DBMS LAB RECORD TEST-2

USN- 1BM19CS079
NAME- LIKITHA B

PROGRAM 1: INSURANCE DATABASE

QUESTION:

Consider the Insurance database given below. The primary keys are underlined

and the data types are specified.

PERSON (driver-id #: String, name: String, address:

String) CAR (Regno: String, model: String, year: int)

ACCIDENT (report-number: int, adate: date, location:

String) OWNS (driver-id #: String, Regno: String)

PARTICIPATED (driver-id: String, Regno: String, report-number: int, damage-amount: int)

i. Create the above tables by properly specifying the primary keys and the foreign keys.

ii. Enter at least five tuples for each relation.

iii. Demonstrate how you

a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to

25000.

b. Add a new accident to the database.

iv. Find the total number of people who owned cars that involved in accidents in 2008.

v. Find the number of accidents in which cars belonging to a specific model were involved.

PROGRAM CODE:

```
create database

Insurance; use

Insurance;

CREATE TABLE PERSON(DRIVER_ID VARCHAR(10),NAME
VARCHAR(20),ADDRESS VARCHAR(15),PRIMARY KEY(DRIVER_ID));

show tables;

desc

PERSON;

SELECT *FROM PERSON;
```

```
create table car(regno varchar(10),Model varchar(20),Year date,Primary key(Regno));

create table Accident(report_no int,ADATE DATE,Location varchar(15),Primary

key(report_no)); create table owns(driver_id varchar(10),regno varchar(10),primary

key(driver_id,regno),

 foreign key(driver_id) references person(driver_id) on delete cascade, foreign key(regno) references car(regno) on delete cascade);

 CREATE TABLE PARTICIPATED(driver_id varchar(10),regno varchar(10),report_no int, damage_amt float,

foreign key (driver_id,regno) references OWNS(driver_id,regno) ON DELETE

CASCADE, foreign key (REPORT_NO) references ACCIDENT(REPORT_NO)

ON DELETE CASCADE); show tables;

insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('1111','RAMU',

'K.S.LAYOUT'); insert into

PERSON(DRIVER_ID,NAME,ADDRESS)values('2222','JOHN', 'INDIRANAGAR');

insert into

PERSON(DRIVER_ID,NAME,ADDRESS)values('3333','PRIYA','JAYANAGAR');

insert into

PERSON(DRIVER_ID,NAME,ADDRESS)values('4444','GOPAL','WHITEFIELD');

insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('5555','LATHA',' 

VIJAYANAGAR'); COMMIT;

desc PERSON;

SELECT *FROM PERSON;

insert into car(regno,Model,Year)values('KA04Q2301','MARUTHI-DX','2000-

10-11'); insert into car(regno,Model,Year)values('KA05P1000',' 

FORDICON','2000-09-08'); insert into

car(regno,Model,Year)values('KA03L1234','ZEN-VXI', '1999-07-06'); insert

into car(regno,Model,Year)values('KA03L9999',' MARUTH-DX', '2002-06-

05'); insert into car(regno,Model,Year)values('KA01P4020',' INDICA-VX',

'2002-05-04'); COMMIT;

desc car;

SELECT *FROM car;

insert into Accident(report_no,ADATE,Location)values('12',' 2002-06-02',' M G

ROAD'); insert into Accident(report_no,ADATE,Location)values('200',' 2002-12-10',' 

DOUBLEROAD'); insert into Accident(report_no,ADATE,Location)values('300',' 

1999-07-10','M G ROAD');

insert into Accident(report_no,ADATE,Location)values('25000',' 2000-06-11',' 
```

RESIDENCY ROAD'); insert into Accident(report_no,ADATE,Location)values('26500',' 2001-08-12',' RICHMOND ROAD'); COMMIT;

```
desc Accident;

SELECT *FROM Accident;

insert into owns(driver_id,regno)values('1111',

'KA04Q2301'); insert into

owns(driver_id,regno)values('1111','KA05P1000'); insert

into owns(driver_id,regno)values('2222','KA03L1234');

insert into

owns(driver_id,regno)values('3333','KA03L9999'); insert

into owns(driver_id,regno)values('4444','KA01P4020');

COMMIT;

desc owns;

SELECT *FROM owns;

insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111',
'KA04Q2301',' 12',' 20000');

insert into
PARTICIPATED(driver_id,regno,report_no,damage_amt)values('2222','KA03L1234','200','
500');

insert into
PARTICIPATED(driver_id,regno,report_no,damage_amt)values('3333','KA03L9999','300','
10000');

insert into
PARTICIPATED(driver_id,regno,report_no,damage_amt)values('4444','KA01P4020','25000
','2375');

insert into
PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111','KA05P1000','26500','
70000');

COMMIT;

desc PARTICIPATED ;

SELECT *FROM PARTICIPATED;

/*

a. Update the damage amount for the car with a specific Regno in the accident with report
number 12 to

25000.

*/

UPDATE PARTICIPATED SET DAMAGE_AMT=25000 WHERE REPORT_NO
=12 AND REGNO='KA04Q2301';

COMMIT;

desc PARTICIPATED ;

SELECT *FROM PARTICIPATED;
```

/*

b. Add a new accident to the database

*/

insert into Accident(report_no,ADATE,Location)values('500',' 2005-06-02','Mysore Road'); desc Accident;

SELECT *FROM Accident;

/*

iv. Find the total number of people who owned cars that involved in accidents in 2008

*/

select count(*) from Accident where year(ADATE)=2008;

/*

V. Find the number of accidents in which cars belonging to a specific model were involved

*/

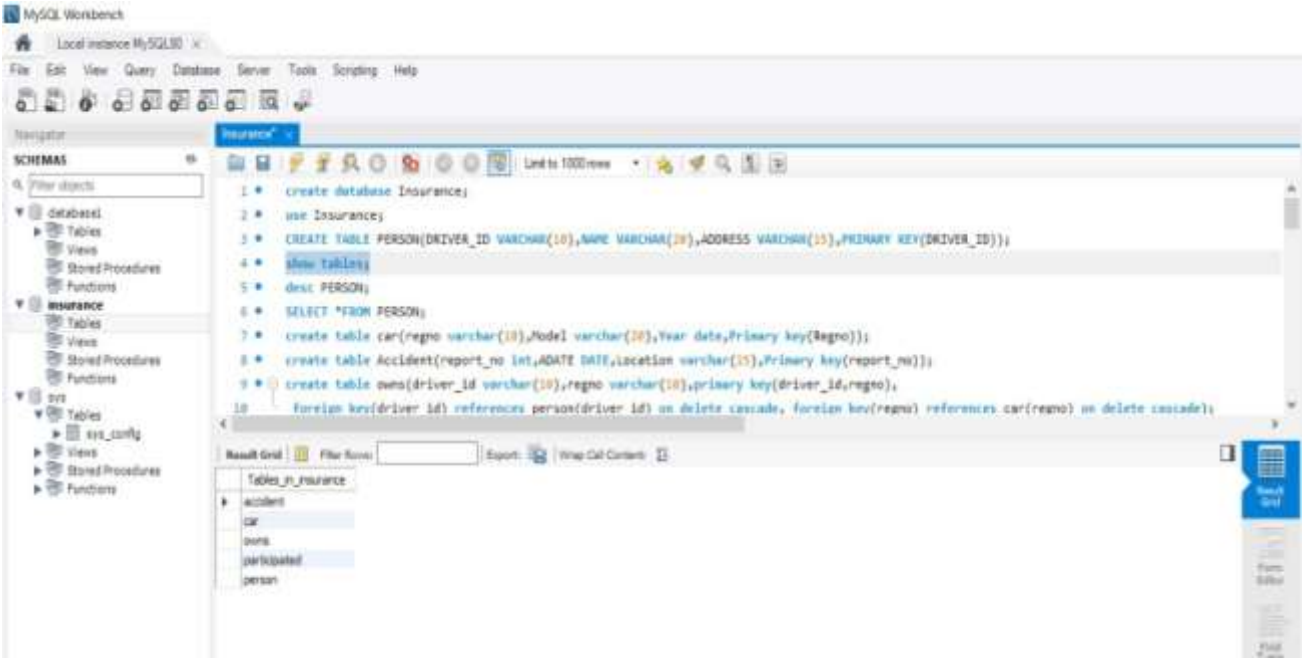
SELECT COUNT(A.REPORT_NO) FROM ACCIDENT A,

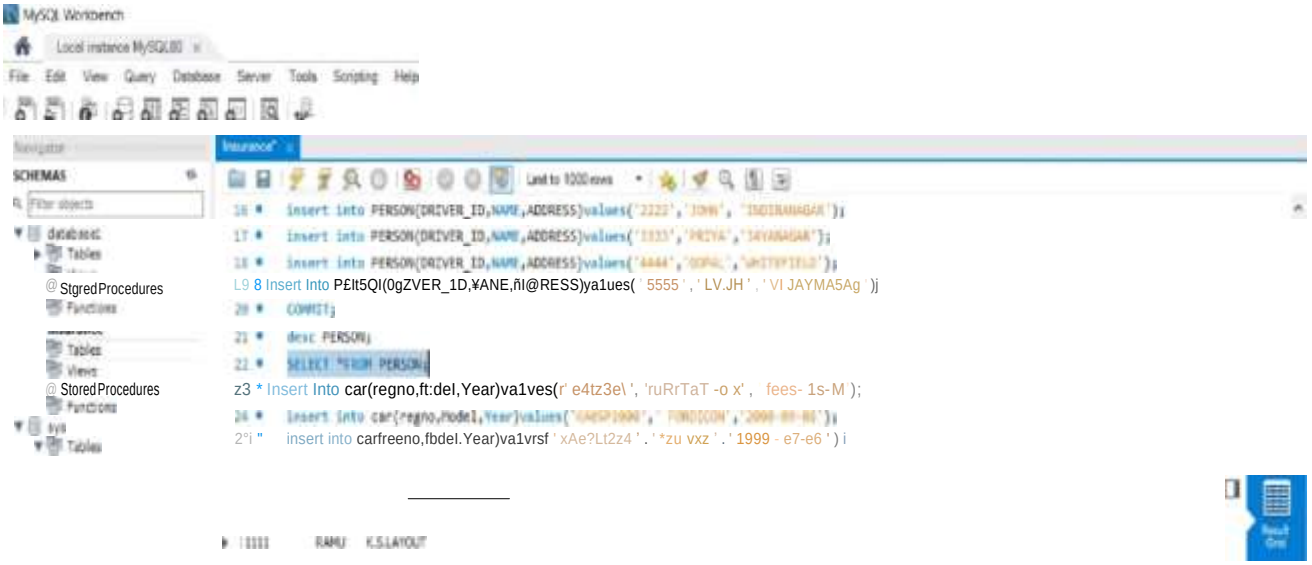
PARTICIPATED P, CAR C WHERE

A.REPORT_NO=P.REPORT_NO AND

P.REGNO=C.REGNO AND C.MODEL='MARUTHI-DX';

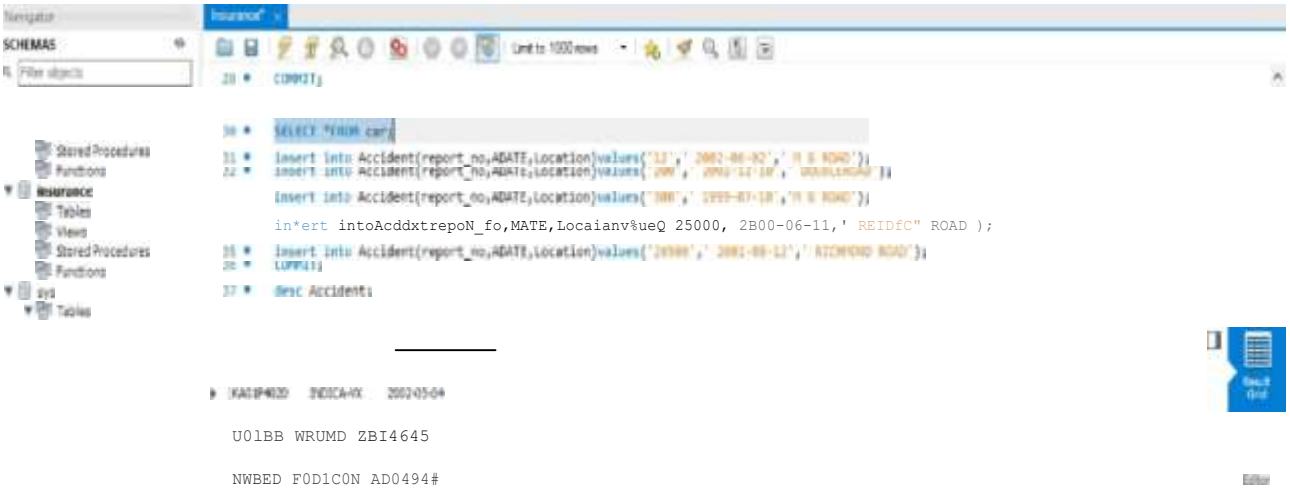
SCREENSHOTS OF OUTPUT:





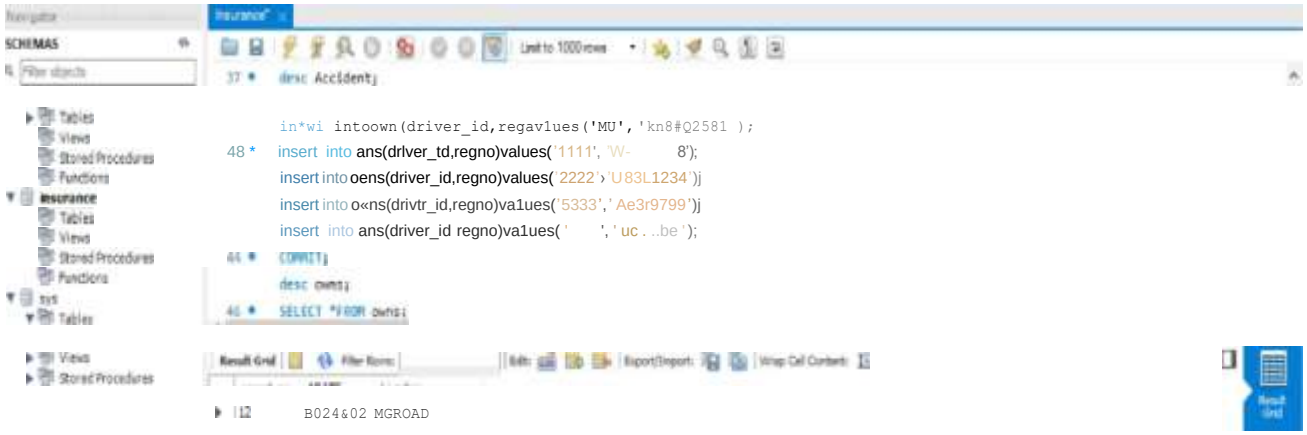
My& Workbench

File Edt Yes Quey Oatabase Seve Todd Sold /g Hdp



MySQL Workbench

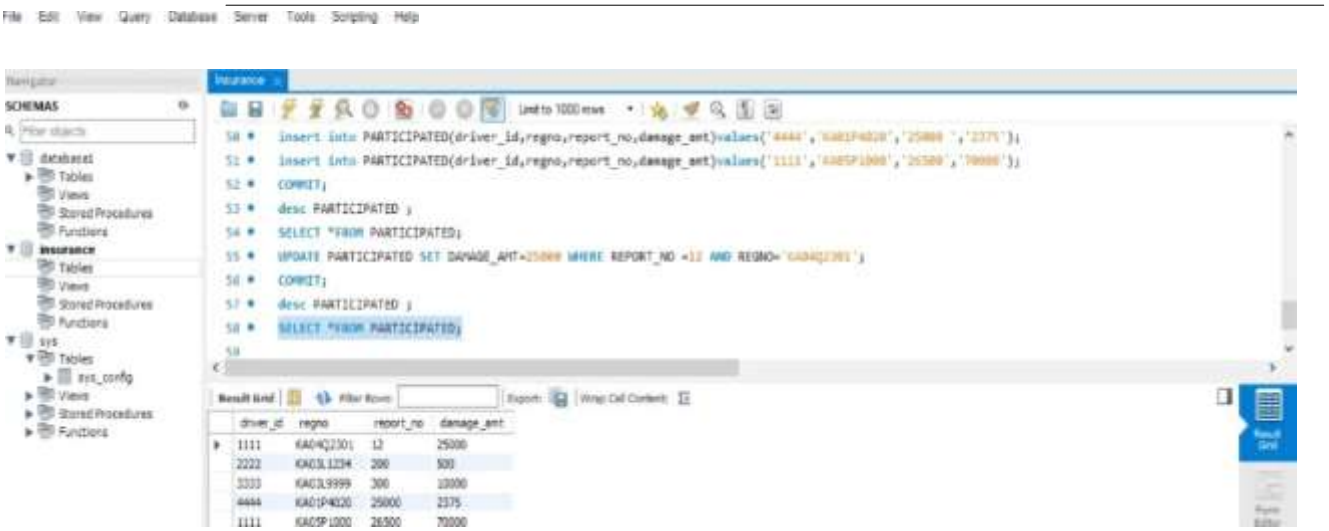
tie Edt Vle« gury Oatahase Sava Tale Sending Hdp

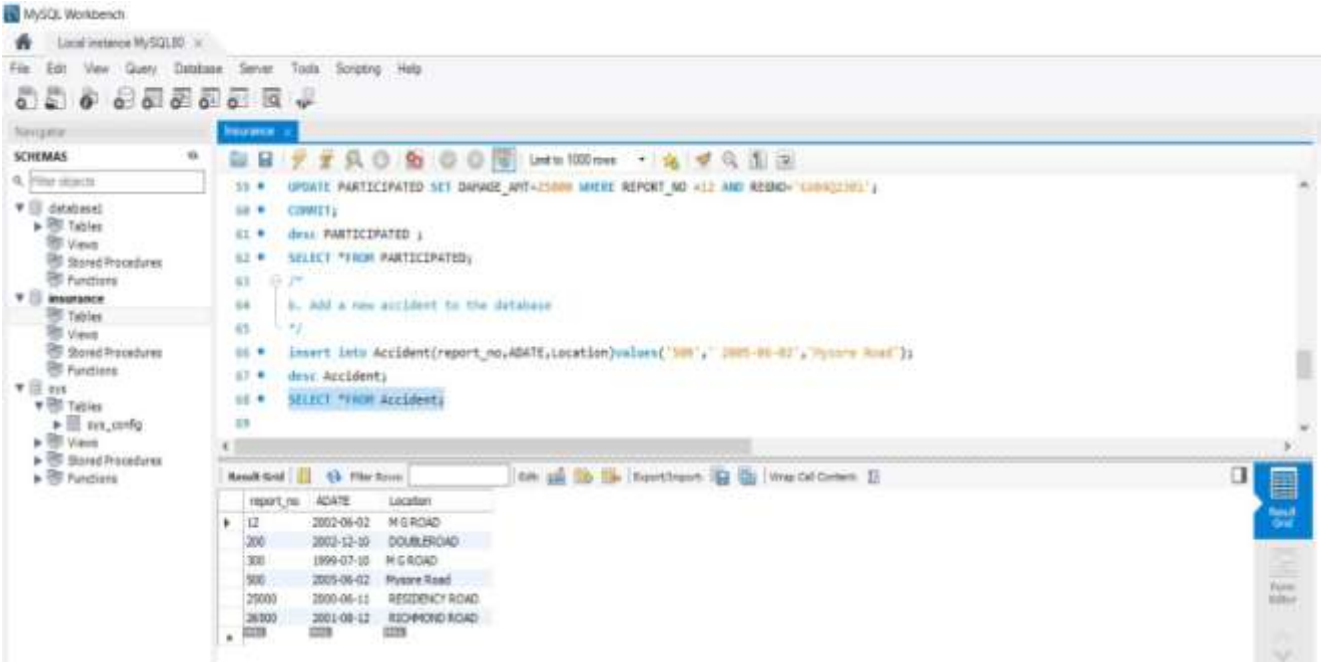


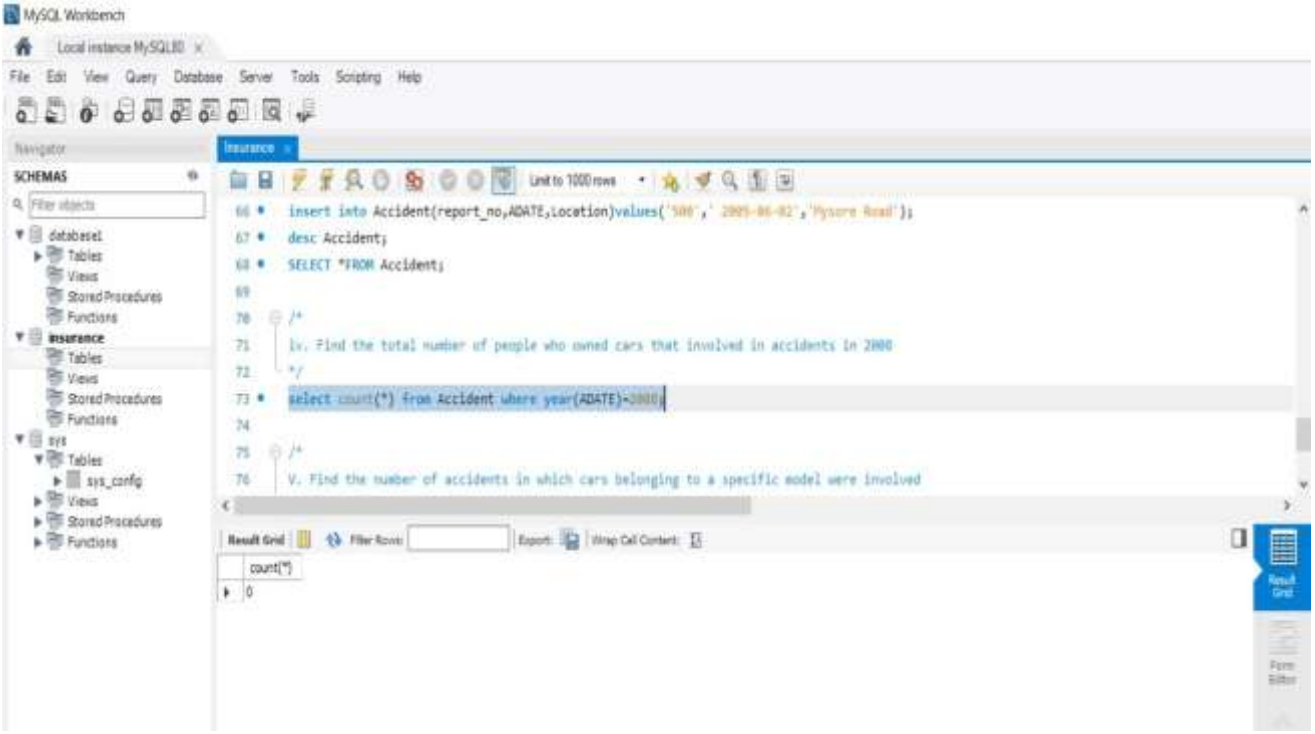
12	B024&02 MGROAD
Z00	I99S47-IO NG RCU0

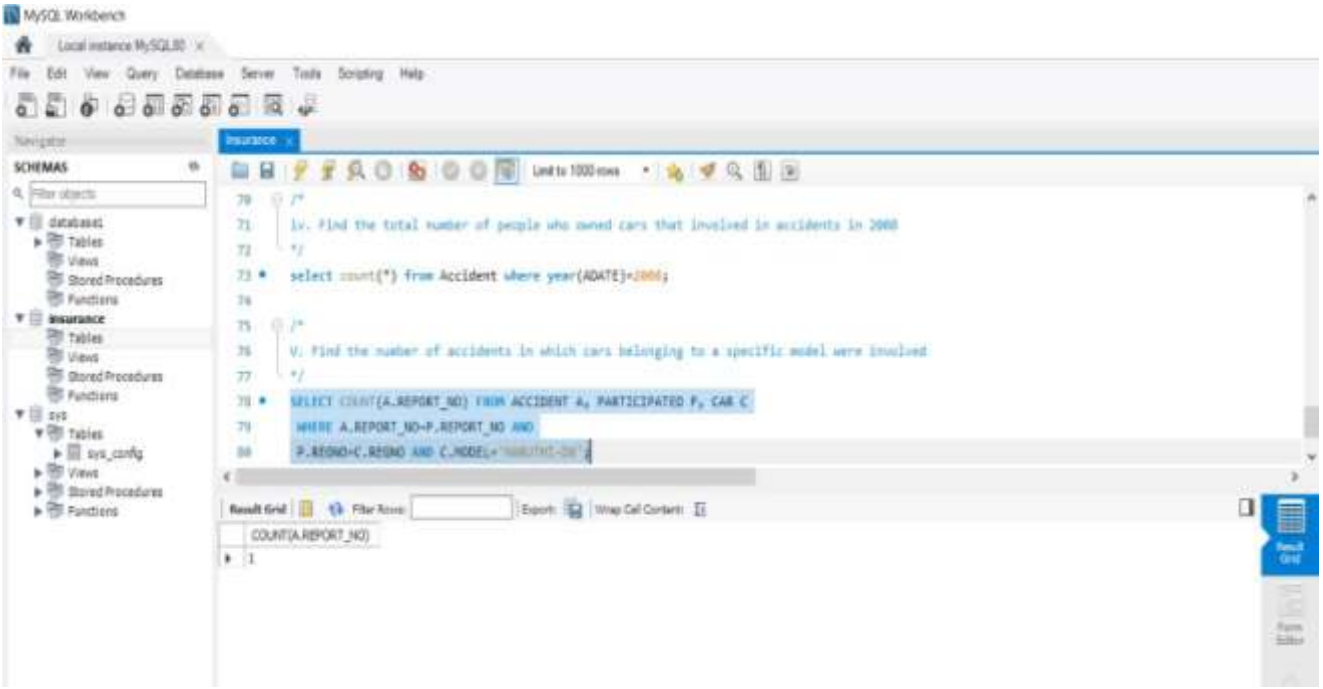
t1t1	2g01
1111	*MOVGPIO*


File Edit View Query Database Server Tools Scripting Help
Limit to 1000 rows
t7 * insert iato PARTICIPATED(driver_id,regno,report_no,deage_W)va1ues(`UIF , 'KA84q230L',' 12',` 200a ')j 48
° Inimrt Ents PARMCIPATED(driver_ld,mgno,repoct_no,damagc_mt)vn1ur*('2222', KAB3t1234', 28B',' 848]; 49
50 • insert Into PARTICIPATED(driver_id,regno,report_no,damage_ant)values('4444','KA01P4020','25000 ','2375');
•••n i•t• Putz c\raTEo(4rt ver_ld, regno, reFert_no, aanagr_+nt)v•t••s‹ `azzs', `KAe3t9s9 s', `sea', ' geese' jJ
51 o insert inte PARDCIPA1f0(driver_id, regna, regrt_r+n,deriage_a•t)vzlues(` HH , ` G5Plm ' ' 26SM ' ' 7M ');
53 • desc PARTICIPATED ;
54 • SELECT *FROM PARTICIPATED;
SS
Stored Procedures
Functions
driver_id regno report_no damage_ant
1111 KA04Q2301 12 20000
2222 KA03L1234 200 500
3333 KA03L9999 300 10000
4444 KA01P4020 25000 2375
1111 KA05P1000 26500 70000


File Edit View Query Database Server Tools Scripting Help
SCHEMAS
Insurance
Limit to 1000 rows
50 • insert into PARTICIPATED(driver_id,regno,report_no,damage_ant)values('4444','KA01P4020','25000 ','2375');
51 • insert into PARTICIPATED(driver_id,regno,report_no,damage_ant)values('1111','KA05P1000','26500','70000');
52 • COMMIT;
53 • desc PARTICIPATED ;
54 • SELECT *FROM PARTICIPATED;
55 • UPDATE PARTICIPATED SET DAMAGE_AMT=25000 WHERE REPORT_NO =12 AND REGNO='KA04Q2301';
56 • COMMIT;
57 • desc PARTICIPATED ;
58 • SELECT *FROM PARTICIPATED;
Result Grid
driver_id regno report_no damage_ant
1111 KA04Q2301 12 25000
2222 KA03L1234 200 500
3333 KA03L9999 300 10000
4444 KA01P4020 25000 2375
1111 KA05P1000 26500 70000


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
database1
Tables
Views
Stored Procedures
Functions
insurance
Tables
Views
Stored Procedures
Functions
sys
Tables
sys_config
Views
Stored Procedures
Functions
Insurance
Limit to 1000 rows
53 • UPDATE PARTICIPATED SET DAMAGE_AMT=25000 WHERE REPORT_NO =12 AND REGNO='KA09Q2301';
60 • COMMIT;
61 • desc PARTICIPATED ;
62 • SELECT *FROM PARTICIPATED;
63 /*
64 b. Add a new accident to the database
65 */
66 • insert into Accident(report_no,ADATE,Location)values('500',' 2005-06-02','Mysore Road');
67 • desc Accident;
68 • SELECT *FROM Accident;
69
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
report_no ADATE Location
12 2002-06-02 M G ROAD
200 2002-12-10 DOUBLEROAD
300 1999-07-10 M G ROAD
500 2005-06-02 Mysore Road
25000 2000-06-11 RESIDENCY ROAD
26300 2001-08-12 RICHMOND ROAD


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
database1
Tables
Views
Stored Procedures
Functions
insurance
Tables
Views
Stored Procedures
Functions
sys
Tables
sys_config
Views
Stored Procedures
Functions
Insurance
Limit to 1000 rows
66 • insert into Accident(report_no,ADATE,Location)values('500',' 2005-06-02','Mysore Road');
67 • desc Accident;
68 • SELECT *FROM Accident;
69
70 /*
71 iv. Find the total number of people who owned cars that involved in accidents in 2008
72 */
73 • select count(*) from Accident where year(ADATE)=2008;
74
75 /*
76 V. Find the number of accidents in which cars belonging to a specific model were involved
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
count(*)
0



PROGRAM-2:BOOK DEALER DATABASE

QUESTIO
N:

The following tables are maintained by a book dealer:

AUTHOR(author-id: int, name: String, city: String, country:

String) PUBLISHER(publisher-id: int, name: String, city: String,

country: String)

CATALOG(book-id: int, title: String, author-id: int, publisher-id: int, category-id: int, year: int, price: int)

CATEGORY(category-id: int, description: String)

ORDER-DETAILS(order-no: int, book-id: int, quantity: int)

i)Create the above tables by properly specifying the primary keys and the foreign keys.

ii) Enter at least five tuples for each relation.

iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books in the

catalog and the year of publication is after 2000.

iv) Find the author of the book which has maximum sales.

v) Demonstrate how you increase the price of books published by a specific publisher by 10%.

PROGRAM CODE:

```
create database
bookdealer; use
bookdealer;
create table
AUTHOR (
author_id int,
name varchar(20),
city varchar(15),
country
varchar(15),
primary key(author_id)
);
show tables;
desc
AUTHOR;
SELECT *FROM AUTHOR;
create table
PUBLISHER (
publisher_id int,
name varchar(20),
city varchar(15),
country
varchar(15),
primary key(publisher_id)
);
create table
CATEGORY(
category_id int,
description
varchar(20), primary
```

```
key(category_id)
);
show tables;
desc
CATEGORY;
```

```
SELECT *FROM CATEGORY;
create table
CATALOG ( book_id
int,
title varchar(15),author_id int,publisher_id int,category_id int,
foreign key(author_id) references AUTHOR(author_id) on delete cascade,
foreign key(publisher_id) references PUBLISHER(publisher_id) on delete
cascade, foreign key(category_id) references CATEGORY(category_id)
on delete cascade, year int,
price int,
primary key(book_id)
);
show tables;
desc
CATALOG;
SELECT *FROM CATALOG;
create table
ORDER_DETAILS (
order_no int,book_id int,
foreign key(book_id) references CATALOG(book_id) on delete
cascade, quantity int
);
show tables;
desc ORDER_DETAILS;
SELECT *FROM ORDER_DETAILS;
insert into AUTHOR(author_id,name,city,country)values(1001,'TERAS CHAN','CA','USA');
insert into
AUTHOR(author_id,name,city,country)values(1002,'STEVENS','ZOMBI','UG
ANDA');
insert into AUTHOR(author_id,name,city,country)values(1003,'M MANO','CAIR','CANADA');
insert into AUTHOR(author_id,name,city,country)values(1004,'KARTHIK B.P','NEW
YORK','USA');
insert into AUTHOR(author_id,name,city,country)values(1005,'WILLIAM
STALLINGS','LAS VEGAS','USA');
```

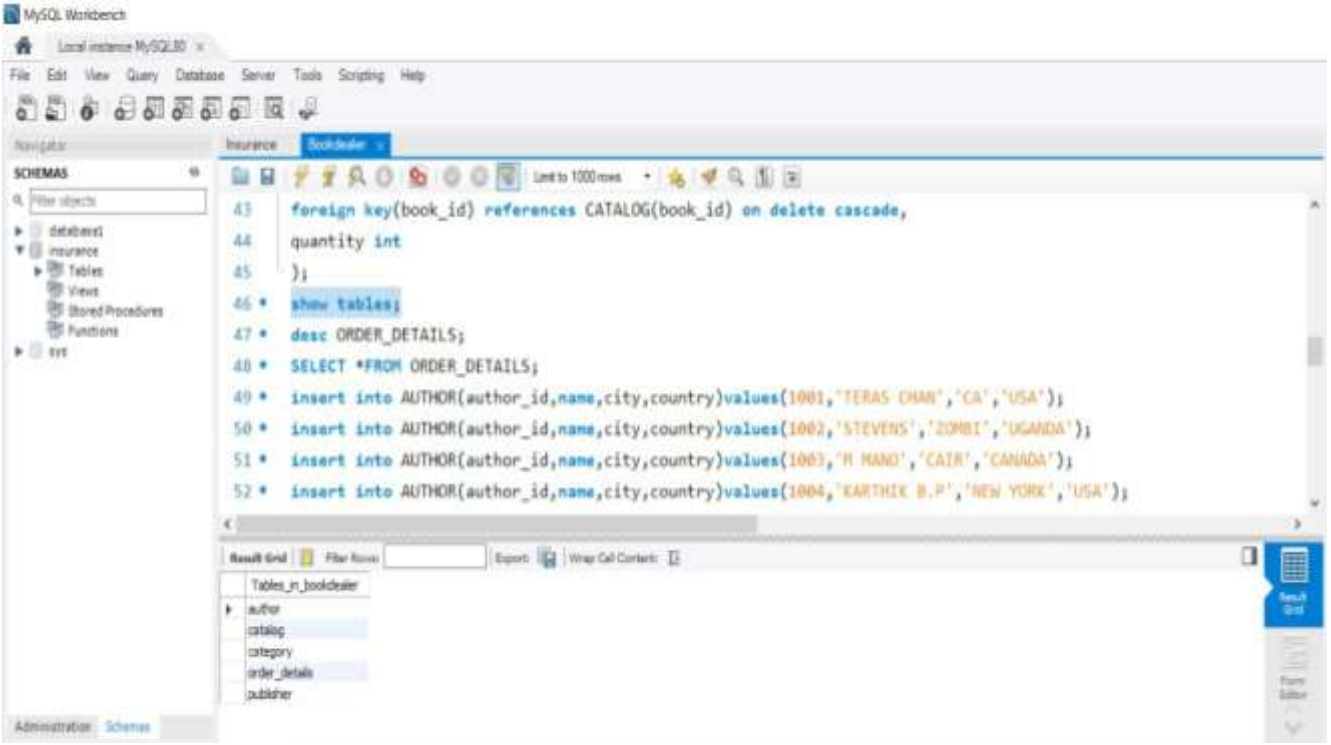
```
COMMIT;

desc AUTHOR;

SELECT *FROM AUTHOR;

insert into
PUBLISHER(publisher_id,name,city,country)values(1,'PEARSON','NEW
YORK','USA');

insert into
PUBLISHER(publisher_id,name,city,country)values(2,'EEE','NEW SOUTH
VALES','USA');

insert into PUBLISHER(publisher_id,name,city,country)values(3,'PHI','DELHI','INDIA');

insert into
PUBLISHER(publisher_id,name,city,country)values(4,'WILLEY','BERLIN','GE
RMANY');

insert into PUBLISHER(publisher_id,name,city,country)values(5,'MGH ','NEW

YORK','USA'); COMMIT;

desc PUBLISHER;

SELECT *FROM PUBLISHER;

insert into CATEGORY(category_id,description)values(1001,'COMPUTER

SCIENCE'); insert into

CATEGORY(category_id,description)values(1002,'ALGORITHM DESIGN');

insert into

CATEGORY(category_id,description)values(1003,'ELECTRONICS');

insert into

CATEGORY(category_id,description)values(1004,'PROGRAMMING'); insert

into CATEGORY(category_id,description)values(1005,'OPERATING

SYSTEMS'); COMMIT;

desc CATEGORY;

SELECT *FROM CATEGORY;

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(11,'
Unix System Prg',1001,1,1001,2000,251);

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(12,'Digit
al Signals',1002,2,1003,2001,425);

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(13,'Logi
c Design',1003,3,1002,1999,225);
```

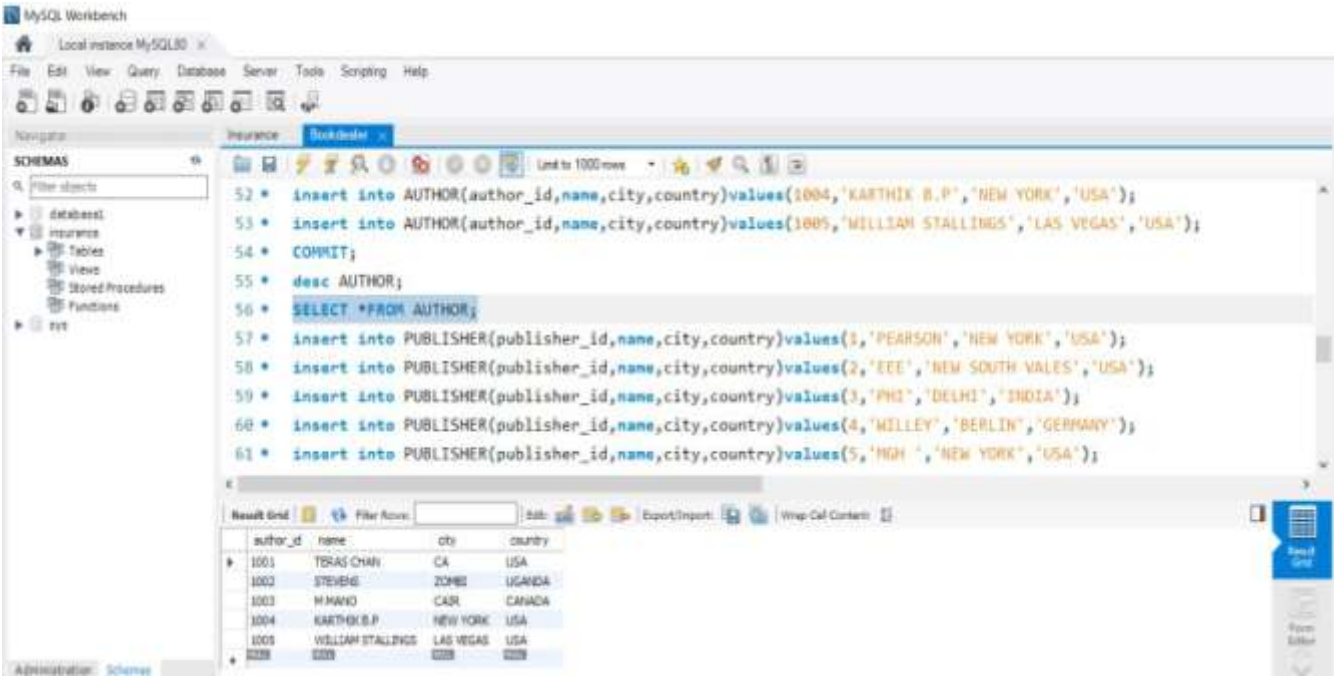
```
insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(14,'Ser
ver Prg',1004,4,1004,2001,333);

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(15,'Linu
x OS',1005,5,1005,2003,326);

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(16,'C++
Bible',1005,5 ,1001,2000,526);

insert into
CATALOG(book_id,title,author_id,publisher_id,category_id,year,price)values(17,'CO
BOL Handbook',1005,4,1001,2000,658);

COMMIT;

desc CATALOG;

SELECT *FROM CATALOG;

insert into

ORDER_DETAILS(order_no,book_id,quantity)values(1,11,5);

insert into

ORDER_DETAILS(order_no,book_id,quantity)values(2,12,8);

insert into

ORDER_DETAILS(order_no,book_id,quantity)values(3,13,15);

insert into

ORDER_DETAILS(order_no,book_id,quantity)values(4,14,22);

insert into

ORDER_DETAILS(order_no,book_id,quantity)values(5,15,3);

insert into

ORDER_DETAILS(order_no,book_id,quantity)values(2,17,10);

COMMIT;

desc ORDER_DETAILS;

SELECT *FROM ORDER_DETAILS;

SELECT AUTHOR.author_id,name,city,country FROM AUTHOR,CATALOG where
AUTHOR.author_id=CATALOG.author_id group by CATALOG.author_id having
count(CATALOG.author_id)>=2;

SELECT PRICE FROM CATALOG where year>2000;

select name from AUTHOR,CATALOG where
AUTHOR.author_id=CATALOG.author_id and book_id in(select book_id from
ORDER_DETAILS where quantity=(select max(quantity) from ORDER_DETAILS));
```

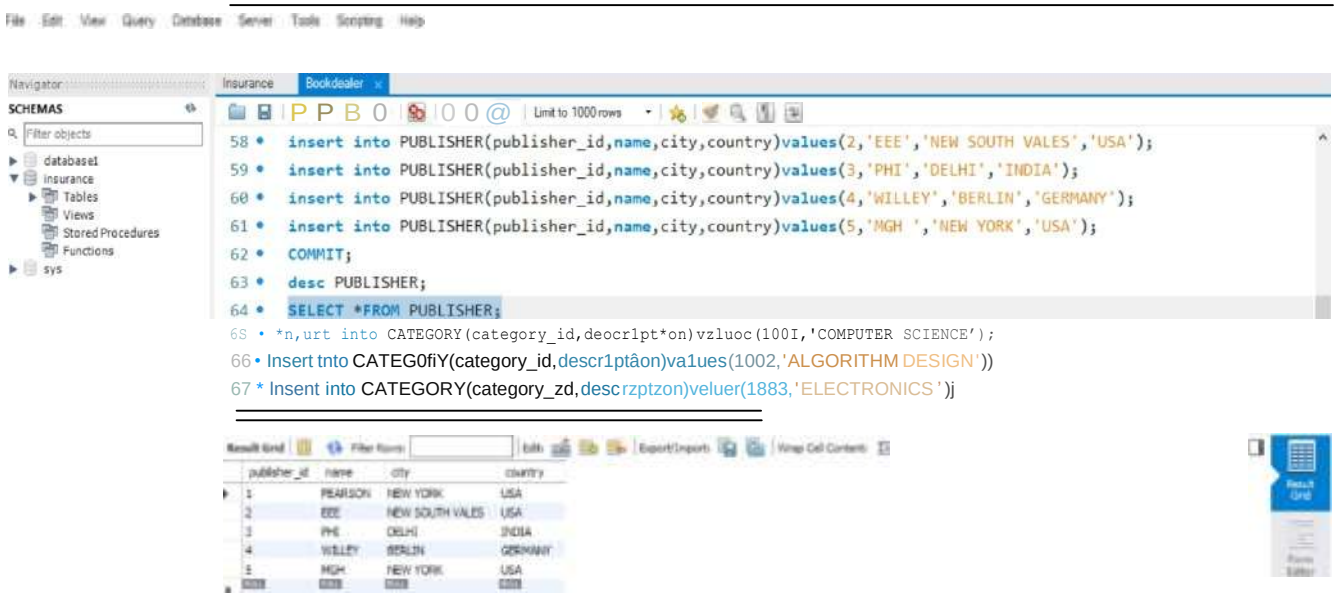
```
update CATALOG set price=1.1*price where publisher_id in(select publisher_id from
PUBLISHER where name='PEARSON');

COMMIT;
```

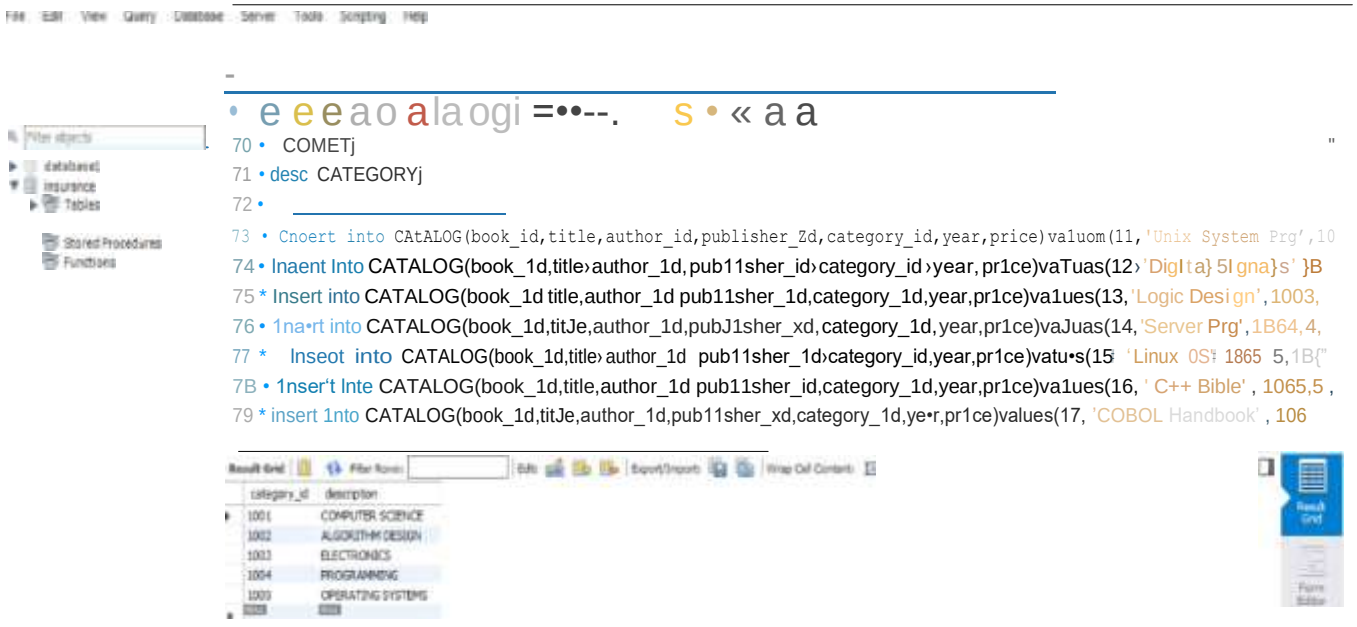
SELECT *FROM CATALOG;

OUTPUT SCREENSHOTS:



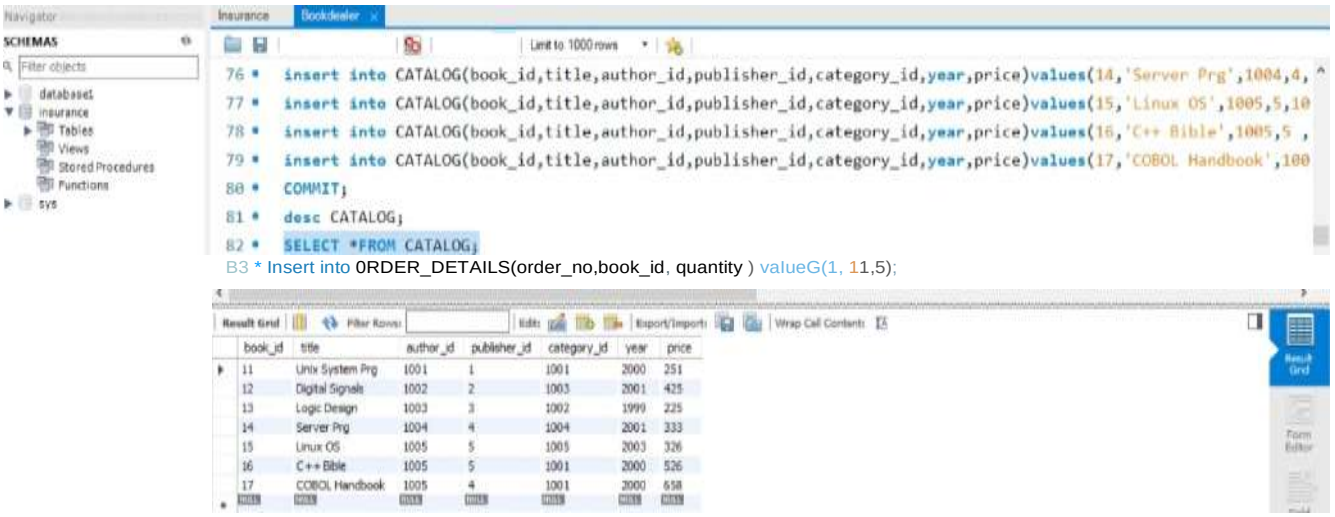



File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
database1
insurance
Tables
Views
Stored Procedures
Functions
sys
Insurance
Bookdealer
Limit to 1000 rows
58 • insert into PUBLISHER(publisher_id,name,city,country)values(2,'EEE','NEW SOUTH VALES','USA');
59 • insert into PUBLISHER(publisher_id,name,city,country)values(3,'PHI','DELHI','INDIA');
60 • insert into PUBLISHER(publisher_id,name,city,country)values(4,'WILLEY','BERLIN','GERMANY');
61 • insert into PUBLISHER(publisher_id,name,city,country)values(5,'MGH ','NEW YORK','USA');
62 • COMMIT;
63 • desc PUBLISHER;
64 • SELECT *FROM PUBLISHER;
6S • *n,urt into CATEGORY(category_id,deocr1pt*on)vzluoc(100I,'COMPUTER SCIENCE');
66 • Insert tnto CATEG0fiY(category_id,descr1ptâon)va1ues(1002,'ALGORITHM DESIGN'))
67 * Insent into CATEGORY(category_zd,descrzptzon)veluer(1883,'ELECTRONICS')j
publisher_id name city country
1 PEARSON NEW YORK USA
2 EEE NEW SOUTH VALES USA
3 PHI DELHI INDIA
4 WILLEY BERLIN GERMANY
5 MGH NEW YORK USA


File Edit View Query Database Server Tools Scripting Help
database1
insurance
Tables
Stored Procedures
Functions
70 • COMETj
71 • desc CATEGORYj
72 •
73 • Cnoert into CAtALOG(book_id,title,author_id,publisher_Zd,category_id,year,price)valuom(11,'Unix System Prg',10
74 • Inaent Into CATALOG(book_1d,title›author_1d,pub11sher_id›category_id›year,pr1ce)vaTuas(12›'DigIta}5Igna}s' }B
75 * Insert into CATALOG(book_1d title,author_1d pub11sher_1d,category_1d,year,pr1ce)va1ues(13,'Logic Design',1003,
76 • 1na•rt into CATALOG(book_1d,titJe,author_1d,pubJ1sher_xd,category_1d,year,pr1ce)vaJuas(14,'Server Prg',1B64,4,
77 * Inseot into CATALOG(book_1d,title›author_1d pub11sher_1d›category_id,year,pr1ce)vatu•s(15 'Linux 0S 1865 5,1B{"
7B • 1nser't Inte CATALOG(book_1d,title,author_1d pub11sher_id,category_1d,year,pr1ce)va1ues(16,' C++ Bible' , 1065,5 ,
79 * insert 1nto CATALOG(book_1d,titJe,author_1d,pub11sher_xd,category_1d,ye•r,pr1ce)values(17, 'COBOL Handbook' , 106
category_id description
1001 COMPUTER SCIENCE
1002 ALGORITHM DESIGN
1003 ELECTRONICS
1004 PROGRAMMING
1005 OPERATING SYSTEMS

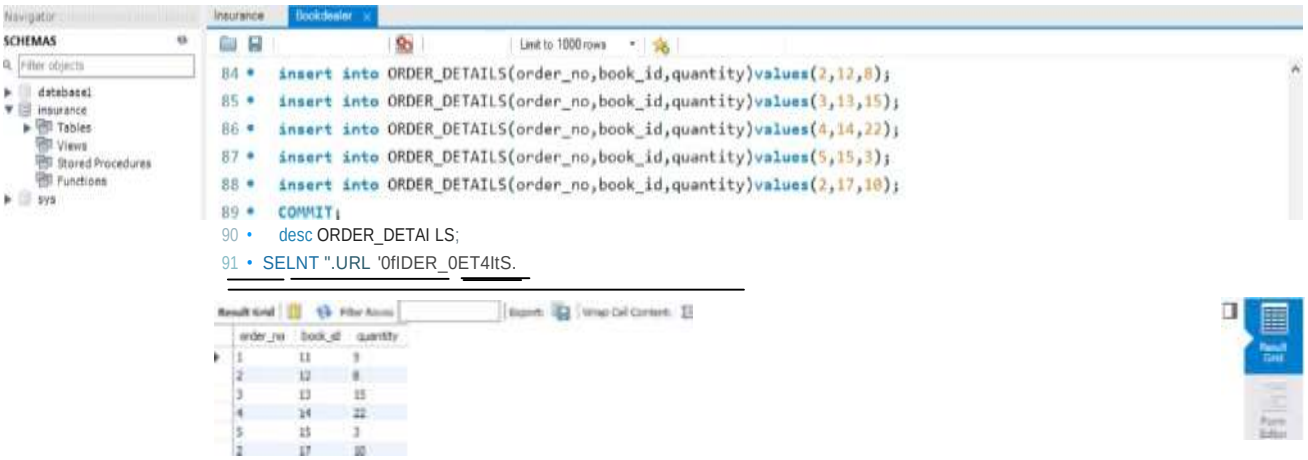
MySQL Workbench

File EdJt View Quay Oatcbcse Seryo Tcds Scripliro Help



B3 * Insert into 0RDER_DETAILS(order_no,book_id, quantity) valueG(1, 11,5);

book_id	title	author_id	publisher_id	category_id	year	price
11	Unix System Prg	1001	1	1001	2000	251
12	Digital Signals	1002	2	1003	2001	425
13	Logic Design	1003	3	1002	1999	225
14	Server Prg	1004	4	1004	2001	333
15	Linux OS	1005	5	1005	2003	326
16	C++ Bible	1005	5	1001	2000	526
17	COBOL Handbook	1005	4	1001	2000	658



MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

```
83 *  1nse rt  into ORDER DETAILS(order  no,book Id, quantity) va1ues( 1, 1,5) j
84 " lnse rt into 0RDER_DETATL5(order_no,book_i d, quantity)va1ues (2, I 2,8) j
85      insert  into  0RDER_DETA1LS(order_no, book_id, quantlt y ) valuest *, I °',18) j
B6 " insert  Snto  0RDER_DETAILS(order_no, book_1d, quantity ) values (4, 14, 22) j
B7 *  1nse rt-  into  ORDER_DETAILS(order_no,book_id, quantlty)va1ues(5,IN,   } j
88 "  insert  into ORDER_DETAILS(order_no,book_id, quant ity  ) values (2, 17, '16) j
89 •   C0f1FII T;
98 •   dasc 0RDER_DETAI LS j
9J •   SELECT *FROM ORDER_DETAILS;
92 •   SELECT AUTHOR.author_id,name,city,country FROM AUTHOR,CATALOG where AUTHOR.author_id=CATALOG.author_id group b
```

author_id	name	city	country
1005	WILLIAM STALLINGS	LAS VEGAS	USA

MySQL Workbench

File Edit Via Query DaBbase $ervel 'mls $cFipting Help

```
84 *  insert into ORDER_DETAILS(order_no,book_id,quantity)values(2,12,8);
85 ° Inse rt into 0RDER_DETAILS(order_no,book_id, quantity)va1ues ( J, t3, 15) ;
86 *  insert into ORDER_DETAILS(order_no,book_id,quantity)values(4,14,22);
87 *  insert Into 0RDER_DETAILS(order_no,book_1d,quantit y) val uas(5, IS , 3) j
88 * Insert into OItDEk 0ET4ILS(order no,book id, quantity)values(2, 17, \0) j
89 •  C0P8IT j
90 • desc 0RD£R_DETAI LS;
91 • SELECT *FR0PI ORDER_DETAI LS j
92 •  SELECT AUTHOR.author id, n••e,city,country FR0h AUTh0R,CATAL0G •h•re 4UTIJ0fi.author id=CAT4L0G.author id group b
93 • SELECT PRICE FRS CATAL& ah•r•  vaap2000
```

PRICE
425
333
326

Administration Schemas

MySQL Workbench

File Edit \fie O Oatabase Servw "cold Scripâng Help

```
BS •  insert into ORDEk DETAILS(order no,book id, quantity)vaIues(3,13, TS) j
86 ° Inse rt into 0RDER_DETAILS(order_no,book_i d,quantity)va1ues (4, J4, 22) ;
87 •  insert into 0RDER_DETAILS(order_no,book_id, quantity) values(S,1S, 3);
BB •  insert Into 0RDER_DETAILS(order_no,book_id, quantity) values(2, 17,16) ;
89 •  COFINIT j
90 ° desc 0RD£R_D£ TAIL S j
91 •  SELECT *lR0ñ 0RDER_DETAILS;
92 •  SELECT AUTHOR.author_id,nams,city,country FRON AUTHOh,CATALOG where AUTHOR.author_id:CATALOG.author_id group b
93 °  SELECT PRICE FR0/1 CATALOG therg year›2600j
94 •  smlcct ntmw from AUTHDR.CATALOG whcrm AUTHDR.authpn SdiCATALoG.:author Zd'nnd boot id in(nmlac€ book ld from OR'
```

MySQL Workbench

File Edit \ñw Q Oatabase Server Tell $oipting Help

```
91 •  SELECT *FRON 0RDER_DETAIL5;
92 •  SELECT AUTHOR.author_id,name,city,country PROM AUTHOA,CATALOG vhere AUTHDR.author_id:CATALOG.author_id gnoup b
93 •  SELECT PRICE FRON CATALOG uhere year›2006j
94 •  sel$cC name from AuTH0R,CATALOG whoro AUTHOA.authon_idzCATALOG.author_id nnd book_id in(*olecC book_id frem OR
95 •  ypdate CATALOG. set 'pr1cenl. 1*pr1ce uher.e publ1sher_id 1n(Pellet pub 1shéñ_1d rrea .PU8L.ISIJER uhere. naaleo 'PEARSO
96 •  0M1Tj
97 •  SELECT *FR08 CATALOG;
```

¥ It L ix Systm Prg 1001 I 1B1 @@ 276

Server Prg l0IH *

C++ Bible t00S 5

PROGRAM-3:ORDER PROCESSING DATABASE

QUESTION:

Consider the following relations for an Order

Processing database application in a company.

CUSTOMER (CUST #: int, cname: String, city:

String) ORDER (order #: int, odate: date, cust #: int,

ord-Amt: int) ITEM (item #: int, unit-price: int)

ORDER-ITEM (order #: int, item #: int,

qty: int) WAREHOUSE (warehouse #: int,

city: String)

SHIPMENT (order #: int, warehouse #: int, ship-date: date)

i) Create the above tables by properly specifying the primary keys and the foreign keys and the

foreig

n

keys.

ii) Enter at least five tuples for each relation.

iii) Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column

is the total

numbers of orders by the customer and the last column is the average order amount for that

customer.

iv) List the order# for orders that were shipped from all warehouses that the company has in a

specific city.

v) Demonstrate how you delete item# 10 from the ITEM table and make that field null in the

ORDER_ITEM

table.

PROGRAM CODE:

```
create database
Order_processing; use
Order_processing;
CREATE TABLE
CUSTOMER (
 cust_no int,
 cname
 VARCHAR(15), city
 VARCHAR(15),
 PRIMARY KEY(cust_no)
);
CREATE TABLE ORDERS(
 order_no
 int, odate
 date,
 cust_no int,
 foreign key(cust_no) references CUSTOMER(cust_no) on delete
 cascade, ord_Amt int,
 primary key(order_no)
 );
 create table ITEM (
 item_no int,
 unit_price int,
 primary
 key(item_no)
 );
create table
ORDER_ITEM (
order_no int,
item_no
int, qty int,
```

foreign key(order_no) references ORDERS(order_no) on delete cascade,

```
foreign key(item_no) references ITEM(item_no) on delete SET NULL
);
create table
WAREHOUSE(
warehouseno int,
city varchar(30),
primary key(warehouseno)
);
create table
SHIPMENT( order_no
int, warehouseno int,
ship_date date,
foreign key(order_no) references ORDERS(order_no) on delete cascade,
foreign key(warehouseno) references WAREHOUSE(warehouseno) on delete cascade
);
show tables;
insert into CUSTOMER(cust_no,cname,city)values(771,'PUSHPA
K','BANGALORE'); insert into
CUSTOMER(cust_no,cname,city)values(772,'SUMAN','MUMBAI'); insert
into CUSTOMER(cust_no,cname,city)values(773,'SOURAV','CALICUT');
insert into
CUSTOMER(cust_no,cname,city)values(774,'LAILA','HYDERABAD');
insert into
CUSTOMER(cust_no,cname,city)values(775,'FAIZAL','BANGALORE');
COMMIT;
desc CUSTOMER;
SELECT *FROM CUSTOMER;
insert into
ORDERS(order_no,odate,cust_no,ord_Amt)values(111,'22-01-
02',771,18000);
insert into
ORDERS(order_no,odate,cust_no,ord_Amt)values(112,'30-07-
02',774,6000);
```

```
insert into
ORDERS(order_no,odate,cust_no,ord_Amt)values(113,'03-04-
03',775,9000);
```

```
insert into
ORDERS(order_no,odate,cust_no,ord_Amt)values(114,'03-11-
03',775,29000);

insert into
ORDERS(order_no,odate,cust_no,ord_Amt)values(115,'10-12-
03',773,29000);

insert into
ORDERS(order_no,odate,cust_no,ord_Amt)values(116,'19-08-
04',772,56000);

insert into
ORDERS(order_no,odate,cust_no,ord_Amt)values(117,'10-09-
04',771,20000);

insert into
ORDERS(order_no,odate,cust_no,ord_Amt)values(118,'20-11-
04',775,29000);

insert into
ORDERS(order_no,odate,cust_no,ord_Amt)values(119,'13-02-
05',774,29000);

insert into
ORDERS(order_no,odate,cust_no,ord_Amt)values(120,'13-10-
05',775,29000);

COMMIT;

desc ORDERS;

SELECT *FROM ORDERS;

insert                                          into

ITEM(item_no,unit_price)values(5001,503);  insert

into    ITEM(item_no,unit_price)values(5002,750);

insert                                          into

ITEM(item_no,unit_price)values(5003,150);  insert

into    ITEM(item_no,unit_price)values(5004,600);

insert                                          into

ITEM(item_no,unit_price)values(5005,890);

COMMIT;

desc ITEM;

SELECT *FROM ITEM;

insert                                                  into
```

```
ORDER_ITEM(order_no,item_no,qty)values(111,5001,50);
insert into
ORDER_ITEM(order_no,item_no,qty)values(112,5003,20);
insert into
ORDER_ITEM(order_no,item_no,qty)values(113,5002,50);
insert into
ORDER_ITEM(order_no,item_no,qty)values(114,5005,60);
insert into
ORDER_ITEM(order_no,item_no,qty)values(115,5004,90);
```

```
insert into ORDER_ITEM(order_no,item_no,qty)values(116,5001,10);
insert into ORDER_ITEM(order_no,item_no,qty)values(117,5003,80);
insert into ORDER_ITEM(order_no,item_no,qty)values(118,5005,50);
insert into ORDER_ITEM(order_no,item_no,qty)values(119,5002,10);
insert into ORDER_ITEM(order_no,item_no,qty)values(120,5004,45);
COMMIT;
desc ORDER_ITEM;
SELECT *FROM ORDER_ITEM;
insert into WAREHOUSE(warehouseno,city)values(1,'DELHI'); insert into WAREHOUSE(warehouseno,city)values(2,'BOMBAY');
insert into WAREHOUSE(warehouseno,city)values(3,'CHENNAI');
insert into WAREHOUSE(warehouseno,city)values(4,'BANGALORE');
insert into WAREHOUSE(warehouseno,city)values(5,'BANGALORE');
insert into WAREHOUSE(warehouseno,city)values(6,'DELHI');
insert into WAREHOUSE(warehouseno,city)values(7,'BOMBAY');
insert into WAREHOUSE(warehouseno,city)values(8,'CHENNAI');
insert into WAREHOUSE(warehouseno,city)values(9,'DELHI');
insert into WAREHOUSE(warehouseno,city)values(10,'BANGALORE');
```

```
COMMIT;
desc WAREHOUSE;
SELECT *FROM WAREHOUSE;
insert into SHIPMENT(order_no,warehouseno,ship_date)values(111,1,'10-02-02');
insert into SHIPMENT(order_no,warehouseno,ship_date)values(112,5,'10-09-02');
insert into SHIPMENT(order_no,warehouseno,ship_date)values(113,8,'10-02-03');
insert into SHIPMENT(order_no,warehouseno,ship_date)values(114,3,'10-12-03');
insert into SHIPMENT(order_no,warehouseno,ship_date)values(115,9,'19-01-04');
insert into SHIPMENT(order_no,warehouseno,ship_date)values(116,1,'20-09-04');
insert into SHIPMENT(order_no,warehouseno,ship_date)values(117,5,'10-09-04');
```

```
insert into SHIPMENT(order_no,warehouseno,ship_date)values(118,7,'30-11-04'); insert into SHIPMENT(order_no,warehouseno,ship_date)values(119,7,'30-04-05');
insert into SHIPMENT(order_no,warehouseno,ship_date)values(120,6,'21-12-05'); COMMIT;
desc SHIPMENT;
SELECT *FROM SHIPMENT;
/*Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column
is the total numbers of orders by the customer and the last column is the average order amount for that customer.*/
SELECT C.CNAME as CUSTNAME, COUNT(*) as no_of_orders,AVG(O.ord_Amt) as AVG_ORDER_AMT FROM CUSTOMER C,
ORDERS O WHERE C.cust_no=O.cust_no GROUP BY C.CNAME;
/*List the order# for orders that were shipped from all warehouses that the company has in a specific city.*/
SELECT order_no FROM WAREHOUSE W, SHIPMENT S WHERE W.warehouseno=S.warehouseno AND CITY='BANGALORE';
/*Demonstrate how you delete item# 10 from the ITEM table and make that field null in the ORDER_ITEM table.*/
delete from ITEM where
item_no=5005; select *from ITEM;
select *from ORDER_ITEM;
```

OUTPUT SCREENSHOTS:

MySQL Workbench

File EOl Ve• 0 OeBbee Savm 'Qas 5c"%'9 H@Q

databases
insurance
Tables
Views
Stored Procedures
Functions
sys

```
GB    ship_date  aata,                                                    ^
39    foreign key(order no) re-Farencas ORDERS (order no) on dalata cascade,
40    foreign kay(warehouseno) raferancas ¥/AREH0USE (warehouseno) on delate cascade

42 •  ehaw taBd.esj
43 • 1nsez't Into CUSTOMER(cust_no cname,city)va2ues(77a 'PUSHPA K', 'BAftGALORE'}j
44 * insert Into CUSTOMER(cust_no cname, city)va2ues(77 2 'SUf'1AfJ ', ' IIU/'IBAI ')j
45    insert Into CUSTOMER(cust_no cname, city) va2ues(773, ' SOURAV ', ' CALICU ')j
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Tables_in_order_processing
customer
item
order_item
orders
shipment
warehouse

Administration Schemas

Field Types

File Edit View Query Database Server Tools Scripting Help

Navigator Insurance Bookdealer OrderProcessing

database
insurance
Tables
Views
sys

```
44 • Insert Into CUSTOMER(cust_no, cnane city)va1uas(772 ' 5U\1AtJ ' , ' I^.UI'18AI ');
43  *  insert  into CUSTOMER(cust no, cnaire, city)vaIues(77a,  'SOURAV' ,  ' CCL ICUT ')j
46 °  insert into CUSTO/tER(cust_no,cnane,city)va1ues(774, ' LAILA' , 'HYDEPABAD ')j
47 *  1nsert into CUSTOMER(cust_no,cnane,c1ty)va1ues(775 ' (4I2AL' ,'BA1tCAL0RE')j

49 • derc CUSTOMER j
56 °  SELECT *€€f@ CtGT RI
51 • insert Into ORDERS(order_no,odate,cust_no,ord_Amt)va1ues(UI, ' 22-81-82 ' ,771,18606)d
```

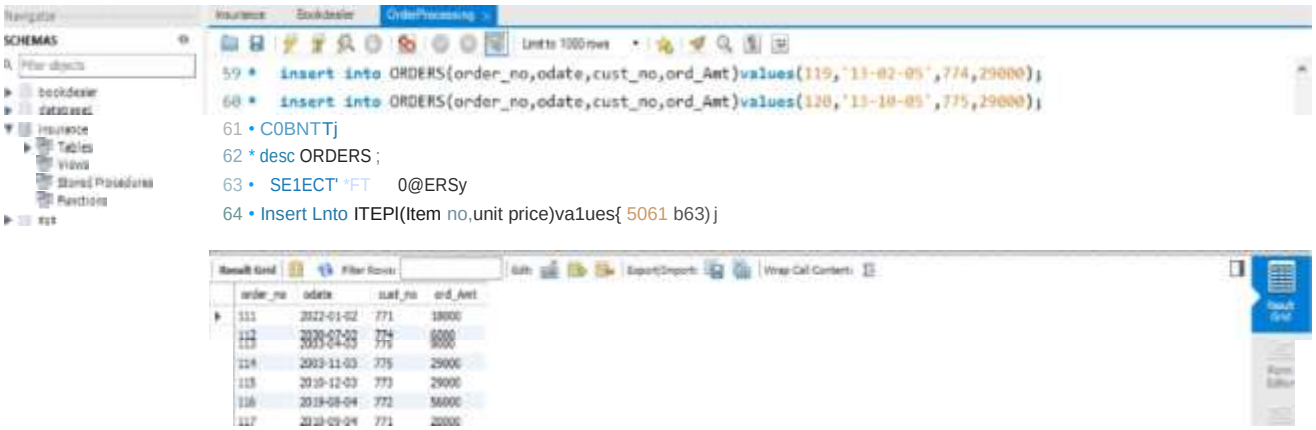
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

cust_no	cname	city
771	PUSHPA K	BANGALORE
772	SUMAN	MUMBAI
773	SOURAV	CALICUT
774	LAILA	HYDERABAD
775	FAIZAL	BANGALORE

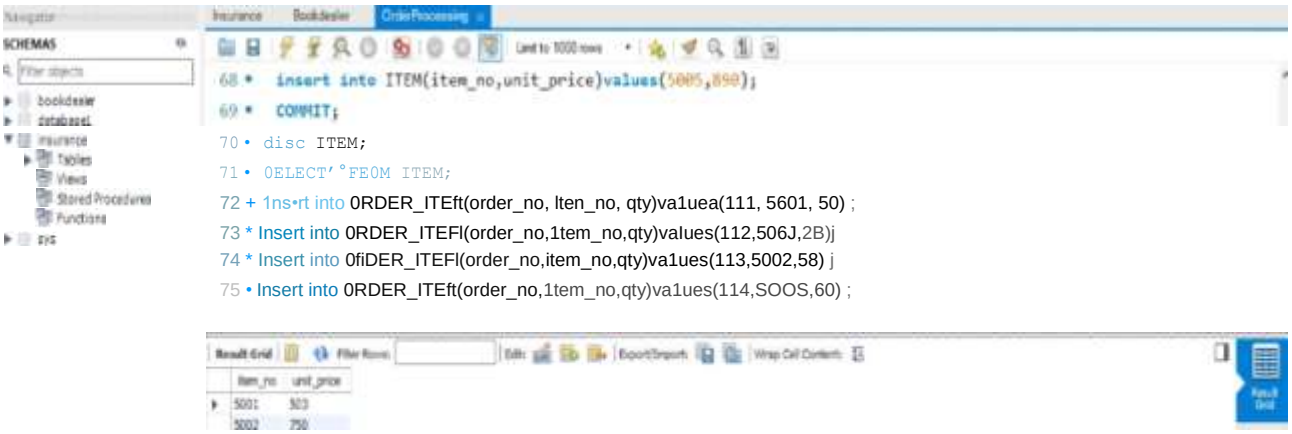
Administration Schemas

Field Types

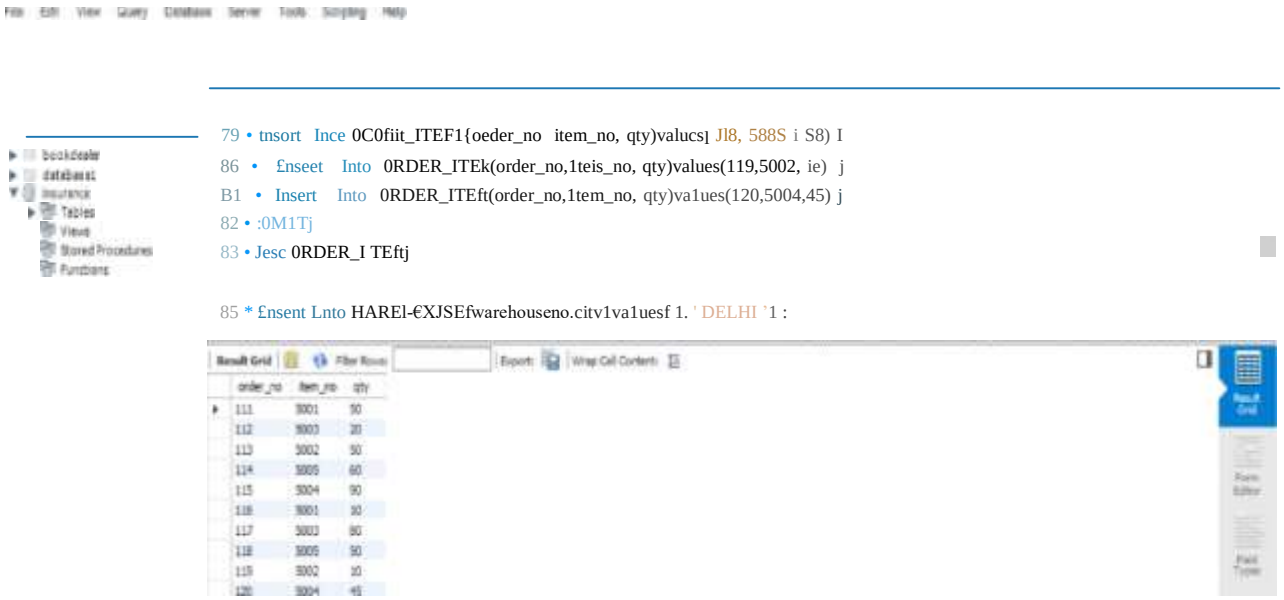
MySQL Workbench



MySQL Workbench

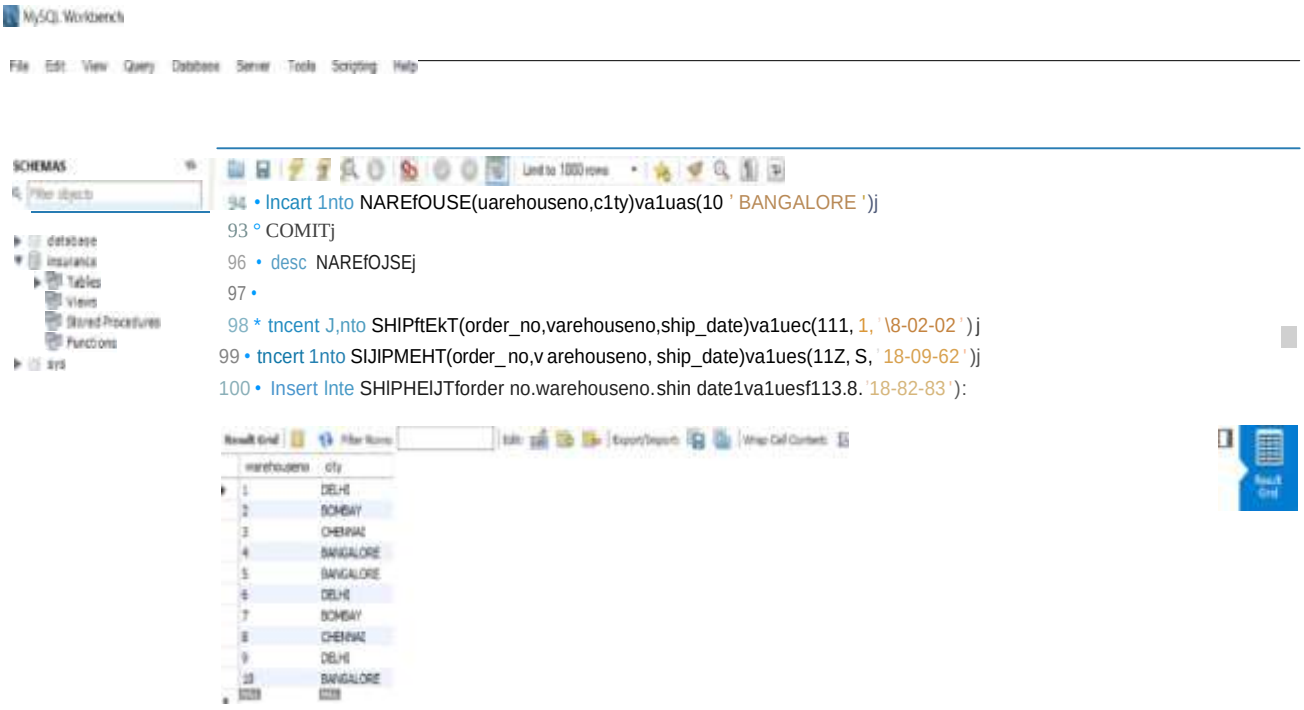


Administration



```
79 • tnsort  Ince 0C0fiit_ITEF1{oeder_no  item_no, qty)valucsl Jl8, 588S i S8) I
86 •  £nseet  Into  0RDER_ITEk(order_no,1teis_no, qty)values(119,5002, ie)  j
B1 •  Insert  Into  0RDER_ITEft(order_no,1tem_no, qty)va1ues(120,5004,45) j
82 • :0M1Tj
83 • Jesc 0RDER_I TEftj

85 * £nsent Lnto HAREl-€XJSEfwarehouseno.citv1va1uesf 1. ' DELHI '1 :
```

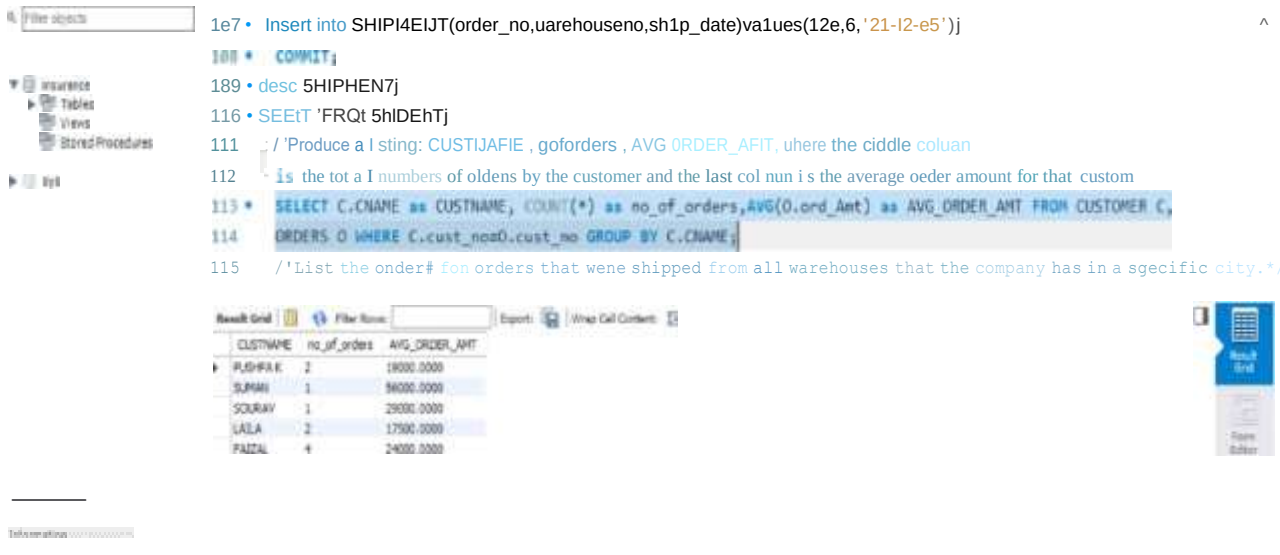
order_no	item_no	qty
111	5001	50
112	5003	20
113	5002	50
114	5005	60
115	5004	90
116	5001	30
117	5003	80
118	5005	50
119	5002	10
120	5004	45



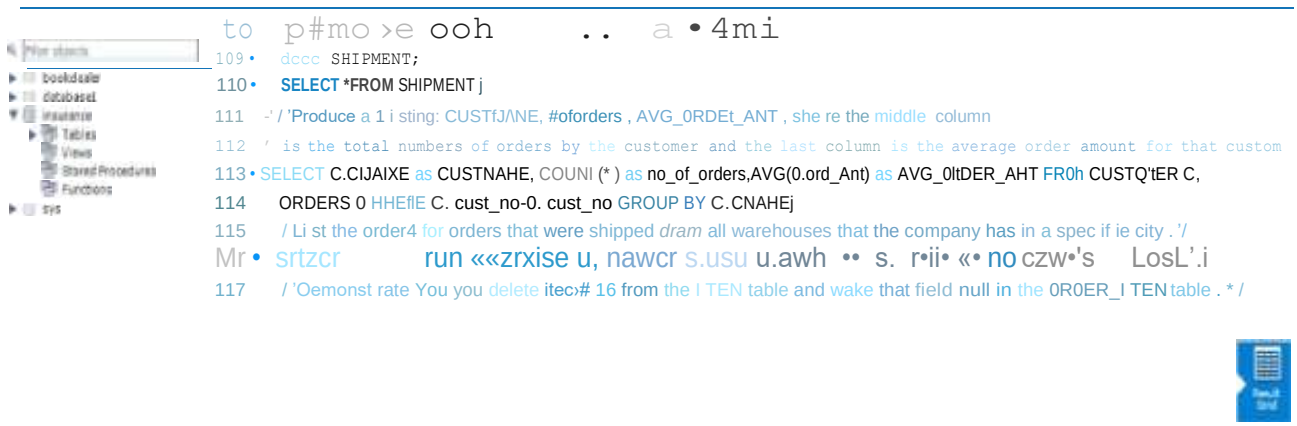
```
94 • Incart 1nto NAREfOUSE(uarehouseno,c1ty)va1uas(10 ' BANGALORE ')j
93 ° COMITj
96 •  desc  NAREfOJSEj
97 •
98 * tncent J,nto SHlPftEkT(order_no,varehouseno,ship_date)va1uec(111, 1, ' \8-02-02 ' ) j
99 • tncert 1nto SIJIPMEHT(order_no,v arehouseno, ship_date)va1ues(11Z, S, ' 18-09-62 ')j
100 •  Insert Inte SHlPHElJTforder no.warehouseno. shin date1va1uesf113.8. '18-82-83 '):
```

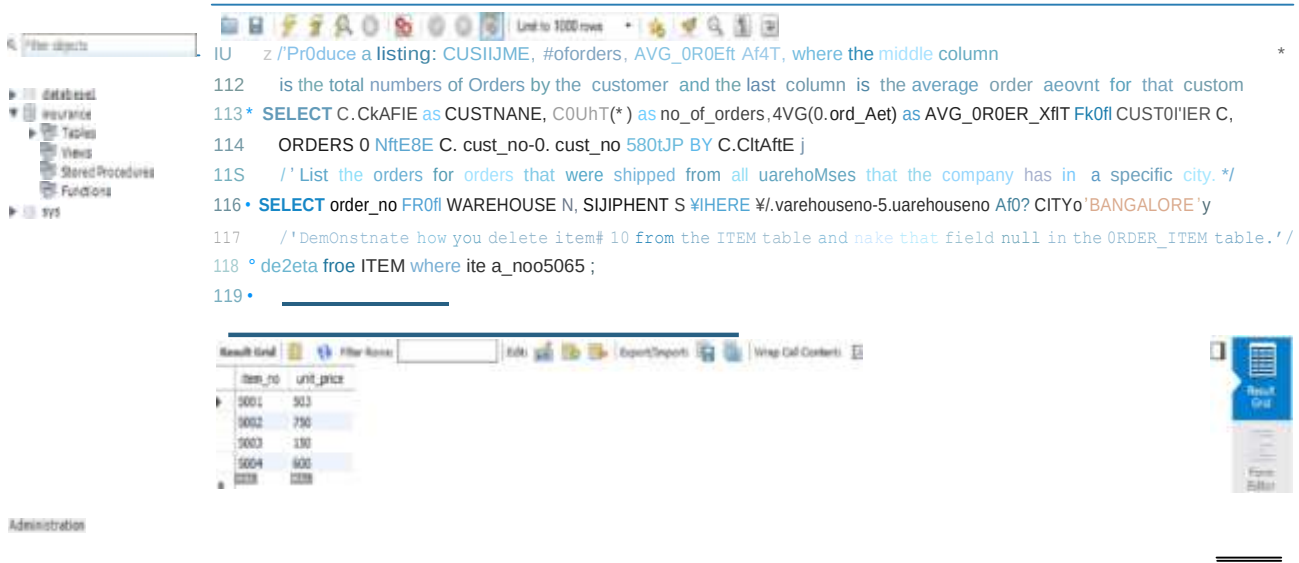
warehouseno	city
1	DELHI
2	BOMBAY
3	CHENNAI
4	BANGALORE
5	BANGALORE
6	DELHI
7	BOMBAY
8	CHENNAI
9	DELHI
10	BANGALORE

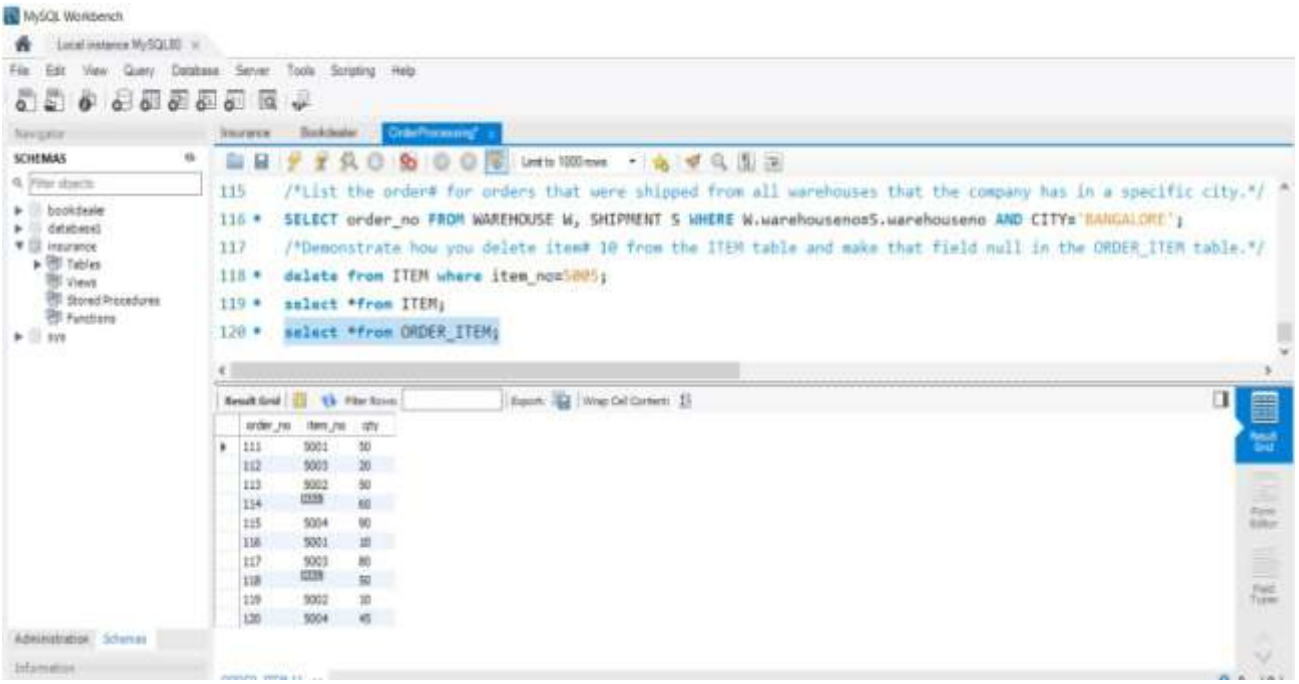

165 * t,nser•t :tnte SftIPMENT(order_no›uarehouseno,ship_date)va1uas(118 7, ' 36-11-B4 ') j
166 • Insent into SHIPMENT(order_no,varehouseno,ship_date)va1ues(119,7, ' 36-B4-65 ')j
167 * Insert Snto SHIPMENT(order_no,warehouseno ship_date)va1ues(129,6, ' 21- 12-65 ') j
108 • t0f¥IITj
109 • dssc SHIPMENT)
110 • SELECT *FROM SHIPMENT;
111 — / 'Produce a li stink: CUSTNANE . Ughorders . AVG ORDER AST . where the middle co1unn
r th 1
t17 5
1t8 7
Administration


1e7 • Insert into SHIPI4EIJT(order_no,uarehouseno,sh1p_date)va1ues(12e,6,'21-I2-e5')j
188 • COMMIT;
189 • desc 5HIPHEN7j
116 • SEEtT 'FRQt 5hlDEhTj
111 / 'Produce a I sting: CUSTIJAFIE , goforders , AVG 0RDER_AFIT, uhere the ciddle coluan
112 is the tot a I numbers of oldens by the customer and the last col nun i s the average oeder amount for that custom
113 • SELECT C.CNAME as CUSTNAME, COUNT(*) as no_of_orders,AVG(O.ord_Amt) as AVG_ORDER_AMT FROM CUSTOMER C,
114 ORDERS O WHERE C.cust_nosO.cust_no GROUP BY C.CNAME;
115 /'List the onder# fon orders that wene shipped from all warehouses that the company has in a sgecific city.*/
CUSTNAME no_of_orders AVG_ORDER_AMT
PUSHPA K 2 19000.0000
SUMAN 1 56000.0000
SOURAV 1 29000.0000
LAILA 2 17500.0000
FAIZAL 4 24000.0000
Information

MySQL Workbench







PROGRAM-4:BANKING DATABASE

Question:

Consider the following database for a banking enterprise.

BRANCH (branch-name: String, branch-city: String,

assets: real) ACCOUNTS (accno: int, branch-name:

String, balance: real) DEPOSITOR (customer-name:

String, customer-street: String, customer-city: String)

LOAN (loan-number: int, branch-name: String, amount:

real) BORROWER (customer-name: String, loan-

number: int)

i) Create the above tables by properly specifying the

primary keys and the foreign keys.

ii) Enter at least five tuples for each relation.

iii) Find all the customers who have at least two

accounts at the Main branch.

iv) Find all the customers who have an account at

all the branches located in a specific city.

v) Demonstrate how you delete all account tuples at every branch located in a specific city.

vi) Generate suitable reports.

vii) Create suitable front end for querying and displaying the results.

PROGRAM CODE:

```
create database
banking; use banking;

create table branch(
branch_name varchar(30) primary
key, branch_city varchar(30),
assets real);

create table accounts(
accno int primary key,
branch_name
varchar(30), balance
real,
foreign key (branch_name) references branch(branch_name) on delete cascade on update cascade);

create table customer(
customer_name varchar(30) primary
key, customer_street varchar(20),
customer_city varchar(20));

create table depositor(
customer_name
varchar(30), accno int,
```

```
primary key(customer_name ,accno),
foreign key (accno) references accounts(accno) on delete cascade on update cascade,
foreign key (customer_name) references customer(customer_name) on delete
cascade on update
cascade);

create table loan(
loan_number int primary
key, branch_name
varchar(30), amount real,
foreign key (branch_name) references branch(branch_name)
);

create table borrower (
customer_name
varchar(30), loan_number
int,
primary key(customer_name, loan_number),
foreign key (customer_name) references customer(customer_name) on delete
cascade on update cascade,
foreign key (loan_number) references loan(loan_number) on delete cascade on
update cascade);
show tables;
insert into branch(branch_name,branch_city,assets) values
('A','Bangalore',190000),
('B','Bangalore',200000),
('C','Delhi',235344),
('D','Chennai',1050560),
('E','Chennai',678909);
select *from branch;
```

```
insert into accounts(accno,branch_name,balance)
VALUES (1001,'A',10000),
(1002,'B',5000),
(1003,'C',7500),
(1004,'D',50000),
(1005,'D',75000),
(1006,'E',560),
(1007,"B",500),
(1008,"B",1500);
select *from accounts;

insert into customer(customer_name,customer_street,customer_city) VALUES
("Ravi","Dasarahalli","Bangalore"),
("Shyam","Indiranagar","Delhi"),
("Seema","Vasantnagar","Chennai")
, ("Arpita","Church
Street","Bangalore"), ("Vinay","MG
Road","Chennai");
select *from customer;

insert into depositor(customer_name,accno)
VALUES ("Ravi",1001),
("Ravi",1002),
("Shyam",1003),
("Seema",1004),
("Seema",1005),
("Arpita",1006),
("Vinay",1007),
("Vinay",1008);
select *from depositor;
```

```
insert into loan(loan_number,branch_name,amount) VALUES
(001,'A',10000),
(002,'B',25000),
(003,'B',250000),
(004,'C',5000),
(005,'E',90000);
select *from loan;

insert into borrower(customer_name,loan_number)
VALUES ("Arpita",001),
("Ravi",002),
("Arpita",003),
("Shyam",004),
("Vinay",005);
select *from borrower;

/*iii. Find all the customers who have at least two accounts at the Main branch */

select customer_name from depositor
join accounts on depositor.accno = accounts.accno where accounts.branch_name =
"D" group by depositor.customer_name having count(depositor.customer_name)
>=2;

/* iv. Find all the customers who have an account at all the branches located in a
specific city.*/

select customer_name from depositor
join accounts on accounts.accno = depositor.accno
join branch on branch.branch_name = accounts.branch_name
```

where branch.branch_city = "Bangalore"

GROUP BY depositor.customer_name

having count(DISTINCT branch.branch_name) = (SELECT

COUNT(branch_name) FROM branch

WHERE branch_city = 'Bangalore');

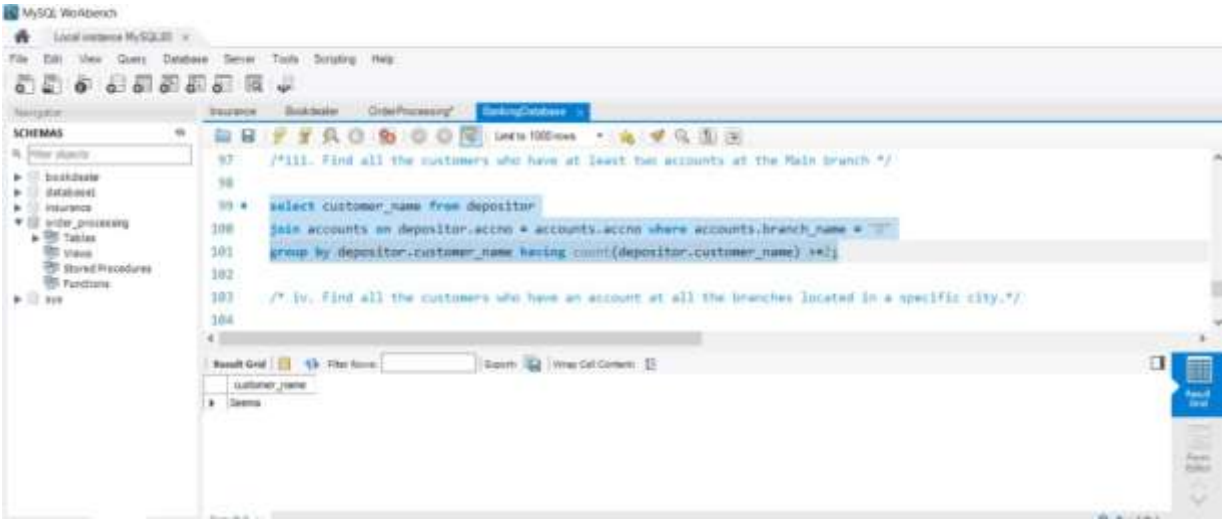
/*v. Demonstrate how you delete all account tuples at every branch located in a specific city.*/

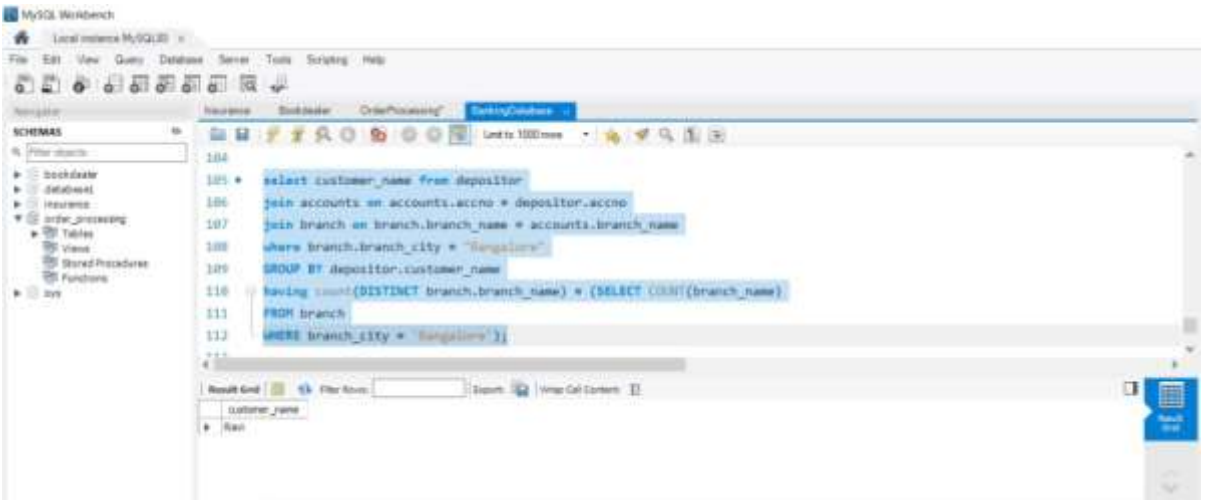
delete from accounts where branch_name in

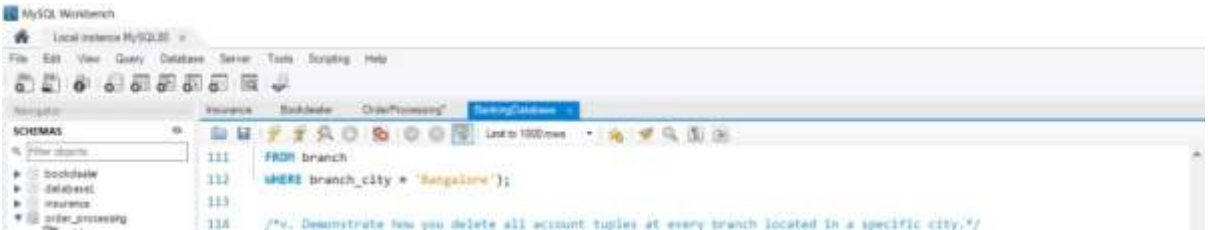
(select branch_name from branch where branch_city="Delhi");

select *from accounts;

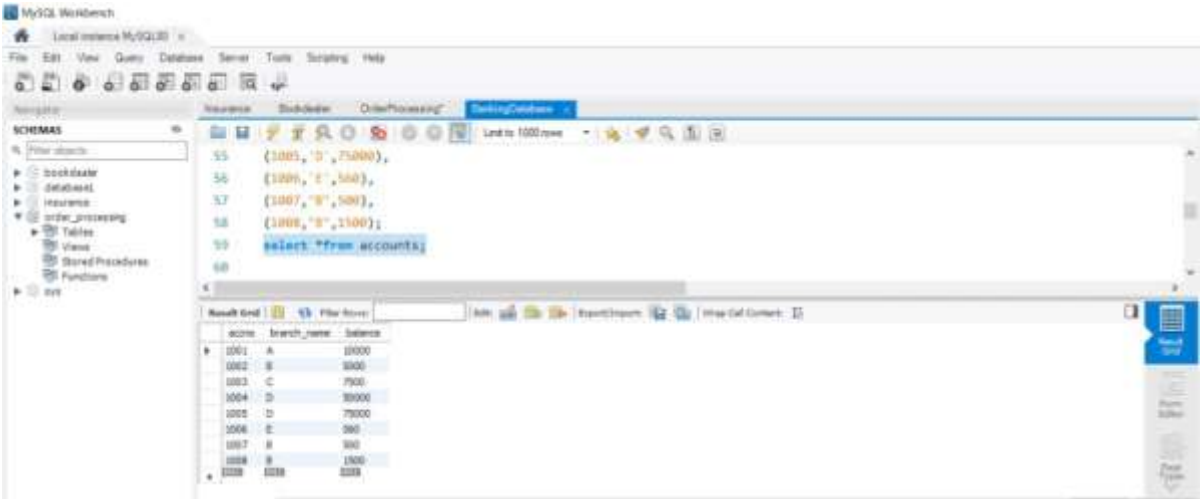
OUTPUT SCREENSHOTS:

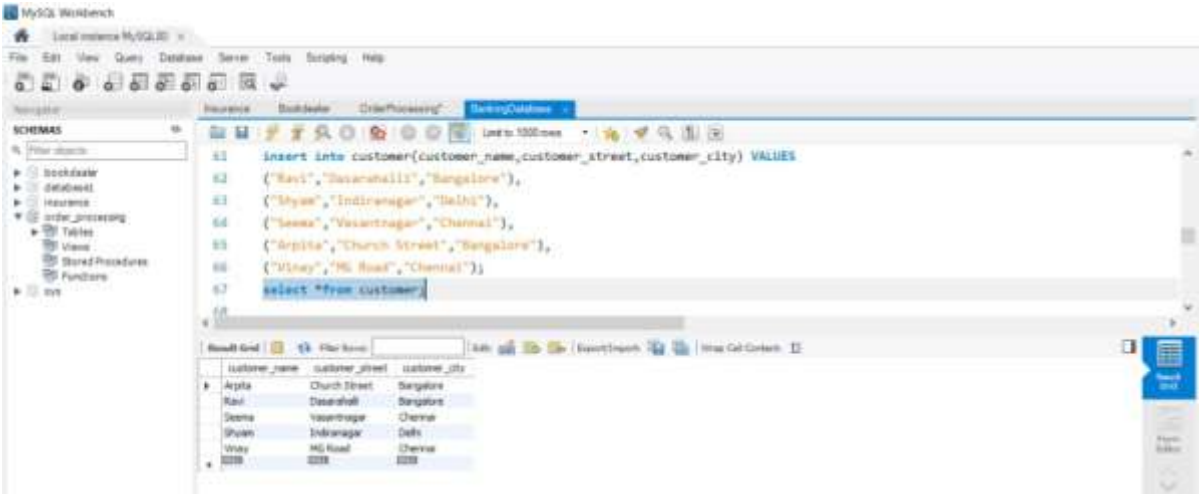


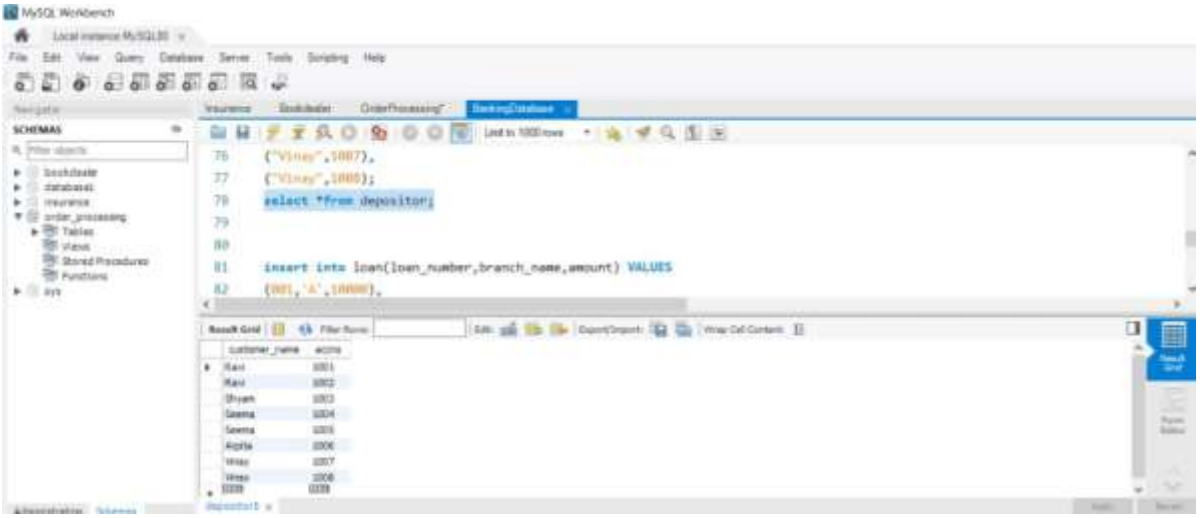


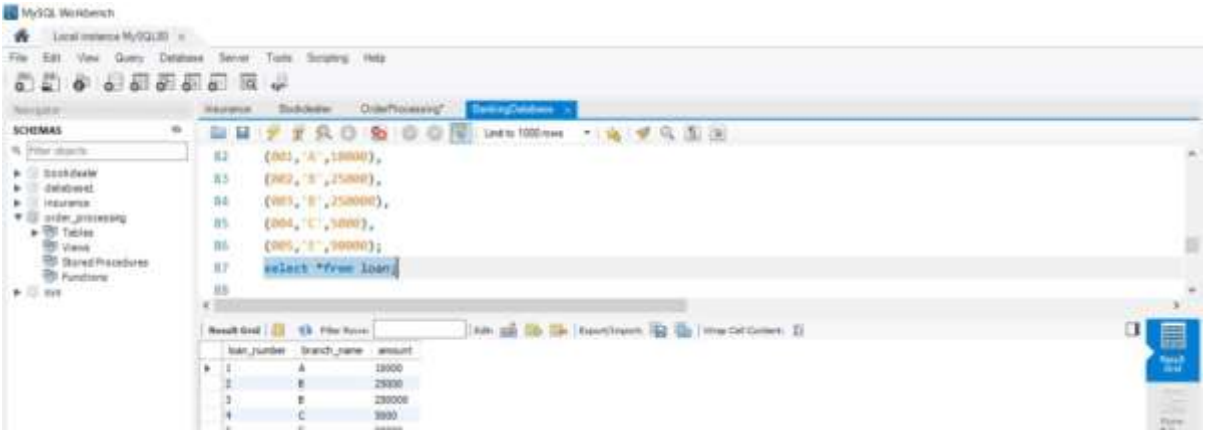


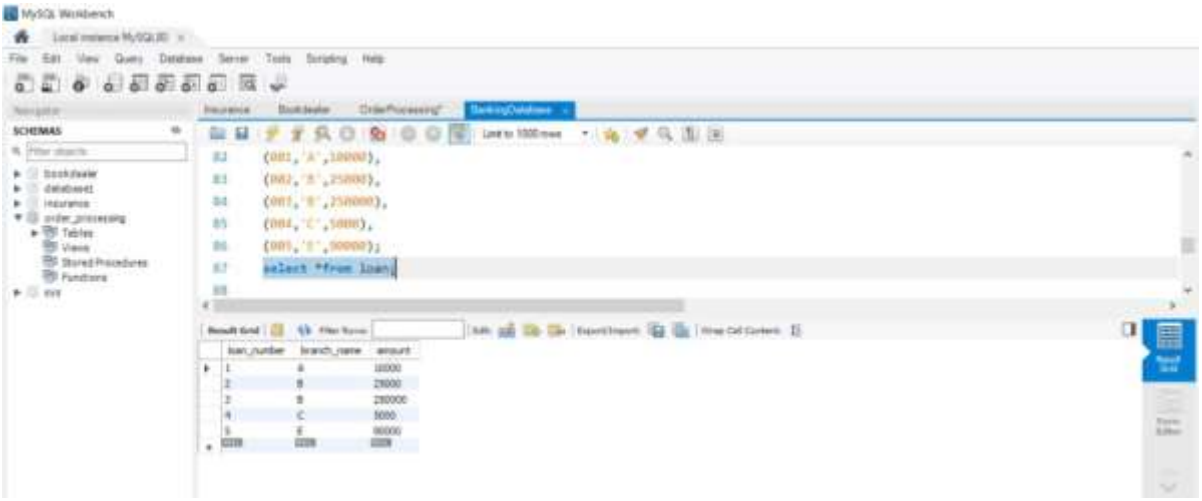
1005 D 75000

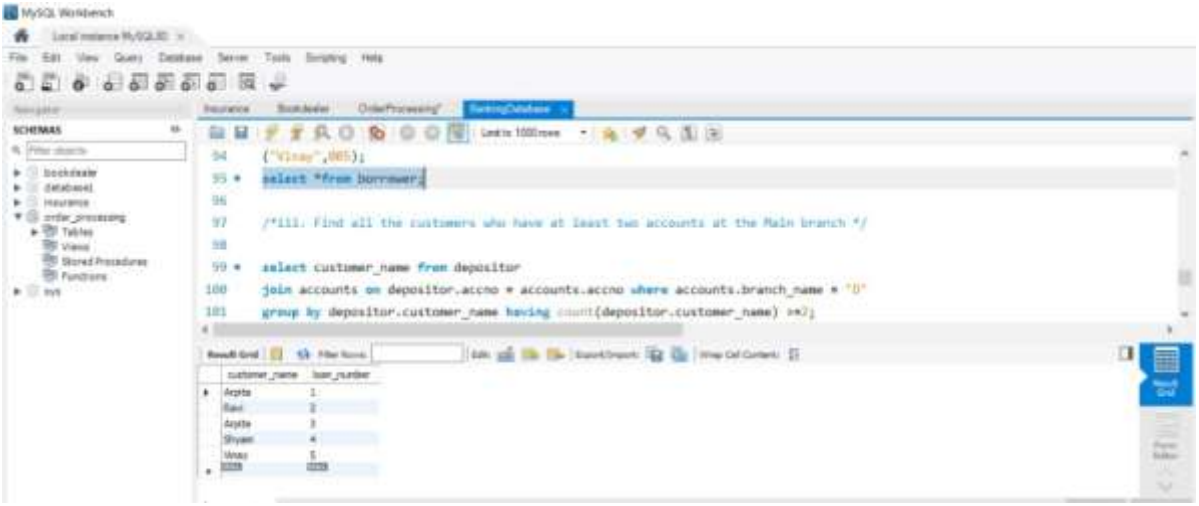

MySQL Workbench
(1005,'D',75000),
(1006,'E',900),
(1007,'B',500),
(1008,'B',1500);
select *from accounts;


MySQL Workbench
insert into customer(customer_name,customer_street,customer_city) VALUES
("Ravi","Dasarahalli","Bangalore"),
("Shyam","Indiranagar","Delhi"),
("Seema","Vasantnagar","Chennai"),
("Arpita","Church Street","Bangalore"),
("Vinay","MG Road","Chennai");
select *from customer;


MySQL Workbench
("Vinay",1007),
("Vinay",1008);
select *from depositor;
insert into loan(loan_number,branch_name,amount) VALUES
(001,'A',10000),


MySQL Workbench
(001,'A',10000),
(002,'B',25000),
(003,'B',250000),
(004,'C',5000),
(005,'D',90000);
select *from loan;





PROGRAM 5-STUDENT ENROLLMENT DATABASE

QUESTION:

Consider the following database of student enrollment in courses and books adopted for each course.

STUDENT (regno: String, name: String, major: String,

bdate: date) COURSE (course #: int, cname: String, dept:

String)

ENROLL (regno: String, cname: String, sem: int,

marks: int) BOOK_ADOPTION (course #: int, sem:

int, book-ISBN: int)

TEXT(book-ISBN:int, book-title: String, publisher:String, author:String)

i) Create the above tables by properly specifying the primary keys and the foreign keys.

ii) Enter at least five tuples for each relation.

iii) Demonstrate how you add a new text book to the database and make this book be adopted by some department.

iv) Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.

v) List any department that has all its adopted books published by a specific publisher.

PROGRAM CODE:

```
CREATE DATABASE STUDENTENROLLMENT;

USE STUDENTENROLLMENT;

CREATE TABLE
STUDENT( REG_NO
VARCHAR(30),
SNAME
VARCHAR(30),
   MAJOR
   VARCHAR(30),
   BDATE DATE,
   PRIMARY KEY(REG_NO)
);


CREATE TABLE
COURSE(
COURSE_ID INT,
   CNAME VARCHAR(30),
   DEPT VARCHAR(30),
   PRIMARY
   KEY(COURSE_ID)
);


CREATE TABLE ENROLL(
```

```
REG_NO
  VARCHAR(30),
  COURSE_ID INT,
  SEM INT,
  MARKS
  INT,
FOREIGN KEY(REG_NO) REFERENCES STUDENT(REG_NO) ON DELETE
CASCADE ON UPDATE CASCADE,
FOREIGN KEY(COURSE_ID) REFERENCES COURSE(COURSE_ID) ON
DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE
BOOK_ADOPTION(
COURSE_ID INT,
SEM INT,
BOOK_ISBN
INT,
PRIMARY KEY(BOOK_ISBN),
FOREIGN KEY(COURSE_ID) REFERENCES COURSE(COURSE_ID) ON
DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE TEXT(
BOOK_ISBN INT,
BOOK_TITLE
VARCHAR(30),
PUBLISHER
VARCHAR(30),
AUTHOR
VARCHAR(30),
FOREIGN KEY(BOOK_ISBN) REFERENCES
BOOK_ADOPTION(BOOK_ISBN) ON DELETE CASCADE ON UPDATE
CASCADE
);
```

```
show tables;

INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES
('CS01', 'RAM', 'DS', '1986-03-12');
```

```
INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('IS02', 'SMITH', 'USP', '1987-12-23');

INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('EC03', 'AHMED', 'SNS', '1985-04-17');

INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('CS03', 'SNEHA', 'DBMS', '1987-01-01');

INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('TC05', 'AKHILA', 'EC', '1986-10-06');

SELECT * FROM STUDENT;

INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (11, 'DS', 'CS'); INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (22, 'USP', 'IS'); INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (33, 'SNS', 'EC'); INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (44, 'DBMS', 'CS'); INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (55, 'EC', 'TC'); SELECT * FROM COURSE;

INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS01', 11, 4, 85); INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('IS02', 22, 6, 80); INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('EC03', 33, 2, 80); INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS03', 44, 6, 75); INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('TC05', 55, 2, 8); SELECT * FROM ENROLL;

INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (11,4,1); INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (11,4,2); INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (44,6,3); INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (44,6,4); INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (55,2,5); INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (22,6,6);
```

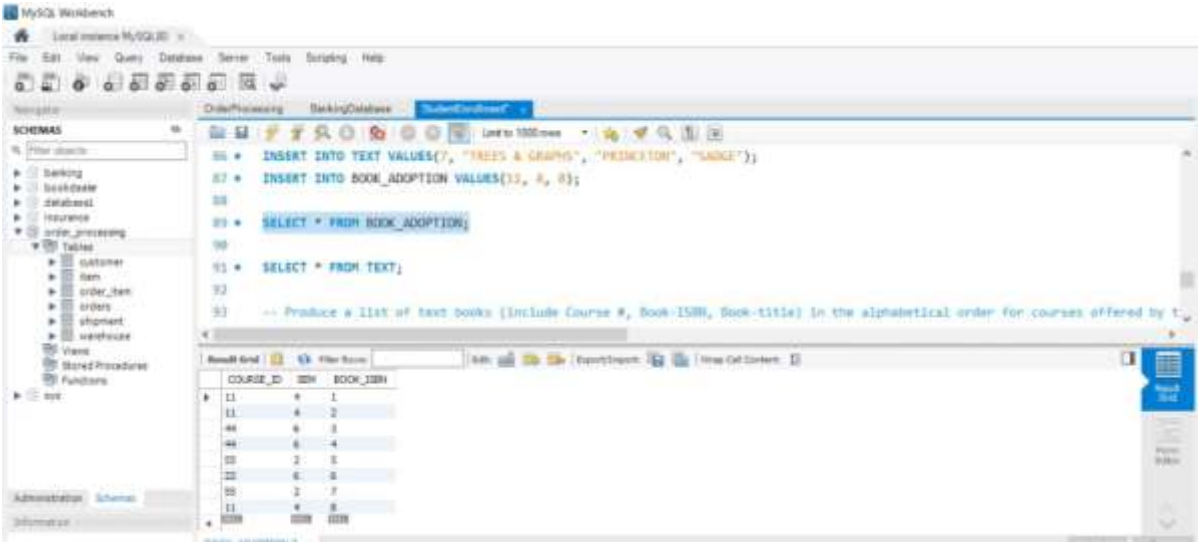
```
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN)
VALUES (55,2,7);
```

```
SELECT * FROM BOOK_ADOPTION;


INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (1, 'DS and C',
'Princeton', 'Padma Reddy');

INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (2,
'Fundamentals of DS', 'Princeton', 'Godse');

INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (3,
'Fundamentals of DBMS', 'Princeton', 'Navathe');

INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (4, 'SQL',
'Princeton', 'Foley');

INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (5,
'Electronic
circuits', 'TMH', 'Elmasri');

INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (6, 'Adv
unix
prog', 'TMH',

 'Stevens'); SELECT *

 FROM TEXT;


 -- Demonstrate how you add a new text book to the database and make this book be
adopted by some department.

INSERT INTO TEXT VALUES(7, "TREES & GRAPHS", "PRINCETON",

"SADGE"); INSERT INTO BOOK_ADOPTION VALUES(11, 4, 8);


SELECT * FROM BOOK_ADOPTION;


SELECT * FROM TEXT;


-- Produce a list of text books (include Course #, Book-ISBN, Book-title) in the
alphabetical order for courses offered by the ‘CS’ department that use more than two
books.

SELECT C.COURSE_ID,T.BOOK_ISBN,T.BOOK_TITLE FROM TEXT
T,COURSE C,BOOK_ADOPTION B WHERE
T.BOOK_ISBN=B.BOOK_ISBN AND

B.COURSE_ID=C.COURSE_ID AND C.DEPT="CS" AND (SELECT COUNT(B.BOOK_ISBN)
FROM BOOK_ADOPTION B WHERE

 C.COURSE_ID=B.COURSE_ID)>=2 ORDER BY T.BOOK_TITLE;
```

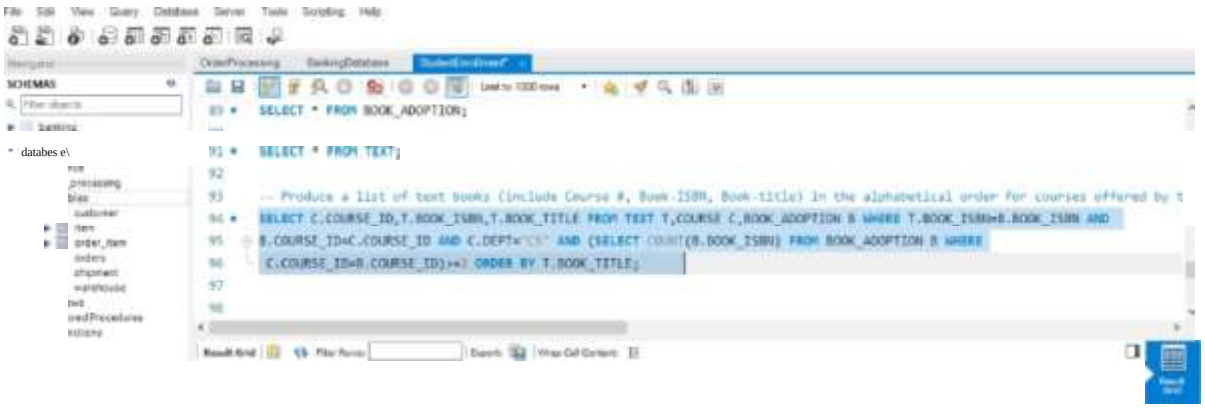
-- List any department that has all its adopted books published by a specific publisher. SELECT DISTINCT C.DEPT

```
FROM
COURSE C
WHERE
C.DEPT IN (
SELECT
C.DEPT
FROM COURSE C,BOOK_ADOPTION
B,TEXT T WHERE
C.COURSE_ID=B.COURSE_ID
AND
T.BOOK_ISBN=B.BOOK_ISB
N AND
T.PUBLISHER='Princeton')
AND C.DEPT NOT IN
(SELECT C.DEPT
FROM COURSE C,BOOK_ADOPTION
B,TEXT T WHERE
C.COURSE_ID=B.COURSE_ID
AND
T.BOOK_ISBN=B.BOOK_ISB
N AND T.PUBLISHER !=
'Princeton');
```

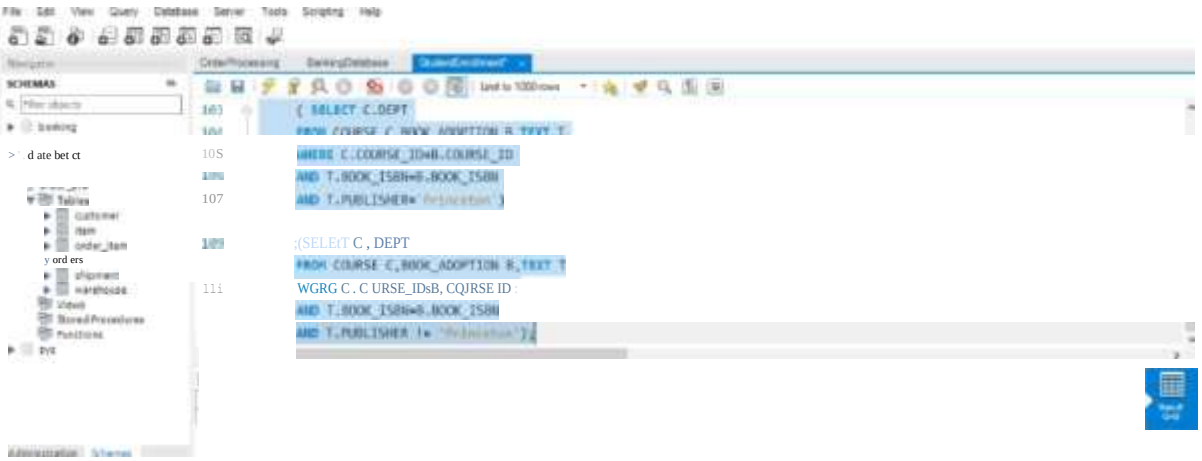
OUTPUT SCREENSHOTS:

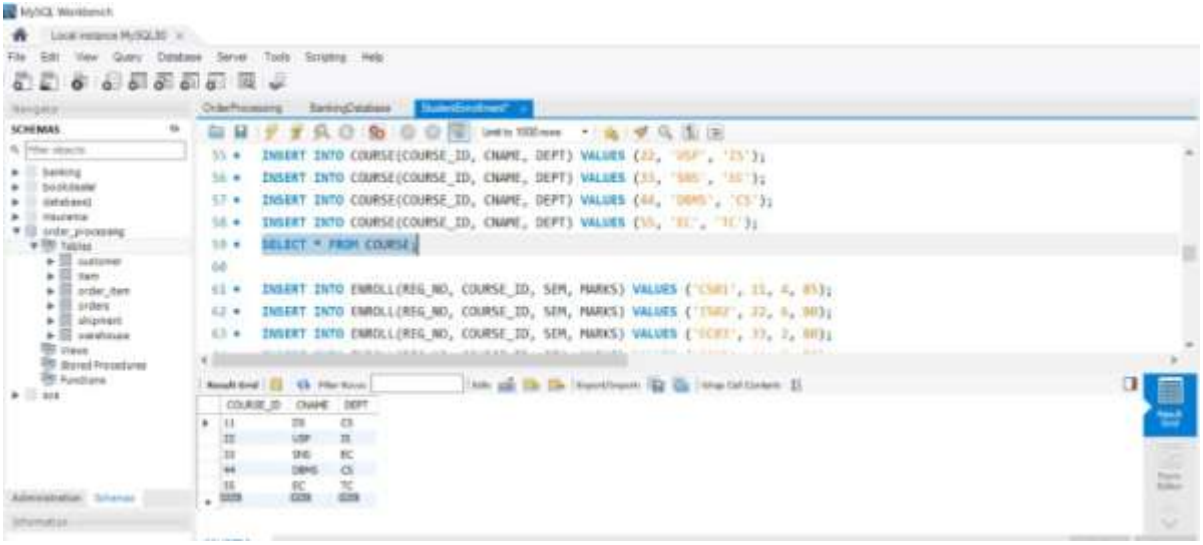

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
banking
bookdata
database1
insurance
order_processing
Tables
customer
item
order_item
orders
shipment
warehouse
Views
Stored Procedures
Functions
sys
Administration Schemas
OrderProcessing BankingDatabase StudentEnrollment
INSERT INTO TEXT VALUES(7, "TREES & GRAPHS", "PRENTICE", "SAGGE");
INSERT INTO BOOK_ADOPTION VALUES(13, 8, 8);
SELECT * FROM BOOK_ADOPTION;
SELECT * FROM TEXT;
-- Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by t
Result Grid
COURSE_ID SEM BOOK_ISBN
11 4 1
11 4 2
44 6 3
44 6 4
33 2 5
22 6 6
55 2 7
11 4 8

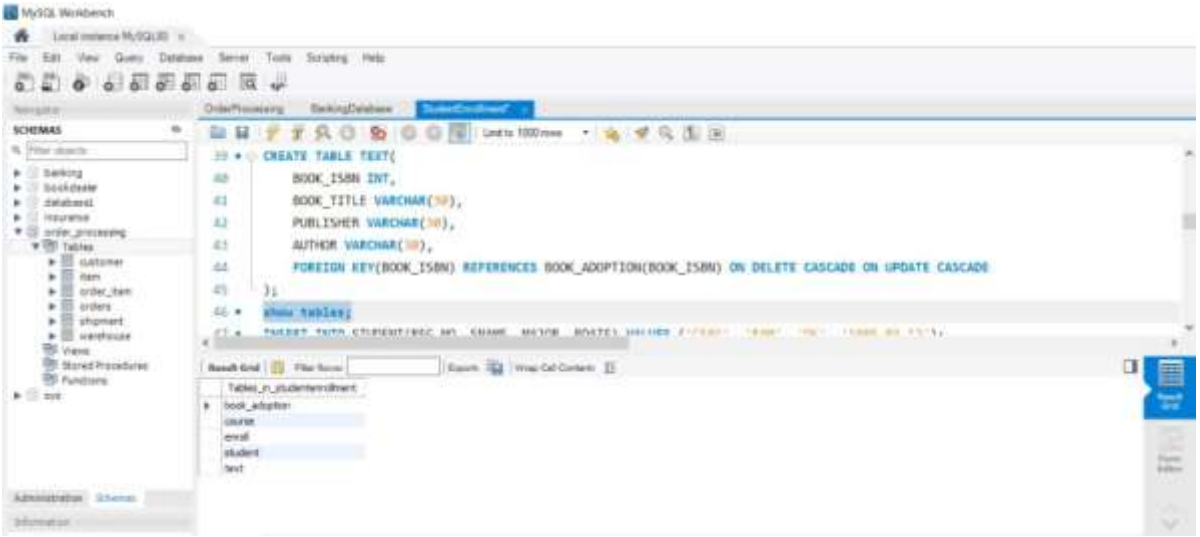
Legal i•sfanc< M\'S0 L80 x

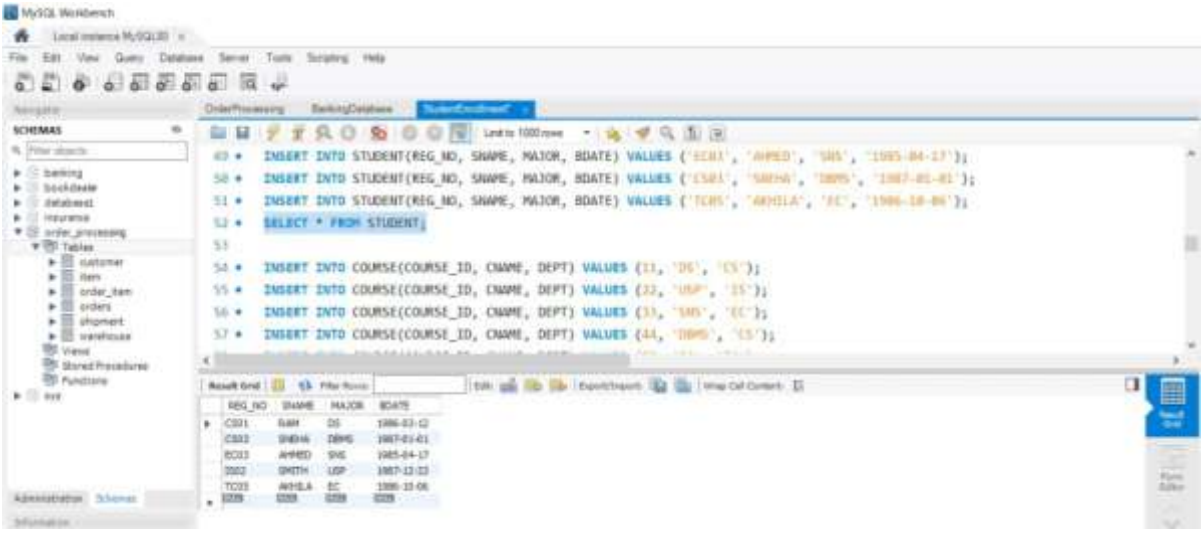


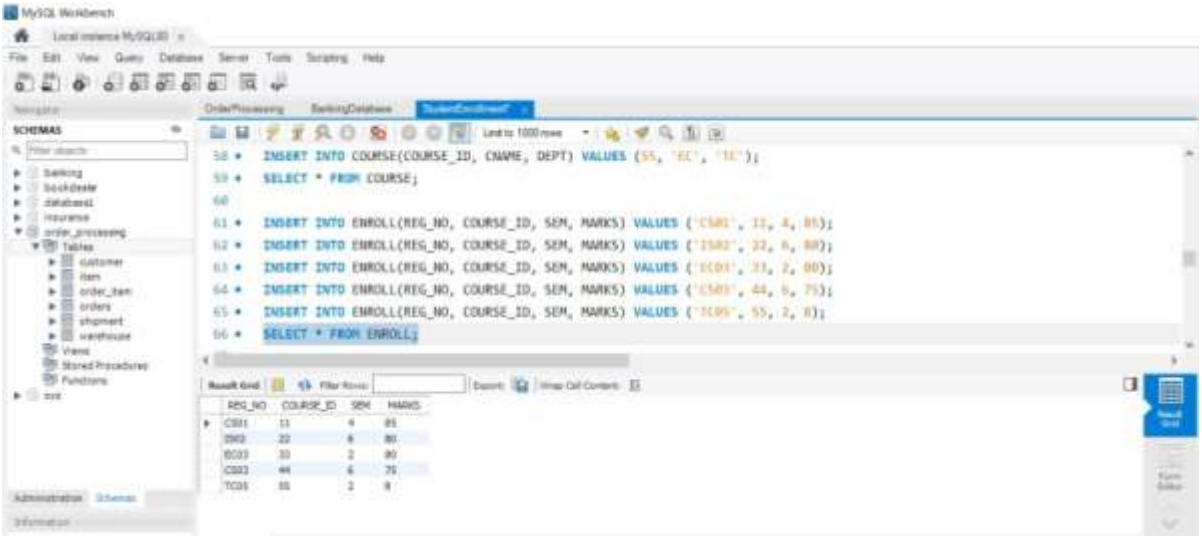
Local ii•sbncc M\'S0 L80 x

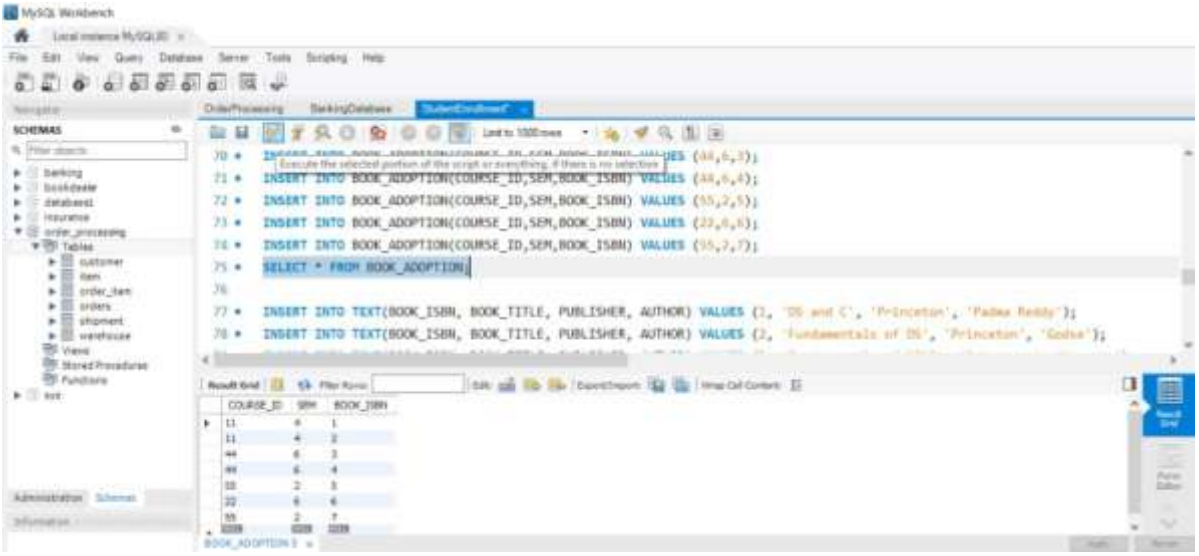


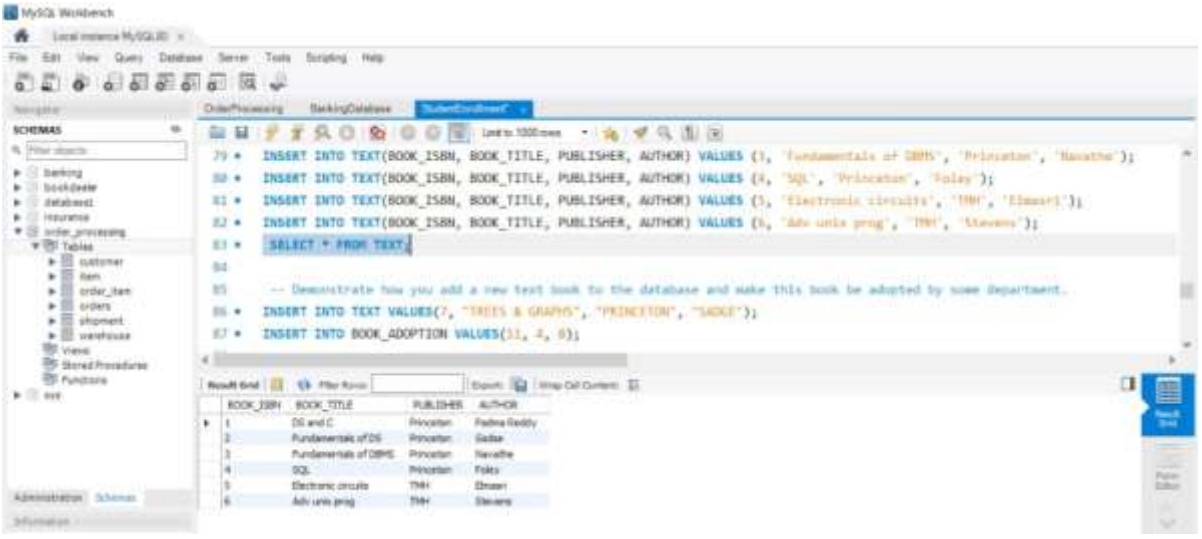

MySQL Workbench
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (22, 'USP', 'IS');
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (33, 'SNS', 'EC');
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (44, 'DBMS', 'CS');
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (55, 'EC', 'EC');
SELECT * FROM COURSE;
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS01', 11, 4, 85);
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('IS02', 22, 6, 90);
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('EC03', 33, 2, 80);


MySQL Workbench
CREATE TABLE TEXT(
BOOK_ISBN INT,
BOOK_TITLE VARCHAR(50),
PUBLISHER VARCHAR(50),
AUTHOR VARCHAR(50),
FOREIGN KEY(BOOK_ISBN) REFERENCES BOOK_ADOPTION(BOOK_ISBN) ON DELETE CASCADE ON UPDATE CASCADE
);
show tables;
book_adoption
course
enroll
student
text


MySQL Workbench
INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('EC01', 'AHMED', 'SNS', '1985-04-17');
INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('CS01', 'SNEHA', 'DBMS', '1987-01-01');
INSERT INTO STUDENT(REG_NO, SNAME, MAJOR, BDATE) VALUES ('EC05', 'AKHILA', 'EC', '1986-10-06');
SELECT * FROM STUDENT;
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (11, 'DS', 'CS');
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (22, 'USP', 'IS');
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (33, 'SNS', 'EC');
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (44, 'DBMS', 'CS');


MySQL Workbench
INSERT INTO COURSE(COURSE_ID, CNAME, DEPT) VALUES (55, 'EC', 'EC');
SELECT * FROM COURSE;
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS01', 11, 4, 85);
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('IS02', 22, 6, 80);
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('EC03', 33, 2, 90);
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('CS03', 44, 6, 75);
INSERT INTO ENROLL(REG_NO, COURSE_ID, SEM, MARKS) VALUES ('EC05', 55, 2, 8);
SELECT * FROM ENROLL;
REG_NO COURSE_ID SEM MARKS
CS01 11 4 85
IS02 22 6 80
EC03 33 2 90
CS03 44 6 75
EC05 55 2 8


MySQL Workbench
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (44,6,4);
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (55,2,5);
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (22,6,6);
INSERT INTO BOOK_ADOPTION(COURSE_ID,SEM,BOOK_ISBN) VALUES (55,2,7);
SELECT * FROM BOOK_ADOPTION;
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (1, 'DS and C', 'Princeton', 'Padma Reddy');
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (2, 'Fundamentals of DS', 'Princeton', 'Godse');
COURSE_ID SEM BOOK_ISBN
11 4 1
11 4 2
44 6 3
44 6 4
55 2 5
22 6 6
55 2 7


MySQL Workbench
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (3, 'Fundamentals of DBMS', 'Princeton', 'Navathe');
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (4, 'SQL', 'Princeton', 'Foley');
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (5, 'Electronic circuits', 'TMH', 'Elmasri');
INSERT INTO TEXT(BOOK_ISBN, BOOK_TITLE, PUBLISHER, AUTHOR) VALUES (6, 'Adv unix prog', 'TMH', 'Stevens');
SELECT * FROM TEXT;
-- Demonstrate how you add a new text book to the database and make this book be adopted by some department.
INSERT INTO TEXT VALUES(7, 'TREES & GRAPHS', 'PRINCETON', 'SAHNI');
INSERT INTO BOOK_ADOPTION VALUES(11, 4, 8);
BOOK_ISBN BOOK_TITLE PUBLISHER AUTHOR
1 DS and C Princeton Padma Reddy
2 Fundamentals of DS Princeton Godse
3 Fundamentals of DBMS Princeton Navathe
4 SQL Princeton Foley
5 Electronic circuits TMH Elmasri
6 Adv unix prog TMH Stevens

PROGRAM-6:MOVIE DATABASE

QUESTION:

Consider the schema for Movie Database:

ACTOR(Act_id, Act_Name, Act_Gender)

DIRECTOR(Dir_id, Dir_Name, Dir_Phone)

MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

MOVIE_CAST(Act_id, Mov_id, Role)

RATING(Mov_id, Rev_Stars)

Write SQL queries to

i. List the titles of all movies directed by 'Hitchcock'.

ii. Find the movie names where one or more actors acted in two or more movies.

iii. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation).

iv. 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.

v. Update rating of all movies directed by 'Steven Spielberg' to 5.

PROGRAM CODE:

```
create database movie;

use movie;

CREATE TABLE ACTOR (

ACT_ID INT,

ACT_NAME VARCHAR (20),

ACT_GENDER CHAR (1),

PRIMARY KEY (ACT_ID));


CREATE TABLE DIRECTOR (

DIR_ID INT,
```

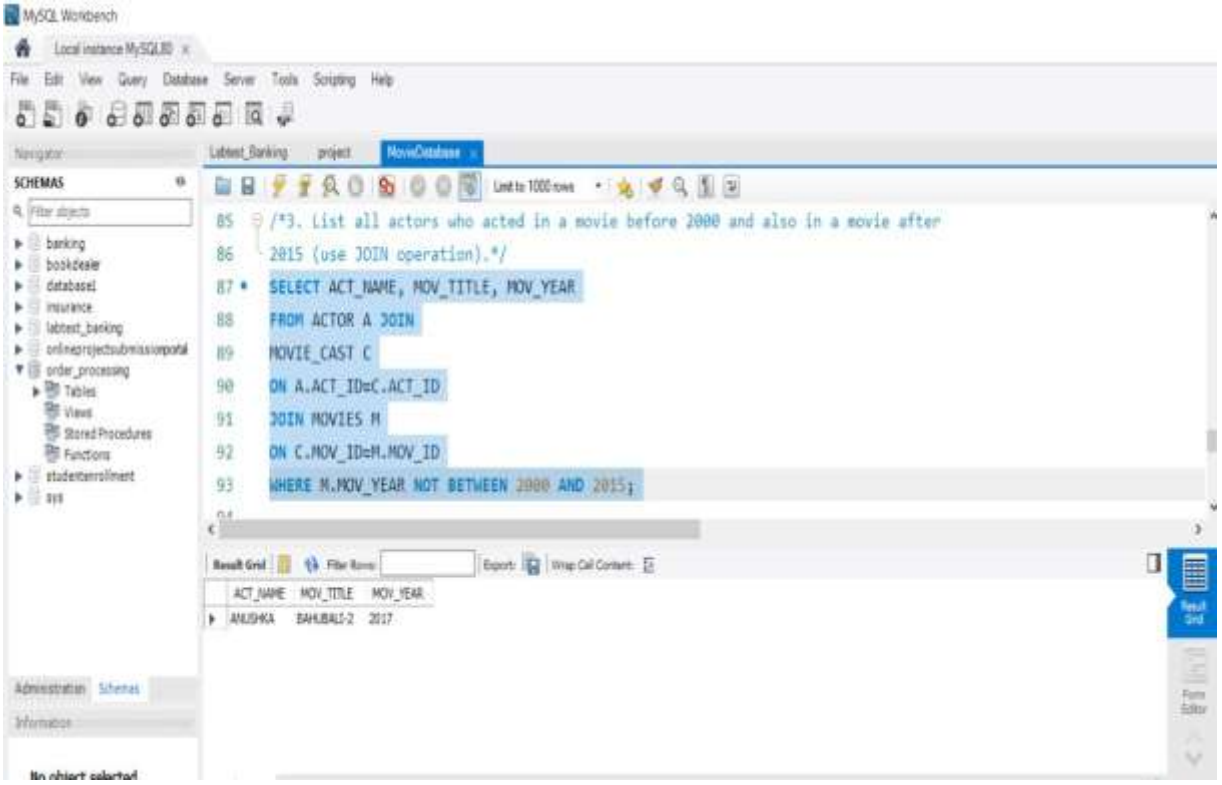
```
DIR_NAME VARCHAR (20),
DIR_PHONE real,
PRIMARY KEY (DIR_ID));

CREATE TABLE MOVIES (
MOV_ID INT,
MOV_TITLE VARCHAR (25),
MOV_YEAR INT,
MOV_LANG VARCHAR (12),
DIR_ID INT,
PRIMARY KEY (MOV_ID),
FOREIGN KEY (DIR_ID) REFERENCES DIRECTOR (DIR_ID));

CREATE TABLE MOVIE_CAST (
ACT_ID INT,
MOV_ID INT,
ROLE VARCHAR(10),
PRIMARY KEY (ACT_ID, MOV_ID),
FOREIGN KEY (ACT_ID) REFERENCES ACTOR (ACT_ID),
FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID));

CREATE TABLE RATING (
MOV_ID INT,
REV_STARS VARCHAR (25),
PRIMARY KEY (MOV_ID),
FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID));

INSERT INTO ACTOR VALUES (301,'ANUSHKA','F');
```

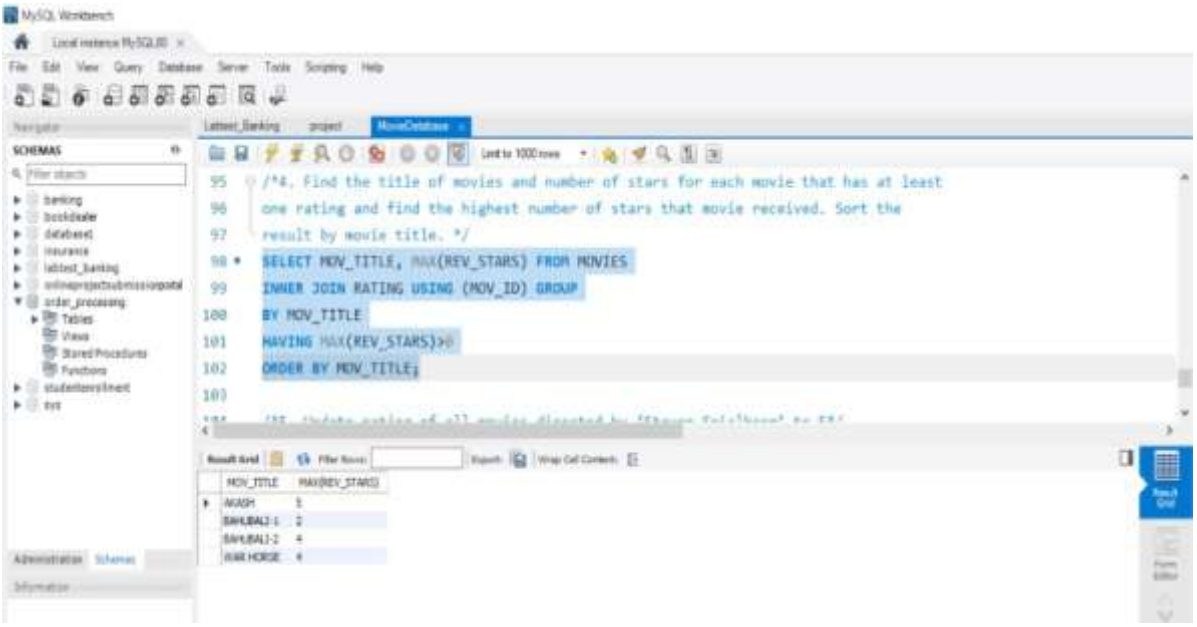
```
INSERT INTO ACTOR VALUES (302,'PRABHAS','M');
INSERT INTO ACTOR VALUES (303,'PUNITH','M');
INSERT INTO ACTOR VALUES (304,'JERMY','M');

INSERT INTO DIRECTOR VALUES (60,'RAJAMOULI', 8751611001);
INSERT INTO DIRECTOR VALUES (61,'HITCHCOCK', 7766138911);
INSERT INTO DIRECTOR VALUES (62,'FARAN', 9986776531);
INSERT INTO DIRECTOR VALUES (63,'STEVEN SPIELBERG', 8989776530);

INSERT INTO MOVIES VALUES (1001,'BAHUBALI-2', 2017, 'TELUGU', 60);
INSERT INTO MOVIES VALUES (1002,'BAHUBALI-1', 2015,'TELUGU', 60);
INSERT INTO MOVIES VALUES (1003,'AKASH', 2008,'KANNADA', 61);
INSERT INTO MOVIES VALUES (1004,'WAR HORSE', 2011, 'ENGLISH', 63);

INSERT INTO MOVIE_CAST VALUES (301, 1002, 'HEROINE');
INSERT INTO MOVIE_CAST VALUES (301, 1001, 'HEROINE');
INSERT INTO MOVIE_CAST VALUES (303, 1003, 'HERO');
INSERT INTO MOVIE_CAST VALUES (303, 1002,'GUEST');
INSERT INTO MOVIE_CAST VALUES (304, 1004, 'HERO');

INSERT INTO RATING VALUES (1001,'4');
INSERT INTO RATING VALUES (1002,'2');
INSERT INTO RATING VALUES (1003,'5');
INSERT INTO RATING VALUES (1004,'4');
SELECT * FROM ACTOR;

/*1. List the titles of all movies directed by‘Hitchcock’.*/
SELECT MOV_TITLE
```

```
FROM MOVIES
WHERE DIR_ID IN (SELECT DIR_ID
FROM DIRECTOR
WHERE DIR_NAME = 'HITCHCOCK');

/*2. Find the movie names where one or more actors acted in two or more movies.*/
SELECT MOV_TITLE
FROM MOVIES M, MOVIE_CAST MV
WHERE M.MOV_ID=MV.MOV_ID AND ACT_ID IN (SELECT ACT_ID
FROM MOVIE_CAST GROUP BY ACT_ID HAVING COUNT(ACT_ID)>1)
GROUP BY MOV_TITLE HAVING COUNT(MOV_TITLE)>1;

/*3. List all actors who acted in a movie before 2000 and also in a movie after
2015 (use JOIN operation).*/
SELECT ACT_NAME, MOV_TITLE, MOV_YEAR
FROM ACTOR A JOIN
MOVIE_CAST C
ON A.ACT_ID=C.ACT_ID
JOIN MOVIES M
ON C.MOV_ID=M.MOV_ID
WHERE M.MOV_YEAR NOT BETWEEN 2000 AND 2015;

/*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. */
SELECT MOV_TITLE, MAX(REV_STARS) FROM MOVIES
INNER JOIN RATING USING (MOV_ID) GROUP
```

```
BY MOV_TITLE
HAVING MAX(REV_STARS)>0
ORDER BY MOV_TITLE;

/*5. Update rating of all movies directed by ‘Steven Spielberg’ to 5*/
UPDATE RATING
SET REV_STARS='5'
WHERE MOV_ID = (SELECT MOV_ID FROM MOVIES
WHERE DIR_ID = (SELECT DIR_ID FROM DIRECTOR
WHERE DIR_NAME ='STEVEN SPIELBERG'));
select * from rating;
```

OUTPUT SCREENSHOTS:

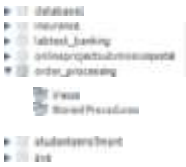




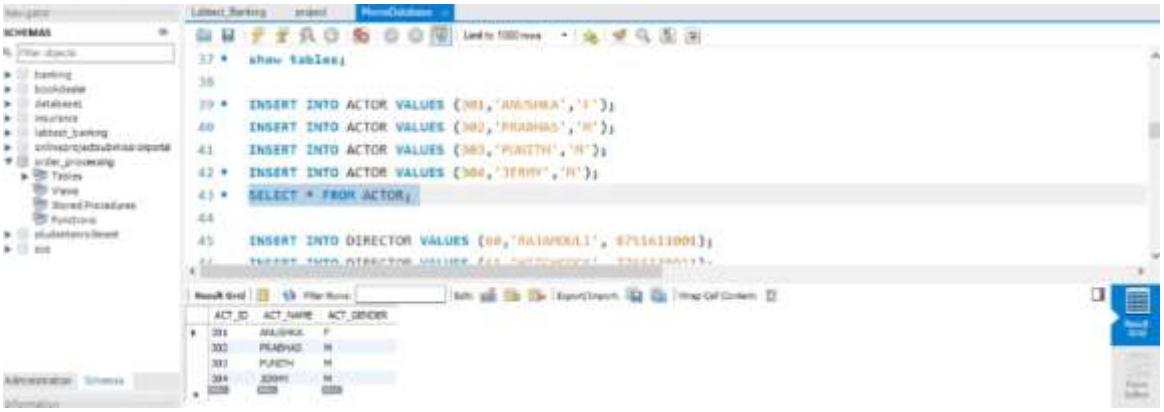
```
182   0R0£t 8Y tDV_IITLE j
183
t04   /' 5 . Update rating of al1 ivies directed by 'Steven Spi e tberg' to 5' /
10S • UPDATE RATING
106   SET REV_STARS='5'
107   WHERE MOV_ID = (SELECT MOV_ID FROM MOVIES
108   WHERE DIR_ID = (SELECT DIR_ID FROM DIRECTOR
109   UIJEREDDR H4IXEn'STEV£IJSPIELBERS'));
110 •
```

MOV_ID	REV_STARS
1001	4
1002	2
1003	5
1004	5

Local instance MySQL80 ×

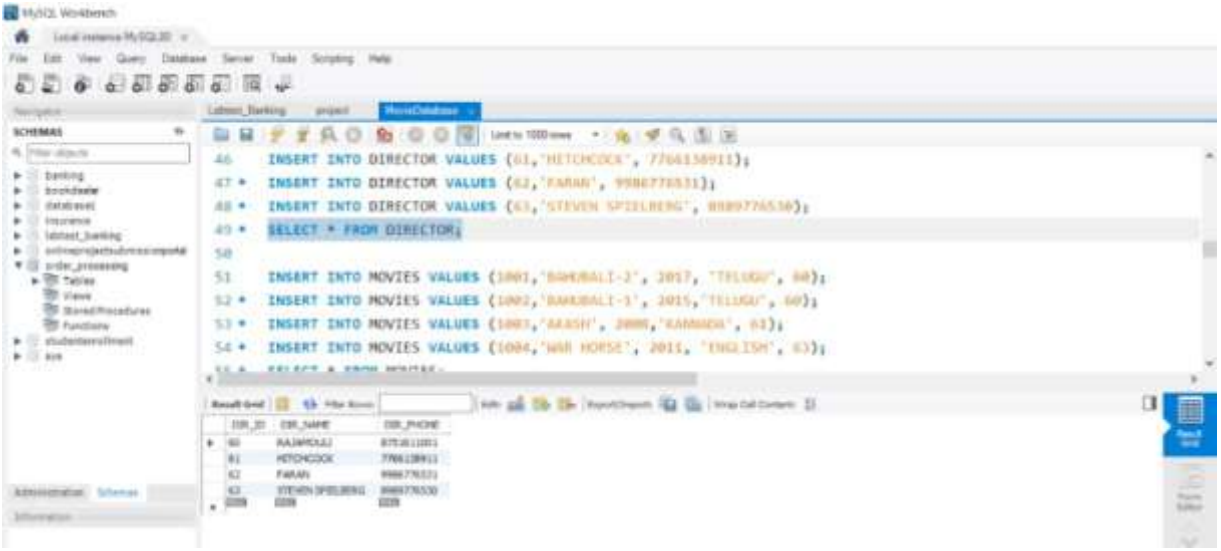


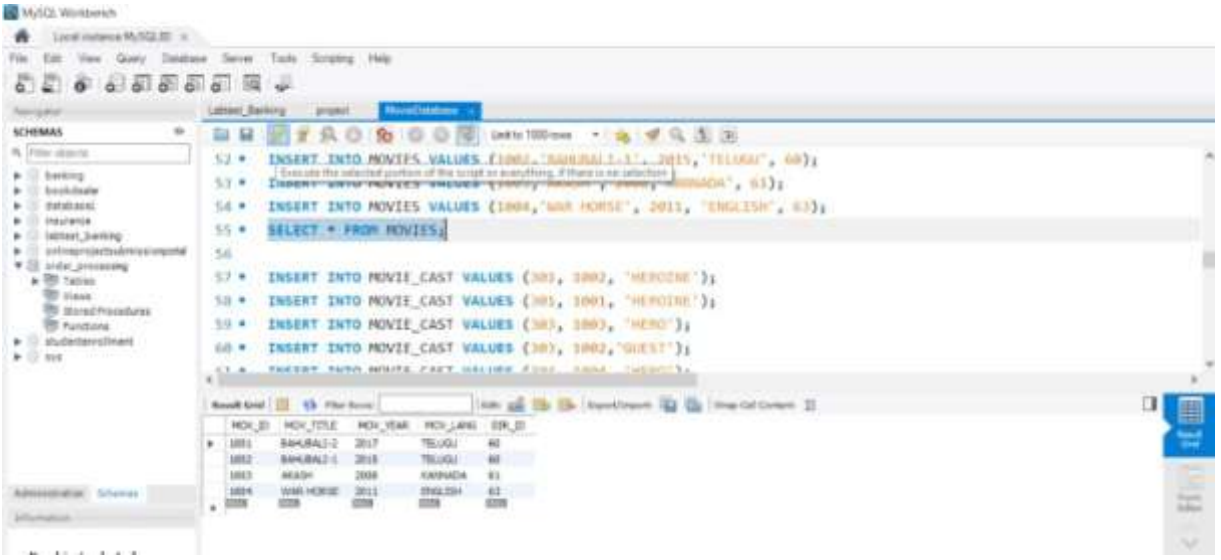
```
50   FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID));
51
52   CREATE TABLE RATING (
54   REV_STARS VARCHAR (25),
56   FOREIGN KEY (MOV_ID) REFERENCES MOVIES (MOV_ID));
```

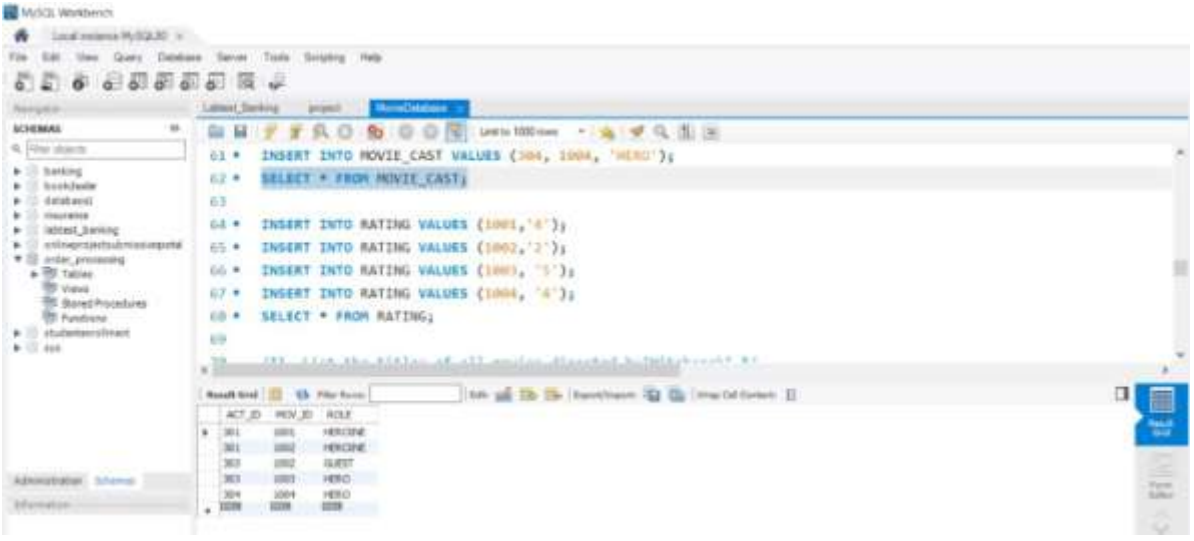


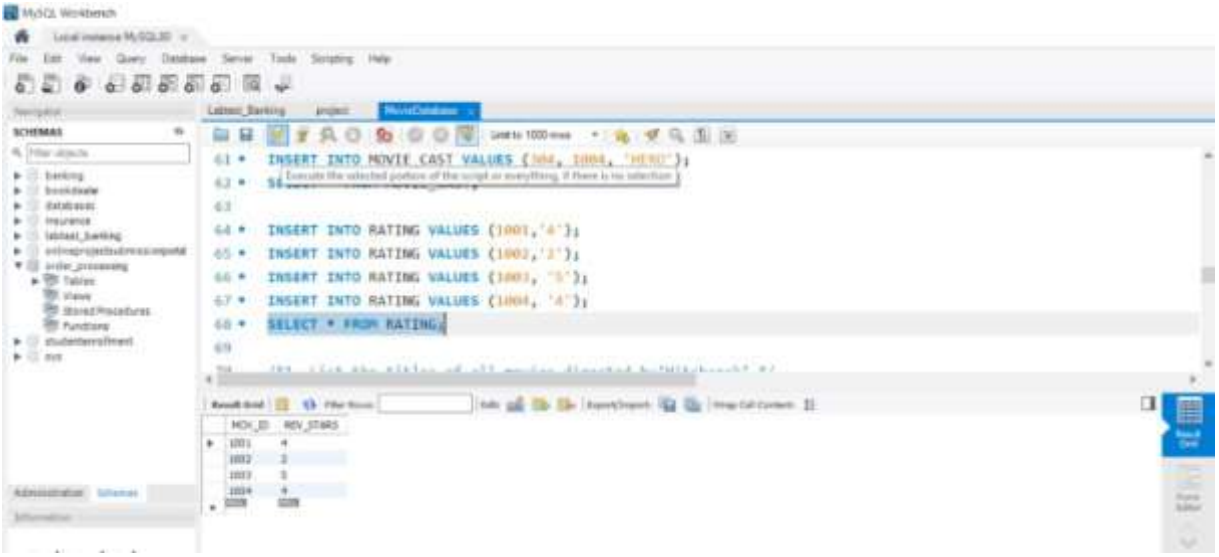
```
37 • show tables;
38
39 • INSERT INTO ACTOR VALUES (301,'ANUSHKA','F');
40   INSERT INTO ACTOR VALUES (302,'PRABHAS','M');
41   INSERT INTO ACTOR VALUES (303,'PUNITH','M');
42 • INSERT INTO ACTOR VALUES (304,'JERRY','M');
43 • SELECT * FROM ACTOR;
44
45   INSERT INTO DIRECTOR VALUES (60,'RAJAMOULI', 8751611001);
```

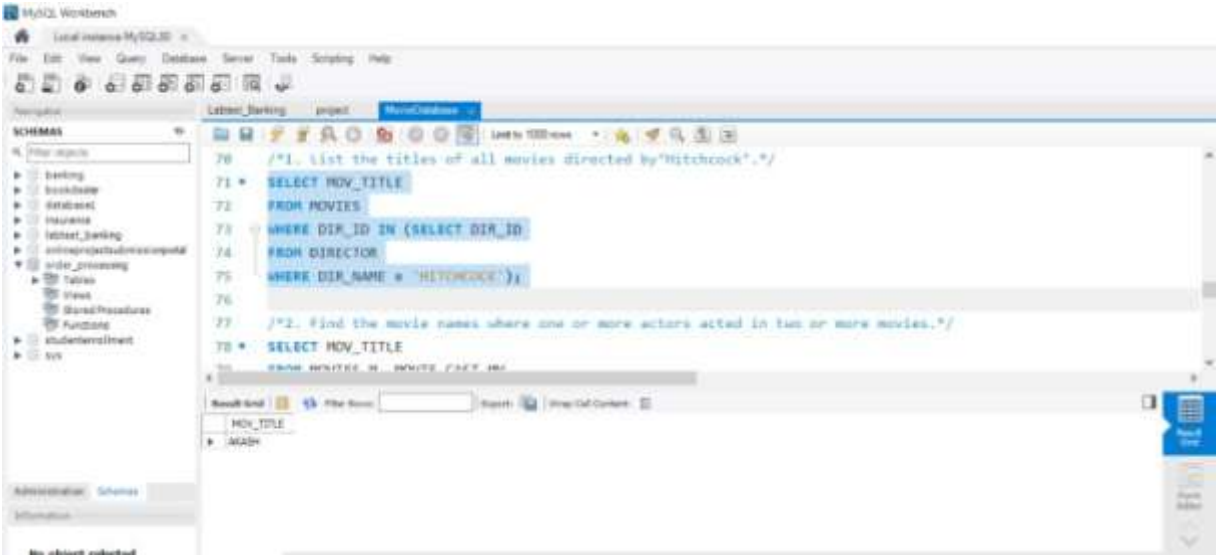
ACT_ID	ACT_NAME	ACT_GENDER
301	ANUSHKA	F
302	PRABHAS	M
303	PUNITH	M
304	JERRY	M

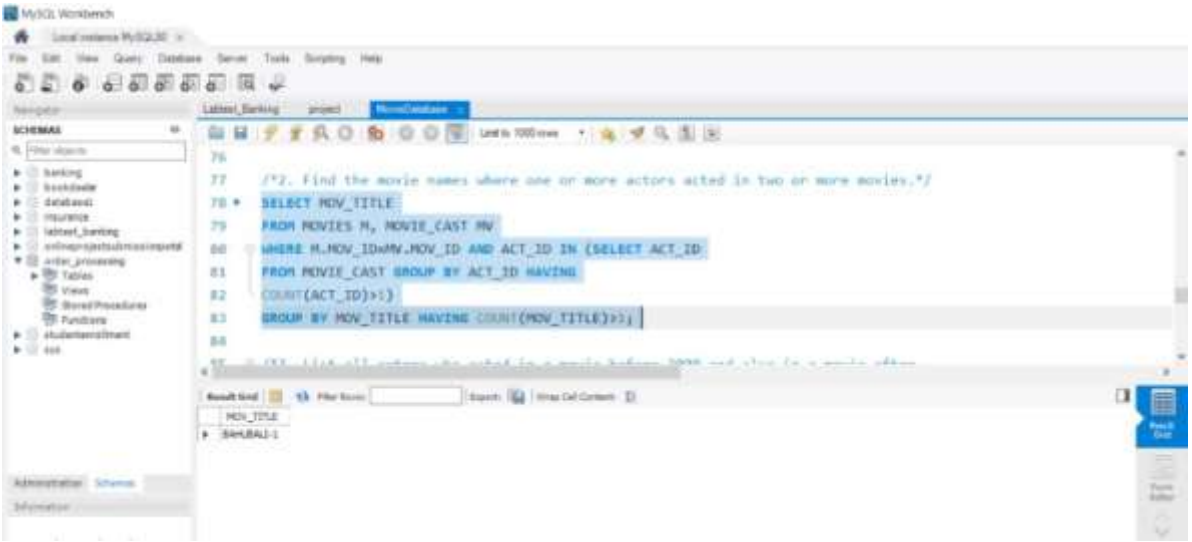

MySQL Workbench
INSERT INTO DIRECTOR VALUES (61,'HITCHCOCK', 7766138911);
INSERT INTO DIRECTOR VALUES (62,'FARAN', 9988776531);
INSERT INTO DIRECTOR VALUES (63,'STEVEN SPIELBERG', 8889776530);
SELECT * FROM DIRECTOR;
INSERT INTO MOVIES VALUES (1001,'BAHUBALI-2', 2017, 'TELUGU', 60);
INSERT INTO MOVIES VALUES (1002,'BAHUBALI-1', 2015,'TELUGU', 60);
INSERT INTO MOVIES VALUES (1003,'AKASH', 2008,'KANNADA', 61);
INSERT INTO MOVIES VALUES (1004,'WAR HORSE', 2011, 'ENGLISH', 63);


MySQL Workbench
INSERT INTO MOVIES VALUES (1002,'BAHUBALI-1', 2015,'TELUGU', 60);
INSERT INTO MOVIES VALUES (1004,'WAR HORSE', 2011, 'ENGLISH', 63);
SELECT * FROM MOVIES;
INSERT INTO MOVIE_CAST VALUES (303, 1002, 'HEROINE');
INSERT INTO MOVIE_CAST VALUES (301, 1001, 'HEROINE');
INSERT INTO MOVIE_CAST VALUES (303, 1003, 'HERO');
INSERT INTO MOVIE_CAST VALUES (307, 1002,'GUEST');


MySQL Workbench
INSERT INTO MOVIE_CAST VALUES (304, 1004, 'HERO');
SELECT * FROM MOVIE_CAST;
INSERT INTO RATING VALUES (1001,'4');
INSERT INTO RATING VALUES (1002,'2');
INSERT INTO RATING VALUES (1003, '5');
INSERT INTO RATING VALUES (1004, '4');
SELECT * FROM RATING;


MySQL Workbench
INSERT INTO MOVIE_CAST VALUES (304, 1004, 'HERO');
INSERT INTO RATING VALUES (1001,'4');
INSERT INTO RATING VALUES (1002,'2');
INSERT INTO RATING VALUES (1003, '5');
INSERT INTO RATING VALUES (1004, '4');
SELECT * FROM RATING;
MOV_ID
REV_STARS
1001
4
1002
2
1003
5
1004
4


MySQL Workbench
/*1. List the titles of all movies directed by'Hitchcock'.*/
SELECT MOV_TITLE
FROM MOVIES
WHERE DIR_ID IN (SELECT DIR_ID
FROM DIRECTOR
WHERE DIR_NAME = 'HITCHCOCK');
/*2. Find the movie names where one or more actors acted in two or more movies.*/
SELECT MOV_TITLE
MOV_TITLE
AKASH


MySQL Workbench
/*2. Find the movie names where one or more actors acted in two or more movies.*/
SELECT MOV_TITLE
FROM MOVIES M, MOVIE_CAST MV
WHERE M.MOV_ID=MV.MOV_ID AND ACT_ID IN (SELECT ACT_ID
FROM MOVIE_CAST GROUP BY ACT_ID HAVING
COUNT(ACT_ID)>1)
GROUP BY MOV_TITLE HAVING COUNT(MOV_TITLE)>1);
MOV_TITLE
BAHUBALI-1

PROGRAM 7: AIRLINE FLIGHT DATABASE

QUESTION:

Consider the following database that keeps track of airline flight information:
FLIGHTS (flno: integer, from: string, to: string, distance: integer, departs: time, arrives: time,
price: integer)
AIRCRAFT (aid: integer, aname: string, cruisingrange: integer)
CERTIFIED (eid: integer, aid: integer)
EMPLOYEE (eid: integer, ename: string, salary: integer)
Note that the Employees relation describes pilots and other kinds of employees as well; Every
pilot is certified
for some aircraft, and only pilots are certified to fly.
Write each of the following queries in SQL.
i. Find the names of aircraft such that all pilots certified to operate them have salaries more than
Rs.80,000.
ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum
cruising range of
the aircraft for which she or he is certified.
iii. Find the names of pilots whose salary is less than the price of the cheapest route from
Bengaluru to
Frankfurt.
iv. For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the
average salary of
all pilots certified for this aircraft.
v. Find the names of pilots certified for some Boeing aircraft.
vi. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.
vii. A customer wants to travel from Madison to New York with no more than two changes of
flight. List the
choice of departure times from Madison if the customer wants to arrive in New York by 6 p.m.
viii. Print the name and salary of every non-pilot whose salary is more than the average salary
for pilots.

PROGRAM CODE:

```
create database airline;
use airline;
CREATE TABLE flights(
flno Int,
`from` Varchar(20),
`to` Varchar(20),
distance INT,
departs time,
arrives time,
price Int,
PRIMARY KEY(flno) );
CREATE TABLE aircraft(
aid INT,
aname VARCHAR(20),
cruisingrange INT,
PRIMARY KEY (aid) );
CREATE TABLE employees(
eid INT,
ename Varchar(20),
salary INT,
PRIMARY KEY (eid) );
CREATE TABLE certified(
eid INT,
aid INT,
PRIMARY KEY (eid,aid),
FOREIGN KEY (eid) REFERENCES employees (eid),
FOREIGN KEY (aid) REFERENCES aircraft (aid) );
show tables;
INSERT INTO flights (flno,`from`,`to`,distance,departs,arrives,price) VALUES
(1,'Bangalore','Chennai',360,'08:45','10:00',10000),
(2,'Bangalore','Delhi',1700,'12:15','15:00',37000),
(3,'Bangalore','Kolkata',1500,'15:15','05:25',30000),
(4,'Mumbai','Delhi',1200,'10:30','12:30',28000),
(5,'Bangalore','New york',14000,'05:45','02:30',90000),
(6,'Delhi','Chicago',12000,'10:00','05:45',95000),
(7,'Bangalore','Frankfurt',15000,'12:00','06:30',98000),
(8,'Madison','New york',1500,'10:15','14:25',30000);
SELECT * FROM flights;
INSERT INTO aircraft (aid,aname,cruisingrange) values
(1,'Airbus 380',1000),
(2,'Boeing 737',4000),
(3,'Lockheed',5500),
(4,'Airbus A220',9500),
```

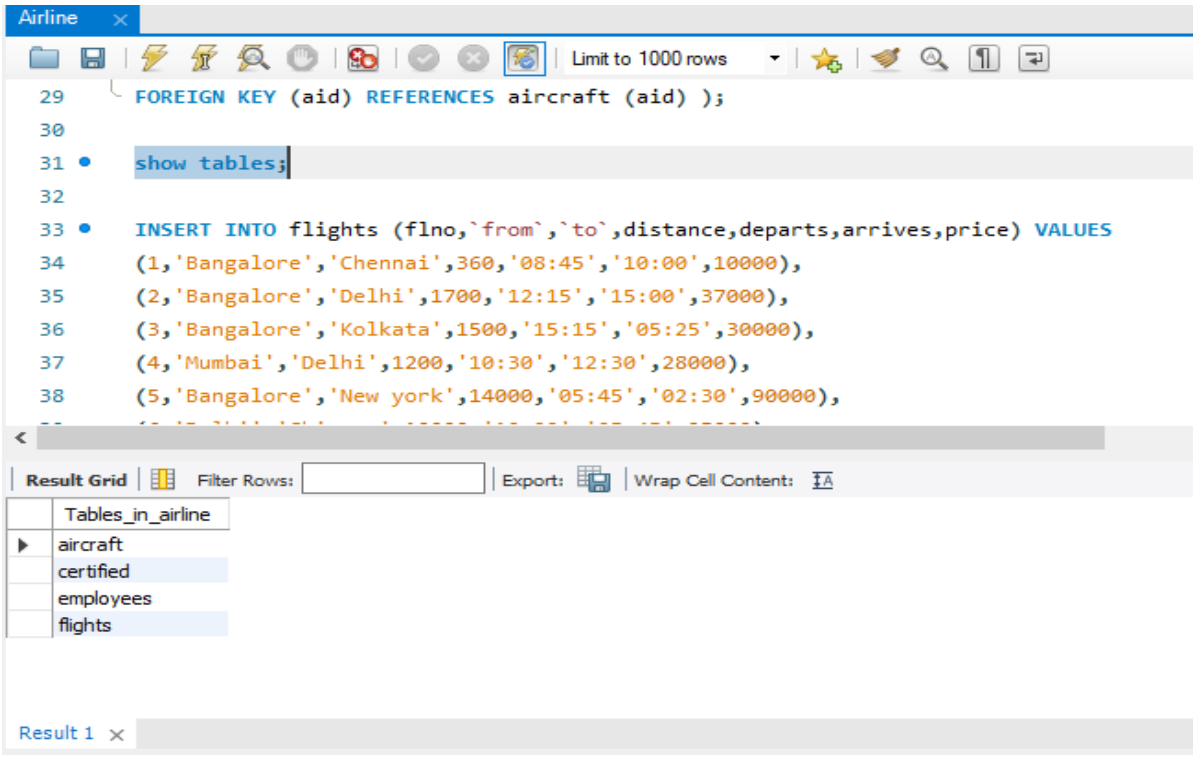
```
(5,'Boeing 747',800),
(6,'Douglas DC3',900);
SELECT * FROM aircraft;
INSERT INTO employees (eid,ename,salary) VALUES
(1,'Zoya',95000),
(2,'Akshay',65000),
(3,'Niveditha',70000),
(4,'Safan',45000),
(5,'Peter',95000),
(6,'Nayan',100000),
(7,'Ajay',50000);
SELECT * FROM employees;
INSERT INTO certified (eid,aid) VALUES
(1,1),
(1,3),
(1,4),
(5,4),
(5,3),
(1,2),
(2,6),
(2,5),
(4,5),
(6,4),
(6,3),
(3,6),
(3,2);
SELECT * FROM certified;
#i. Find the names of aircraft such that all pilots certified to operate them have salaries more
than Rs.80,000.
SELECT DISTINCT A.aname
FROM Aircraft A
WHERE A.Aid IN (SELECT C.aid
FROM Certified C, Employees E
WHERE C.eid = E.eid AND
NOT EXISTS ( SELECT *
FROM Employees E1
WHERE E1.eid = E.eid AND E1.salary < 80000 ));
#ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum
cruising range of the aircraft for which she or he is certified.
SELECT C.eid, MAX(A.cruisingrange)
FROM Certified C, Aircraft A
WHERE C.aid = A.aid
```

```
GROUP BY C.eid
HAVING COUNT(*) > 3;
#iii. Find the names of pilots whose salary is less than the price of the cheapest route from
Bengaluru toFrankfurt.
SELECT DISTINCT e.ename
FROM employees e
WHERE e.salary<
(SELECT MIN(f.price)
FROM flights f
WHERE f.from='Bangalore' AND f.to='Frankfurt');
#iv. For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the
average salary of all pilots certified for this aircraft.
SELECT a.aid,a.aname,AVG(e.salary)
FROM aircraft a,certified c,employees e
WHERE a.aid=c.aid
AND c.eid=e.eid
AND a.cruisingrange>1000
GROUP BY a.aid,a.aname;
#v. Find the names of pilots certified for some Boeing aircraft.
SELECT distinct e.ename
FROM employees e,aircraft a,certified c
WHERE e.eid=c.eid AND c.aid=a.aid AND a.aname like 'Boeing%';
#vi. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.
SELECT a.aid
FROM aircraft a
WHERE a.cruisingrange>
(SELECT  MIN(f.distance)
FROM flights f
WHERE f.from='Bangalore' AND f.to='Delhi');
#vii. A customer wants to travel from Madison to New York with no more than two changes of
flight. List the choice of departure times from Madison if the customer wants to arrive in New
York by 6 p.m.
SELECT F.departs
FROM Flights F WHERE F.flno IN ( SELECT F0.flno
FROM Flights F0
WHERE F0.from = 'Madison' AND F0.to = 'New york' AND F0.arrives < '18:00' );
#viii. Print the name and salary of every non-pilot whose salary is more than the average salary
for pilots.0
```

```
SELECT E.ename, E.salary
FROM Employees E
WHERE E.eid NOT IN ( SELECT DISTINCT C.eid
FROM Certified C )
AND E.salary > ( SELECT AVG (E1.salary)
FROM Employees E1
WHERE E1.eid IN
( SELECT DISTINCT C1.eid
FROM Certified C1 ) );
```

OUTPUT:

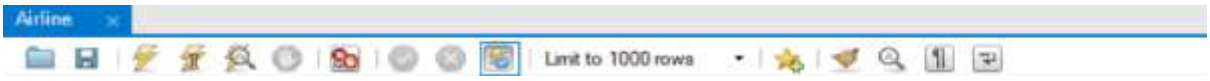
OUTPUT SCREENSHOTS:




```
-2 °    SELECT ” FRODO flights;
```

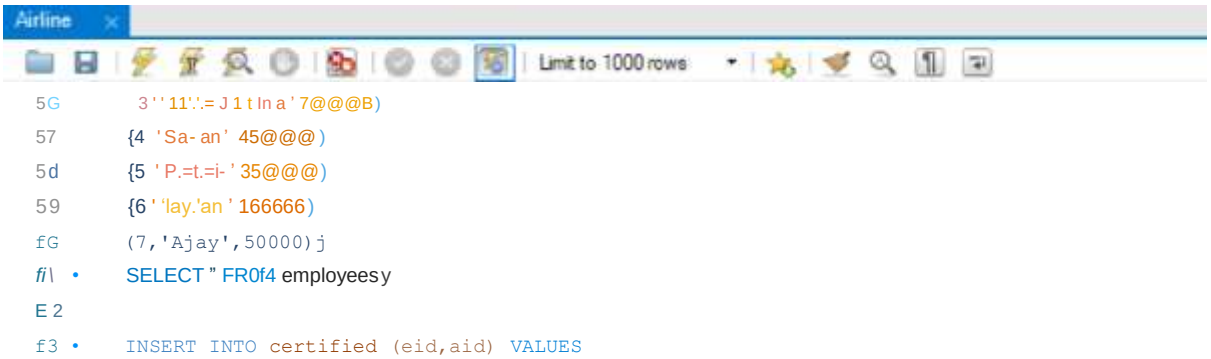
Result Grid '# . I F"ter Rz-. >: Eo I: Exg-z-rt. m g-z-rt: @ '.".'r- g- ?e ? z-rter t :

	fino	from	to	distance	departs	arrives	price
¥	1	Bangalore	Chennai	360	08:45:00	10:00:00	10000
	2	Bangalore	Delhi	1700	12:15:00	15:00:00	37000
	3	Bangalore	Kolkata	1500	15:15:00	05:25:00	30000
	4	Mumbai	Delhi	1Z00	10:50:00	12:50:00	Z8000
	5	Bangalore	Ne'x' york	14000	05:45:00	02:30:00	90000
	6	Delhi	Chicago	1Z000	10:00:00	05:45:00	95000
	7	Bangalore	Frankfurt	15000	12:00:00	06:30:00	98000
	8	Madison	Ne'w' york	1500	10:15:00	14:25:00	30000




```
                (3,'Lockheed',5500),
                (4,'Airbus A220',9500),
49              (5,'Boeing 747',800),
50              (6,'Douglas DC3',900);
ñi °    SELECT * PROD aircraft_I

 a ^    lr.sEnT I t.TC' e z ploy'ees  e id , e n arre , s ala n\' ) ' '" .LLES
54      (1,'Zoya',95000),
```

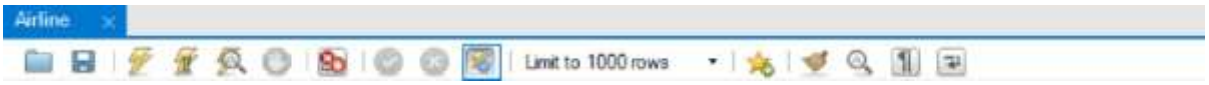
	aid	aname	cruisingrange
p	1	Airbus380	1000
	2	Boeing 7J7	4000
		Lockheed	5500
	4	Airbus A2Z0	9500
	5	Boeing 747	800
	6	Douglas DC3	900



Result Grid @ 't\ l"iher Rev.s: _____ Edit: Export•'J mport: '."Yrap Céll Cont=ut: @

	eid	ename	salary
¥'	1	Zoya	95000
	Z	Akshay	65000
	3	Niveditha	70000
	1	Safan	15000
	5	Peter	95000
	6	Nayan	100000
	7	Ajay	50000

znlplo›'zzsH x



Result Grid \ Eiker Roe.si _____ Editi mport i '.Wrap Cell Cont t i @

	eid	aid
D	1	1
	1	Z
	5	2
	5	3
	6	]
	1	9
	6	9
	2	6
	3	6

cE rtifiE d 5 x

Airline

```
80    -b1. F1nd the nanes of aInc naft such that all pilots ce stifled to ope rate then have sala ries more than Rs . 80, 000.
81 •  SELECT DISTINCT A. anare
82    rRow Aircraft A
83    WHERE A.Aid IN (SELECT C.aid
84    FROM Certified C, Employees E
B5    WHERE C.eid =E.eid AND
86    NOT EXISTS ( SELECT
87    PROM Employees El
88    WHERE E1.eid = E.eid AND EI.salary < 80000 ));
89
9B    iii. For each pilot who is certified for more than three aircrafts. find the eid and the maximum cruising ranRe of
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

aname
Airbus 380
Boeing 737
Lockheed
Airbus A220

Aircraft 6 x

Airline

Limit to 1000 rows

```
89
98    #ii. For each pilot who is certified for more thau three aircrafts. fiud the eid and
91 •  SELECT C.eid, MAX(A.cruisingramge}
92    FRC#M Certified C, Aircraft  A
93    WHERE C.aid = A.aid
94    GROUP BY C.eid
95    HAVIfOG COUNT(*) > 3;
96
97    4t111. F1nd I he nanes of p11ot s \Vh ose s a La cy 1s 1e s s I han I he pn1c e of I he che ape sI nc
9B •  SELE[T DISTINCT e.emame
99    PRCVfl emplovees e
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

eid	MAX(A.cruisingramge)
1	9500

Result 7 x

```
95    HAVING COUIT T (") › 3 j
9G
97    4ii  i . 'ind +he  nan e 3  cf  pit  cts   'hc3 e zala    iz less than the p i ce c f the cheape 3t cute f c r Bengal u -u +c' ankf u -I .
95 •  SELECT DISTINCT e. ename
99    FR0Jt employees  e
LEE   MHERE e .sa1aryc
\G1   (SELECT i"›II4 (f. price )
IG2   FR0Jt flight s f
11?3  MHERE f. f-rcxn= ' B anga More ' AND f. to= ' Frank-furt ');)

105   #Iv. For all aircraft with cruising range over 1000 Kms, find the name of the aircraft and the average salary of all
```

Result Grid

ename
Zoya
Akshay
Nivedtha
Safan
Peter
Ajay

Employ'EEs 8 x

Airline

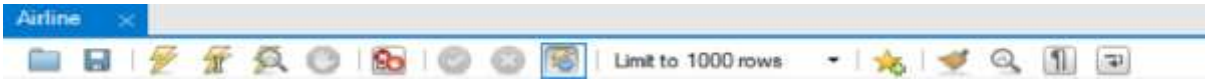
Limit to 1000 rows

```
1 it 5      t1 • . ' o - a11  a1 - c - al *    * h c - u 1 1 n g - an ge o • e - 1 C C C K n z , f 1 nd * he n ane
1I2E  •     5 EL EC T a . ald t a . a n amet AVG e . s a la r•y)
1G7         FRDf4  a1r-c said  a  ceW171ed  d  empl oy ees  e
1Gd         THE RE a . a id =c . aid
1G9         AND c . eid =e . eid
IT D        ANO a . o r-u is in gr a n ge at B B B
111         GROUP  BY  a. ald   a . a ◦_ .
112
113         4'«. 'ind the nares ci pilcts ce'ti{ied {c' scre Bceing  ai'c'a+t.
114  •      5ELE[f distinct  e.ename
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

aid	aname	AVG(e.salary)
2	Boeing 737	82500.0000
S	Loddneed	96666.6667
4	Airbus AZZ0	96666.6667

RE sult 9 x




```
113     €v. 'ind the names cf pilcts ce'tified fc' scme Bceing ai'c'aft.
114 •   SELECT distinct e.ename
115     PRoM employees e aircraft a certified c
11E     âdHERE e. e1d =c.e1d AT c.ald =a.ald AT a.anarae 11ke 'Boe1ng@
117
118     lvi . 'ind the aids cf all ai'c 'aft that cam be used cm 'cutes f'cm Bemgalu 'u
119 •   !5ELECf a.aid
120     IFROffl aircraft a
121     lQHERE a.ccuisimgoange>
122     (SELECT i-'.IN(T. d1st ance)
123     FRO7ñ f1:xeht s I
```

Result Grid // Aker Row:_____I ^: W I Wrap CéJI Content:

ename

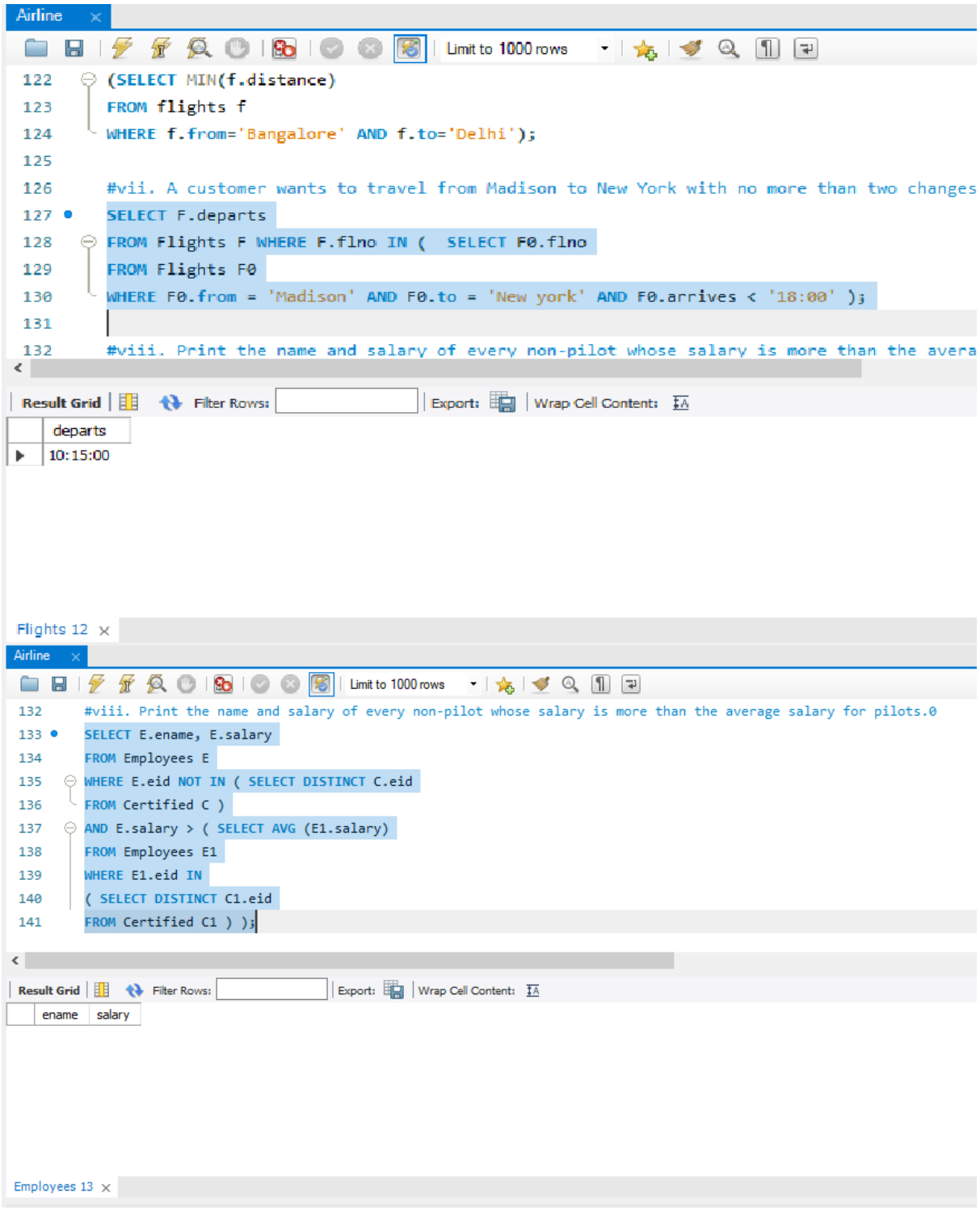
¥• Zoya

Niveditha

Res ult 10 x


```
1 1f    \dHERE e . e1d=c . e1d AND c . a1d =a . a1d AND a . aname I1ke ' 6a .=1 ng1 ';
117
118     4vi. 'ind the aids ct all ai'c'aft that cau be used cr 'cutes f'cm tc Nen' Delhi
119 •   SELE CT a . aid
120     FROM aircraft a
121     1dHERE a . c PM is in grange>
122     (SELECT i'1If4(f. dEst ance)
123     FRON f1igh-Is f
124     WHERE I.I-t*<xn=' Banga1or e' AT I.I:a=' De th i ');|
125
12f     4vii. $ custcme' warts tc t'avel f'cm 'ladiscu tc New Yc'k with rc mc'e thar tivc changes ct #
```

aid



PROGRAM-8:SUPPLIERS DATABASE

QUESTION :

CONSIDER THE FOLLOWING SCHEMA:
SUPPLIERS (SID: INTEGER, SNAME: STRING,
ADDRESS: STRING) PARTS (PID: INTEGER, PNAME:
STRING, COLOR: STRING) CATALOG (SID: INTEGER,
PID: INTEGER, COST: REAL)

THE CATALOG RELATION LISTS THE PRICES CHARGED FOR PARTS BY SUPPLIERS.
WRITE THE FOLLOWING QUERIES IN SQL:

I. FIND THE PNAMES OF PARTS FOR WHICH THERE IS SOME SUPPLIER.
II. FIND THE SNAMES OF SUPPLIERS WHO SUPPLY EVERY PART.
III. FIND THE SNAMES OF SUPPLIERS WHO SUPPLY EVERY RED PART.
IV. FIND THE PNAMES OF PARTS SUPPLIED BY ACME WIDGET SUPPLIERS AND BY NO ONE ELSE.
V. FIND THE SIDS OF SUPPLIERS WHO CHARGE MORE FOR SOME PART THAN THE AVERAGE COST OF THAT PART (AVERAGED OVER ALL THE SUPPLIERS WHO SUPPLY THAT PART).
VI. FOR EACH PART, FIND THE SNAME OF THE SUPPLIER WHO CHARGES THE MOST FOR THAT PART.
VII. FIND THE SIDS OF SUPPLIERS WHO SUPPLY ONLY RED PARTS.

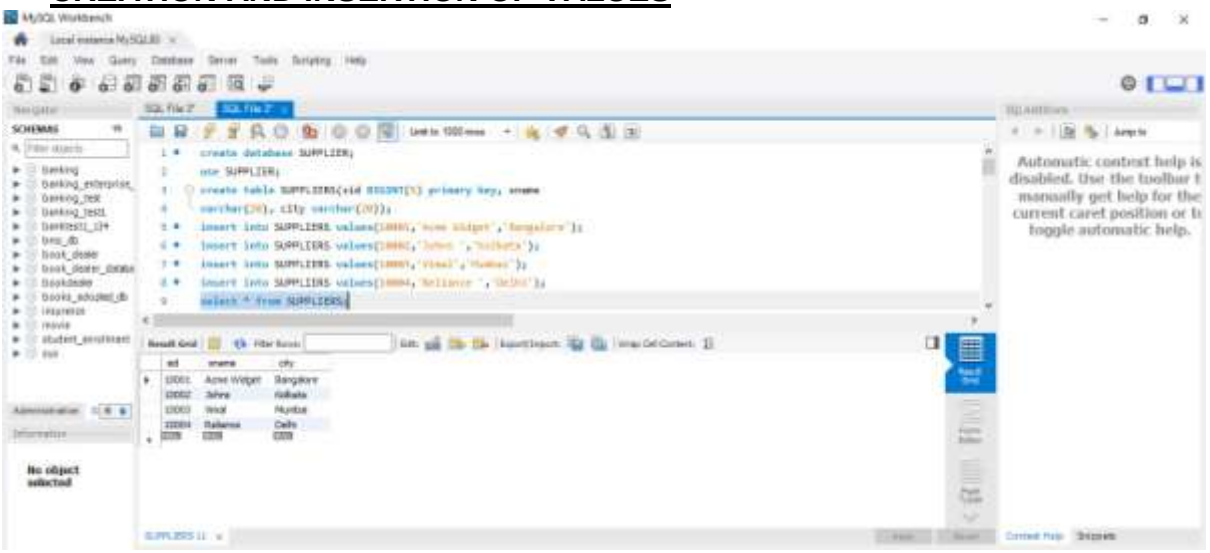
PROGRAM CODE :

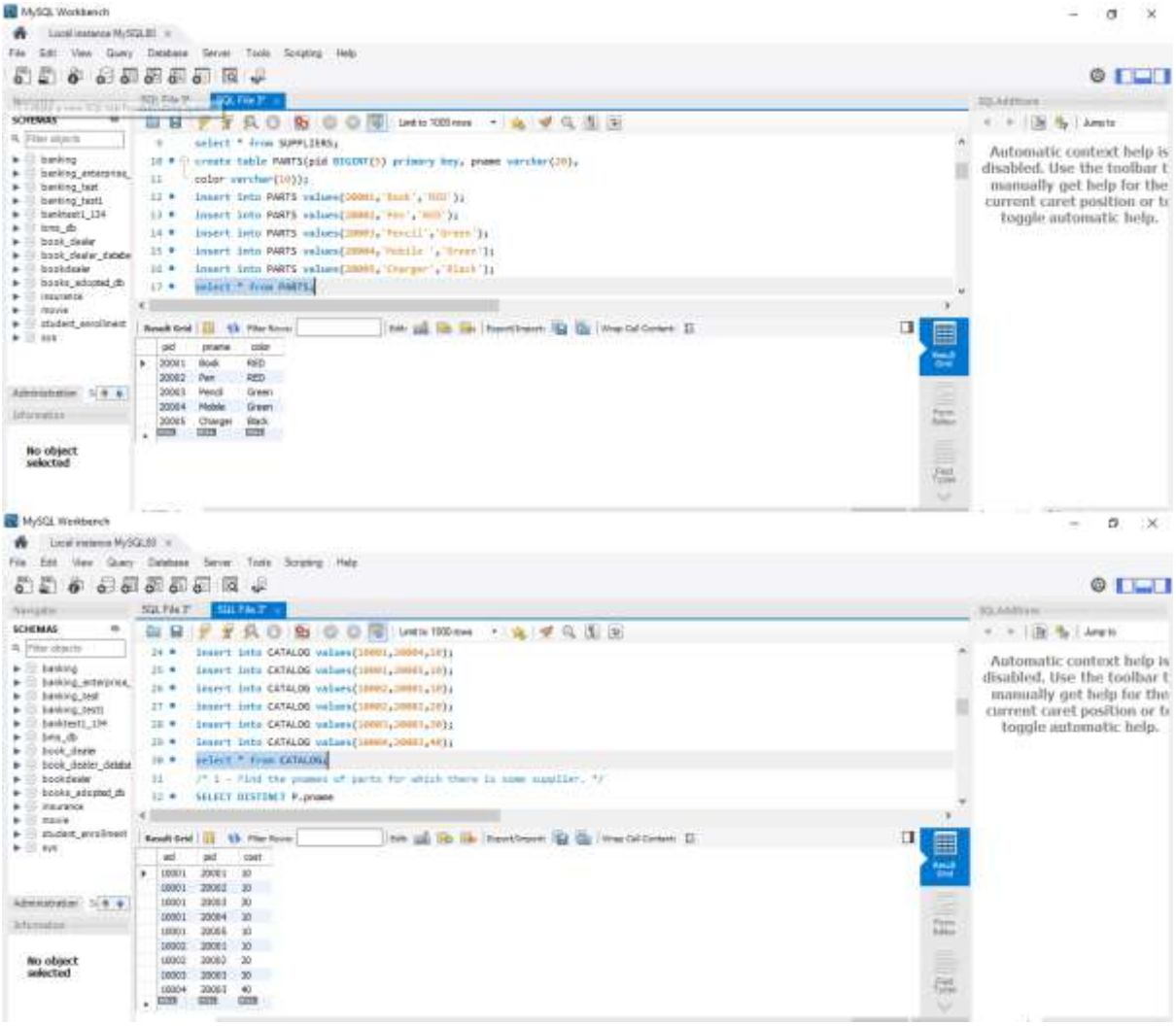
```
CREATE DATABASE SUPPLIER;
USE SUPPLIER;
CREATE TABLE SUPPLIERS(SID BIGINT(5) PRIMARY KEY, SNAME
VARCHAR(20), CITY VARCHAR(20));
INSERT INTO SUPPLIERS VALUES(10001,'ACME WIDGET','BANGALORE');
INSERT INTO SUPPLIERS VALUES(10002,'JOHNS ','KOLKATA');
INSERT INTO SUPPLIERS VALUES(10003,'VIMAL','MUMBAI');
INSERT INTO SUPPLIERS VALUES(10004,'RELIANCE ','DELHI');
SELECT * FROM SUPPLIERS;
CREATE TABLE PARTS(PID BIGINT(5) PRIMARY KEY, PNAME VARCHAR(20),
COLOR VARCHAR(10));
INSERT INTO PARTS VALUES(20001,'BOOK','RED');
INSERT INTO PARTS VALUES(20002,'PEN','RED');
INSERT INTO PARTS VALUES(20003,'PENCIL','GREEN');
INSERT INTO PARTS VALUES(20004,'MOBILE ','GREEN');
INSERT INTO PARTS VALUES(20005,'CHARGER','BLACK');
SELECT * FROM PARTS;
CREATE TABLE CATALOG(SID BIGINT(5), PID BIGINT(5), FOREIGN KEY(SID)
REFERENCES SUPPLIERS(SID), FOREIGN KEY(PID) REFERENCES PARTS(PID),
COST FLOAT(6), PRIMARY KEY(SID, PID));
INSERT INTO CATALOG VALUES(10001,20001,10);
INSERT INTO CATALOG VALUES(10001,20002,10);
INSERT INTO CATALOG VALUES(10001,20003,30);
INSERT INTO CATALOG VALUES(10001,20004,10);
INSERT INTO CATALOG VALUES(10001,20005,10);
INSERT INTO CATALOG VALUES(10002,20001,10);
INSERT INTO CATALOG VALUES(10002,20002,20);
INSERT INTO CATALOG VALUES(10003,20003,30);
INSERT INTO CATALOG VALUES(10004,20003,40);
SELECT * FROM CATALOG;
/* 1 - FIND THE PNAMES OF PARTS FOR WHICH THERE IS SOME SUPPLIER. */
SELECT DISTINCT P.PNAME
 FROM PARTS P, CATALOG C
 WHERE P.PID = C.PID;

/* FIND THE SNAMES OF SUPPLIERS WHO SUPPLY EVERY PART */
SELECT S.SNAME FROM SUPPLIERS S WHERE NOT EXISTS (SELECT P.PID FROM
PARTS P WHERE NOT EXISTS (SELECT C.SID FROM CATALOG C WHERE C.SID =
S.SID AND C.PID = P.PID));

/* FIND THE SNAMES OF SUPPLIERS WHO SUPPLY EVERY RED PART. */
SELECT S.SNAME FROM SUPPLIERS S WHERE NOT EXISTS (SELECT P.PID FROM
PARTS P WHERE P.COLOR = 'RED' AND (NOT EXISTS (SELECT C.SID FROM
CATALOG C WHERE C.SID = S.SID AND C.PID = P.PID)));

/* FIND THE PNAMES OF PARTS SUPPLIED BY ACME WIDGET SUPPLIERS AND BY NO ONE ELSE */
SELECT P.PNAME FROM PARTS P, CATALOG C, SUPPLIERS S WHERE P.PID
= C.PID AND C.SID = S.SID AND S.SNAME = 'ACME WIDGET' AND NOT EXISTS
```

```
(SELECT * FROM CATALOG C1, SUPPLIERS S1 WHERE P.PID = C1.PID AND
C1.SID = S1.SID AND S1.SNAME <> 'ACME WIDGET');

/* FIND THE SIDS OF SUPPLIERS WHO CHARGE MORE FOR SOME PART THAN THE AVERAGE COST OF
THAT PART (AVERAGED OVER
ALL THE SUPPLIERS WHO SUPPLY THAT PART).
 */
SELECT DISTINCT C.SID FROM CATALOG C
WHERE C.COST > ( SELECT AVG (C1.COST)
FROM CATALOG C1
WHERE C1.PID = C.PID );

/* FOR EACH PART, FIND THE SNAME OF THE SUPPLIER WHO CHARGES THE MOST FOR THAT PART.*/
SELECT P.PID, S.SNAME
FROM PARTS P, SUPPLIERS S, CATALOG C
WHERE C.PID = P.PID
AND C.SID = S.SID
AND C.COST = (SELECT MAX(C1.COST)
FROM CATALOG C1
WHERE C1.PID = P.PID);

/* FIND THE SIDS OF SUPPLIERS WHO SUPPLY ONLY RED PARTS.*/
SELECT DISTINCT C.SID
FROM CATALOG C
WHERE NOT EXISTS ( SELECT *
FROM PARTS P
WHERE P.PID = C.PID AND P.COLOR <> 'RED' );
```

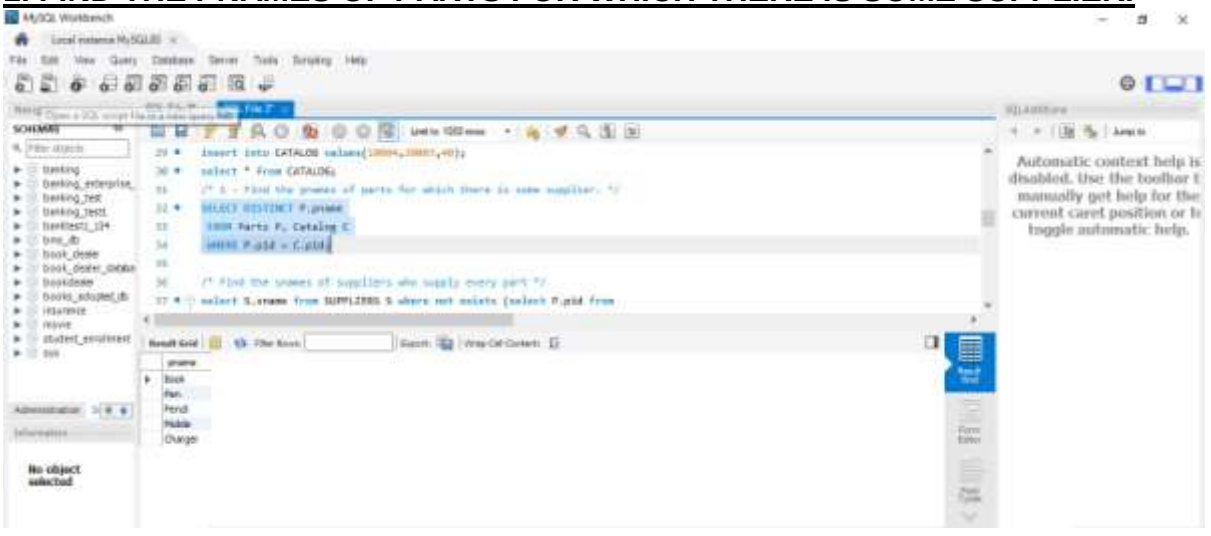
SCREENSHOTS OF THE PROGRAM OUTPUT :

- **CREATION AND INSERTION OF VALUES**

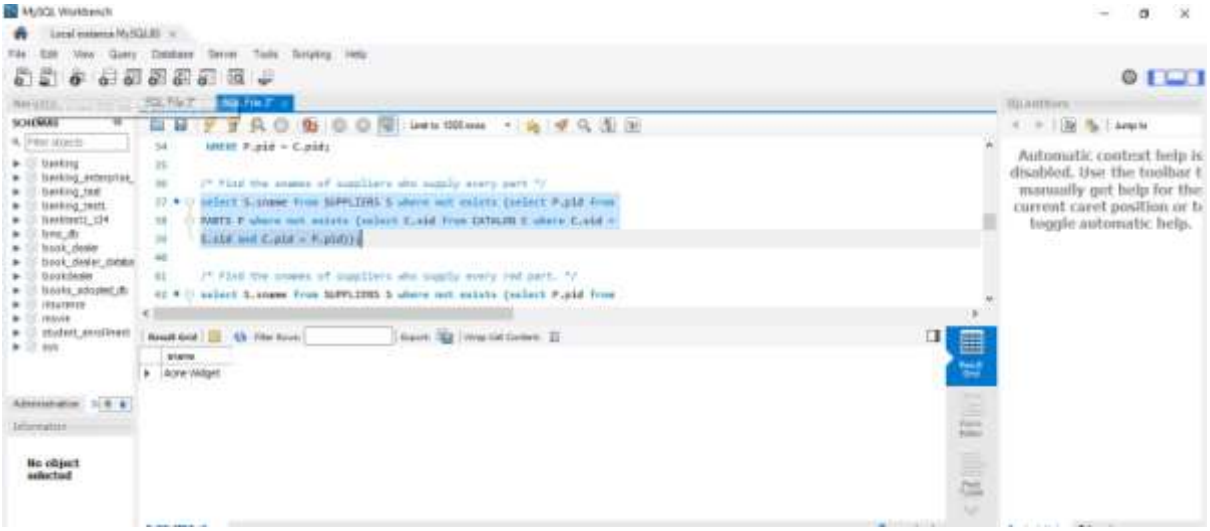




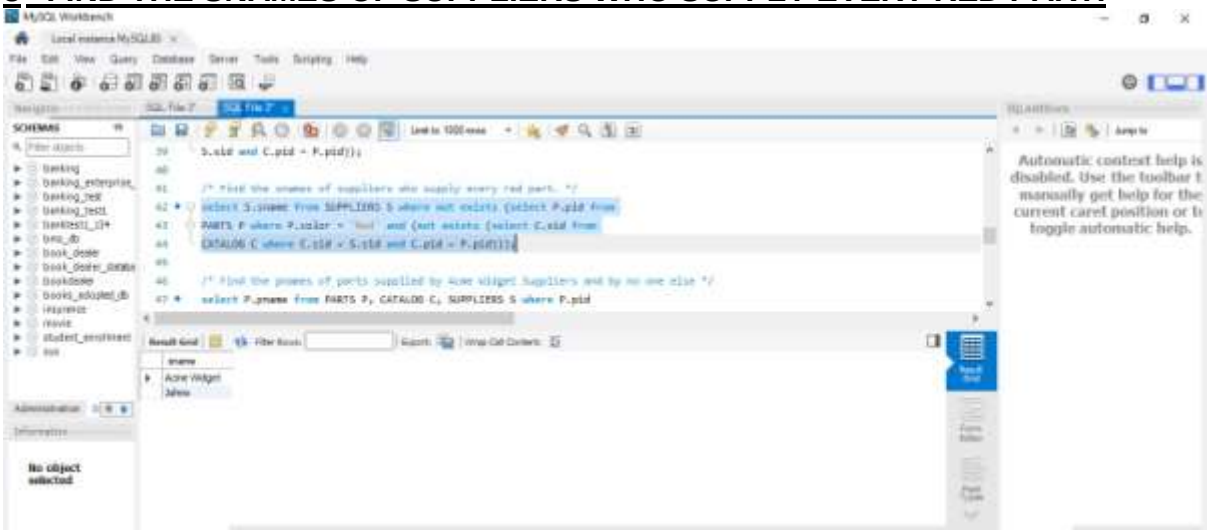
1. FIND THE PNAMES OF PARTS FOR WHICH THERE IS SOME SUPPLIER.



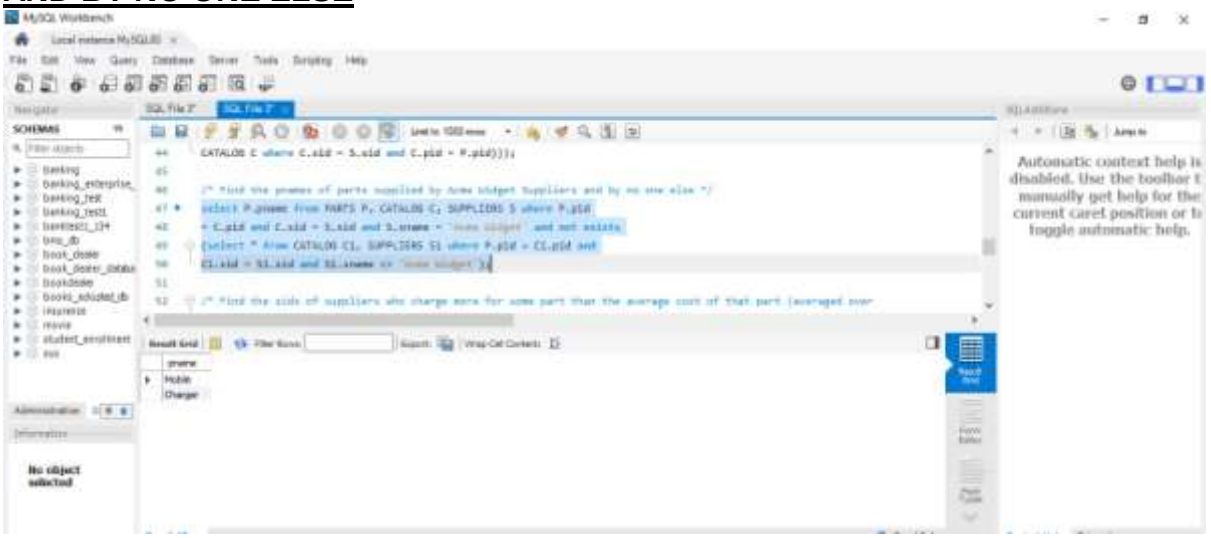
2 - FIND THE SNAMES OF SUPPLIERS WHO SUPPLY EVERY PART.



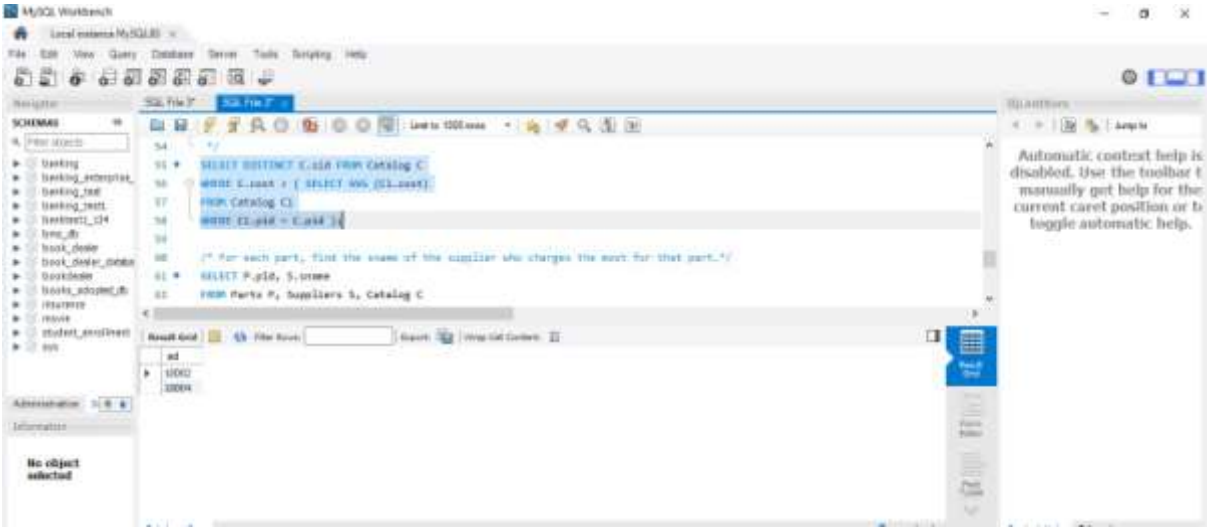
3 - FIND THE SNAMES OF SUPPLIERS WHO SUPPLY EVERY RED PART.



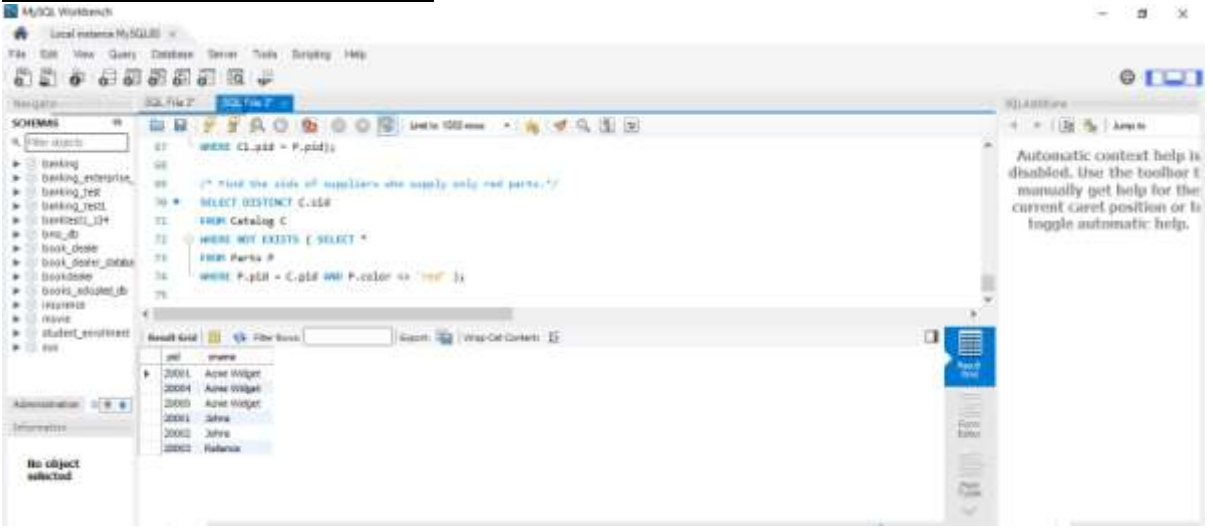
4 - FIND THE PNAMES OF PARTS SUPPLIED BY ACME WIDGET SUPPLIERS AND BY NO ONE ELSE



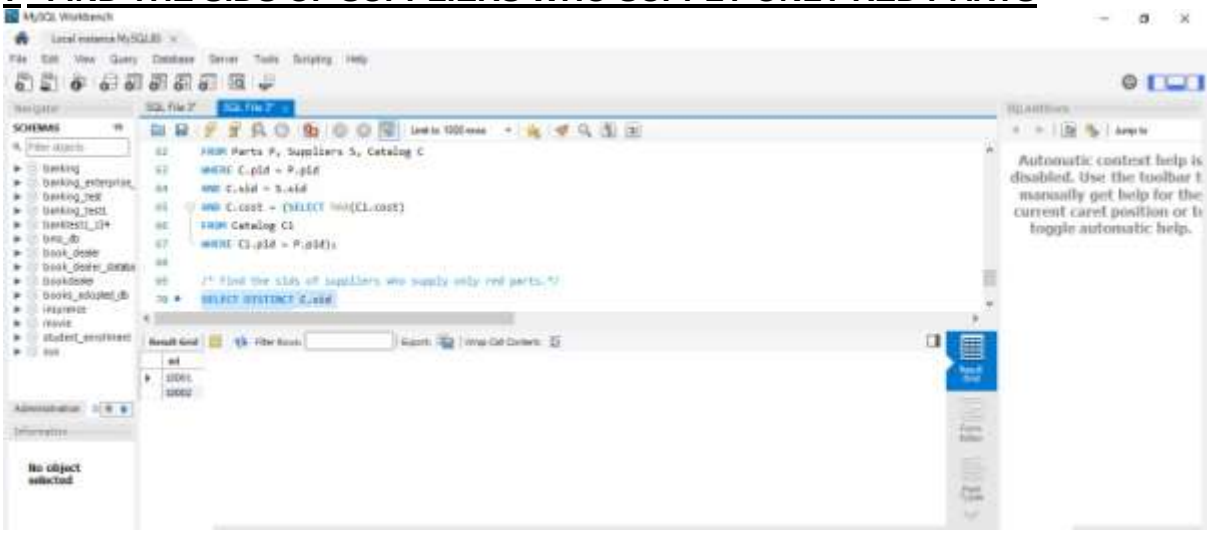
5 - FIND THE SIDS OF SUPPLIERS WHO CHARGE MORE FOR SOME PART THAN THE AVERAGE COST OF THAT PART (AVERAGED OVER ALL THE SUPPLIERS WHO SUPPLY THAT PART).



6 - FOR EACH PART, FIND THE SNAME OF THE SUPPLIER WHO CHARGES THE MOST FOR THAT PART



7 - FIND THE SIDS OF SUPPLIERS WHO SUPPLY ONLY RED PARTS



PROGRAM-9:STUDENT-FACULTY DATABASE

QUESTION :

CONSIDER THE FOLLOWING DATABASE FOR STUDENT ENROLMENT FOR COURSE:
STUDENT (SNUM: INTEGER, SNAME: STRING, MAJOR: STRING, LEVEL: STRING, AGE: INTEGER) CLASS (NAME: STRING, MEETS AT: TIME, ROOM: STRING, FID: INTEGER)
ENROLLED (SNUM: INTEGER, CNAME: STRING)
FACULTY (FID: INTEGER, FNAME: STRING, DEPTID: INTEGER)
THE MEANING OF THESE RELATIONS IS STRAIGHTFORWARD; FOR EXAMPLE, ENROLLED HAS ONE RECORD PER STUDENT-CLASS PAIR SUCH THAT THE STUDENT IS ENROLLED IN THE CLASS. LEVEL IS A TWO CHARACTER CODE WITH 4 DIFFERENT VALUES (EXAMPLE:
JUNIOR: JR ETC)
WRITE THE FOLLOWING QUERIES IN SQL. NO DUPLICATES SHOULD BE PRINTED IN ANY OF THE ANSWERS.

I. FIND THE NAMES OF ALL JUNIORS (LEVEL = JR) WHO ARE ENROLLED IN A CLASS TAUGHT BY
II. FIND THE NAMES OF ALL CLASSES THAT EITHER MEET IN ROOM R128 OR HAVE FIVE OR MORE STUDENTS ENROLLED.
III. FIND THE NAMES OF ALL STUDENTS WHO ARE ENROLLED IN TWO CLASSES THAT MEET AT THE SAME TIME.
IV. FIND THE NAMES OF FACULTY MEMBERS WHO TEACH IN EVERY ROOM IN WHICH SOME CLASS IS TAUGHT.
V. FIND THE NAMES OF FACULTY MEMBERS FOR WHOM THE COMBINED ENROLMENT OF THE COURSES THAT THEY TEACH IS LESS THAN FIVE.
VI. FIND THE NAMES OF STUDENTS WHO ARE NOT ENROLLED IN ANY CLASS.
VII. FOR EACH AGE VALUE THAT APPEARS IN STUDENTS, FIND THE LEVEL VALUE THAT APPEARS MOST OFTEN. FOR EXAMPLE, IF THERE ARE MORE FR LEVEL STUDENTS AGED 18 THAN SR, JR, OR SO STUDENTS AGED 18, YOU SHOULD PRINT THE PAIR (18, FR).

PROGRAM CODE :

```
CREATE DATABASE
STUDENT_FACULTY; USE
STUDENT_FACULTY;
CREATE TABLE
STUDENT( SNUM INT,
SNAME
VARCHAR(10),
MAJOR
VARCHAR(2), LVL
VARCHAR(2),
AGE INT, PRIMARY KEY(SNUM));

CREATE TABLE FACULTY(
FID INT,FNAME
VARCHAR(20), DEPTID
INT,
 PRIMARY KEY(FID));

CREATE TABLE
CLASS( CNAME
VARCHAR(20),
```

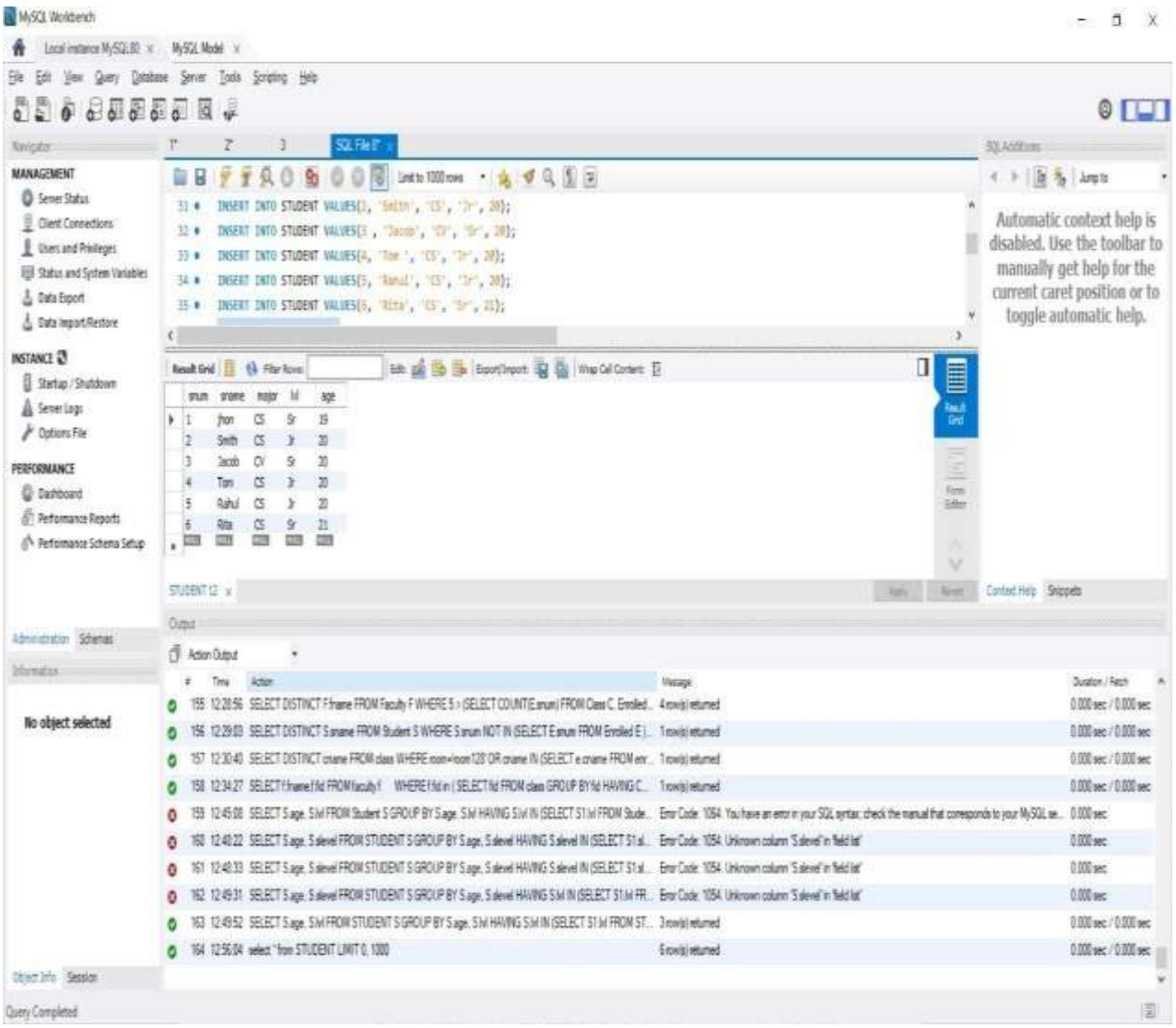
```
FOREIGN KEY(FID) REFERENCES FACULTY(FID));

CREATE TABLE
ENROLLED( SNUM
INT,
CNAME VARCHAR(20),
PRIMARY
KEY(SNUM,CNAME),
FOREIGN KEY(SNUM) REFERENCES
STUDENT(SNUM), FOREIGN KEY(CNAME)
REFERENCES CLASS(CNAME));

INSERT INTO STUDENT VALUES(1, 'JHON', 'CS',
'SR', 19); INSERT INTO STUDENT VALUES(2,
'SMITH', 'CS', 'JR', 20); INSERT INTO STUDENT
VALUES(3 , 'JACOB', 'CV', 'SR', 20); INSERT INTO
STUDENT VALUES(4, 'TOM ', 'CS', 'JR', 20); INSERT
INTO STUDENT VALUES(5, 'RAHUL', 'CS', 'JR', 20);
INSERT INTO STUDENT VALUES(6, 'RITA', 'CS',
'SR', 21); SELECT * FROM STUDENT;
INSERT INTO FACULTY VALUES(11,
'HARISH', 1000); INSERT INTO FACULTY
VALUES(12, 'MV', 1000); INSERT INTO
FACULTY VALUES(13 , 'MIRA', 1001); INSERT
INTO FACULTY VALUES(14, 'SHIVA', 1002);
INSERT INTO FACULTY VALUES(15, 'NUPUR',
1000); SELECT * FROM FACULTY;
INSERT INTO CLASS VALUES('CLASS1', '12/11/15 10:15:16',
'R1', 14); INSERT INTO CLASS VALUES('CLASS10', '12/11/15
10:15:16', 'R128', 14); INSERT INTO CLASS VALUES('CLASS2',
'12/11/15 10:15:20', 'R2', 12); INSERT INTO CLASS
VALUES('CLASS3', '12/11/15 10:15:25', 'R3', 11); INSERT INTO
CLASS VALUES('CLASS4', '12/11/15 20:15:20', 'R4', 14); INSERT
INTO CLASS VALUES('CLASS5', '12/11/15 20:15:20', 'R3', 15);
INSERT INTO CLASS VALUES('CLASS6', '12/11/15 13:20:20',
'R2', 14); INSERT INTO CLASS VALUES('CLASS7', '12/11/15
10:10:10', 'R3', 14); SELECT * FROM CLASS;
INSERT INTO ENROLLED VALUES(1,
'CLASS1'); INSERT INTO ENROLLED
VALUES(2, 'CLASS1'); INSERT INTO
ENROLLED VALUES(3, 'CLASS3');
INSERT INTO ENROLLED VALUES(4,
'CLASS3'); INSERT INTO ENROLLED
VALUES(5, 'CLASS4'); INSERT INTO
ENROLLED VALUES(1, 'CLASS5');
INSERT INTO ENROLLED VALUES(2,
'CLASS5'); INSERT INTO ENROLLED
VALUES(3, 'CLASS5'); INSERT INTO
ENROLLED VALUES(4, 'CLASS5');
INSERT INTO ENROLLED VALUES(5,
'CLASS5'); SELECT * FROM
ENROLLED;
-- QUERY 1
SELECT DISTINCT S.SNAME
FROM STUDENT S, CLASS C, ENROLLED E, FACULTY F
WHERE S.SNUM = E.SNUM AND E.CNAME = C.CNAME AND C.FID =
F.FID AND F.FNAME = 'HARISH' AND S.LVL = 'JR';
-- QUERY 2
SELECT DISTINCT
```

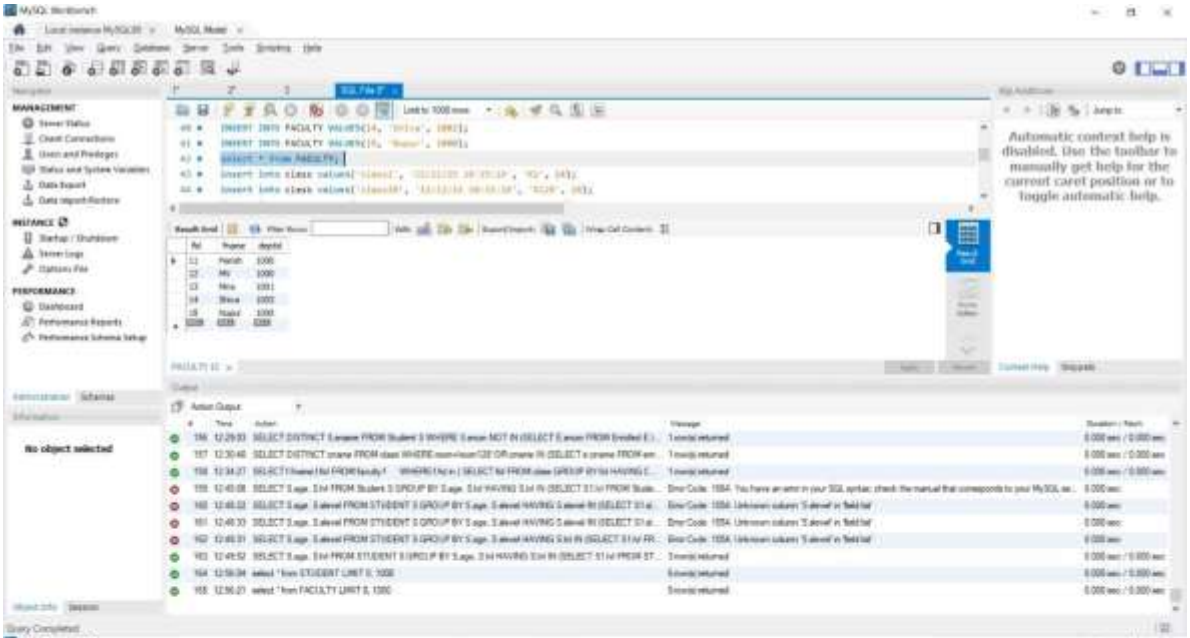
```
CNAME FROM
CLASS
WHERE ROOM='ROOM128'
```

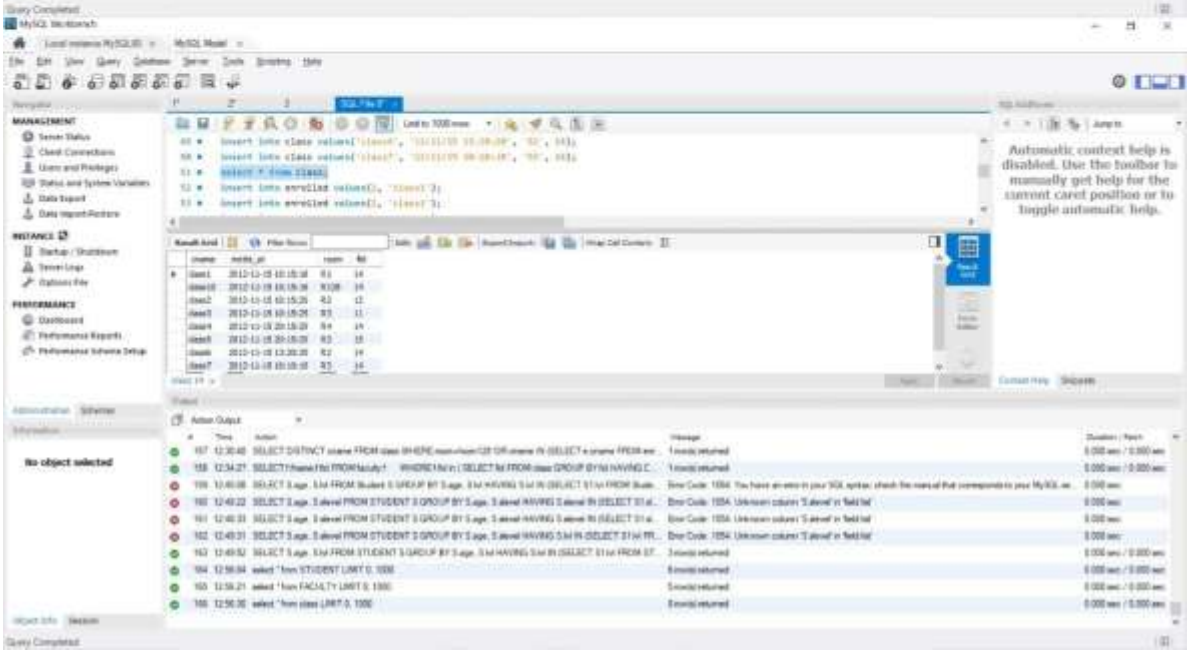
```
OR
CNAME IN (SELECT E.CNAME FROM ENROLLED E GROUP BY E.CNAME HAVING
COUNT(*)>=5);
-- QUERY 3
SELECT DISTINCT
S.SNAME FROM
STUDENT S
WHERE S.SNUM IN (SELECT E1.SNUM
FROM ENROLLED E1, ENROLLED E2, CLASS C1,
CLASS C2 WHERE E1.SNUM = E2.SNUM AND
E1.CNAME <> E2.CNAME AND E1.CNAME =
C1.CNAME
AND E2.CNAME = C2.CNAME AND C1.METTS_AT = C2.METTS_AT);
-- QUERY 4
SELECT
F.FNAME,F.FID
FROM FACULTY F
WHERE F.FID IN ( SELECT FID FROM CLASS
GROUP BY FID HAVING COUNT(*)=(SELECT COUNT(DISTINCT ROOM) FROM
CLASS) );
-- QUERY 5
SELECT DISTINCT
F.FNAME FROM FACULTY
F
WHERE 5 > (SELECT
COUNT(E.SNUM) FROM CLASS C,
ENROLLED E
WHERE C.CNAME =
E.CNAME AND C.FID =
F.FID);
-- QUERY 6
SELECT DISTINCT
S.SNAME FROM
STUDENT S
WHERE S.SNUM NOT IN (SELECT
E.SNUM FROM ENROLLED E );
-- QUERY 7
SELECT S.AGE,
S.LVL FROM
STUDENT S GROUP
BY S.AGE, S.LVL
HAVING S.LVL IN (SELECT
```

SCREENSHOTS OF THE PROGRAM OUTPUT :

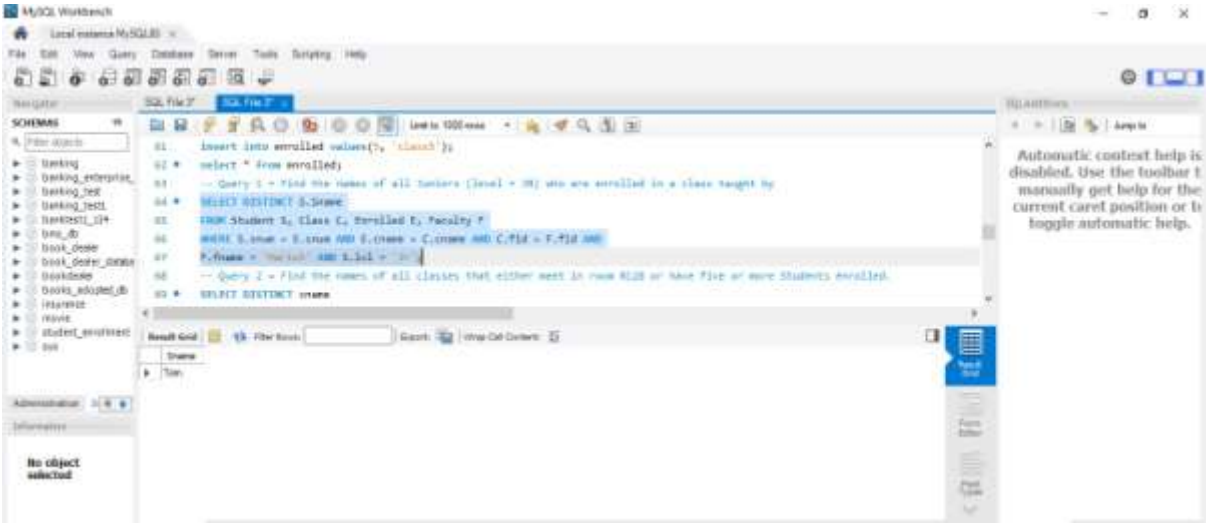
- ## CREATION AND INSERTION OF VALUES



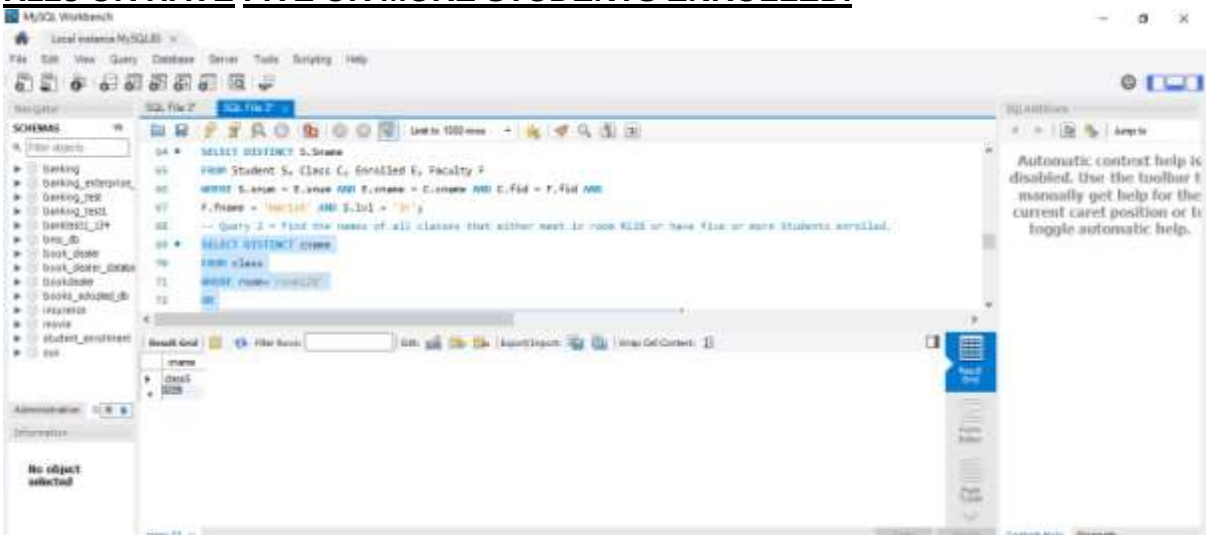




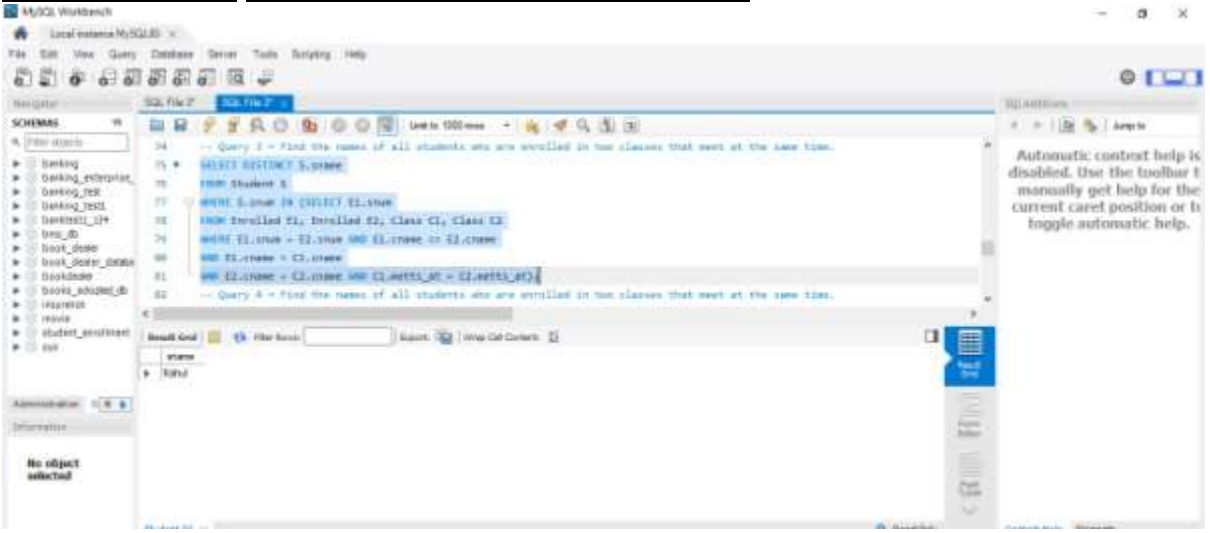
QUERY 1 = FIND THE NAMES OF ALL JUNIORS (LEVEL = JR) WHO ARE ENROLLED IN A CLASS TAUGHT BY



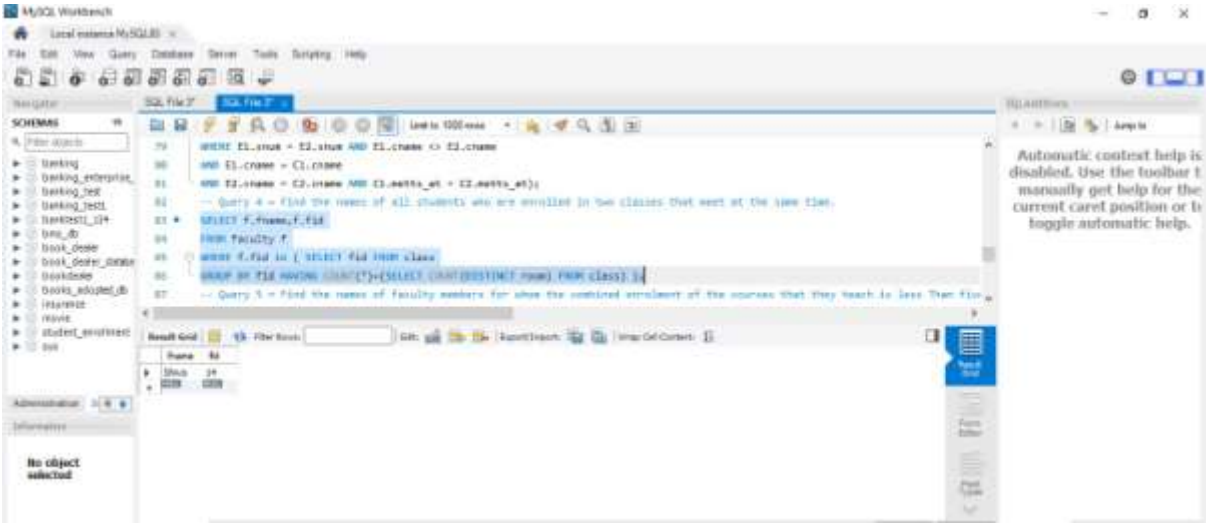
QUERY 2 = FIND THE NAMES OF ALL CLASSES THAT EITHER MEET IN ROOM R128 OR HAVE FIVE OR MORE STUDENTS ENROLLED.



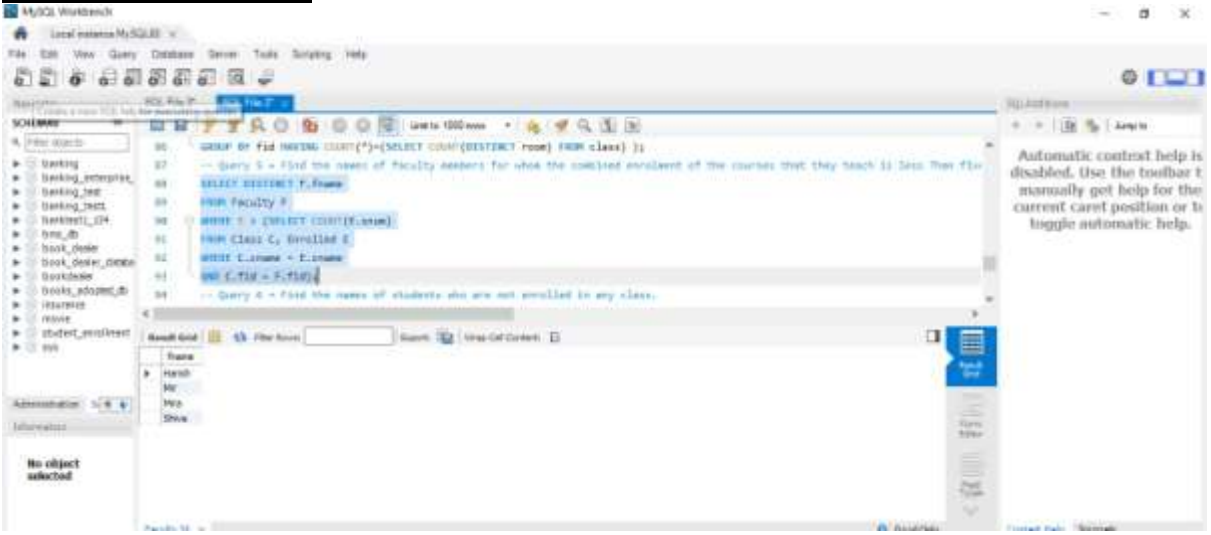
QUERY 3 = FIND THE NAMES OF ALL STUDENTS WHO ARE ENROLLED IN TWO CLASSES THAT MEET AT THE SAME TIME.



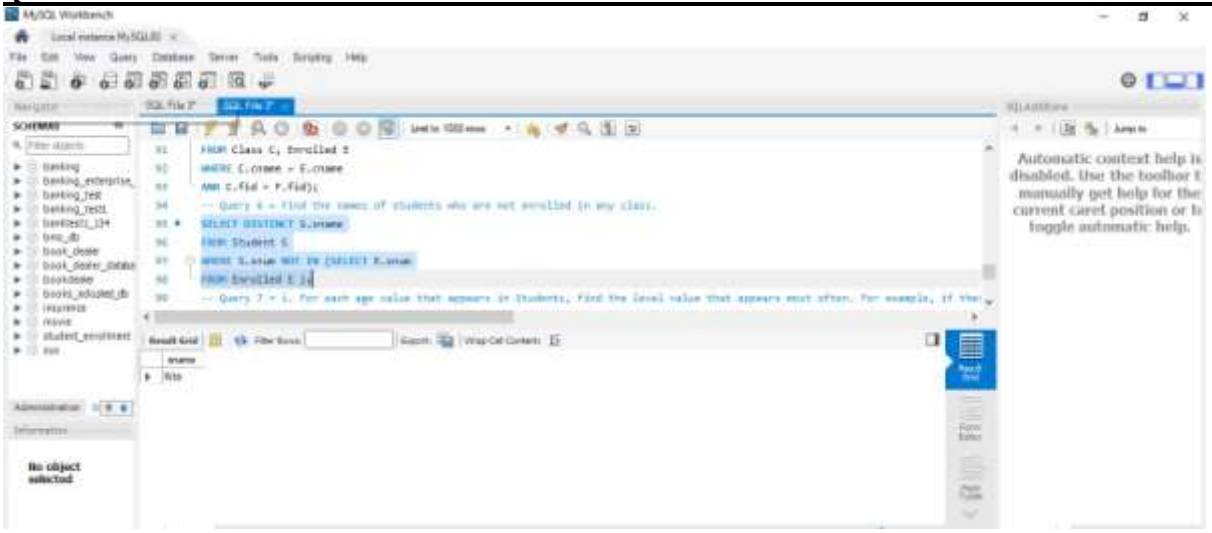
QUERY 4 = FIND THE NAMES OF ALL STUDENTS WHO ARE ENROLLED IN TWO CLASSES THAT MEET AT THE SAME TIME.



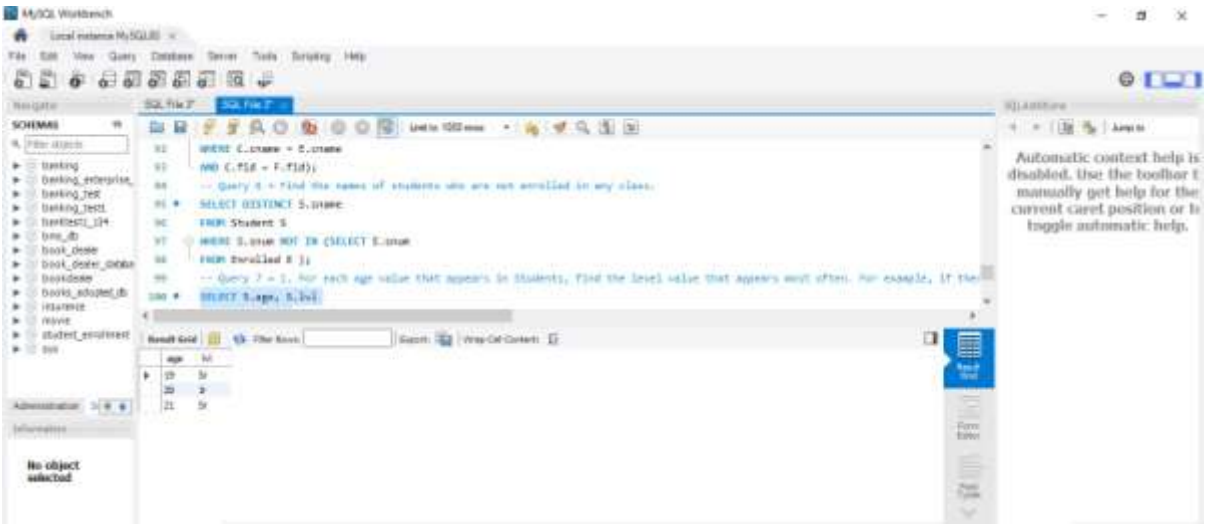
QUERY 5 = FIND THE NAMES OF FACULTY MEMBERS FOR WHOM THE COMBINED ENROLMENT OF THE COURSES THAT THEY TEACH IS LESS THAN FIVE.



QUERY 6 = FIND THE NAMES OF STUDENTS WHO ARE NOT ENROLLED IN ANY CLASS.



QUERY 7 = I. FOR EACH AGE VALUE THAT APPEARS IN STUDENTS, FIND THE LEVEL VALUE THAT APPEARS MOST OFTEN. FOR EXAMPLE, IF THERE ARE MORE FR LEVEL STUDENTS AGED 18 THAN SR, JR, OR SO STUDENTS AGED 18, YOU SHOULD PRINT THE PAIR (18,FR)



PROGRAM-10:COLLEGE DATABASE

QUESTION:

Consider the schema for College Database:
STUDENT(USN, SName, Address, Phone, Gender)
SEMSEC(SSID, Sem, Sec)
CLASS(USN, SSID)
SUBJECT(Subcode, Title, Sem, Credits)
IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)
Write SQL queries to
i. List all the student details studying in fourth semester ‘C’ section.
ii. Compute the total number of male and female students in each semester and in each section.
iii. Create a view of Test1 marks of student USN ‘1BI15CS101’ in all subjects.
iv. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students.
v. 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.

PROGRAM CODE:

```
create database college;
use college;
CREATE TABLE STUDENT (
USN VARCHAR (10) PRIMARY KEY,
SNAME VARCHAR (25),
ADDRESS VARCHAR (25),
PHONE real,
GENDER CHAR (1));
CREATE TABLE SEMSEC (
```

```
SSID VARCHAR (5) PRIMARY KEY,
SEM INT (2),
SEC CHAR (1));
CREATE TABLE CLASS (
USN VARCHAR (10),
SSID VARCHAR (5), PRIMARY
KEY (USN, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN),
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID));
CREATE TABLE SUBJECT (
SUBCODE VARCHAR (8),
TITLE VARCHAR (20),
SEM INT (2),
CREDITS INT (2),
PRIMARY KEY (SUBCODE));
CREATE TABLE IAMARKS (
USN VARCHAR (10),
SUBCODE VARCHAR (8),
SSID VARCHAR(5),
TEST1 INT(2),
TEST2 INT(2),
TEST3 INT(2),
FINALIA INT (2),
PRIMARY KEY (USN, SUBCODE, SSID),
FOREIGN KEY (USN) REFERENCES STUDENT (USN),
FOREIGN KEY (SUBCODE) REFERENCES SUBJECT (SUBCODE),
FOREIGN KEY (SSID) REFERENCES SEMSEC (SSID));
INSERT INTO STUDENT VALUES('1RN13CS020','AKSHAY','BELAGAVI',8877881122,'M');
INSERT INTO STUDENT
VALUES('1RN13CS062','SANDHYA','BENGALURU',7722829912,'F');
INSERT INTO STUDENT
VALUES('1RN13CS091','TEESHA','BENGALURU',7712312312,'F');
INSERT INTO STUDENT
VALUES('1RN13CS066','SUPRIYA','MANGALURU',8877881122,'F');
INSERT INTO STUDENT
VALUES('1RN14CS010','ABHAY','BENGALURU',9900211201,'M');
INSERT INTO STUDENT
VALUES('1RN14CS032','BHASKAR','BENGALURU',9923211099,'M');
INSERT INTO STUDENT VALUES ('1RN14CS025','ASMI','BENGALURU', 7894737377,'F');
INSERT INTO STUDENT VALUES ('1RN15CS011','AJAY','TUMKUR', 9845091341,'M');
INSERT INTO STUDENT VALUES
('1RN15CS029','CHITRA','DAVANGERE',7696772121,'F');
INSERT INTO STUDENT VALUES ('1RN15CS045','JEEVA','BELLARY', 9944850121,'M');
INSERT INTO STUDENT VALUES
('1RN15CS091','SANTOSH','MANGALURU',8812332201,'M');
INSERT INTO STUDENT VALUES('1RN16CS045','ISMAIL','KALBURGI',9900232201,'M');
INSERT INTO STUDENT VALUES
('1RN16CS088','SAMEERA','SHIMOGA',9905542212,'F');
INSERT INTO STUDENT VALUES
('1RN16CS122','VINAYAKA','CHIKAMAGALUR',8800880011,'M');

INSERT INTO SEMSEC VALUES ('CSE8A', 8,'A');
INSERT INTO SEMSEC VALUES ('CSE8B', 8,'B');
INSERT INTO SEMSEC VALUES ('CSE8C',8,'C');
INSERT INTO SEMSEC VALUES ('CSE7A',7,'A');
```

```
INSERT INTO SEMSEC VALUES ('CSE7B',7,'B');
INSERT INTO SEMSEC VALUES ('CSE7C',7,'C');
INSERT INTO SEMSEC VALUES ('CSE6A',6,'A');
INSERT INTO SEMSEC VALUES ('CSE6B', 6,'B');
INSERT INTO SEMSEC VALUES ('CSE6C', 6,'C');
INSERT INTO SEMSEC VALUES ('CSE5A', 5,'A');
INSERT INTO SEMSEC VALUES ('CSE5B', 5,'B');
INSERT INTO SEMSEC VALUES ('CSE5C', 5,'C');
INSERT INTO SEMSEC VALUES ('CSE4A',4,'A');
INSERT INTO SEMSEC VALUES ('CSE4B', 4,'B');
INSERT INTO SEMSEC VALUES('CSE4C',4,'C');
INSERT INTO SEMSEC VALUES ('CSE3A', 3,'A');
INSERT INTO SEMSEC VALUES ('CSE3B', 3,'B');
INSERT INTO SEMSEC VALUES('CSE3C',3,'C');
INSERT INTO SEMSEC VALUES ('CSE2A', 2,'C');
INSERT INTO SEMSEC VALUES ('CSE2B', 2,'B');
INSERT INTO SEMSEC VALUES ('CSE2C', 2,'C');
INSERT INTO SEMSEC VALUES ('CSE1A', 1,'A');
INSERT INTO SEMSEC VALUES ('CSE1B', 1,'B');
INSERT INTO SEMSEC VALUES ('CSE1C', 1,'C');

INSERT INTO CLASS VALUES('1RN13CS020','CSE8A');
INSERT INTO CLASS VALUES('1RN13CS062','CSE8A');
INSERT INTO CLASS VALUES('1RN13CS066','CSE8B');
INSERT INTO CLASS VALUES('1RN13CS091','CSE8C');
INSERT INTO CLASS VALUES('1RN14CS010','CSE7A');
INSERT INTO CLASS VALUES('1RN14CS025','CSE7A');
INSERT INTO CLASS VALUES('1RN14CS032','CSE7A');
INSERT INTO CLASS VALUES('1RN15CS011','CSE4A');
INSERT INTO CLASS VALUES('1RN15CS029','CSE4A');
INSERT INTO CLASS VALUES('1RN15CS045','CSE4B');
INSERT INTO CLASS VALUES('1RN15CS091','CSE4C');
INSERT INTO CLASS VALUES('1RN16CS045','CSE3A');
INSERT INTO CLASS VALUES('1RN16CS088','CSE3B');
INSERT INTO CLASS VALUES('1RN16CS122','CSE3C');

INSERT INTO SUBJECT VALUES ('10CS81','ACA', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS82','SSM', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS83','NM', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS84','CC', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS85','PW', 8, 4);
INSERT INTO SUBJECT VALUES ('10CS71','OOAD', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS72','ECS', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS73','PTW', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS74','DWDM', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS75','JAVA', 7, 4);
INSERT INTO SUBJECT VALUES ('10CS76','SAN', 7, 4);
INSERT INTO SUBJECT VALUES ('15CS51', 'ME', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS52','CN', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS53','DBMS', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS54','ATC', 5, 4);
INSERT INTO SUBJECT VALUES ('15CS55','JAVA', 5, 3);
INSERT INTO SUBJECT VALUES ('15CS56','AI', 5, 3);
INSERT INTO SUBJECT VALUES ('15CS41','M4', 4, 4);
```

```
INSERT INTO SUBJECT VALUES ('15CS42','SE', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS43','DAA', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS44','MPMC', 4, 4);
INSERT INTO SUBJECT VALUES ('15CS45','OOC', 4, 3);
INSERT INTO SUBJECT VALUES ('15CS46','DC', 4, 3);
INSERT INTO SUBJECT VALUES ('15CS31','M3', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS32','ADE', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS33','DSA', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS34','CO', 3, 4);
INSERT INTO SUBJECT VALUES ('15CS35','USP', 3, 3);
INSERT INTO SUBJECT VALUES ('15CS36','DMS', 3, 3);

INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS81','CSE8C', 15, 16,18);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS82','CSE8C', 12, 19,14);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS83','CSE8C', 19, 15,20);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS84','CSE8C', 20, 16,19);
INSERT INTO IAMARKS (USN, SUBCODE, SSID, TEST1, TEST2, TEST3)VALUES
('1RN13CS091','10CS85','CSE8C', 15, 15,12);
SELECT * FROM STUDENT;
SELECT * FROM SEMSEC;
SELECT * FROM CLASS;
SELECT * FROM SUBJECT;
SELECT * FROM IAMARKS;

SELECT S.*, SS.SEM, SS.SEC
FROM STUDENT S, SEMSEC SS, CLASS C
WHERE S.USN = C.USN AND
SS.SSID = C.SSID AND
SS.SEM = 4 AND
SS.SEc='C';

SELECT SS.SEM, SS.SEC, S.GENDER, COUNT(S.GENDER) AS COUNT
FROM STUDENT S, SEMSEC SS, CLASS C
WHERE S.USN = C.USN AND
SS.SSID = C.SSID
GROUP BY SS.SEM, SS.SEC, S.GENDER
ORDER BY SEM;

CREATE VIEW STU_TEST1_MARKS_VIEW
AS
SELECT TEST1, SUBCODE
FROM IAMARKS
WHERE USN = '1RN13CS091';

-- QUERY 4

DELIMITER //

CREATE PROCEDURE AVG_MARKS()
BEGIN
```

```
DECLARE C_A INTEGER;
DECLARE C_B INTEGER;
DECLARE C_C INTEGER;
DECLARE C_SUM INTEGER;
DECLARE C_AVG INTEGER;
DECLARE C_USN VARCHAR(10);
DECLARE C_SUBCODE VARCHAR(8);
DECLARE C_SSID VARCHAR(5);
DECLARE C_IAMARKS CURSOR FOR
SELECT GREATEST(TEST1,TEST2) AS A, GREATEST(TEST1,TEST3) AS B,
GREATEST(TEST3,TEST2) AS C, USN, SUBCODE, SSID
FROM IAMARKS
WHERE FINALIA IS NULL
FOR UPDATE;
OPEN C_IAMARKS;
LOOP
FETCH C_IAMARKS INTO C_A, C_B, C_C, C_USN, C_SUBCODE, C_SSID;
IF (C_A != C_B) THEN
 SET C_SUM=C_A+C_B;
ELSE
 SET C_SUM=C_A+C_C;
END IF;
SET C_AVG=C_SUM/2;
UPDATE IAMARKS SET FINALIA = C_AVG
WHERE USN = C_USN AND SUBCODE = C_SUBCODE AND SSID = C_SSID;
END LOOP;
CLOSE C_IAMARKS;
END;
//

CALL AVG_MARKS();

SELECT * FROM IAMARKS;
-- QUERY 5

SELECT S.USN,S.SNAME,S.ADDRESS,S.PHONE,S.GENDER,
(CASE
WHEN IA.FINALIA BETWEEN 17 AND 20 THEN 'OUTSTANDING'
WHEN IA.FINALIA BETWEEN 12 AND 16 THEN 'AVERAGE'
ELSE 'WEAK'
END) AS CAT
FROM STUDENT S, SEMSEC SS, IAMARKS IA, SUBJECT SUB
WHERE S.USN = IA.USN AND
SS.SSID = IA.SSID AND
SUB.SUBCODE = IA.SUBCODE AND
SUB.SEM = 8;
```

OUTPUT SCREENSHOTS:

LABS_MOVIES | SQL File 3* | SQL File 4*

```
148 e    IN*iERT ZUT0 I NARKS (USN , SUBCODE, SSID, TE5T1, TE5T2. TE5T3) VALUES
149      \'1RN13C3B^.1', '10CS83', 'USE6C', 13, 13, AZ);
13B e SELECT s FR0B 'STUDENT i
151 e SELECT s EROB SEPISEC;
152      SELECT s EROB CLASS ;
153 e SELECT + EROB SUB 3ECT ;
13 4 e SELECT T FR0B IANAftKS ;
135
156 e                                "
157
158
159                    .
160


162
163      SELECT SS . SEBI , SS . SEC , 5. GENDER , CDUNT { S . GMDER ) AS COUNT
164      FR0B STUDEHT S , SENSEC 5S , CLASS C
165      I#HERE S.USN = C.USN AND
```

look ¿ land , zD errorsFund

Reeult Grid @- •@• Fil@r Rose: *ll ' Export:

1RN15CS091 SANTOSH MANGALURU 8012332 M '4 'C

```
1f5
1Z6     SELECT S.v, SS.SEM, SS.5EC
157     FROM STUDENT S, SEMSkC 5S. [LAWS [
        WHERE S.NSW = C.U5N AND
159     SS.SSID - C.SSID AND
160     SS.SEM = 4 AND
161     SS.SEc='C';
162
163 0
164
165
166
167
168
1£Q
```

SEM	SEC	GENDER	COUwT
g	A	M	1
g	B	F	1
4	A	F	1
4	A	M	1
‹	C	U	1
8	A	M	1
8	B	F	1
8	C	F	1

LABS_MOVIES | SQL File 3* | SQL File 4*

```
DEC LARE C_A ZtJTEEER ;
DEC LARE C_B ZtJTEEER ;
BEC LARE C_E ZtJTEEER ;
DEC LARE C_SMf4 INTEGER ;
DEC LARE C_AVG INTEGER ;
DEC LARE C_USN VARCHAR ( be ) ;
DEC LARE C_SUBCODE VARCHAR (8) ;
DEC LARE C_SSID VA6CLIAR 5) ;
DEC LARE C_IAI4xRrs cuRsoe r0R
SELECT GREATEST(TEST1 ,TEST2) AT A, GREATEST(TEST1, TEST3 j AS B, GREATEST(TEST3, TEST2j xs c, usN. suecODE,   SSID
FRON IANARKS
MERE FINAL IA IS fJULL
FOR UPDATE ;
oPeN E_J t'l Rxs i
```

USN	SUBCODG	SSID	TEST1	TGST2	TEST3	FTNALLA
1RN1g IS091	10 ISBN	CSE8C 1'2	16	18	17	
1RN1g 5091	1 82	CBE8C 12	19	14	17	
1RN1gCS091	10 S82	CSE8C 18	15	@	@	
1RN1g 1509	1 10 ISBN	C9E8C 15	15	12	15	

```
SELECT r FROB IJV4ABKS ;

SELECT * ER04 IAKAPKS

"    QUERY 5
```

USN	SNAME	ADDRESS	PHONE	GENDER	CAT
1RN13CS091	TEESHA	BENGALURU	77123123	F	OUTSTANDING
1RN1gCS091	TEEsnx	eEnaxLunu	zz1a3las	x	aiJTsTx+JoiHB
1RN1g S091	TEEsnx	BENDALunu	771lZ31za	F	auTsTxuoino
1RN1 081	wEsnx	eEnoxuunu	zz1zolzs	i=	aiJTsTuoino
1RN1g S091	TEEsnA	BENDALunu	771lZ31za	F	xveRxoE