# Lab 03 – SQL – Single Table Queries

## Objectives

The purpose of this lab is to start learning SQL by writing basic DML statements involving a single table. You will learn to create basic CRUD statements (queries as well as insert, update and delete).

### Submission:

***Your submission will be a single Word document with the question#, SQL query, solutions if it is a text or a screen shot provided. (with a .docx file extension)***. **Create a comment header that includes your name, student id, the date and the purpose of the file (i.e. DBS211 – Lab 03_firstname).**

### Setup

Create a new worksheet in SQL developer and add an appropriate comment header that includes your name, student id, the date and the purpose of the file (i.e. DBS211 – Lab 03). After every command and result in each question paste them in your Word file

Immediately under the comment header, enter the following line and then execute it:

```
SET AUTOCOMMIT ON;
```

### Style Guide

Your SQL should be written using the standard coding style:

- all keywords are to be upper case,
- all user-defined names are to be lower case, (example: table and field names)
- there should be a carriage return before each major part of the SQL statements (i.e. before SELECT, FROM, WHERE and ORDER BY)

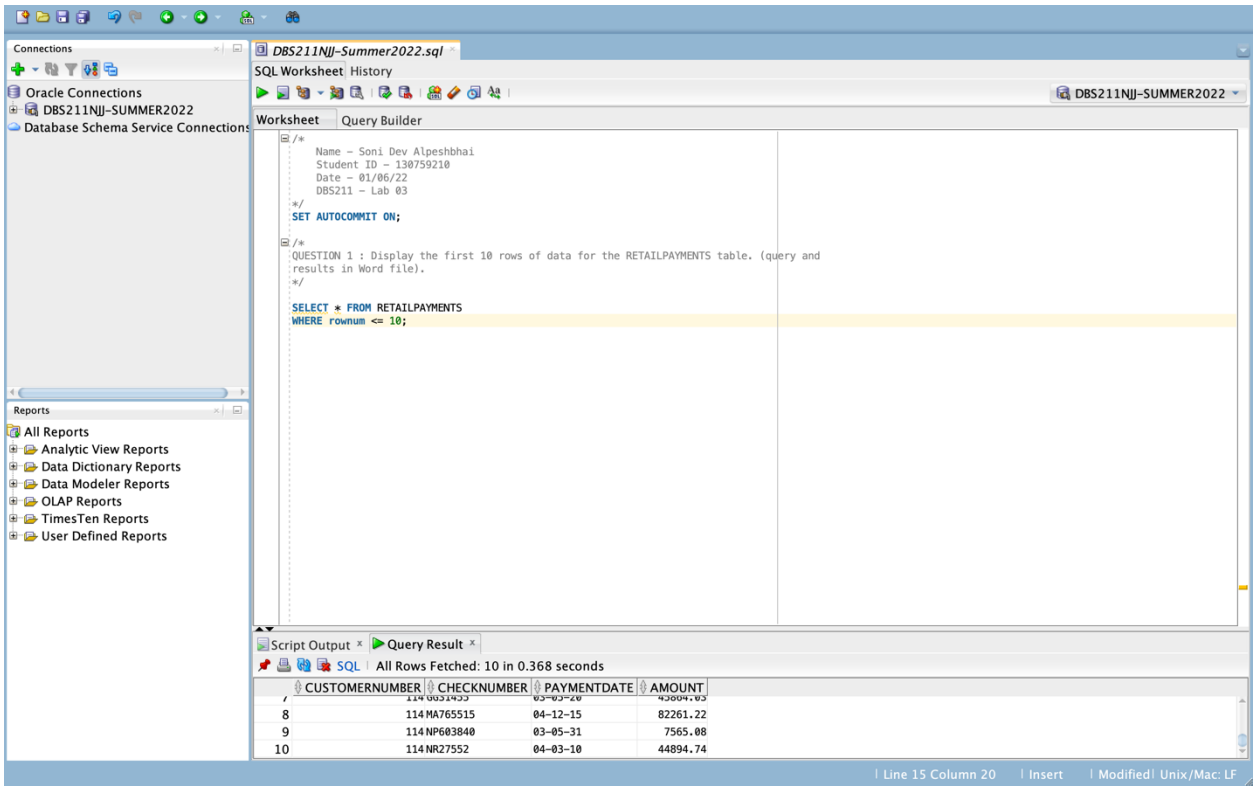
Using comments to number the question answers, write the SQL code to complete the following tasks.

## Tasks:

1. Display the first 10 rows of data for the RETAILPAYMENTS table. (query and results in Word file).

   QUERY :-

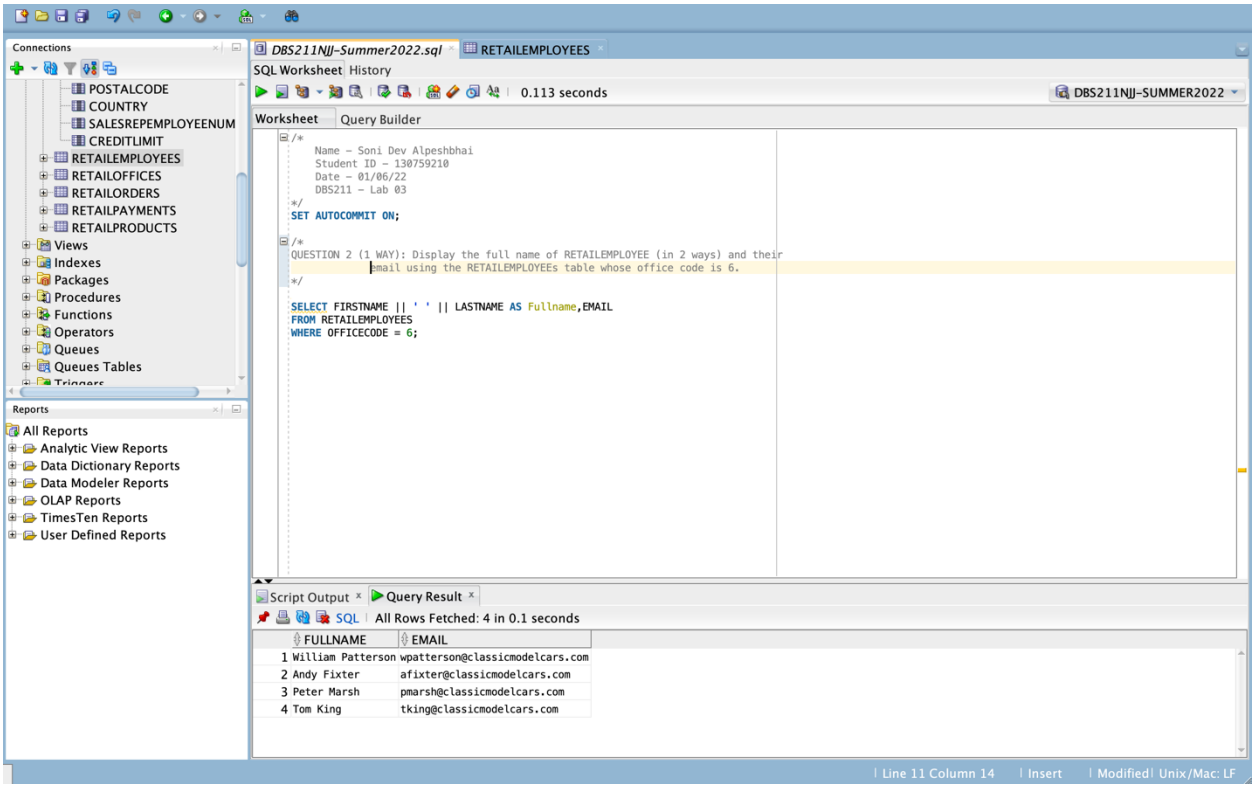
   ```
   SELECT * FROM RETAILPAYMENTS

   WHERE rownum <= 10;
   ```




2. Display the full name of RETAILEMPLOYEE (in 2 ways) and their email using the RETAILEMPLOYEEs table whose office code is 6.

1-WAY :

```
SELECT FIRSTNAME AS FIRST,LASTNAME AS LAST,EMAIL

FROM RETAILEMPLOYEES

WHERE OFFICECODE = 6;
```

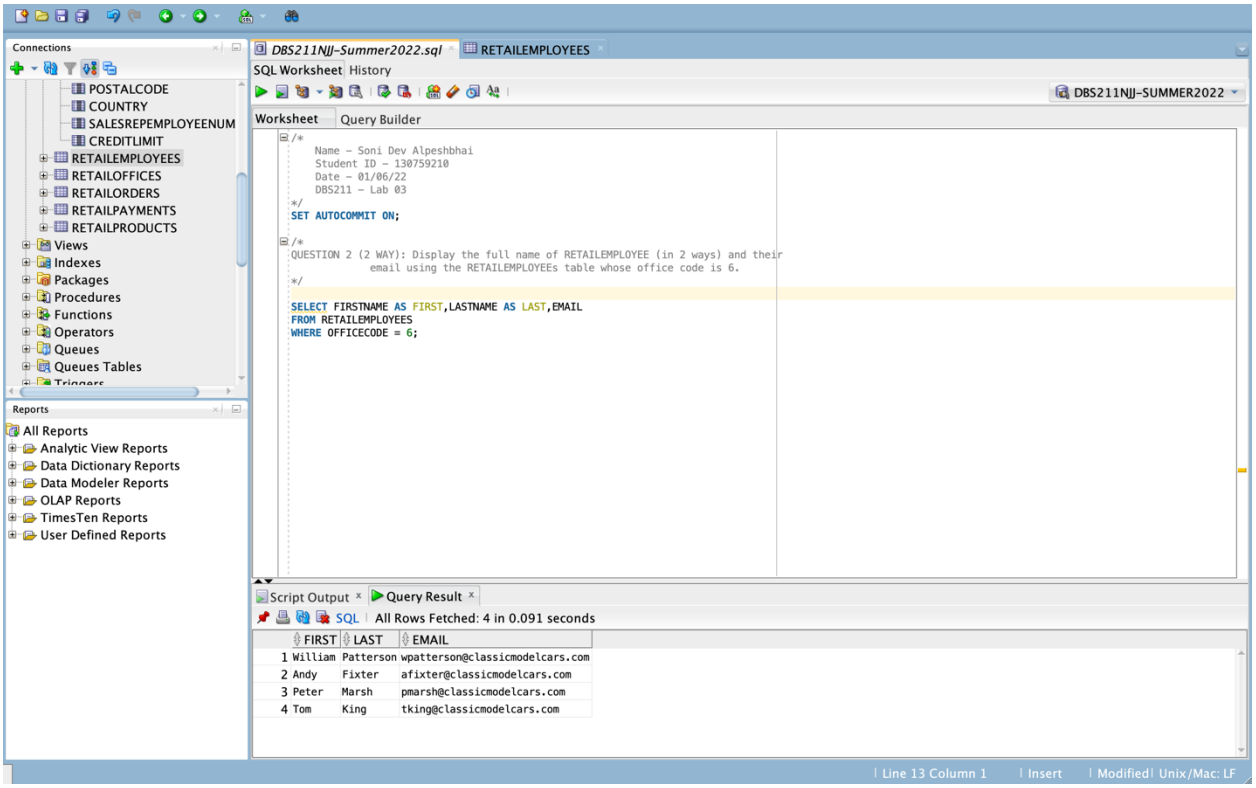



2-WAY :

```
SELECT FIRSTNAME || ‘ ‘ || LASTNAME AS FULLNAME,EMAIL
```

FROM RETAILEMPLOYEES

WHERE OFFICECODE = 6;

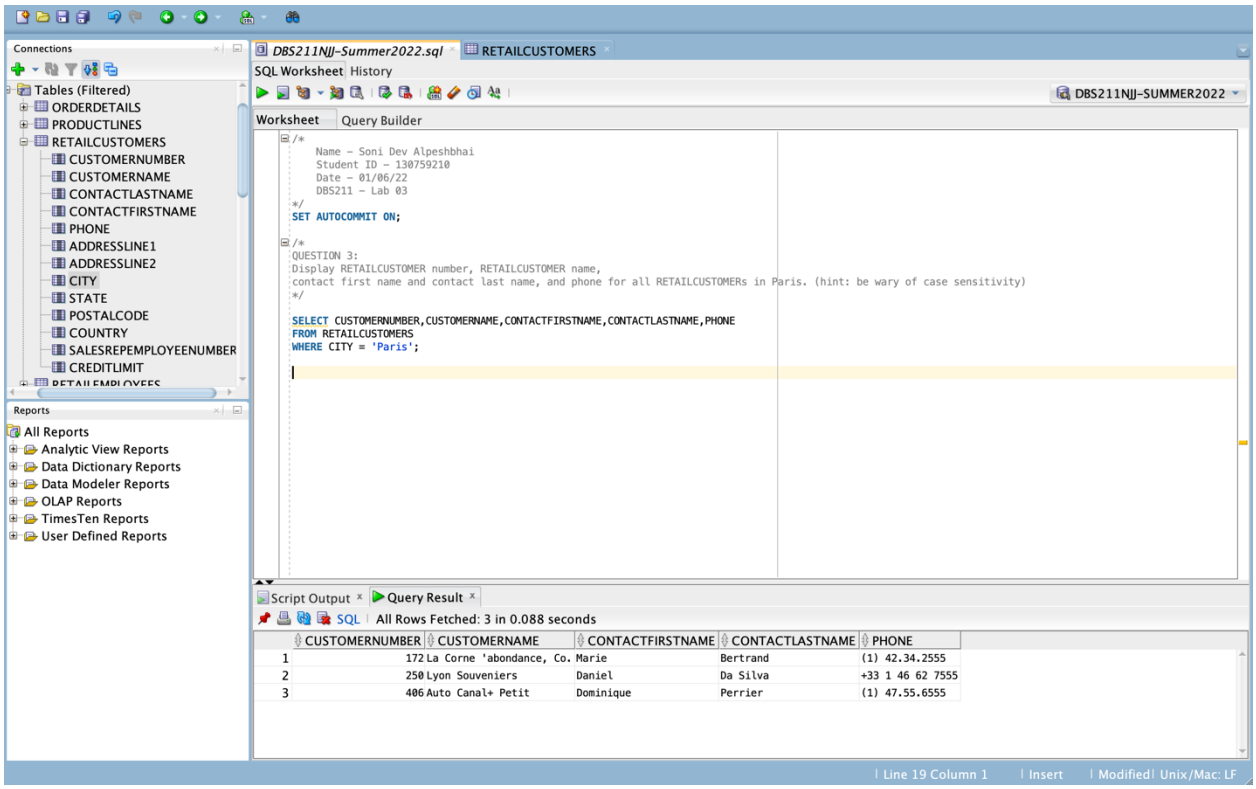



3. Display RETAILCUSTOMER number, RETAILCUSTOMER name, contact first name and contact last name, and phone for all RETAILCUSTOMERs in Paris. (**hint**: be wary of case sensitivity)

SELECT CUSTOMERNUMBER,CUSTOMERNAME,CONTACTFIRSTNAME,CONTACTLASTNAME,PHONE

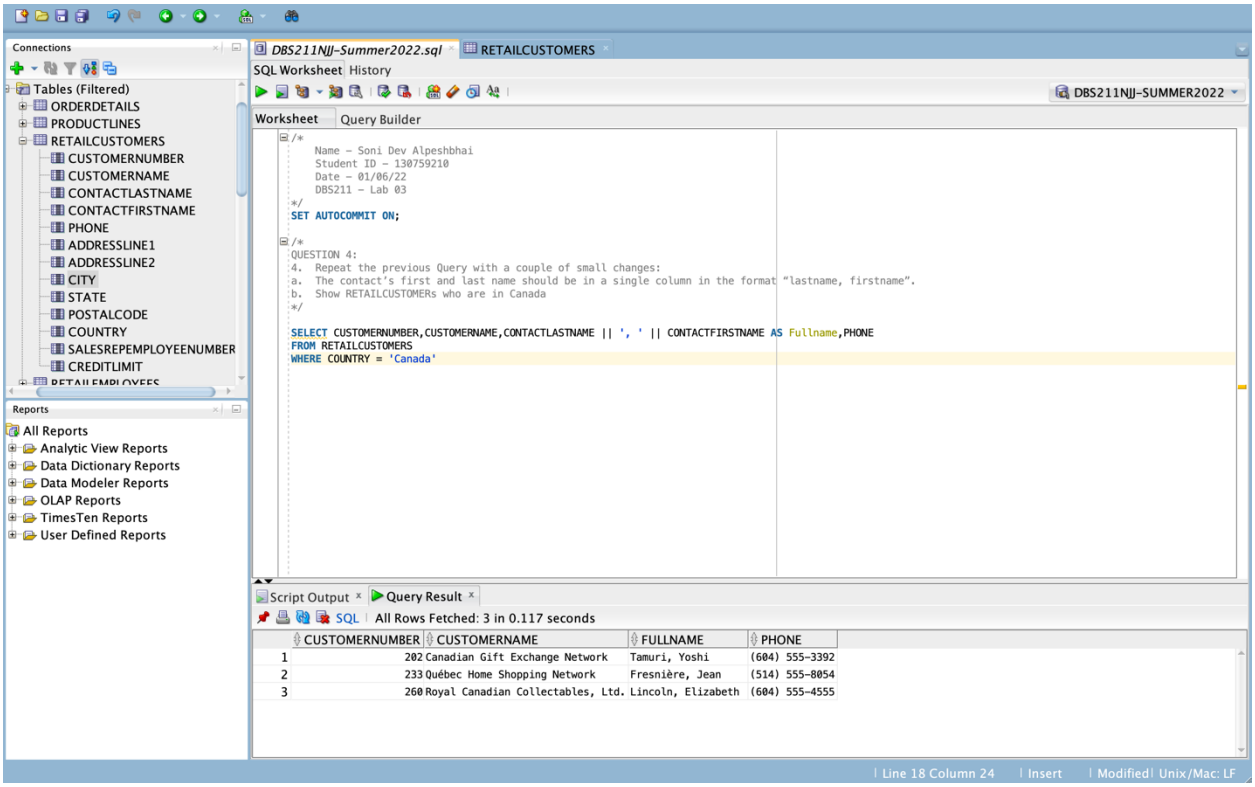
FROM RETAILCUSTOMERS

WHERE CITY = 'Paris'




4. Repeat the previous Query with a couple of small changes:
    a. The contact's first and last name should be in a single column in the format "lastname, firstname".
    b. Show RETAILCUSTOMERs who are in Canada

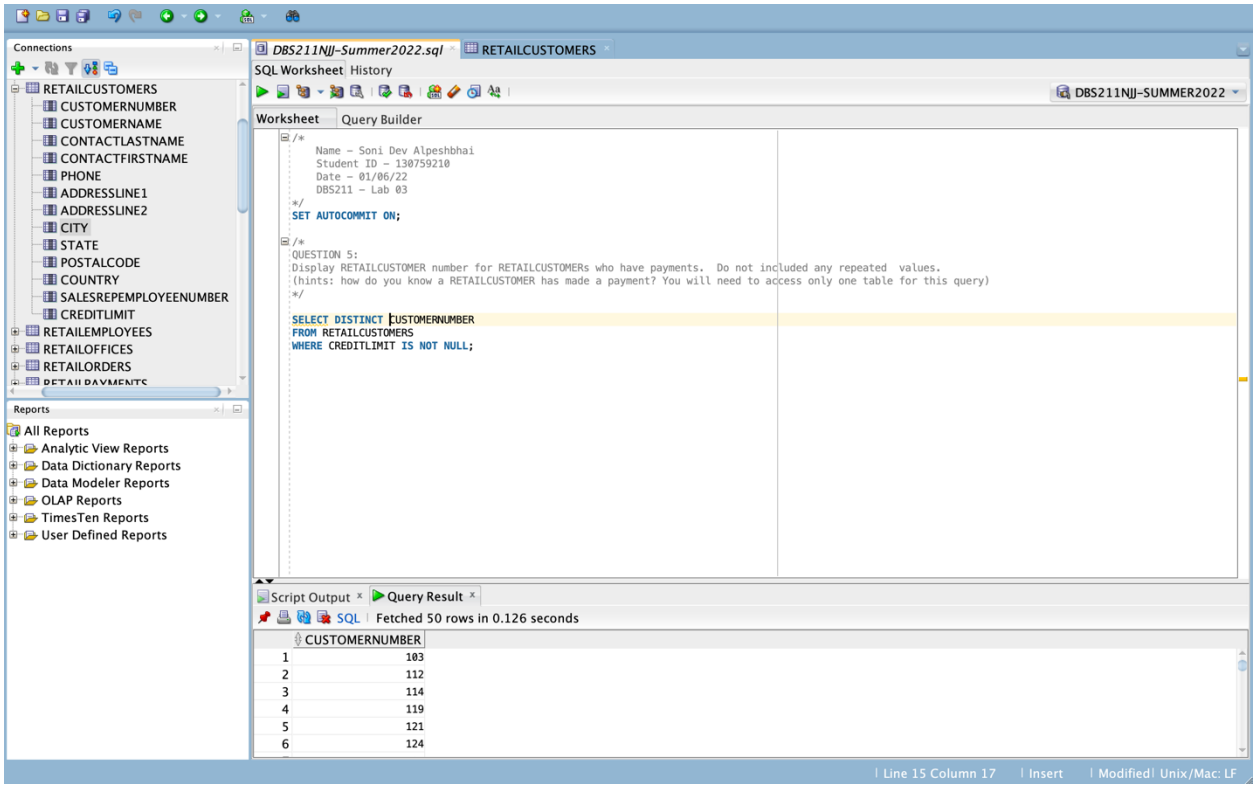
```
SELECT CUSTOMERNUMBER,CUSTOMERNAME, CONTACTLASTNAME || ‘,’ || CONTACTFIRSTNAME
AS Fullname, PHONE

FROM RETAILCUSTOMERS

        WHERE COUNTRY = 'Canada’;
```




5. Display RETAILCUSTOMER number for RETAILCUSTOMERs who have payments. Do not included any repeated values. (**hints**: how do you know a RETAILCUSTOMER has made a payment? You will need to access only one

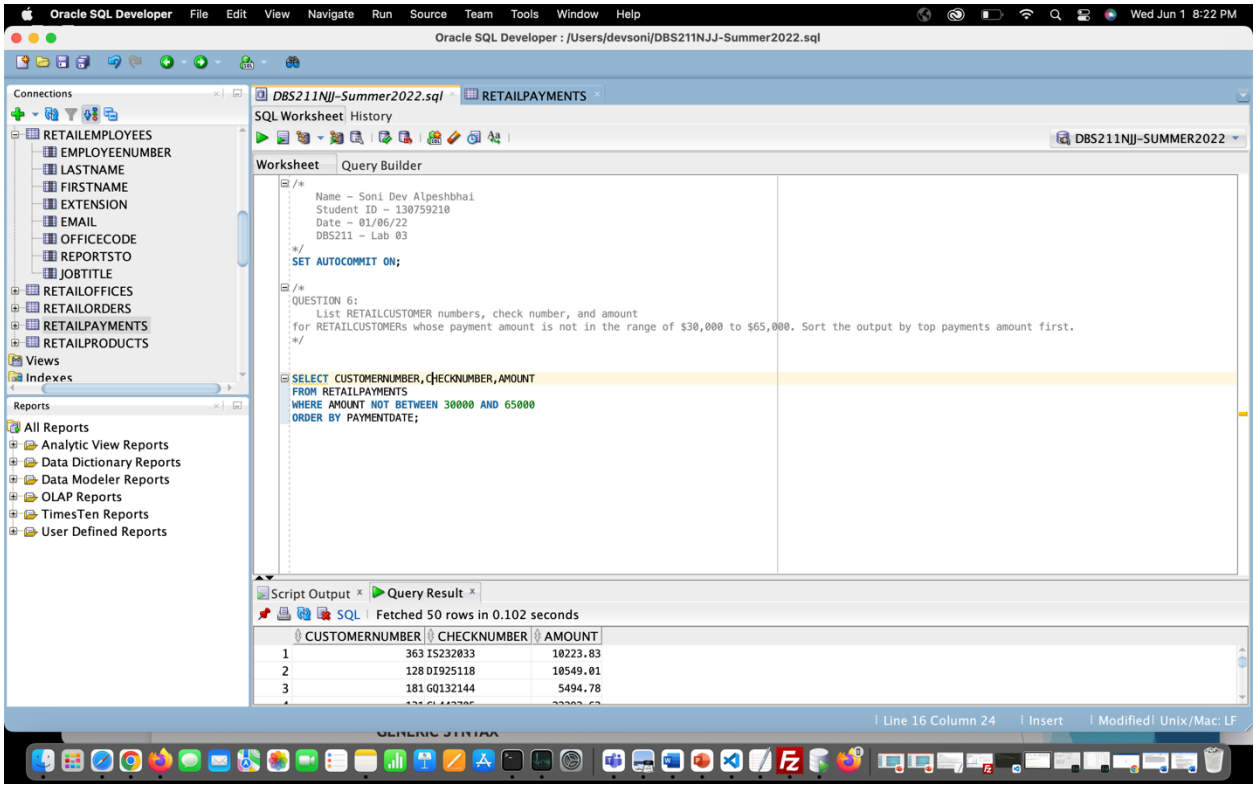
table for this query)

```
SELECT DISTINCT CUSTOMERNUMBER

FROM RETAILCUSTOMERS

WHERE CREDITLIMIT IS NOT NULL;
```



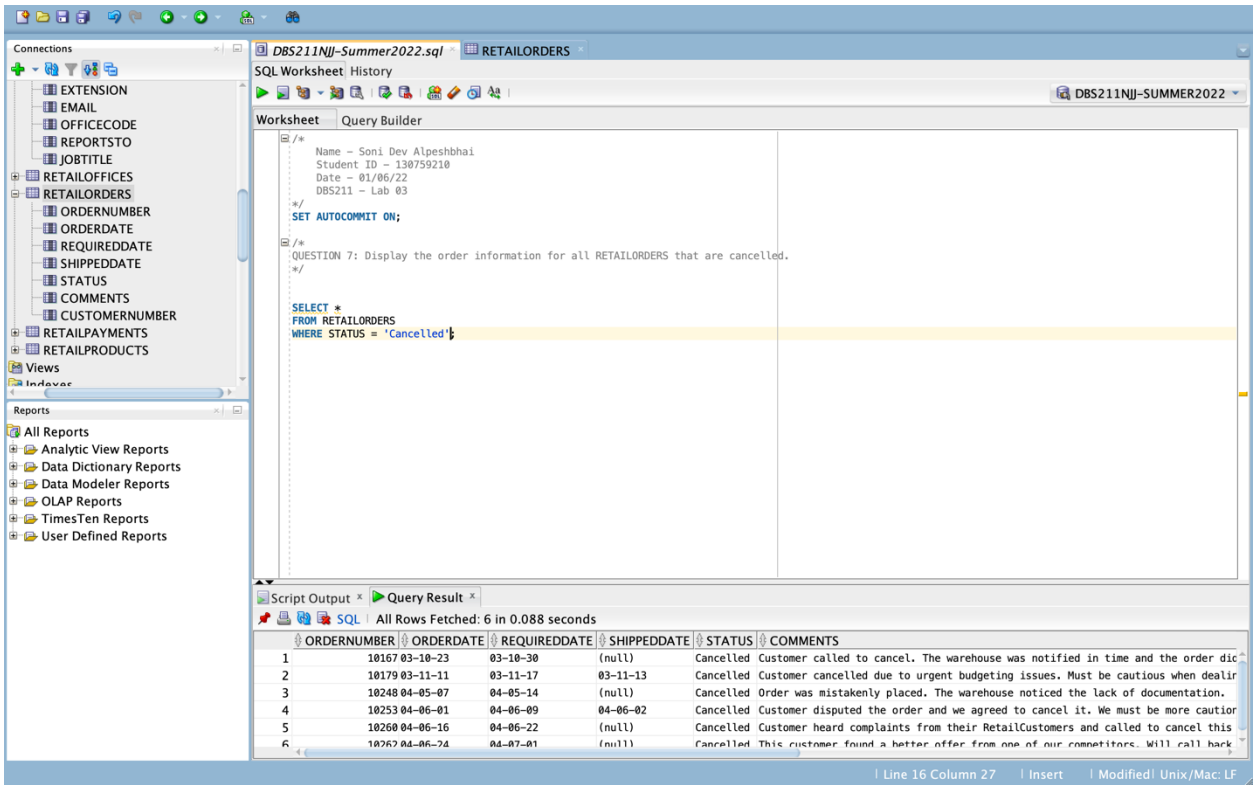

6. List RETAILCUSTOMER numbers, check number, and amount for RETAILCUSTOMERs whose payment amount is not in the range of $30,000 to $65,000. Sort the output by top payments amount first.

```
SELECT CUSTOMERNUMBER,CHECKNUMBER,AMOUNT
FROM RETAILPAYMENTS
WHERE AMOUNT NOT BETWEEN 30000 AND 65000
ORDER BY PAYMENTDATE;
```



7. Display the order information for all RETAILORDERS that are cancelled.

```
SELECT *

FROM RETAILORDERS

WHERE STATUS = 'Cancelled';
```

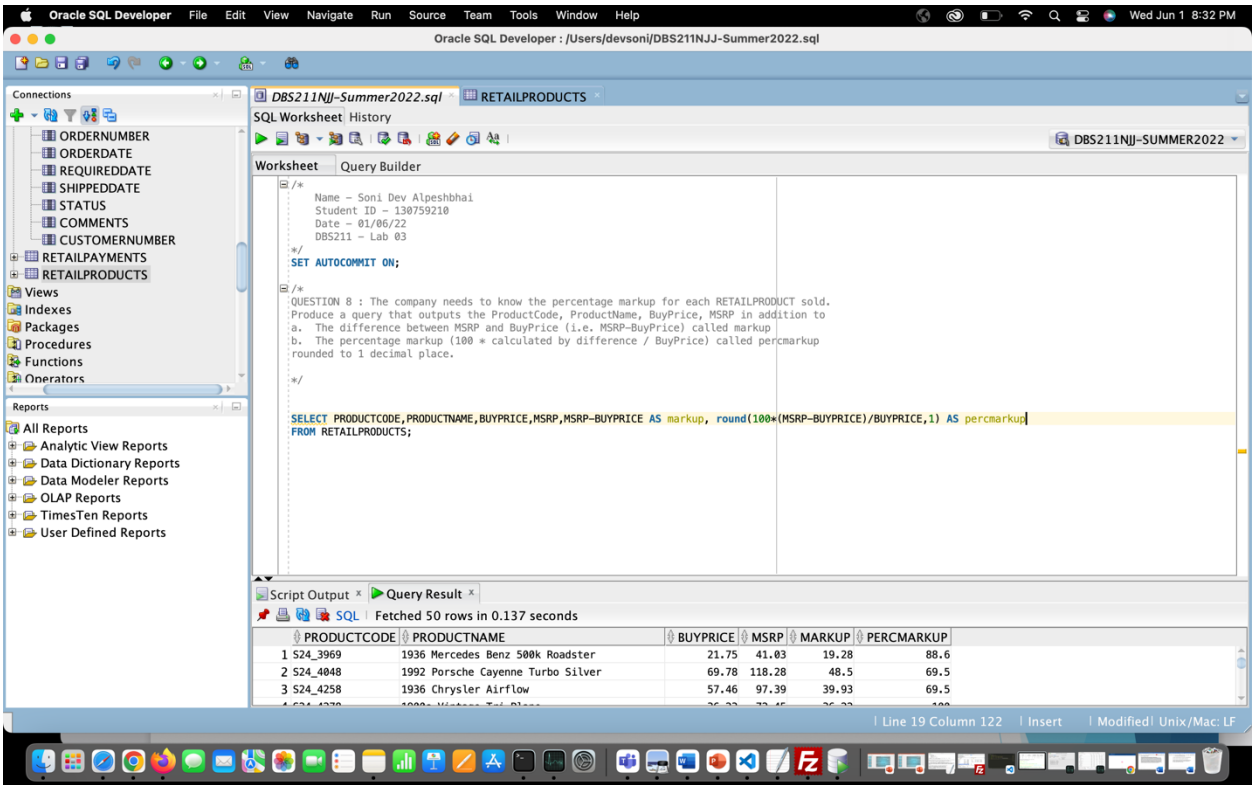


8. The company needs to know the percentage markup for each RETAILPRODUCT sold. Produce a query that outputs the ProductCode, ProductName, BuyPrice, MSRP in addition to

a. The difference between MSRP and BuyPrice (i.e. MSRP-BuyPrice) called *markup*

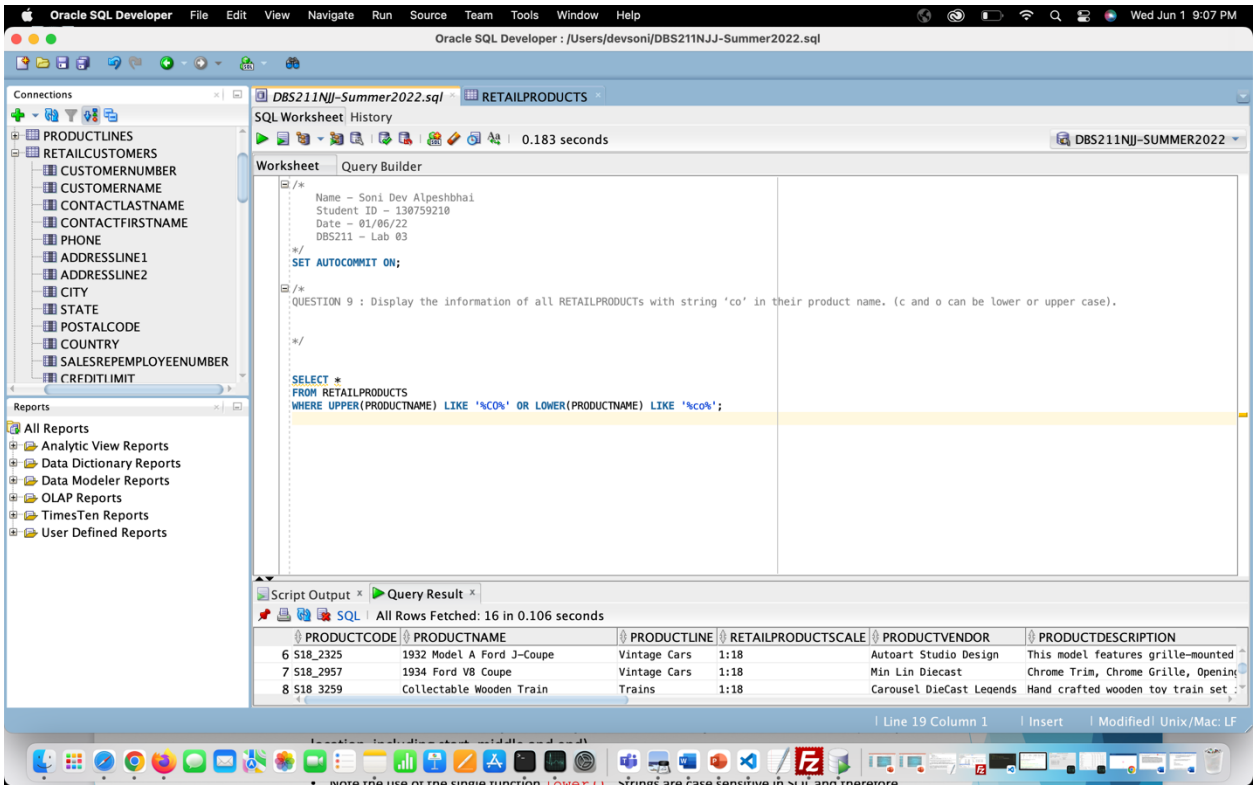
b. The percentage markup (100 * calculated by difference / BuyPrice) called *percmarkup*
rounded to 1 decimal place.

```
SELECT PRODUCTCODE,PRODUCTNAME,BUYPRICE,MSRP,MSRP-BUYPRICE AS
markup, round(100*(MSRP-BUYPRICE)/BUYPRICE,1) AS percmarkup

FROM RETAILPRODUCTS;
```



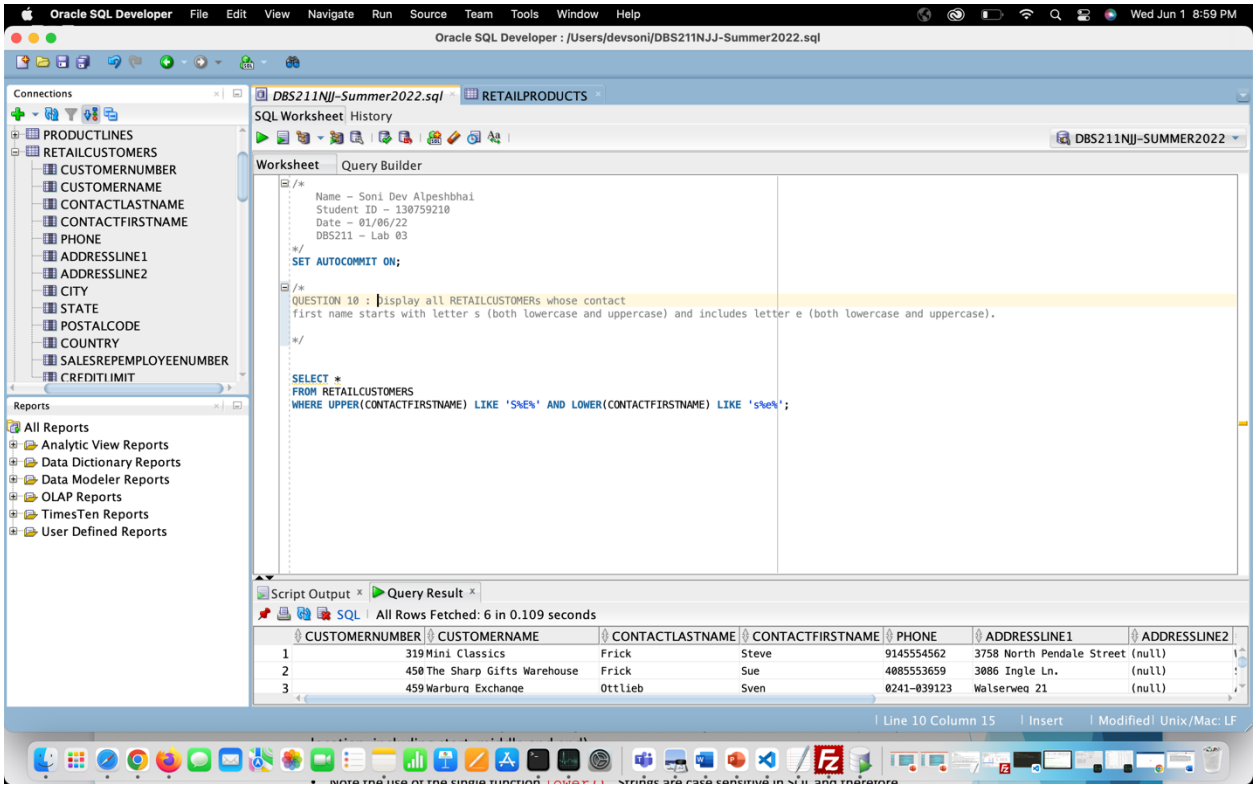

9. Display the information of all RETAILPRODUCTs with string ***'co'*** in their product name. (c and o can be lower or upper case).

```
SELECT *

FROM RETAILPRODUCTS

WHERE UPPER(PRODUCTNAME) LIKE '%CO%' OR LOWER(PRODUCTNAME) LIKE '%co%';
```




10.Display all RETAILCUSTOMERs whose contact first name starts with letter ***s*** (both lowercase and uppercase) and includes letter ***e*** (both lowercase and uppercase).

```
SELECT *

FROM RETAILCUSTOMERS

WHERE UPPER(CONTACTFIRSTNAME) LIKE 'S%E%' AND
LOWER(CONTACTFIRSTNAME) LIKE 's%e%';
```




11.Create a statement that will insert yourself as an RETAILEMPLOYEE of the company.

a. Use a unique RETAILEMPLOYEE number of your choice

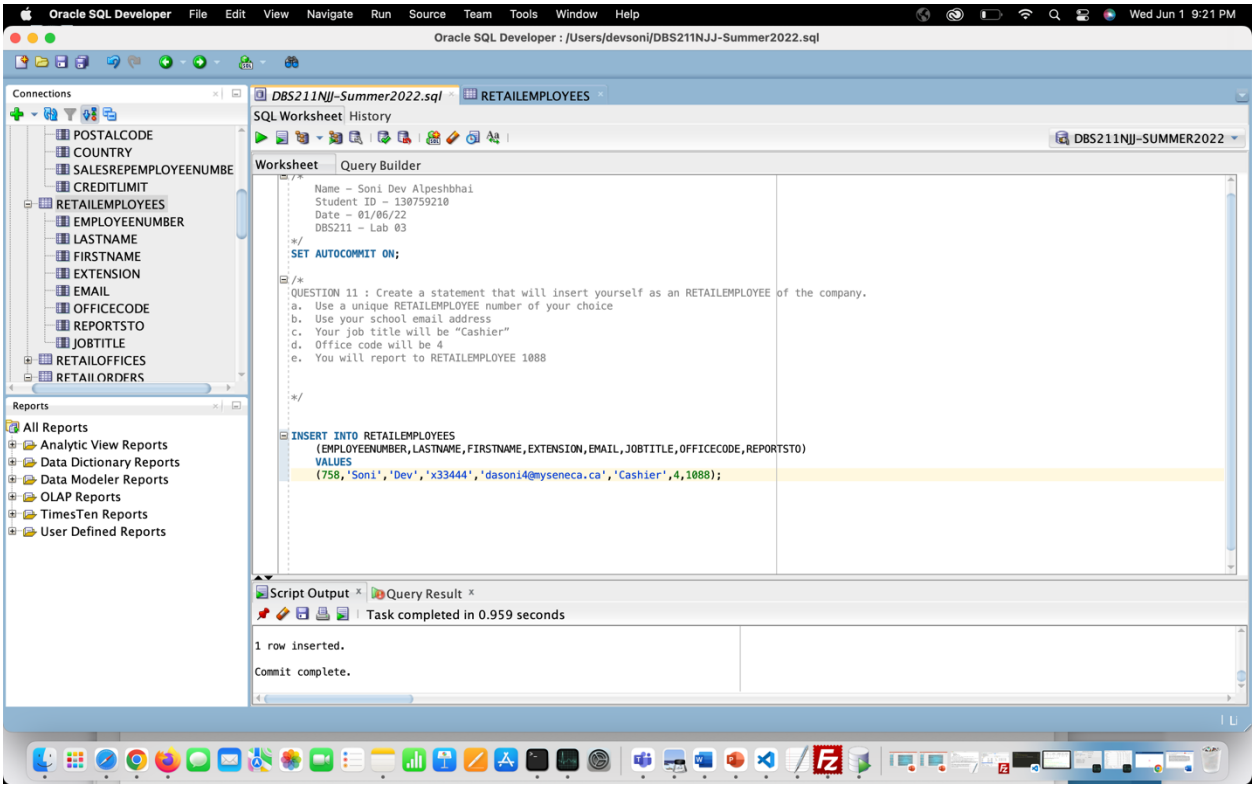
b. Use your school email address
c. Your job title will be “Cashier”
d. Office code will be 4
e. You will report to RETAILEMPLOYEE 1088

INSERT INTO RETAILEMPLOYEES

(EMPLOYEENUMBER,LASTNAME,FIRSTNAME,EXTENSION,EMAIL,JOBTITLE,OFFICECODE,REPORTSTO)

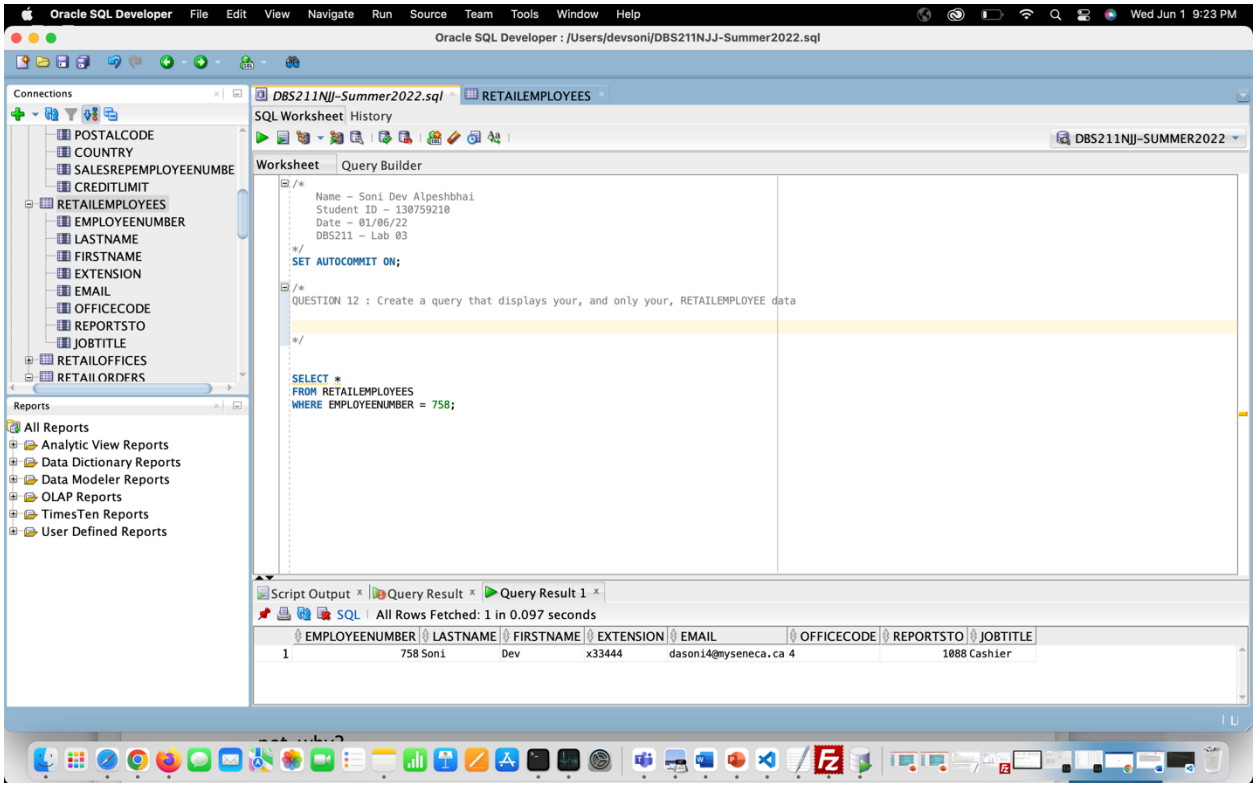
VALUES

(758,'Soni','Dev','x33444','dasoni4@myseneca.ca','Cashier',4,1088);



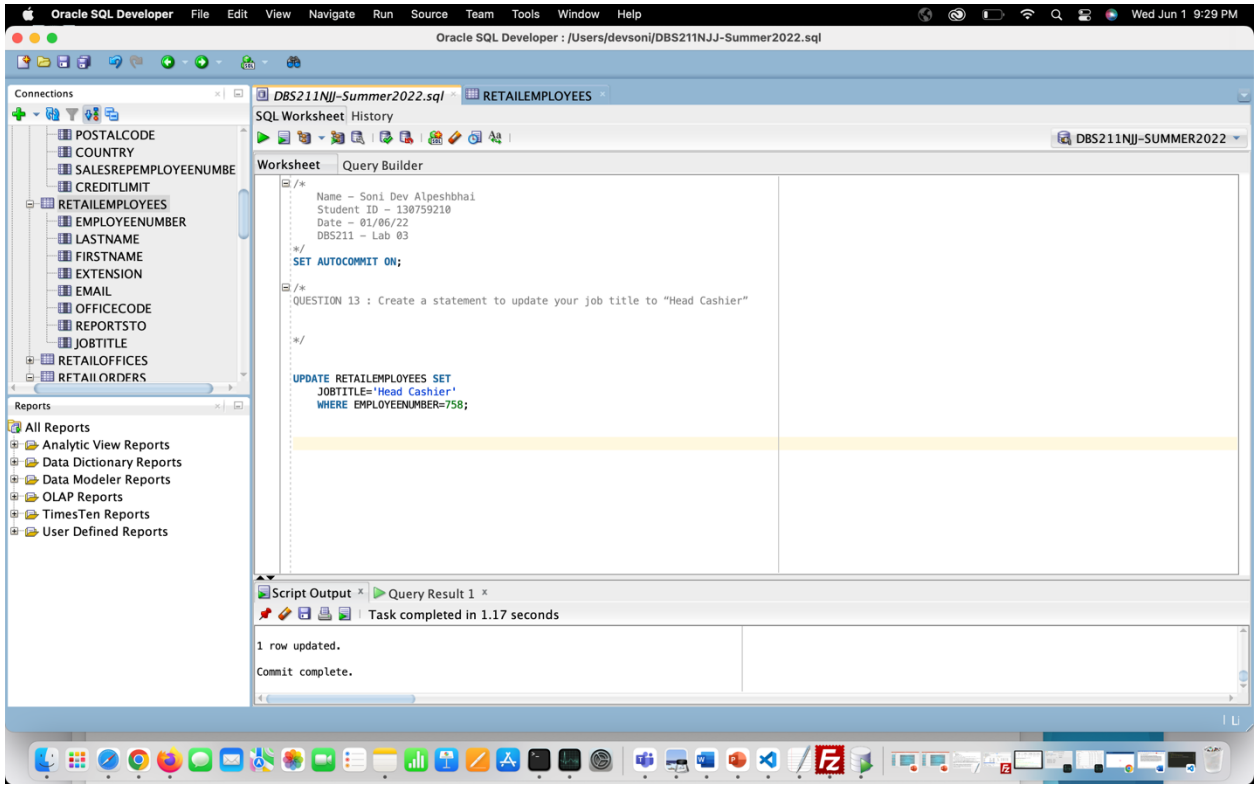
12.Create a query that displays your, and only your, RETAILEMPLOYEE data

```
SELECT *

FROM RETAILEMPLOYEES

WHERE EMPLOYEENUMBER = 758;
```



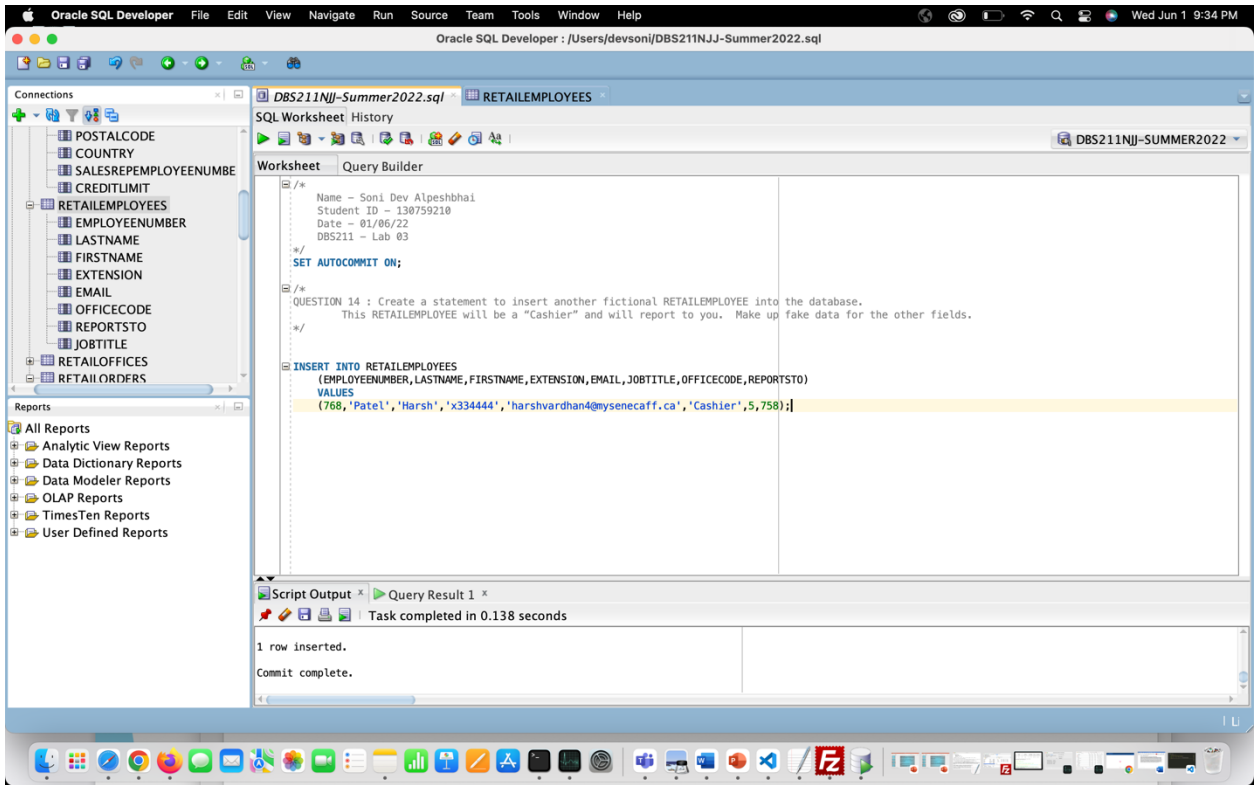
13.Create a statement to update your job title to “Head Cashier”

```
UPDATE RETAILEMPLOYEES SET
   JOBTITLE='Head Cashier'
   WHERE EMPLOYEENUMBER=758;
```



14.Create a statement to insert another fictional RETAILEMPLOYEE into the database. This RETAILEMPLOYEE will be a “Cashier” and will report to you. Make up fake data for the other fields.

```
INSERT INTO RETAILEMPLOYEES

(EMPLOYEENUMBER,LASTNAME,FIRSTNAME,EXTENSION,EMAIL,JOBTITLE,OFFICE
CODE,REPORTSTO)

  VALUES

  (768,'Patel','Harsh','x334444','harshvardhan4@mysenecaff.ca','Cashier',5,758);
```



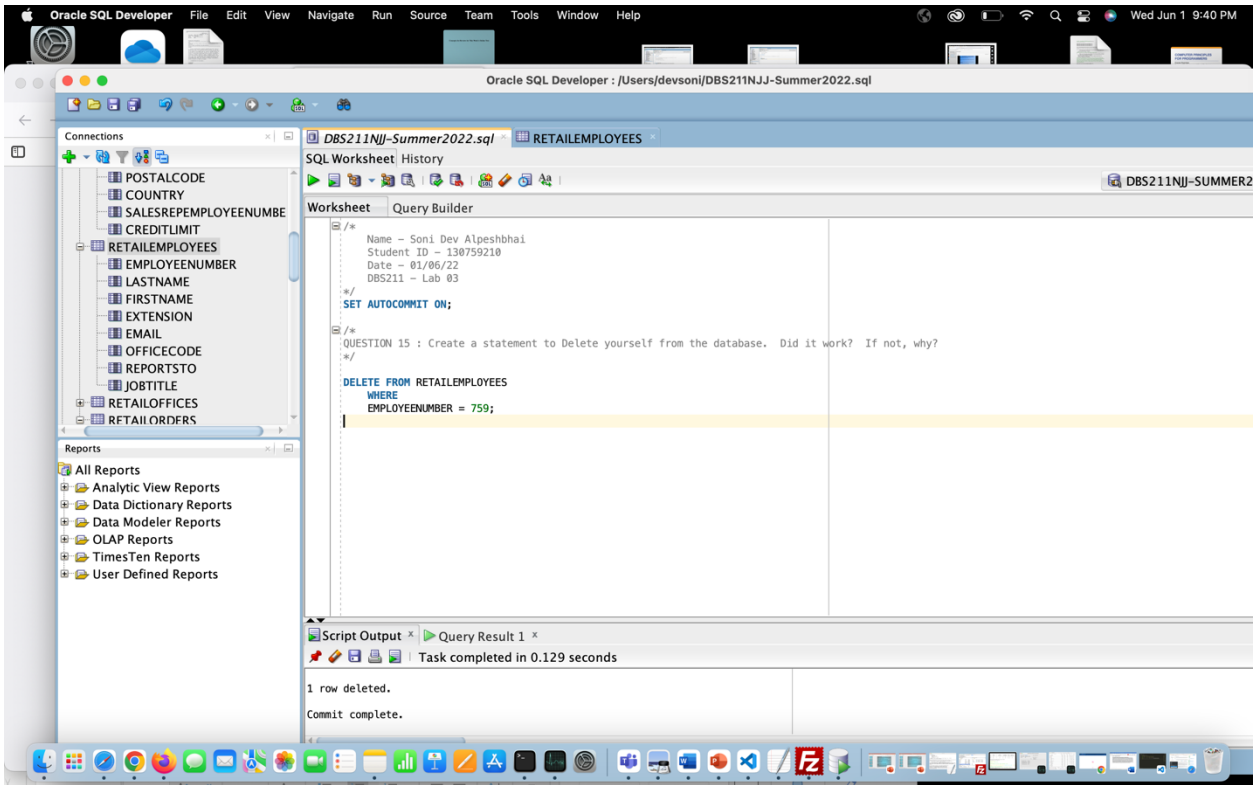

15.Create a statement to Delete yourself from the database. Did it work? If not, why?

DELETE FROM RETAILEMPLOYEES

WHERE

EMPLOYEENUMBER = 759;

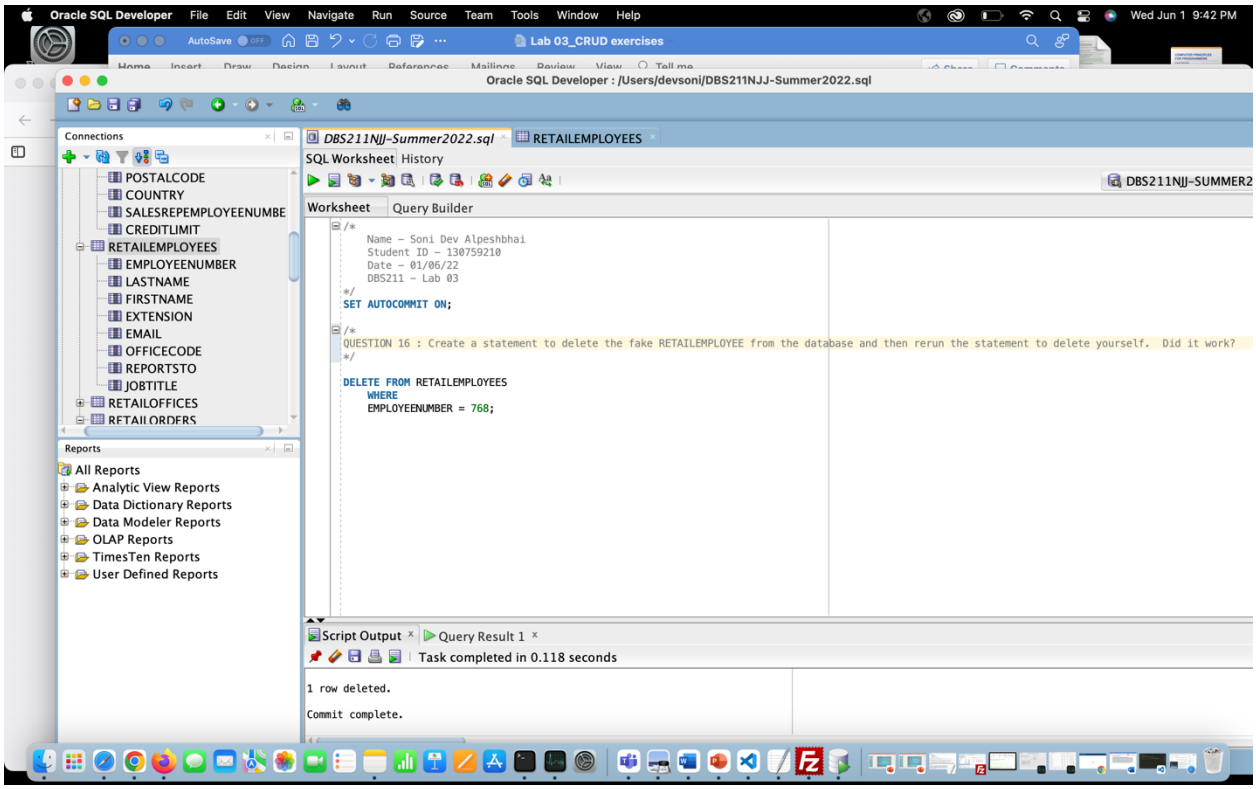
YES, the query executed successfully.




16.Create a statement to delete the fake RETAILEMPLOYEE from the database and then rerun the statement to delete yourself. Did it work?

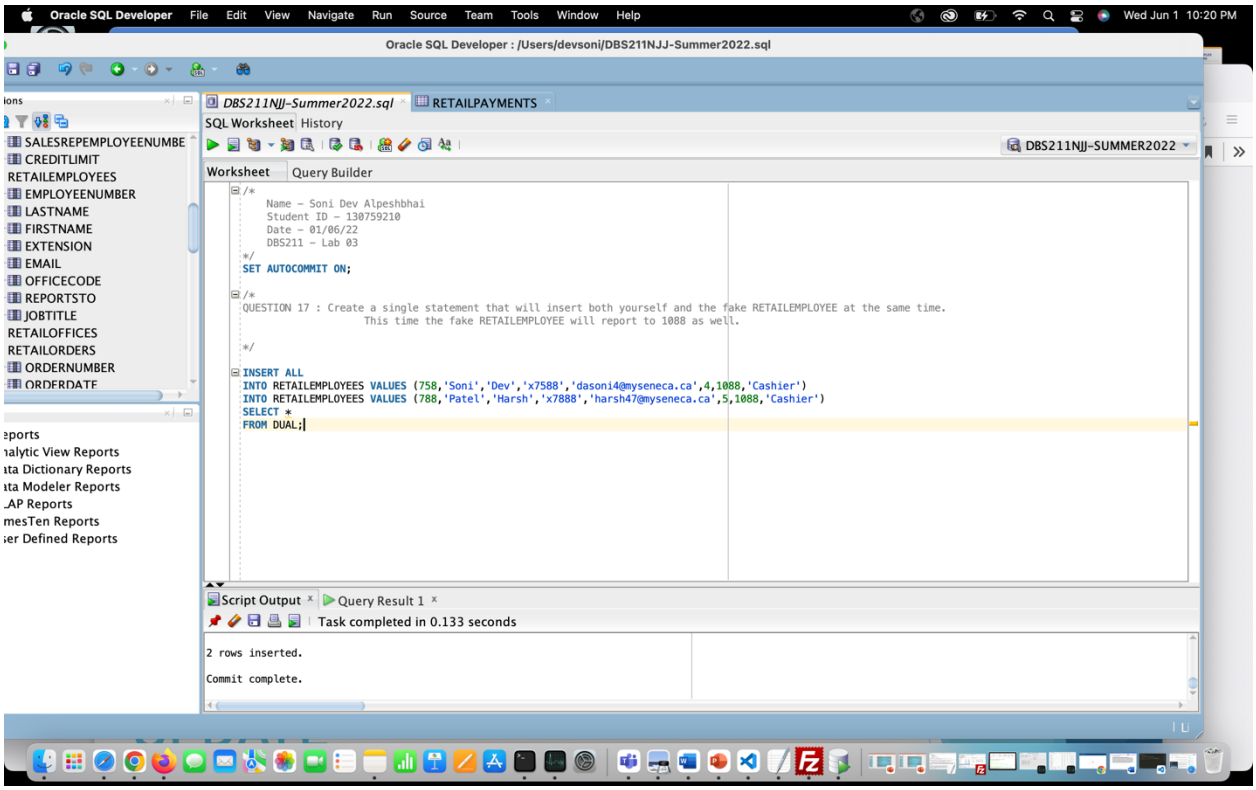
```
DELETE FROM RETAILEMPLOYEES
    WHERE
    EMPLOYEENUMBER = 768;
```

YES, the query executed successfully.



17.Create a **single** statement that will insert both yourself and the fake RETAILEMPLOYEE at the same time. This time the fake RETAILEMPLOYEE will report to 1088 as well.

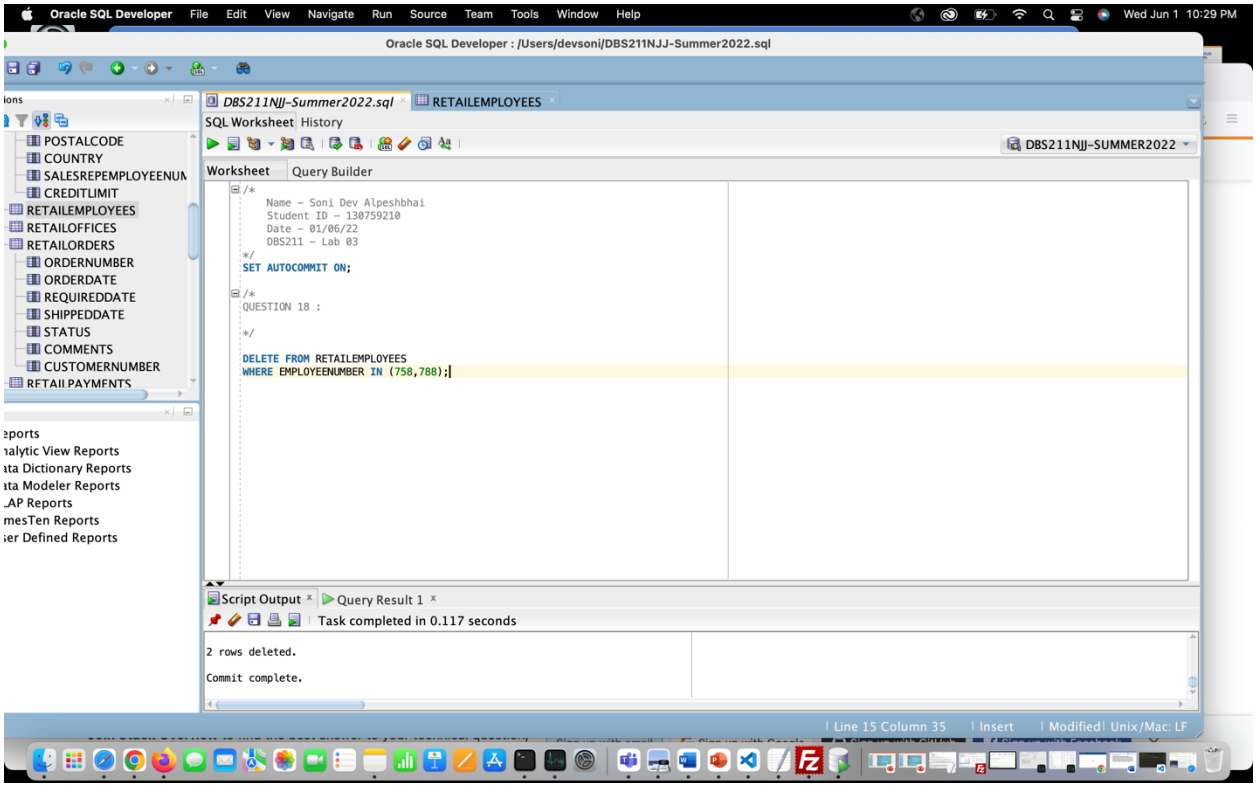
```
INSERT ALL

INTO RETAILEMPLOYEES VALUES
(758,'Soni','Dev','x7588','dasoni4@myseneca.ca',4,1088,'Cashier')

INTO RETAILEMPLOYEES VALUES
(788,'Patel','Harsh','x7888','harsh47@myseneca.ca',5,1088,'Cashier')

SELECT *

FROM DUAL;
```



18.Create a **single** statement to delete both yourself and the fake RETAILEMPLOYEE.

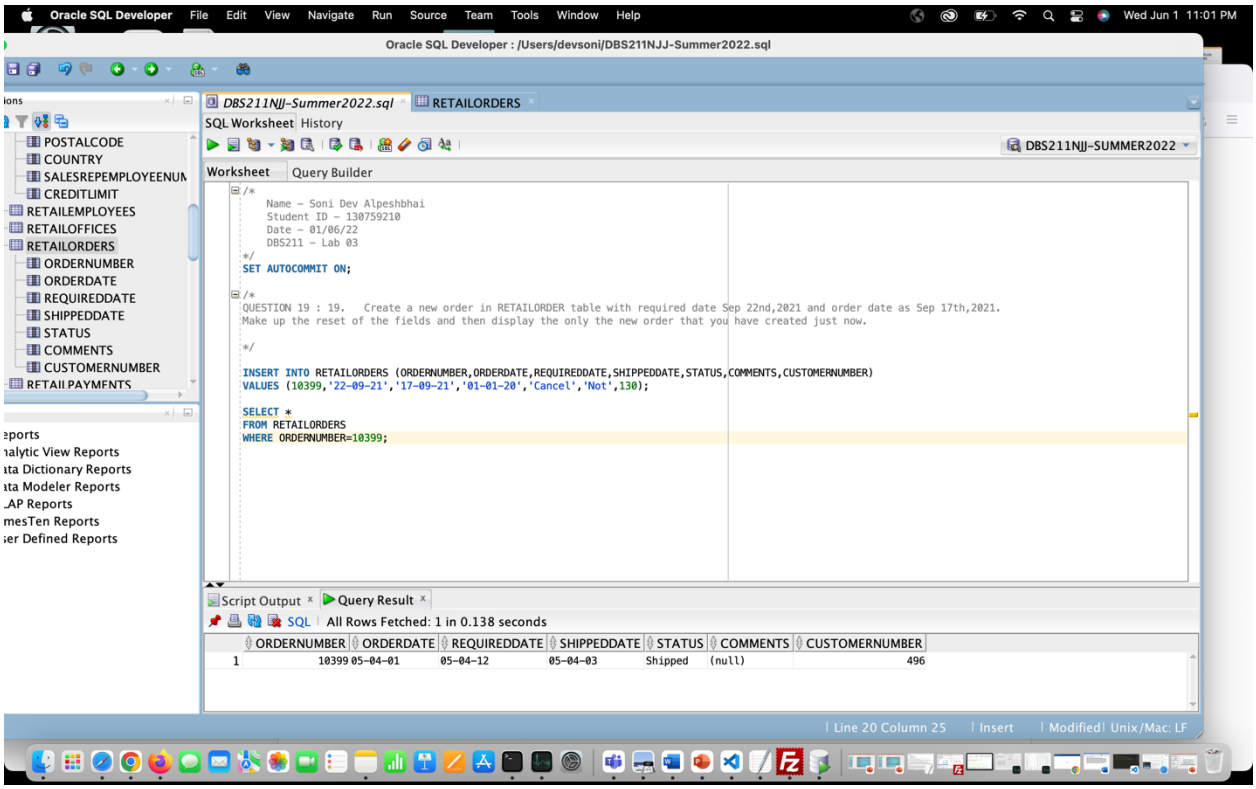
DELETE FROM RETAILEMPLOYEES

WHERE EMPLOYEENUMBER IN (758,788);



19. Create a new order in RETAILORDER table with required date Sep 22nd,2021 and order date as Sep 17th,2021. Make up the reset of the fields and then display the only the new order that you have created just now.

```
INSERT INTO RETAILORDERS
(ORDERNUMBER,ORDERDATE,REQUIREDDATE,SHIPPEDDATE,STATUS,COM
MENTS,CUSTOMERNUMBER)

VALUES (10399,'22-09-21','17-09-21','01-01-20','Cancel','Not',130);

SELECT *

FROM RETAILORDERS

WHERE ORDERNUMBER=10399;
```




20. Insert a new product into product table with product name as “2020 Bugatti Veyron” and productcode as “S111_111” and make up the rest of the fields.

INSERT INTO RETAILPRODUCTS (PRODUCTCODE,PRODUCTNAME,PRODUCTLINE,RETAILPRODUCTSCALE,PRODUCTVENDOR,PRODUCTDESCRIPTION,QUANTITYINSTOCK,BUYPRICE,MSRP)

VALUES ('S111_111','2020 Bugatti Veyron','Vintage Cars','1:25','NEXUS','Not Good',130,130.7,180.9);