**ASSIGNMENT-SQL**

1. 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

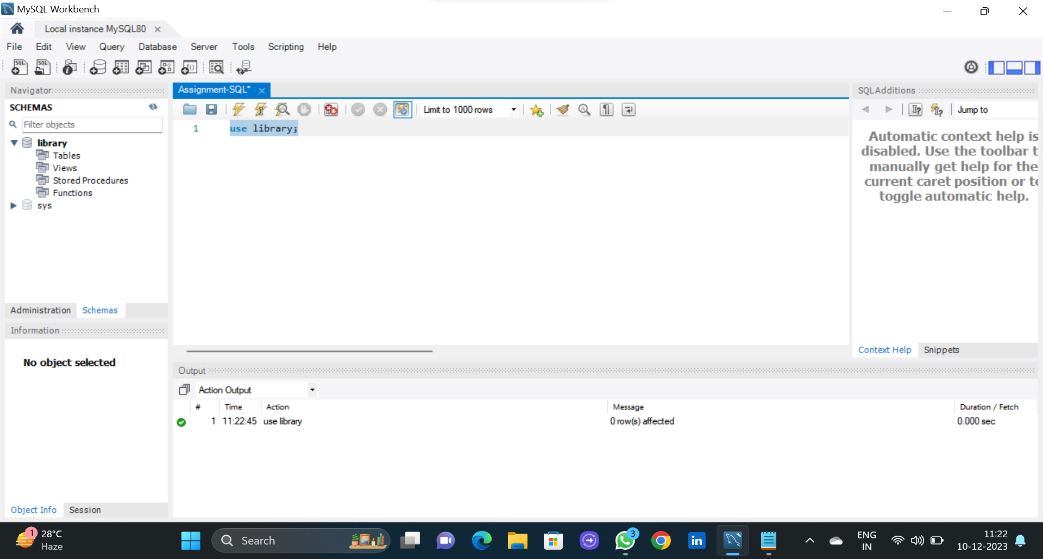
Cost: MaxPages, Cost/day

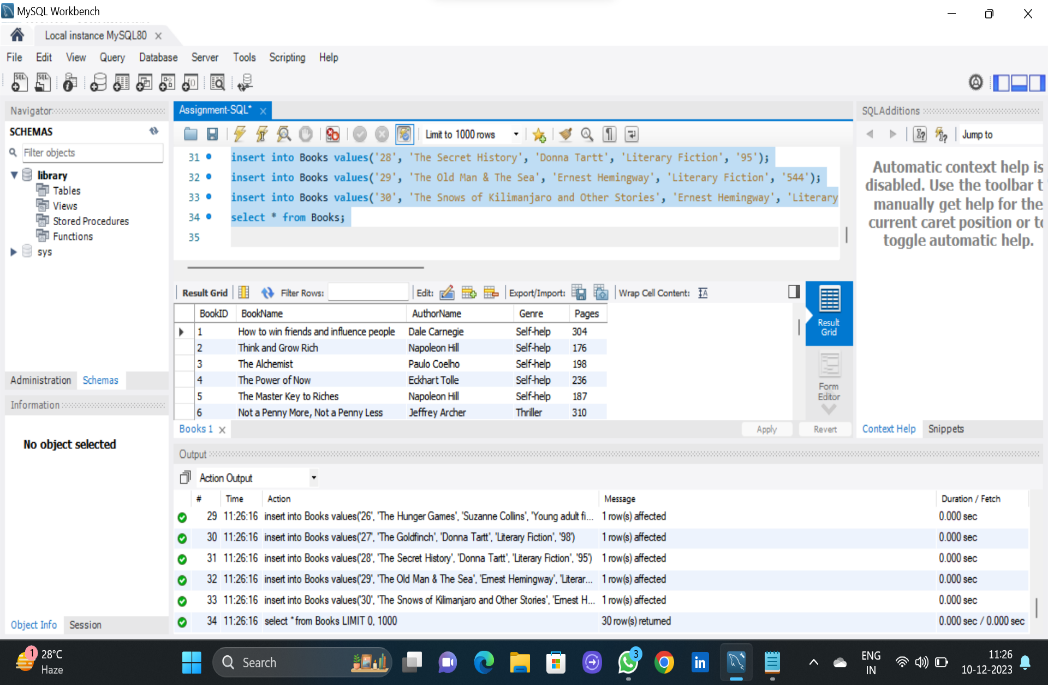
3.Google Search for books along with authors, genre and pages. Insert atleast 5 authors and 5 different genres into Books Table. Make sure you have atleast 30 records.

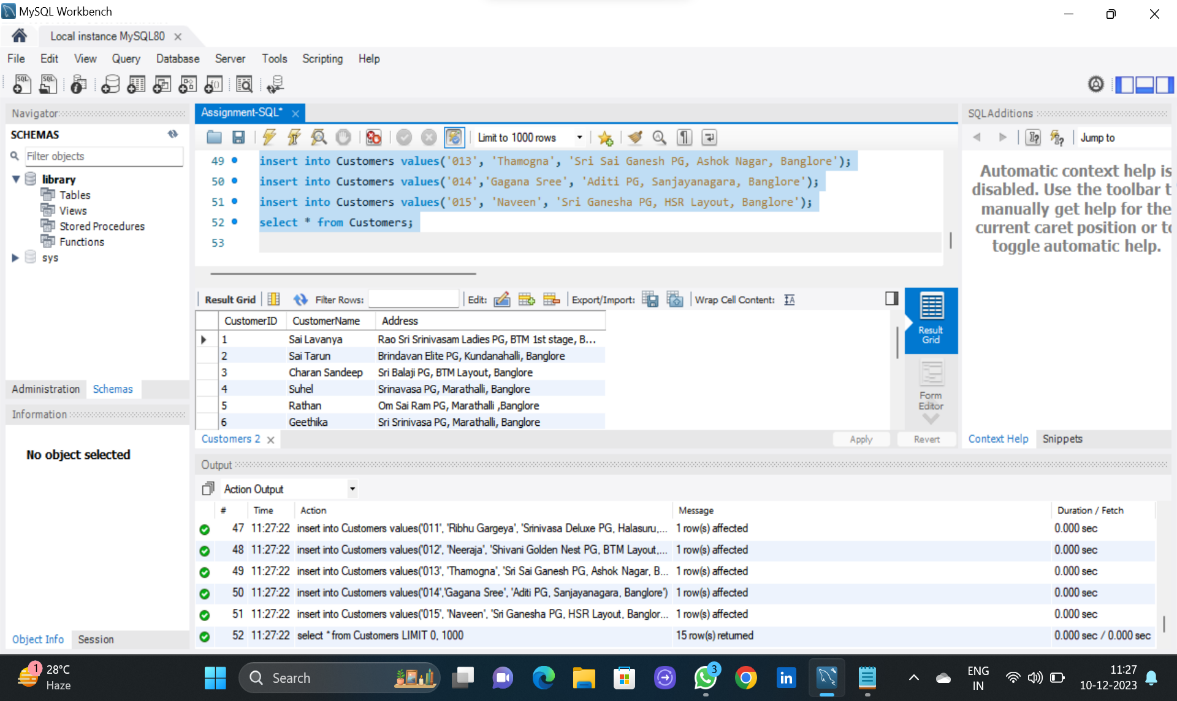
4. Insert 15 customers into customer table by providing appropriate details

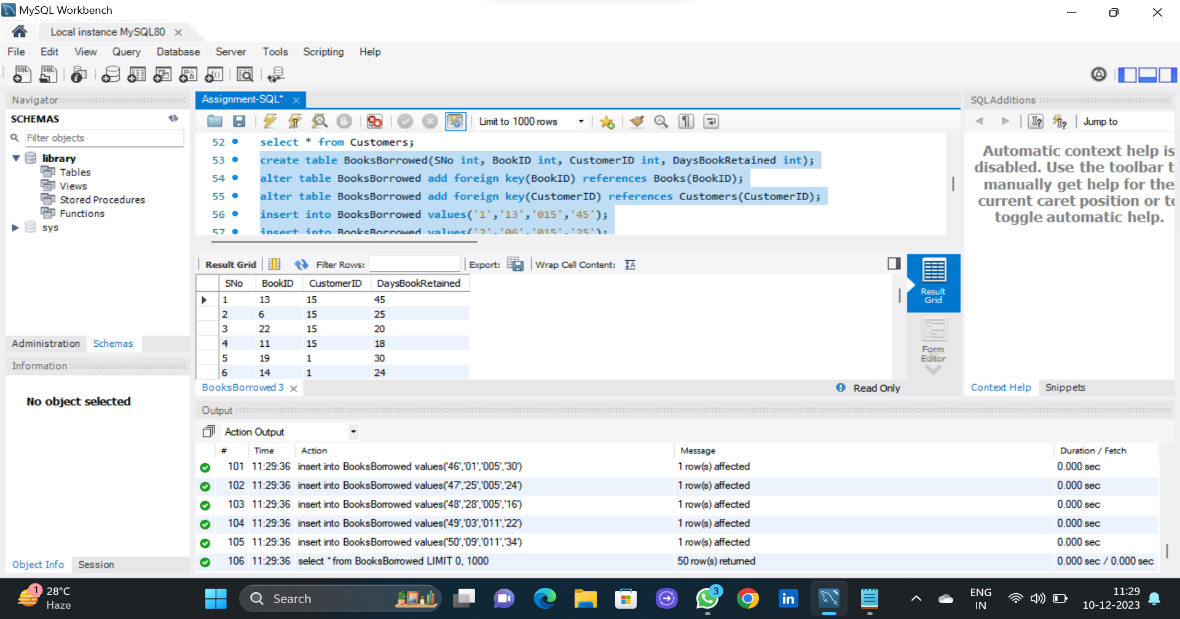
5. Using BookId in Books Table and CustomerID in Customer Table insert data into BooksBorrowed table to have atleast 50 records.

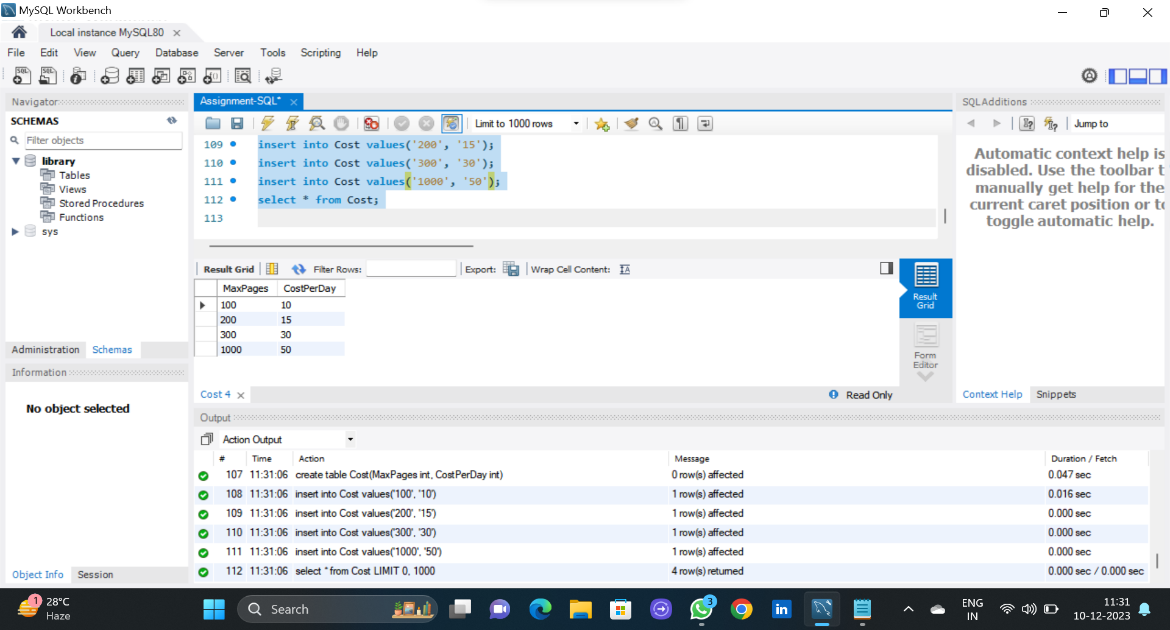
10. Insert data into cost table 100pages -10 rs, 200pages -15rs, 300 pages - 30rs, 1000 pages-50Rs



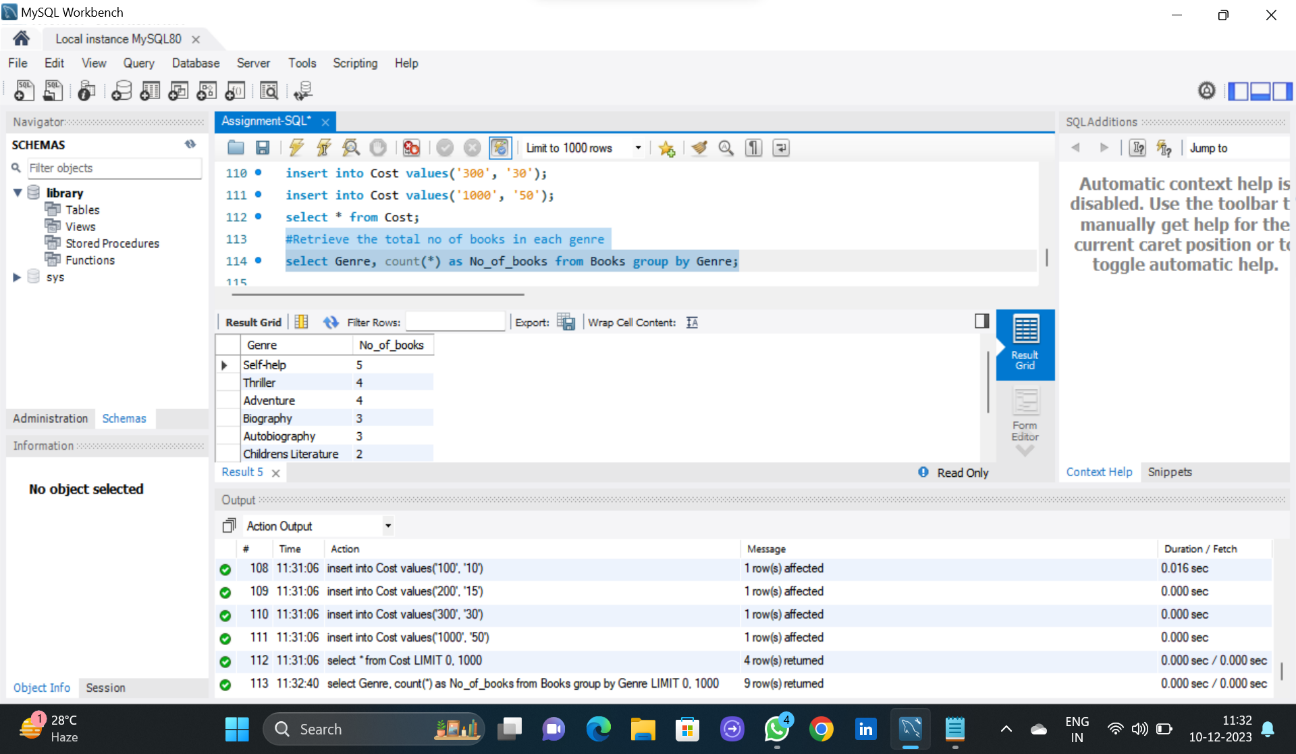




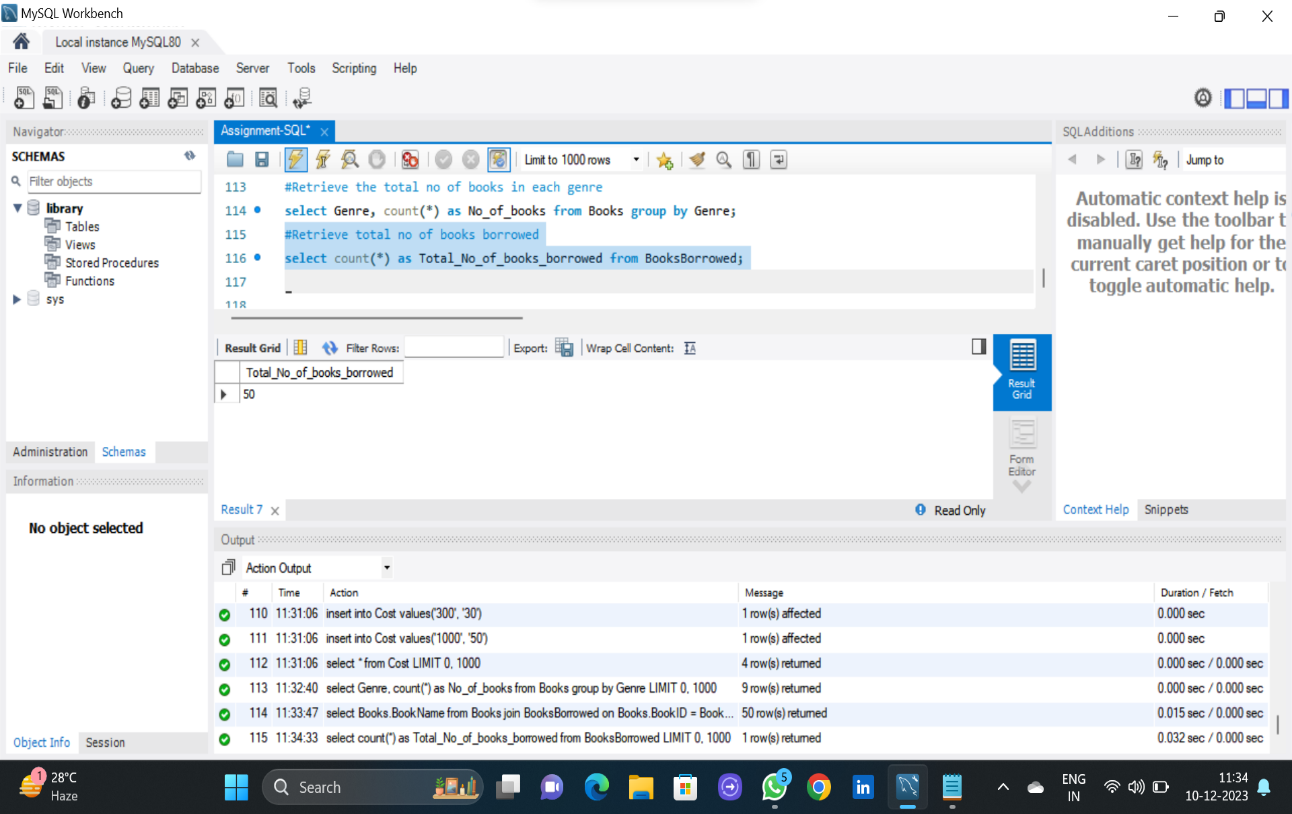




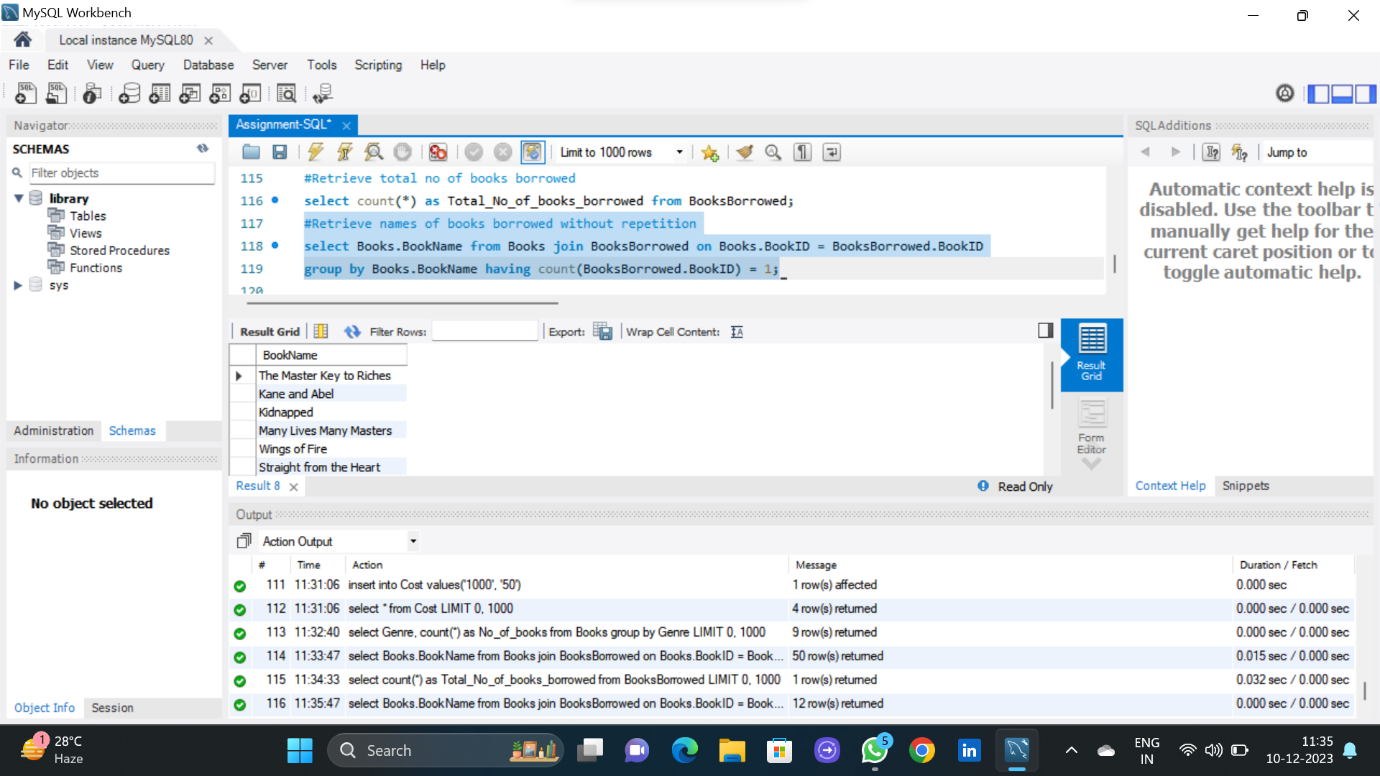
6. Retrieve the total number of books in each genre.



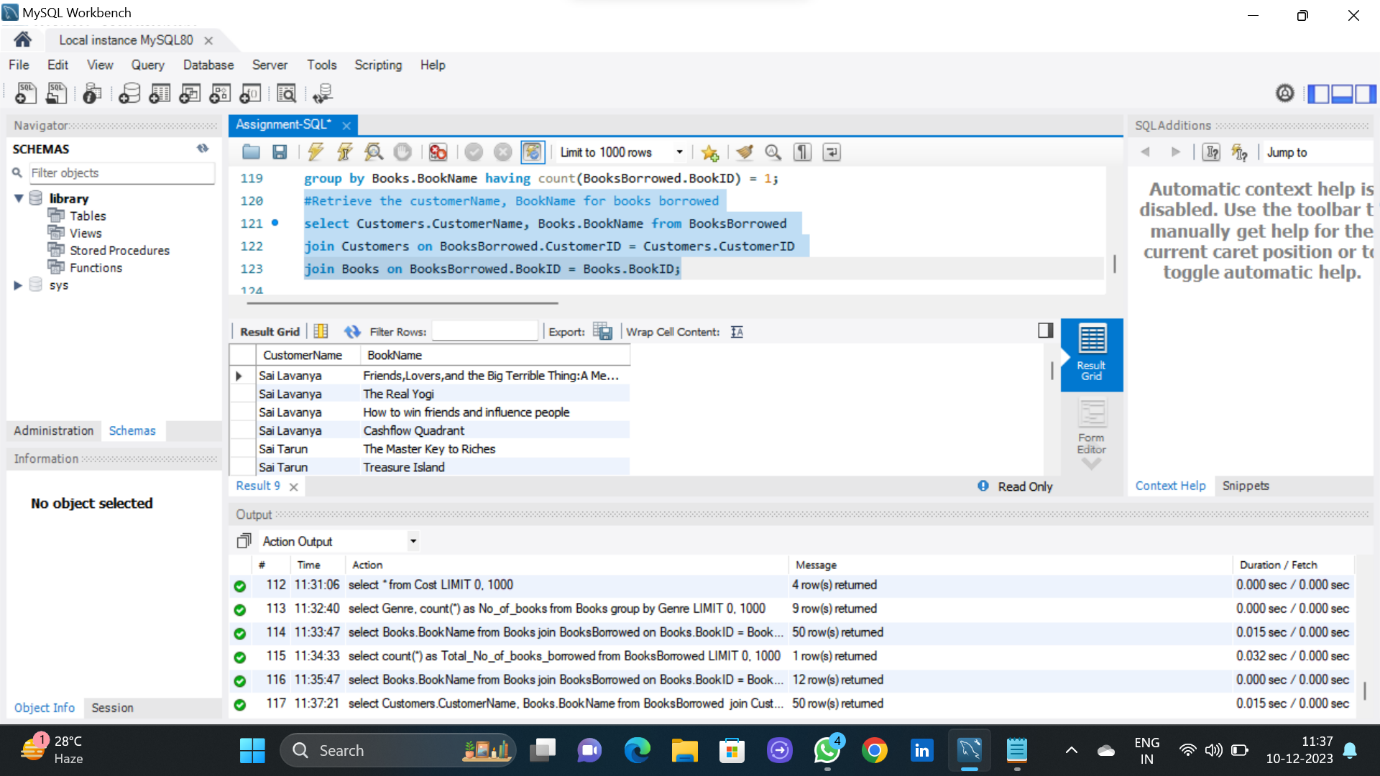
7. Retrieve total number of books borrowed.



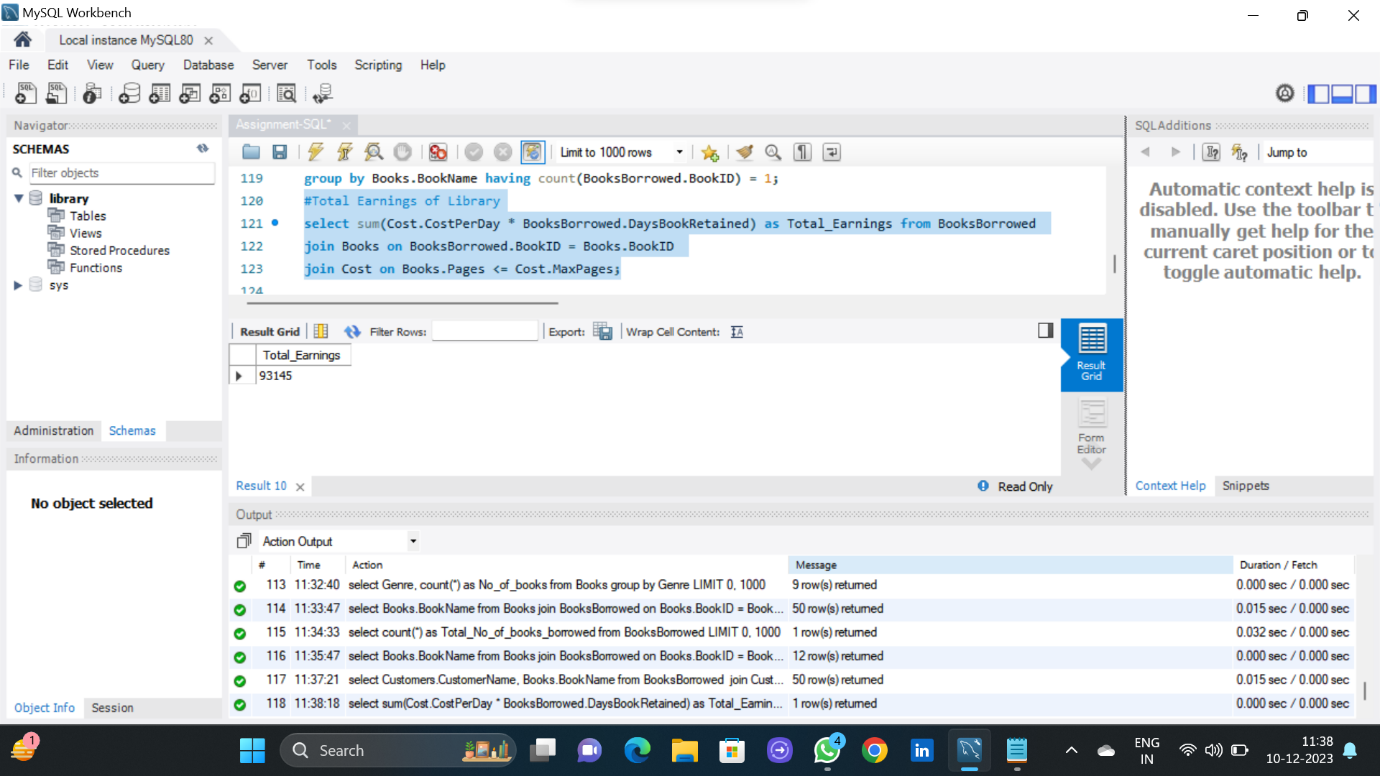
8. Retrieve names of books borrowed without repetition.



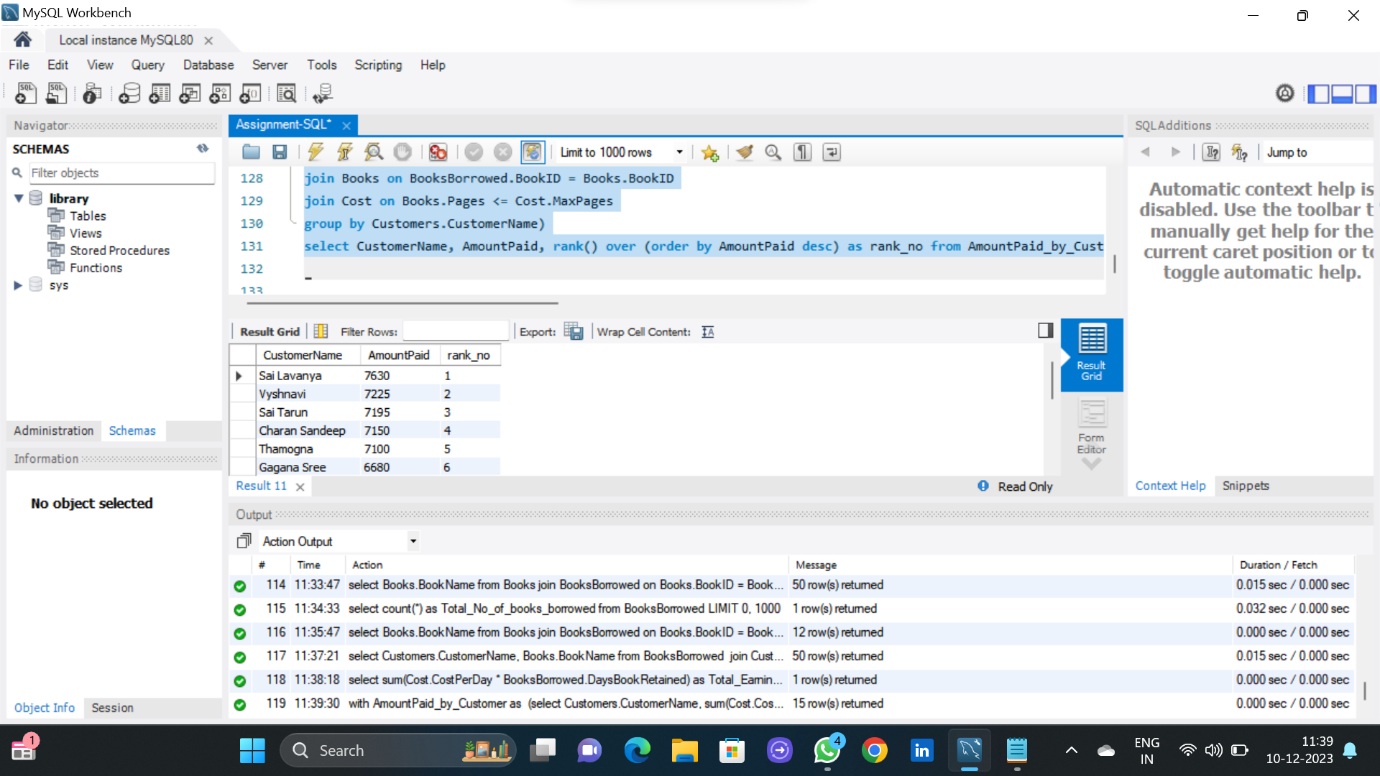
9. Retrieve the CustomerName, BookName for books borrowed.



11. Total Earnings by Library.



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



use library;

create table Books(BookID int, BookName varchar(100), AuthorName varchar(50), Genre varchar(50), Pages int);

alter table Books add primary key(BookID);

insert into Books values('01', 'How to win friends and influence people', 'Dale Carnegie', 'Self-help', '304');

insert into Books values('02', 'Think and Grow Rich', 'Napoleon Hill', 'Self-help', '176');

insert into Books values('03', 'The Alchemist', 'Paulo Coelho', 'Self-help', '198');

insert into Books values('04', 'The Power of Now', 'Eckhart Tolle', 'Self-help', '236');

insert into Books values('05', 'The Master Key to Riches', 'Napoleon Hill', 'Self-help','187');

insert into Books values('06', 'Not a Penny More, Not a Penny Less', 'Jeffrey Archer', 'Thriller', '310');

insert into Books values('07', 'Kane and Abel', 'Jeffrey Archer', 'Thriller', '560');

insert into Books values('08', 'A Prisoner of Birth', 'Jeffrey Archer', 'Thriller', '624');

insert into Books values('09', 'Gone Girl', 'Gillian Flynn', 'Thriller', '466');

insert into Books values('10', 'Kidnapped', 'Robert Louis Stevenson', 'Adventure', '188');

insert into Books values('11', 'Treasure Island', 'Robert Louis Stevenson', 'Adventure', '236');

insert into Books values('12', 'The Three Musketeers', 'Alexandre Dumas', 'Adventure', '696');

insert into Books values('13', 'The Count of Monte Cristo', 'Alexandre Dumas', 'Adventure', '989');

insert into Books values('14','The Real Yogi', 'Gana', 'Biography', '165');

insert into Books values('15', 'Into the Wild', 'Jon Krakauer', 'Biography', '240');

insert into Books values('16', 'Many Lives Many Masters','Brian Weiss', 'Biography', '224');

insert into Books values('17', 'Wings of Fire', 'APJ Abdul Kalam', 'Autobiography', '180');

insert into Books values('18', 'Straight from the Heart' ,'Kapil Dev', 'Autobiography', '374');

insert into Books values('19', 'Friends,Lovers,and the Big Terrible Thing:A Memoir', 'Mathew Perry', 'Autobiography', '272');

insert into Books values('20', 'The Jungle Book', 'Robert Louis Stevenson', 'Childrens Literature', '206');

insert into Books values('21', 'The Little Prince', 'Antoine de Saint-Exupery', 'Childrens Literature', '96');

insert into Books values('22', 'Rich Dad Poor Dad', 'Robert Kiyosaki', 'Personal Finance', '336');

insert into Books values('23', 'Cashflow Quadrant', 'Robert Kiyosaki', 'Personal Finance', '360');

insert into Books values('24', 'The fault in our stars', 'John Green', 'Young adult fiction', '290');

insert into Books values('25', 'Looking for Alaska', 'John Green', 'Young adult fiction', '217');

insert into Books values('26', 'The Hunger Games', 'Suzanne Collins', 'Young adult fiction', '384');

insert into Books values('27', 'The Goldfinch', 'Donna Tartt', 'Literary Fiction', '98');

insert into Books values('28', 'The Secret History', 'Donna Tartt', 'Literary Fiction', '95');

insert into Books values('29', 'The Old Man & The Sea', 'Ernest Hemingway', 'Literary Fiction', '544');

insert into Books values('30', 'The Snows of Kilimanjaro and Other Stories', 'Ernest Hemingway', 'Literary Fiction', '771');

select \* from Books;

create table Customers(CustomerID int, CustomerName varchar(50), Address varchar(100));

alter table Customers add primary key(CustomerID);

insert into Customers values('001', 'Sai Lavanya', 'Rao Sri Srinivasam Ladies PG, BTM 1st stage, Banglore');

insert into Customers values('002', 'Sai Tarun', 'Brindavan Elite PG, Kundanahalli, Banglore');

insert into Customers values('003', 'Charan Sandeep', 'Sri Balaji PG, BTM Layout, Banglore');

insert into Customers values('004', 'Suhel', 'Srinavasa PG, Marathalli, Banglore');

insert into Customers values('005', 'Rathan', 'Om Sai Ram PG, Marathalli ,Banglore');

insert into Customers values('006', 'Geethika', 'Sri Srinivasa PG, Marathalli, Banglore');

insert into Customers values('007', 'Sri Charan', 'Pelagia Paradise PG, Hosur road, Banglore');

insert into Customers values('008', 'Vyshnavi', 'Nandhini PG, BTM 1st stage, Banglore');

insert into Customers values('009', 'Samyuktha', 'Om Sai Balaji PG, BTM Layout, Banglore');

insert into Customers values('010', 'Purvic', 'GoodLands PG, SRK Arcade, Taverkare, Banglore');

insert into Customers values('011', 'Ribhu Gargeya', 'Srinivasa Deluxe PG, Halasuru, Banglore');

insert into Customers values('012', 'Neeraja', 'Shivani Golden Nest PG, BTM Layout, Banglore');

insert into Customers values('013', 'Thamogna', 'Sri Sai Ganesh PG, Ashok Nagar, Banglore');

insert into Customers values('014','Gagana Sree', 'Aditi PG, Sanjayanagara, Banglore');

insert into Customers values('015', 'Naveen', 'Sri Ganesha PG, HSR Layout, Banglore');

select \* from Customers;

create table BooksBorrowed(SNo int, BookID int, CustomerID int, DaysBookRetained int);

alter table BooksBorrowed add foreign key(BookID) references Books(BookID);

alter table BooksBorrowed add foreign key(CustomerID) references Customers(CustomerID);

insert into BooksBorrowed values('1','13','015','45');

insert into BooksBorrowed values('2','06','015','25');

insert into BooksBorrowed values('3','22','015','20');

insert into BooksBorrowed values('4','11','015','18');

insert into BooksBorrowed values('5','19','001','30');

insert into BooksBorrowed values('6','14','001','24');

insert into BooksBorrowed values('7','01','001','26');

insert into BooksBorrowed values('8','23','001','33');

insert into BooksBorrowed values('9','02','009','19');

insert into BooksBorrowed values('10','09','009','31');

insert into BooksBorrowed values('11','25','009','21');

insert into BooksBorrowed values('12','30','006','50');

insert into BooksBorrowed values('13','26','006','23');

insert into BooksBorrowed values('14','29','006','37');

insert into BooksBorrowed values('15','02','012','20');

insert into BooksBorrowed values('16','15','012','27');

insert into BooksBorrowed values('17','03','012','23');

insert into BooksBorrowed values('18','17','007','25');

insert into BooksBorrowed values('19','21','007','13');

insert into BooksBorrowed values('20','08','007','40');

insert into BooksBorrowed values('21','14','013','28');

insert into BooksBorrowed values('22','04','013','16');

insert into BooksBorrowed values('23','07','013','38');

insert into BooksBorrowed values('24','27','013','12');

insert into BooksBorrowed values('25','05','002','19');

insert into BooksBorrowed values('26','11','002','26');

insert into BooksBorrowed values('27','16','002','22');

insert into BooksBorrowed values('28','23','002','31');

insert into BooksBorrowed values('29','12','014','39');

insert into BooksBorrowed values('30','18','014','29');

insert into BooksBorrowed values('31','24','014','20');

insert into BooksBorrowed values('32','28','014','16');

insert into BooksBorrowed values('33','12','003','38');

insert into BooksBorrowed values('34','19','003','22');

insert into BooksBorrowed values('35','08','003','41');

insert into BooksBorrowed values('36','20','003','18');

insert into BooksBorrowed values('37','10','008','23');

insert into BooksBorrowed values('38','04','008','31');

insert into BooksBorrowed values('39','15','008','32');

insert into BooksBorrowed values('40','11','004','29');

insert into BooksBorrowed values('41','06','004','29');

insert into BooksBorrowed values('42','13','004','52');

insert into BooksBorrowed values('43','14','010','21');

insert into BooksBorrowed values('44','30','010','46');

insert into BooksBorrowed values('45','26','010','43');

insert into BooksBorrowed values('46','01','005','30');

insert into BooksBorrowed values('47','25','005','24');

insert into BooksBorrowed values('48','28','005','16');

insert into BooksBorrowed values('49','03','011','22');

insert into BooksBorrowed values('50','09','011','34');

select \* from BooksBorrowed;

create table Cost(MaxPages int, CostPerDay int);

insert into Cost values('100', '10');

insert into Cost values('200', '15');

insert into Cost values('300', '30');

insert into Cost values('1000', '50');

select \* from Cost;

#Retrieve the total no of books in each genre

select Genre, count(\*) as No\_of\_books from Books group by Genre;

#Retrieve total no of books borrowed

select count(\*) as Total\_No\_of\_books\_borrowed from BooksBorrowed;

#Retrieve names of books borrowed without repetition

select Books.BookName from Books join BooksBorrowed on Books.BookID = BooksBorrowed.BookID

group by Books.BookName having count(BooksBorrowed.BookID) = 1;

#Retrieve the customerName, BookName for books borrowed

select Customers.CustomerName, Books.BookName from BooksBorrowed

join Customers on BooksBorrowed.CustomerID = Customers.CustomerID

join Books on BooksBorrowed.BookID = Books.BookID;

#Total Earnings of Library

select sum(Cost.CostPerDay \* BooksBorrowed.DaysBookRetained) as Total\_Earnings from BooksBorrowed

join Books on BooksBorrowed.BookID = Books.BookID

join Cost on Books.Pages <= Cost.MaxPages;

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

with AmountPaid\_by\_Customer as

(select Customers.CustomerName, sum(Cost.CostPerDay \* BooksBorrowed.DaysBookRetained) as AmountPaid from Customers

join BooksBorrowed on Customers.CustomerID = BooksBorrowed.CustomerID

join Books on BooksBorrowed.BookID = Books.BookID

join Cost on Books.Pages <= Cost.MaxPages

group by Customers.CustomerName)

select CustomerName, AmountPaid, rank() over (order by AmountPaid desc) as rank\_no from AmountPaid\_by\_Customer;