DBMS LAB RECORD (CIE-1)

NAME: KIZHAKEL SHARAT PRASAD

SEC: 4-B

USN:1BM19CS074

BATCH NO: B2

BATCH IN CHARGE: Dr. K. PANIMOZHI

CONTENTS: PROGRAMS(1 TO 5)

WEEK-1 DBMS LAB

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.

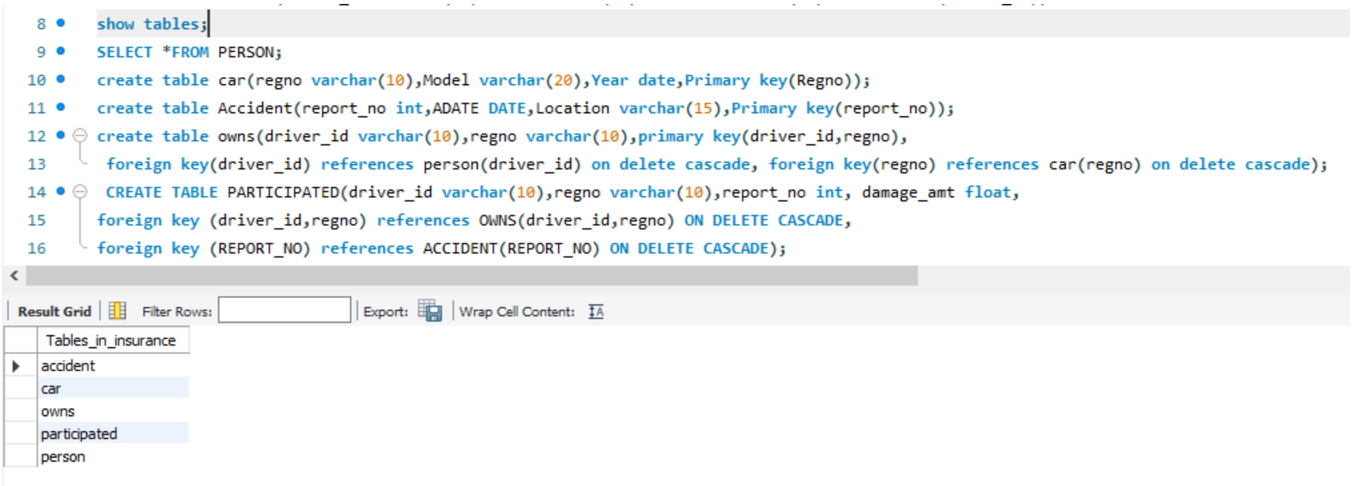
iii. Demonstrate how you

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.

OUTPUT:

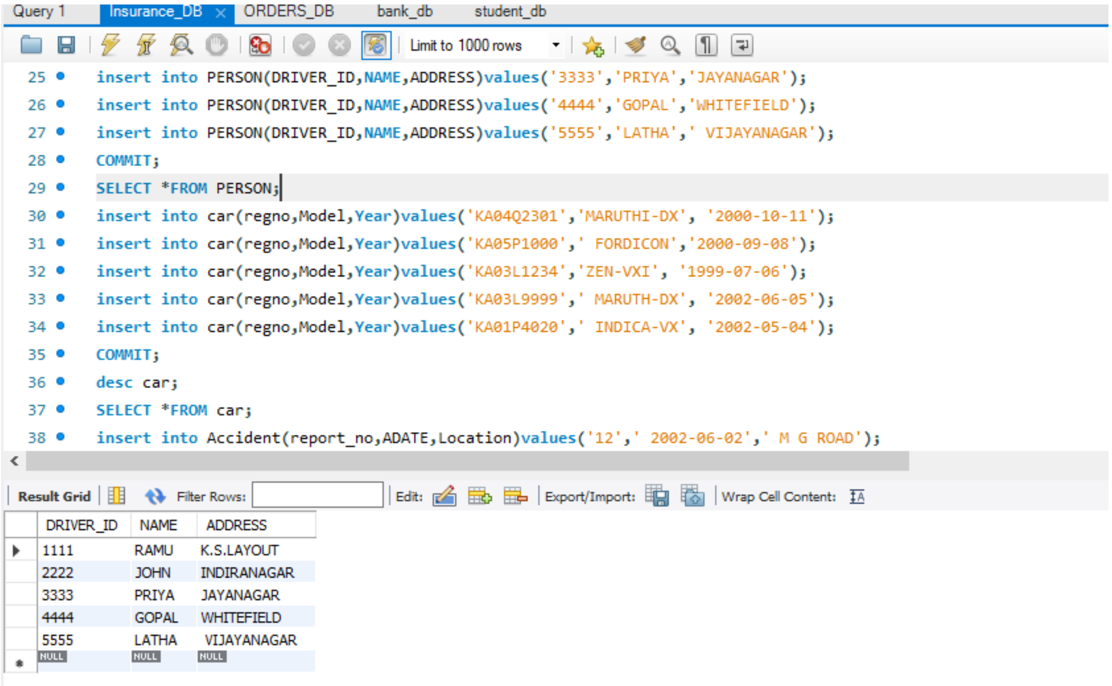
i. Create the above tables by properly specifying the primary keys and the foreign keys.

ii. Enter at least five tuples for each relation.



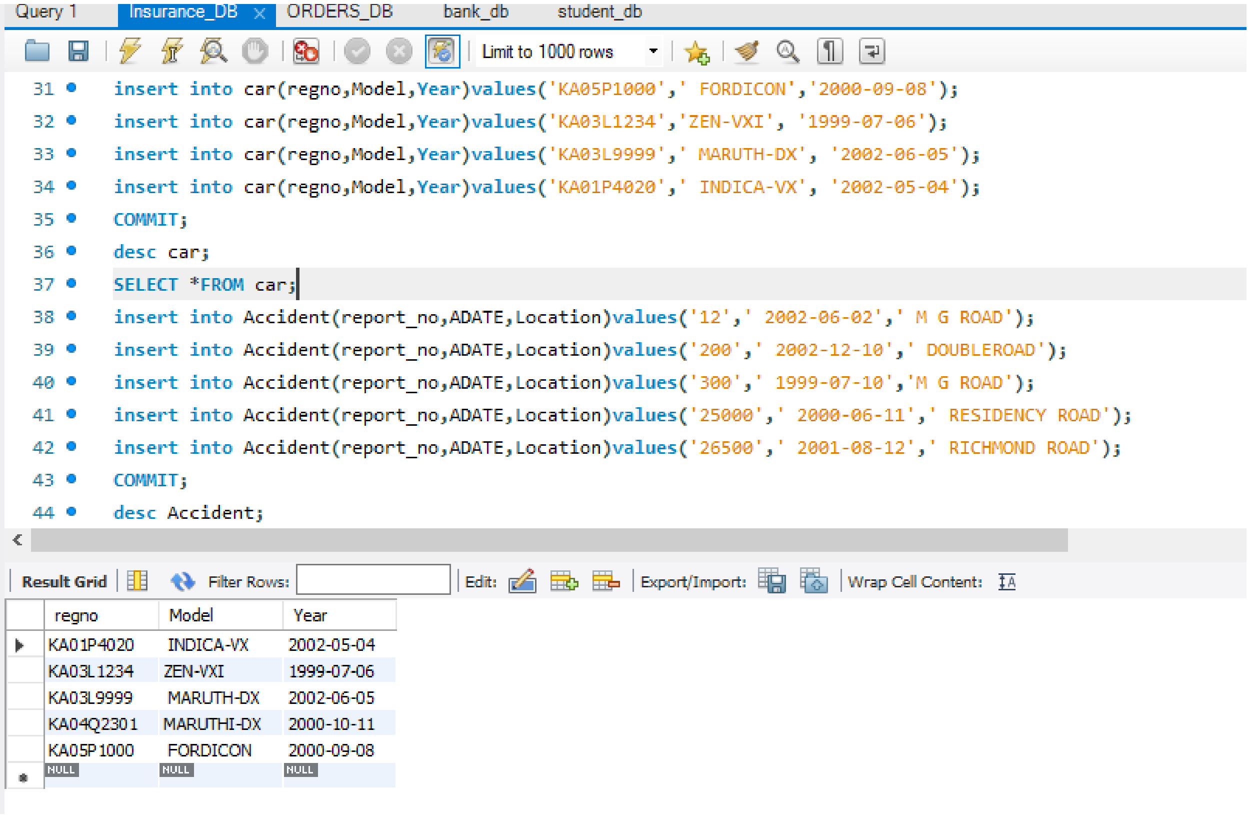
```sql
show tables;
SELECT *FROM PERSON;
create table car(regno varchar(10),Model varchar(20),Year date,Primary key(Regno));
create table Accident(report_no int,ADATE DATE,Location varchar(15),Primary key(report_no));
create table owns(driver_id varchar(10),regno varchar(10),primary key(driver_id,regno),
 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);
```

Tables_in_insurance
accident
car
owns
participated
person



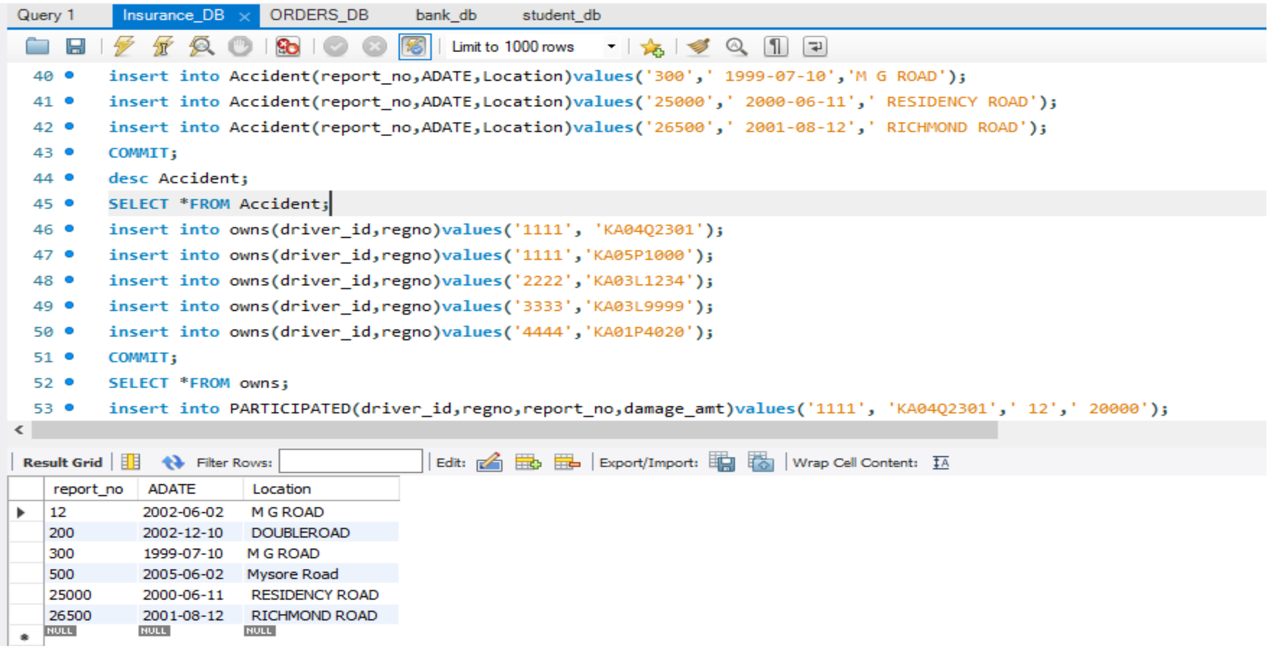
```sql
insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('3333','PRIYA','JAYANAGAR');
insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('4444','GOPAL','WHITEFIELD');
insert into PERSON(DRIVER_ID,NAME,ADDRESS)values('5555','LATHA',' VIJAYANAGAR');
COMMIT;
SELECT *FROM PERSON;
insert into car(regno,Model,Year)values('KA04Q2301','MARUTHI-DX', '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');
```

DRIVER_ID	NAME	ADDRESS
1111	RAMU	K.S.LAYOUT
2222	JOHN	INDIRANAGAR
3333	PRIYA	JAYANAGAR
4444	GOPAL	WHITEFIELD
5555	LATHA	VIJAYANAGAR
NULL	NULL	NULL



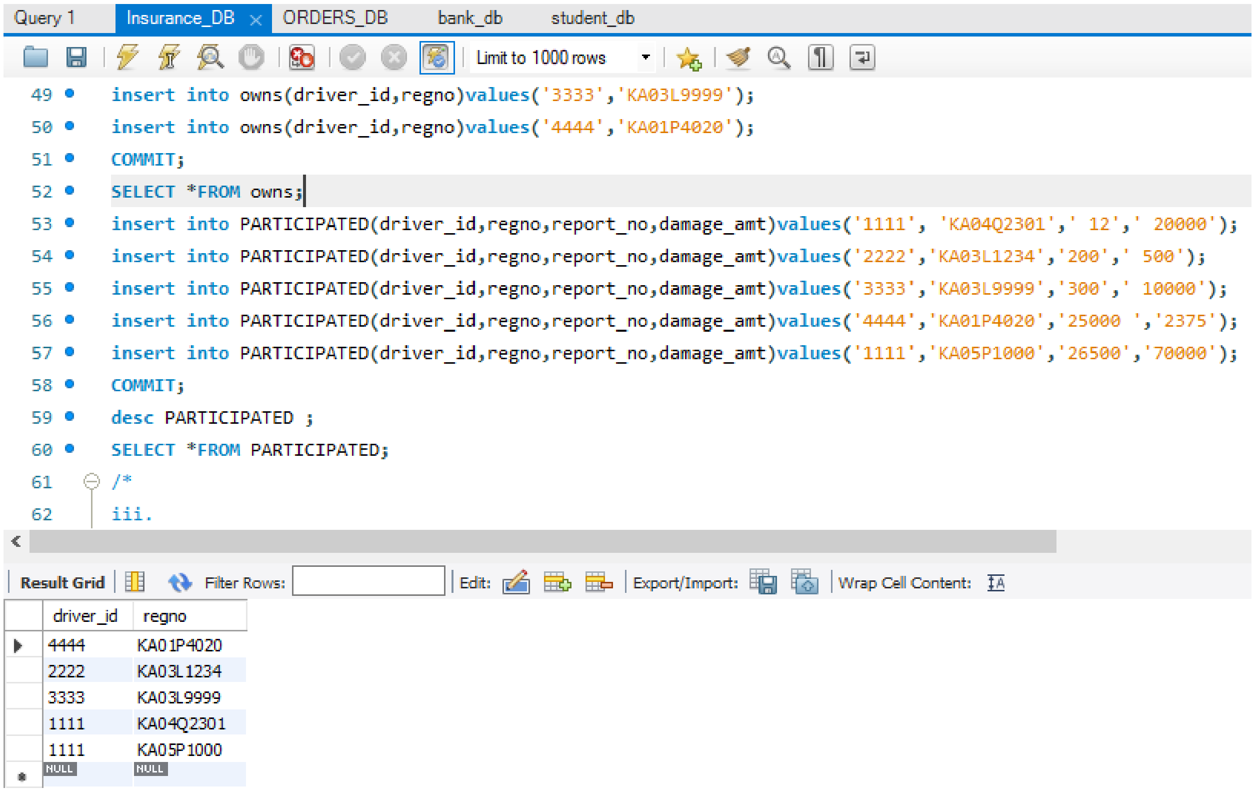
```sql
insert into car(regno,Model,Year)values('KA05P1000',' FORDICON','2000-09-08');
insert into car(regno,Model,Year)values('KA03L1234','ZEN-VXI', '1999-07-06');
insert into car(regno,Model,Year)values('KA03L9999',' MARUTH-DX', '2002-06-05');
insert into car(regno,Model,Year)values('KA01P4020',' INDICA-VX', '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;
```

regno	Model	Year
KA01P4020	INDICA-VX	2002-05-04
KA03L1234	ZEN-VXI	1999-07-06
KA03L9999	MARUTH-DX	2002-06-05
KA04Q2301	MARUTHI-DX	2000-10-11
KA05P1000	FORDICON	2000-09-08
NULL	NULL	NULL



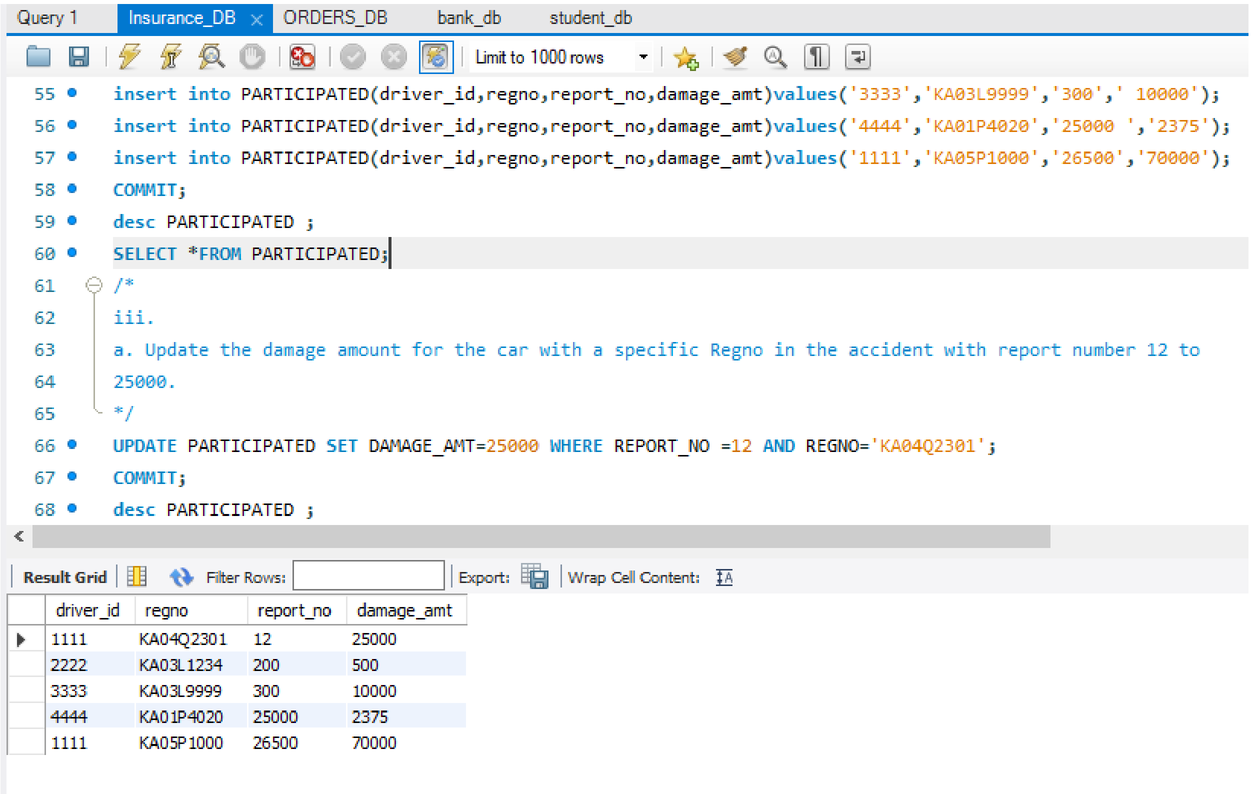
```sql
insert into Accident(report_no,ADATE,Location)values('300',' 1999-07-10','M G ROAD');
insert into Accident(report_no,ADATE,Location)values('25000',' 2000-06-11',' RESIDENCY ROAD');
insert into Accident(report_no,ADATE,Location)values('26500',' 2001-08-12',' RICHMOND ROAD');
COMMIT;
desc Accident;
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;
SELECT *FROM owns;
insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111', 'KA04Q2301',' 12',' 20000');
```

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
26500	2001-08-12	RICHMOND ROAD
NULL	NULL	NULL



```
insert into owns(driver_id,regno)values('3333','KA03L9999');
insert into owns(driver_id,regno)values('4444','KA01P4020');
COMMIT;
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;
/*
iii.
```

driver_id	regno
4444	KA01P4020
2222	KA03L1234
3333	KA03L9999
1111	KA04Q2301
1111	KA05P1000
NULL	NULL

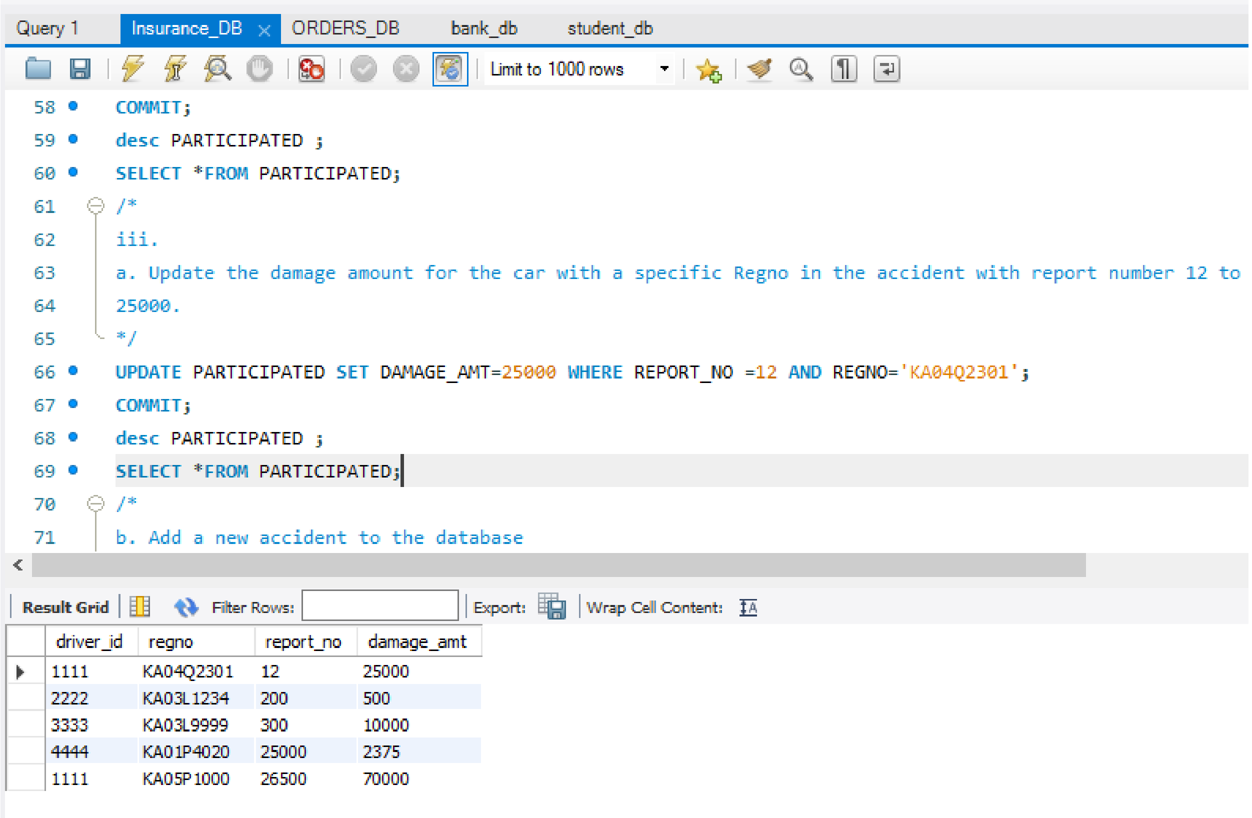


```
insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('3333','KA03L9999','300',' 10000');
insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('4444','KA01P4020','25000 ','2375');
insert into PARTICIPATED(driver_id,regno,report_no,damage_amt)values('1111','KA05P1000','26500','70000');
COMMIT;
desc PARTICIPATED ;
SELECT *FROM PARTICIPATED;
/*
iii.
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 ;
```

driver_id	regno	report_no	damage_amt
1111	KA04Q2301	12	25000
2222	KA03L1234	200	500
3333	KA03L9999	300	10000
4444	KA01P4020	25000	2375
1111	KA05P1000	26500	70000

iii. Demonstrate how you

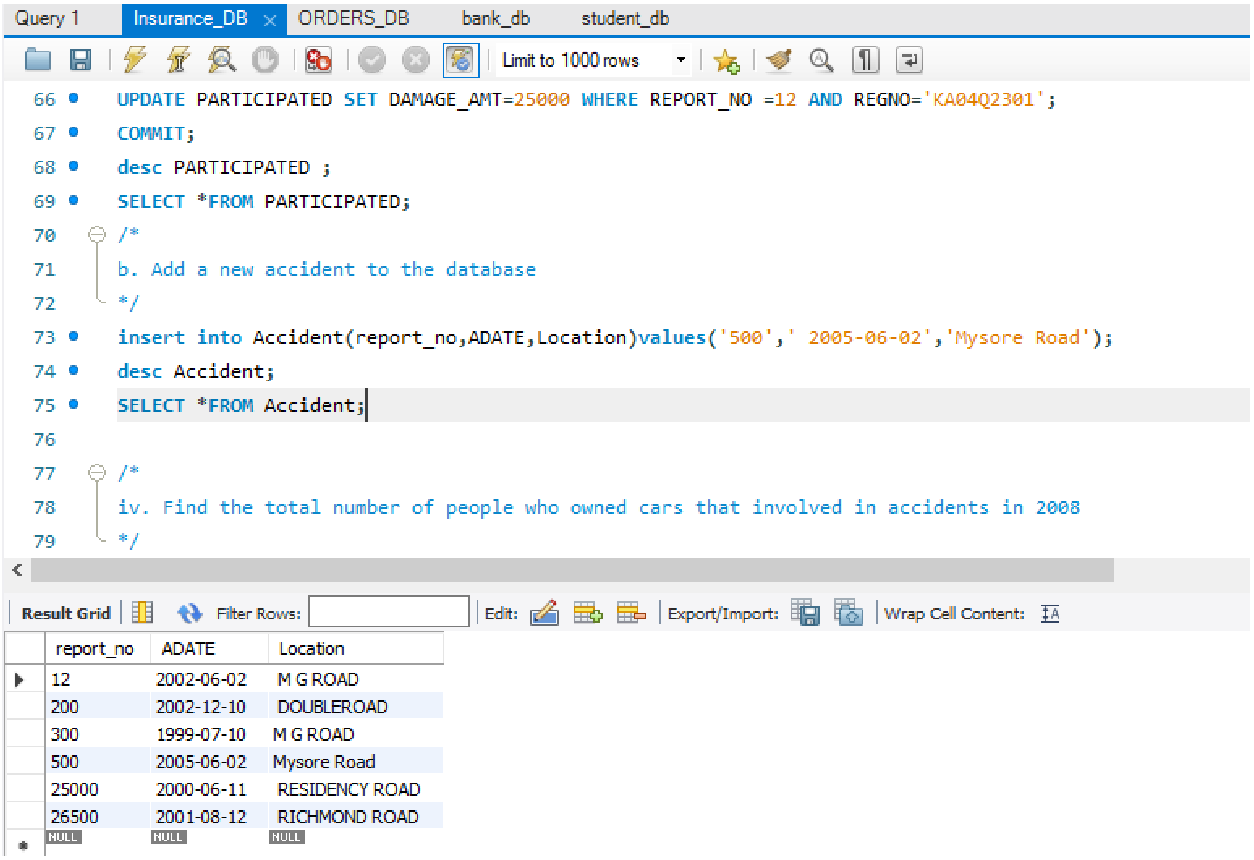
a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000.



```
COMMIT;
desc PARTICIPATED ;
SELECT *FROM PARTICIPATED;
/*
iii.
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
```

driver_id	regno	report_no	damage_amt
1111	KA04Q2301	12	25000
2222	KA03L1234	200	500
3333	KA03L9999	300	10000
4444	KA01P4020	25000	2375
1111	KA05P1000	26500	70000

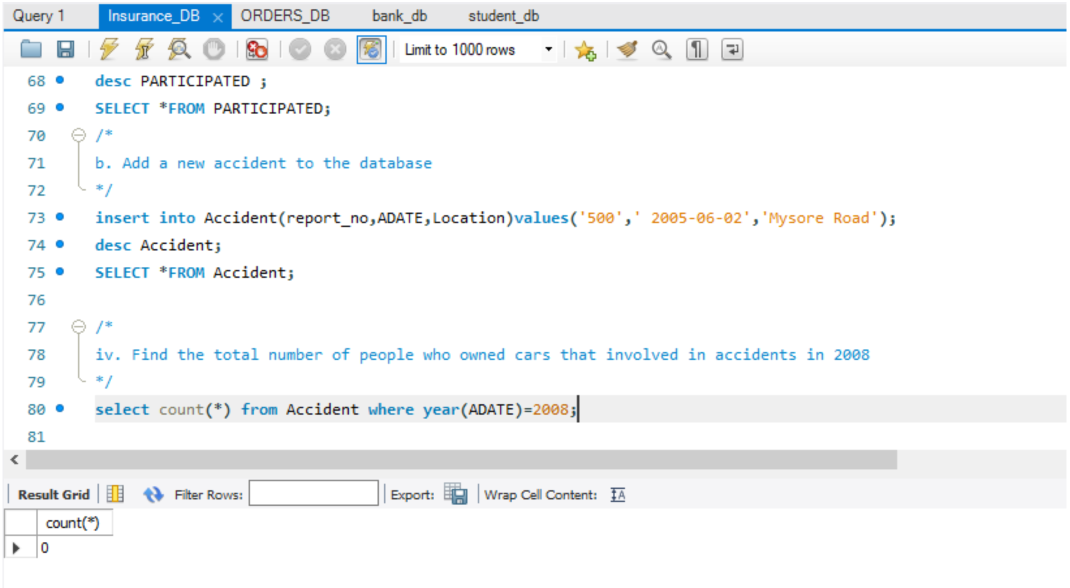
b. Add a new accident to the database.



```sql
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
*/
```

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
26500	2001-08-12	RICHMOND ROAD
NULL	NULL	NULL

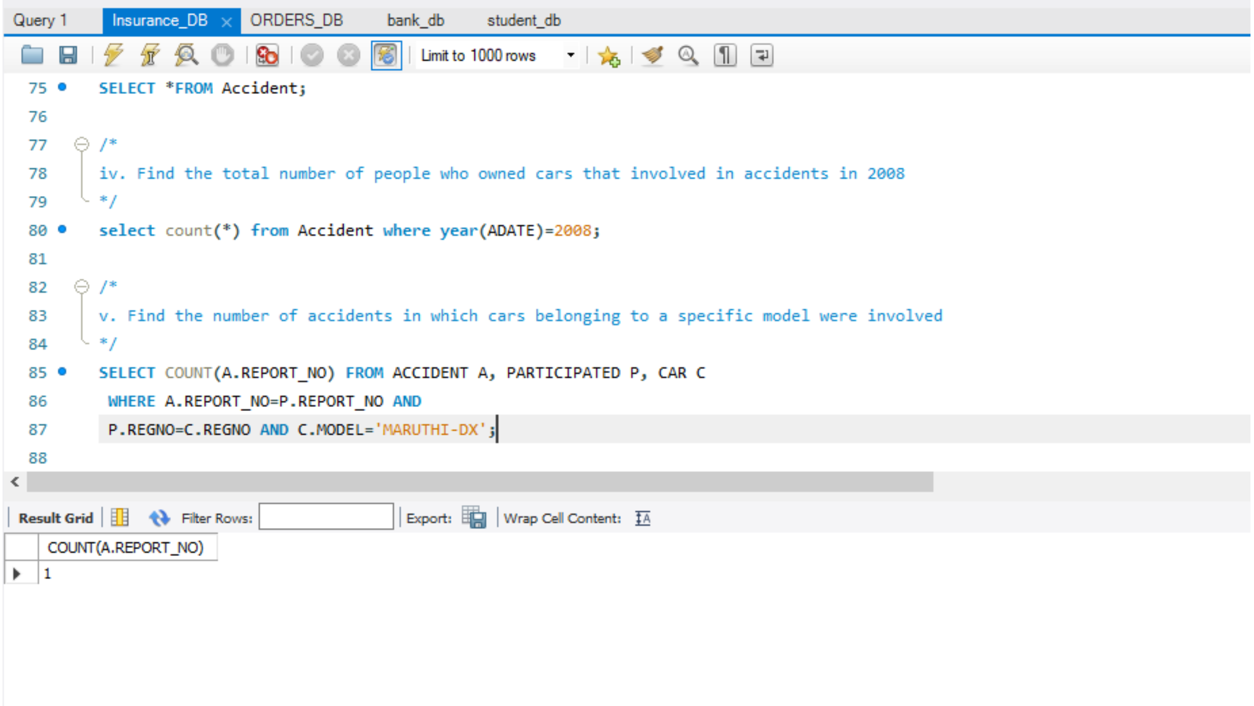
iv. Find the total number of people who owned cars that involved in accidents in 2008.



```
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;

```

```
count(*)
0
```

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



```
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';

```

COUNT(A.REPORT_NO)
1

****************LAB 1 ENDS*****************

WEEK-2 DBMS LAB

LAB PROGRAM 2:

BOOKDEALER DATABASE:

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)

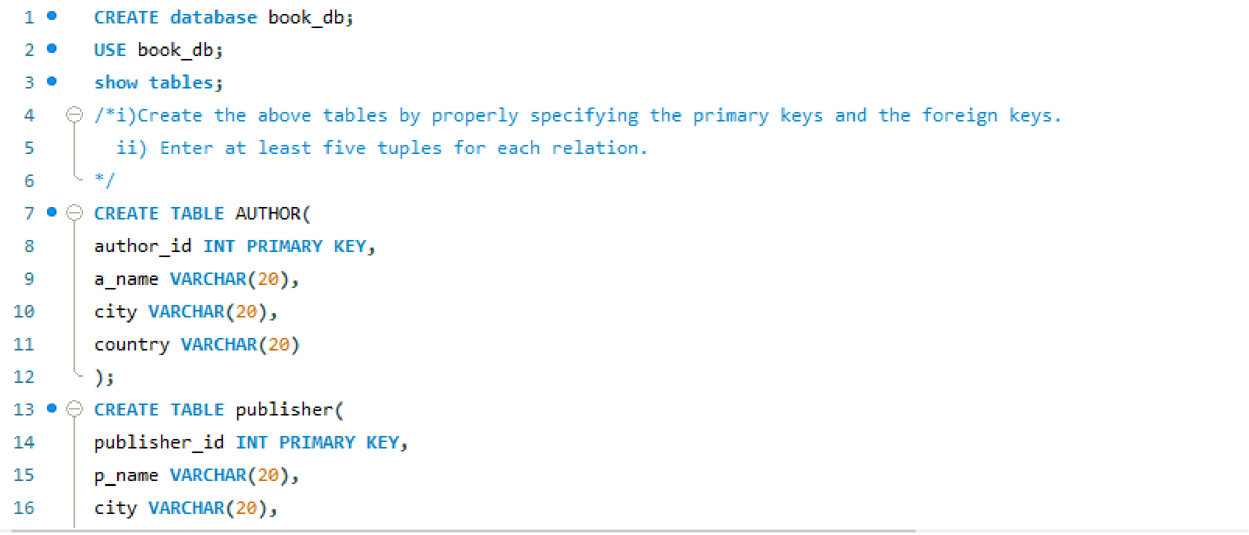
CATEGORY(category-id: int, description: String)

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

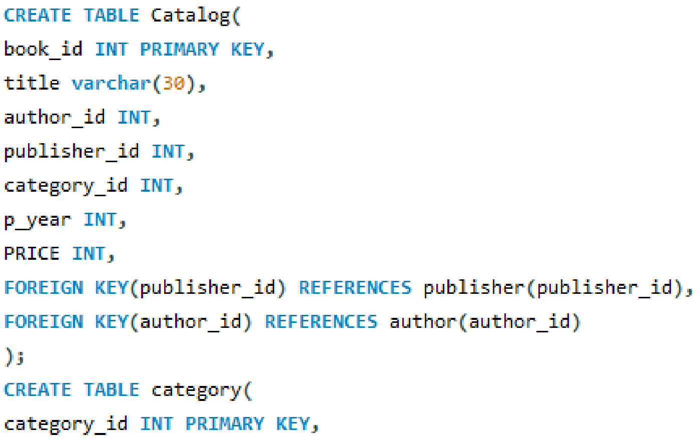
i)Create the above tables by properly specifying the primary keys and the foreign keys.
ii) Enter at least five tuples for each relation.
iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books in the
catalog and the year of publication is after 2000.
iv) 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%.

SCREENSHOTS OF OUTPUT:

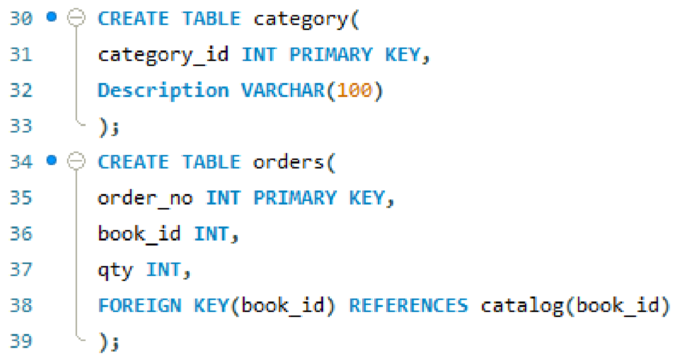
i)Create the above tables by properly specifying the primary keys and the foreign keys.
ii) Enter at least five tuples for each relation.



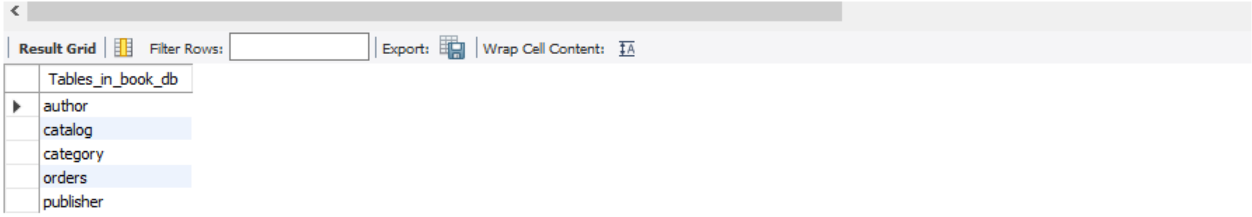
```
CREATE database book_db;
USE book_db;
show tables;
/*i)Create the above tables by properly specifying the primary keys and the foreign keys.
  ii) Enter at least five tuples for each relation.
*/
CREATE TABLE AUTHOR(
author_id INT PRIMARY KEY,
a_name VARCHAR(20),
city VARCHAR(20),
country VARCHAR(20)
);
CREATE TABLE publisher(
publisher_id INT PRIMARY KEY,
p_name VARCHAR(20),
city VARCHAR(20),
```



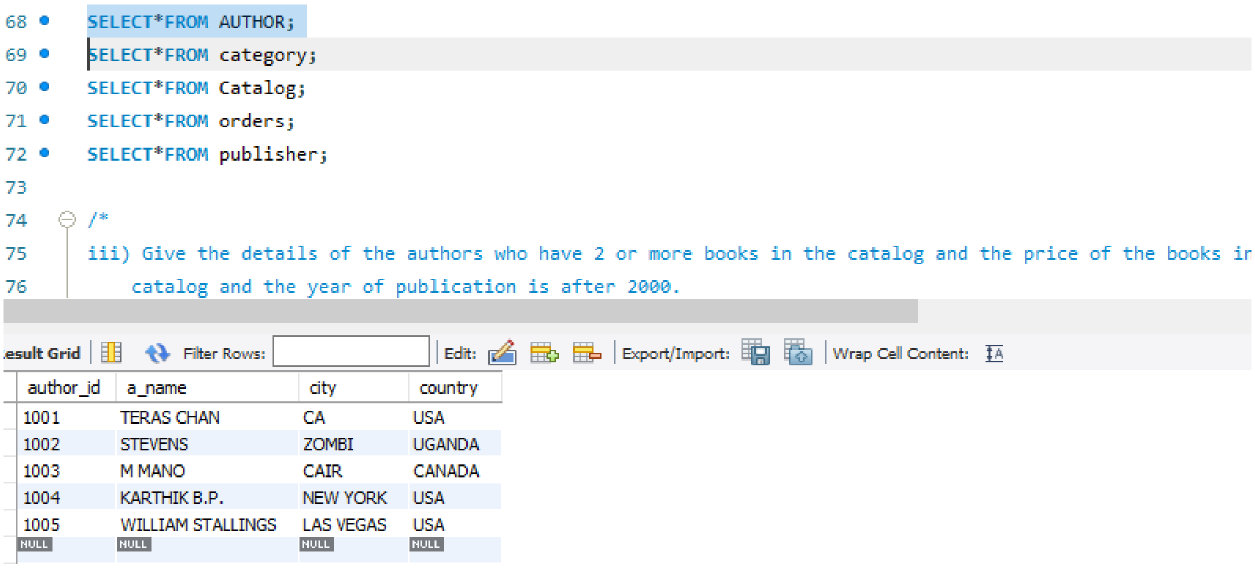
```
CREATE TABLE Catalog(
book_id INT PRIMARY KEY,
title varchar(30),
author_id INT,
publisher_id INT,
category_id INT,
p_year INT,
PRICE INT,
FOREIGN KEY(publisher_id) REFERENCES publisher(publisher_id),
FOREIGN KEY(author_id) REFERENCES author(author_id)
);
CREATE TABLE category(
category_id INT PRIMARY KEY,
```



```
CREATE TABLE category(
category_id INT PRIMARY KEY,
Description VARCHAR(100)
);
CREATE TABLE orders(
order_no INT PRIMARY KEY,
book_id INT,
qty INT,
FOREIGN KEY(book_id) REFERENCES catalog(book_id)
);
```

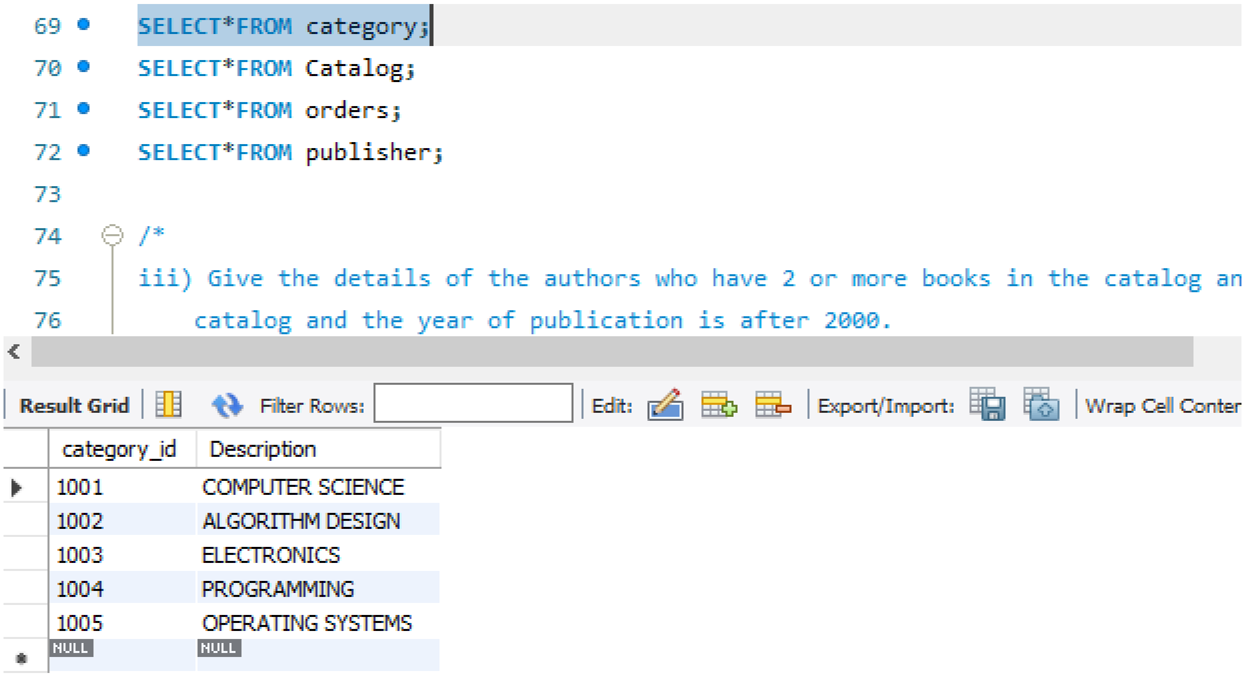


Tables_in_book_db
author
catalog
category
orders
publisher



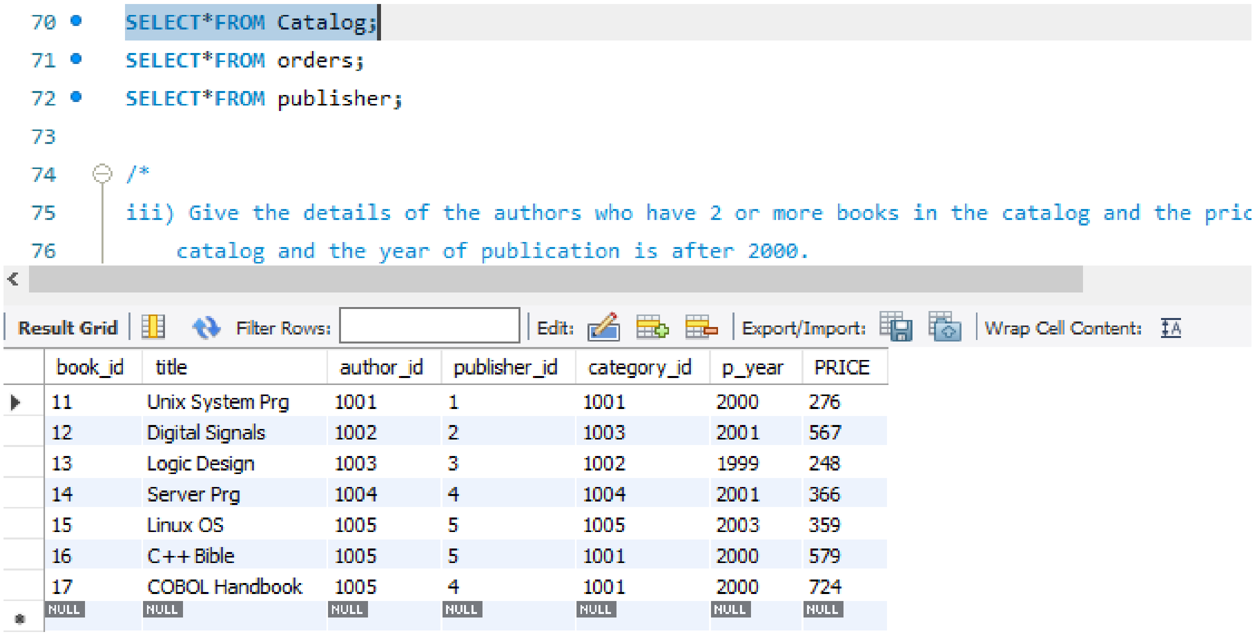
```
SELECT*FROM AUTHOR;
SELECT*FROM category;
SELECT*FROM Catalog;
SELECT*FROM orders;
SELECT*FROM publisher;

/*
iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books ir
    catalog and the year of publication is after 2000.
```

author_id	a_name	city	country
1001	TERAS CHAN	CA	USA
1002	STEVENS	ZOMBI	UGANDA
1003	M MANO	CAIR	CANADA
1004	KARTHIK B.P.	NEW YORK	USA
1005	WILLIAM STALLINGS	LAS VEGAS	USA
NULL	NULL	NULL	NULL



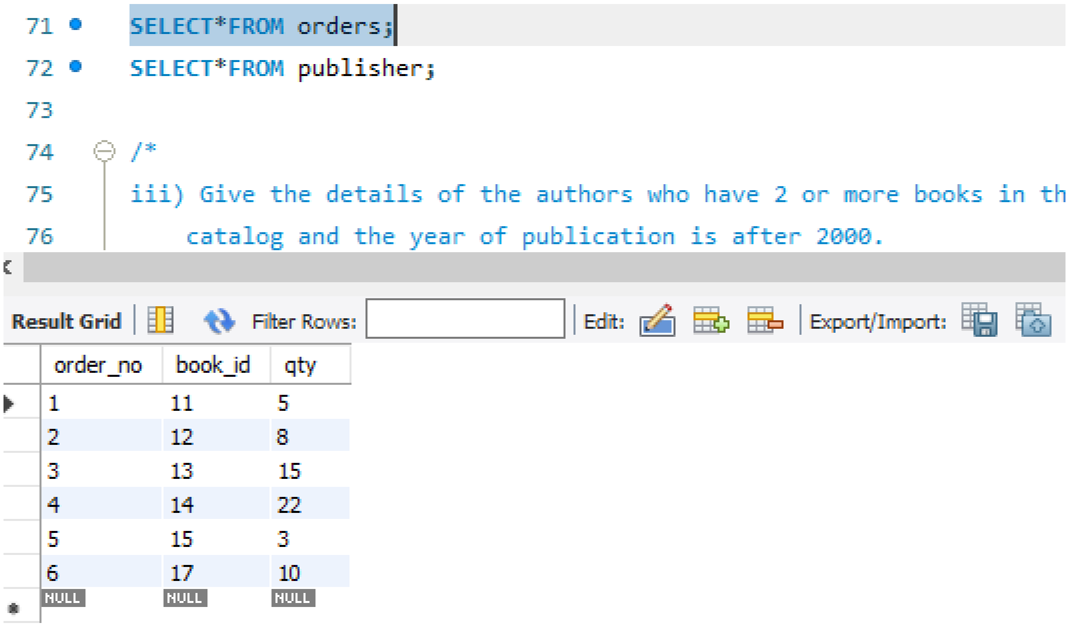
```
SELECT*FROM category;
SELECT*FROM Catalog;
SELECT*FROM orders;
SELECT*FROM publisher;

/*
iii) Give the details of the authors who have 2 or more books in the catalog an
    catalog and the year of publication is after 2000.
```

category_id	Description
1001	COMPUTER SCIENCE
1002	ALGORITHM DESIGN
1003	ELECTRONICS
1004	PROGRAMMING
1005	OPERATING SYSTEMS
NULL	NULL



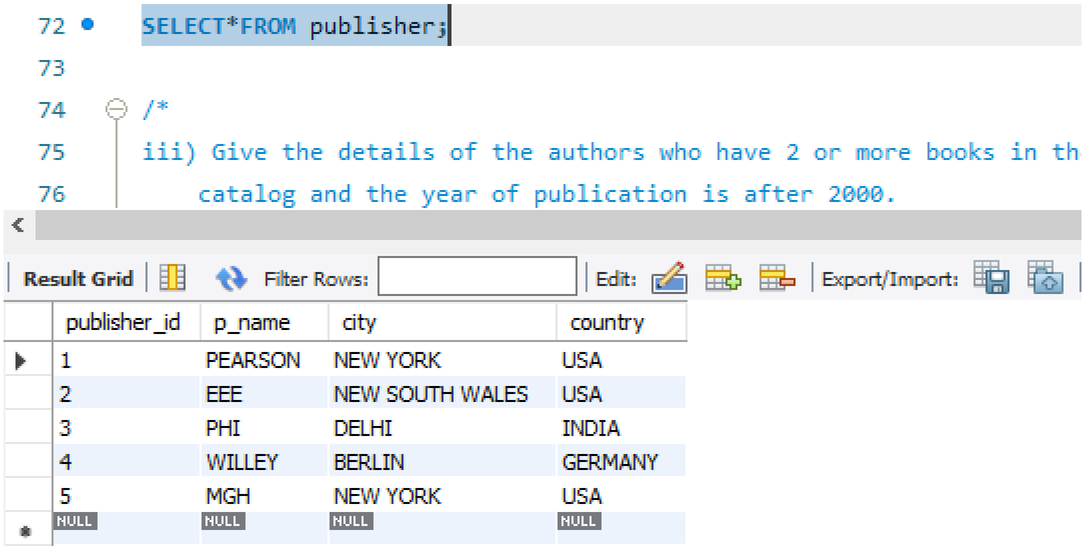
```
SELECT*FROM Catalog;
SELECT*FROM orders;
SELECT*FROM publisher;

/*
iii) Give the details of the authors who have 2 or more books in the catalog and the pric
    catalog and the year of publication is after 2000.
```

book_id	title	author_id	publisher_id	category_id	p_year	PRICE
11	Unix System Prg	1001	1	1001	2000	276
12	Digital Signals	1002	2	1003	2001	567
13	Logic Design	1003	3	1002	1999	248
14	Server Prg	1004	4	1004	2001	366
15	Linux OS	1005	5	1005	2003	359
16	C++ Bible	1005	5	1001	2000	579
17	COBOL Handbook	1005	4	1001	2000	724
NULL	NULL	NULL	NULL	NULL	NULL	NULL



```
SELECT*FROM orders;
SELECT*FROM publisher;

/*
iii) Give the details of the authors who have 2 or more books in th
    catalog and the year of publication is after 2000.
```

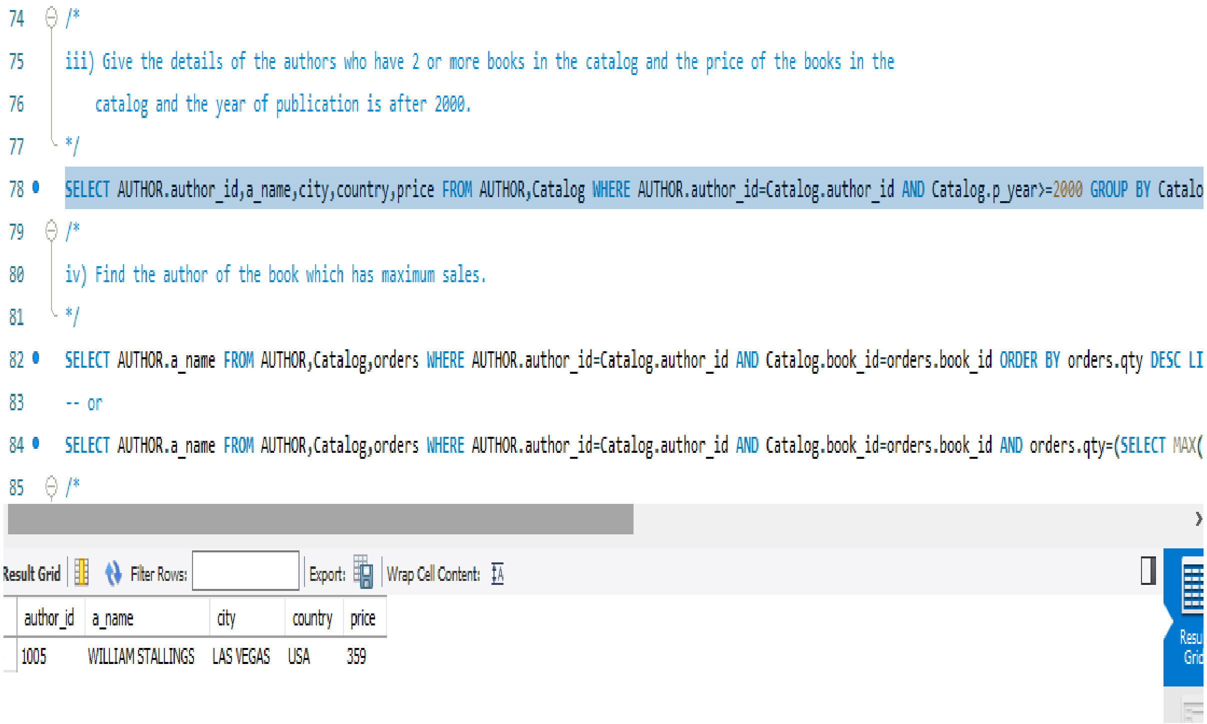
order_no	book_id	qty
1	11	5
2	12	8
3	13	15
4	14	22
5	15	3
6	17	10
NULL	NULL	NULL



```
SELECT*FROM publisher;

/*
iii) Give the details of the authors who have 2 or more books in th
    catalog and the year of publication is after 2000.
```

publisher_id	p_name	city	country
1	PEARSON	NEW YORK	USA
2	EEE	NEW SOUTH WALES	USA
3	PHI	DELHI	INDIA
4	WILLEY	BERLIN	GERMANY
5	MGH	NEW YORK	USA
NULL	NULL	NULL	NULL

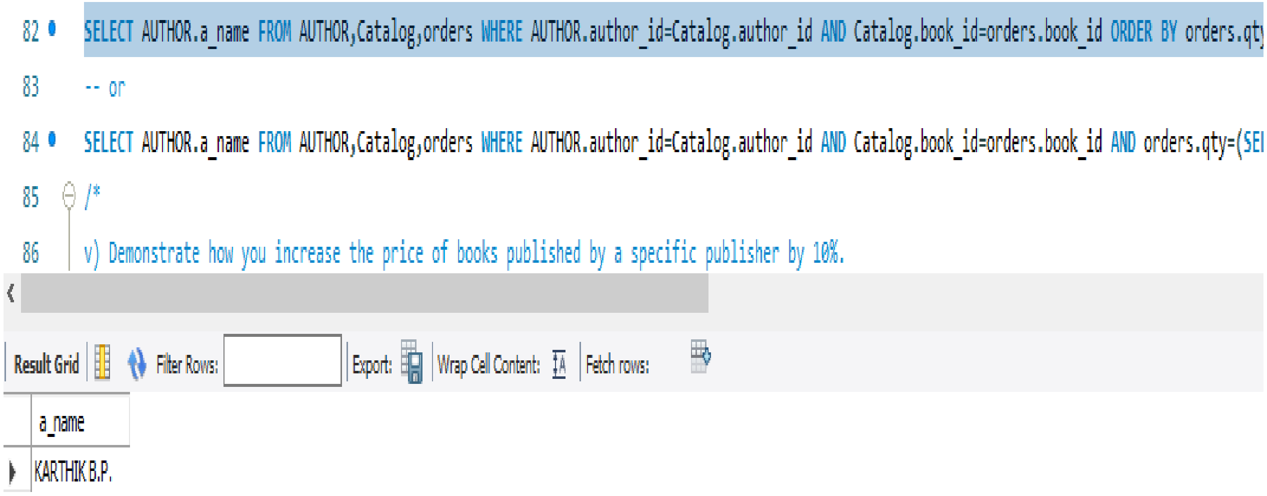
iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books in the catalog and the year of publication is after 2000.



```
/*
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.
*/
SELECT AUTHOR.author_id,a_name,city,country,price FROM AUTHOR,Catalog WHERE AUTHOR.author_id=Catalog.author_id AND Catalog.p_year>=2000 GROUP BY Catalo
/*
iv) Find the author of the book which has maximum sales.
*/
SELECT AUTHOR.a_name FROM AUTHOR,Catalog,orders WHERE AUTHOR.author_id=Catalog.author_id AND Catalog.book_id=orders.book_id ORDER BY orders.qty DESC LI
-- or
SELECT AUTHOR.a_name FROM AUTHOR,Catalog,orders WHERE AUTHOR.author_id=Catalog.author_id AND Catalog.book_id=orders.book_id AND orders.qty=(SELECT MAX(
/*
```

author_id	a_name	city	country	price
1005	WILLIAM STALLINGS	LAS VEGAS	USA	359

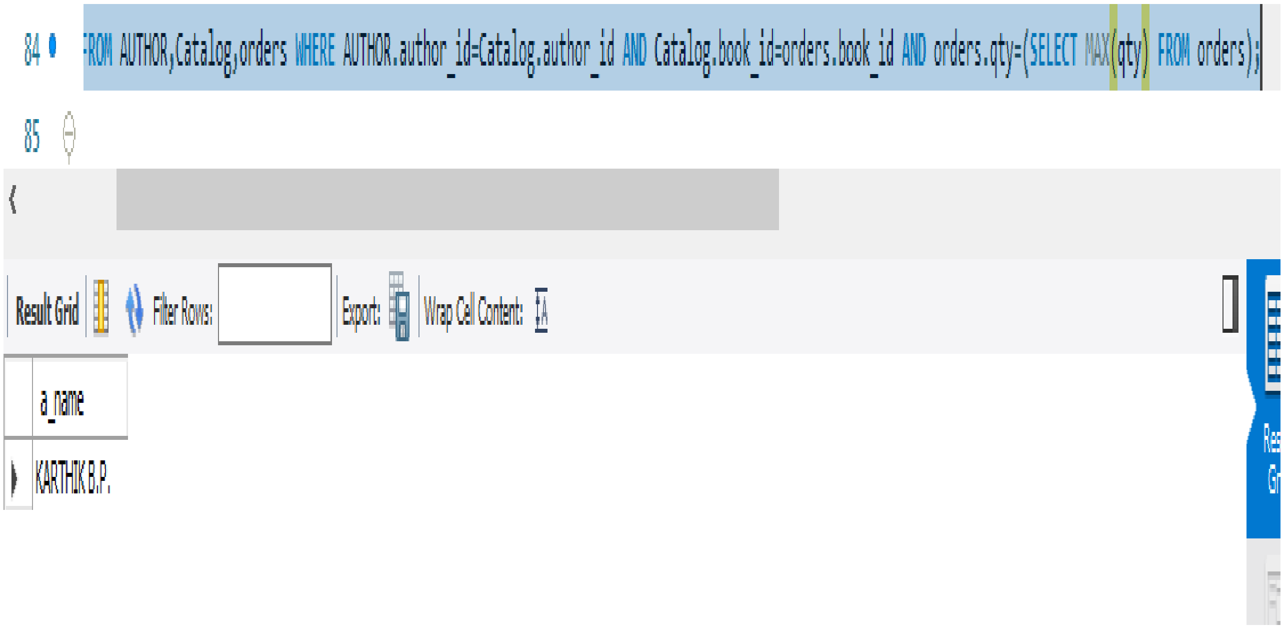
iv) Find the author of the book which has maximum sales.



```
SELECT AUTHOR.a_name FROM AUTHOR,Catalog,orders WHERE AUTHOR.author_id=Catalog.author_id AND Catalog.book_id=orders.book_id ORDER BY orders.qty
-- or
SELECT AUTHOR.a_name FROM AUTHOR,Catalog,orders WHERE AUTHOR.author_id=Catalog.author_id AND Catalog.book_id=orders.book_id AND orders.qty=(SEL
/*
v) Demonstrate how you increase the price of books published by a specific publisher by 10%.
```

```
a_name
KARTHIK B.P.
```

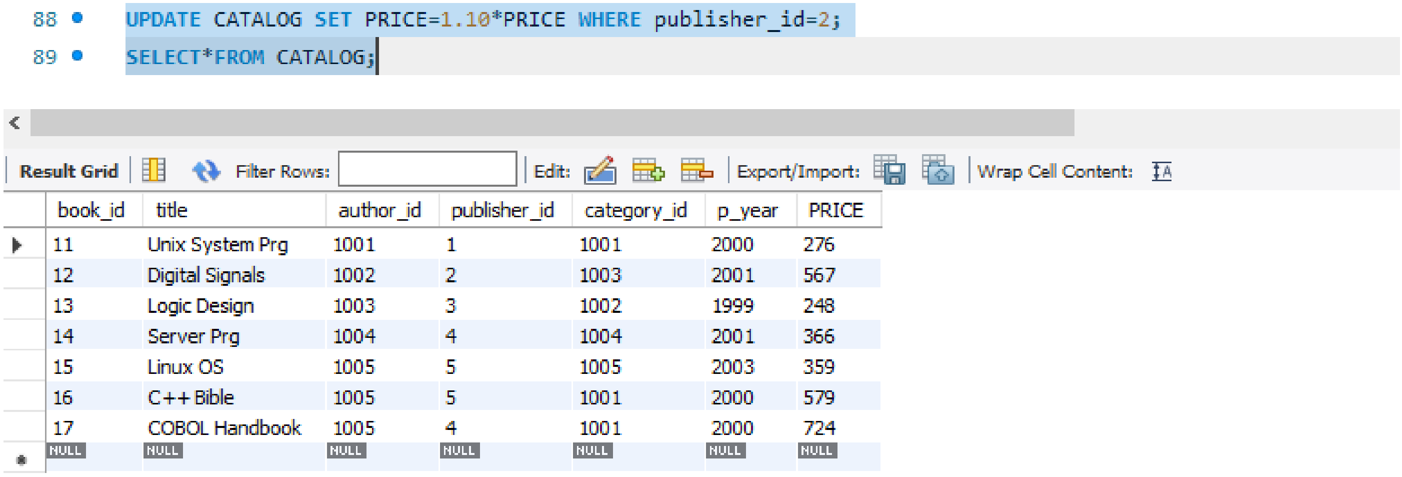
BY SECOND QUERY



```
ROM AUTHOR,Catalog,orders WHERE AUTHOR.author_id=Catalog.author_id AND Catalog.book_id=orders.book_id AND orders.qty=(SELECT MAX(qty) FROM orders);
```

```
a_name
KARTHIK B.P.
```

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



```sql
UPDATE CATALOG SET PRICE=1.10*PRICE WHERE publisher_id=2;
SELECT*FROM CATALOG;
```

book_id	title	author_id	publisher_id	category_id	p_year	PRICE
11	Unix System Prg	1001	1	1001	2000	276
12	Digital Signals	1002	2	1003	2001	567
13	Logic Design	1003	3	1002	1999	248
14	Server Prg	1004	4	1004	2001	366
15	Linux OS	1005	5	1005	2003	359
16	C++ Bible	1005	5	1001	2000	579
17	COBOL Handbook	1005	4	1001	2000	724
NULL	NULL	NULL	NULL	NULL	NULL	NULL

WEEK-3 DBMS LAB

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
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.

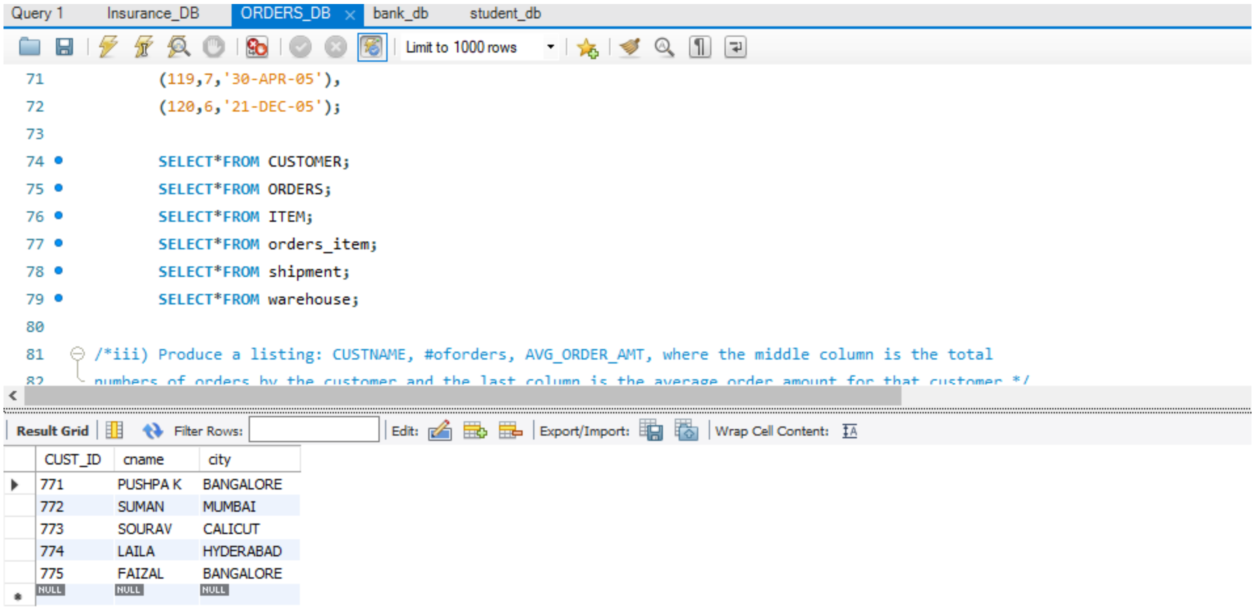
iv) List the order# for orders that were shipped from all warehouses that the company has in a specific city.

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

OUTPUT:

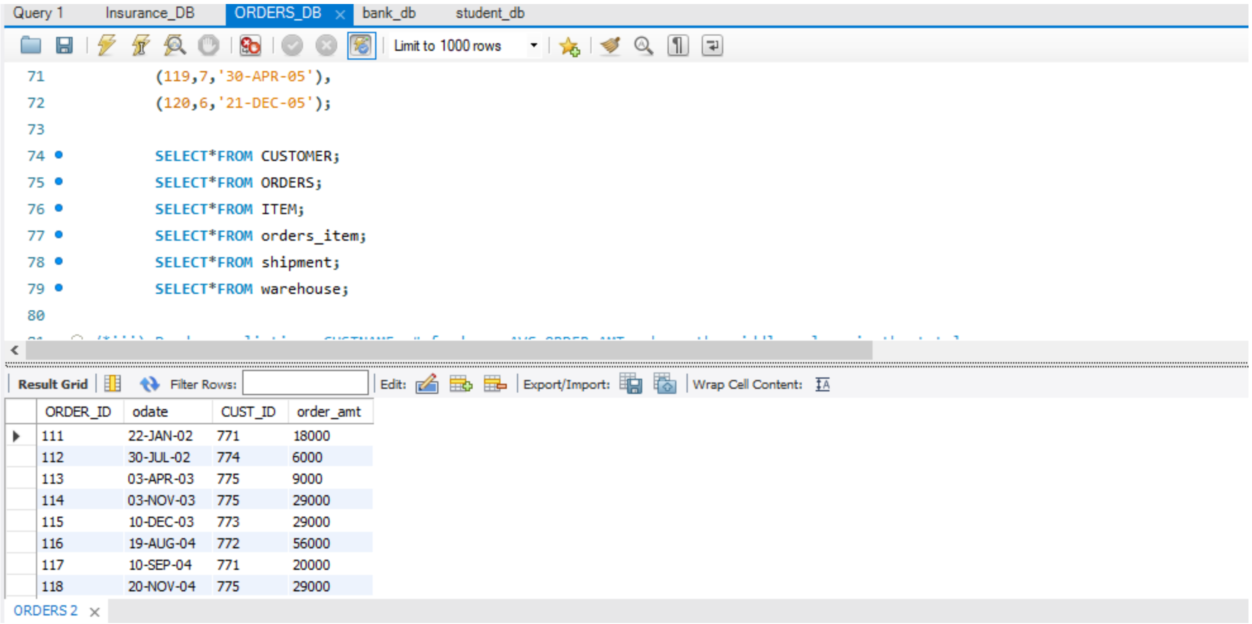
i) Create the above tables by properly specifying the primary keys and the foreign keys and the foreign keys.

ii) Enter at least five tuples for each relation.



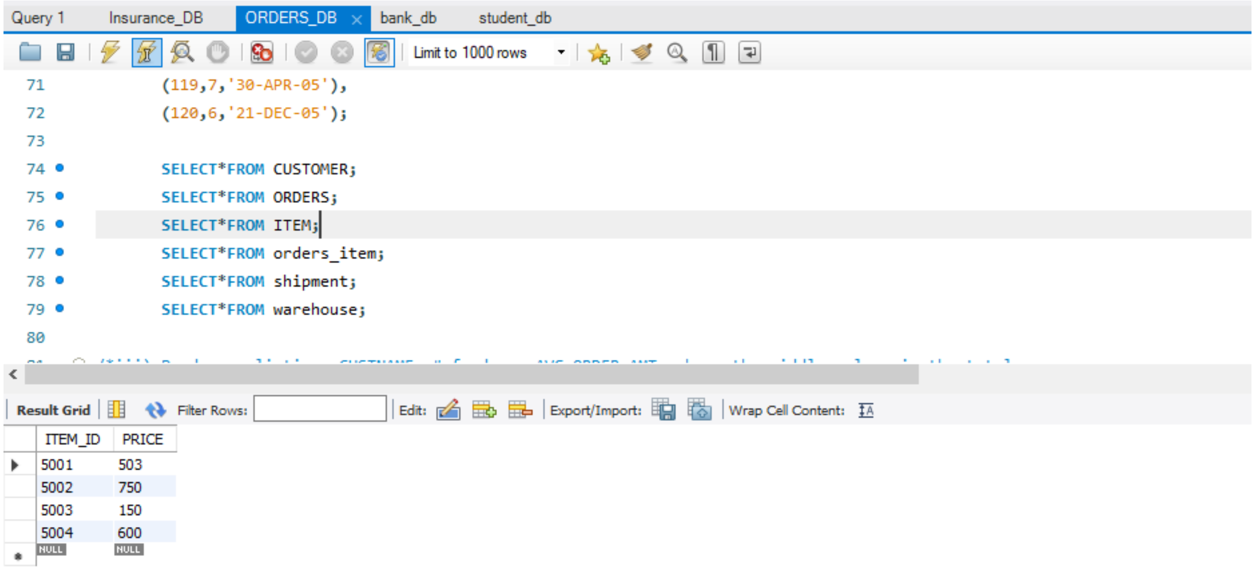
```
            (119,7,'30-APR-05'),
            (120,6,'21-DEC-05');

            SELECT*FROM CUSTOMER;
            SELECT*FROM ORDERS;
            SELECT*FROM ITEM;
            SELECT*FROM orders_item;
            SELECT*FROM shipment;
            SELECT*FROM warehouse;

    /*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 */
```

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



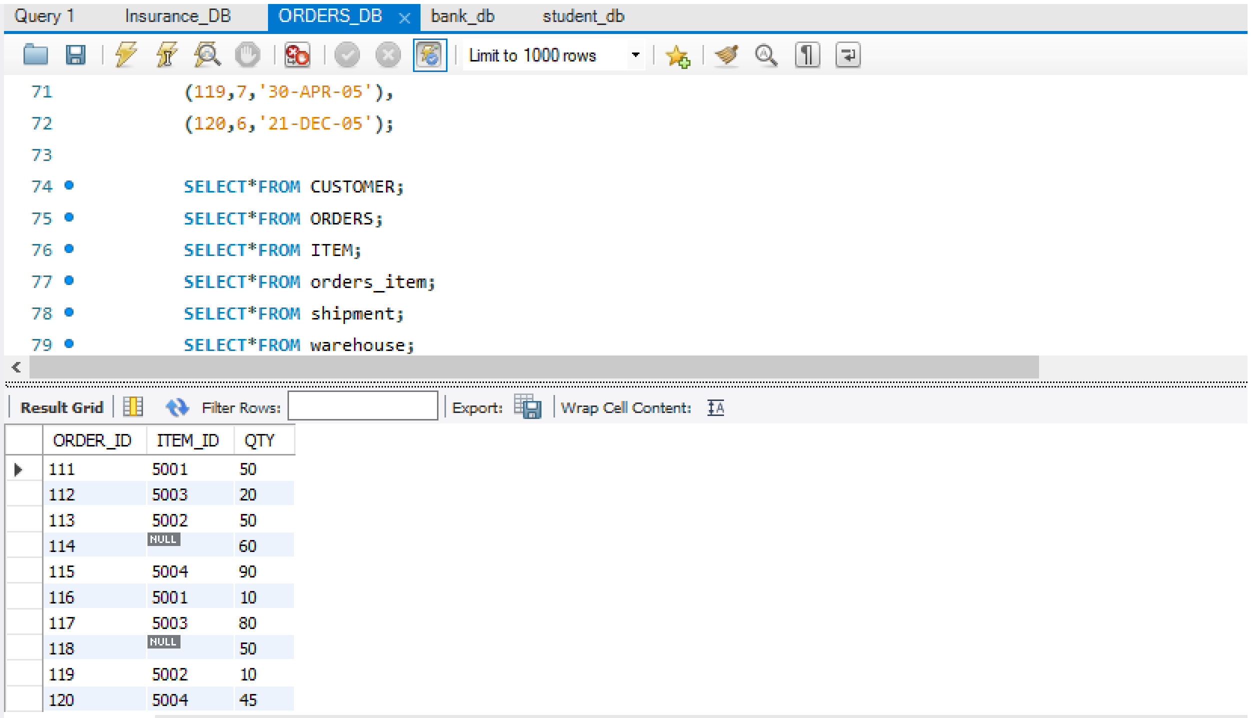
```
            (119,7,'30-APR-05'),
            (120,6,'21-DEC-05');

            SELECT*FROM CUSTOMER;
            SELECT*FROM ORDERS;
            SELECT*FROM ITEM;
            SELECT*FROM orders_item;
            SELECT*FROM shipment;
            SELECT*FROM warehouse;
```

ORDER_ID	odate	CUST_ID	order_amt
111	22-JAN-02	771	18000
112	30-JUL-02	774	6000
113	03-APR-03	775	9000
114	03-NOV-03	775	29000
115	10-DEC-03	773	29000
116	19-AUG-04	772	56000
117	10-SEP-04	771	20000
118	20-NOV-04	775	29000



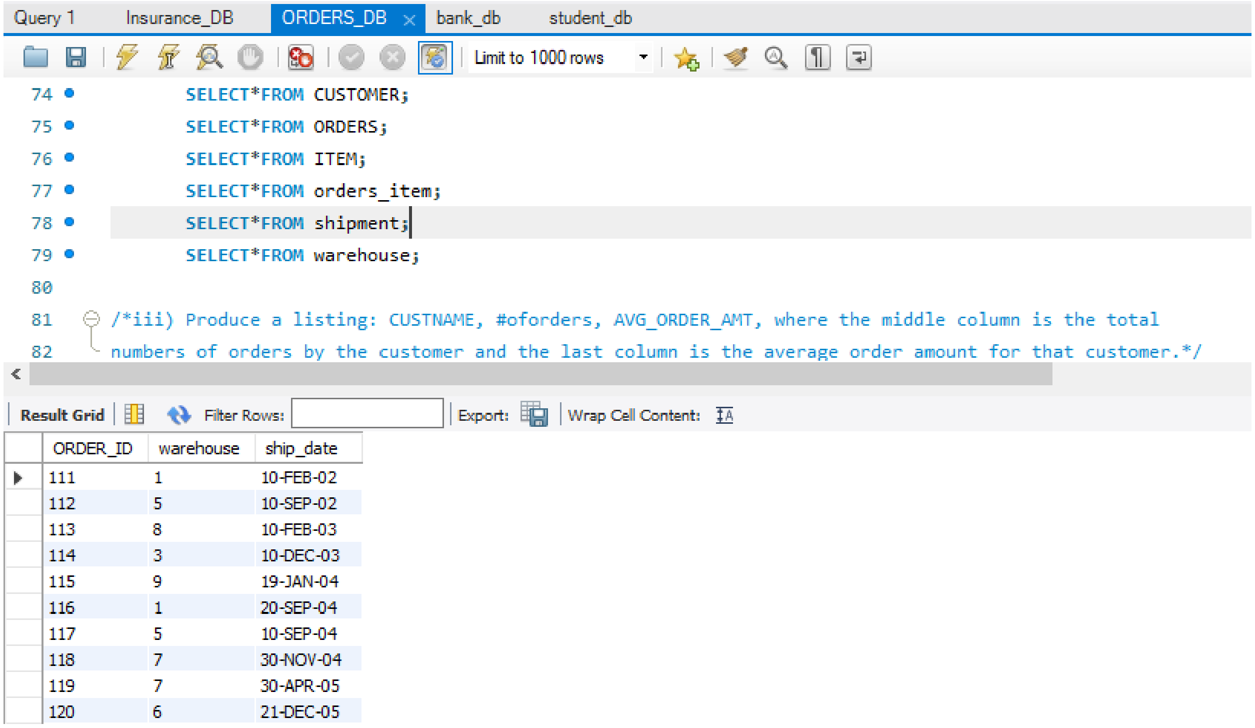
```
        (119,7,'30-APR-05'),
        (120,6,'21-DEC-05');

        SELECT*FROM CUSTOMER;
        SELECT*FROM ORDERS;
        SELECT*FROM ITEM;
        SELECT*FROM orders_item;
        SELECT*FROM shipment;
        SELECT*FROM warehouse;
```

ITEM_ID	PRICE
5001	503
5002	750
5003	150
5004	600
NULL	NULL



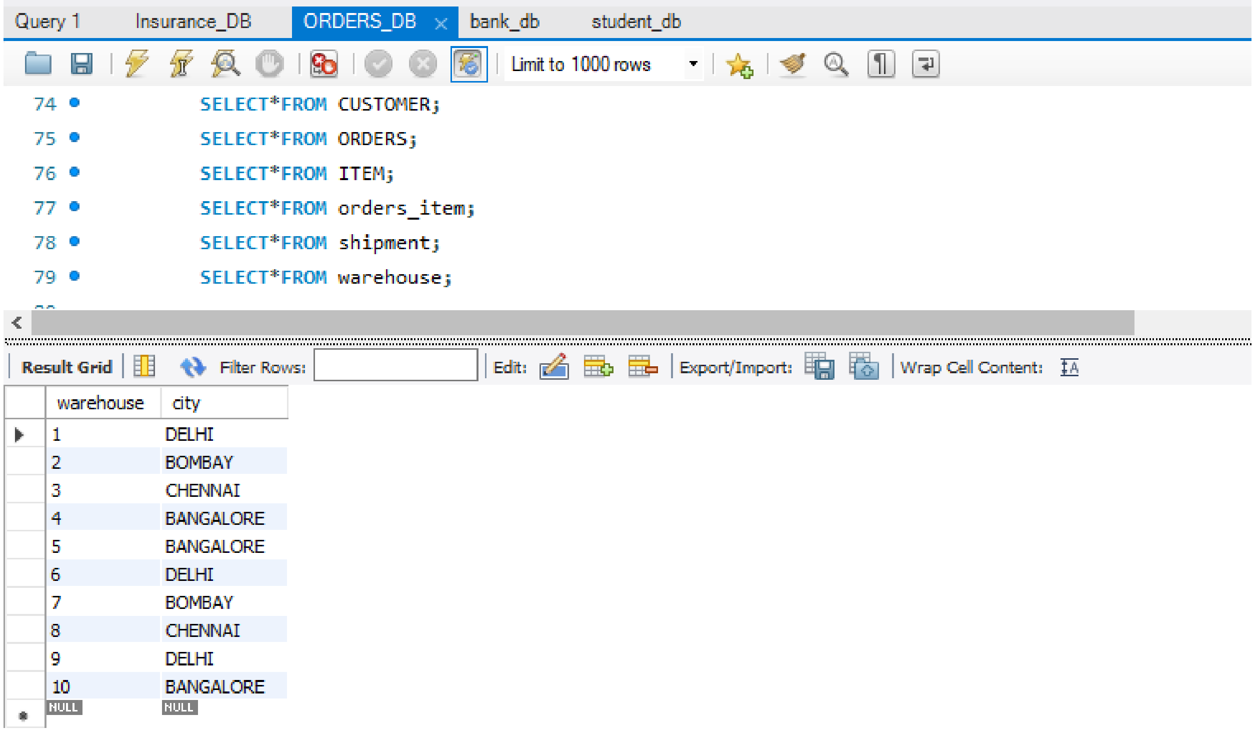
```
        (119,7,'30-APR-05'),
        (120,6,'21-DEC-05');

        SELECT*FROM CUSTOMER;
        SELECT*FROM ORDERS;
        SELECT*FROM ITEM;
        SELECT*FROM orders_item;
        SELECT*FROM shipment;
        SELECT*FROM warehouse;
```

ORDER_ID	ITEM_ID	QTY
111	5001	50
112	5003	20
113	5002	50
114	NULL	60
115	5004	90
116	5001	10
117	5003	80
118	NULL	50
119	5002	10
120	5004	45



```sql
        SELECT*FROM CUSTOMER;
        SELECT*FROM ORDERS;
        SELECT*FROM ITEM;
        SELECT*FROM orders_item;
        SELECT*FROM shipment;
        SELECT*FROM warehouse;

/*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.*/
```

ORDER_ID	warehouse	ship_date
111	1	10-FEB-02
112	5	10-SEP-02
113	8	10-FEB-03
114	3	10-DEC-03
115	9	19-JAN-04
116	1	20-SEP-04
117	5	10-SEP-04
118	7	30-NOV-04
119	7	30-APR-05
120	6	21-DEC-05

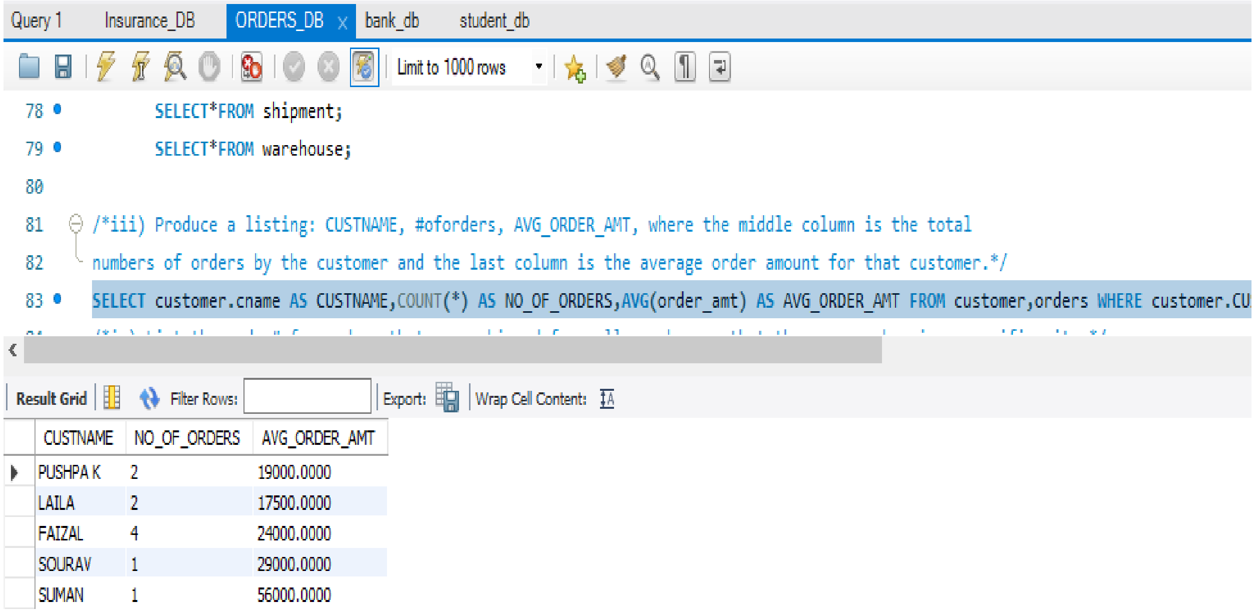


```sql
        SELECT*FROM CUSTOMER;
        SELECT*FROM ORDERS;
        SELECT*FROM ITEM;
        SELECT*FROM orders_item;
        SELECT*FROM shipment;
        SELECT*FROM warehouse;
```

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

warehouse 6 ×

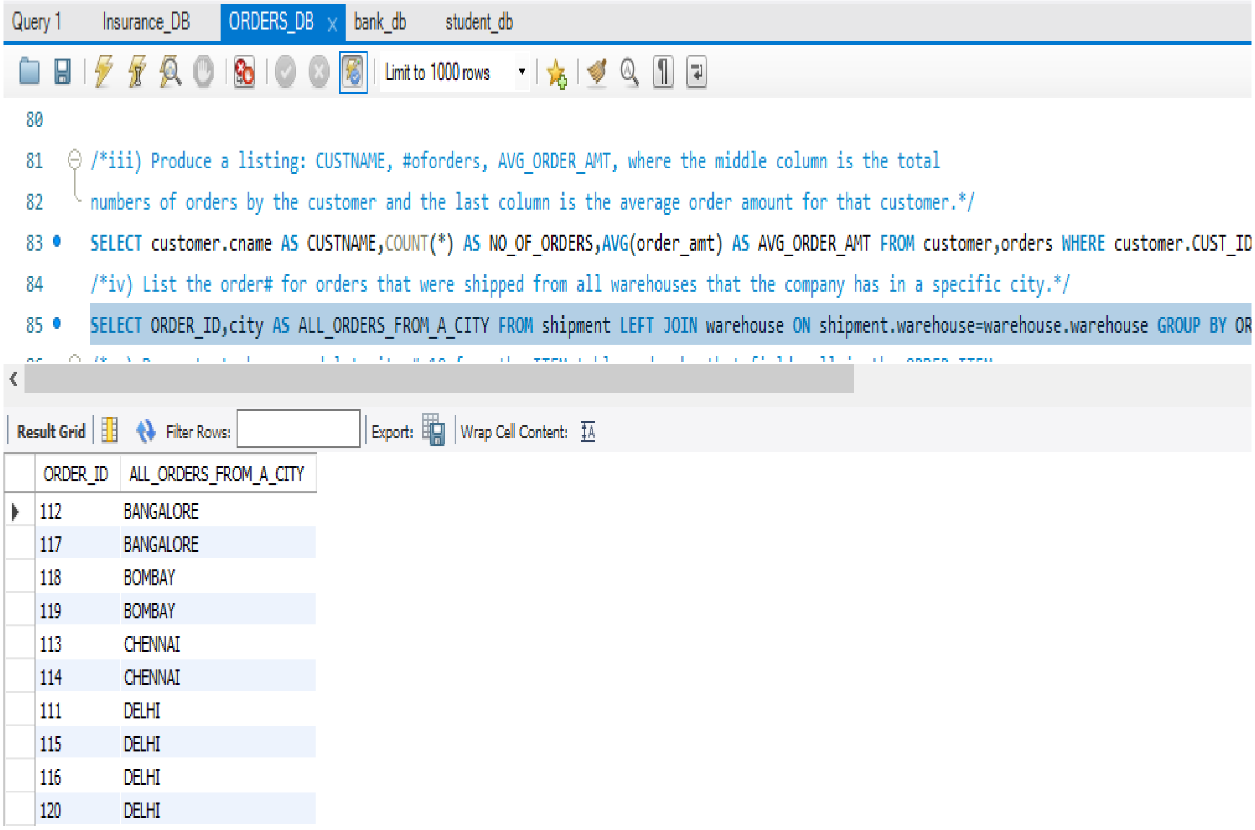
iii) Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column is the total numbers of orders by the customer and the last column is the average order amount for that customer.



```
        SELECT*FROM shipment;
        SELECT*FROM warehouse;

/*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.*/
SELECT customer.cname AS CUSTNAME,COUNT(*) AS NO_OF_ORDERS,AVG(order_amt) AS AVG_ORDER_AMT FROM customer,orders WHERE customer.CU
```

CUSTNAME	NO_OF_ORDERS	AVG_ORDER_AMT
PUSHPA K	2	19000.0000
LAILA	2	17500.0000
FAIZAL	4	24000.0000
SOURAV	1	29000.0000
SUMAN	1	56000.0000

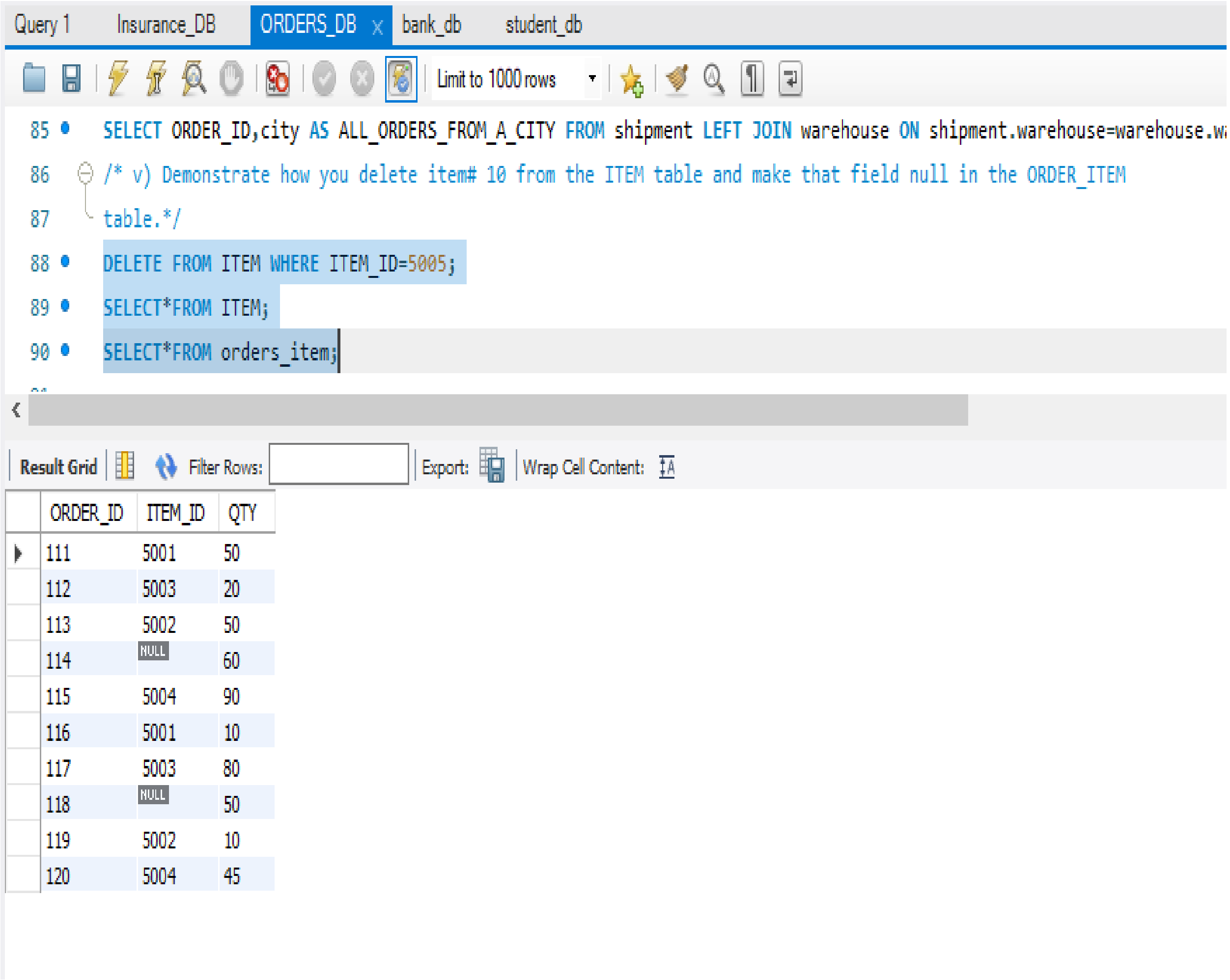
iv) List the order# for orders that were shipped from all warehouses that the company has in a specific city.



```
/*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.*/
SELECT customer.cname AS CUSTNAME,COUNT(*) AS NO_OF_ORDERS,AVG(order_amt) AS AVG_ORDER_AMT FROM customer,orders WHERE customer.CUST_ID
/*iv) List the order# for orders that were shipped from all warehouses that the company has in a specific city.*/
SELECT ORDER_ID,city AS ALL_ORDERS_FROM_A_CITY FROM shipment LEFT JOIN warehouse ON shipment.warehouse=warehouse.warehouse GROUP BY OR
```

ORDER_ID	ALL_ORDERS_FROM_A_CITY
112	BANGALORE
117	BANGALORE
118	BOMBAY
119	BOMBAY
113	CHENNAI
114	CHENNAI
111	DELHI
115	DELHI
116	DELHI
120	DELHI

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



```
SELECT ORDER_ID,city AS ALL_ORDERS_FROM_A_CITY FROM shipment LEFT JOIN warehouse ON shipment.warehouse=warehouse.wa
/* v) 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_ID=5005;
SELECT*FROM ITEM;
SELECT*FROM orders_item;
```

ORDER_ID	ITEM_ID	QTY
111	5001	50
112	5003	20
113	5002	50
114	NULL	60
115	5004	90
116	5001	10
117	5003	80
118	NULL	50
119	5002	10
120	5004	45

*******************LAB 3 ENDS*******************

WEEK-4 DBMS LAB

Consider the following database for a banking enterprise.

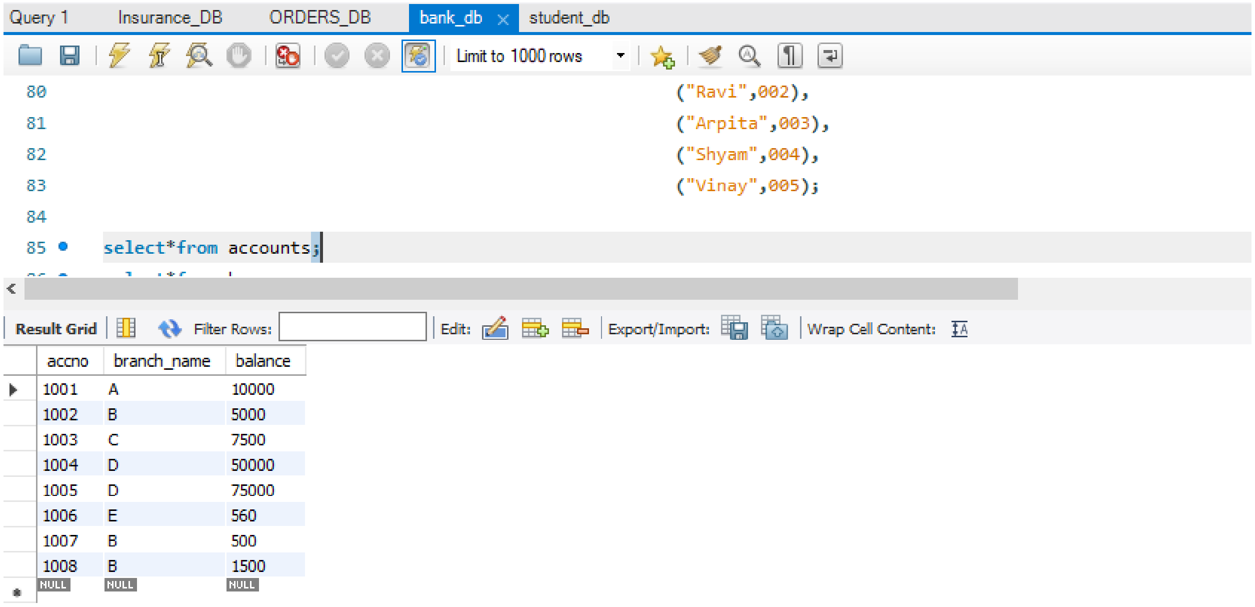
BRANCH (branch-name: String, branch-city: String, assets: real)
ACCOUNTS (accno: int, branch-name: String, balance: real)
DEPOSITOR (customer-name: String, customer-street: String, customer-city: String)
LOAN (loan-number: int, branch-name: String, amount: real)
BORROWER (customer-name: String, loan-number: 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.**

OUTPUT:

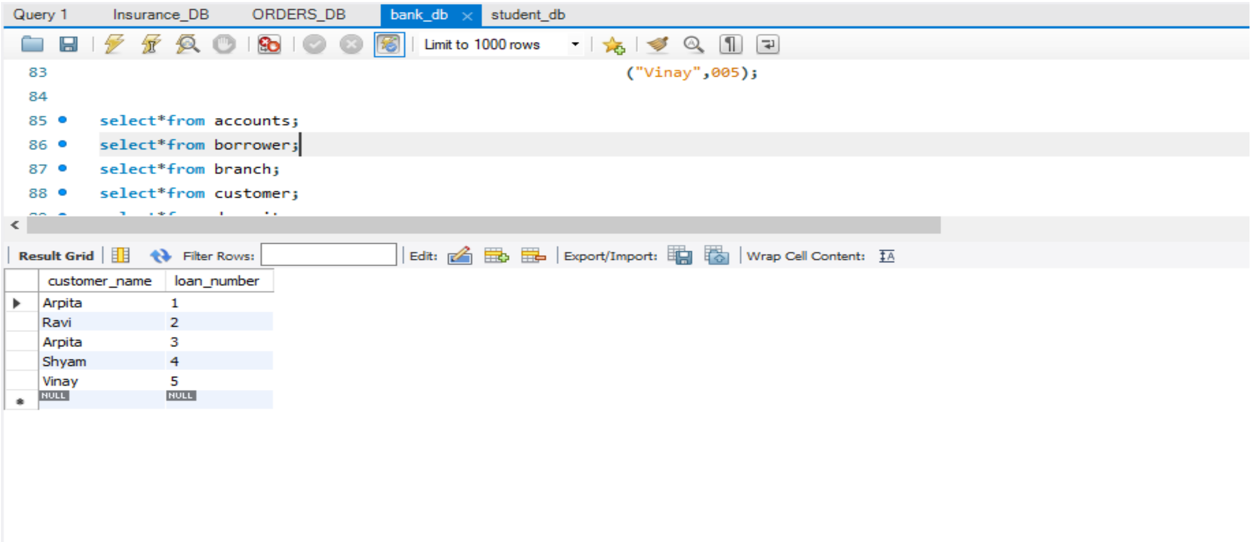
i. **Create the above tables by properly specifying the primary keys and the foreign keys.**

ii. **Enter at least five tuples for each relation.**



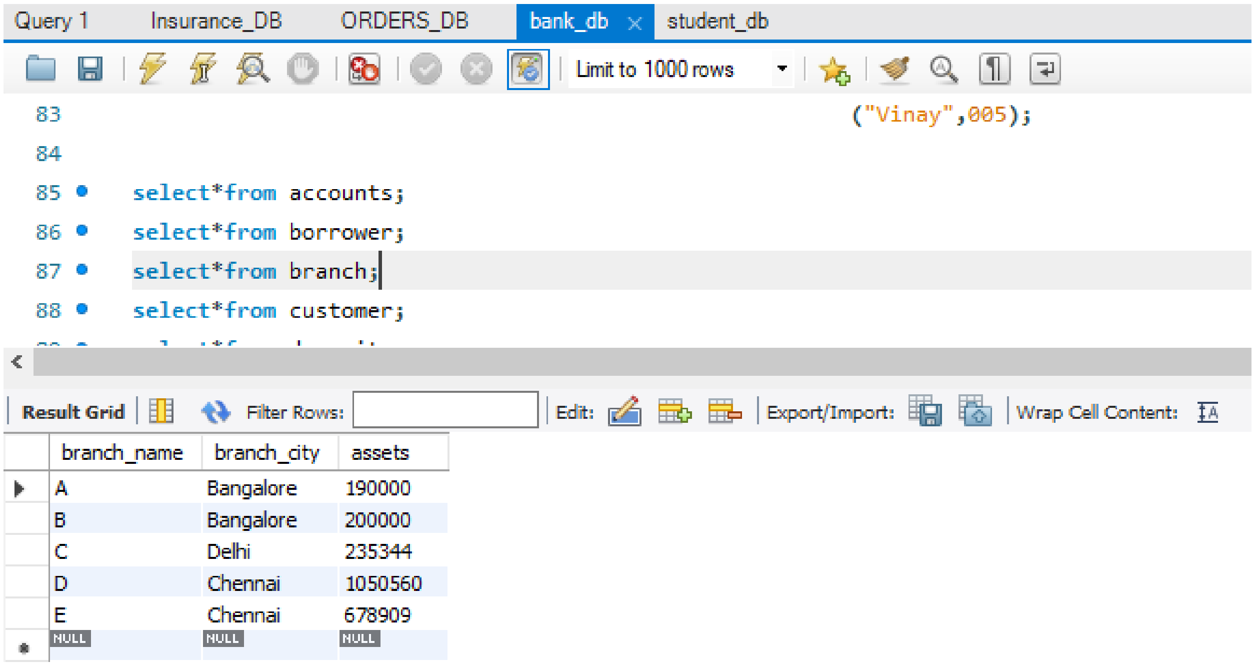
```
                                  ("Ravi",002),
                                  ("Arpita",003),
                                  ("Shyam",004),
                                  ("Vinay",005);

select*from accounts;
```

accno	branch_name	balance
1001	A	10000
1002	B	5000
1003	C	7500
1004	D	50000
1005	D	75000
1006	E	560
1007	B	500
1008	B	1500
NULL	NULL	NULL



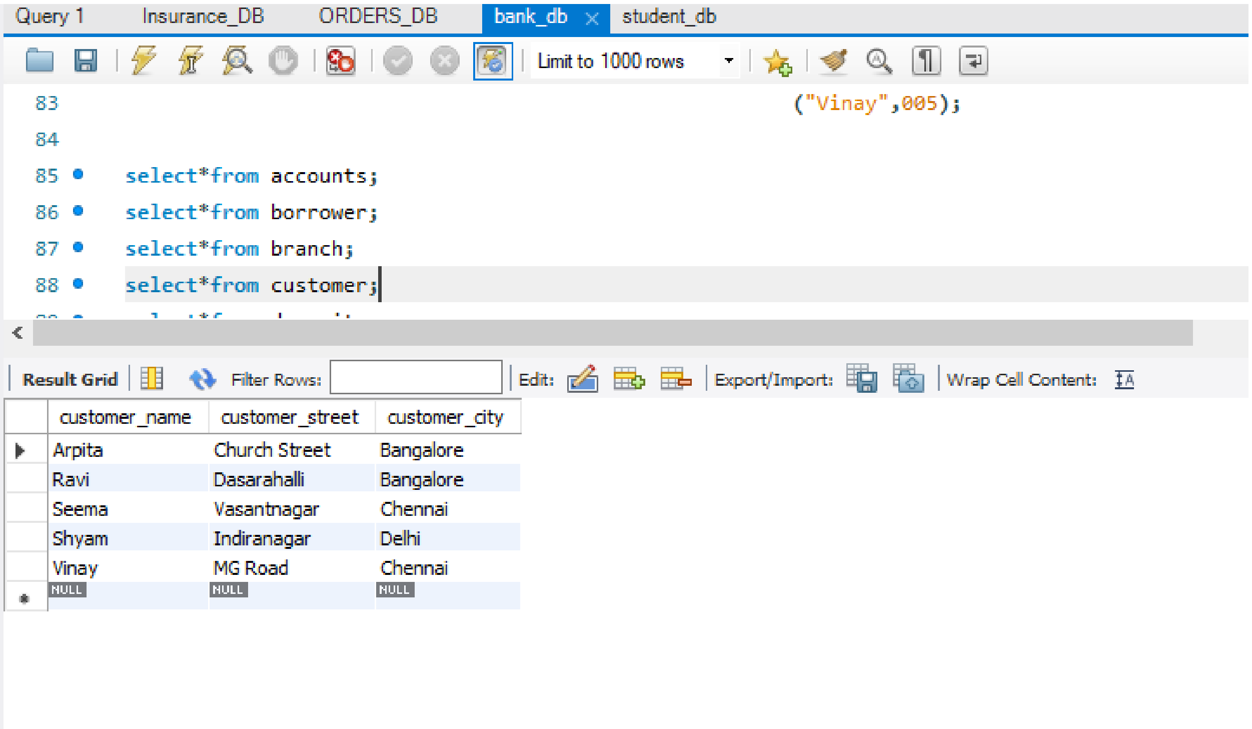
```
                                  ("Vinay",005);

select*from accounts;
select*from borrower;
select*from branch;
select*from customer;
```

customer_name	loan_number
Arpita	1
Ravi	2
Arpita	3
Shyam	4
Vinay	5
NULL	NULL



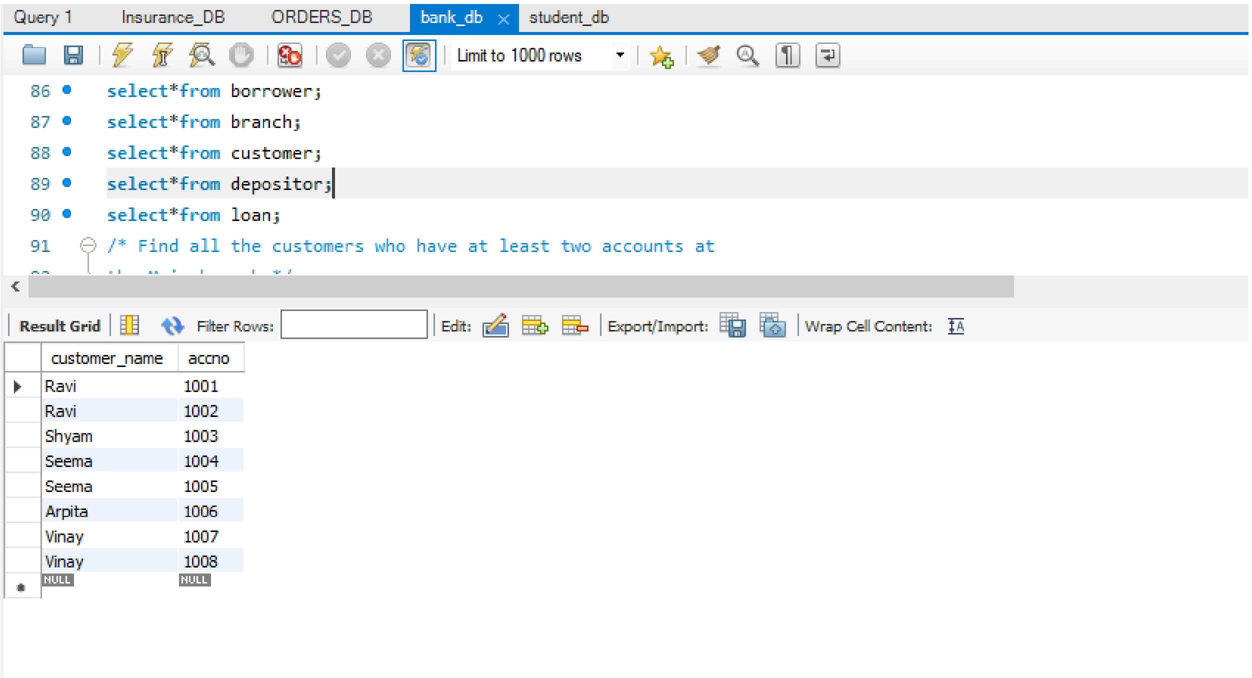
```
                                                    ("Vinay",005);

select*from accounts;
select*from borrower;
select*from branch;
select*from customer;
```

branch_name	branch_city	assets
A	Bangalore	190000
B	Bangalore	200000
C	Delhi	235344
D	Chennai	1050560
E	Chennai	678909
NULL	NULL	NULL



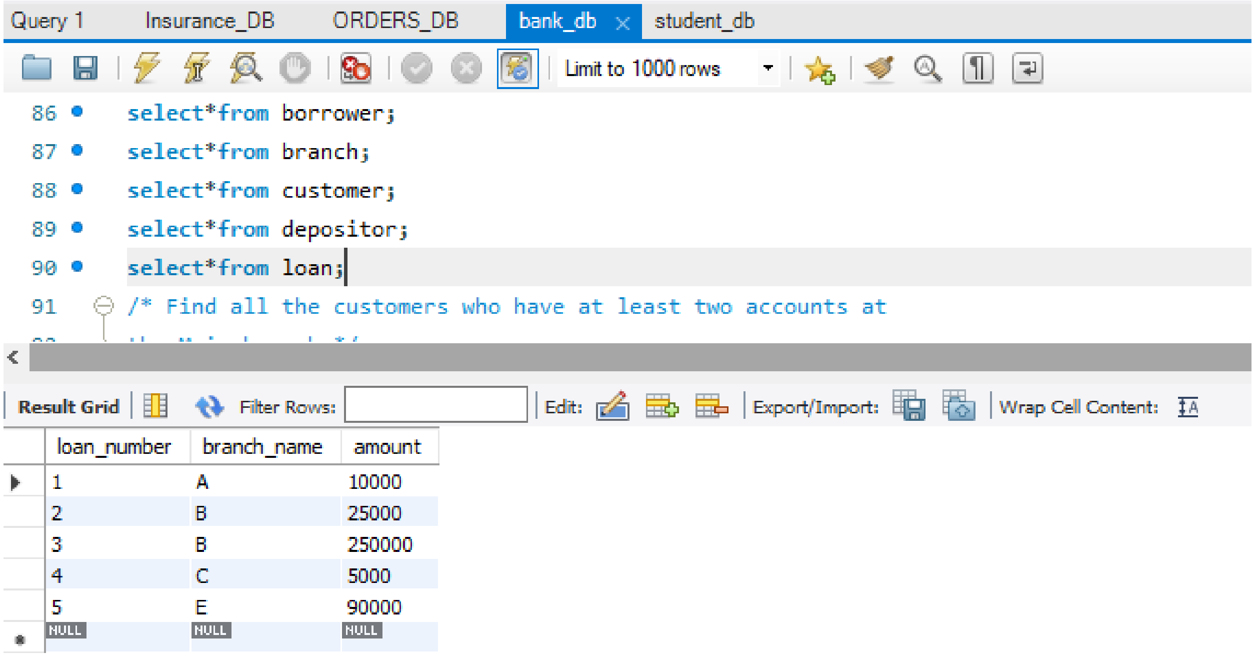
```
                                                    ("Vinay",005);

select*from accounts;
select*from borrower;
select*from branch;
select*from customer;
```

customer_name	customer_street	customer_city
Arpita	Church Street	Bangalore
Ravi	Dasarahalli	Bangalore
Seema	Vasantnagar	Chennai
Shyam	Indiranagar	Delhi
Vinay	MG Road	Chennai
NULL	NULL	NULL



```
select*from borrower;
select*from branch;
select*from customer;
select*from depositor;
select*from loan;
/* Find all the customers who have at least two accounts at
```

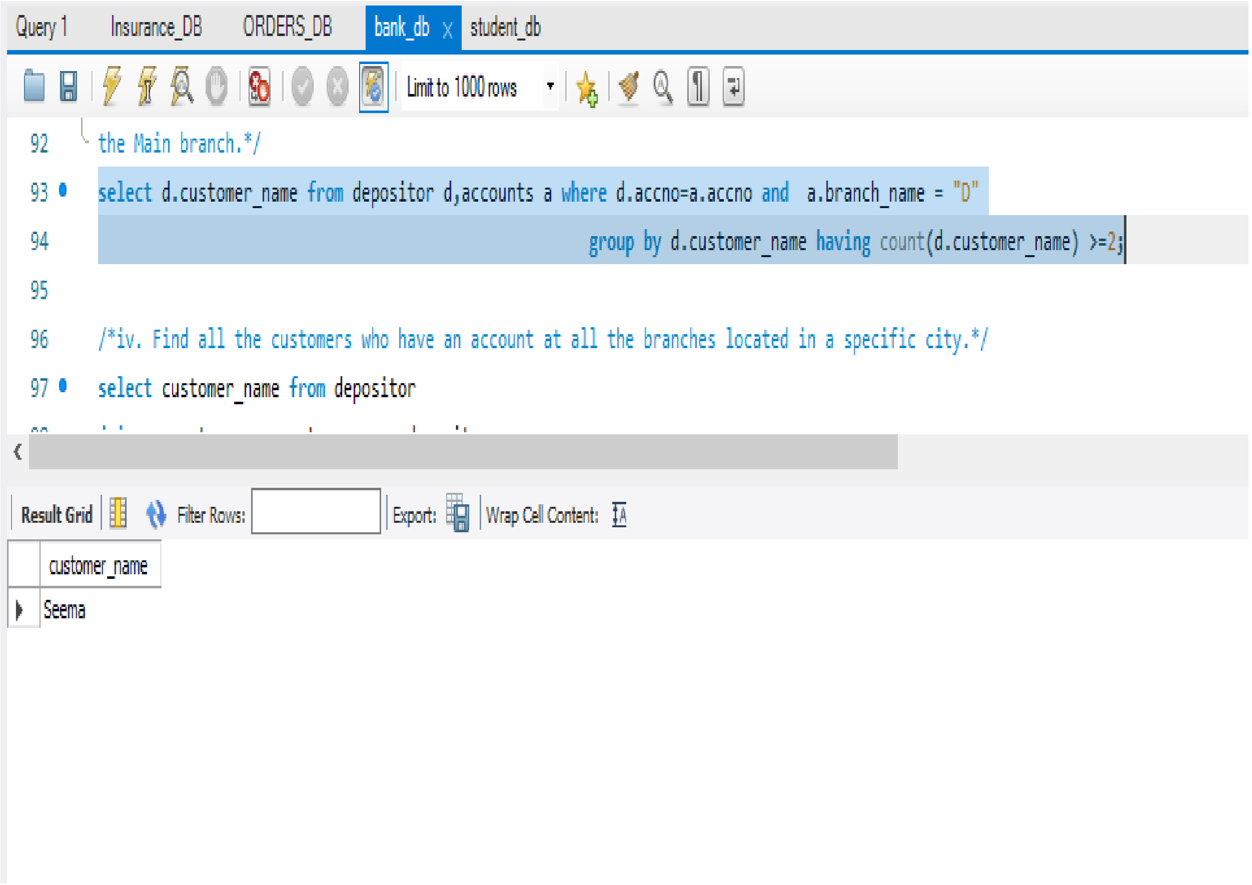
customer_name	accno
Ravi	1001
Ravi	1002
Shyam	1003
Seema	1004
Seema	1005
Arpita	1006
Vinay	1007
Vinay	1008
NULL	NULL



```
select*from borrower;
select*from branch;
select*from customer;
select*from depositor;
select*from loan;
/* Find all the customers who have at least two accounts at
```

loan_number	branch_name	amount
1	A	10000
2	B	25000
3	B	250000
4	C	5000
5	E	90000
NULL	NULL	NULL

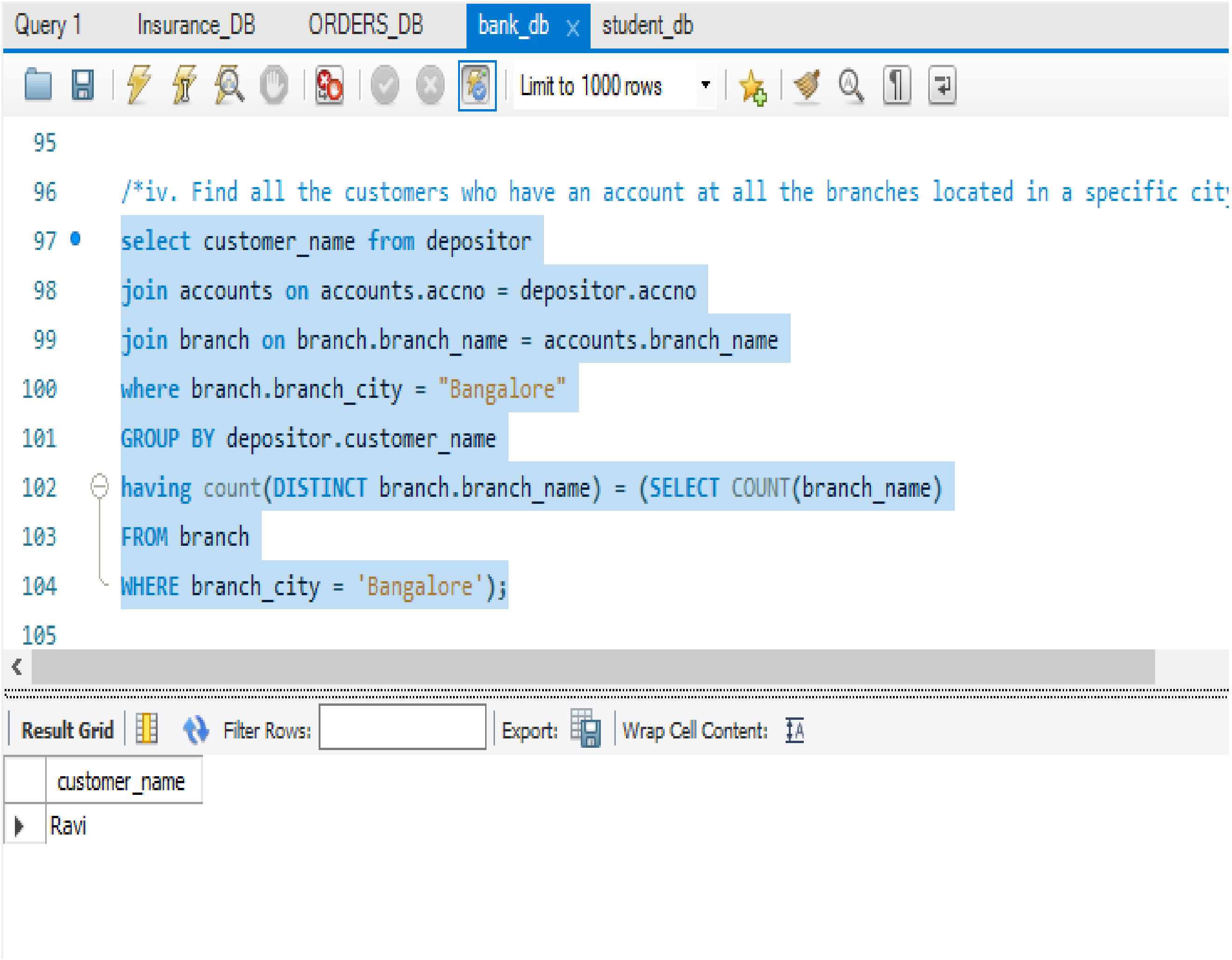
iii.Find all the customers who have at least two accounts at the *Main* branch.



```
the Main branch.*/
select d.customer_name from depositor d,accounts a where d.accno=a.accno and  a.branch_name = "D"
                                                  group by d.customer_name having count(d.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
```

customer_name
Seema

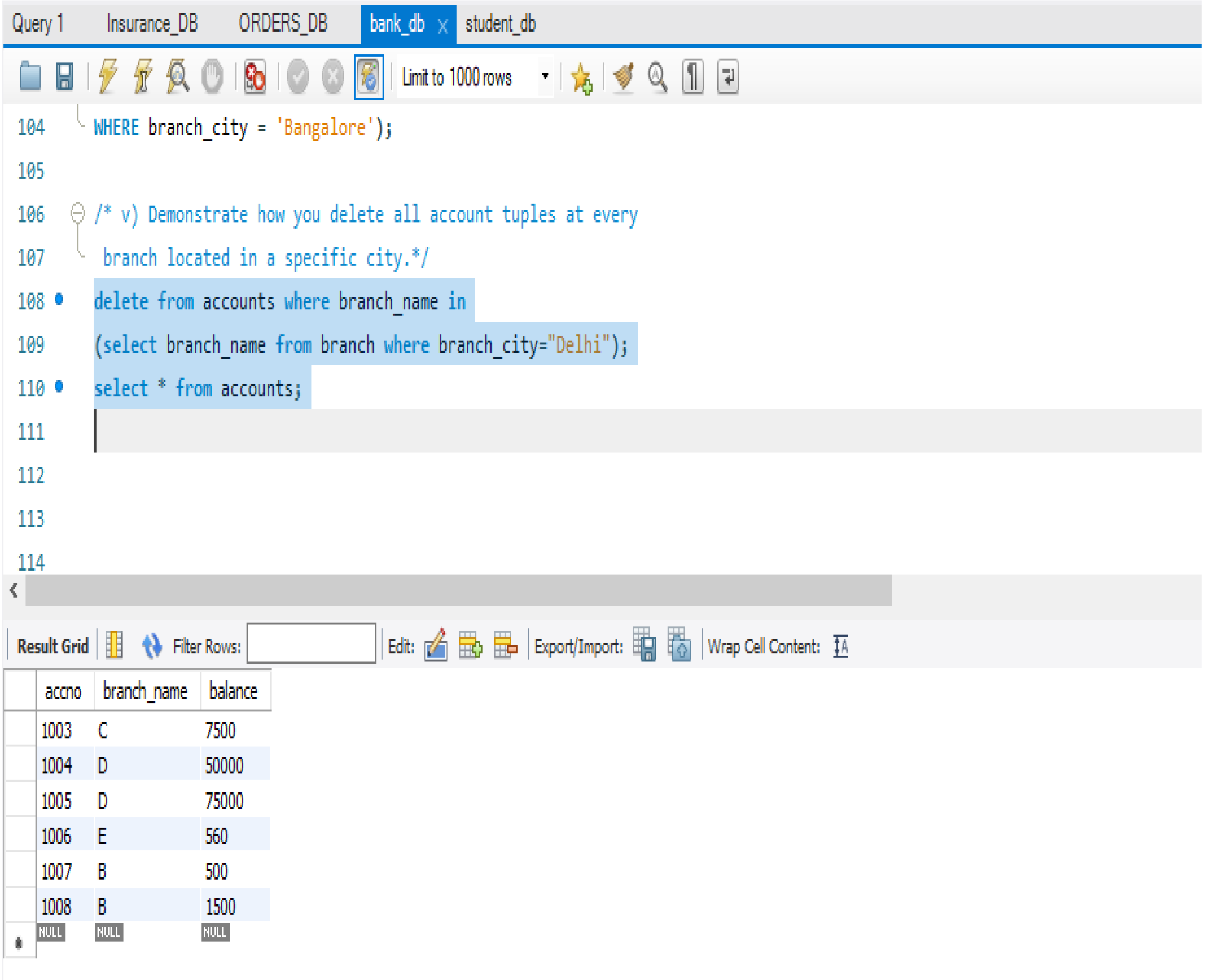
iv.Find all the customers who have an account at *all* the branches located in a specific city.



```
/*iv. Find all the customers who have an account at all the branches located in a specific cit
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');
```

customer_name
Ravi

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



```
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;
```

accno	branch_name	balance
1003	C	7500
1004	D	50000
1005	D	75000
1006	E	560
1007	B	500
1008	B	1500
NULL	NULL	NULL

*******************LAB-4 ENDS*********************

WEEK-5 DBMS LAB

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

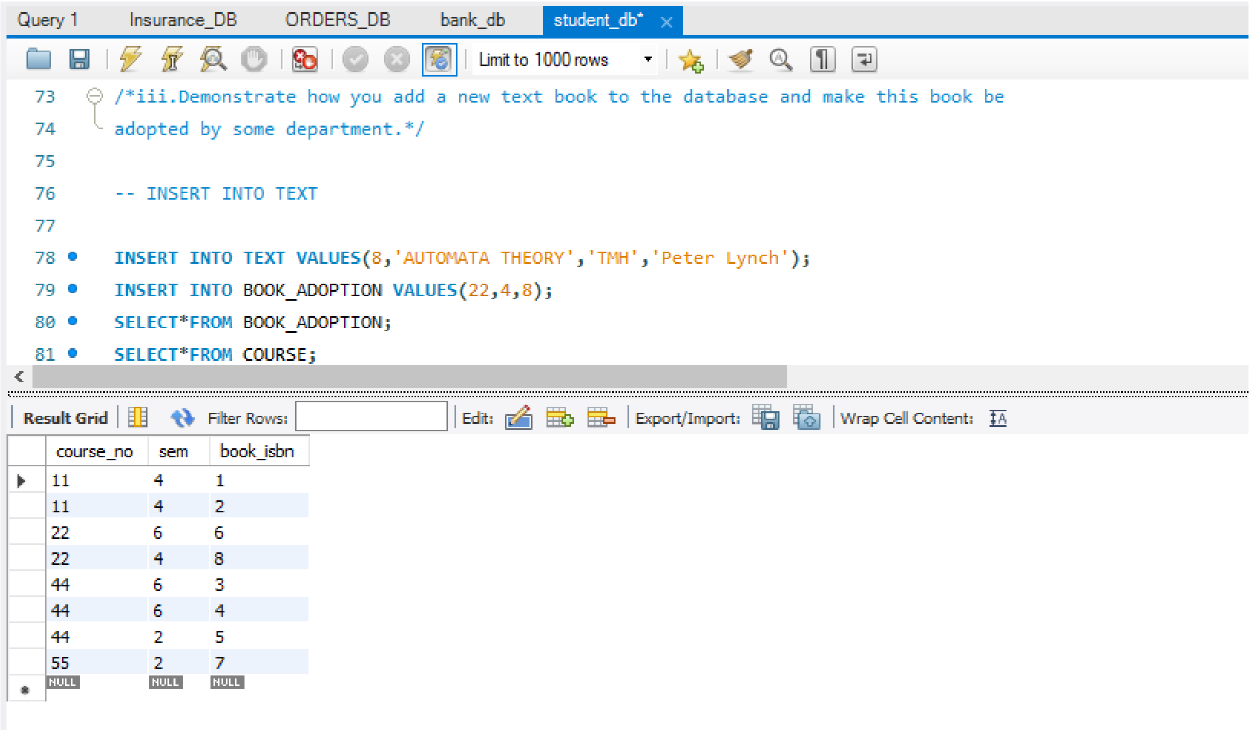
STUDENT (regno: String, name: String, major: String, bdate: date)
COURSE (course #: int, cname: String, dept: String)
ENROLL (regno: String, cname: String, sem: int, marks: int)
BOOK_ADOPTION (course #: int, sem: int, book-ISBN: int)
TEXT(book-ISBN:int, book-title: String, publisher:String, author:String)

- **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.**

OUTPUT:

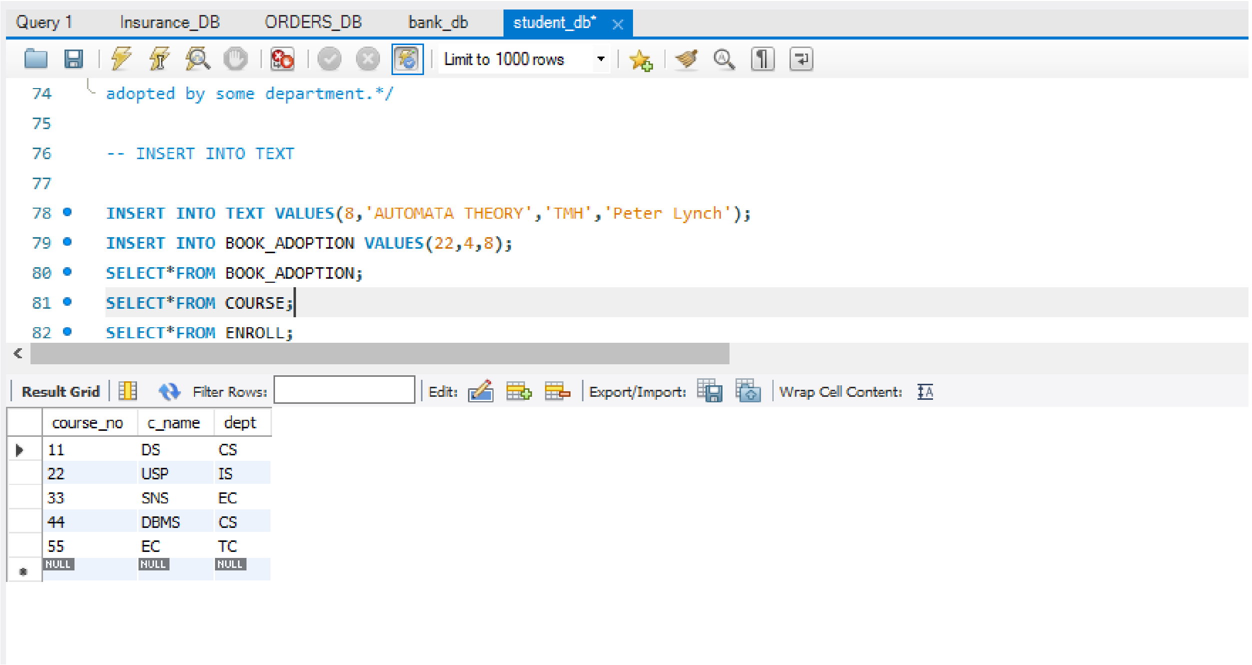
i. **Create the above tables by properly specifying the primary keys and the foreign keys.**

ii. **Enter at least five tuples for each relation.**



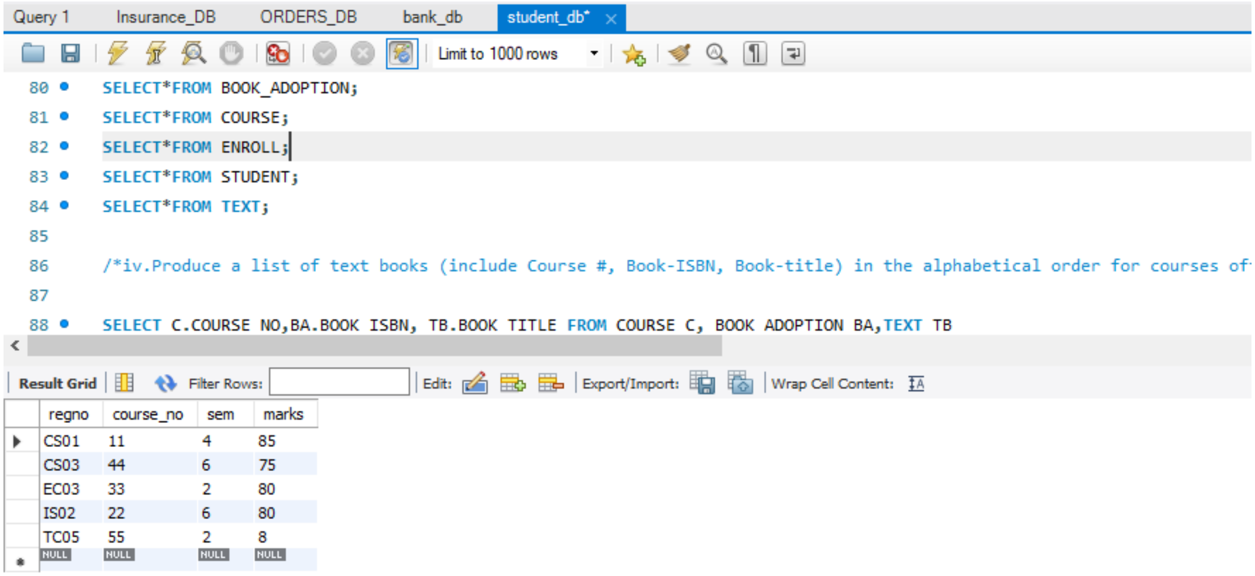
```
/*iii.Demonstrate how you add a new text book to the database and make this book be
adopted by some department.*/

-- INSERT INTO TEXT

INSERT INTO TEXT VALUES(8,'AUTOMATA THEORY','TMH','Peter Lynch');
INSERT INTO BOOK_ADOPTION VALUES(22,4,8);
SELECT*FROM BOOK_ADOPTION;
SELECT*FROM COURSE;
```

course_no	sem	book_isbn
11	4	1
11	4	2
22	6	6
22	4	8
44	6	3
44	6	4
44	2	5
55	2	7
NULL	NULL	NULL



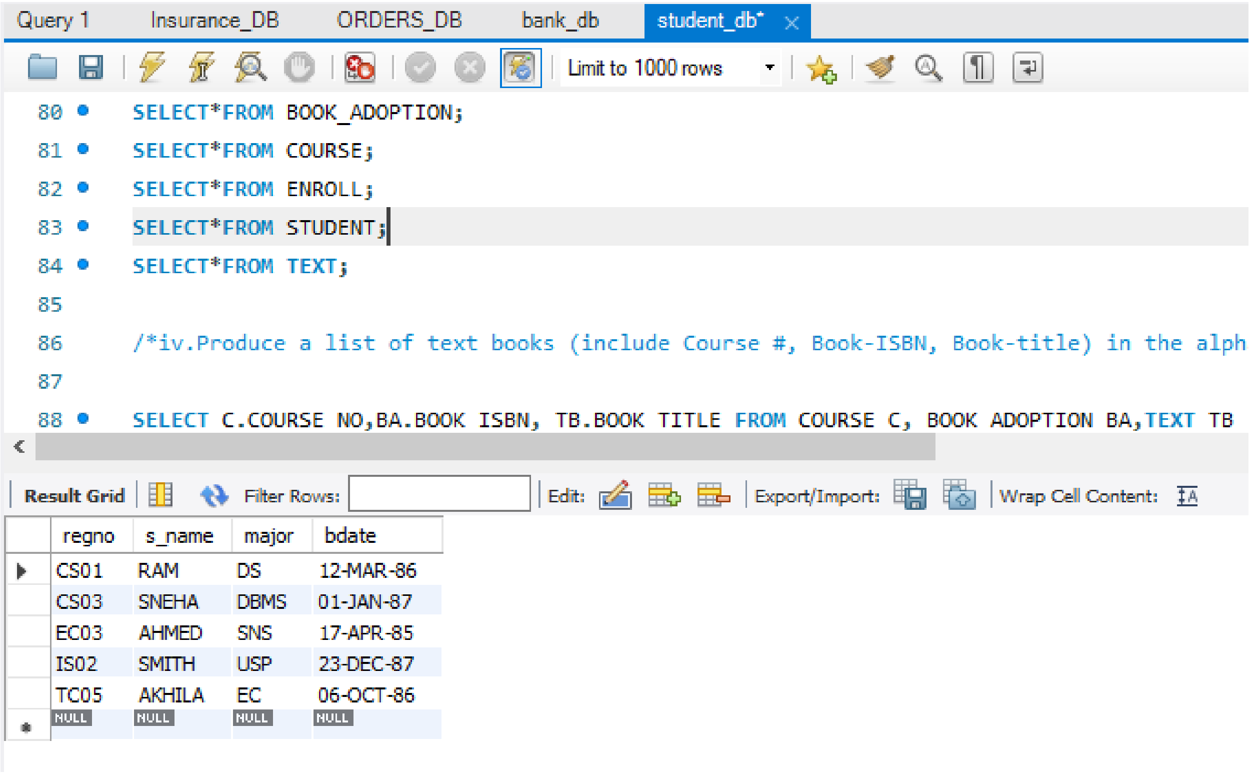
```
adopted by some department.*/

-- INSERT INTO TEXT

INSERT INTO TEXT VALUES(8,'AUTOMATA THEORY','TMH','Peter Lynch');
INSERT INTO BOOK_ADOPTION VALUES(22,4,8);
SELECT*FROM BOOK_ADOPTION;
SELECT*FROM COURSE;
SELECT*FROM ENROLL;
```

course_no	c_name	dept
11	DS	CS
22	USP	IS
33	SNS	EC
44	DBMS	CS
55	EC	TC
NULL	NULL	NULL



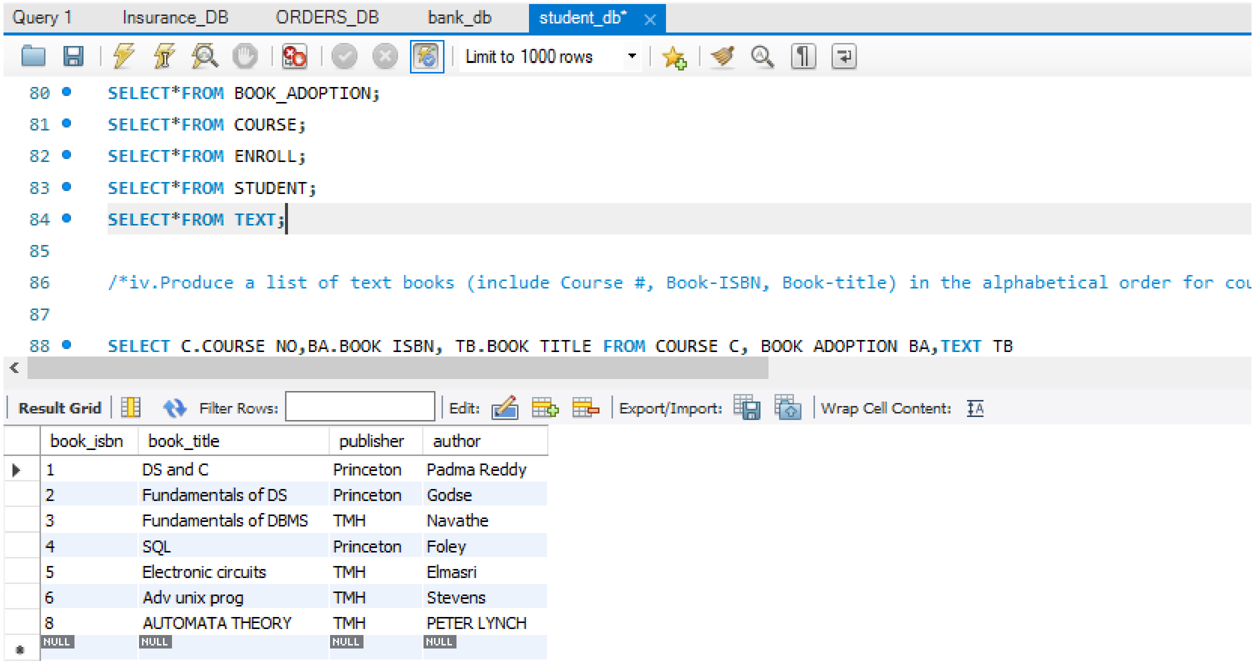
```
SELECT*FROM BOOK_ADOPTION;
SELECT*FROM COURSE;
SELECT*FROM ENROLL;
SELECT*FROM STUDENT;
SELECT*FROM TEXT;

/*iv.Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses of

SELECT C.COURSE NO,BA.BOOK ISBN, TB.BOOK TITLE FROM COURSE C, BOOK ADOPTION BA,TEXT TB
```

regno	course_no	sem	marks
CS01	11	4	85
CS03	44	6	75
EC03	33	2	80
IS02	22	6	80
TC05	55	2	8
NULL	NULL	NULL	NULL



```
SELECT*FROM BOOK_ADOPTION;
SELECT*FROM COURSE;
SELECT*FROM ENROLL;
SELECT*FROM STUDENT;
SELECT*FROM TEXT;

/*iv.Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alph

SELECT C.COURSE NO,BA.BOOK ISBN, TB.BOOK TITLE FROM COURSE C, BOOK ADOPTION BA,TEXT TB
```

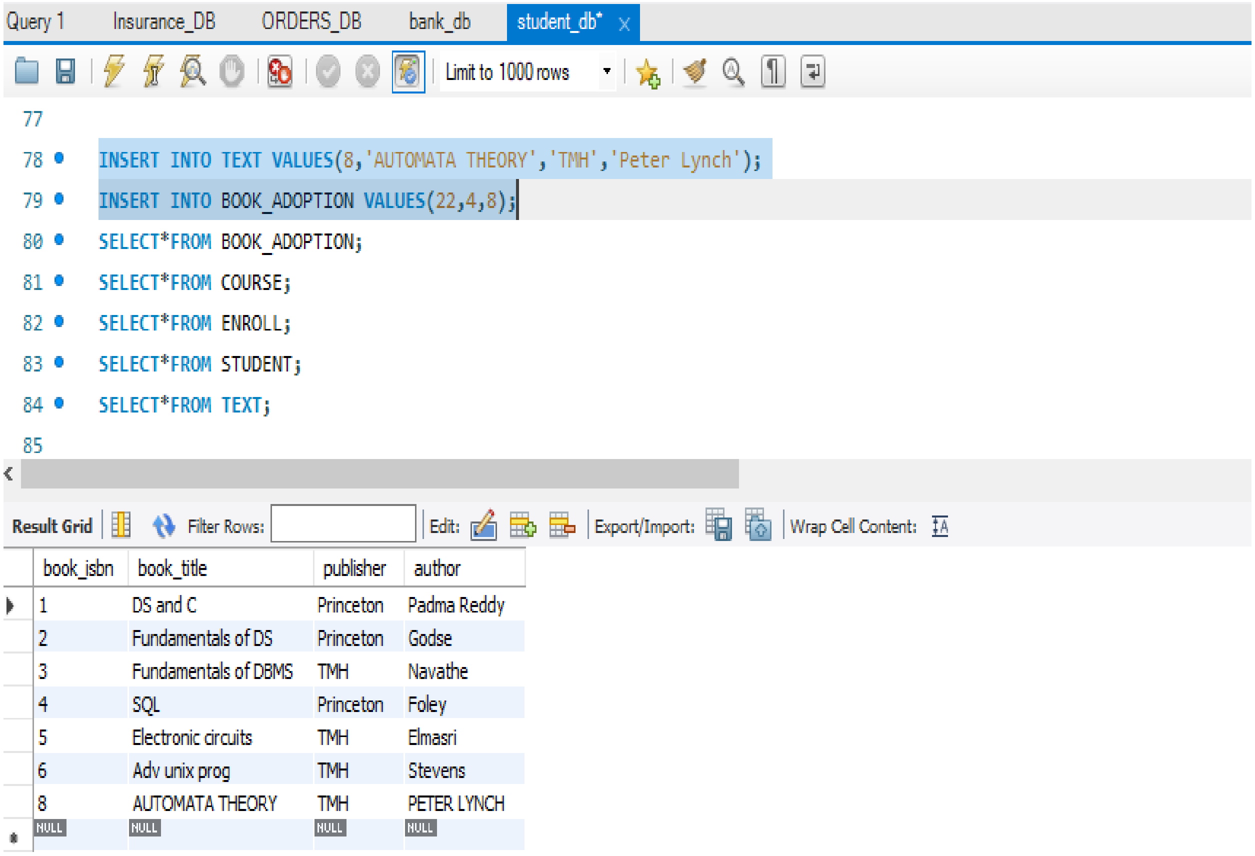
regno	s_name	major	bdate
CS01	RAM	DS	12-MAR-86
CS03	SNEHA	DBMS	01-JAN-87
EC03	AHMED	SNS	17-APR-85
IS02	SMITH	USP	23-DEC-87
TC05	AKHILA	EC	06-OCT-86
NULL	NULL	NULL	NULL



```
SELECT*FROM BOOK_ADOPTION;
SELECT*FROM COURSE;
SELECT*FROM ENROLL;
SELECT*FROM STUDENT;
SELECT*FROM TEXT;

/*iv.Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for cou

SELECT C.COURSE NO,BA.BOOK ISBN, TB.BOOK TITLE FROM COURSE C, BOOK ADOPTION BA,TEXT TB
```

book_isbn	book_title	publisher	author
1	DS and C	Princeton	Padma Reddy
2	Fundamentals of DS	Princeton	Godse
3	Fundamentals of DBMS	TMH	Navathe
4	SQL	Princeton	Foley
5	Electronic circuits	TMH	Elmasri
6	Adv unix prog	TMH	Stevens
8	AUTOMATA THEORY	TMH	PETER LYNCH
NULL	NULL	NULL	NULL

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

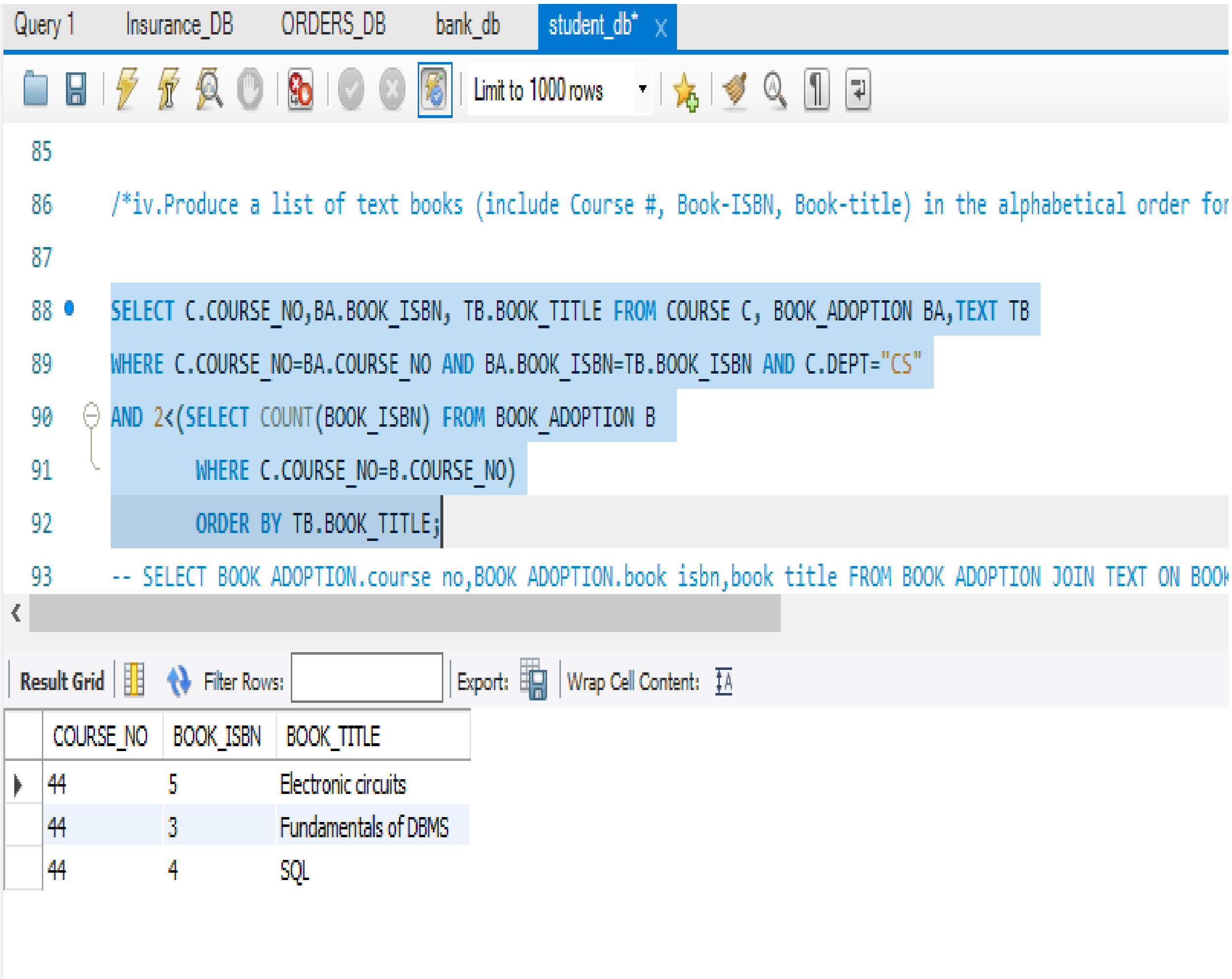


Limit to 1000 rows

```
INSERT INTO TEXT VALUES(8,'AUTOMATA THEORY','TMH','Peter Lynch');
INSERT INTO BOOK_ADOPTION VALUES(22,4,8);
SELECT*FROM BOOK_ADOPTION;
SELECT*FROM COURSE;
SELECT*FROM ENROLL;
SELECT*FROM STUDENT;
SELECT*FROM TEXT;
```

book_isbn	book_title	publisher	author
1	DS and C	Princeton	Padma Reddy
2	Fundamentals of DS	Princeton	Godse
3	Fundamentals of DBMS	TMH	Navathe
4	SQL	Princeton	Foley
5	Electronic circuits	TMH	Elmasri
6	Adv unix prog	TMH	Stevens
8	AUTOMATA THEORY	TMH	PETER LYNCH
NULL	NULL	NULL	NULL

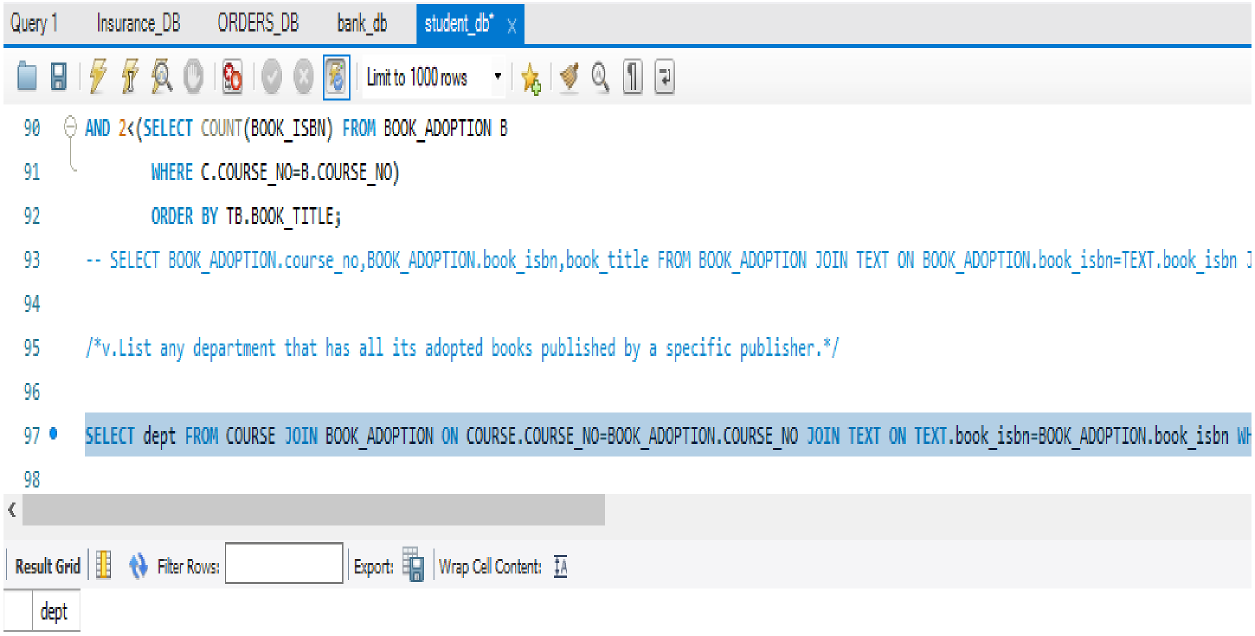
iv.Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the ‘CS’ department that use more than two books.



```
/*iv.Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for

SELECT C.COURSE_NO,BA.BOOK_ISBN, TB.BOOK_TITLE FROM COURSE C, BOOK_ADOPTION BA,TEXT TB
WHERE C.COURSE_NO=BA.COURSE_NO AND BA.BOOK_ISBN=TB.BOOK_ISBN AND C.DEPT="CS"
AND 2<(SELECT COUNT(BOOK_ISBN) FROM BOOK_ADOPTION B
        WHERE C.COURSE_NO=B.COURSE_NO)
        ORDER BY TB.BOOK_TITLE;
-- SELECT BOOK_ADOPTION.course_no,BOOK_ADOPTION.book_isbn,book_title FROM BOOK_ADOPTION JOIN TEXT ON BOOK
```

COURSE_NO	BOOK_ISBN	BOOK_TITLE
44	5	Electronic circuits
44	3	Fundamentals of DBMS
44	4	SQL

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



```
AND 2<(SELECT COUNT(BOOK_ISBN) FROM BOOK_ADOPTION B
        WHERE C.COURSE_NO=B.COURSE_NO)
        ORDER BY TB.BOOK_TITLE;
-- SELECT BOOK_ADOPTION.course_no,BOOK_ADOPTION.book_isbn,book_title FROM BOOK_ADOPTION JOIN TEXT ON BOOK_ADOPTION.book_isbn=TEXT.book_isbn J

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

SELECT dept FROM COURSE JOIN BOOK_ADOPTION ON COURSE.COURSE_NO=BOOK_ADOPTION.COURSE_NO JOIN TEXT ON TEXT.book_isbn=BOOK_ADOPTION.book_isbn WH
```

	dept

*************LAB 5 ENDS****************