_Issue bi ON b.Book_ISBN = bi.Book_ISBN
WHERE bi.Member_Id = 3;
```