**Lab Assignment 1**

**Aim:** Consider the insurance database given below. The primary keys are made bold  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 , accd\_date:date , location:string )  
OWNS( **driver\_id**:string , **regno**:string )  
PARTICIPATED( **driver\_id**:string , **regno**:string , **report\_number**:int , damage\_amount:int)

1)Create the above tables by properly specifying the primary keys and foreign keys.  
2)Enter at least five tuples for each relation.  
3)Demonstrate how you

a.Update the damage amount for the car with specific regno in the accident with report number 12 to 25000.  
b.Add a new accident to the database.

4)Find the total number of people who owned cars that were involved in accidents in the year 2008.  
5)Find the number of accidents in which cars belonging to a specific model were involved.

**Description:**

1.)

SQL> create table person(driver\_id varchar2(10) primary key,name char(15),address varchar2(25)); Table created.

SQL> create table car(regno varchar2(10) primary key,model varchar2(15),year number(4));

Table created.

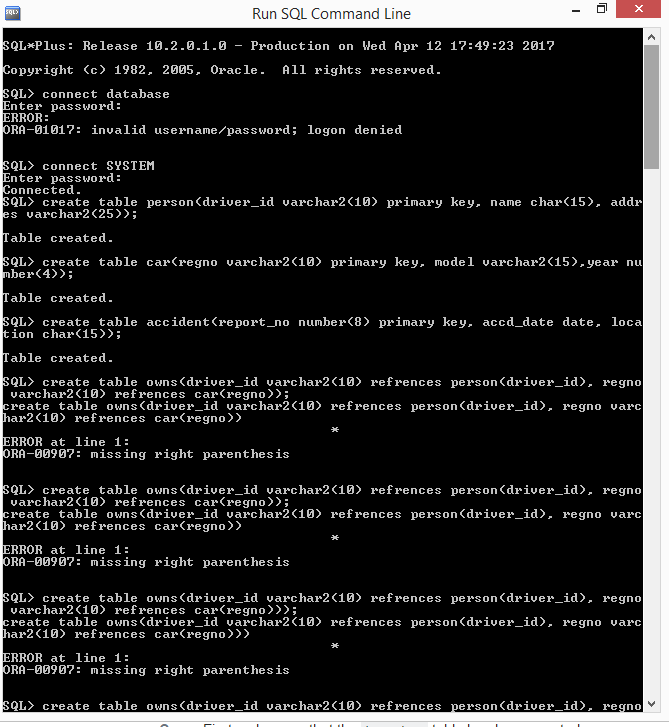
SQL> create table accident(report\_no number(8) primary key,accd\_date date,location char(15));

Table created.

SQL> create table owns(driver\_id varchar2(10) references person(driver\_id),regno varchar2(10) references car(regno));

Table created.

SQL>create table participated(driver\_id varchar2(10) references person(driver\_id),regno varchar2(10) references car(regno),report\_no number(8)references accident(report\_no),damage\_amt number(10));

Table created.**** 

2.)

insert into person values(‘**a111’ ,‘atul ‘,‘gwalior’**);

1 row created.

SQL> insert into person values(‘**a112 ‘,‘aman ‘,‘gwalior’**);

1 row created.

SQL> insert into person values(‘**a113 ‘,‘partha‘,‘delhi’**);

1 row created.

SQL> insert into person values(‘**a114 ‘,‘alom ‘,’tripura’**);

1 row created.

SQL> insert into person values(‘**a115 ‘,‘bharti ‘,‘gwalior**‘);

1 row created.

SQL> SELECT \* FROM PERSON;

**DRIVER\_ID NAME ADDRESS**

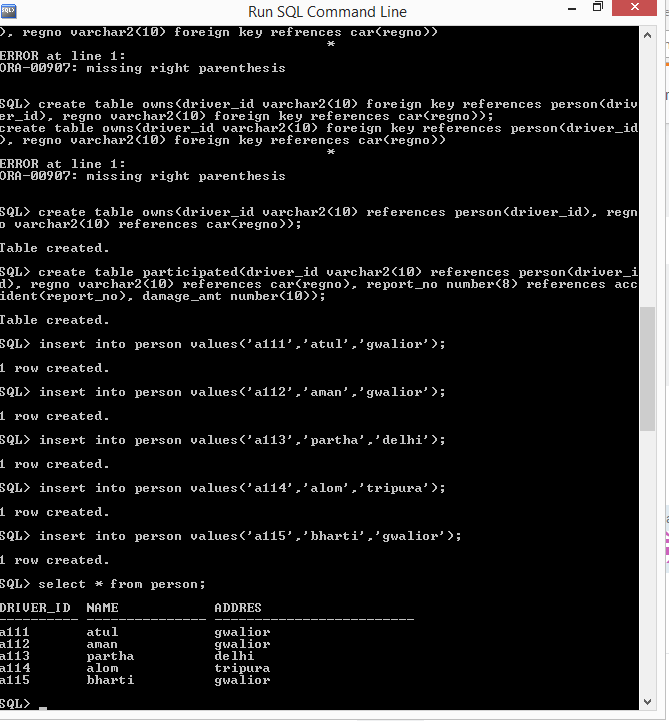
**---------- --------------- - ------------------------**

**a111 atul gwalior**

**a112 aman gwalior**

**a113 partha delhi**

**a114 alom tripura**

**a115 bharti gwalior**

SQL> insert into car values(‘**K234gdh67 ‘,‘swift’, 2016**);

1 row created.

SQL> insert into car values(‘**K23gd4567’,‘fiat’,2017**);

1 row created.

SQL> insert into car values(‘**K2334rfgt’,‘bmw’,2009**);

1 row created.

SQL> insert into car values(‘**K345fd45t ‘,‘lamborghini’,2008**);

1 row created.

SQL> insert into car values(‘**K343er45t ‘,‘jaguar’,2008**);

1 row created.

SQL> SELECT \* FROM CAR;

**REGNO MODEL YEAR**

**---------- -------------- ---------**

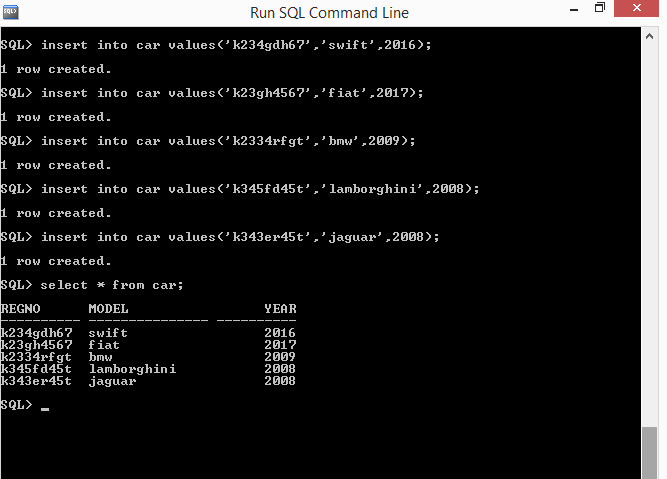
**K234gdh67 swift 2016**

**K23gd4567 fiat 2017**

**K2334rfgt bmw 2009**

**K345fd45t lamborghini 2008**

**K343er45t jaguar 2008**



SQL> insert into accident values(**101,‘01-FEB-10 ‘,‘thatipur ‘**);

1 row created.

SQL> insert into accident values(**102 ,‘09-FEB-15 ‘,‘turari’**);

1 row created.

SQL> insert into accident values(**103,‘25-APR-17‘,‘sithaouli’**);

1 row created.

SQL> insert into accident values(**104 ,‘27-SEP-17 ‘,‘sithaouli’**);

1 row created.

SQL> insert into accident values(**105,’21-DEC-12‘,‘ govindpuri’**);

1 row created.

SQL> select \* from accident;

**REPORT\_NO ACCD\_DATE LOCATION**

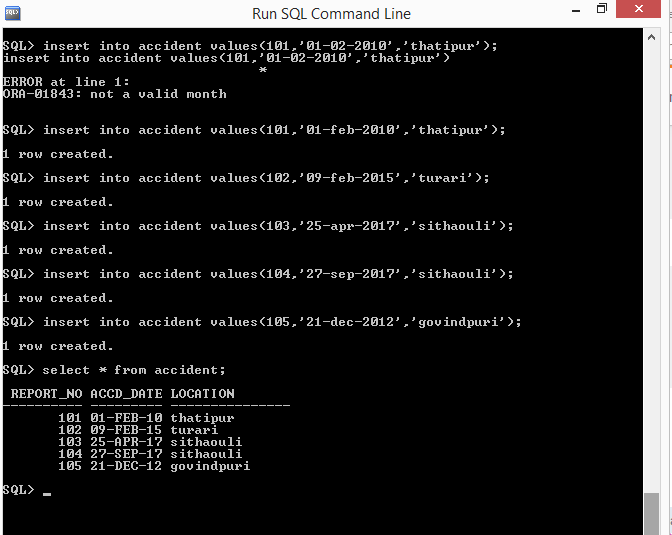
**--------- --------- ---------------**

**101 01-FEB-10 thatipur**

**102 09-FEB-15 turari**

**103 25-APR-17 sithaouli**

**104 27-SEP-17 sithaouli**

**105 21-DEC-12 govindpuri**

SQL> insert into owns values(‘**a111’,’K234gdh67’**);

1 row created.

SQL> insert into owns values(‘**a112’, ‘K23gd4567‘**);

1 row created.

SQL> insert into owns values(‘**a113 ‘,‘K343er45t’**);

1 row created.

SQL> insert into owns values(‘**a114 ‘,‘K343er45t’**);

1 row created.

SQL> insert into owns values(‘**a115 ‘,‘K2334rfgt’**);

1 row created.

SQL> select \* from owns;

**DRIVER\_ID REGNO**

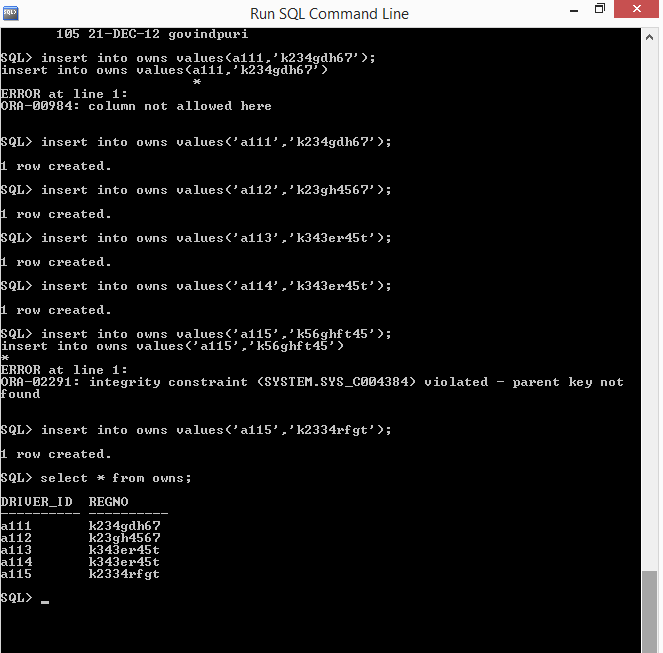
**---------- ----------**

**a111 K234gdh67**

**a112 K23gd4567**

**a113 K343er45t**

**a114 K343er45t**

**a115 K2334rfgt **

SQL> insert into participated values(‘a111’,’K234gdh6**7**’,101,700);

1 row created.

SQL> insert into participated values(‘a111 ‘,‘**K2334rfgt**‘,102,25000);

1 row created.

SQL> insert into participated values(‘a111 ‘,‘**K2334rfgt**’,103,7000);

1 row created.

SQL> insert into participated values(‘a115 ‘,‘**K2334rfgt**’,10, 12000);

1 row created.

SQL> insert into participated value(‘a113‘, ‘**K2334rfgt**’,105,2000);

1 row created.

SQL> select \* from participated;

**DRIVER\_ID REGNO REPORT\_NO DAMAGE\_AMT**

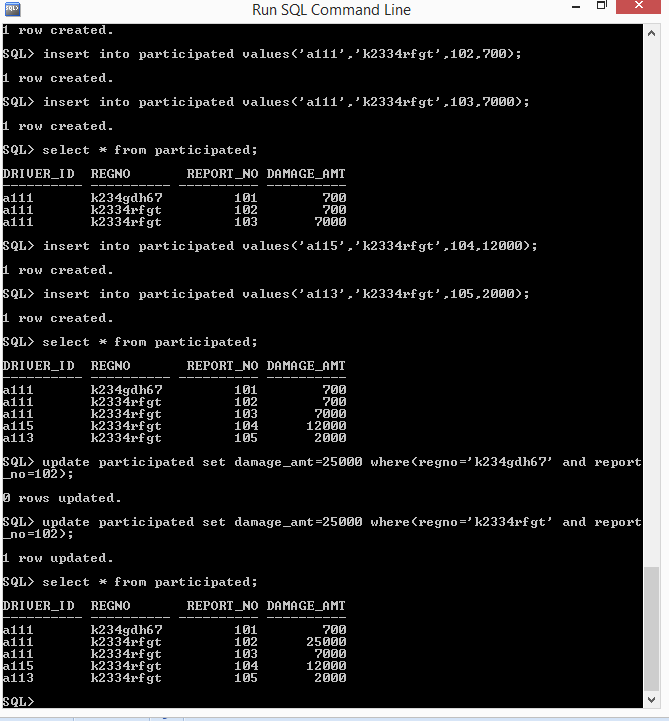
**---------- ---------- --------- ----------**

a111 K234gdh6**7** 101 700

a111 **K2334rfgt** 102 25000

a111 **K2334rfgt** 103 7000

a115 **K2334rfgt** 104 12000

a113 **K2334rfgt** 105 2000****

3.-a.)

SQL> update participated set damage\_amt=25000 where (regno=’K234gdh6**7**’' and report\_no=102);

0 rows updated.

SQL> update participated set damage\_amt=25000 where (regno=’ **K2334rfgt**’' and report\_no=102);

1 row updated.

SQL> select \* from participated;

**DRIVER\_ID REGNO REPORT\_NO DAMAGE\_AMT**

**---------- - --------- --------- ----------**

a111 K234gdh6**7** 101 700

a111 **K2334rfgt** 102 25000

a111 **K2334rfgt** 103 7000

a115 **K2334rfgt** 104 12000

a113 **K2334rfgt** 105 2000

3-b.)

SQL> insert into accident values(504,'18-oct-2008','krpuram');

1 row created.

SQL> select \* from accident;

**REPORT\_NO ACCD\_DATE LOCATION**

**--------- --------- ---------------**

**101 01-FEB-10 thatipur**

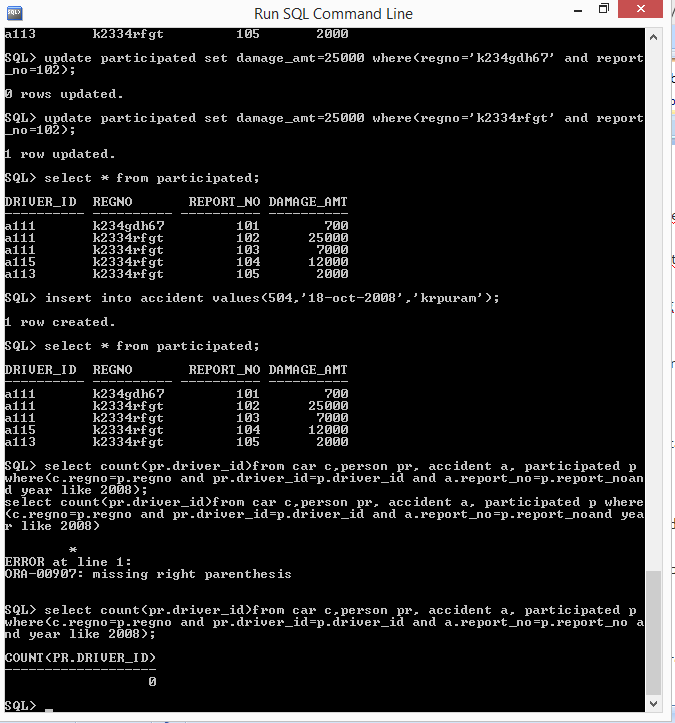
**102 09-FEB-15 turari**

**103 25-APR-17 sithaouli**

**104 27-SEP-17 sithaouli**

**105 21-DEC-12 govindpuri**

**504 18-oct-2008 krpuram**



4.)

SQL> select count(pr.driver\_id) from car c,person pr,accident a,participated p where(c.regno=p.regno and pr.driver\_id=p.driver\_id and a.report\_no=p.report\_no and year like 2008);

**COUNT(PR.DRIVER\_ID)**

**-------------------**

**0**

**5.)**

SQL> select count(a.report\_no) from car c,accident a,participated p where(c.regno=p.regno and a.report\_no=p.report\_no and model like 'gypsy');

**COUNT(A.REPORT\_NO)**

**------------------**

**1**



**Lab Assignment 2**

**Aim:** Consider the following relations for a order processing database application in a company.

CUSTOMER( **custno**:int , cname:string , city:string )  
ORDER( **orderno**:int , odate:date , custno:int , ord\_amt:int )  
ORDER\_ITEM( **orderno**:int , **itemno**:int , quantity:int )  
ITEM( **itemno**:int , unitprice:int )  
SHIPMENT( **orderno**:int , **warehouseno**:int , ship\_date:date )  
WAREHOUSE( **warehouseno**:int , city:string )

1)Create the above tables by properly specifying the primary keys and foreign keys.  
2)Enter at least five tuples for each relation.  
3)Produce a listing: custname , No\_of\_orders , Avg\_order\_amount ,  where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer.  
4)List the orderno for orders that were shipped from ***all*** the warehouses that the company has in a specific city.  
5)Demonstrate the deletion of an item from the ITEM table and demonstrate a method of handling the rows in the ORDER\_ITEM table that contains this particular item.

**Description:**

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

SQL> create table customer(cust\_no number(5) primary key,cname varchar2(10),city varchar2(10)); Table created.

SQL> create table ordr(order\_no number(5) primary key,odatedate,cust\_no number(5) references customer(cust\_no),ord\_amt number(6));

Table created.

SQL> create table item(item\_no number(4) primary key,unit\_pr number(5));

Table created.

SQL> create table order\_item(order\_no number(5) references ordr(order\_no),item\_no number(3) references item(item\_no),qty number(4));

Table created.

SQL> create table warehouse(warehouse\_no number(5) primary key,city varchar2(10));

Table created.

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

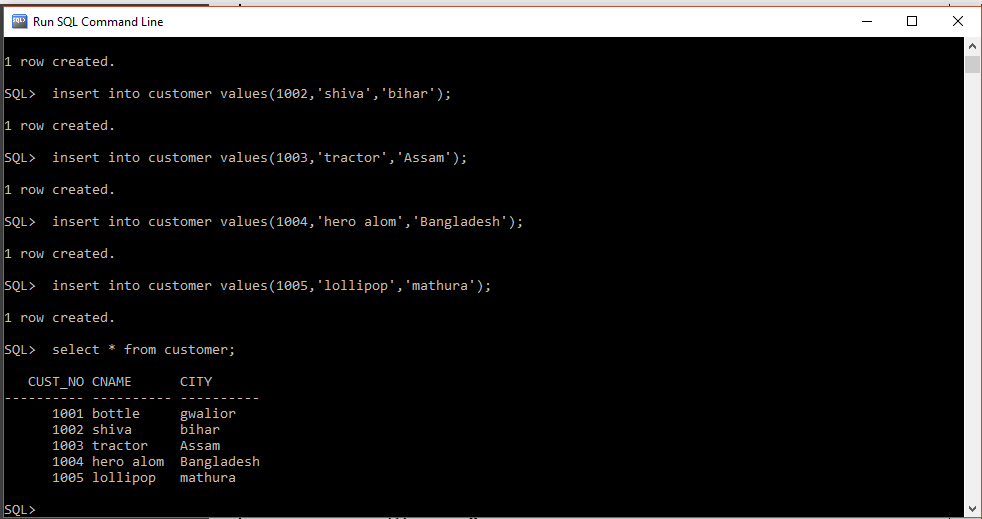
SQL> insert into customer values(1001,'bottle','gwalior'); 1 row created.

SQL> insert into customer values(1002,'shiva','bihar'); 1 row created.

SQL> insert into customer values(1003,'tractor','assam'); 1 row created.

SQL> insert into customer values(1004,'hero alom','bangladesh'); 1 row created.

SQL> insert into customer values(1005,'lolipop','mathura'); 1 row created.



SQL> insert into ordr values(501,'11-feb-2010',1002,5500);

1 row created.

SQL> insert into ordr values(502,'20-oct-2009',1001,9700);

1 row created.

SQL> insert into ordr values(505,'13-aug-2008',1005,5600);

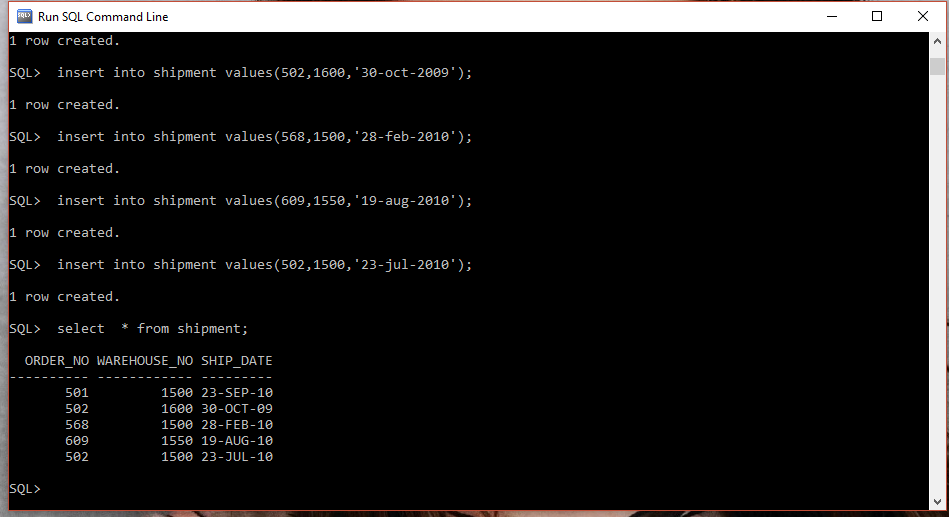
1 row created.

SQL> insert into ordr values(568,'29-jun-2010',1004,4766);

1 row created.

SQL> insert into ordr values(609,'31-aug-2010',1003,9831);

1 row created.



SQL> insert into item values(111,255); 1 row created.

SQL> insert into item values(121,199);

1 row created.

SQL> insert into item values(131,177);

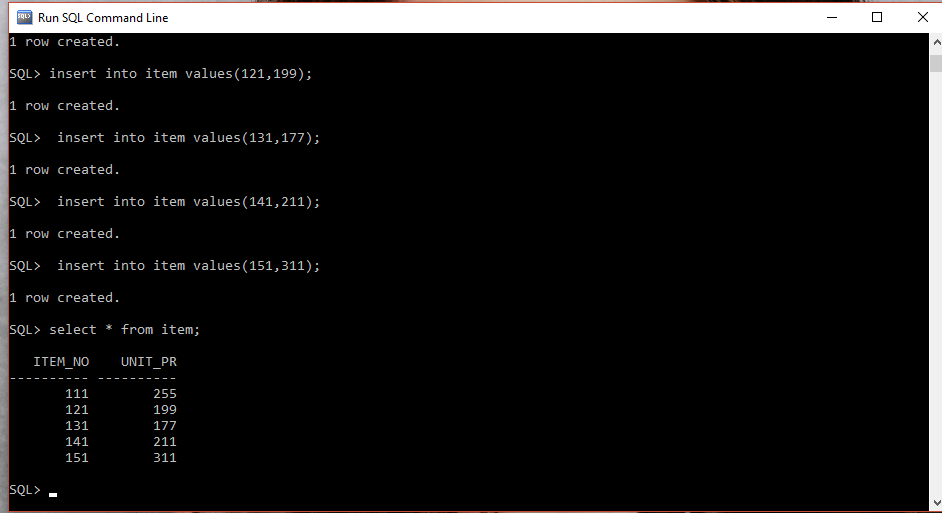
1 row created.

SQL> insert into item values(141,211);

1 row created.

SQL> insert into item values(151,311);

1 row created.



SQL> insert into order\_item values(501,111,200);

\1 row created.

SQL> insert into order\_item values(502,121,100);

1 row created.

SQL> insert into order\_item values(505,131,150);

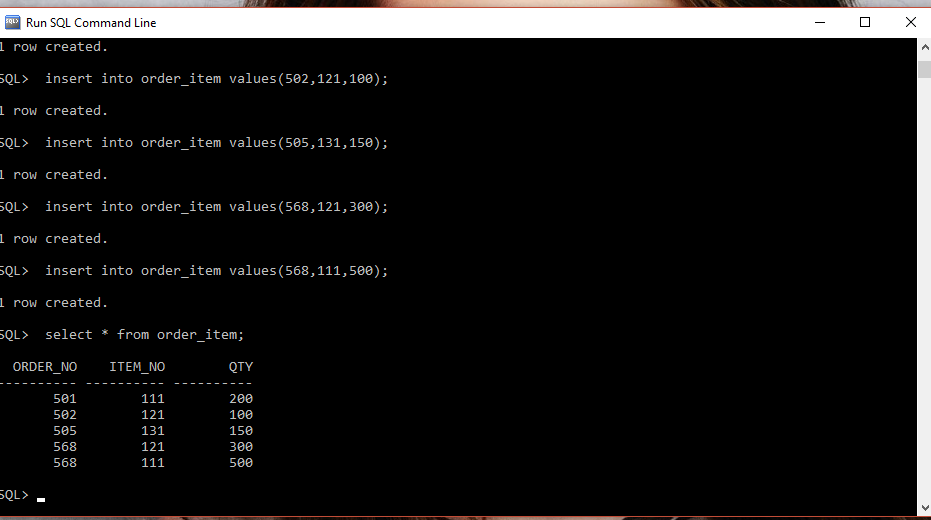
1 row created.

SQL> insert into order\_item values(568,121,300);

1 row created.

SQL> insert into order\_item values(568,111,500);

1 row created.



SQL> insert into warehouse values(1500,'bihar');

1 row created.

SQL> insert into warehouse values(1550,'gwalior');

1 row created.

SQL> insert into warehouse values(1600,'paradeep');

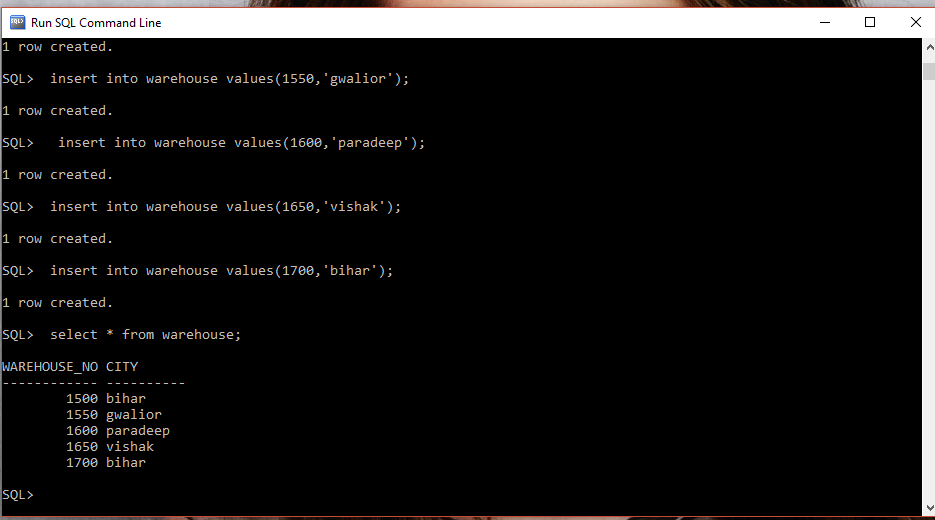
1 row created.

SQL> insert into warehouse values(1650,'vishak');

1 row created.

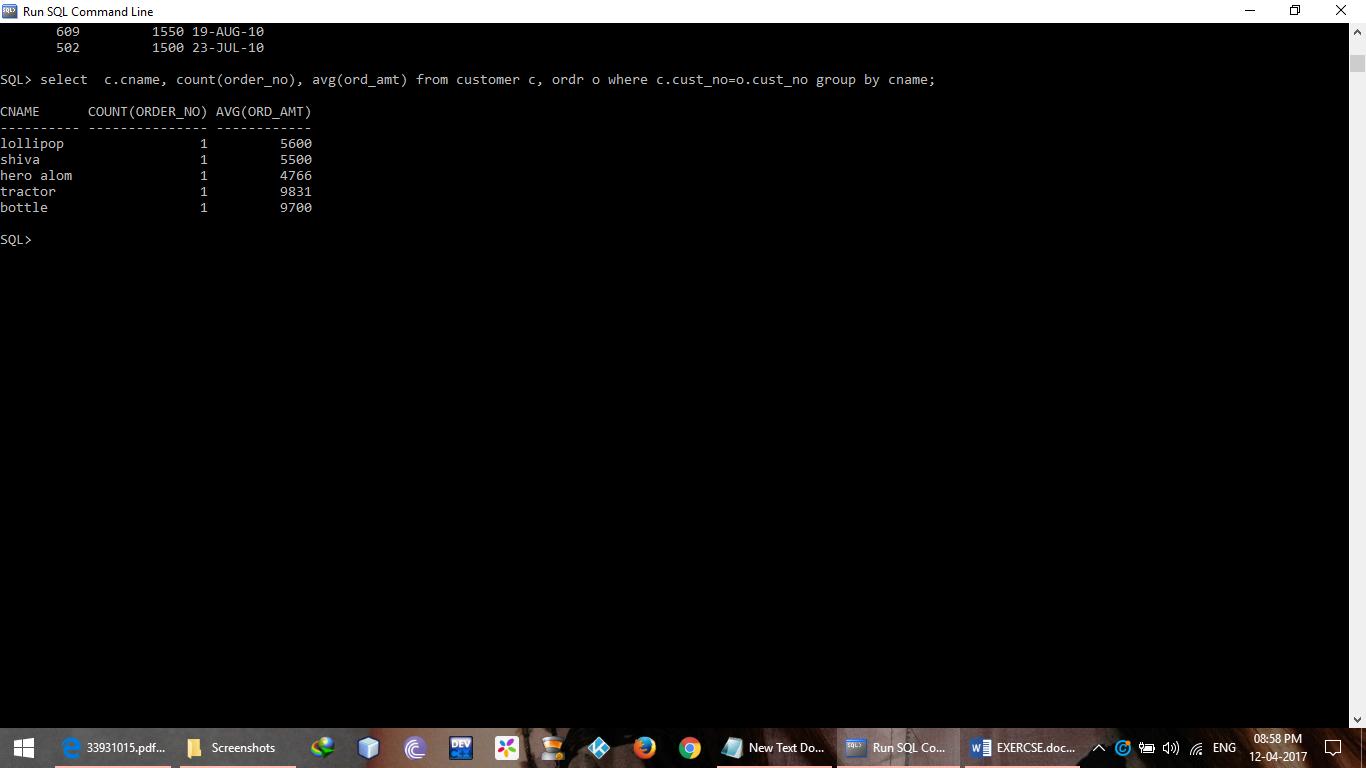
SQL> insert into warehouse values(1700,'bihar');

1 row created



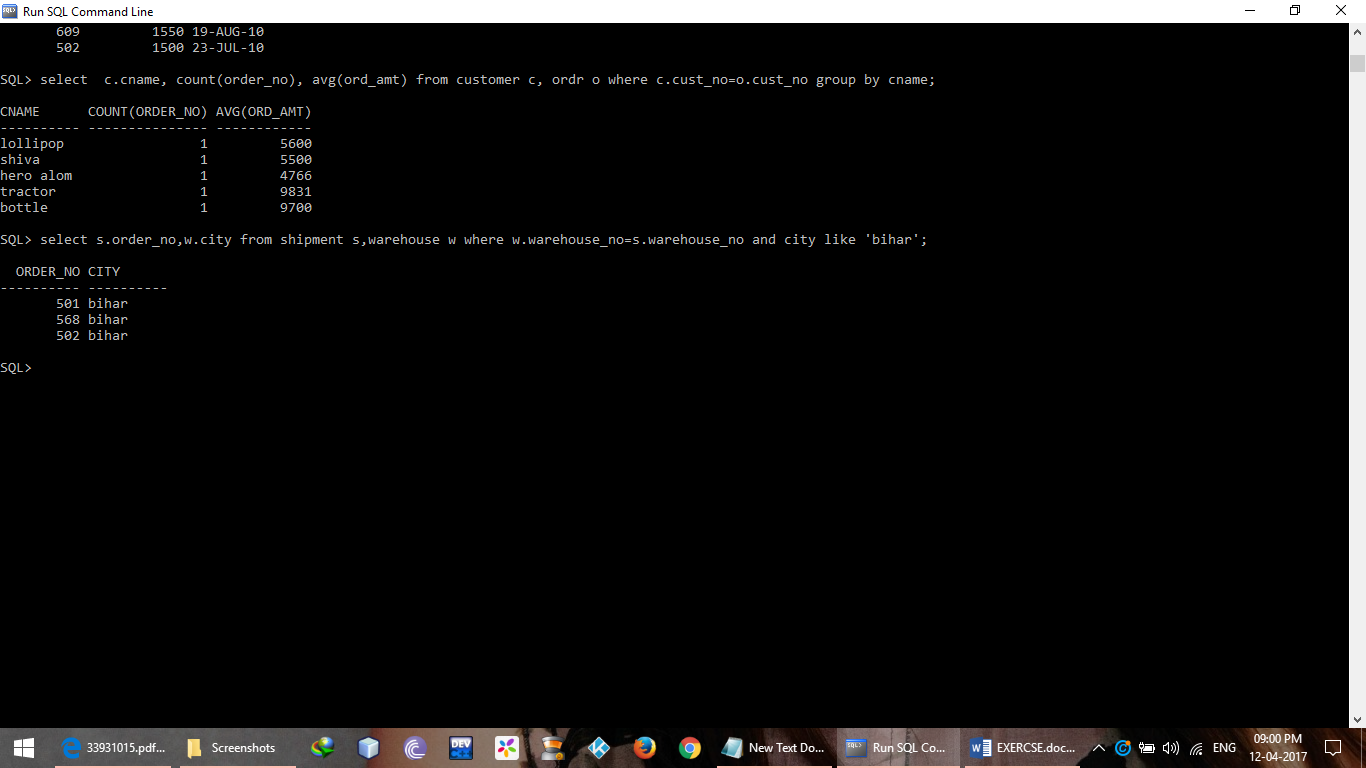
**(iii) Produce a listing: CUSTNAME, #oforders, AVG\_ORDER\_AMT, where the middle column is the total numbers of orders by the customer and the last column is the average order amount for that customer.**

SQL>select c.cname, count(order\_no), avg(ord\_amt) from customer c, ordr o where c.cust\_no=o.cust\_no group by cname;



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

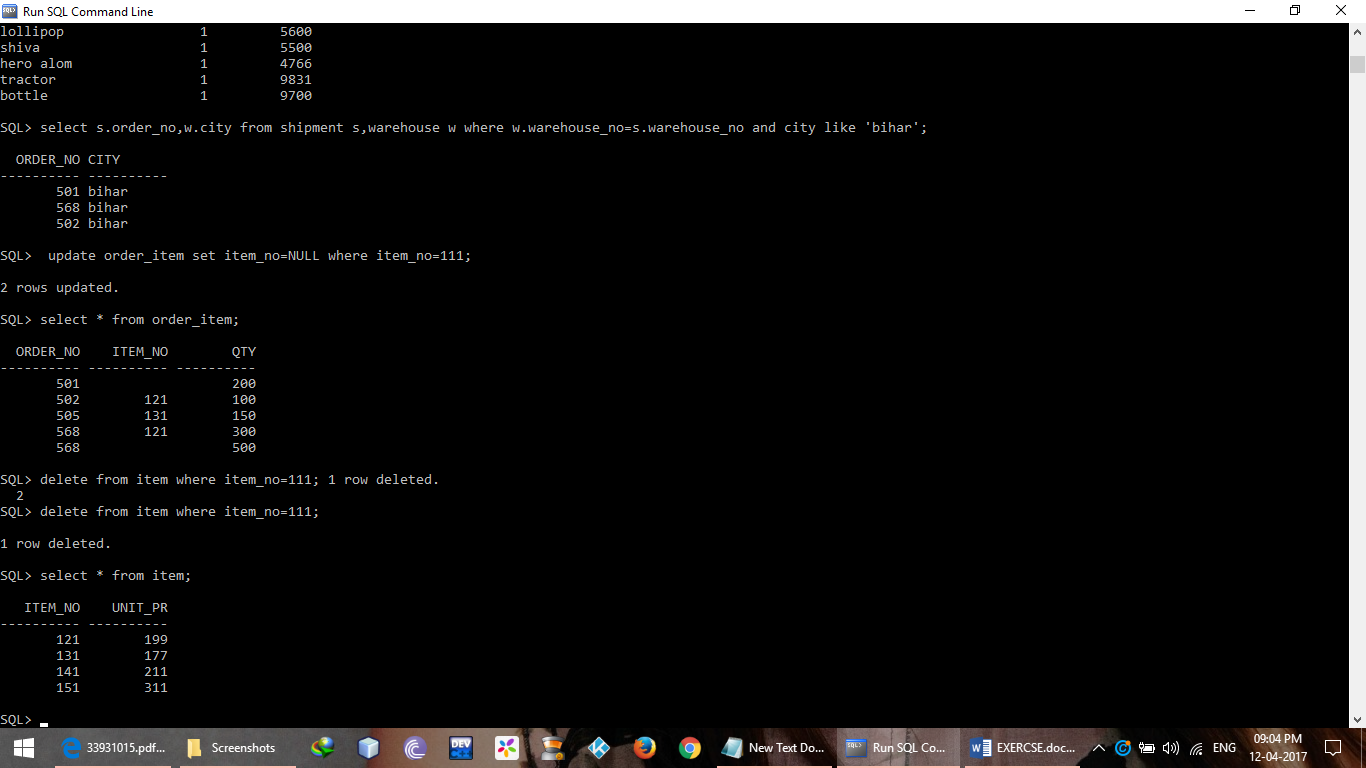
SQL> select s.order\_no,w.city from shipment s,warehouse w where w.warehouse\_no=s.warehouse\_no and city like 'bihar';



**(v) Demonstrate the deletion of an item from the ITEM table and demonstrate a method of handling the rows in the ORDER\_ITEM table that contain this particular item.**

SQL> update order\_item set item\_no=NULL where item\_no=111; 2 rows updated.

SQL> delete from item where item\_no=111; 1 row deleted.

****

**Lab Assignment 3**

**Aim:** Consider the following database of student enrollment in courses and books adopted for that course.

STUDENT( **regno**:string , name:string , major:string , bdate:date )  
COURSE( **courseno**:int , cname:string , dept:string )  
ENROLL( **regno**:string , **courseno**:int , **sem**:int , marks:int )  
BOOK\_ADOPTION( **courseno**:int , **sem**:int , book\_isbn:int )  
TEXT( **book\_isbn**:int , book\_title:string , publisher:string , author:string )

1)Create the above tables by properly specifying the primary keys and foreign keys.  
2)Enteratleast five tuples for each relation.  
3)Demonstrate how you add a new text book to the database and make this book to be adopted by some department.  
4)Produce a list of text books ( includes courseno , book\_isbn , book\_title ) in the alphabetical order for courses offered by the 'CS' department that use more than two books.  
5)List any department that has ***all*** its books published by a specific publisher.

**Description:**

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

SQL>create table student(regno varchar2(5) primary key,name varchar2(10),major varchar2(5),bdate date);

Table created.

SQL> create table course(course\_no number(5) primary key,cname varchar2(10),dept varchar2(10));

Table created.

SQL> create table enroll(regno varchar2(5) references student(regno),course\_no number(5) references course(course\_no),sem number(3),marks number(4));

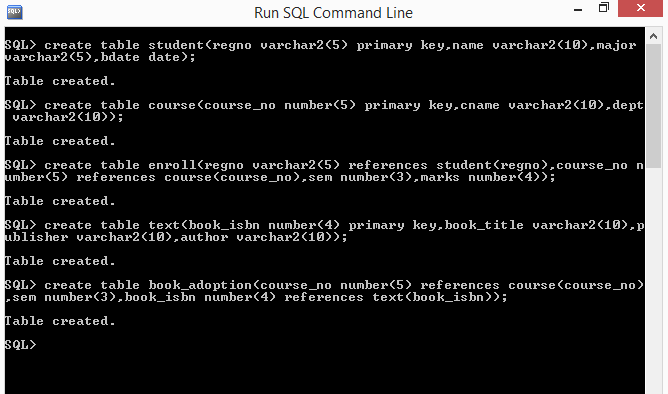
Table created.

SQL> create table text(book\_isbn number(4) primary key,book\_title varchar2(10),publisher varchar2(10),author varchar2(10));

Table created.

SQL> create table book\_adoption(course\_no number(5) references course(course\_no),sem number(3),book\_isbn number(4) references text(book\_isbn));

Table created.



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

SQL> insert into student values('15','aman','btech','17-aug-1998');

1 row created.

SQL> insert into student values('16','atul','btech','17-aug-1996');

1 row created.

SQL> insert into student values('20','jateen','mba','10-may-1996');

1 row created.

SQL> insert into student values('22','priya','bca','10-may-1997');

1 row created.

SQL> insert into student values('11','shivani','btech','10-may-1998');

1 row created.

SQL> select \* from student;

**REGNO NAME MAJOR BDATE**

**----- ---------- ----- ---------**

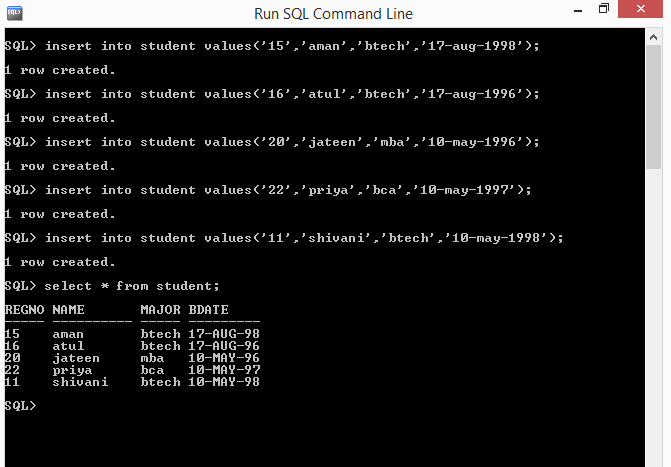
**15 amanbtech 17-AUG-98**

**16 atulbtech 17-AUG-96**

**20 jateenmba 10-MAY-96**

**22 priyabca 10-MAY-97**

**11 shivanibtech 10-MAY-98**



SQL> insert into course values(1001,'mca','dca');

1 row created.

SQL> insert into course values(1002,'bsc','cs');

1 row created.

SQL> insert into course values(1003,'bbm','dom');

1 row created.

SQL>> insert into course values(1004,'msc','cs');

SP2-0734: unknown command beginning "> insert i..." - rest of line ignored.

SQL> insert into course values(1004,'msc','cs');

1 row created.

SQL> insert into course values(1005,'bca','dca');

1 row created.

SQL> select \* from course;

**COURSE\_NO CNAME DEPT**

**---------- ---------- ----------**

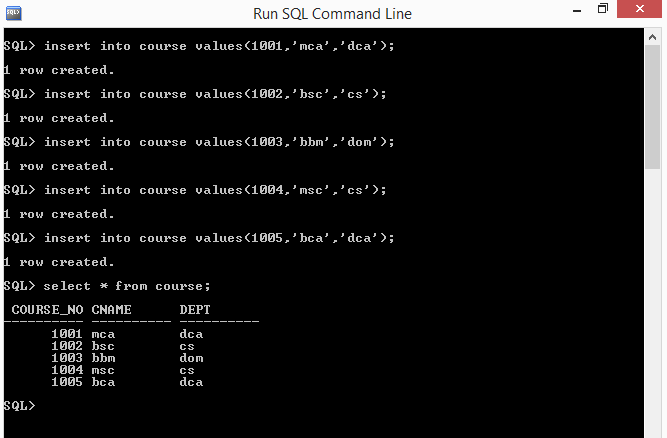
**1001 mca dca**

**1002 bsc cs**

**1003 bbm dom**

**1004 msc cs**

**1005 bca dca**

****

SQL> insert into enroll values('15',1001,2,678);

1 row created.

SQL> insert into enroll values('16',1002,2,678);

1 row created.

SQL> insert into enroll values('20',1003,2378);

insert into enroll values('20',1003,2378)

\*

ERROR at line 1:

ORA-00947: not enough values

SQL> insert into enroll values('20',1003,2,3378);

1 row created.

SQL> insert into enroll values('22',1004,4,3238);

1 row created.

SQL> select \* from enroll;

**REGNO COURSE\_NO SEM MARKS**

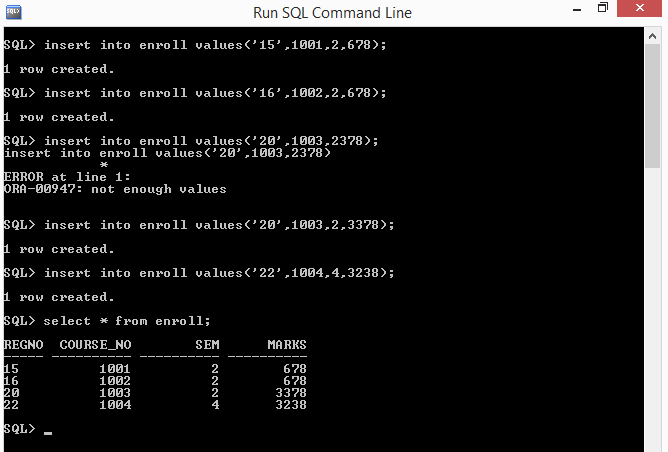
**----- ---------- ---------- ----------**

**15 1001 2 678**

**16 1002 2 678**

**20 1003 2 3378**

**22 1004 4 3238**



SQL> insert into text values(2100,'networks','pearson','stallings'); 1 row created.

SQL> insert into text values(3123,'dbms','mcgraw','navathe'); 1 row created.

SQL> insert into text values(4567,'oops','tata','venugopal'); 1 row created.

SQL> insert into text values(6412,'sys soft','reilly','dhamdere'); 1 row created.

SQL> insert into text values(8455,'tech com','peterson','mathew'); 1 row created.

SQL> select \* from text;

**BOOK\_ISBN BOOK\_TITLE PUBLISHER AUTHOR**

**---------- ---------- ---------- ----------**

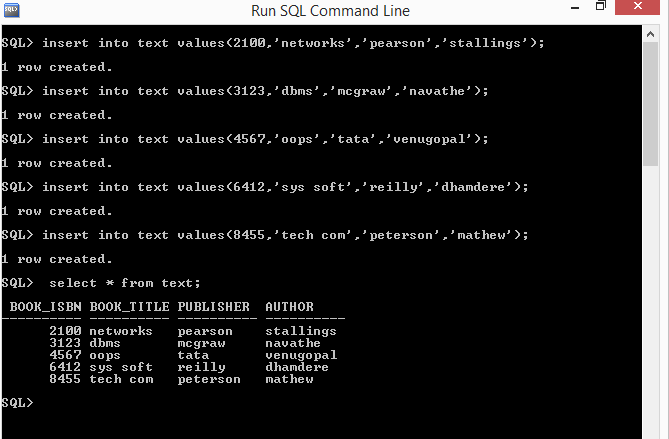
**2100 networks pearson stallings**

**3123 dbms mcgrawn avathe**

**4567 oops tata venugopal**

**6412 sys soft reilly dhamdere**

**8455 tech com Peterson Mathew**

****

SQL> insert into book\_adoption values(1002,2,2100);

1 row created.

SQL> insert into book\_adoption values(1002,2,2100);

1 row created.

SQL> insert into book\_adoption values(1004,2,2100);

1 row created.

SQL> insert into book\_adoption values(1005,4,8455);

1 row created.

SQL> insert into book\_adoption values(1003,3,4567);

1 row created.

SQL> select \* from book\_adoption;

**COURSE\_NO SEM BOOK\_ISBN**

**---------- ---------- ----------**

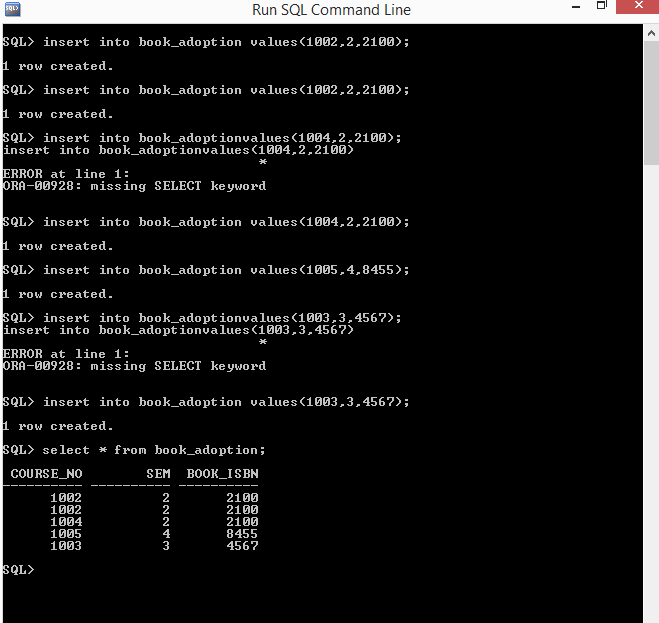
**1002 2 2100**

**1002 2 2100**

**1004 2 2100**

**1005 4 8455**

**1003 3 4567**

****

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

SQL> insert into text values(5398,'java','sonic','ranchor');

1 row created.

SQL> insert into book\_adoption values(1004,4,5398);

1 row created.

**(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.**

SQL> select c.course\_no,cname,t.book\_isbn,book\_title from course c,textt,book\_adoption b where c.course\_no=b.course\_no and b.book\_isbn=t.book\_isbn and dept='cs' and c.course\_no in(select course\_no from book\_adoption group by course\_no having count(\*) >= 1) order by c.cname;

**COURSE\_NO CNAME BOOK\_ISBN BOOK\_TITLE**

**---------- ---------- ---------- ----------**

**1002 bsc 2100 networks**

**1002 bsc 2100 networks**

**1004 msc 2100 networks**

**1004 msc 5398 java**

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

SQL> select c.dept from course c where course\_noin(select course\_no from book\_adoption where book\_isbn=(select book\_isbn from text where publisher='mcgraw'));

**DEPT**

**----------**

**Dca**

****

**Lab Assignment 4**

**Aim:** The following are maintained by abook dealer.  
AUTHOR( **author\_id**:int , name:string , city:string , country:string )  
PUBLISHER( **publisher\_id**:int , name:string , city:string , country:string )  
CATALOGE( **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 )

1)Create the above tables by properly specifying the primary keys and foreign keys.  
2)Enter at least five tuples for each relation.  
3)Give the details of the authors who have 2 or more books in the cataloge and the price of the books is greater than the average price of the books in the cataloge and the year of publication is after 2000.  
4)Find the author of the book that has maximum sales.  
5)Demonstrate how you increase the price of books published by a specific publisher by 10%.

**Description:**

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

SQL> create table author(author\_id number(3) primary key,name varchar2(15),city varchar2(10),country varchar2(10));

Table created.

SQL> create table publisher(publisher\_id number(3) primary key,name varchar2(10),city varchar2(10), country varchar2(10));

Table created.

SQL> create table category(category\_id number(4) primary key,descrip varchar2(10));

Table created.

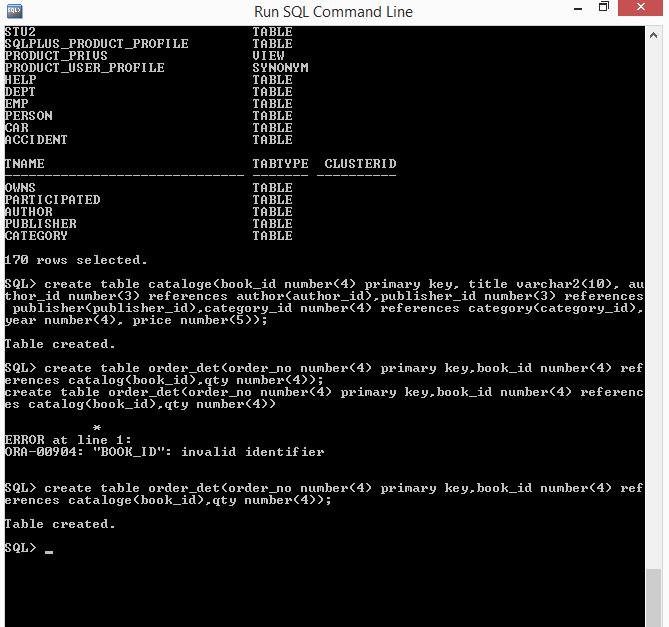
SQL> create table cataloge(book\_id number(4) primary key,title varchar2(10),author\_id number(3) references author(author\_id),publisher\_id number(3) references publisher(publisher\_id),category\_id number(4) references category(category\_id),year number(4),price number(5));

Table created.

SQL> create table order\_det(order\_no number(4) primary key,book\_id number(4) references cataloge(book\_id),qty number(4));

Table created**.**

****

****

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

SQL> insert into author values(**101 ,‘atul‘,‘pilani ‘,‘india’**);

1 row created.

SQL> insert into author values(**102 ,‘kammo‘,‘virati ‘,‘pakistan’**);

1 row created.

SQL> insert into author values(**103 ,‘bharti‘,‘virati’,‘india’**);

1 row created.

SQL> insert into author values(**104,‘partha ‘,‘delhi‘,‘hongkong’**);

1 row created.

SQL> insert into author values(**105 ,‘alom’,‘beijing ‘,‘china’**);

1 row created.

SQL> select \* from author;

**AUTHOR\_ID NAME CITY COUNTRY**

**--------- --------------- ---------- ----------**

**101 atul pilani india**

**102 kammo virati pakistan**

**103 bharti virati india**

**104 partha delhi hongkong**

**105 alom beijing china**

****

SQL> insert into publisher values(550,'pearson','arab','afgan');

1 row created.

SQL> insert into publisher values(660,'peterson','patiala','india');

1 row created.

SQL> insert into publisher values(770,'ranchor','laskar','india');

1 row created.

SQL> insert into publisher values(880,'chanchad','dehradun','india');

1 row created.

SQL> insert into publisher values(990,'munna','mumbai','india');

1 row created.

SQL> select \* from publisher;

**PUBLISHER\_ID NAME CITY COUNTRY**

**------------ ---------- ---------- ----------**

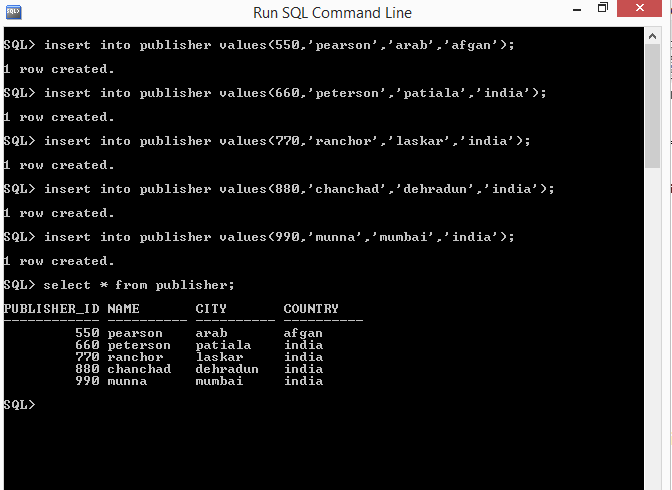
**550 pearson arab afgan**

**660 peterson patiala india**

**770 ranchor laskar india**

**880 chanchad dehradun india**

**990 munna mumbai india**

****

SQL> insert into category values(011,'java');

1 row created.

SQL> insert into category values(012,'dbms');

1 row created.

SQL> insert into category values(013,'os');

1 row created.

SQL> insert into category values(014,'maths');

1 row created.

SQL> insert into category values(015,'adc');

1 row created.

SQL> select \* from category;

**CATEGORY\_ID DESCRIP**

**----------- ----------**

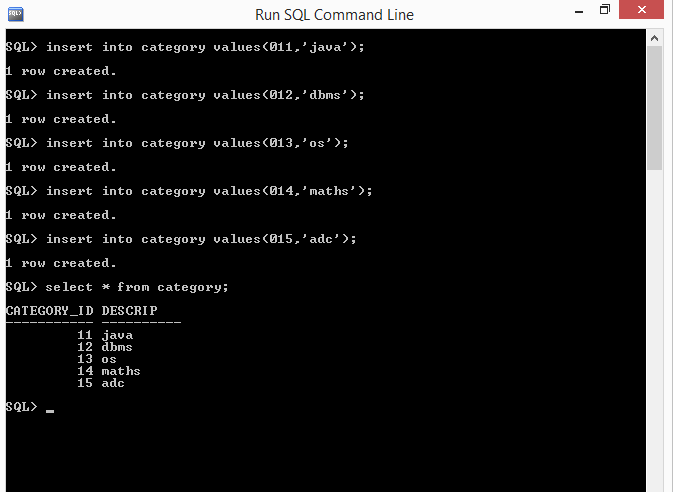
**11 java**

**12 dbms**

**13 os**

**14 maths**

**15 adc**

****

SQL> insert into cataloge values(**700 ,‘oracle’,101,660,13,2009,667**);

1 row created.

SQL> insert into cataloge values(**702 ,‘accounting’,101,660,13,2009,667**);

1 row created.

SQL> insert into cataloge values(**705 ,‘threading’,101,660,13,2009,667**);

1 row created.

SQL> insert into cataloge values(**709 ,‘algebra’,101,660,13,2009,667**);

1 row created.

SQL> select \* from cataloge;

**BOOK\_ID TITLE AUTHOR\_ID PUBLISHER\_ID CATEGORY\_ID YEAR PRICE**

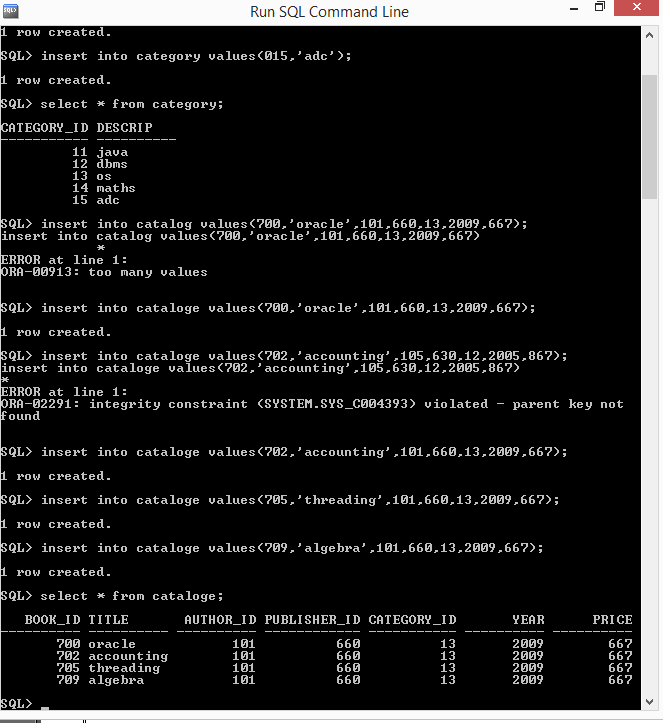
**--------- --------- - --------- ------------ ----------- --------- ---------**

**700 oracle 101 660 13 2009 667**

**702 accounting 101 660 13 2009 667**

**705 threading 101 660 13 2009 667**

**709 algebra 101 660 13 2009 667**

****

SQL> insert into order\_det values(001,700,200); 1 row created.

SQL> insert into order\_det values(002,702,150); 1 row created.

SQL> insert into order\_det values(003,705,170); 1 row created.

SQL> insert into order\_det values(004,709,190); 1 row created.

SQL> insert into order\_det values(005,705,290); 1 row created.

SQL> select \* from order\_det;

**ORDER\_NO BOOK\_ID QTY**

**--------- --------- ---------**

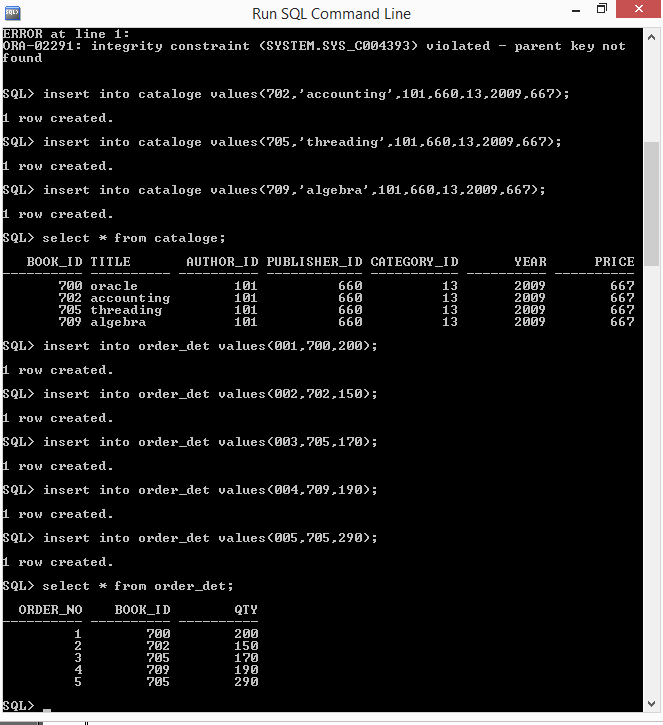
**1 700 200**

**2 702 150**

**3 705 170**

**4 709 190**

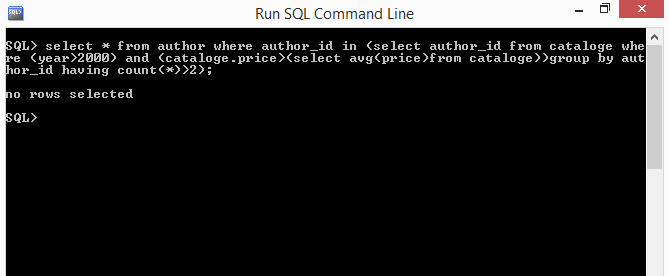
**5 705 290**

****

1. **Give the details of the authors who have 2 or more books in the cataloge and the price of the books is greater than the average price of the books in the cataloge and the year of publication is after 2000.**

SQL> select \* from author where author\_id in (select author\_id from cataloge where (year>2000) and (c atalog.price>(select avg(price)from cataloge))group by author\_id having count(\*)>2);

**no rows selected**



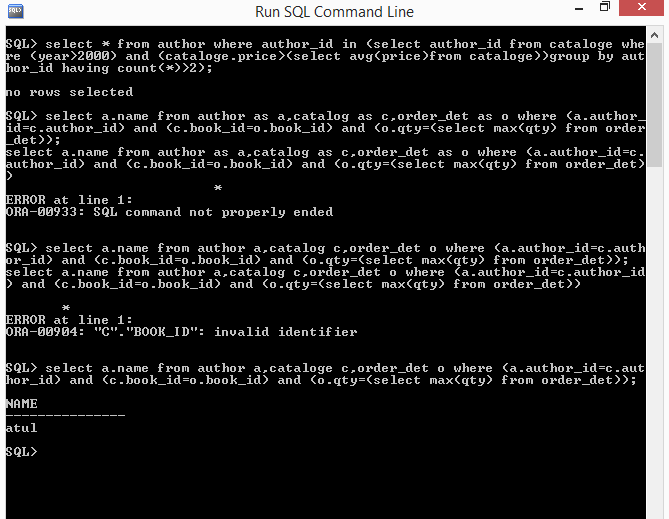
1. **Find the author of the book which has maximum sales.**

SQL> select a.name from author a,cataloge c,order\_det o where (a.author\_id=c.author\_id) and (c.book\_id=o.book\_id) and (o.qty=(select max(qty) from order\_det));

**NAME**

**---------------**

**atul**

****

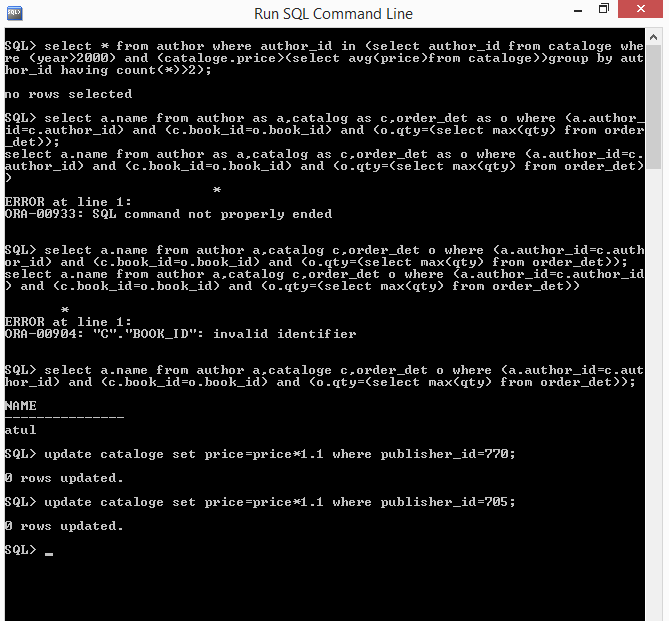
1. **Demonstrate how you increase the price of books published by a specific publisher by 10%.**

SQL> update cataloge set price=price\*1.1 where publisher\_id=770;

**no rows selected**

SQL> update cataloge set price=price\*1.1 where publisher\_id=705;

**no rows selected**



**Lab Assignment 5**

**Aim:**

Consider the following database for a banking enterprise.

BRANCH( **branch\_name**:string , branch\_city:string , assets:real )  
ACCOUNT( **accno**:int , branch\_name:string , balance:real )  
DEPOSITOR( **customer\_name**:string , **accno**:int )  
CUSTOMER( **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 )

1)Create the above tables by properly specifying the primary keys and foreign keys.  
2)Enter at least five tuples for each relation.  
3)Find ***all*** the customers who have at least two accounts at the ***main*** branch.  
4)Find all the customers who have an account at ***all*** the branches located in a specific city.  
5)Demonstrate how you delete all account tuples at every branch located in a specific city.

**Description:**

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

SQL> create table branch(br\_name varchar2(10) primary key,br\_city varchar2(10),assets number(6,2));

Table created.

SQL> create table account(accno number(5) primary key,br\_name varchar2(10) references branch(br\_name),balance number(8,2));

Table created.

SQL> create table cust\_det(cust\_name varchar2(10) primary key,cust\_strt varchar2(10),cust\_city varchar2(10));

Table created.

SQL> create table depositor(cust\_name varchar2(10) references cust\_det(cust\_name),accno number(5) references account(accno));

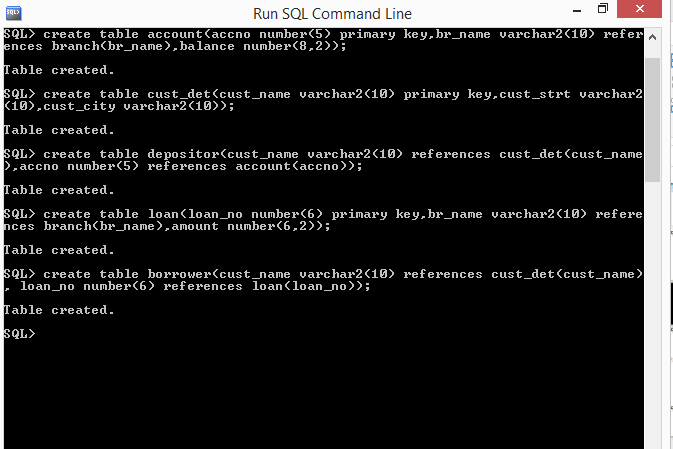
Table created.

SQL> create table loan(loan\_no number(6) primary key,br\_name varchar2(10) references branch(br\_name),amount number(6,2));

Table created.

SQL> create table borrower(cust\_name varchar2(10) references cust\_det(cust\_name), loan\_no number(6) references loan(loan\_no));

Table created.



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

SQL> insert into branch values(‘**AXIS‘,‘GWALIOR’,25000**);

1 row created.

SQL> insert into branch values(‘**HDFC’ ,‘BANGALORE’,80000**);

1 row created.

SQL> insert into branch values(‘**IDBI ‘,‘DELHI’,50000**);

1 row created.

SQL> insert into branch values(‘**SBOI ‘,‘DELHI’,55000**);

1 row created.

SQL> insert into branch values(‘**BOB ‘,‘BANGALORE’,60000**);

1 row created.

SQL> select \* from branch;

**BR\_NAME BR\_CITY ASSETS**

**---------- ---------- ---------**

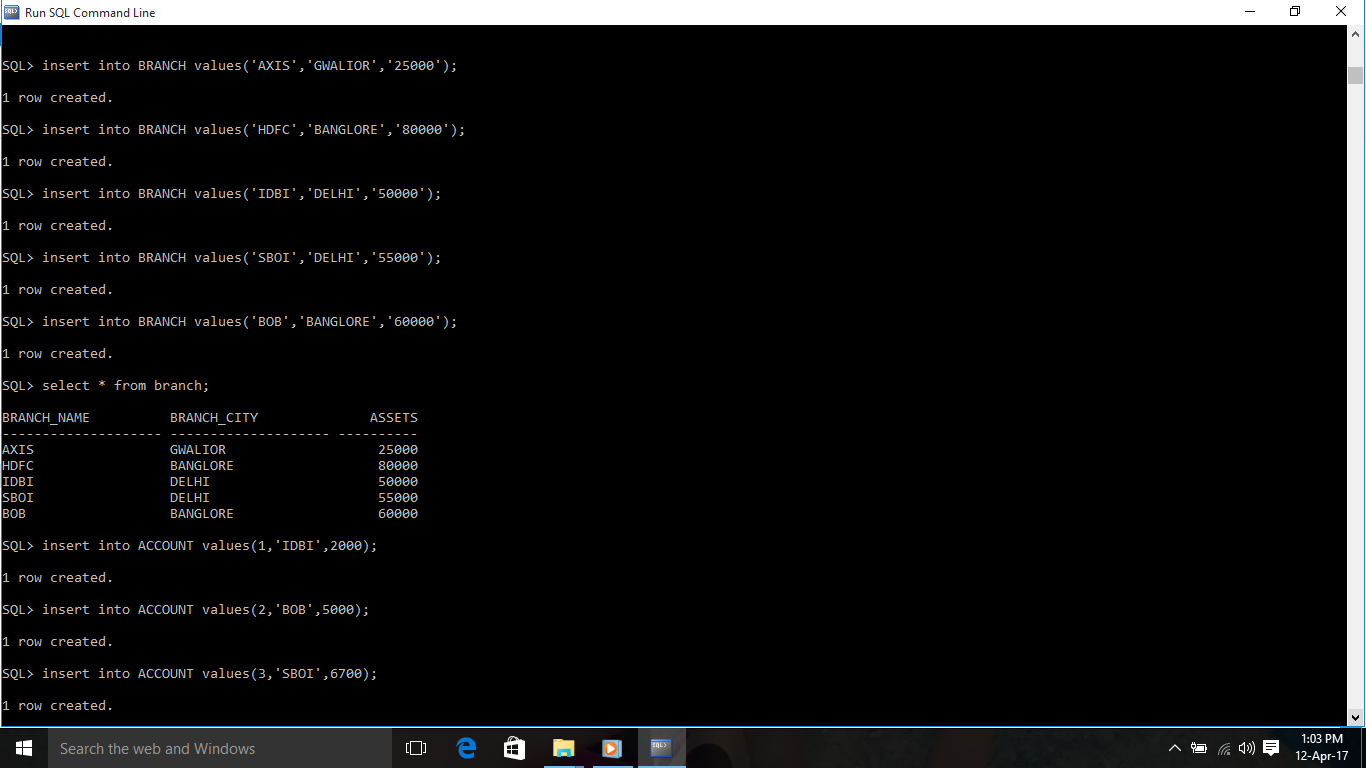
**AXIS GWALIOR 25000**

**HDFC BANGALORE 80000**

**IDBI DELHI 50000**

**SBOI DELHI 55000**

**BOB BANGALORE 60000**

****

SQL> insert into account values(**1,‘IDBI’,2000**); 1 row created.

SQL> insert into account values(**2 ,‘BOB’,5000**); 1 row created.

SQL> insert into account values(**3,‘SBOI’,6700**); 1 row created.

SQL> insert into account values(**4,‘BOB’,6700**); 1 row created.

SQL> insert into account values(**5,‘HDFC’,5600**); 1 row created.

SQL> select \* from account;

**ACCNO BR\_NAME BALANCE**

**--------- ---------- ---------**

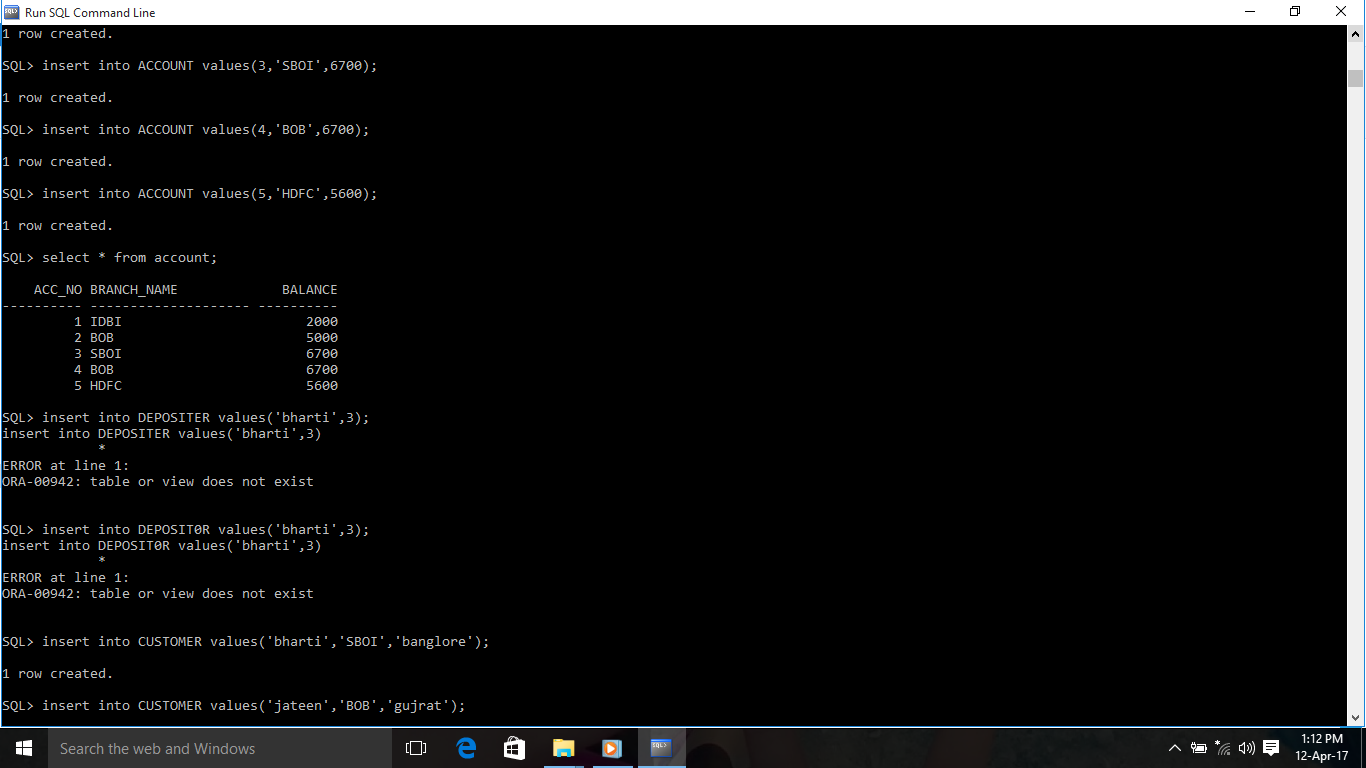
**1 IDBI 2000**

**2 BOB 5000**

**3 SBOI 6700**

**4 BOB 6700**

**5 HDFC 5600**



SQL> insert into customer values(‘**bharti‘,‘SBOI’,‘banglore’**); 1 row created.

SQL> insert into customer values(‘**jateen’,‘BOB ‘,‘gujrat’**); 1 row created.

SQL> insert into customer values(‘**atul‘,’IDBI’,’delhi’**);1 row created.

SQL> insert into customer values(‘**partha ‘,‘HDFC’,‘delhi’**); 1 row created.

SQL> insert into customer values(‘**kamlesh‘,‘SBOI ‘,‘gwalior’**); 1 row created.

SQL> select \* from customer;

**CUST\_NAME CUST\_STRT CUST\_CITY**

**---------- ---------- ----------**

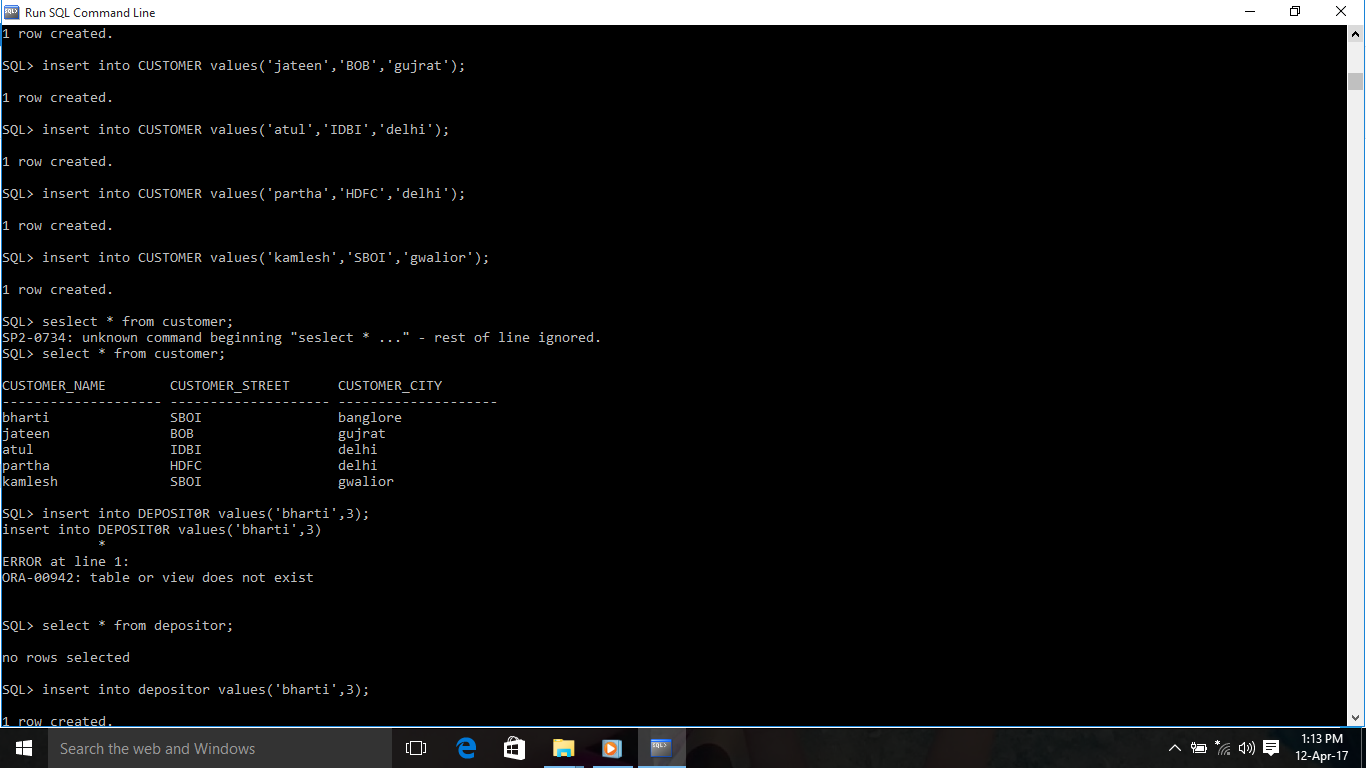
**bharti SBOI banglore**

**jateen BOB gujrat**

**atul IDBI delhi**

**partha HDFC delhi**

**kamlesh SBOI gwalior**

****

SQL> insert into depositor values(‘**atul ‘,2**); 1 row created.

SQL> insert into depositor values(‘**bharti ‘,3** ); 1 row created.

SQL> insert into depositor values(‘**jateen ‘,4**); 1 row created.

SQL> insert into depositor values(‘**kamlesh ‘,5**); 1 row created.

SQL> insert into depositor values(‘**partha’,1**); 1 row created.

SQL> select \* from depositor;

**CUST\_NAME ACCNO**

**---------- ---------**

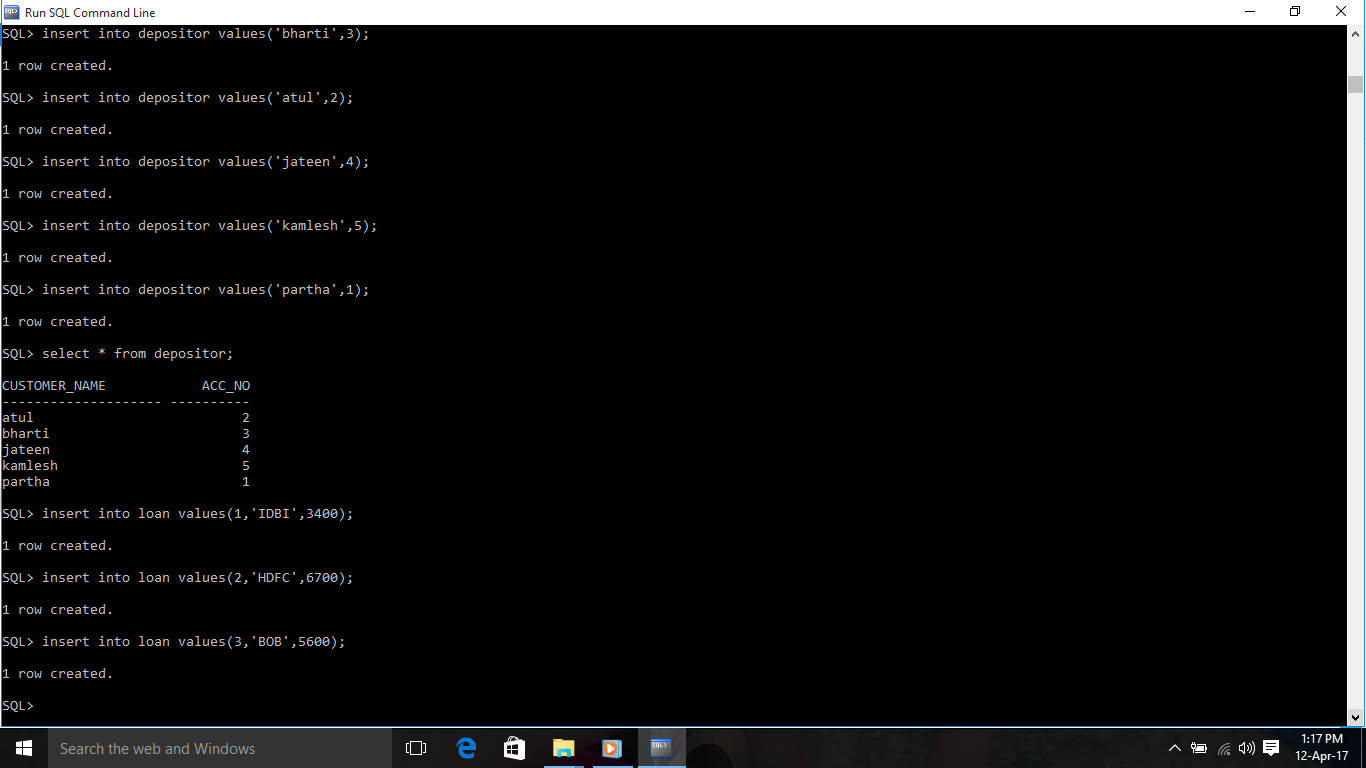
**atul 2**

**bharti 3**

**jateen 4**

**kamlesh 5**

**partha 1**

****

SQL> insert into loan values(**1,‘IDBI ‘,3400**); 1 row created.

SQL> insert into loan values(**2 ,‘HDFC ‘,6700**); 1 row created.

SQL> insert into loan values(**3,‘BOB’,5600**); 1 row created.

SQL> insert into loan values(**4 ,‘SBOI ‘,75000**); 1 row created.

SQL> insert into loan values(**5 ,’AXIS ‘,67000**); 1 row created.

SQL> select \* from loan;

**LOAN\_NO BR\_NAME AMOUNT**

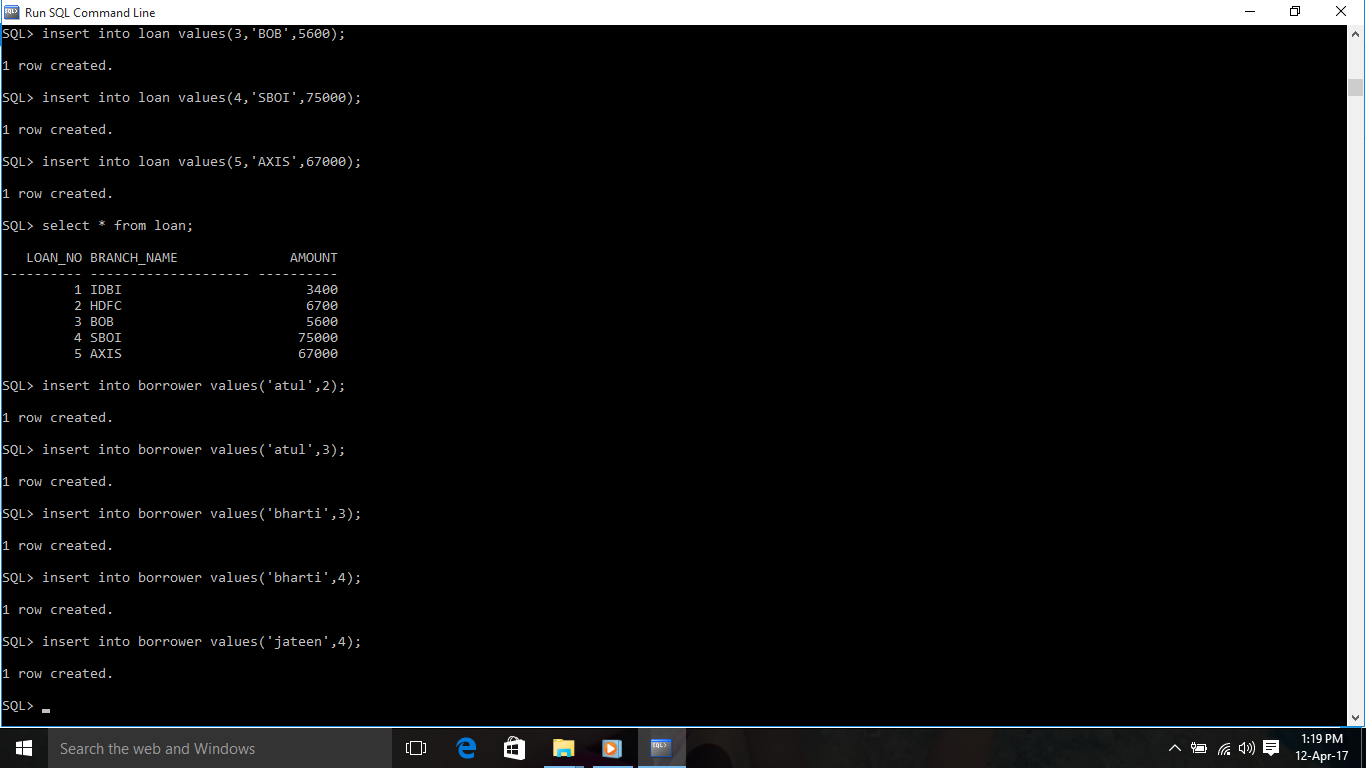
**--------- ---------- ---------**

**1 IDBI 3400**

**2 HDFC 6700**

**3 BOB 5600**

**4 SBOI 75000**

**5 AXIS 67000**

SQL> insert into borrower values(‘**atul’,2**); 1 row created.

SQL> insert into borrower values(**atul’,3**); 1 row created.

SQL> insert into borrower values(‘**bharti‘,3**); 1 row created.

SQL> insert into borrower values(‘**bharti‘,4**); 1 row created.

SQL> insert into borrower values(‘**jateen’,4**); 1 row created.

SQL> select \* from borrower;

**CUST\_NAME LOAN\_NO**

**---------- ---------**

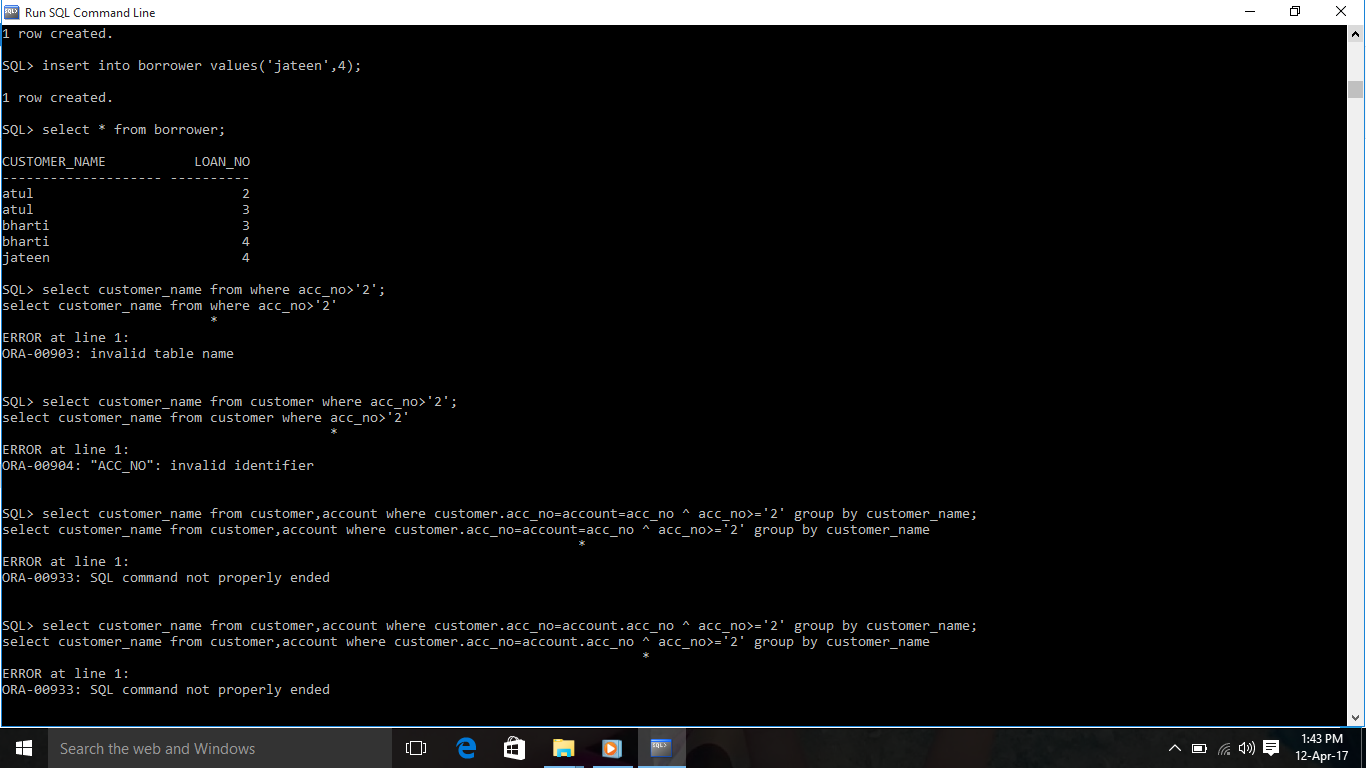
**atul 2**

**atul 3**

**bharti 3**

**bharti 4**

**jateen 4**



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

SQL> select customer.customer\_name from branch,customer,account,depositor where(depositor.acc\_no=account.acc\_no) and (account.accno=depositor.acc\_no) and (branch.branch\_name=account.branch\_name) group by customer.customer\_name having count(depositor.acc\_no)>=2;

**No rows selected**

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

SQL> select distinct depositior.customer\_name from depositor where accno in (select account.accno from account where branch\_name in(select branch\_name from branch where branch\_city='bangalore'));

**No rows selected**

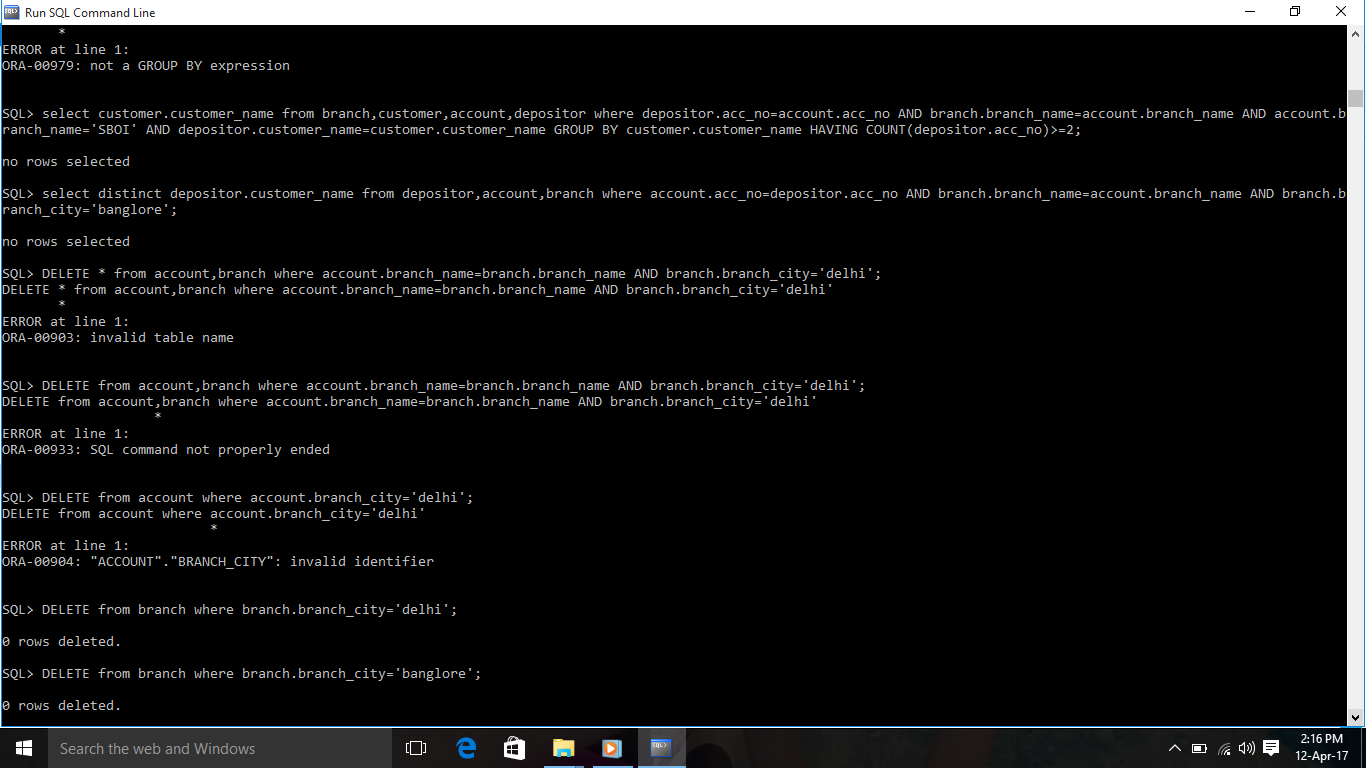
**(V)Demonstrate how you delete all account tuples at every branch located in a specific city.**

SQL> delete from depositor where accno in(select acc\_no from account ,branch where account.branch\_name=branch.branch\_name and branch\_city='delhi');

0 **rows deleted.**

SQL> delete from account where branch\_name in(select branch\_name from branch where branch\_city='bangalore');

**0 rows deleted.**

****