**DBMS ASSIGNMENT: 1**

**Name: Poojan Gandhi**

**Enroll No: AU1940125**

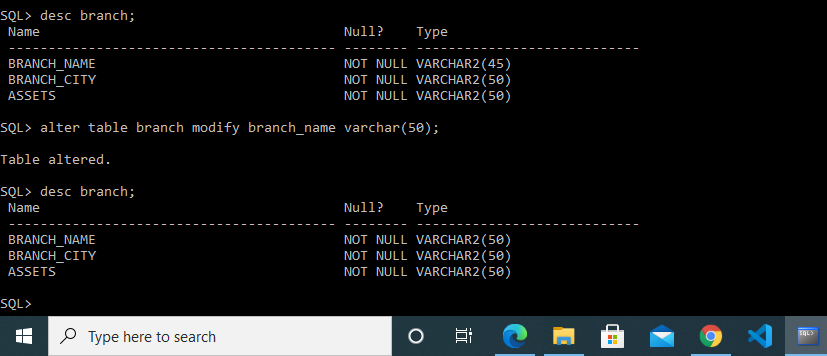
**Faculty: Shefali Naik**

Bank Table (Table 1)

Part 1

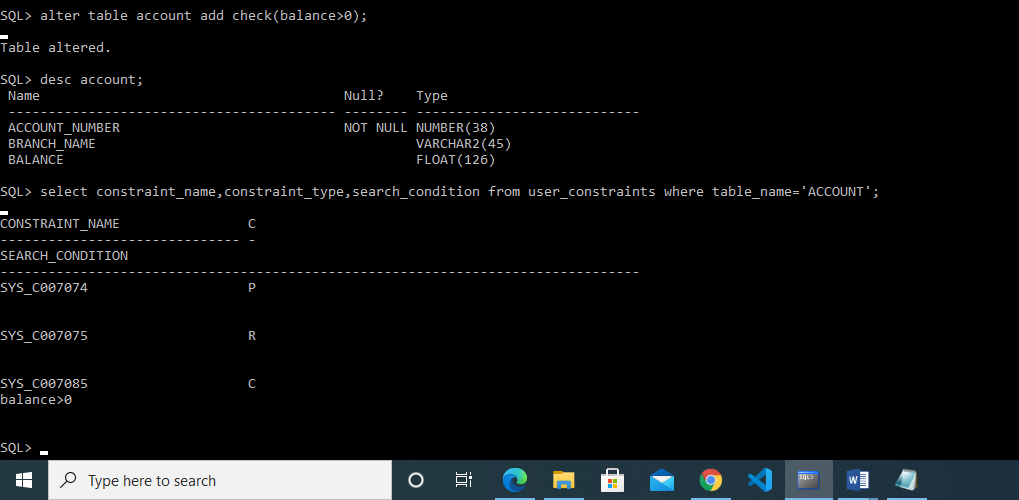
1. Increase the size of branch\_name field.

* Query: **alter table branch modify branch\_name varchar(50);**
* Image:



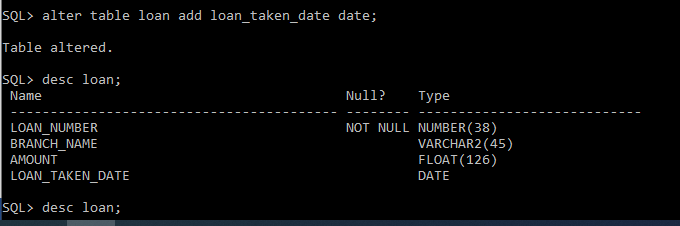
2. Add constraint to check whether balance is more than zero.

* Query: **alter table account add check(balance>0);**
* Image:



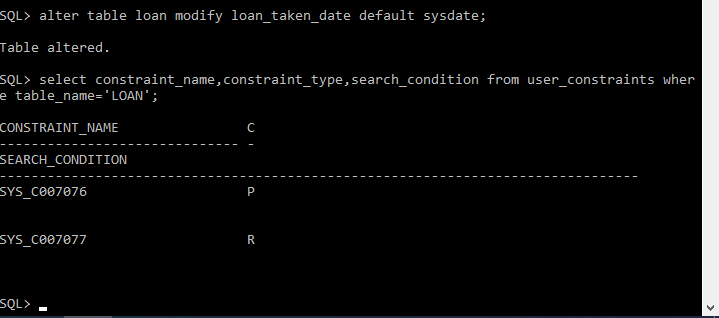
3. Add a column loan\_taken\_date in loan table.

* Query: **alter table loan add loan\_taken\_date date;**
* Image:



4. Add default constraint on loan\_taken\_date to insert current date

* Query: **alter table loan modify loan\_taken\_date default sysdate;**
* Image:



5. Drop foreign key constraint from loan\_number field of borrower table

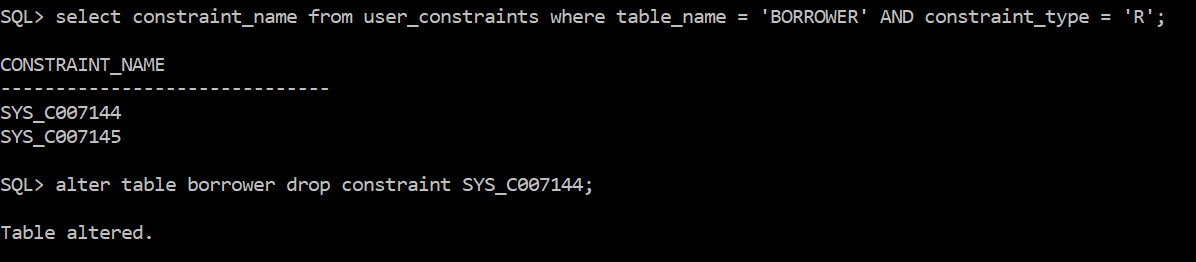
* Query:
  + For getting constraint name of foreign key:

**select constraint\_name from user\_constraints where table\_name = 'BORROWER' and constraint\_type = 'R';**

* + For deleting that key by its name:

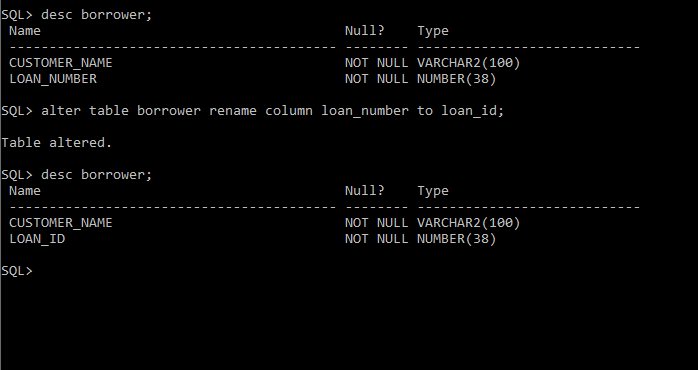
**alter table borrower drop constraint SYS\_C007144;**

* Image:



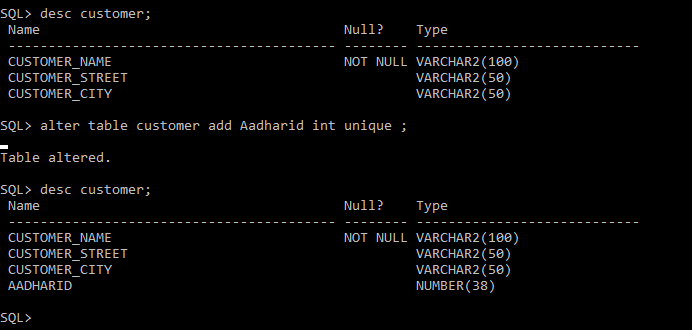
6. Rename column loan\_number of borrower table as “loan\_id”

* Query: **alter table borrower rename column loan\_number to loan\_id;**
* Image:



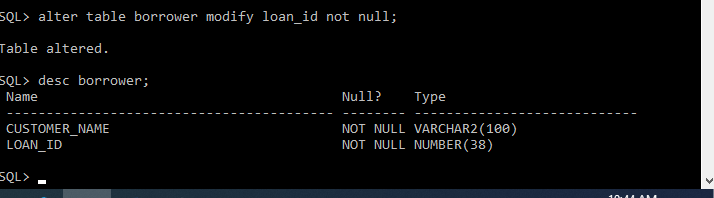
7. Add a field “Aadharid” with unique constraint in customer table

* Query: **alter table customer add Aadharid int unique ;**
* Image:



8. Add “Not Null” constraint on loan\_number field of borrower table

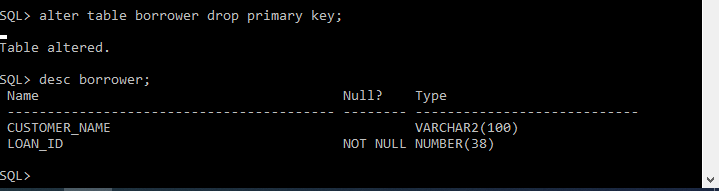
* Query: **alter table borrower modify loan\_id not null;**
* Image:



9. Drop primary key constraint from borrower table.

Query: **alter table borrower drop primary key;**

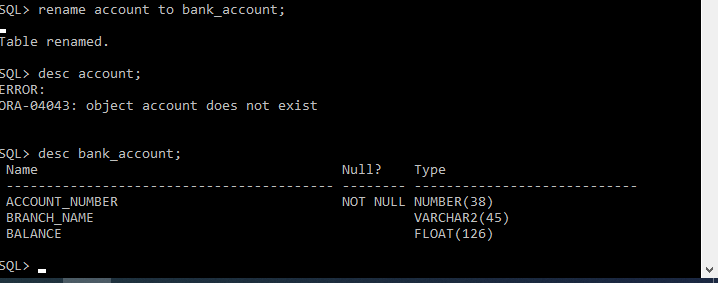
Image:



10. Rename table account to bank\_account.

Query: **rename account to bank\_account;**

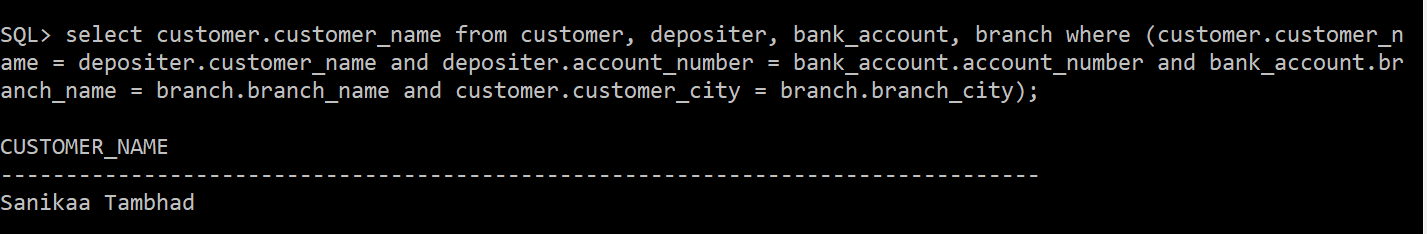
Image:



**Part 2**

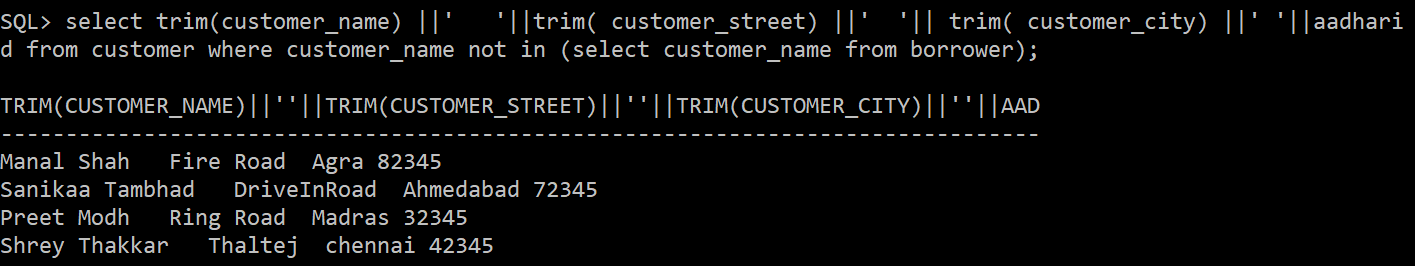
1. Display names of the customers who have account in the city where they live.

* Query: **select customer.customer\_name from customer, depositer, bank\_account, branch where (customer.customer\_name = depositer.customer\_name and depositer.account\_number = bank\_account.account\_number and bank\_account.branch\_name = branch.branch\_name and customer.customer\_city = branch.branch\_city);**
* Image:



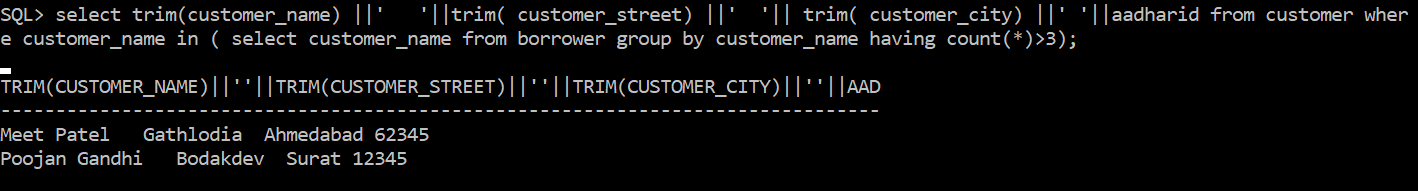
1. Display details of customers who are not borrower.

* Query: **select trim(customer\_name) ||' '||trim( customer\_street) ||' '|| trim( customer\_city) ||' '||aadharid from customer where customer\_name not in (select customer\_name from borrower);**
* Image:

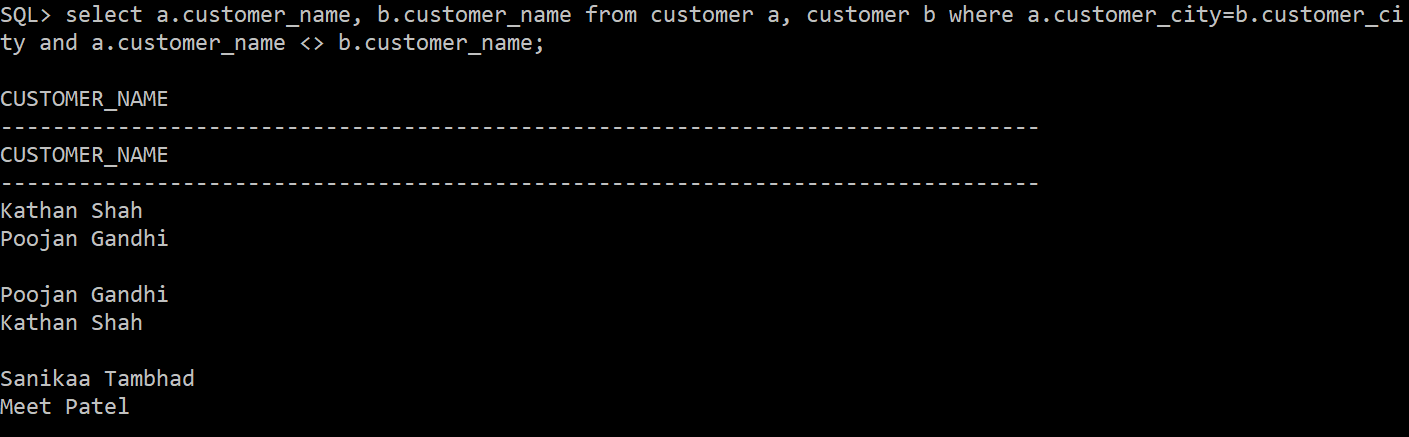


3. Display customer details who have taken loan more than 3 times

* Query: **select trim(customer\_name) ||' '||trim( customer\_street) ||' '|| trim( customer\_city) ||' '||aadharid from customer where customer\_name in ( select customer\_name from borrower group by customer\_name having count(\*)>3);**
* Image:

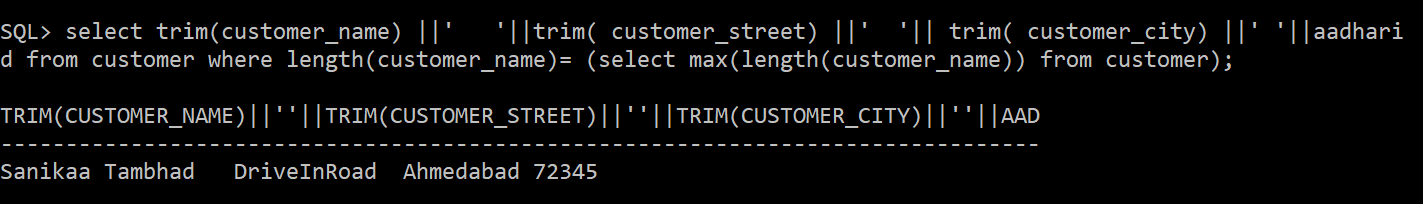


4. Display pairs of customers who live in the same city.

* Query: **select a.customer\_name, b.customer\_name from customer a, customer b where a.customer\_city=b.customer\_city and a.customer\_name <> b.customer\_name;**
* Image: 

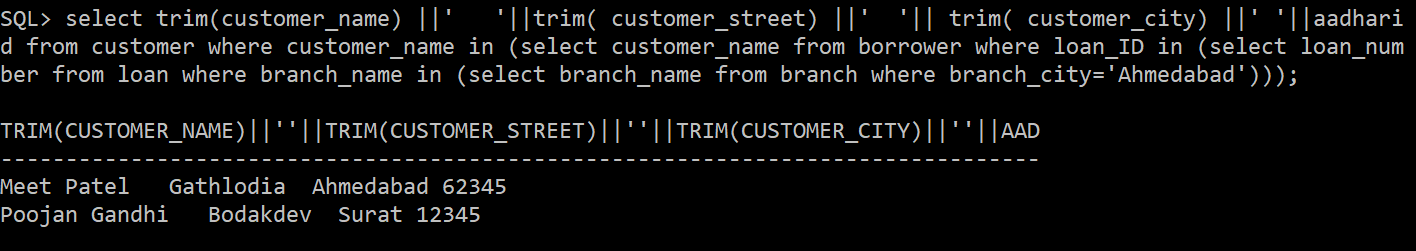
5. Display details of the customers whose name is the longest.

* Query: **select trim(customer\_name) ||' '||trim( customer\_street) ||' '|| trim( customer\_city) ||' '||aadharid from customer where length(customer\_name)= (select max(length(customer\_name)) from customer);**
* Image:



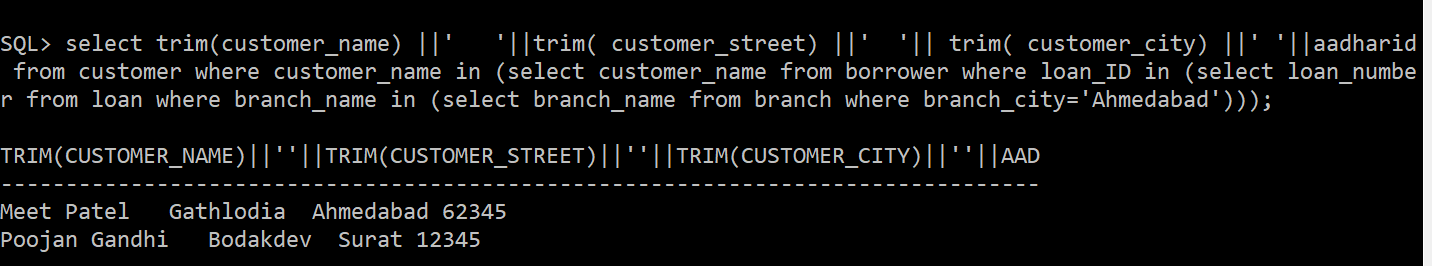
6. Display details of borrowers who have taken loan from 'Ahmedabad' branch

* Query: **select trim(customer\_name) ||' '||trim( customer\_street) ||' '|| trim( customer\_city) ||' '||aadharid from customer where customer\_name in (select customer\_name from borrower where loan\_ID in (select loan\_number from loan where branch\_name in (select branch\_name from branch where branch\_city='Ahmedabad')));**
* Image:



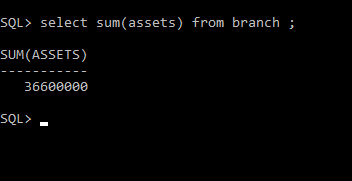
7. Display details of borrowers who have taken **total** loan of more than Rs. 50,00,000.

* Query: **select trim(customer\_name) ||' '||trim( customer\_street) ||' '|| trim( customer\_city) ||' '||aadharid from customer where customer\_name in (select customer\_name from borrower where loan\_ID in (select loan\_number from loan where branch\_name in (select branch\_name from branch where branch\_city='Ahmedabad')));**
* Image:



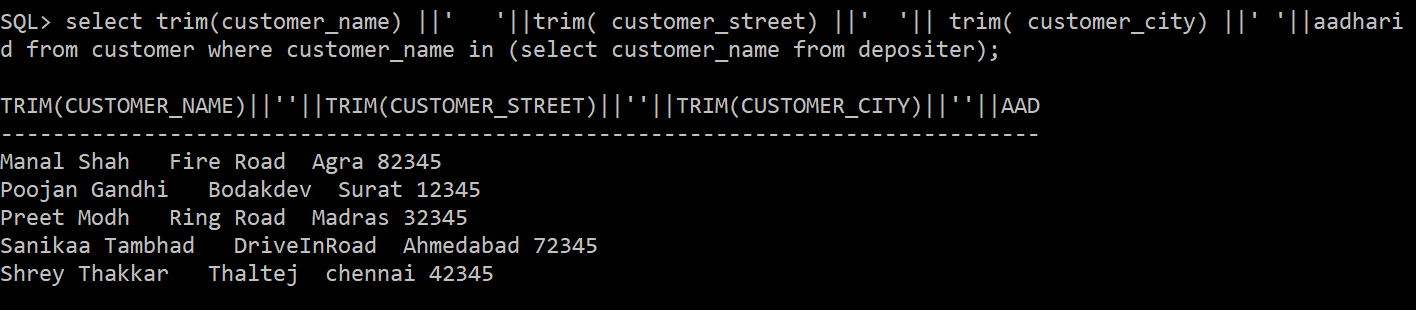
8. Display total assets of all the branches.

* Query: **select sum(assets) from branch ;**
* Image:



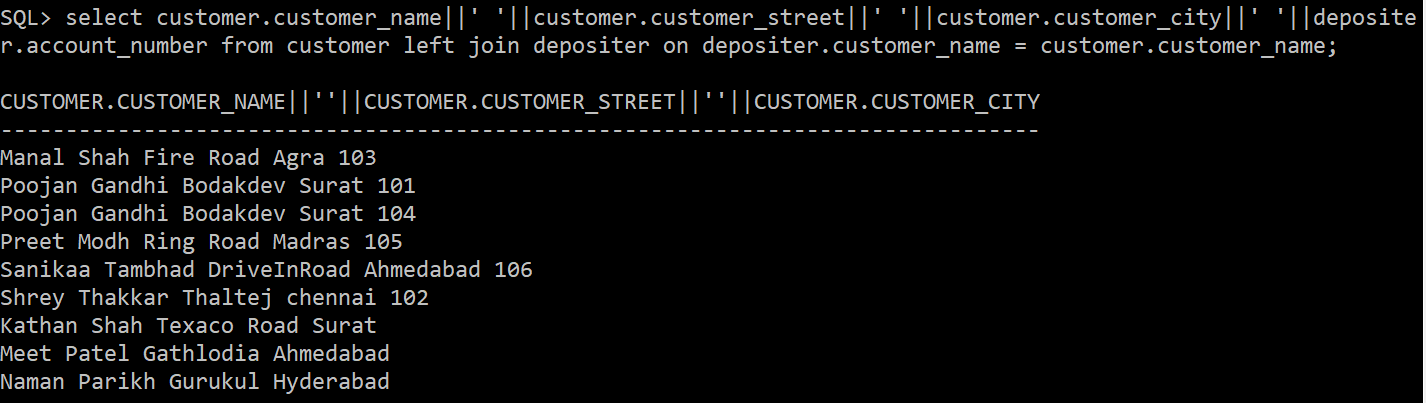
9. Display details of customers who are depositors

* Query: **select trim(customer\_name) ||' '||trim( customer\_street) ||' '|| trim( customer\_city) ||' '||aadharid from customer where customer\_name in (select customer\_name from depositer);**
* Image:



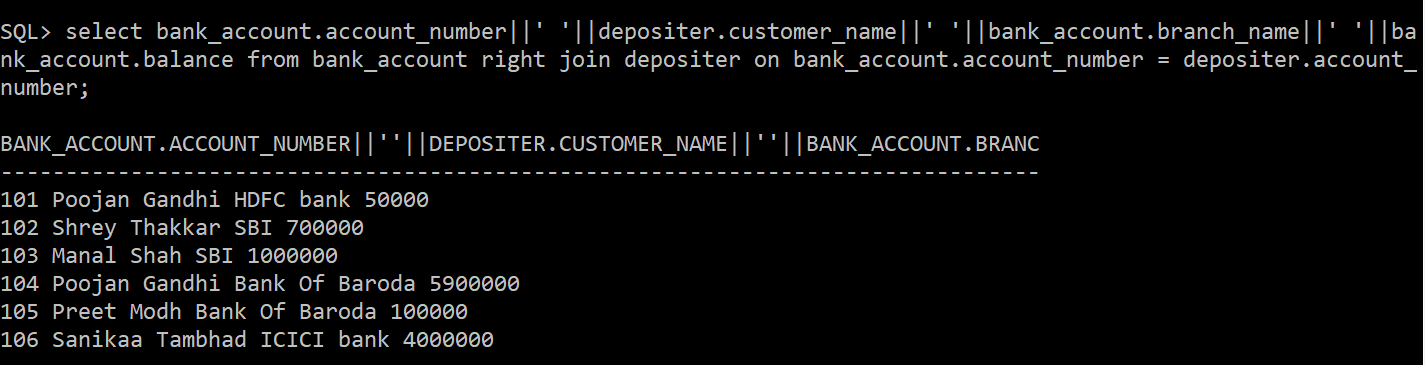
10. Display left outer join of customer and depositor table

* Query: **select customer.customer\_name||' '||customer.customer\_street||' '||customer.customer\_city||' '||depositer.account\_number from customer left join depositer on depositer.customer\_name = customer.customer\_name;**
* Image:



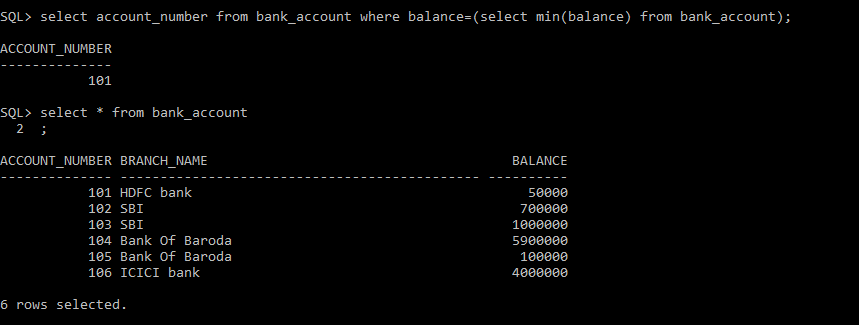
11. Display right outer join of account and depositor table.

* Query: **select bank\_account.account\_number||' '||depositer.customer\_name||' '||bank\_account.branch\_name||' '||bank\_account.balance from bank\_account right join depositer on bank\_account.account\_number = depositer.account\_number;**
* Image:



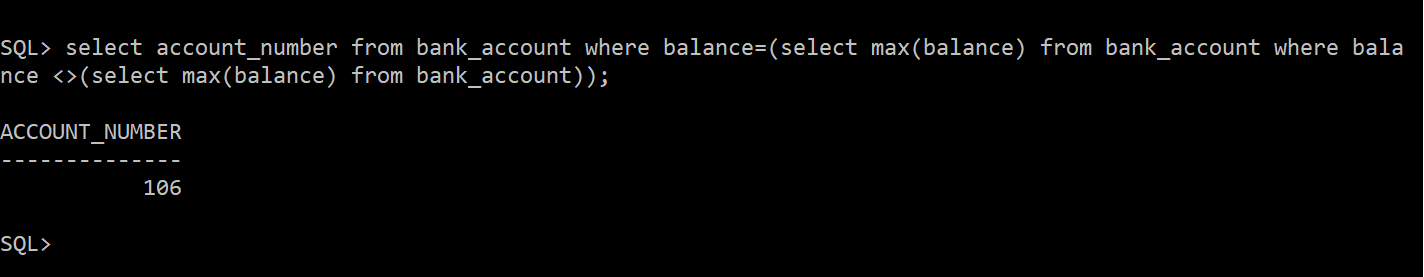
12. Display account number in which balance is minimum.

* Query: **select account\_number from bank\_account where balance=(select min(balance) from bank\_account);**
* Image:



13. Display account number in which balance is second highest

* Query: **select account\_number from bank\_account where balance=(select max(balance) from bank\_account where balance <>(select max(balance) from bank\_account));**
* Image:

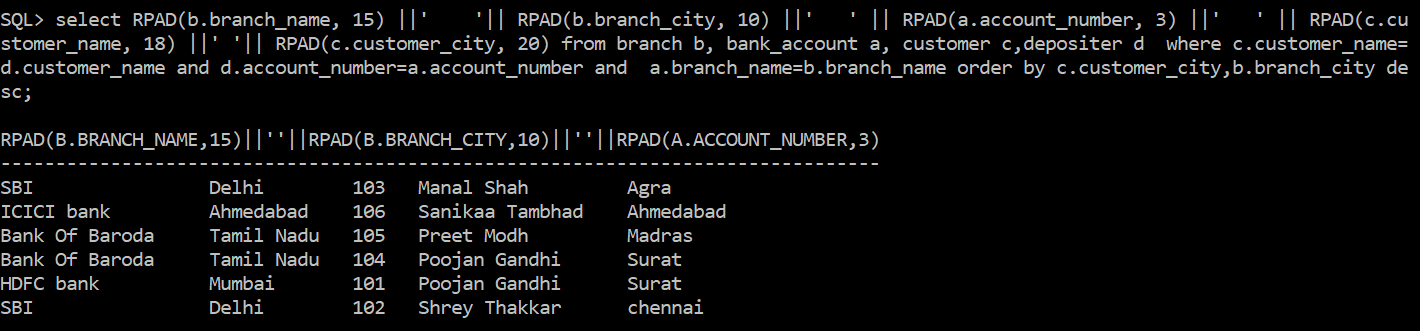


14. Display branch name, branch city, account number, customer name and customer city in ascending order of customer city and descending order of branch city

Query: **select b.branch\_name||' '||b.branch\_city||' ' ||a.account\_number||' ' ||c.customer\_name||' '||c.customer\_city from branch b, bank\_account a, customer c,depositer d where c.customer\_name=d.customer\_name and d.account\_number=a.account\_number and a.branch\_name=b.branch\_name order by c.customer\_city,b.branch\_city desc;**

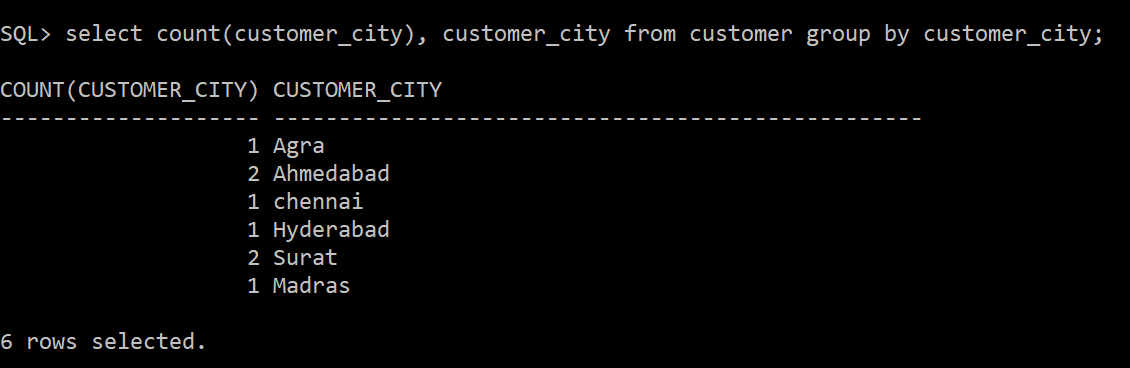
**OR**

**select RPAD(b.branch\_name, 15) ||' '|| RPAD(b.branch\_city, 10) ||' ' || RPAD(a.account\_number, 3) ||' ' || RPAD(c.customer\_name, 18) ||' '|| RPAD(c.customer\_city, 20) from branch b, bank\_account a, customer c,depositer d where c.customer\_name=d.customer\_name and d.account\_number=a.account\_number and a.branch\_name=b.branch\_name order by c.customer\_city,b.branch\_city desc;**

Image:

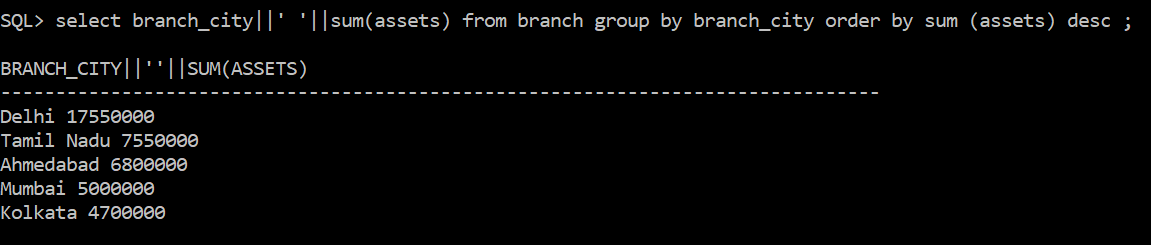
15. Display total no. of customers in each city

* Query: **select count(customer\_city), customer\_city from customer group by customer\_city;**
* Image:



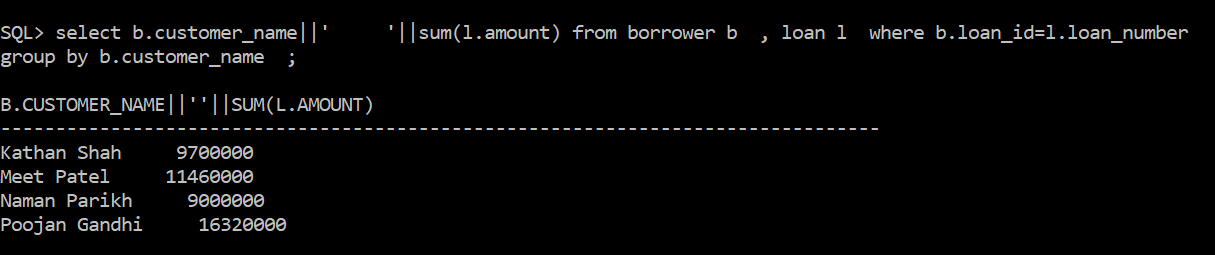
16. Display city wise total assets in descending order of total assets.

* Query: **select branch\_city||' '||sum(assets) from branch group by branch\_city order by sum (assets) desc ;**
* Image:



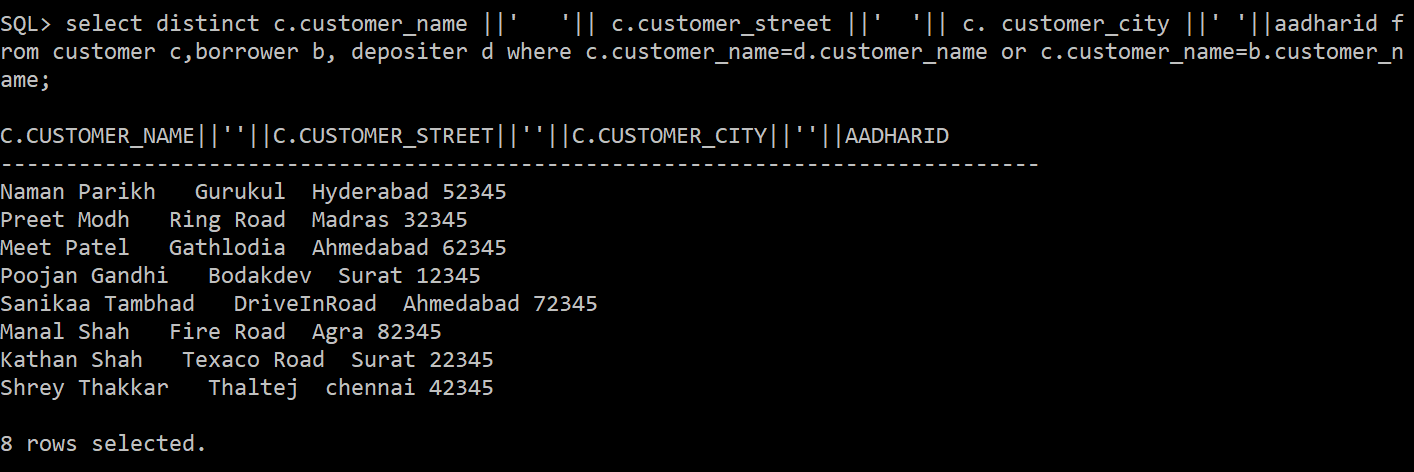
17. Display borrower names with total loan amount taken.

* Query: **select b.customer\_name||' '||sum(l.amount) from borrower b , loan l where b.loan\_id=l.loan\_number group by b.customer\_name ;**
* Image:



18. Display details of the customers who are depositors as well as borrower.

* Query: **select distinct c.customer\_name ||' '|| c.customer\_street ||' '|| c. customer\_city ||' '||aadharid from customer c,borrower b, depositer d where c.customer\_name=d.customer\_name or c.customer\_name=b.customer\_name;**
* Image:



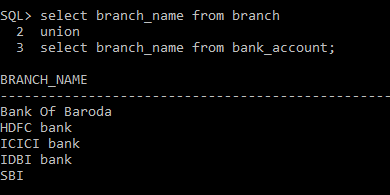
19. Display inner join of customer, borrower, loan and branch.

* Query: **select customer.customer\_name||' '||branch.branch\_name||' '|| loan.loan\_number from branch inner join loan on branch.branch\_name= loan.branch\_name inner join borrower on loan.loan\_number=borrower.loan\_id inner join customer on customer.customer\_name =borrower.customer\_name;**
* Image:



20. Display union of branch and account. (Use “union” operator)

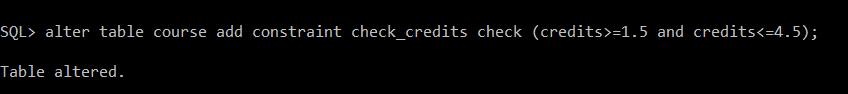
* Query: **select branch\_name from branch union select branch\_name from bank\_account;**
* Image:



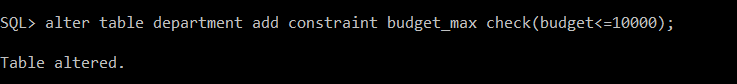
**2ND TABLE**

1. Add constraints:
   * + 1. Credits in course table should be minimum 1.5 and maximum 4.5

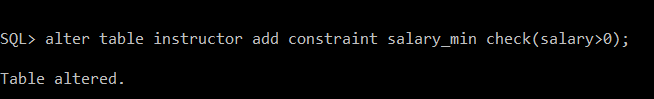
* Query:**alter table course add constraint check\_credits check (credits>=1.5 and credits<=4.5);**
* Image:



* + - 1. Budget in department table can’t exceed Rs. 10,000.
* Query:**alter table department add constraint budget\_max check(budget<=10000);**
* Image:



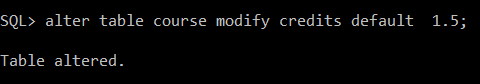
* + - 1. Salary in instructor table should be more than 0.
* Query: **alter table instructor add constraint salary\_min check(salary>0);**
* Image:



1. Default constraints

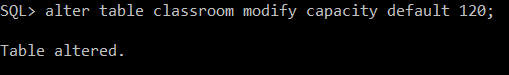
i. 1.5 credits in course table.

* + - Query: **alter table course modify credits default 1.5;**
    - Image:



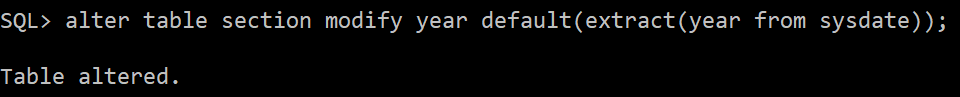
ii. 120 capacity in classroom table.

* + Query: **alter table classroom modify capacity default 120;**
  + Image



iii. 2019 (Don’t set value directly. Extract year from current date)

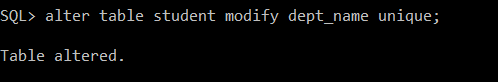
* + Query:**alter table section modify year default(extract(year from sysdate));**
  + Image:

****

1. Unique constraints:

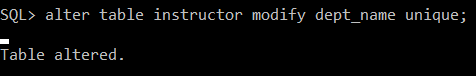
i. Dept\_name in student table.

* + Query: **alter table student modify dept\_name unique;**
  + Image:



ii. Dept\_name in instructor table.

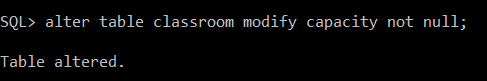
* + Query: **alter table instructor modify dept\_name unique;**
  + Image:



1. NOT NULL

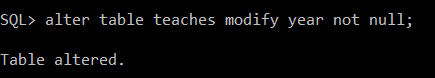
i. Capacity in classroom table.

* + Query: **alter table classroom modify capacity not null;**
  + Image:



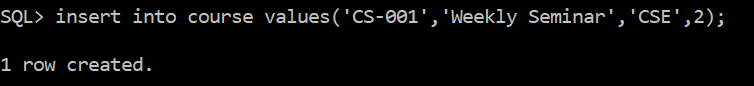
ii. Year in teaches table

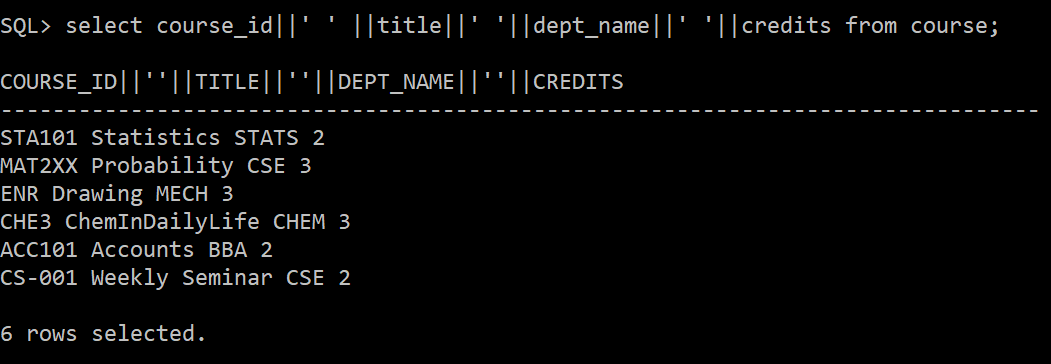
* + Query: **alter table teaches modify year not null;**
  + Image:



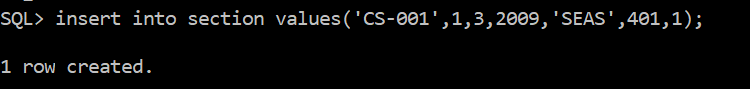
1. Create a new course “CS-001”, titled “Weekly Seminar”, with 2 credits.

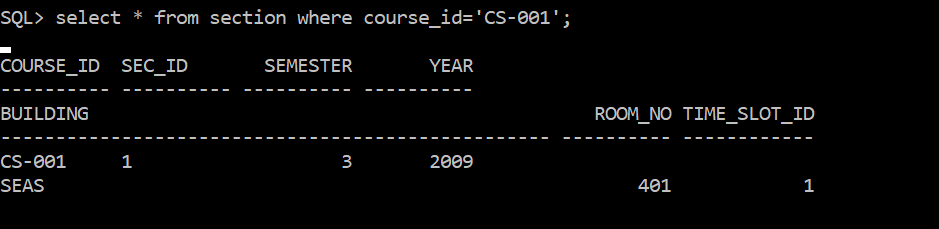
* Query: **insert into course values('CS-001','Weekly Seminar','CSE',2);**
* Image:





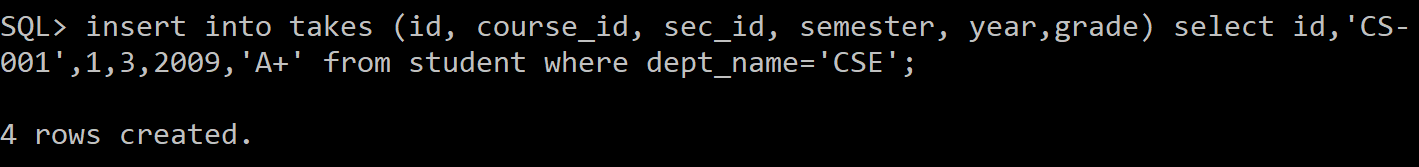
1. Create a section of the course “CS-001” in Autumn 2009, with sec\_id of 1
   * Query:**insert into section values('CS-001',1,3,2009,'SEAS',401,1);**
   * Image:

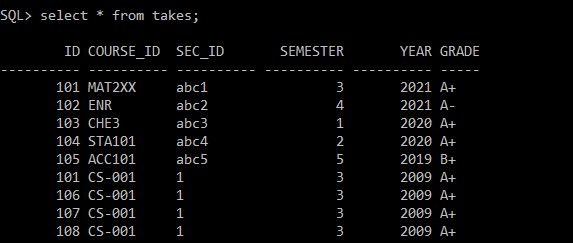


****

1. Enroll every student in the Comp. Sci. department in section 1.

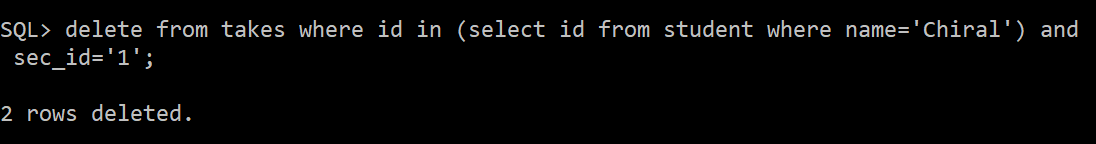
* Query: **insert into takes (id, course\_id, sec\_id, semester, year,grade) select id,'CS-001',1,3,2009,'A+' from student where dept\_name='CSE';**
* Image:





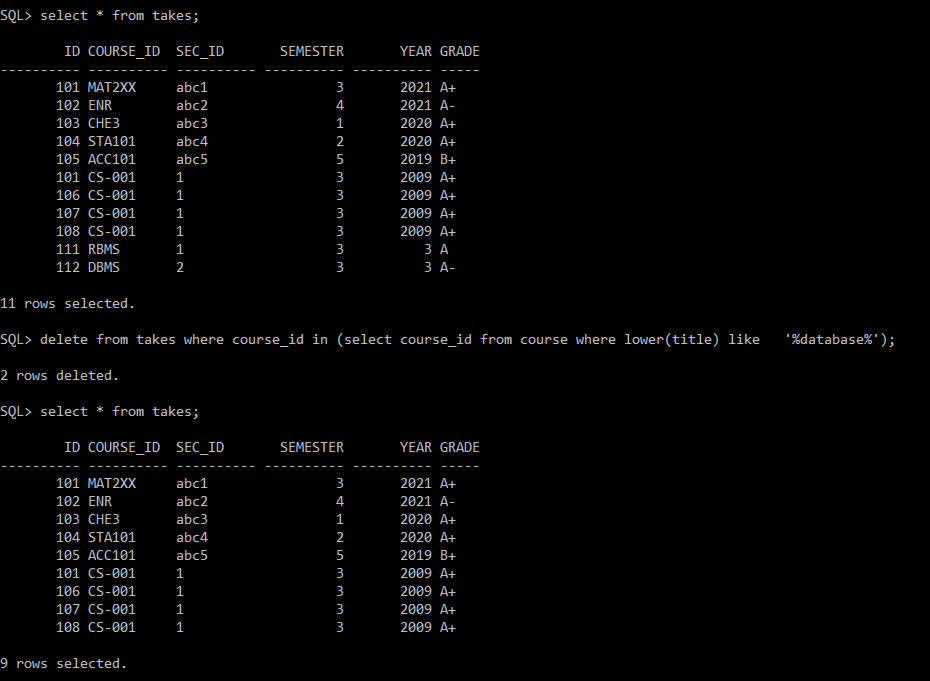
1. Delete enrollment in section 1 where student name is Chiral

* Query: **delete from takes where id in (select id from student where name='Chiral') and sec\_id='1';**
* Image:



1. Delete all “takes” tuples corresponding to any section of any course with the word “database” as a part of the title, ignore case when matching the word with the title.

* Query: **delete from takes where course\_id in (select course\_id from course where lower(title) like '%database%');**
* Image:

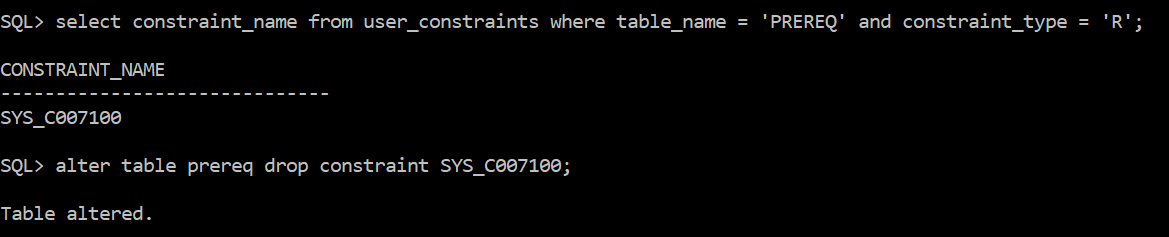


1. Drop “foreign key” constraint from “prereq” table.

Query:

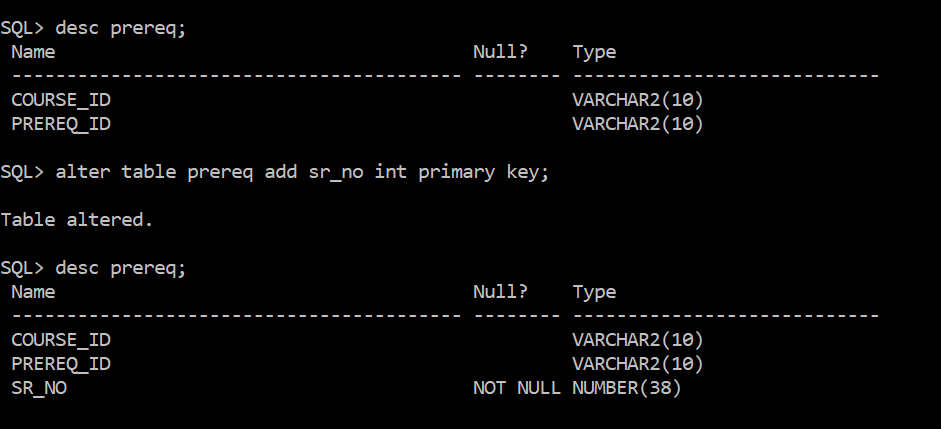
* **select constraint\_name from user\_constraints where table\_name = 'PREREQ' and constraint\_type = 'R';**
* **alter table prereq drop constraint SYS\_C007100;**

Image:



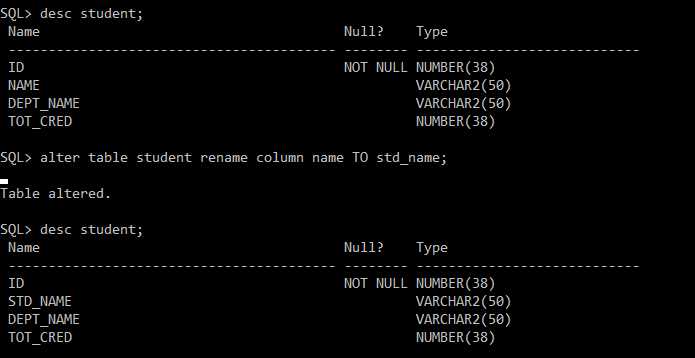
1. Add one field sr\_no “prereq” table and make it a primary key.

* Query: **alter table prereq add sr\_no int primary key;**
* Image:



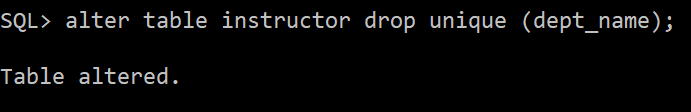
1. Rename “name” field of student table to “std\_name”.

* Query: **alter table student rename column name TO std\_name;**
* Image:



1. Drop unique constraint from “dept\_name” field of instructor table.

* Query: **alter table instructor drop unique (dept\_name);**
* Image:



1. Drop not null constraint from “capacity” field of “classroom” table.

* Query: **alter table classroom modify capacity null;**
* Image:

