Lab-7

Book database

**Code:**

create database book\_db;

use book\_db;

CREATE TABLE Publisher

(name VARCHAR(20) PRIMARY KEY,

phone real,

address VARCHAR(20));

CREATE TABLE Book

(book\_id int PRIMARY KEY,

title VARCHAR(20),

pub\_year VARCHAR(20),

publisher\_name varchar(20) REFERENCES Publisher (name) ON DELETE CASCADE);

CREATE TABLE Book\_Authors

(author\_name VARCHAR(20),

book\_id int REFERENCES Book (book\_id) ON DELETE CASCADE,

PRIMARY KEY (book\_id, author\_name));

CREATE TABLE Library\_Branch

(branch\_id int PRIMARY KEY,

branch\_name VARCHAR(50),

address VARCHAR(50));

CREATE TABLE Book\_Copies

(no\_of\_copies int,

book\_id int REFERENCES Book(book\_id) ON DELETE CASCADE,

branch\_id int REFERENCES Library\_Branch (branch\_id) ON DELETE CASCADE,

PRIMARY KEY (book\_id, branch\_id));

CREATE TABLE Card

(card\_no int PRIMARY KEY);

CREATE TABLE Book\_Lending

(date\_out DATE,

due\_date DATE,

book\_id int REFERENCES book (book\_id) ON DELETE CASCADE,

branch\_id int REFERENCES Library\_Branch (branch\_id) ON DELETE CASCADE,

card\_no int REFERENCES Card (card\_no) ON DELETE CASCADE,

PRIMARY KEY (book\_id, branch\_id, card\_no));

INSERT INTO Publisher VALUES

("MCGRAW-HILL", 9989076587, "BANGALORE"),

("PEARSON",9889076565,"NEWDELHI"),

("RANDOM HOUSE",7455679345,"HYDRABAD"),

("HACHETTE LIVRE",8970862340,"CHENNAI"),

("GRUPO PLANETA",7756120238,"BANGALORE");

INSERT INTO Book VALUES

(1,"DBMS","JAN-2017","MCGRAW-HILL"),

(2,"ADBMS","JUN-2016","MCGRAW-HILL"),

(3,"CN","SEP-2016","PEARSON"),

(4,"CG","SEP-2015","GRUPO PLANETA"),

(5,"OS","MAY-2016","PEARSON");

INSERT INTO Book\_Authors VALUES

("NAVATHE", 1),

("NAVATHE", 2),

("TANENBAUM", 3),

("EDWARD ANGEL", 4),

("GALVIN", 5);

INSERT INTO Library\_Branch VALUES

(10,"RR NAGAR","BANGALORE"),

(11,"RNSIT","BANGALORE"),

(12,"RAJAJI NAGAR","BANGALORE"),

(13,"NITTE","MANGALORE"),

(14,"MANIPAL","UDUPI");

INSERT INTO Book\_Copies VALUES

(10, 1, 10),

(5, 1, 11),

(2, 2, 12),

(5, 2, 13),

(7, 3, 14),

(1, 5, 10),

(3, 4, 11);

INSERT INTO Card VALUES

(100),

(101),

(102),

(103),

(104);

INSERT INTO Book\_Lending VALUES

("2017-01-01","2017-06-01", 1, 10, 101),

("2017-01-11","2017-03-11", 3, 14, 101),

("2017-02-21","2017-04-21", 2, 13, 101),

("2017-03-17","2017-07-15", 4, 11, 101),

("2017-04-12","2017-05-12", 1, 11, 104);

Queries and screenshot:

**1.** Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each branch, etc.

**2.** Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017

**3.** Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.

**4.** Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.

**5.** Create a view of all books and its number of copies that are currently available in the Library.

**1.** select l.branch\_id,b.book\_id,b.title,p.name,ba.author\_name,bc.no\_of\_copies from Library\_Branch l, Book\_Copies bc,Publisher p,Book b,Book\_Authors ba

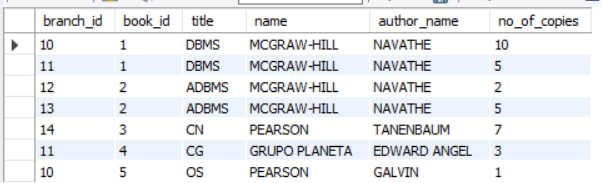
where

l.branch\_id=bc.branch\_id and

bc.book\_id=b.book\_id and

b.book\_id=ba.book\_id and

b.publisher\_name=p.name;

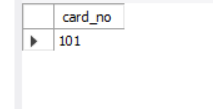


**2.** select card\_no from Book\_Lending

where date\_out between "2017-01-01" and "2017-06-30"

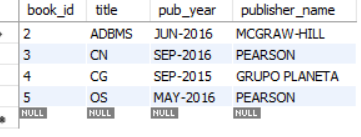
group by card\_no

having count(\*)>3;



**3.** delete from Book where book\_id=1;

Select \* from book;

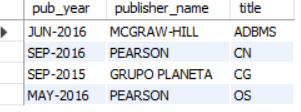


**4.**

create view Year\_Publishing as

select pub\_year,publisher\_name,title from Book;

select \* from Year\_Publishing;



**5.**

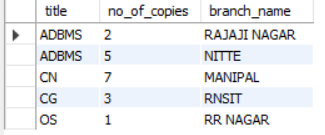
create view Book\_Copies\_Library as

select b.title,bc.no\_of\_copies,bl.branch\_name from Library\_Branch bl,Book\_Copies bc,Book b

where bl.branch\_id=bc.branch\_id and

bc.book\_id=b.book\_id;

select \* from Book\_Copies\_Library;



Lab-8

Queries and screenshot:

1. Produce a list of text books (include Course #, Book-ISBN, Booktitle) in the alphabetical order for courses offered by the 'CS' department that use more than two books
2. List any department that has all its adopted books published by a specific publisher.

**1** select b.courseno,b.book\_isbn,t.book\_title from book\_adoption b,text t

where

b.book\_isbn=t.book\_isbn and

b.courseno in (

select c.courseno from course c,book\_adoption b

where

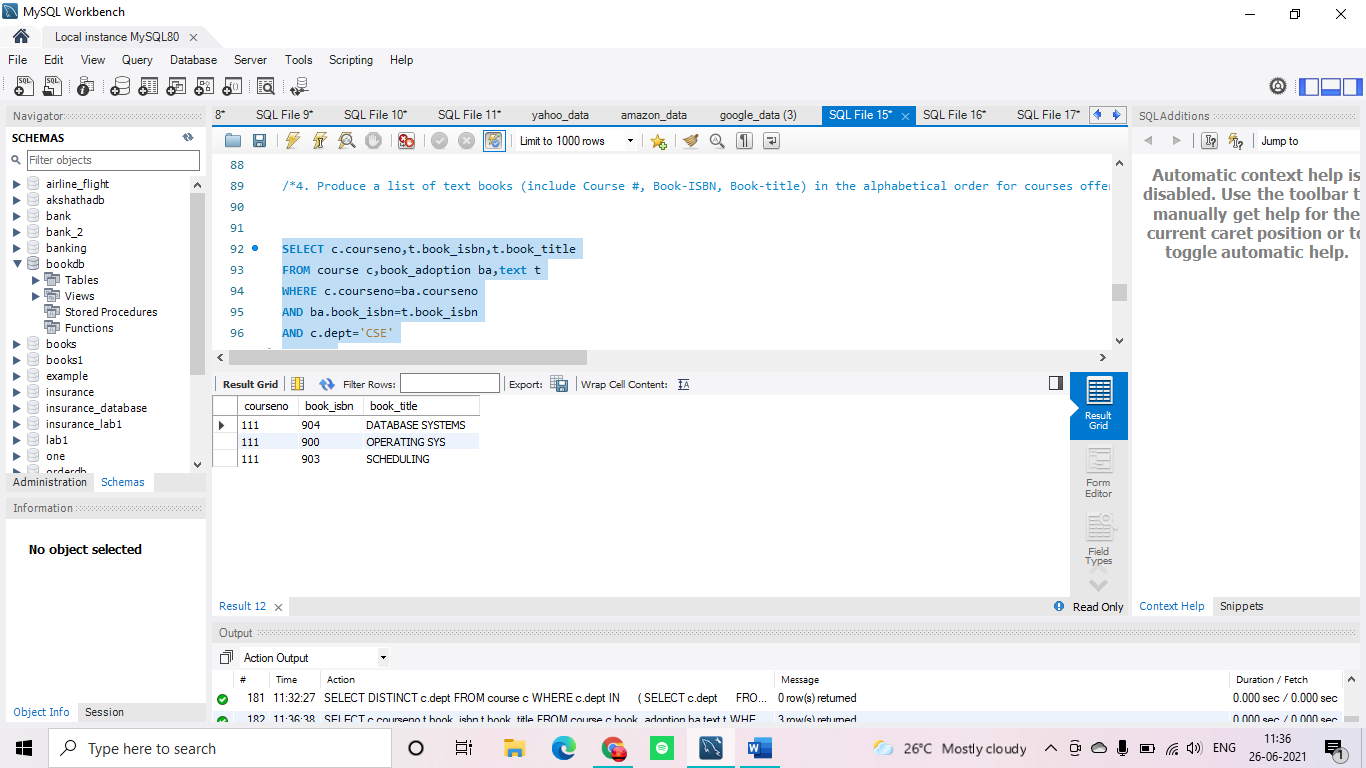
dept="CSE" and

c.courseno=b.courseno

group by b.courseno

having count(\*)>2

);



**2**  select c.dept from course c where c.dept in

(

select c1.dept from course c1,text t1,book\_adoption b1

where

b1.courseno=c1.courseno and

t1.book\_isbn=b1.book\_isbn and

t1.publisher='HALL INDIA' and

c1.dept not in (

select c2.dept from course c2,text t2,book\_adoption b2

where

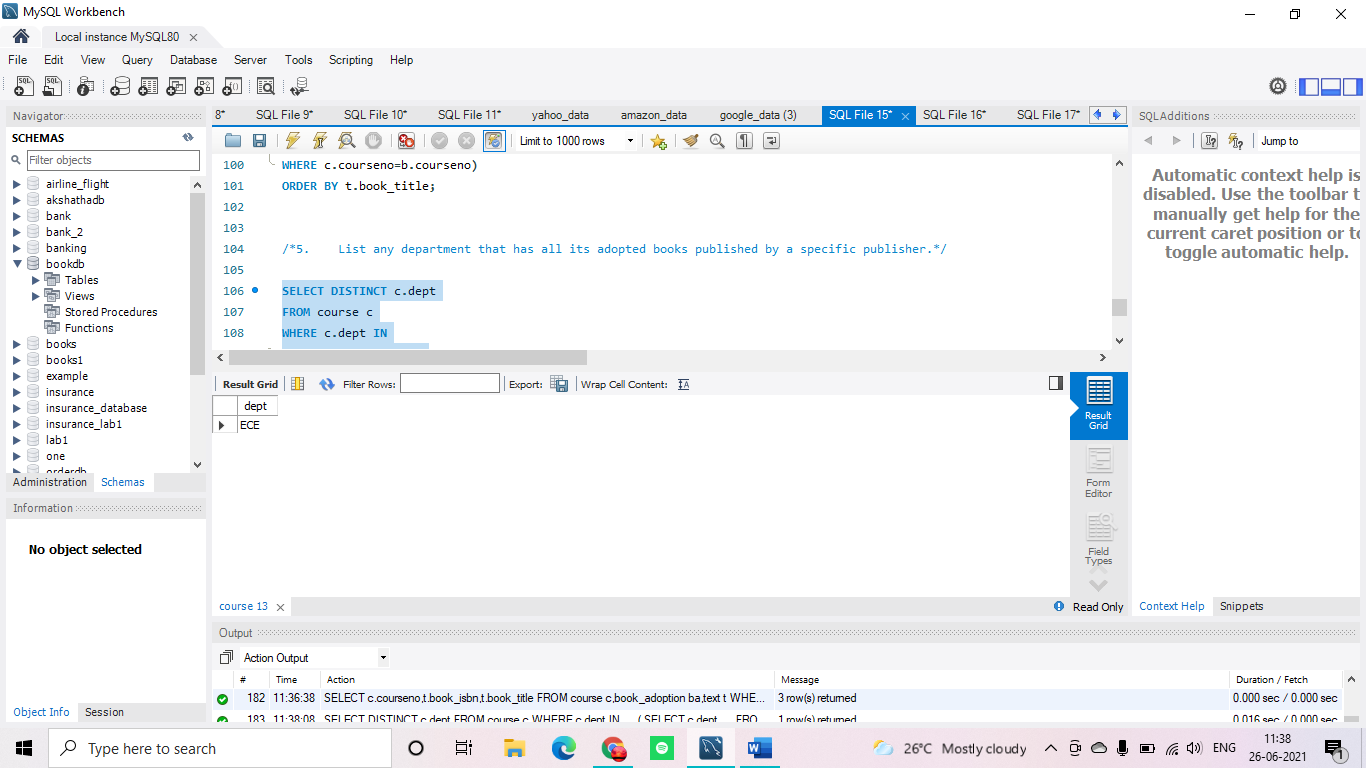
b2.courseno=c2.courseno and

t2.book\_isbn=b2.book\_isbn and

t2.publisher!='HALL INDIA'

)

);



Lab-9

Queries and screenshot:

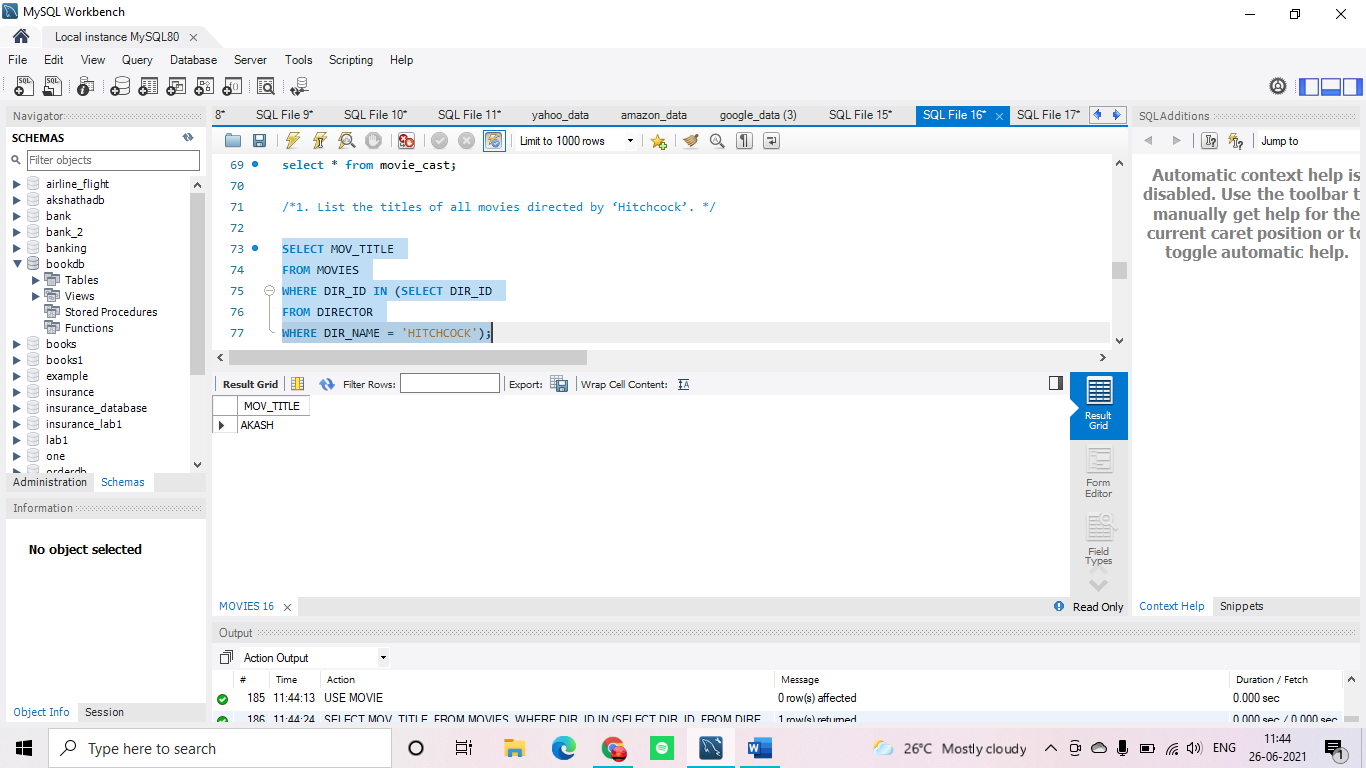
1. List the titles of all movies directed by ‘Hitchcock’.
2. Find the movie names where one or more actors acted in two or more movies.
3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).
4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.
5. Update rating of all movies directed by ‘Steven Spielberg’ to 5.

**1.** select m.MOV\_TITLE from MOVIES m,DIRECTOR d

where

d.DIR\_NAME="Hitchcock" and

d.DIR\_ID=M.DIR\_ID;



**2.** select m.MOV\_TITLE from MOVIES m,MOVIE\_CAST mc

where

mc.MOV\_ID=m.MOV\_ID and

mc.ACT\_ID in (

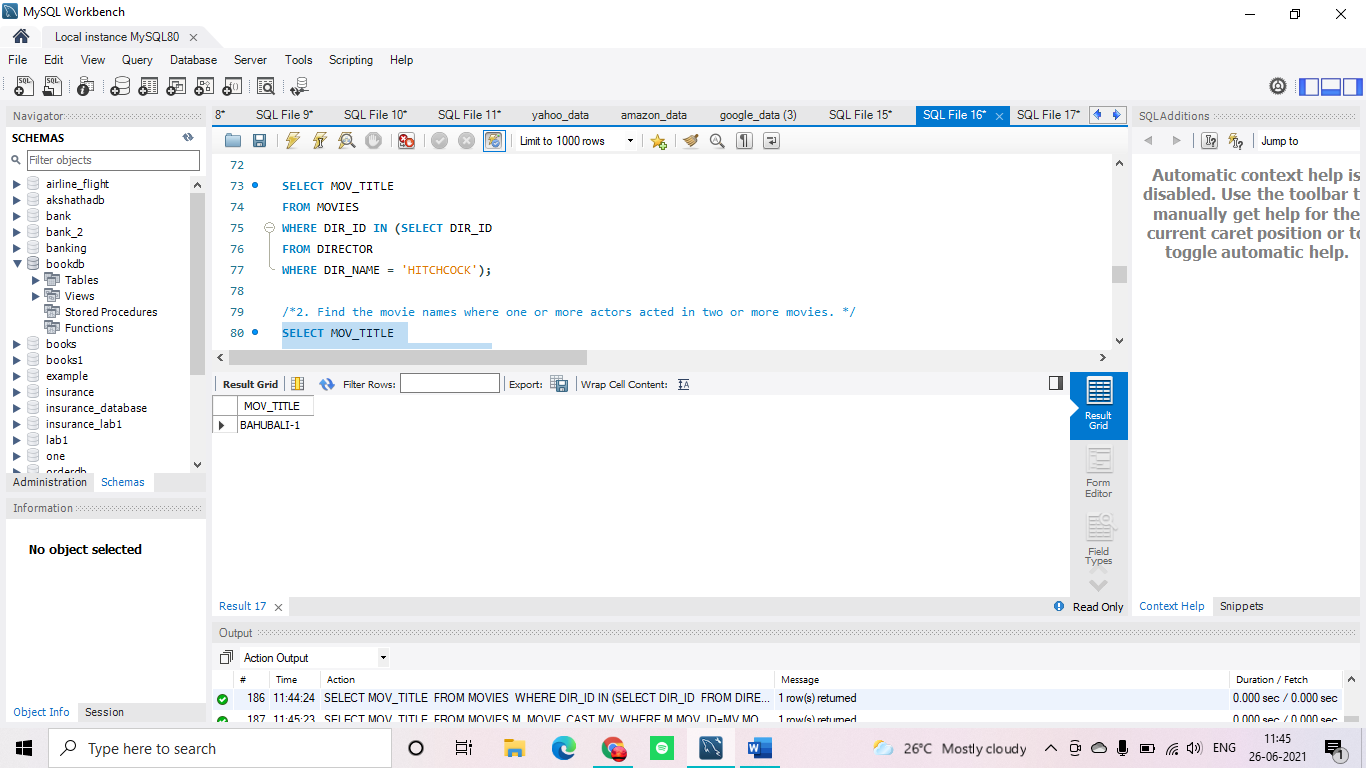
select mc1.ACT\_ID from MOVIE\_CAST mc1

group by mc1.ACT\_ID

having count(\*)>=2)

group by m.MOV\_TITLE

having count(\*)>=1;



**3.** select a.ACT\_NAME,m.MOV\_TITLE,m.MOV\_YEAR from ACTOR a

join MOVIE\_CAST mc

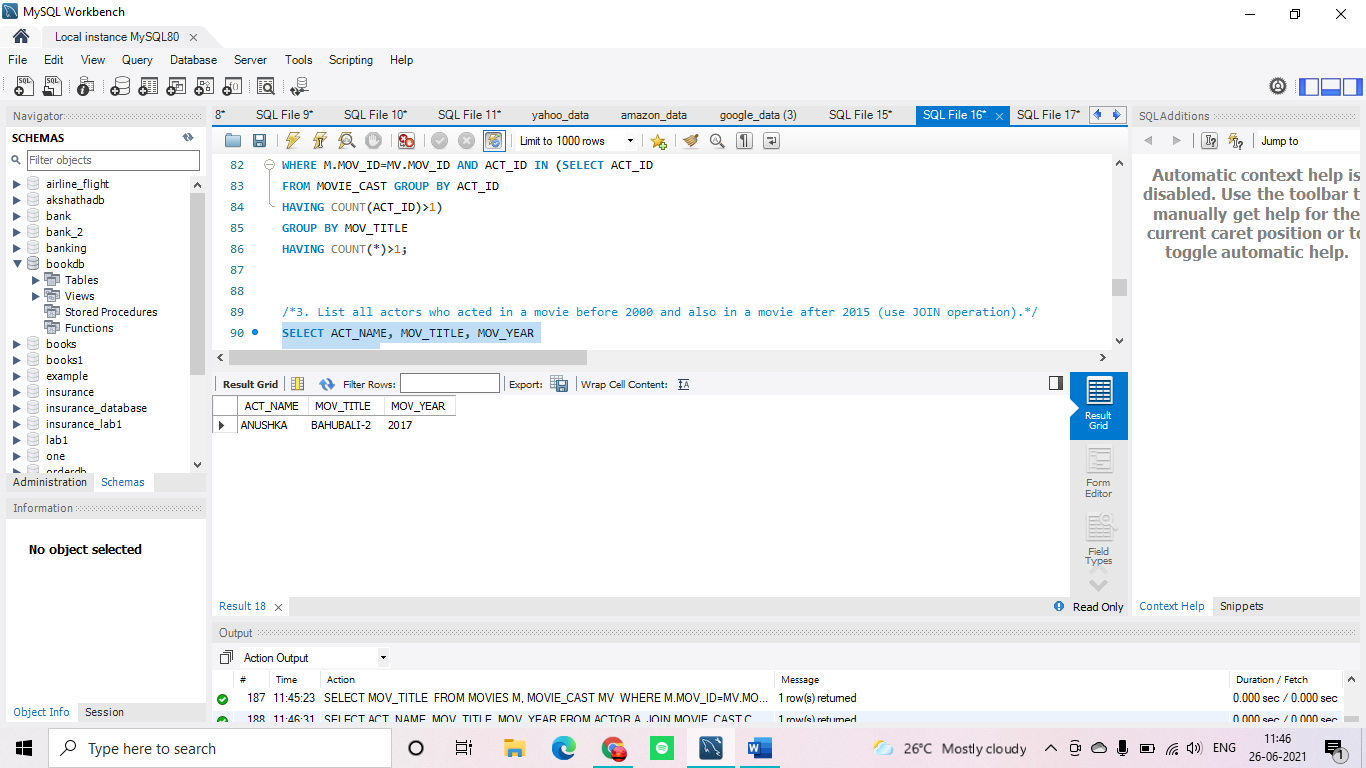
on mc.ACT\_ID =a.ACT\_ID

join MOVIES m

on m.MOV\_ID =mc.MOV\_ID

where

m.MOV\_YEAR not between 2000 and 2015



;

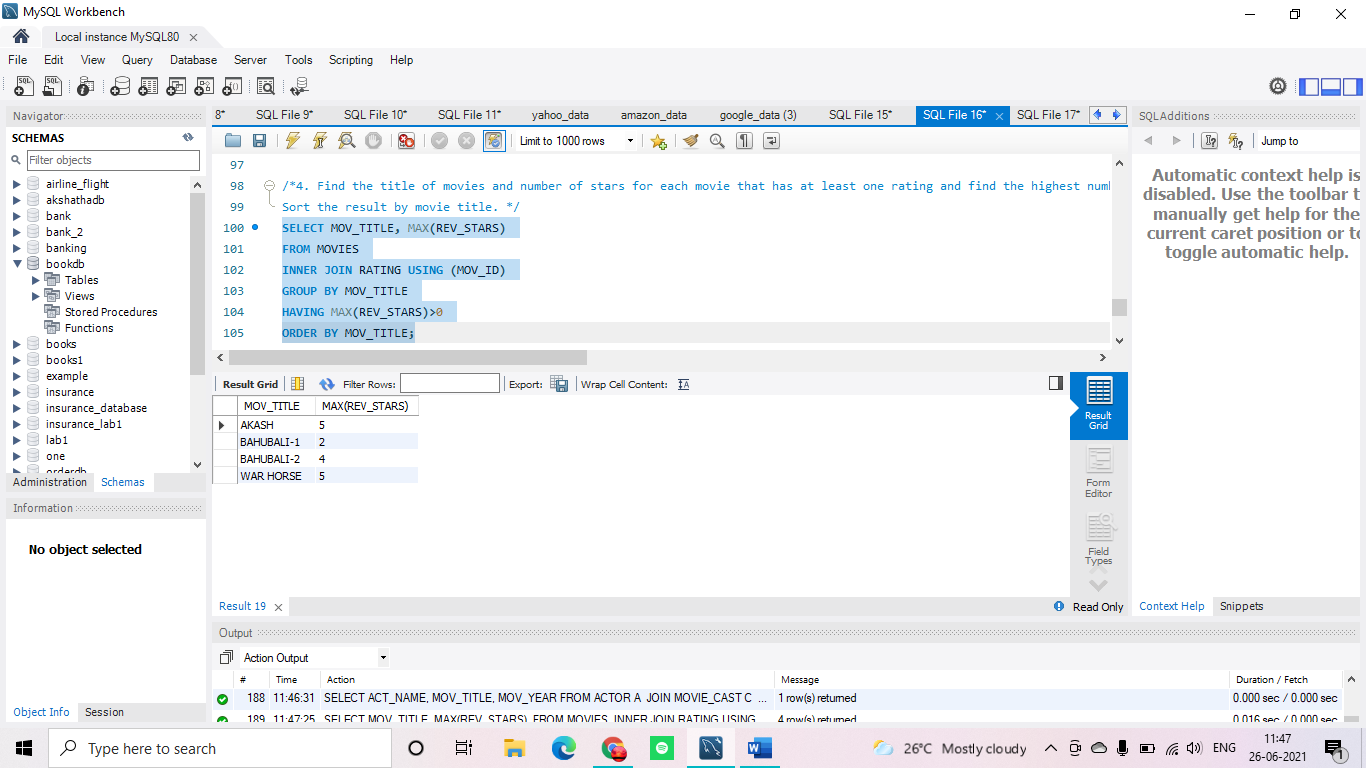
**4.** SELECT MOV\_TITLE , MAX(REV\_STARS) FROM MOVIES M,RATING R

WHERE M.MOV\_ID=R.MOV\_ID

group by M.MOV\_ID

having COUNT(\*)>=1

order by MOV\_TITLE;



**5.** UPDATE RATING SET REV\_STARS=5

WHERE MOV\_ID IN (

SELECT MOV\_ID FROM MOVIES

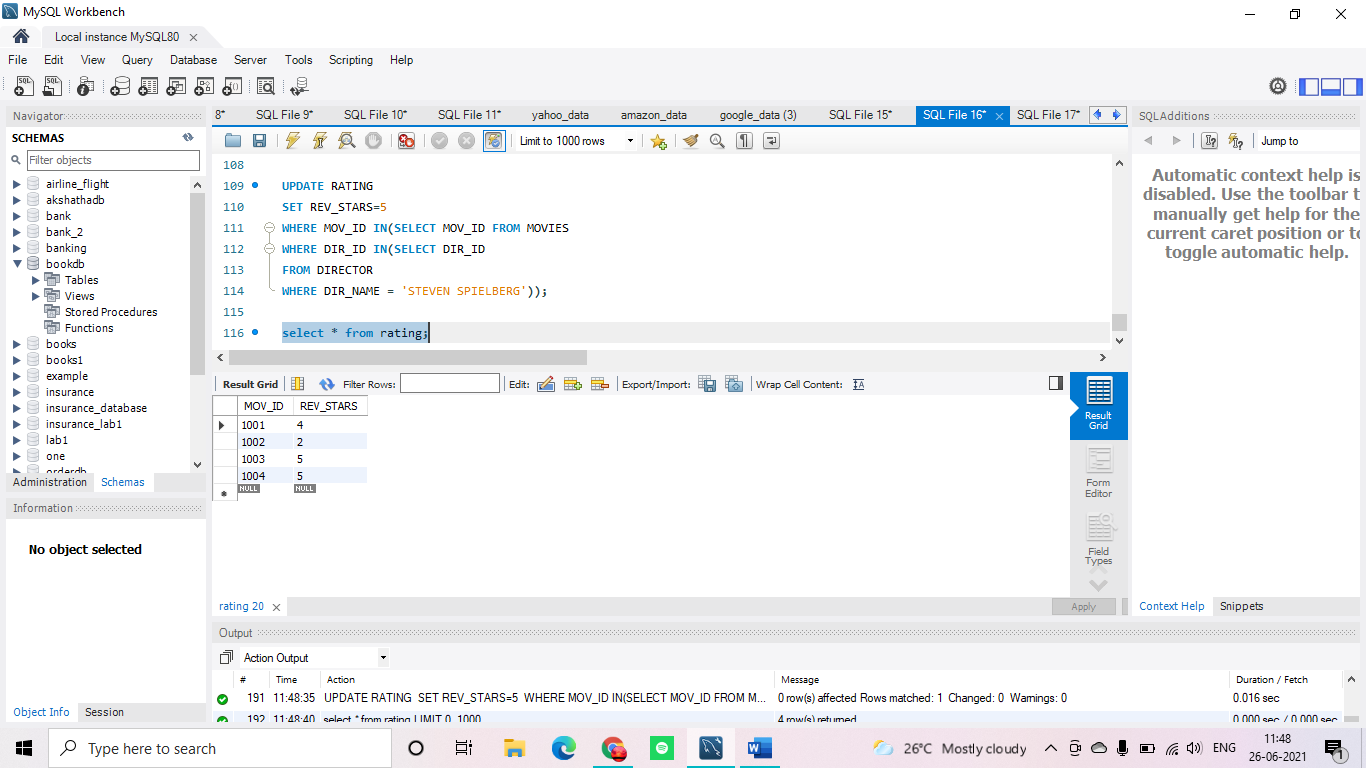
WHERE DIR\_ID =(

SELECT DIR\_ID FROM DIRECTOR

WHERE DIR\_NAME="STEVEN SPIELBERG"

)

);



Lab-10

Queries and screenshot:

**1**. List all the student details studying in fourth semester ‘C’ section.

**2**. Compute the total number of male and female students in each semester and in each section.

**3**. Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects.

**4**. Categorize students based on the following criterion:

If FinalIA = 17 to 20 then CAT = ‘Outstanding’ If FinalIA = 12 to 16 then CAT = ‘Average’ If FinalIA< 12 then CAT = ‘Weak’ Give these details only for 8th semester A, B, and C section students.

**1.** SELECT \* FROM STUDENT S,SEMSEC M,CLASS C

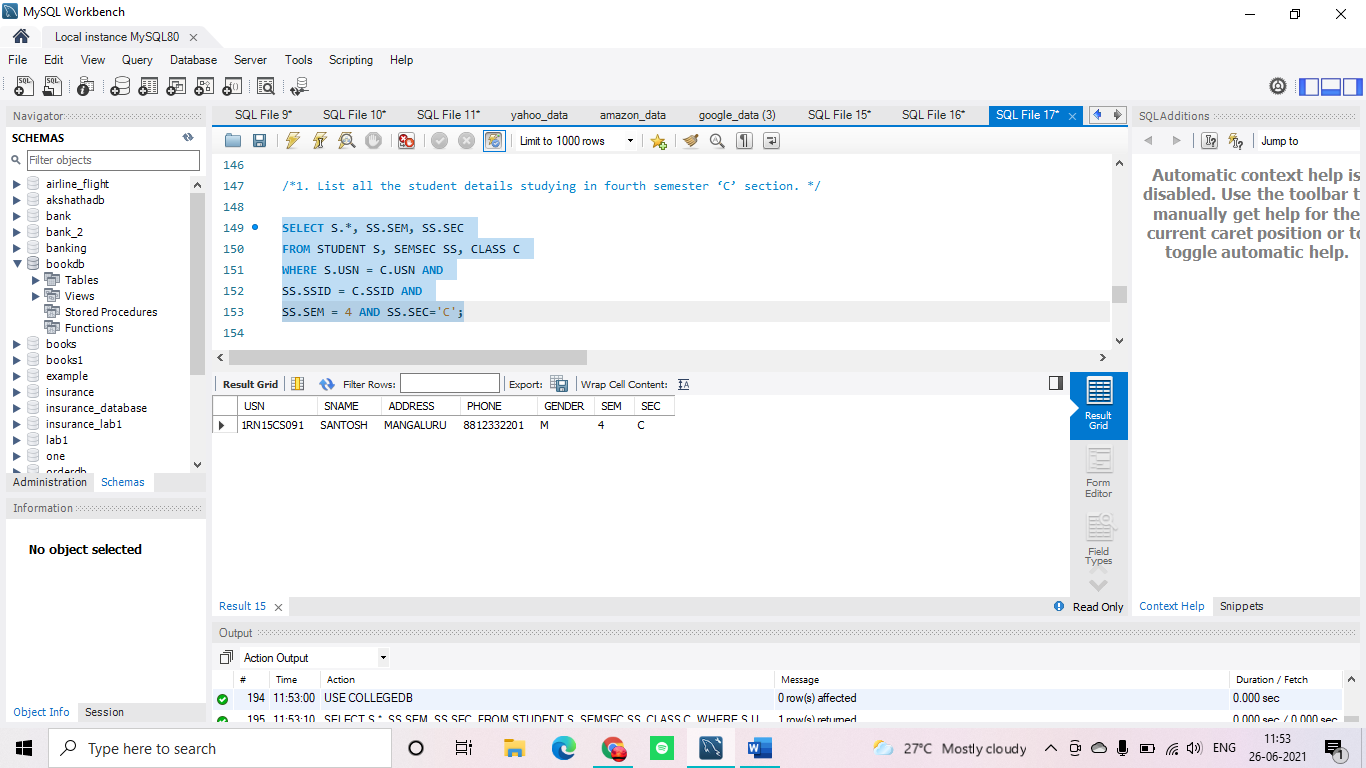
WHERE

S.USN=C.USN AND

C.SSID=M.SSID AND

M.SEM=4 AND

M.SEC="C"



;

**2.** SELECT M.SEC,M.SEM,S.GENDER,COUNT(\*) FROM STUDENT S,SEMSEC M,CLASS C

WHERE

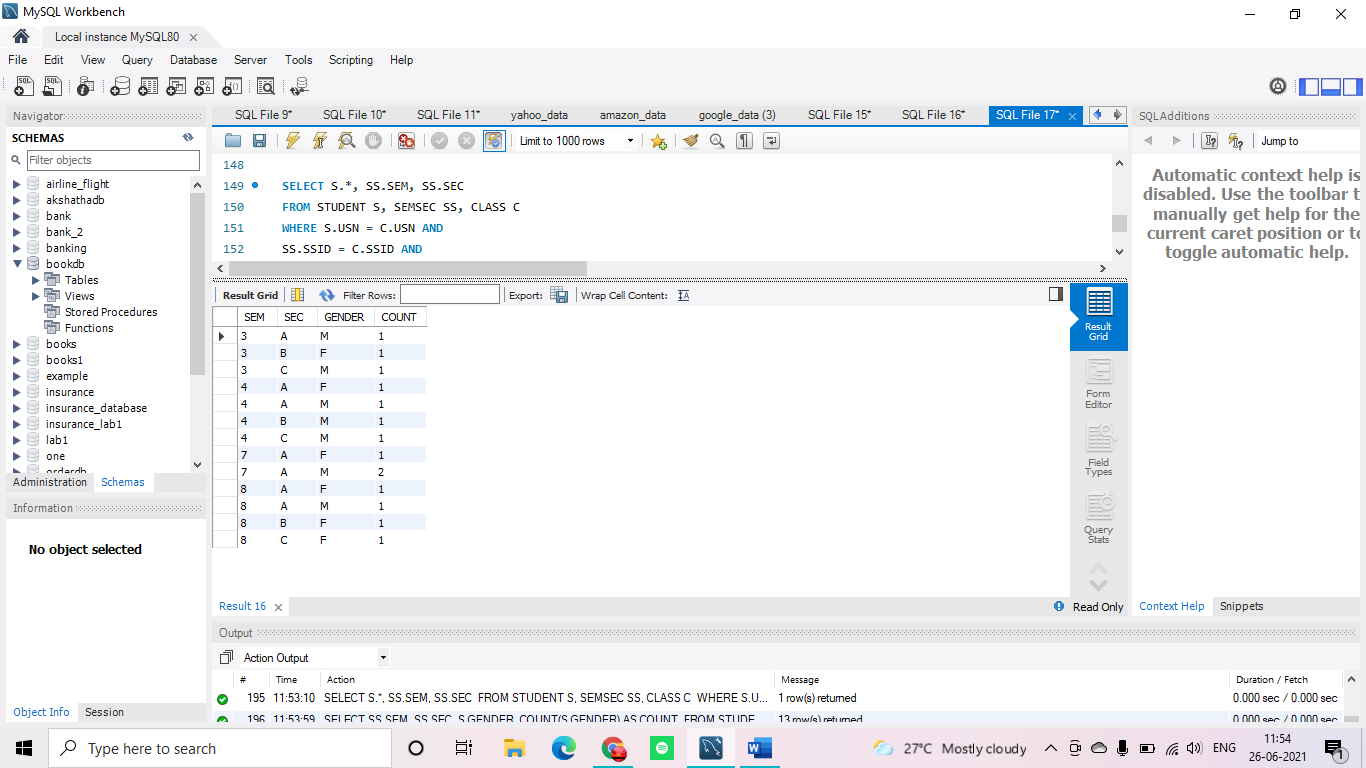
S.USN=C.USN AND

C.SSID=M.SSID

group by

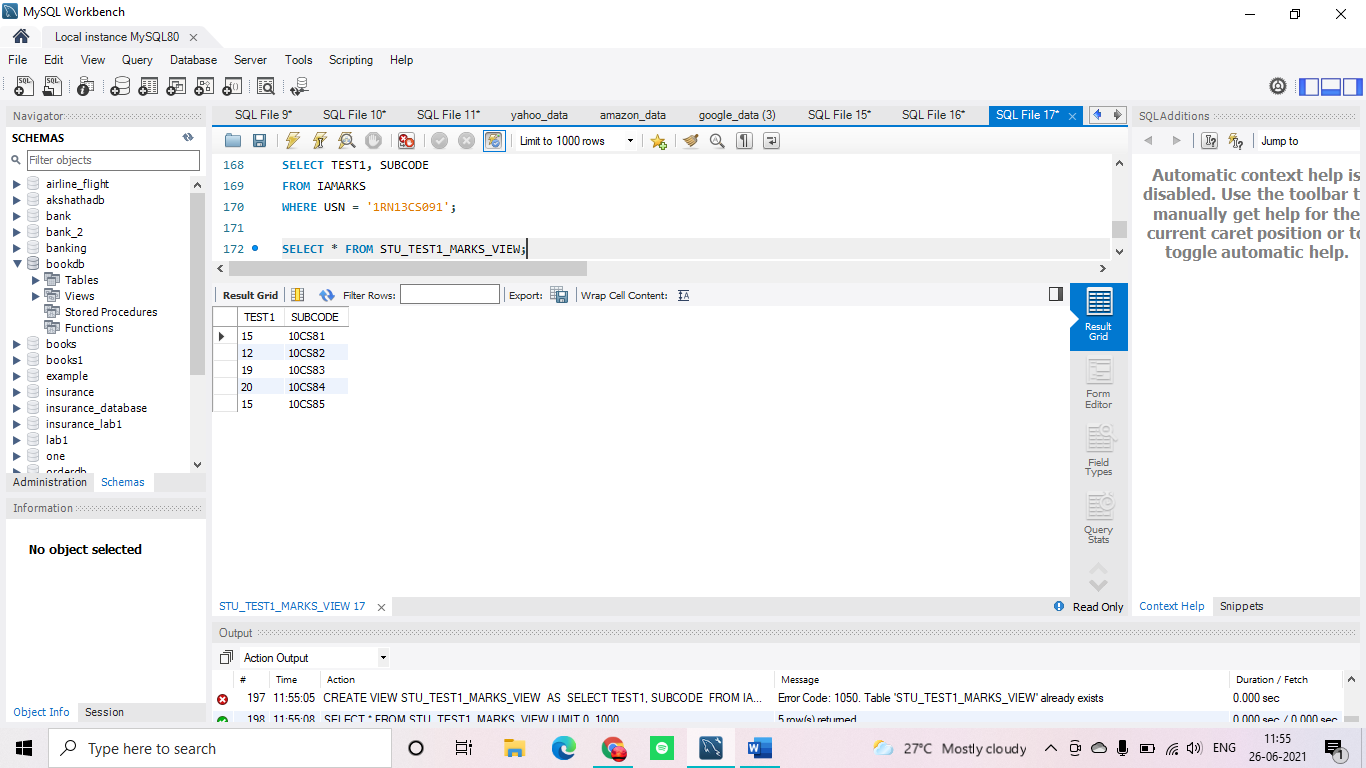
S.GENDER,M.SEM,M.SEC

ORDER BY M.SEM;



**3.** CREATE VIEW TEST1\_1RN13CS091 AS SELECT SUBCODE, TEST1 FROM IAMARKS WHERE USN="1RN13CS091";

SELECT \* FROM TEST1\_1RN13CS091;



**4.** UPDATE IAMARKS SET FINALIA=(TEST1+TEST2+TEST3)/3;

select \* from iamarks;

(SELECT I.USN,SUBCODE,FINALIA,"OUTSTANDING" FROM IAMARKS I,SEMSEC S WHERE S.SSID=I.SSID AND S.SEM=8 AND (S.SEC="A"OR S.SEC="B"OR S.SEC="C") AND FINALIA between 17 AND 20)

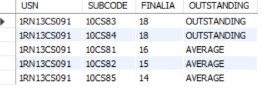
UNION

(SELECT I.USN,SUBCODE,FINALIA,"AVERAGE" FROM IAMARKS I,SEMSEC S WHERE S.SSID=I.SSID AND S.SEM=8 AND (S.SEC="A"OR S.SEC="B"OR S.SEC="C") AND FINALIA between 12 AND 17)

UNION

(SELECT I.USN,SUBCODE,FINALIA,"WEAK" FROM IAMARKS I,SEMSEC S WHERE S.SSID=I.SSID AND S.SEM=8 AND (S.SEC="A"OR S.SEC="B"OR S.SEC="C") AND FINALIA <12)

;



Lab-6

Order database

**Code:**

create database order\_dab;

use order\_dab;

create table Salesman

(salesman\_id int primary key not null,

name varchar(25),

city varchar(25),

commission real);

create table Customer

(customer\_id int primary key not null,

cust\_name varchar(25),

city varchar(25),

grade int,

salesman\_id int,

foreign key(salesman\_id) references Salesman(salesman\_id)on delete set null);

create table Orders(

order\_no int,

purchase\_amt real,

ord\_date date,

customer\_id int,

salesman\_id int,

foreign key(customer\_id) references Customer(customer\_id) on delete cascade,

foreign key(salesman\_id) references Salesman(salesman\_id) on delete cascade);

insert into salesman values

(1000,"JHON","BANGALORE",25),

(2000,"RAVI","BANGALORE",20),

(3000,"KUMAR","MYSORE",15),

(4000,"SMITH","DELHI",30),

(5000,"HARSHA","HYDRABAD",15);

insert into Customer values

(10,"PREETHI","BANGALORE",100,1000),

(11,"VIVEK","MANGALORE",300,1000),

(12,"BHASKAR","CHENNAI",400,2000),

(13,"CHETHAN","BANGALORE",200,2000),

(14,"MAMATHA","BANGALORE",400,3000);

insert into Orders values

(50,5000,"2017-05-04",10,1000),

(51,450,"2017-01-20",10,2000),

(52,1000,"2017-02-24",13,2000),

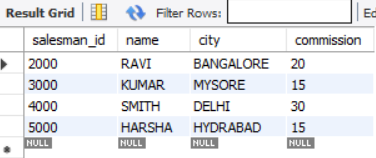
(53,3500,"2017-04-13",14,3000),

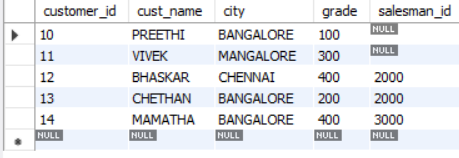
(54,550,"2017-03-09",12,2000);

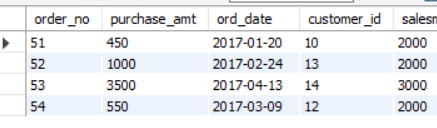
select \* from salesman;

select \*from customer;

select \* from orders;

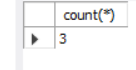






**1.** select count(\*) from Customer

where grade>(select avg(grade) from Customer where city="BANGALORE");



**2.**

select s.salesman\_id,s.name from Salesman s,Customer c

where

s.salesman\_id=c.salesman\_id

group by c.salesman\_id

having count(\*)>1;



**3**.

SELECT s.salesman\_id, name, cust\_name, commission

FROM Salesman s, Customer c

WHERE s.city = c.city

UNION

SELECT salesman\_id, name, 'NO MATCH', commission

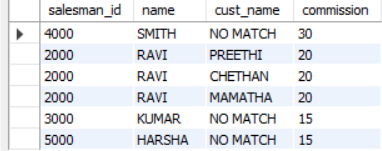
FROM Salesman

WHERE NOT city = ANY

(SELECT city

FROM Customer)

ORDER BY 2 DESC;



**4.**

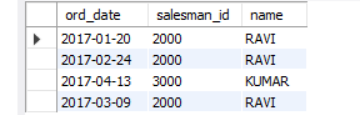
create view HighestOrder as

select o.ord\_date,s.salesman\_id,s.name from Salesman s,Orders o

where s.salesman\_id=o.salesman\_id and

o.purchase\_amt= (select max(purchase\_amt) from Orders where ord\_date=o.ord\_date);

select \* from HighestOrder;



**5.**

delete from Salesman where salesman\_id=1000;

select \* from Salesman;

select \* from Orders;

select \* from Customer;

