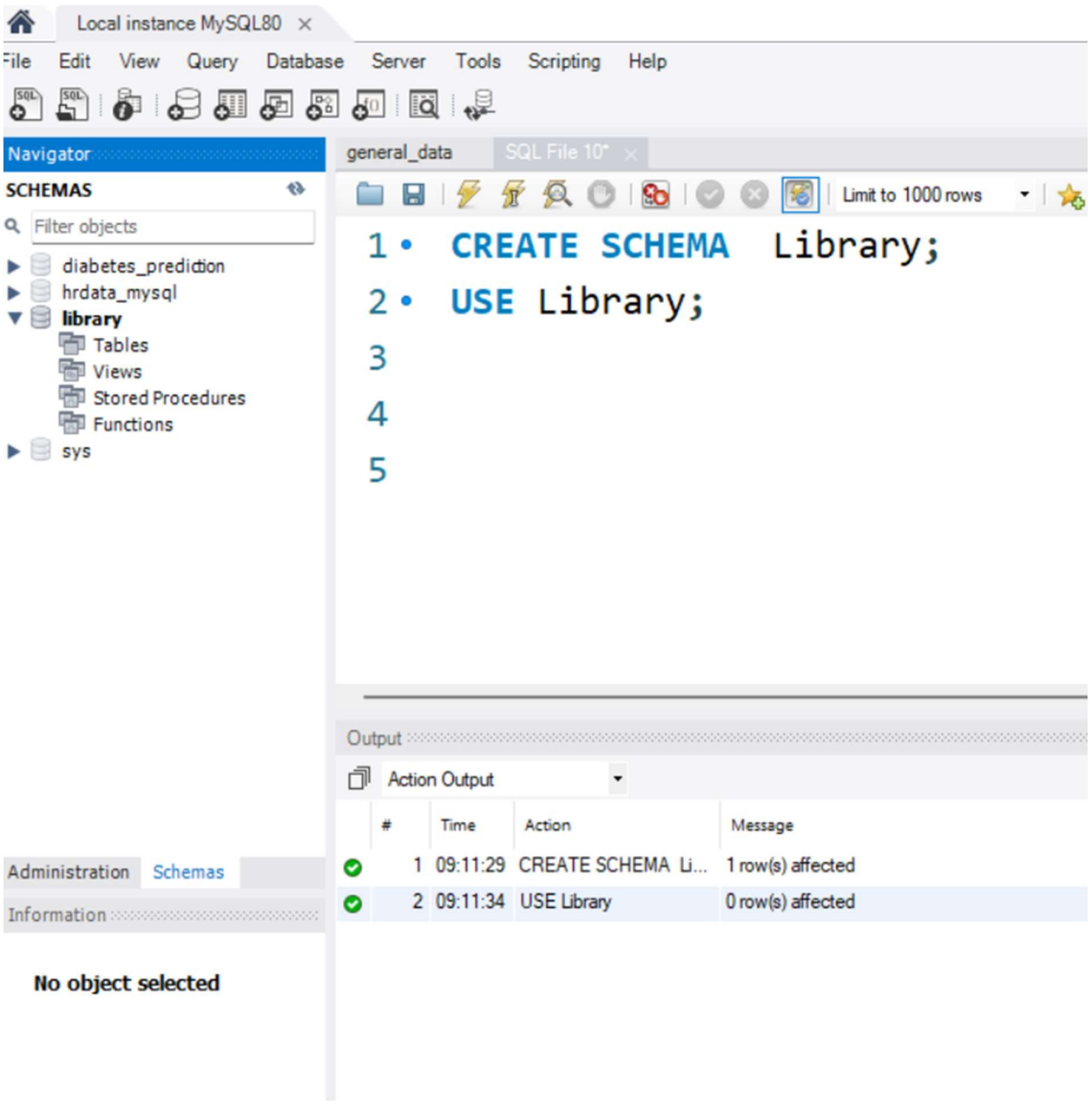
Assignment – 1

Ch.Lakshmipriya-G8-Ds

1.Create Schema Library

Created schema using command -**create schema library;**



2. Create tables

Books: BookID - Pk,BookName,AuthorName,Genre,pages

Customers : CustomerId - PK, CustomerName, address

BooksBorrowed: SlNo,BookID - FK,CustomerID - FK,DaysBookRetained

Cost: MaxPages, Cost/day

A)Creating Books table:

Code: CREATE TABLE Books (

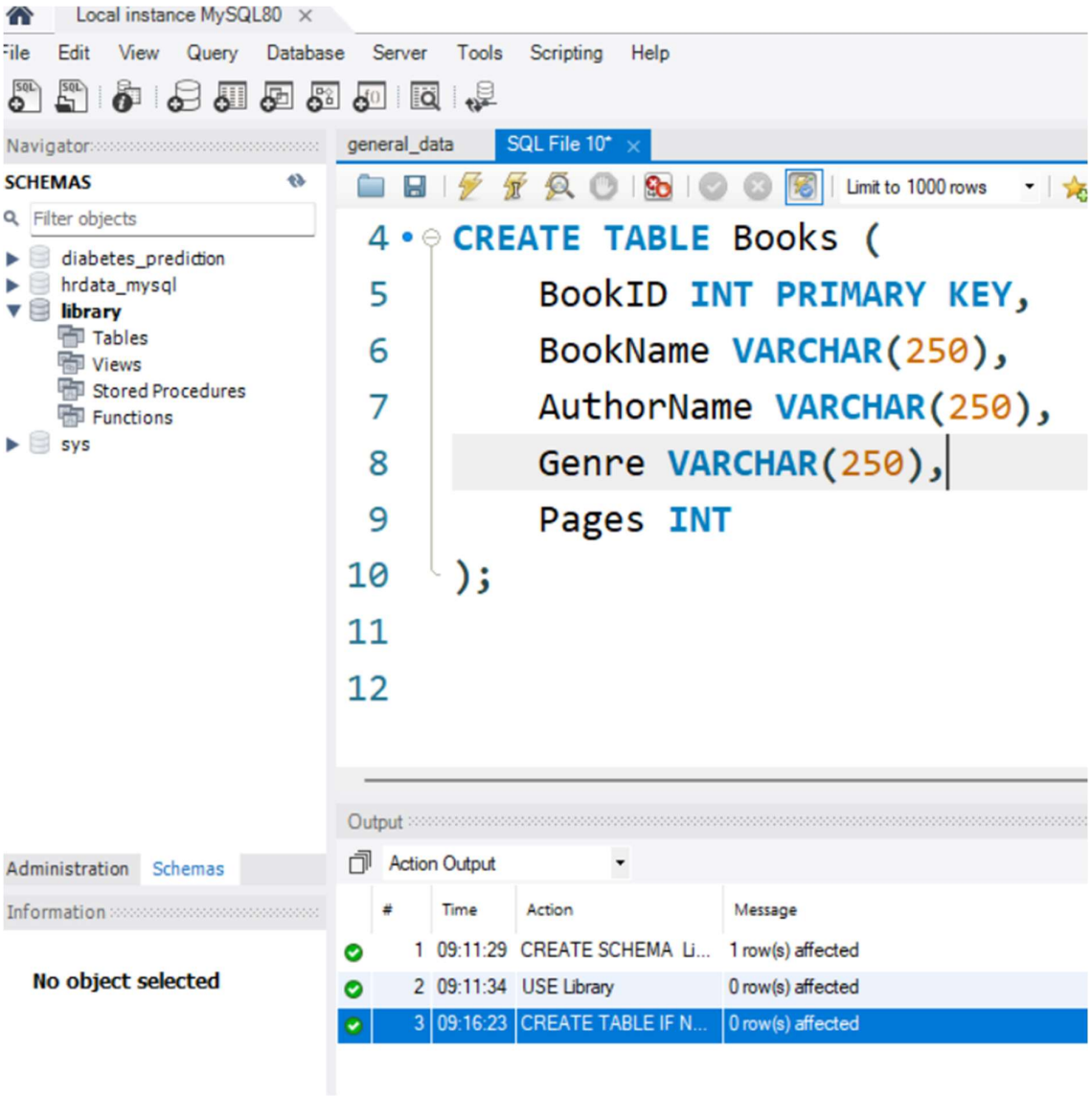
BookID INT PRIMARY KEY,

BookName VARCHAR(250),

AuthorName VARCHAR(250),

Genre VARCHAR(250),

Pages INT);



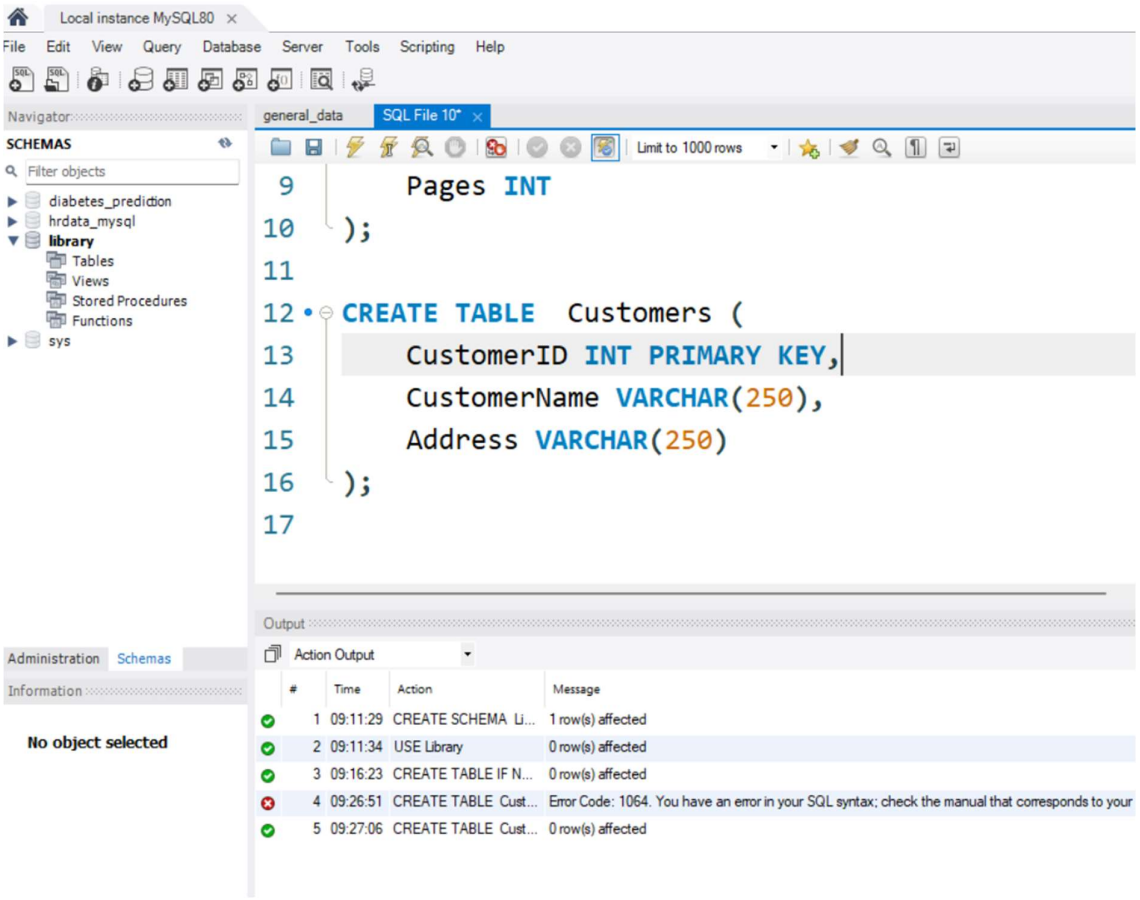
B)Creating Customers table:

Code: CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

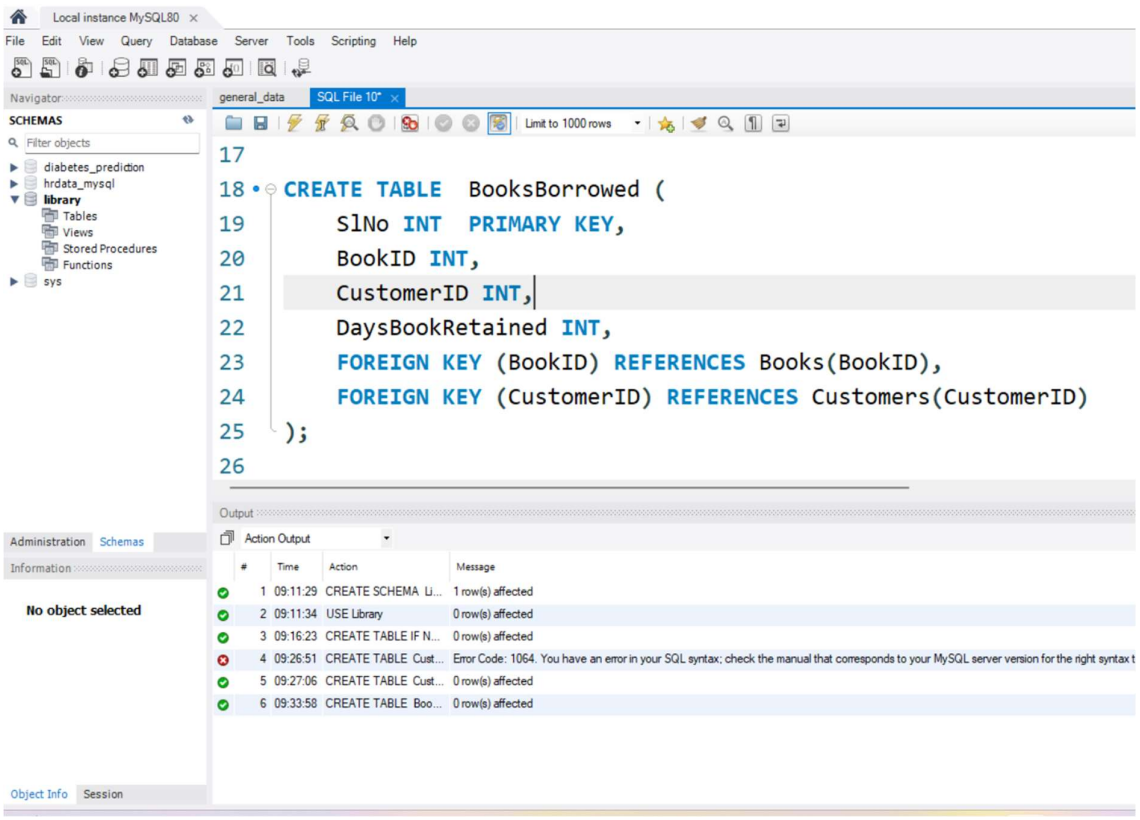
CustomerName VARCHAR(250),

Address VARCHAR(250));



C) Creating Books Borrowed Table:

Code: CREATE TABLE BooksBorrowed (SlNo INT PRIMARY KEY, BookID INT, CustomerID INT, DaysBookRetained INT,FOREIGN KEY (BookID) REFERENCES Books(BookID), FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID));



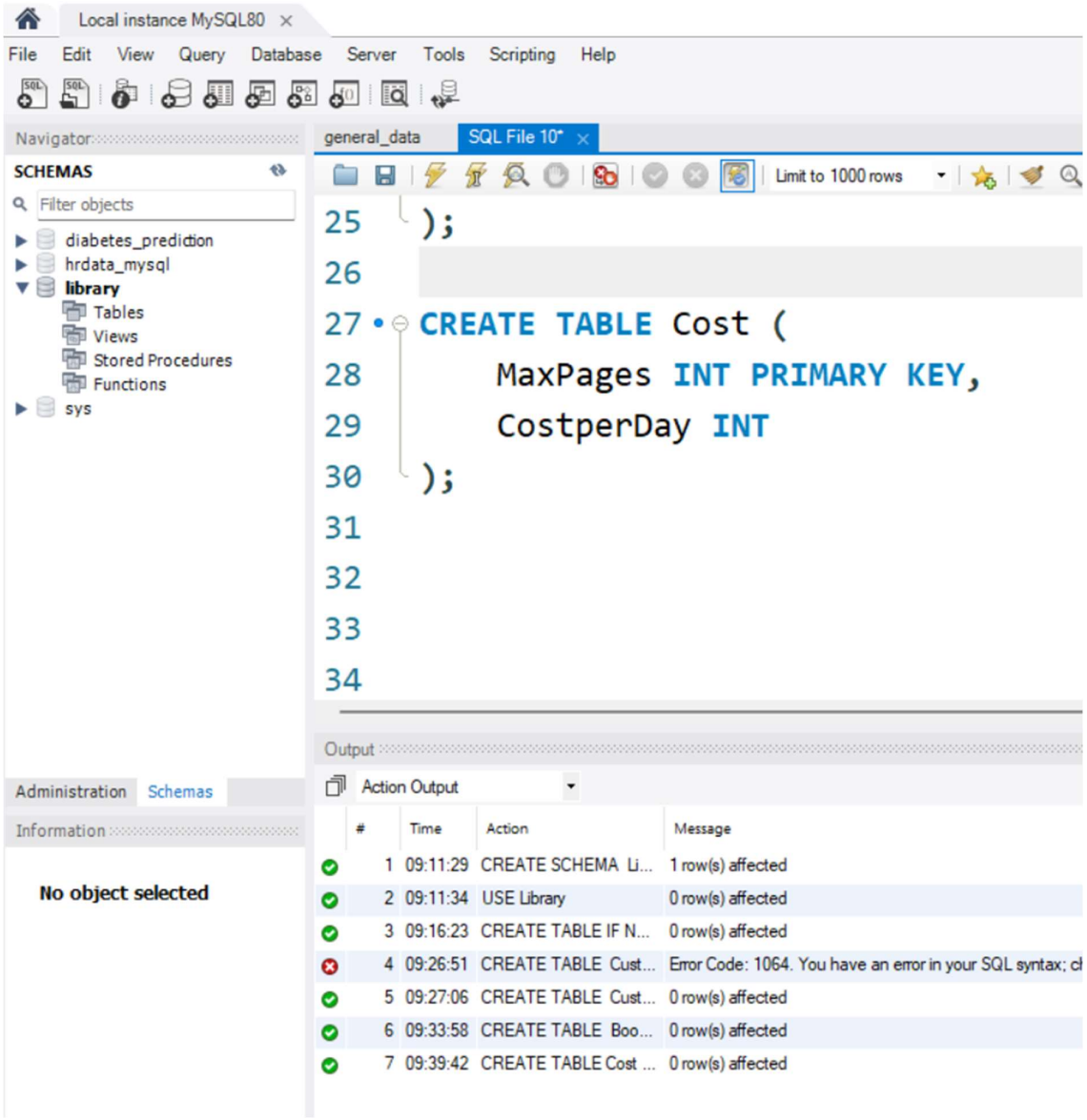
D) Creating Cost table:

Code: CREATE TABLE Cost (

MaxPages INT PRIMARY KEY,

CostperDay INT

);

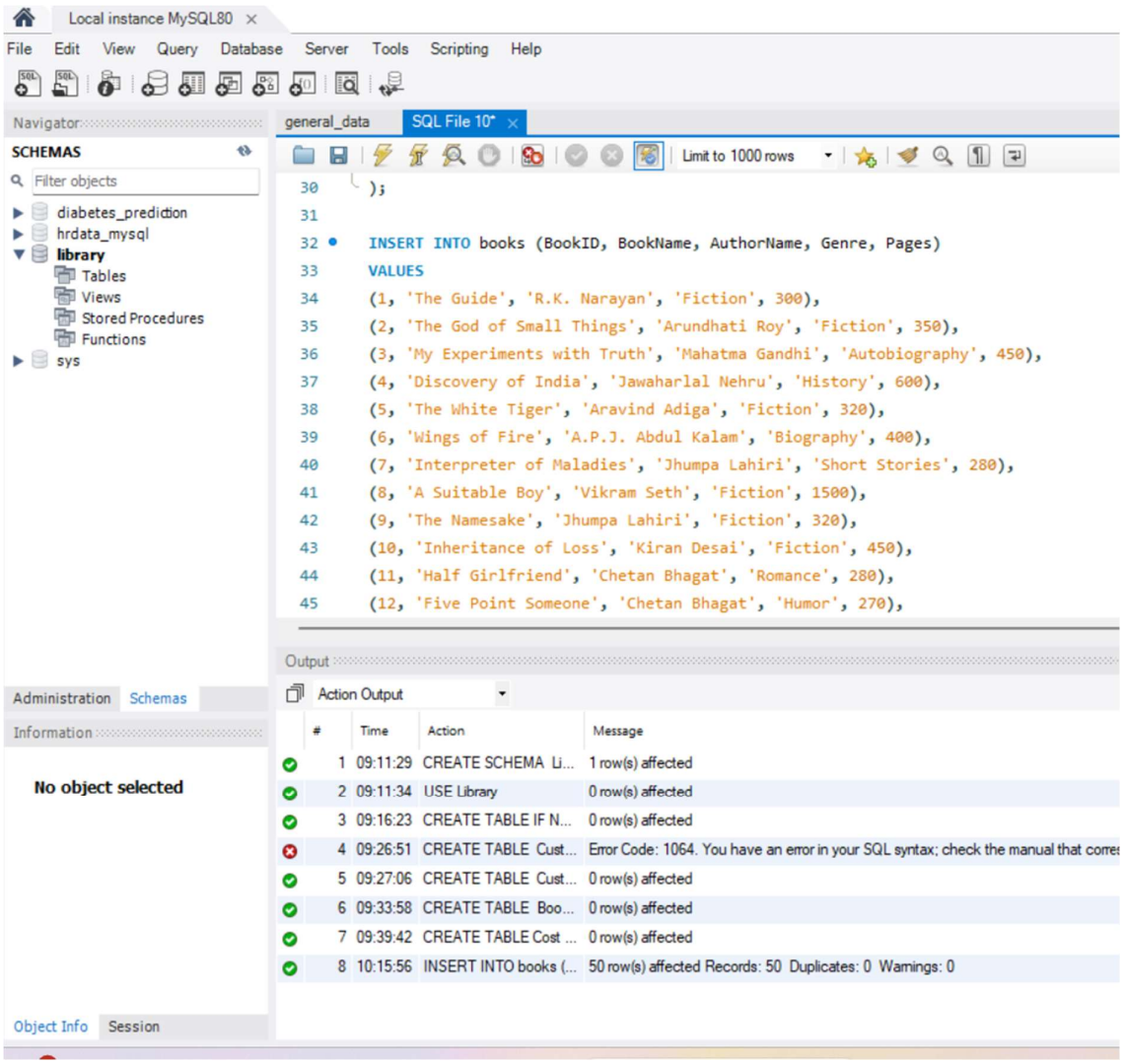

Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
diabetes_prediction
hrdata_mysql
library
Tables
Views
Stored Procedures
Functions
sys
Administration Schemas
Information
No object selected
general_data
SQL File 10*
Limit to 1000 rows
25);
26
27 • CREATE TABLE Cost (
28 MaxPages INT PRIMARY KEY,
29 CostperDay INT
30);
31
32
33
34
Output
Action Output
Time Action Message
1 09:11:29 CREATE SCHEMA Li... 1 row(s) affected
2 09:11:34 USE Library 0 row(s) affected
3 09:16:23 CREATE TABLE IF N... 0 row(s) affected
4 09:26:51 CREATE TABLE Cust... Error Code: 1064. You have an error in your SQL syntax; cl
5 09:27:06 CREATE TABLE Cust... 0 row(s) affected
6 09:33:58 CREATE TABLE Boo... 0 row(s) affected
7 09:39:42 CREATE TABLE Cost ... 0 row(s) affected

3) 3.Google Search for books along with authors, genre and pages. Insert atleast 5 authors and 5 different genres into Books Table. Make sure you have atleast 30 records.

A) Insering values into books table :

Code: INSERT INTO books (BookID, BookName, AuthorName, Genre, Pages)

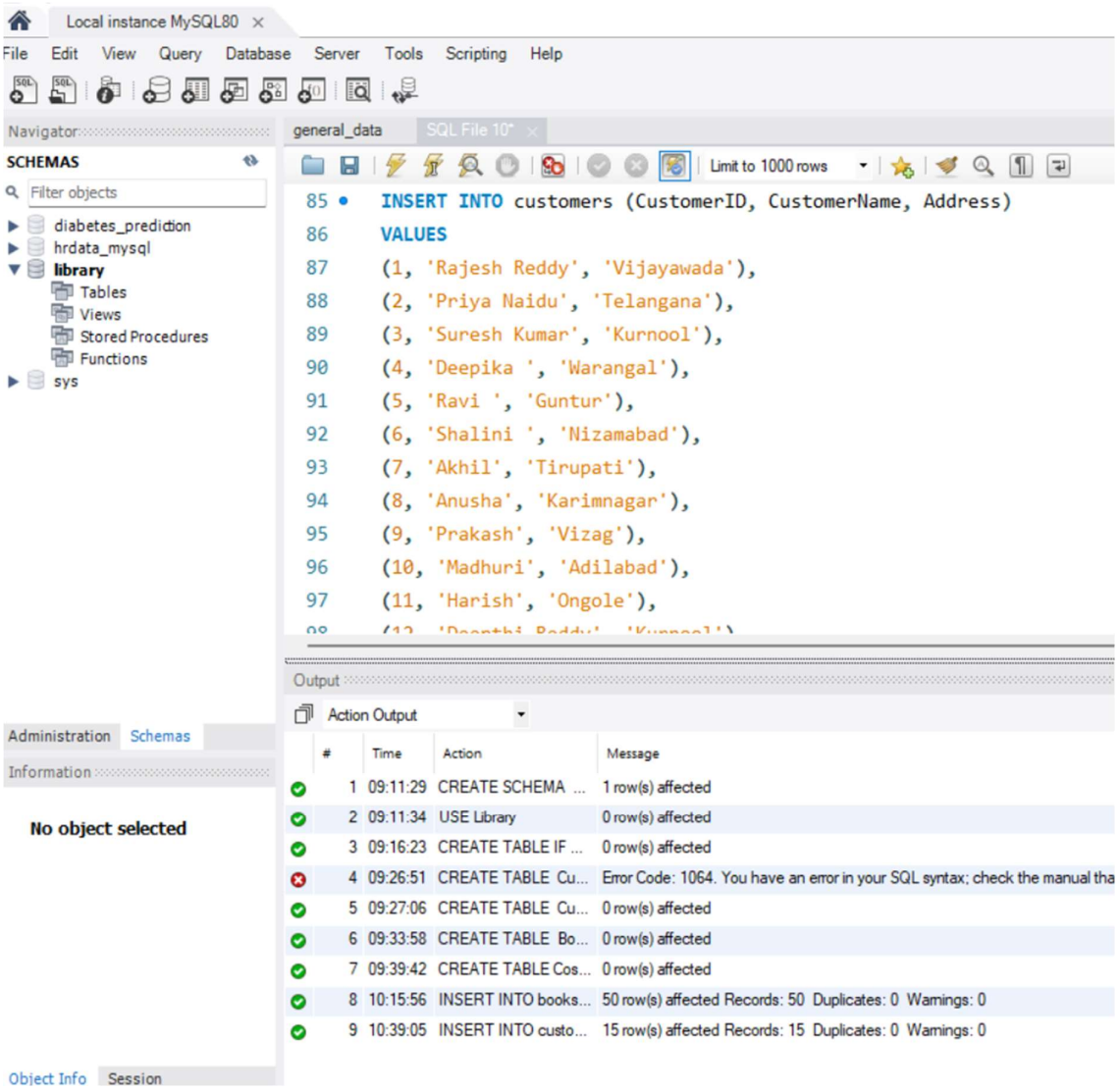
VALUES (1, 'The Guide', 'R.K. Narayan', 'Fiction', 300);



4) Insert 15 customers into customer table by providing appropriate details.

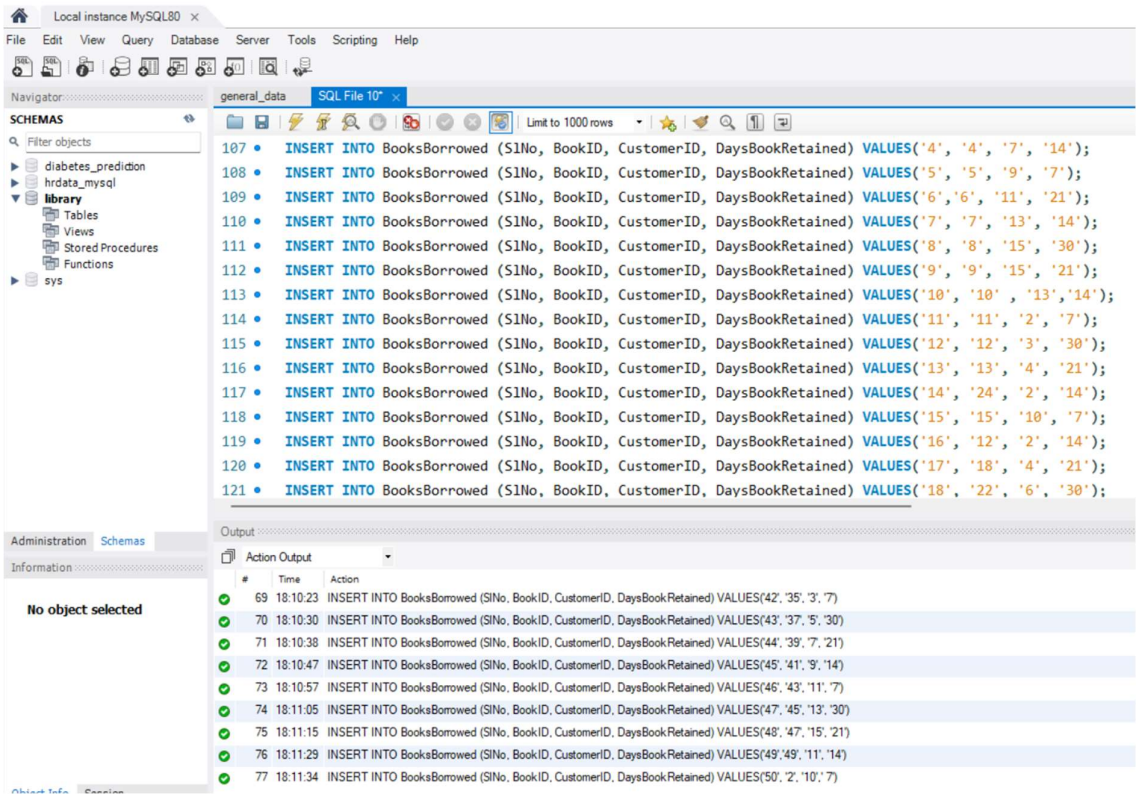
Code: INSERT INTO customers (CustomerID, CustomerName, Address) VALUES

(1, 'Rajesh Reddy', 'Vijayawada'),



5. Using BookId in Books Table and CustomerID in Customer Table insert data into BooksBorrowed table to have atleast 50 records.

Code: INSERT INTO BooksBorrowed (SlNo, BookID, CustomerID, DaysBookRetained) VALUES ('1',' 1', '1', '14');

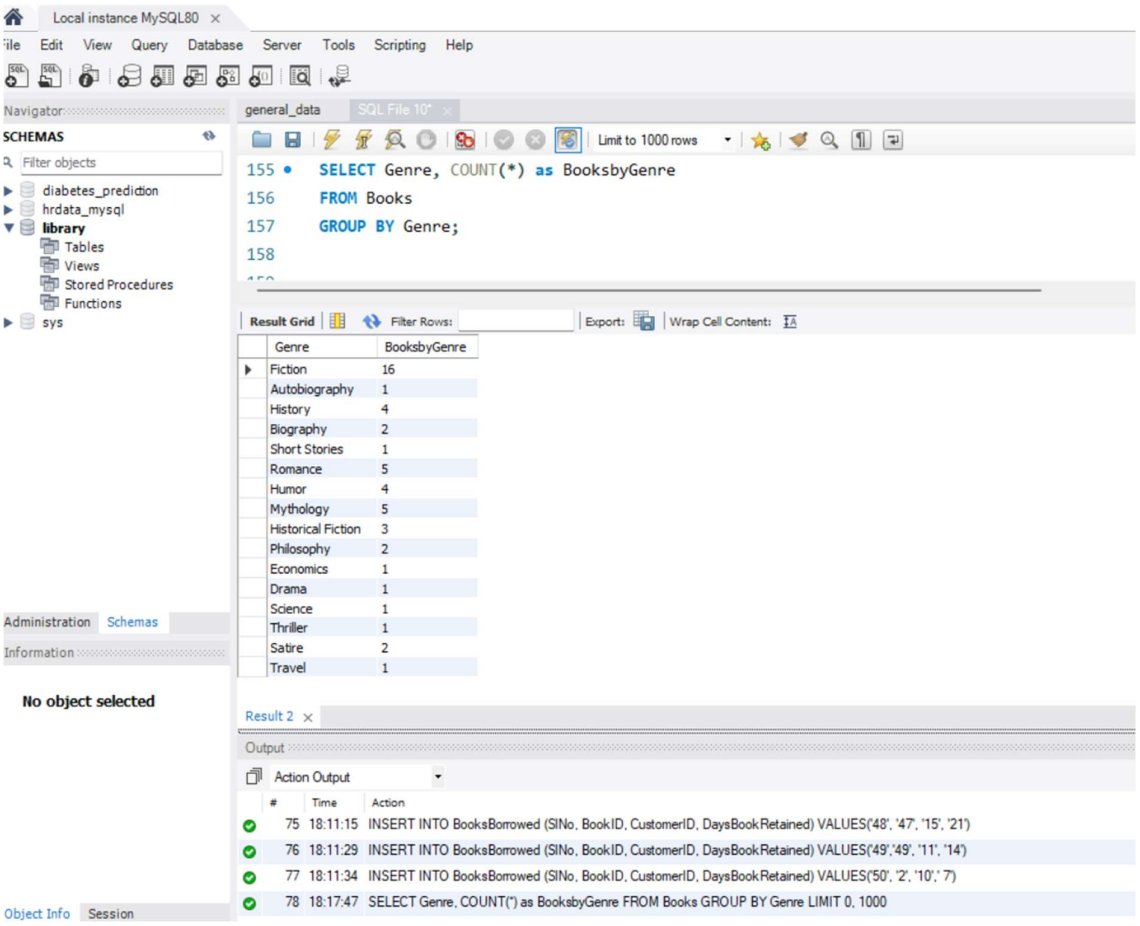


6. Retrieve the total number of books in each genre.

Code: SELECT Genre, COUNT(*) as BooksbyGenre

FROM Books

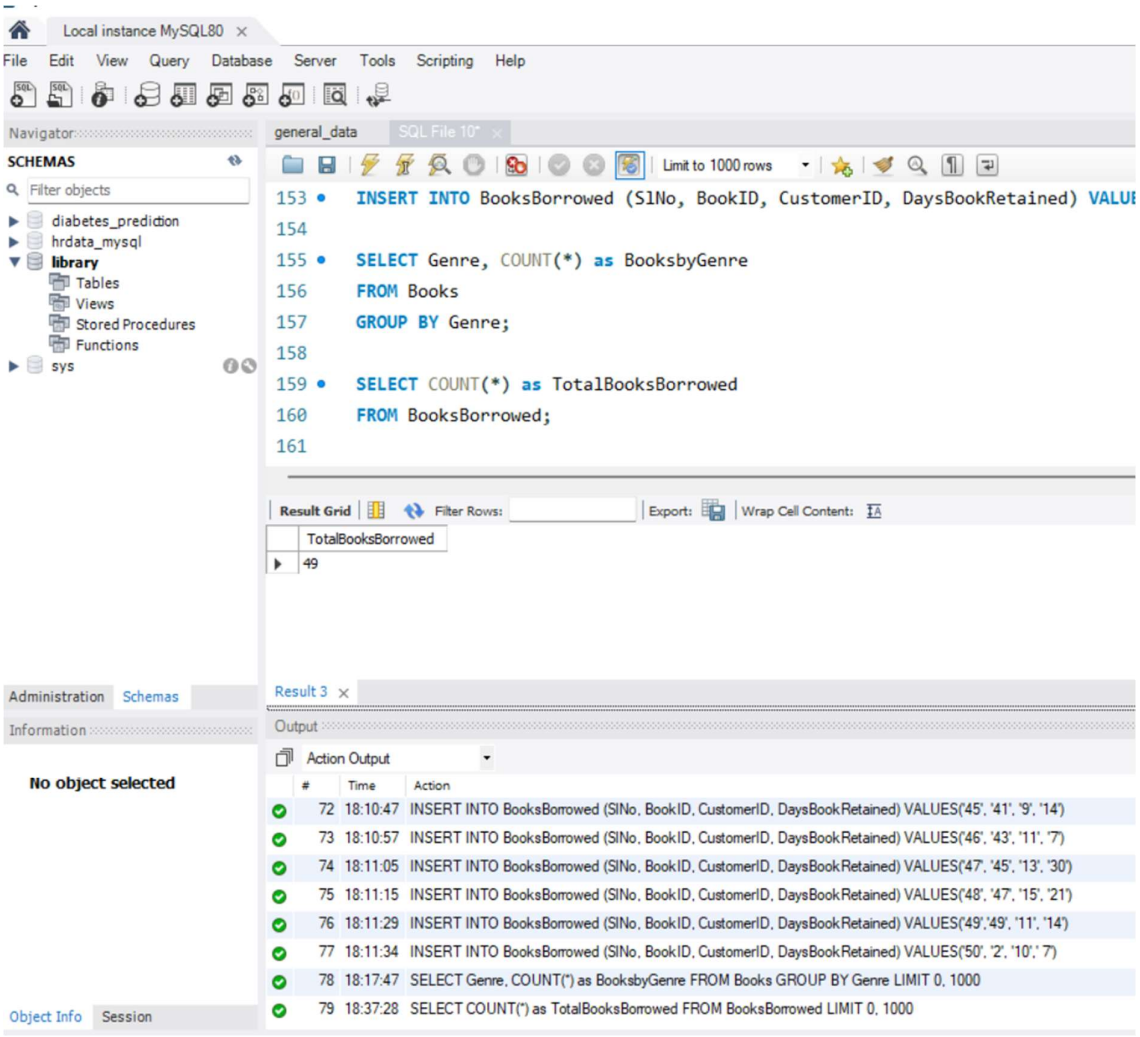
GROUP BY Genre;



7. Retrieve total number of books borrowed.

Code: SELECT COUNT(*) as TotalBooksBorrowed

FROM BooksBorrowed;



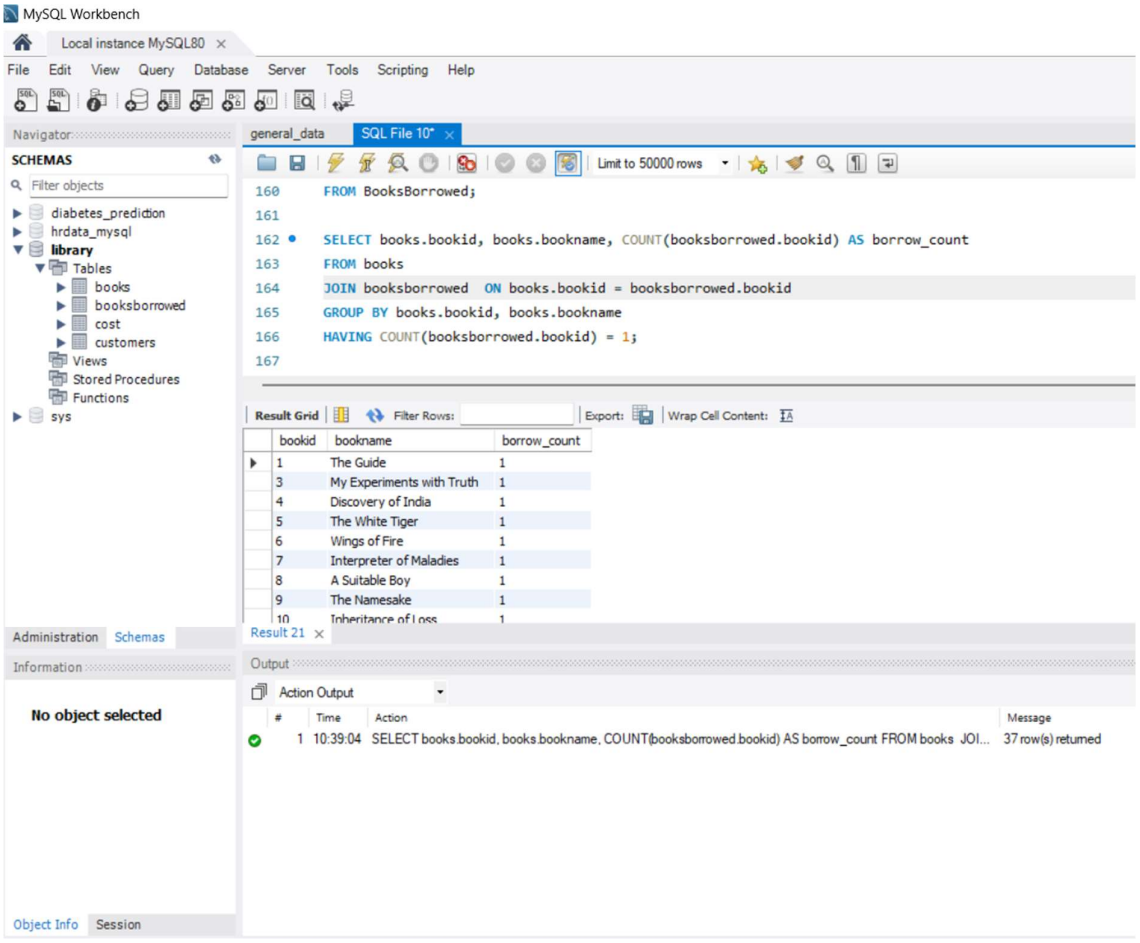
8. retrieve names of books borrowed without repetition:

Code: SELECT books.bookid, books.bookname, COUNT(booksborrowed.bookid) AS borrow_count FROM books

JOIN booksborrowed ON books.bookid = booksborrowed.bookid

GROUP BY books.bookid, books.bookname

HAVING COUNT(booksborrowed.bookid) = 1;



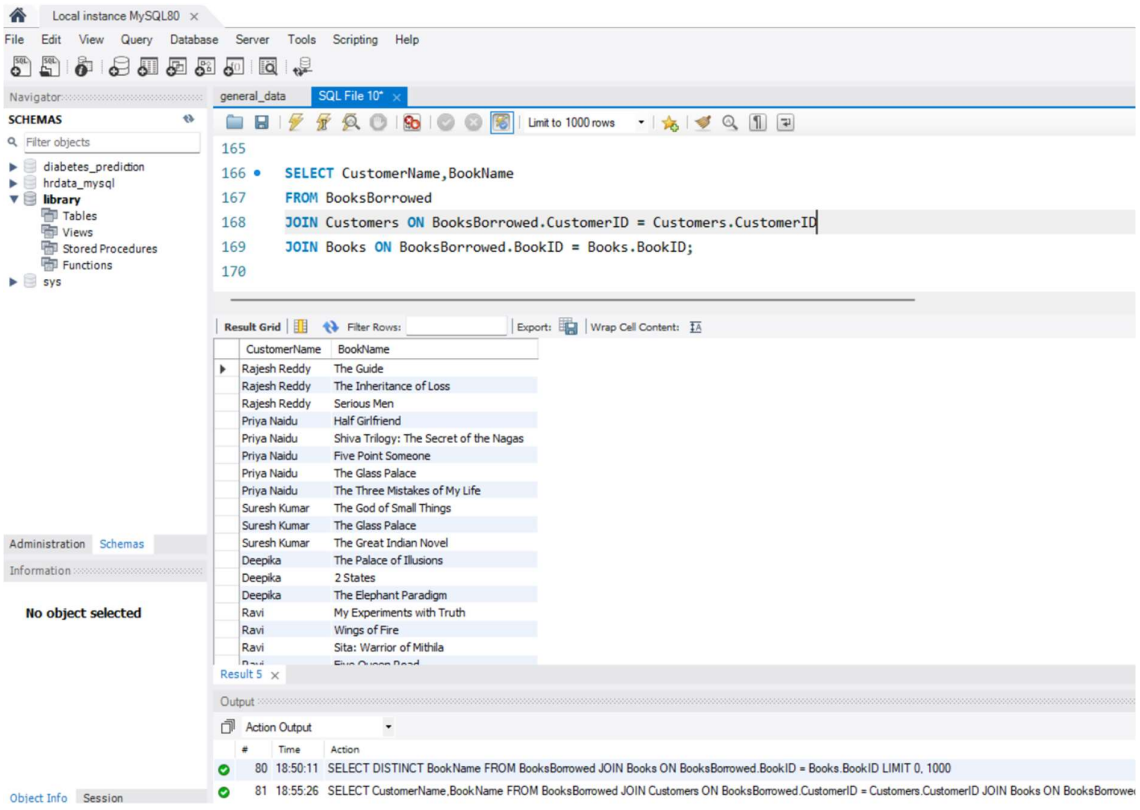
9. Retrieve the customerName, BookName for books borrowed.

Code: SELECT Customers.CustomerName, Books.BookName

FROM BooksBorrowed

JOIN Customers ON BooksBorrowed.CustomerID = Customers.CustomerID

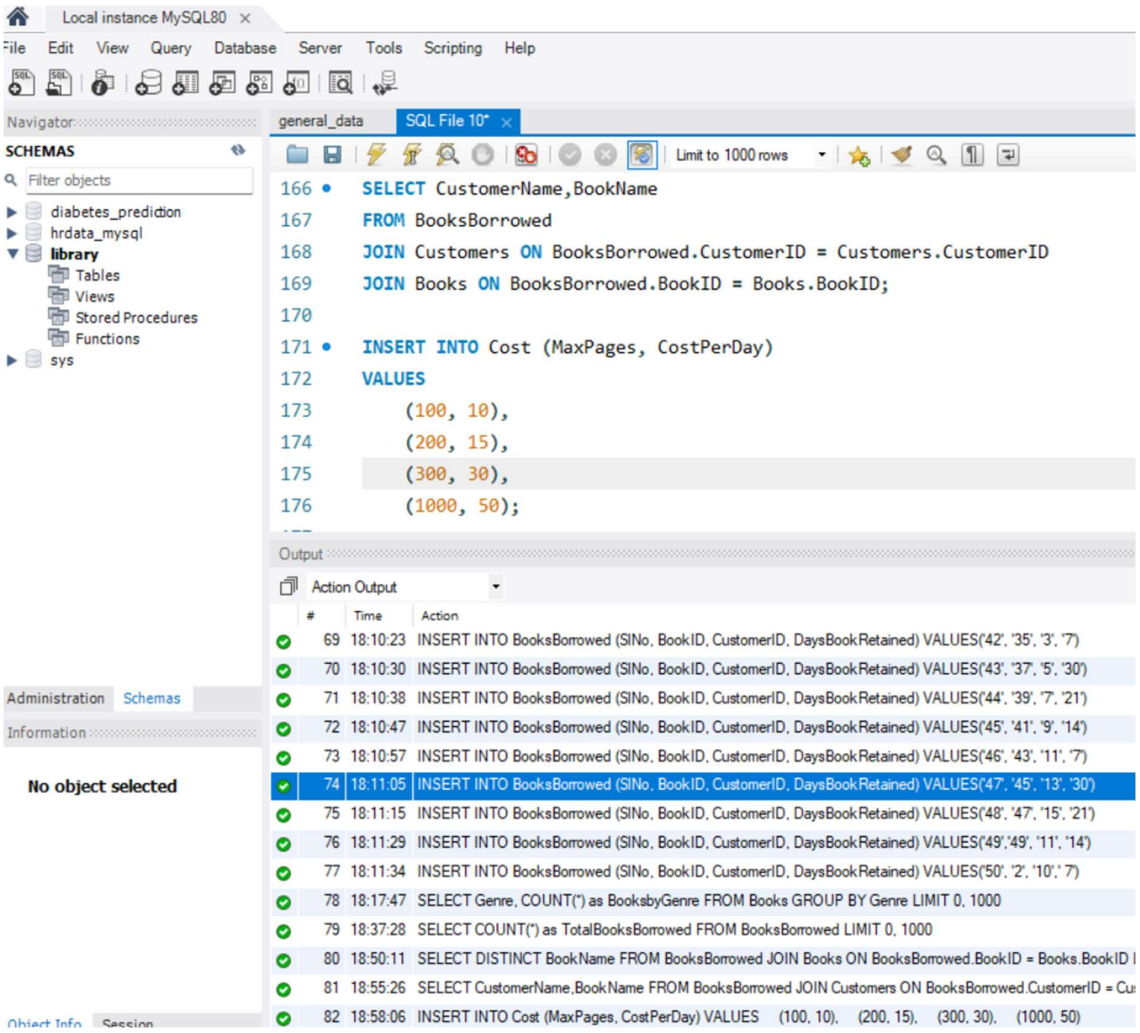
JOIN Books ON BooksBorrowed.BookID = Books.BookID;



10. Insert below data into Cost table 100pages -10 rs, 200pages -15rs, 300 pages - 30rs, 1000 pages -50Rs.

Code: INSERT INTO Cost (MaxPages, CostPerDay)

VALUES (100, 10),(200, 15),(300, 30), (1000, 50);

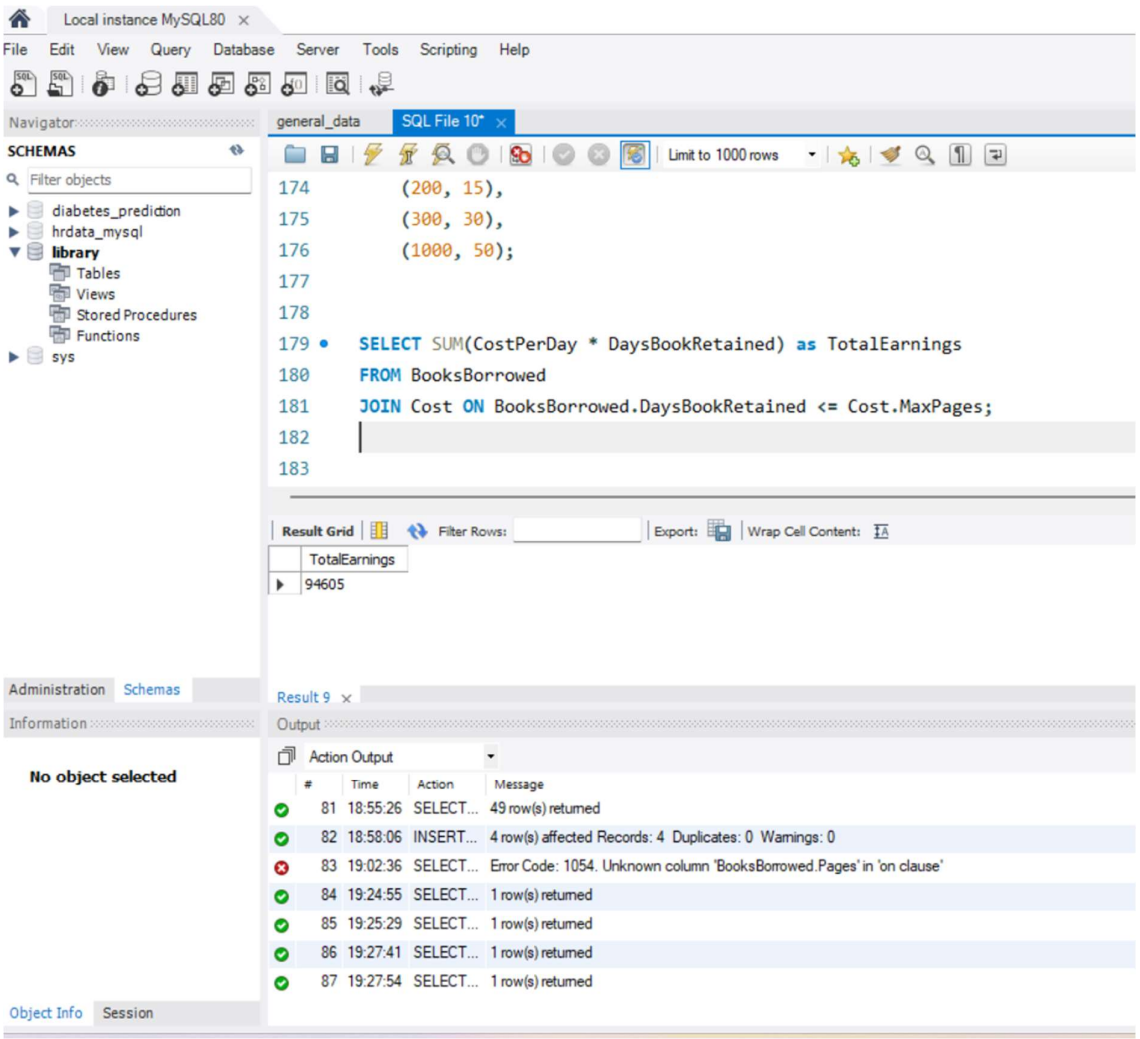


11. total Earnings by Library.

Code: SELECT SUM(CostPerDay * DaysBookRetained) as TotalEarnings

FROM BooksBorrowed

JOIN Cost ON BooksBorrowed.DaysBookRetained <= Cost.MaxPages;



12. Amount paid by each customer. Rank the customers based in amount paid.

Code: SELECT Customers.CustomerName, SUM(CostPerDay * DaysBookRetained) as AmountPaid FROM BooksBorrowed

JOIN Customers ON BooksBorrowed.CustomerID = Customers.CustomerID

JOIN Books ON BooksBorrowed.BookID = Books.BookID

JOIN Cost ON Books.Pages <= Cost.MaxPages

GROUP BY Customers.CustomerName

ORDER BY AmountPaid DESC;