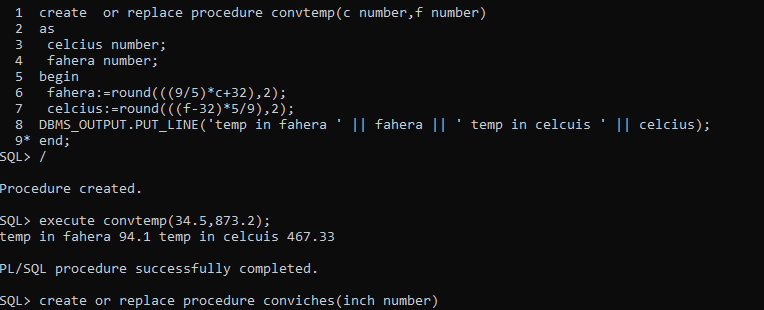
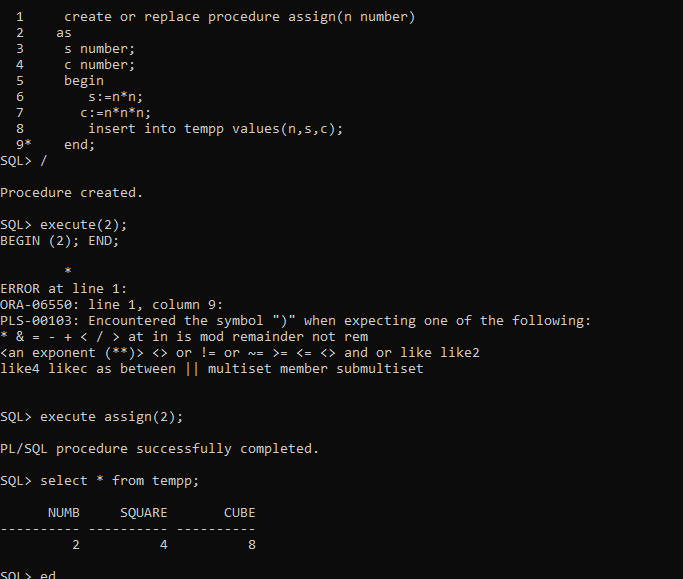
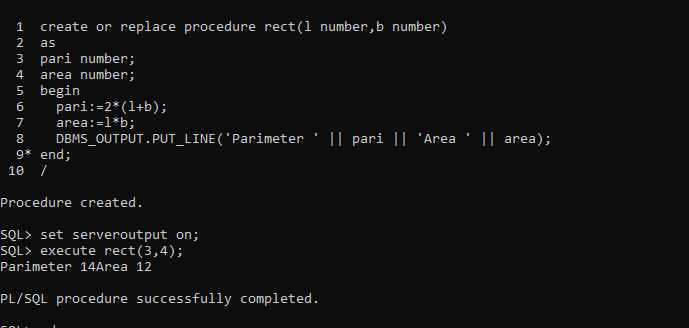
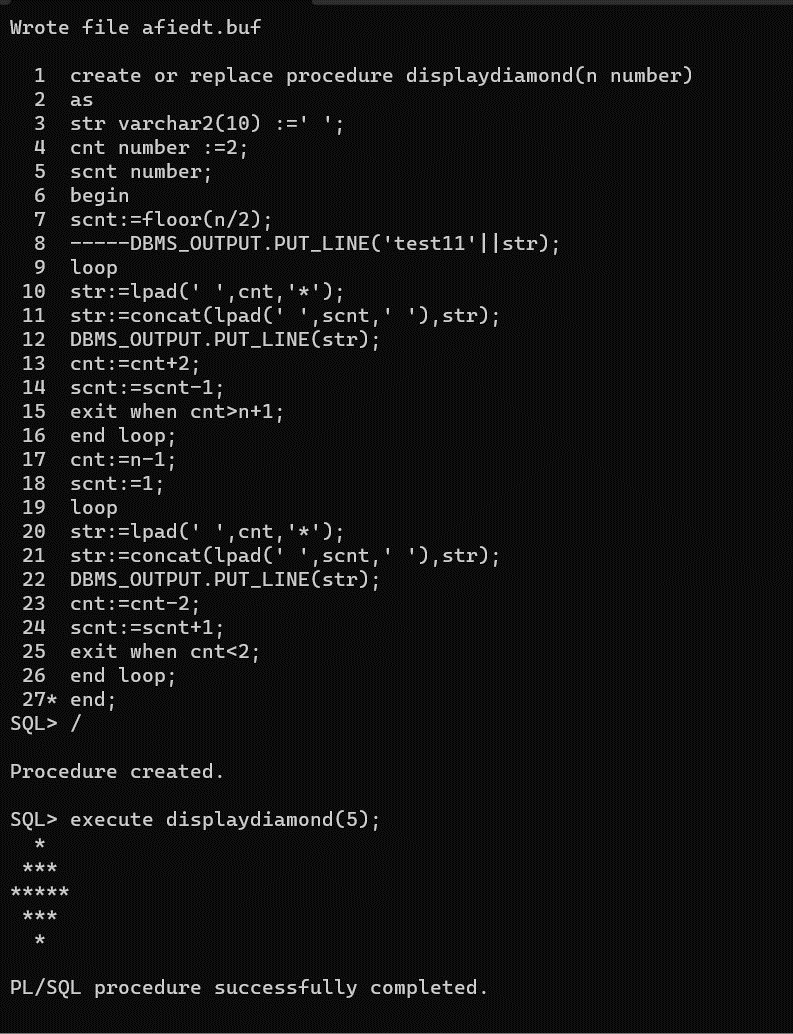
**

**

**

**

*PS-SQL assignment*

*1)*

*1 create or replace procedure perimeter(l number,b number)*

*2 as*

*3 peri number;*

*4 area number;*

*5 begin*

*6 peri:=2\*(l+b);*

*7 area:=l\*b;*

*8 DBMS\_OUTPUT.PUT\_LINE('perimeter=' || peri || 'area= ' || area);*

*9\* end;*

*SQL> /*

*Procedure created.*

*SQL> set serveroutput on;*

*SQL> execute perimeter(1,2);*

*perimeter=6area= 2*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*2)*

*create table number\_1*

*2 (*

*3 num number,*

*4 square number,*

*5\* cube number)*

*SQL> /*

*Table created.*

*SQL> ed*

*Wrote file afiedt.buf*

*1 create or replace procedure sq\_cube(n number)*

*2 as*

*3 sq number;*

*4 cube number;*

*5 begin*

*6 sq:=n\*n;*

*7 cube:=n\*n\*n;*

*8 insert into number\_1 values(n,sq,cube);*

*9\* end;*

*SQL> /*

*Procedure created.*

*SQL> execute sq\_cube(5);*

*PL/SQL procedure successfully completed.*

*SQL> select \* from number\_1;*

*NUM SQUARE CUBE*

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

*5 25 125*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*3)*

*create or replace procedure tempconverter(C number,F number)*

*2 as*

*3 celcius number;*

*4 fahrenheit number;*

*5 begin*

*6 celcius:=round(((F-32)\*5/9),2);*

*7 fahrenheit:=(9/5)\*C+35;*

*8 DBMS\_OUTPUT.PUT\_LINE('Celcius=' || celcius || 'Fahrenheit= ' || fahrenheit);*

*9\* end;*

*SQL> /*

*Procedure created.*

*SQL> execute tempconverter(35,298);*

*Celcius=147.78Fahrenheit= 98*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*4)*

*create or replace procedure inchconverter(inch number)*

*2 as*

*3 yard number;*

*4 foot number;*

*5 inc number;*

*6 begin*

*7 yard:=floor(inch/36);*

*8 foot:=floor(mod(inch,36)/12);*

*9 inc:=floor(mod(mod(inch,36),12));*

*10 DBMS\_OUTPUT.PUT\_LINE('Yards =' || yard|| 'Foot= ' || foot || 'Inches=' || inc);*

*11\* end;*

*SQL> /*

*Procedure created.*

*SQL> execute inchconverter(124);*

*PL/SQL procedure successfully completed.*

*SQL> set serveroutput on;*

*SQL> execute inchconverter(124);*

*Yards =3Foot= 1Inches=4*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*5)*

*create or replace procedure divisible(num number)*

*2 as*

*3 begin*

*4 if(mod(num,5)=0)*

*5 then*

*6 DBMS\_OUTPUT.PUT\_LINE(num ||'is Divisible by 5');*

*7 else*

*8 DBMS\_OUTPUT.PUT\_LINE(num ||'is not Divisible by 5');*

*9 end if;*

*10\* end;*

*SQL> /*

*Procedure created.*

*SQL> execute divisible(15);*

*15is Divisible by 5*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*6)*

*create or replace procedure divisible(num1 number,num2 number)*

*2 as*

*3 begin*

*4 if((num1\*num2)>100)*

*5 then*

*6 DBMS\_OUTPUT.PUT\_LINE('Product is greater than 100');*

*7 elsif((num1\*num2)=100)*

*8 then*

*9 DBMS\_OUTPUT.PUT\_LINE('Product is equal to 100');*

*10 end if;*

*11\* end;*

*SQL> /*

*Procedure created.*

*SQL> execute divisible(12,10);*

*Product is greater than 100*

*PS-SQL assignment*

*1)*

*1 create or replace procedure perimeter(l number,b number)*

*2 as*

*3 peri number;*

*4 area number;*

*5 begin*

*6 peri:=2\*(l+b);*

*7 area:=l\*b;*

*8 DBMS\_OUTPUT.PUT\_LINE('perimeter=' || peri || 'area= ' || area);*

*9\* end;*

*SQL> /*

*Procedure created.*

*SQL> set serveroutput on;*

*SQL> execute perimeter(1,2);*

*perimeter=6area= 2*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*2)*

*create table number\_1*

*2 (*

*3 num number,*

*4 square number,*

*5\* cube number)*

*SQL> /*

*Table created.*

*SQL> ed*

*Wrote file afiedt.buf*

*1 create or replace procedure sq\_cube(n number)*

*2 as*

*3 sq number;*

*4 cube number;*

*5 begin*

*6 sq:=n\*n;*

*7 cube:=n\*n\*n;*

*8 insert into number\_1 values(n,sq,cube);*

*9\* end;*

*SQL> /*

*Procedure created.*

*SQL> execute sq\_cube(5);*

*PL/SQL procedure successfully completed.*

*SQL> select \* from number\_1;*

*NUM SQUARE CUBE*

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

*5 25 125*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*3)*

*create or replace procedure tempconverter(C number,F number)*

*2 as*

*3 celcius number;*

*4 fahrenheit number;*

*5 begin*

*6 celcius:=round(((F-32)\*5/9),2);*

*7 fahrenheit:=(9/5)\*C+35;*

*8 DBMS\_OUTPUT.PUT\_LINE('Celcius=' || celcius || 'Fahrenheit= ' || fahrenheit);*

*9\* end;*

*SQL> /*

*Procedure created.*

*SQL> execute tempconverter(35,298);*

*Celcius=147.78Fahrenheit= 98*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*4)*

*create or replace procedure inchconverter(inch number)*

*2 as*

*3 yard number;*

*4 foot number;*

*5 inc number;*

*6 begin*

*7 yard:=floor(inch/36);*

*8 foot:=floor(mod(inch,36)/12);*

*9 inc:=floor(mod(mod(inch,36),12));*

*10 DBMS\_OUTPUT.PUT\_LINE('Yards =' || yard|| 'Foot= ' || foot || 'Inches=' || inc);*

*11\* end;*

*SQL> /*

*Procedure created.*

*SQL> execute inchconverter(124);*

*PL/SQL procedure successfully completed.*

*SQL> set serveroutput on;*

*SQL> execute inchconverter(124);*

*Yards =3Foot= 1Inches=4*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*5)*

*create or replace procedure divisible(num number)*

*2 as*

*3 begin*

*4 if(mod(num,5)=0)*

*5 then*

*6 DBMS\_OUTPUT.PUT\_LINE(num ||'is Divisible by 5');*

*7 else*

*8 DBMS\_OUTPUT.PUT\_LINE(num ||'is not Divisible by 5');*

*9 end if;*

*10\* end;*

*SQL> /*

*Procedure created.*

*SQL> execute divisible(15);*

*15is Divisible by 5*

*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*6)*

*create or replace procedure divisible(num1 number,num2 number)*

*2 as*

*3 begin*

*4 if((num1\*num2)>100)*

*5 then*

*6 DBMS\_OUTPUT.PUT\_LINE('Product is greater than 100');*

*7 elsif((num1\*num2)=100)*

*8 then*

*9 DBMS\_OUTPUT.PUT\_LINE('Product is equal to 100');*

*10 end if;*

*11\* end;*

*SQL> /*

*Procedure created.*

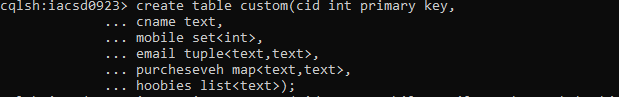
*SQL> execute divisible(12,10);*

*Product is greater than 100*

1. create a customer table to store following columns, customer id as primary key

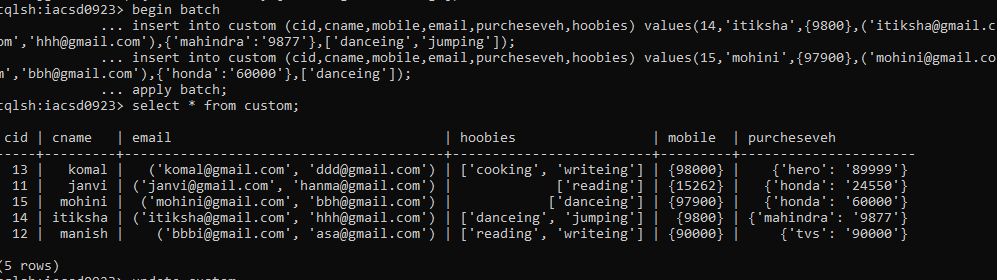
customer id, name, mobile set<int>,email tuple(text,text),purchaseveh map<text,int>,hoobies list<text>

ANS:



2. insert 5 records with customer id 11,12,13,14,15

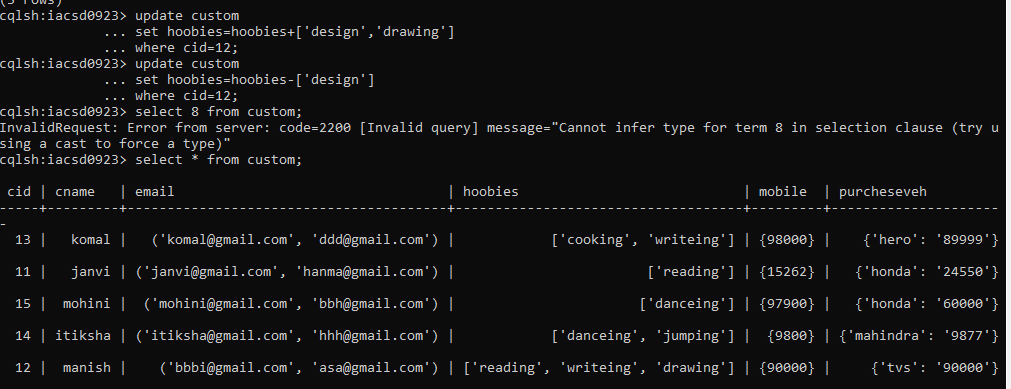
Ans:



3. add hoobies design, drawing for custid=12

4. remove hoobies design for custid=12

ANS:



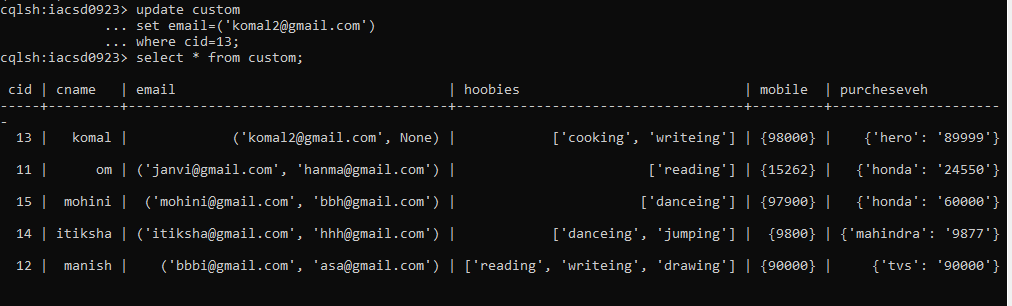
5. update name of customer with id 11

Ans:

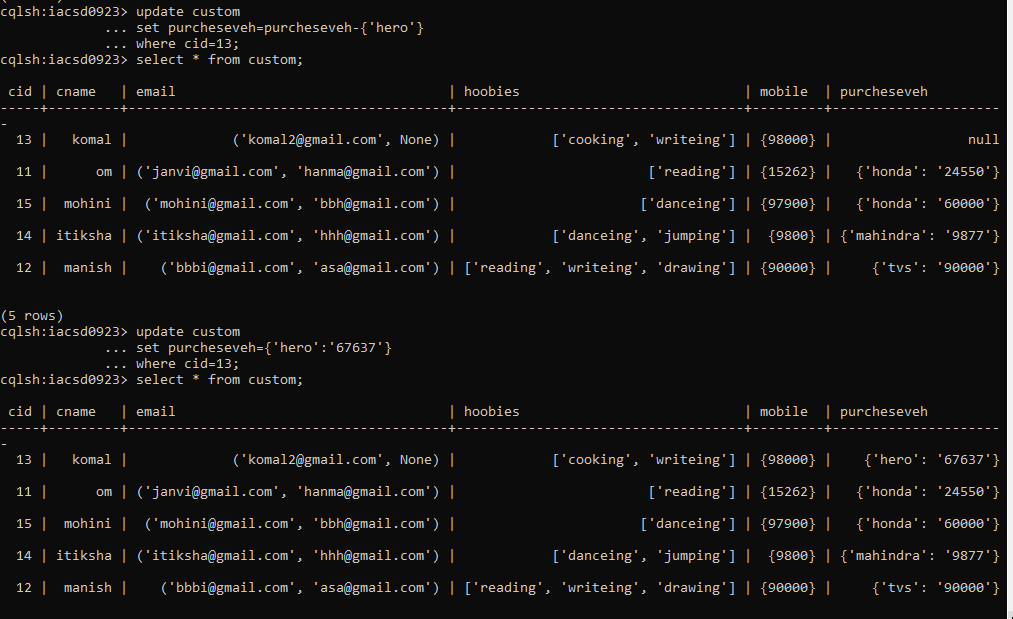


6. modify emails of customer 13

ANS:



7. add and delete one of the purchase vehicle

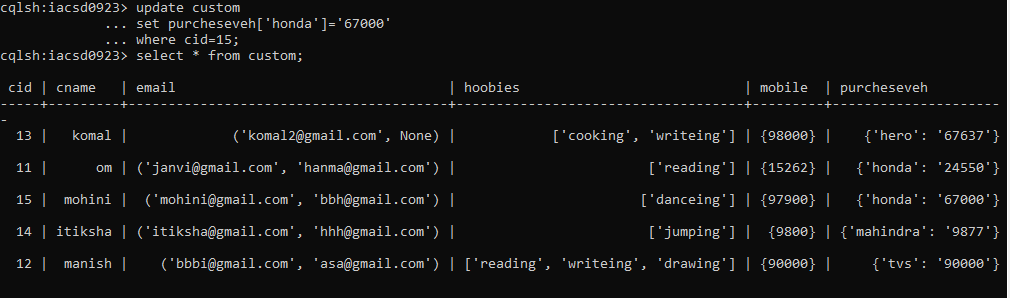


8. add and delete one of the hobbies for customer 14

Ans:



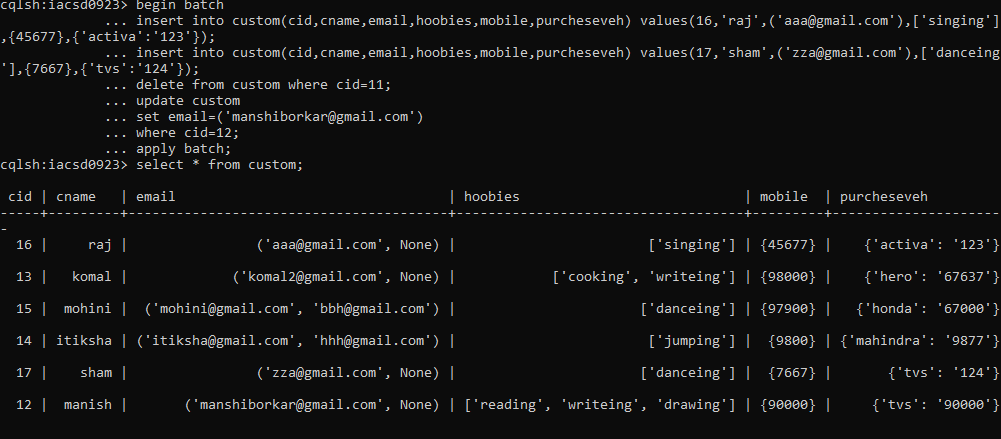
9.change the quantity of vehicle activa from customer id 15

Ans: 

Q2create a batch to add 2 records in customer table, delete record for customer 13

and delete only email from customer 11

and update emails for customer 12

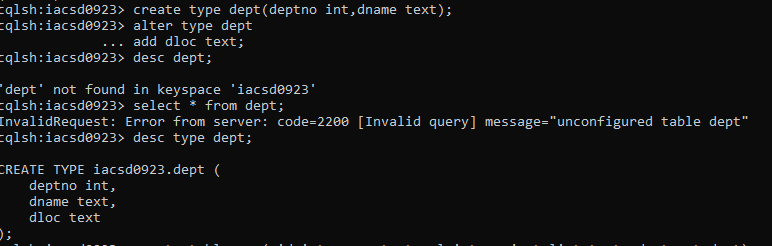
ANS: 

Q3. create type department to store deptno, dname

a.

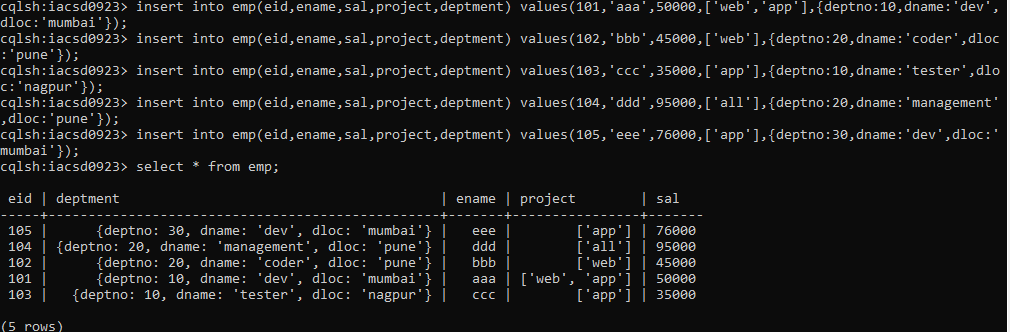
change the type department to add dloc in it.

ANS:



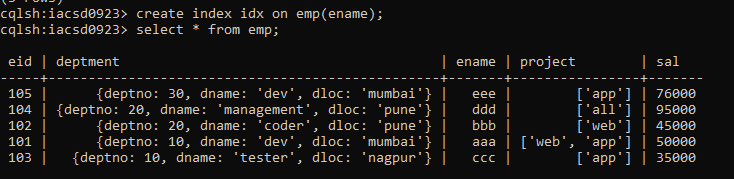
create table emp to store eid, ename, sal, projects List<text>, dept department

add 5 records in the table



create index on ename column

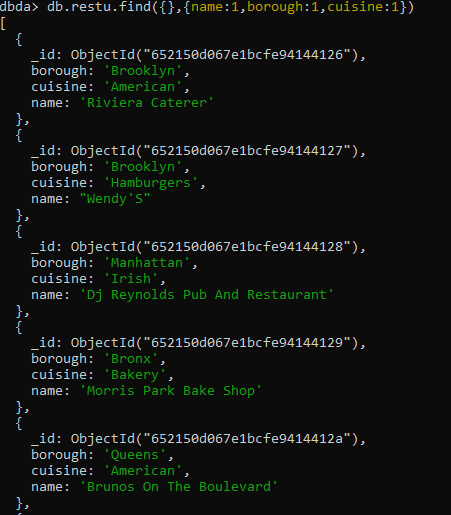
ANS:



*------MONGODB*

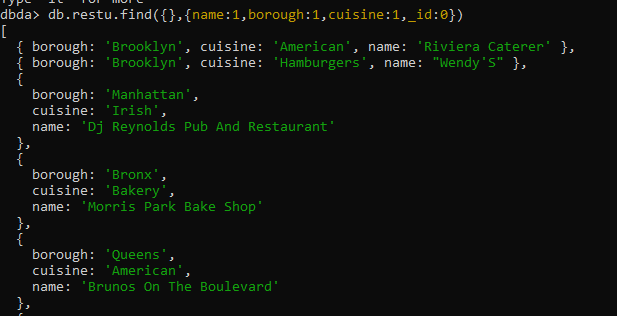
1. *Write a MongoDB query to display all the documents in the collection restaurants*

*2. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine for*

*all the documents in the collection restaurant. *

*3. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine,*

*but exclude the field \_id for all the documents in the collection restaurant.*

**

*4. Write a MongoDB query to display the fields restaurant\_id, name, borough and zip code,*

*but exclude the field \_id for all the documents in the collection restaurant.*

**

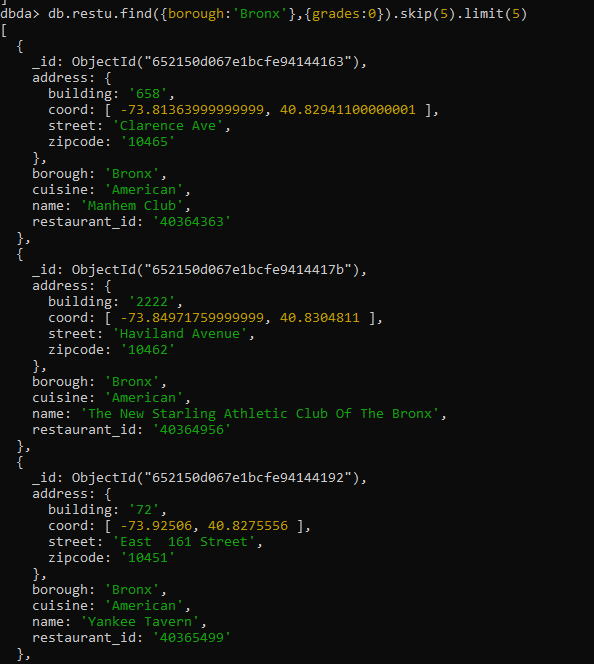
*5. Write a MongoDB query to display all the restaurant which is in the borough Bronx *

*6. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.*

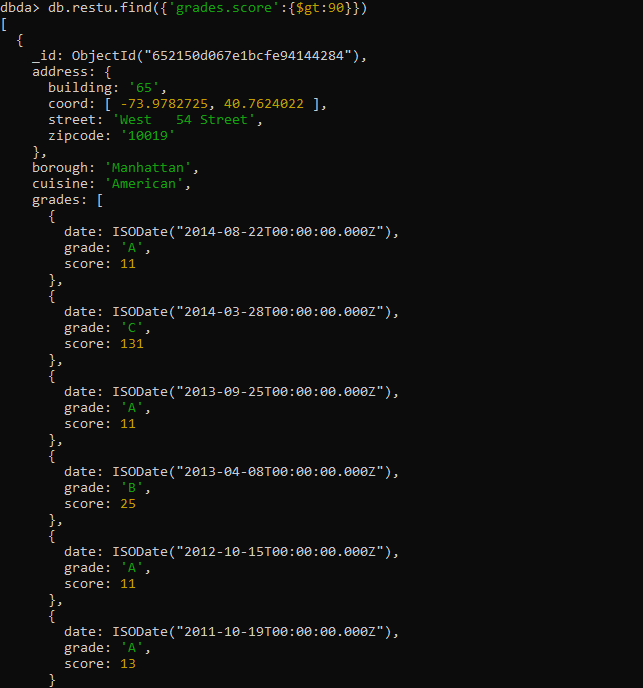
**

*7.Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in*

*the borough Bronx.*

**

*8. Write a MongoDB query to find the restaurants who achieved a score more than 90.*

**

*9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but*

*less than 100.*

**

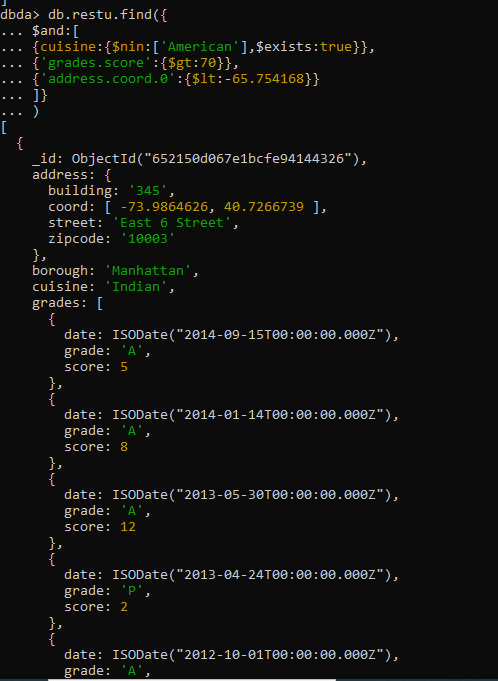
*10. Write a MongoDB query to find the restaurants which locate in latitude value less than -*

*95.754168.*

**

*11. Write a MongoDB query to find the restaurants that do not prepare any cuisine of*

*'American' and their grade score more than 70 and latitude less than -65.754168.*

**

*12. Write a MongoDB query to find the restaurants which do not prepare any cuisine of*

*'American' and achieved a score more than 70 and located in the longitude less than -*

*65.754168.*

**

*13. Write a MongoDB query to find the restaurants which do not prepare any cuisine of*

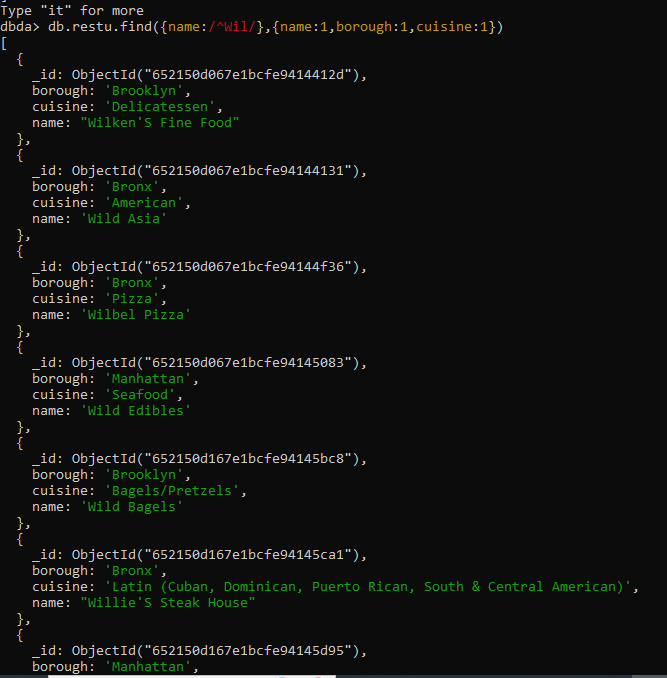
*'American ' and achieved a grade point 'A' not belongs to the borough Brooklyn. The*

*document must be displayed according to the cuisine in descending order.*

**

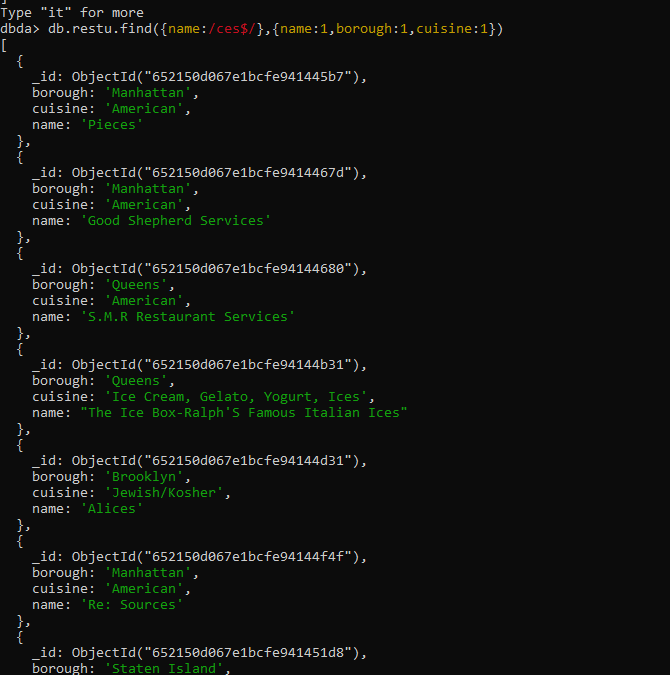
*14. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those*

*restaurants which contain 'Wil' as first three letters for its name.*

**

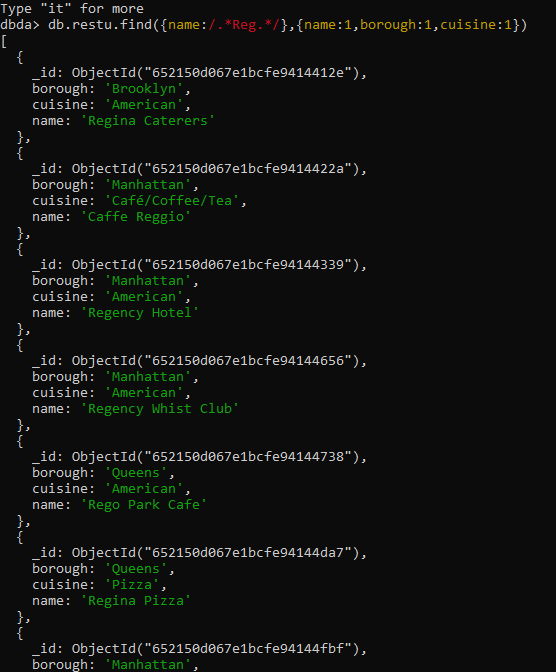
*15. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those*

*restaurants which contain 'ces' as last three letters for its name.*

**

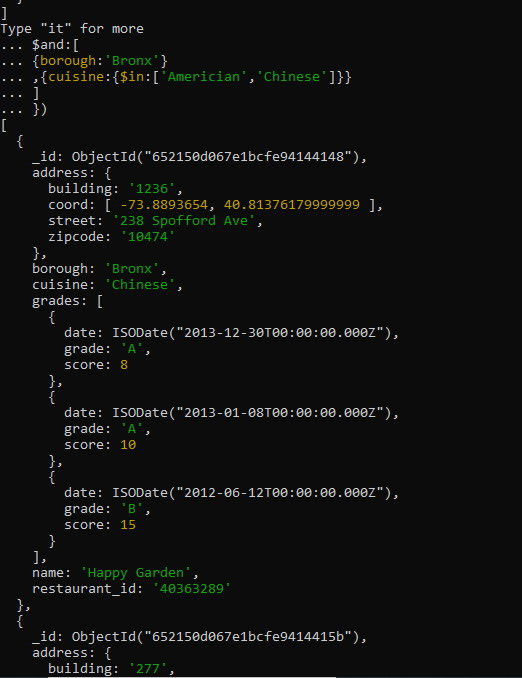
*16. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those*

*restaurants which contain 'Reg' as three letters somewhere in its name.*

**

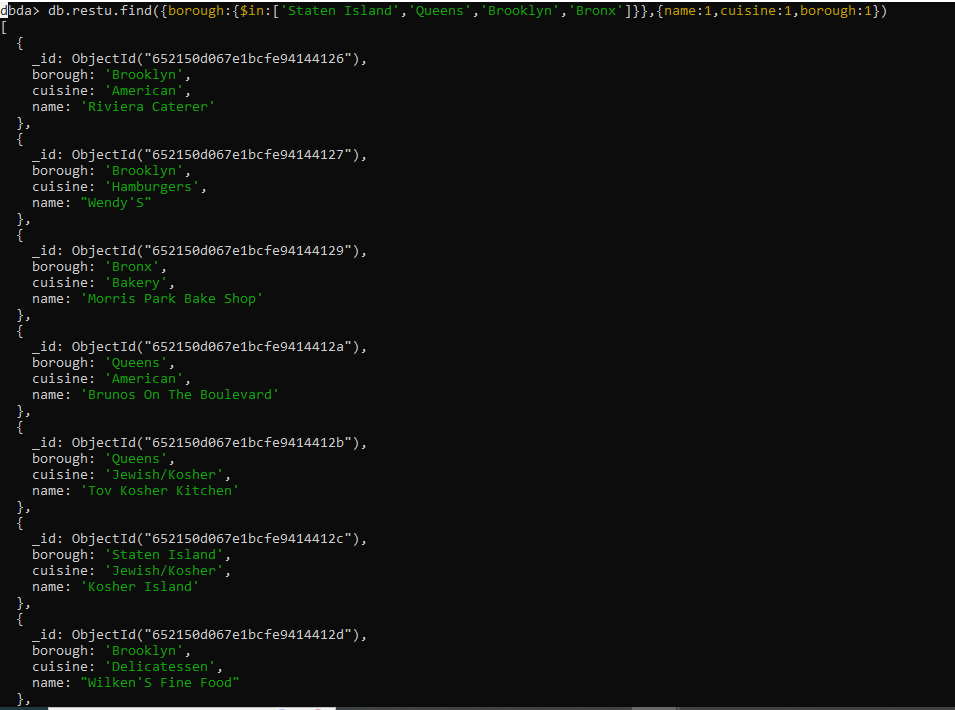
*17. Write a MongoDB query to find the restaurants which belong to the borough Bronx and*

*prepared either American or Chinese dish.*

**

*18. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those*

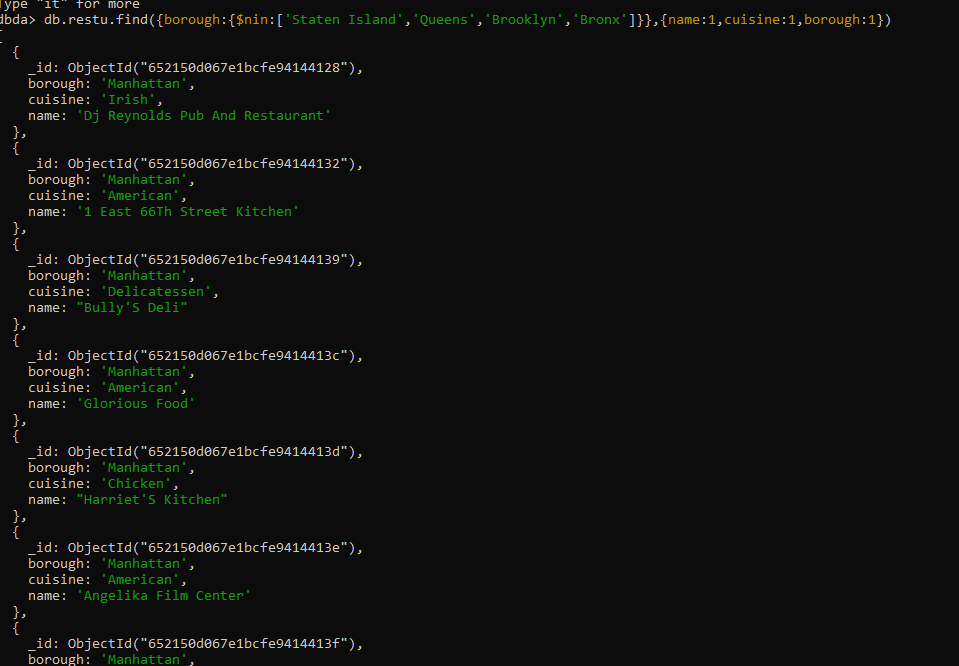
*restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn.*

**

*19. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those*

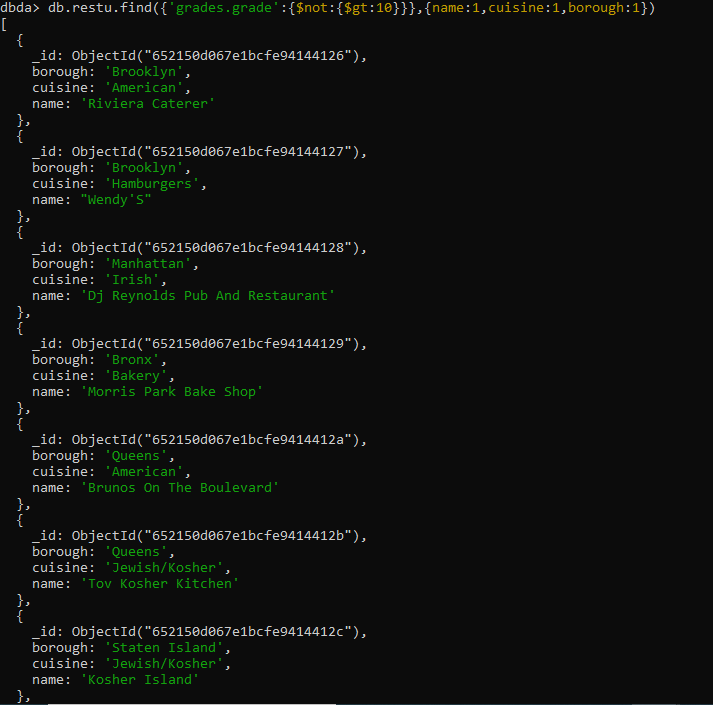
*restaurants which are not belonging to the borough Staten Island or Queens or Bronxor*

*Brooklyn.*

**

*20. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those*

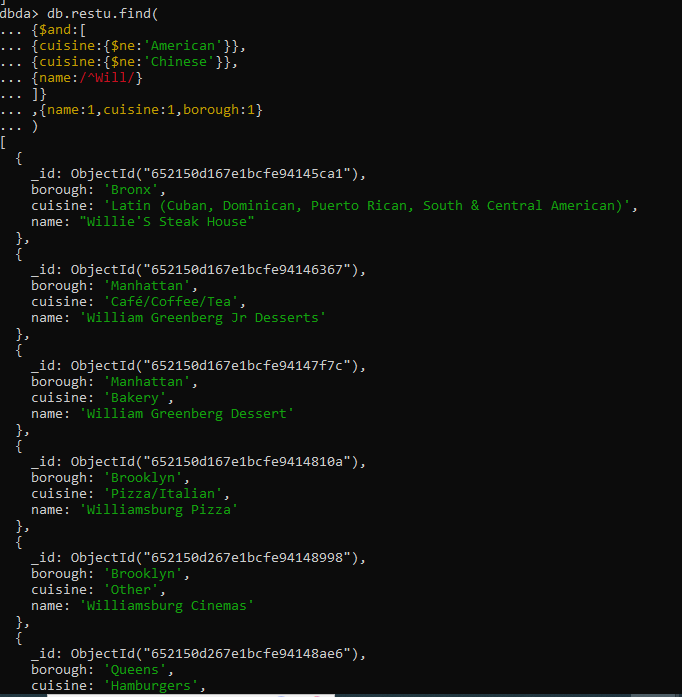
*restaurants which achieved a score which is not more than 10.*

**

*21. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those*

*restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins*

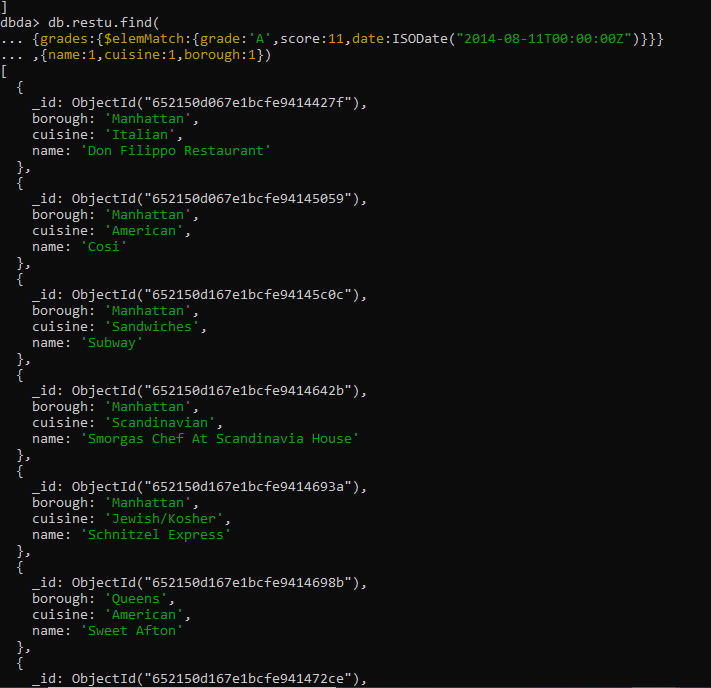
*with letter 'Wil'.*

**

*22. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants*

*which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z"*

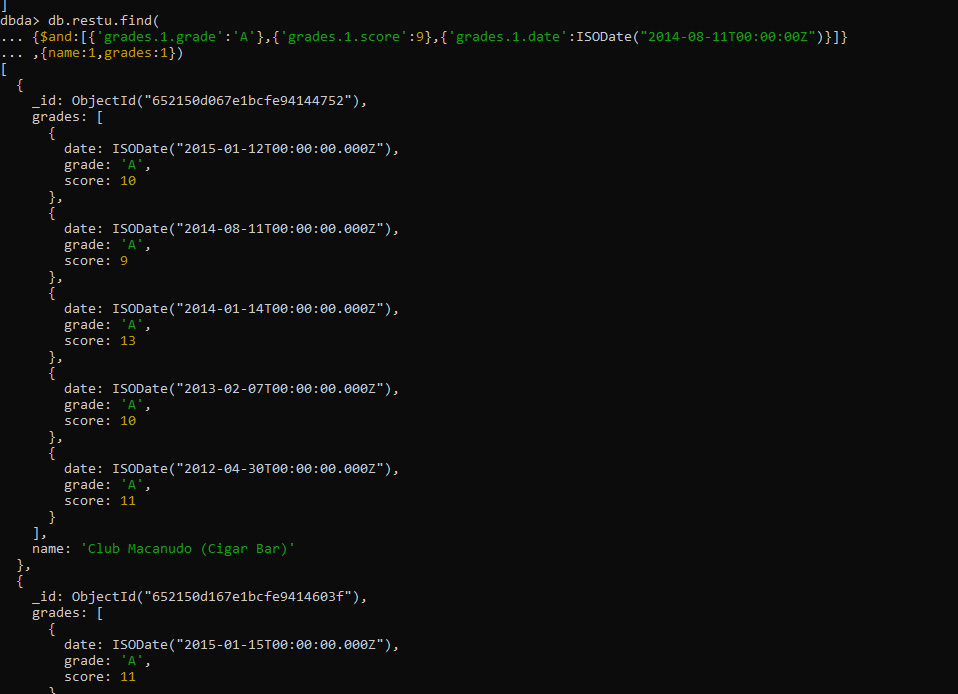
*among many of survey dates*

**

*23. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants*

*where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate*

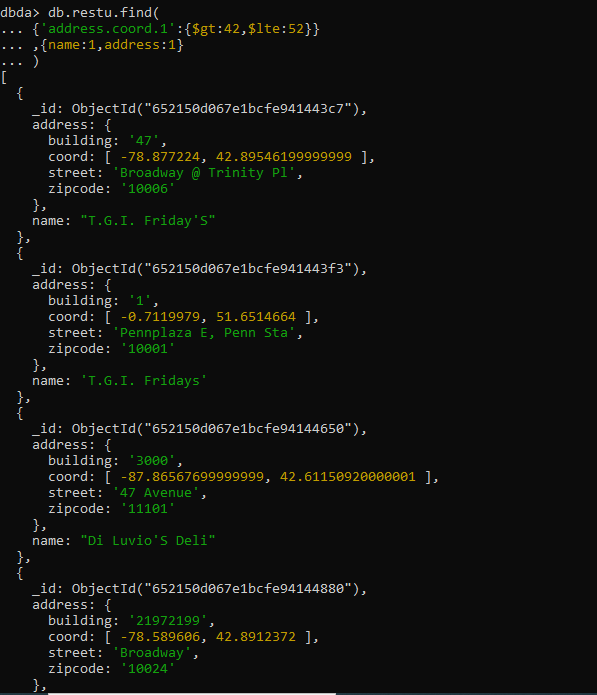
*"2014-08-11T00:00:00Z".*

**

*24. Write a MongoDB query to find the restaurant Id, name, address and geographical*

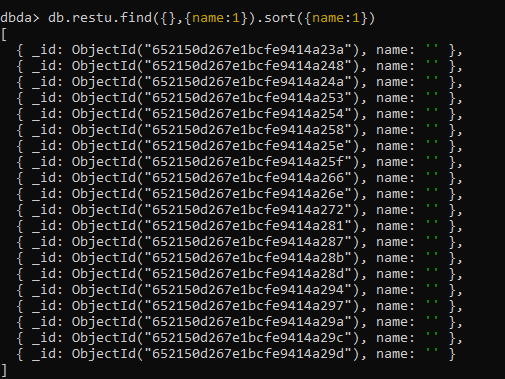
*location for those restaurants where 2nd element of coord array contains a value which is*

*more than 42 and upto 52*

**

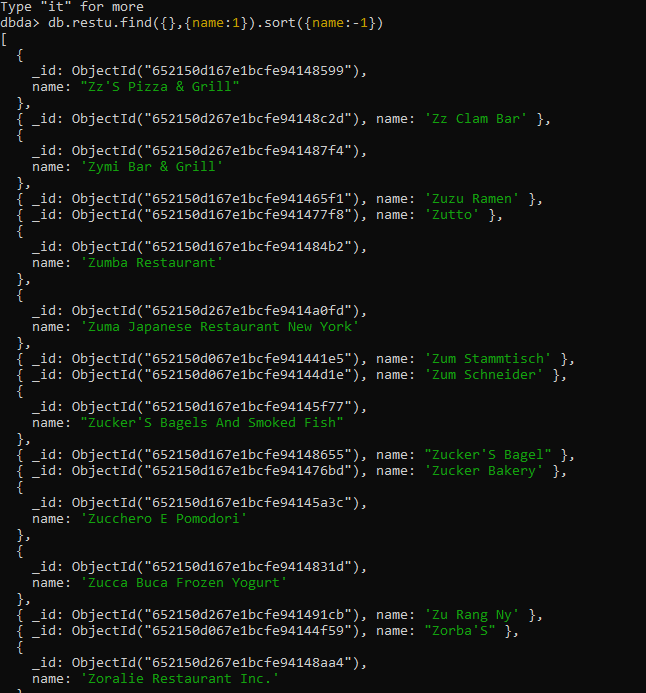
*25. Write a MongoDB query to arrange the name of the restaurants in ascending order along*

*with all the columns.*

**

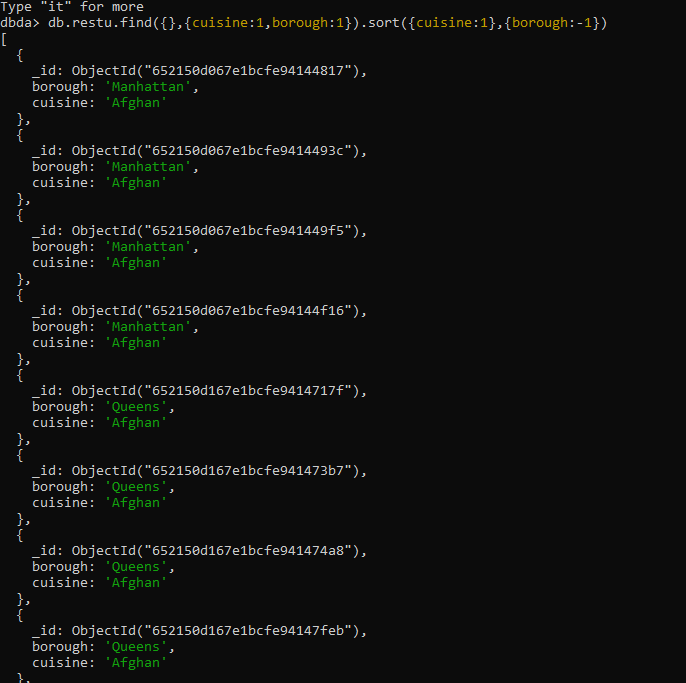
*26. Write a MongoDB query to arrange the name of the restaurants in descending along with*

*all the columns.*

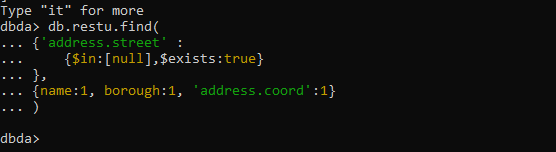
**

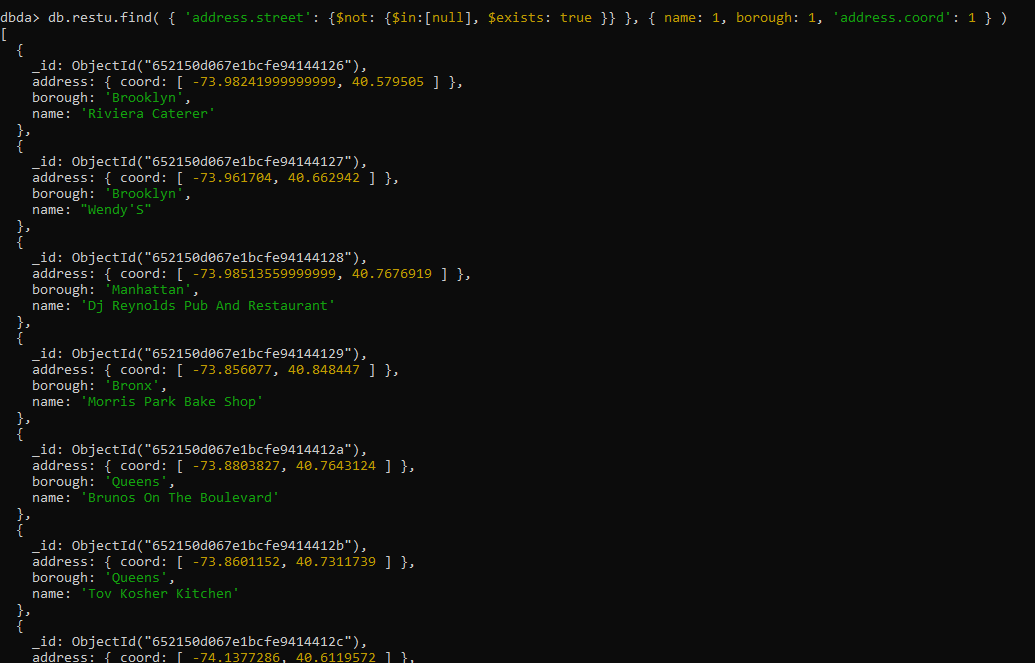
*27. Write a MongoDB query to arranged the name of the cuisine in ascending order and for*

*that same cuisine borough should be in descending order*

**

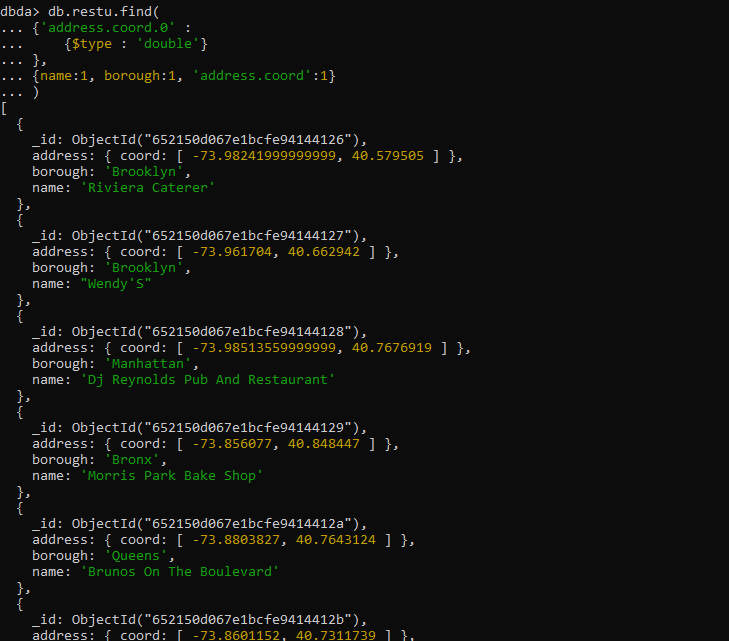
*28. Write a MongoDB query to know whether all the addresses contains the street or not.*

**

**

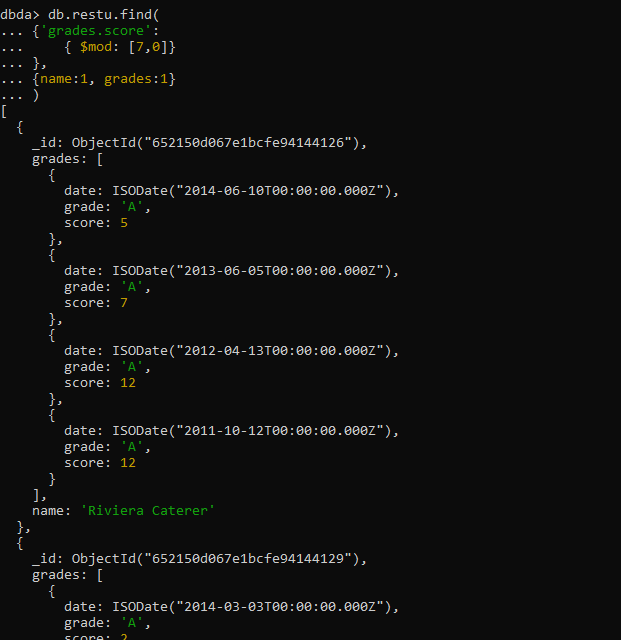
*29. Write a MongoDB query which will select all documents in the restaurants collection*

*where the coord field value is Double.*

**

*30. Write a MongoDB query which will select the restaurant Id, name and grades for those*

*restaurants which returns 0 as a remainder after dividing the score by 7.*

**

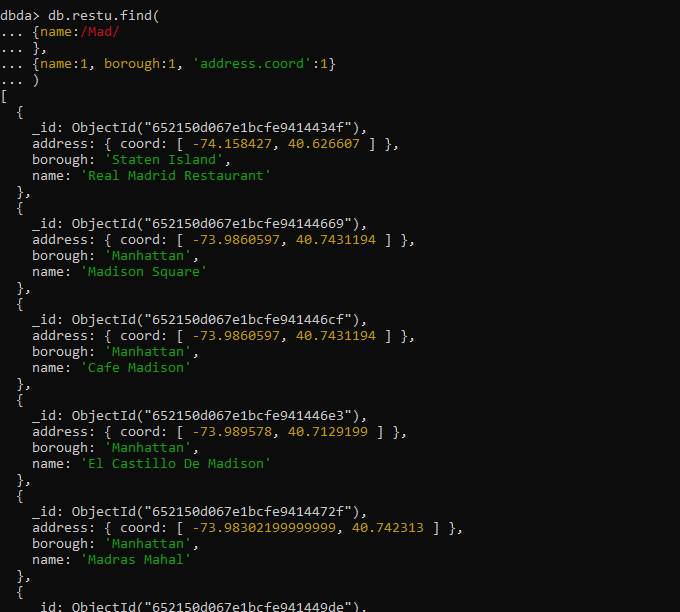
*31. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and*

*cuisine for those restaurants which contains 'mon' as three letters somewhere in its name.*

**

*32. Write a MongoDB query to find the restaurant name, borough, longitude and latitude and*

*cuisine for those restaurants which contain 'Mad' as first three letters of its name.*

**