Lab 05

# Objectives:

The purpose of this lab is to familiarize you with tables and set operations.

# LAB 05 – SUBMISSION

You will be submitting a Word document with the questions listed followed by the answers with the SQL queries and screenshots of the results. Use () brackets for subqueries and try to run the subquery first if it executes fine without any errors

## In the Word document header, you should have your Name, Student ID number, section.

**QUESTION 1**

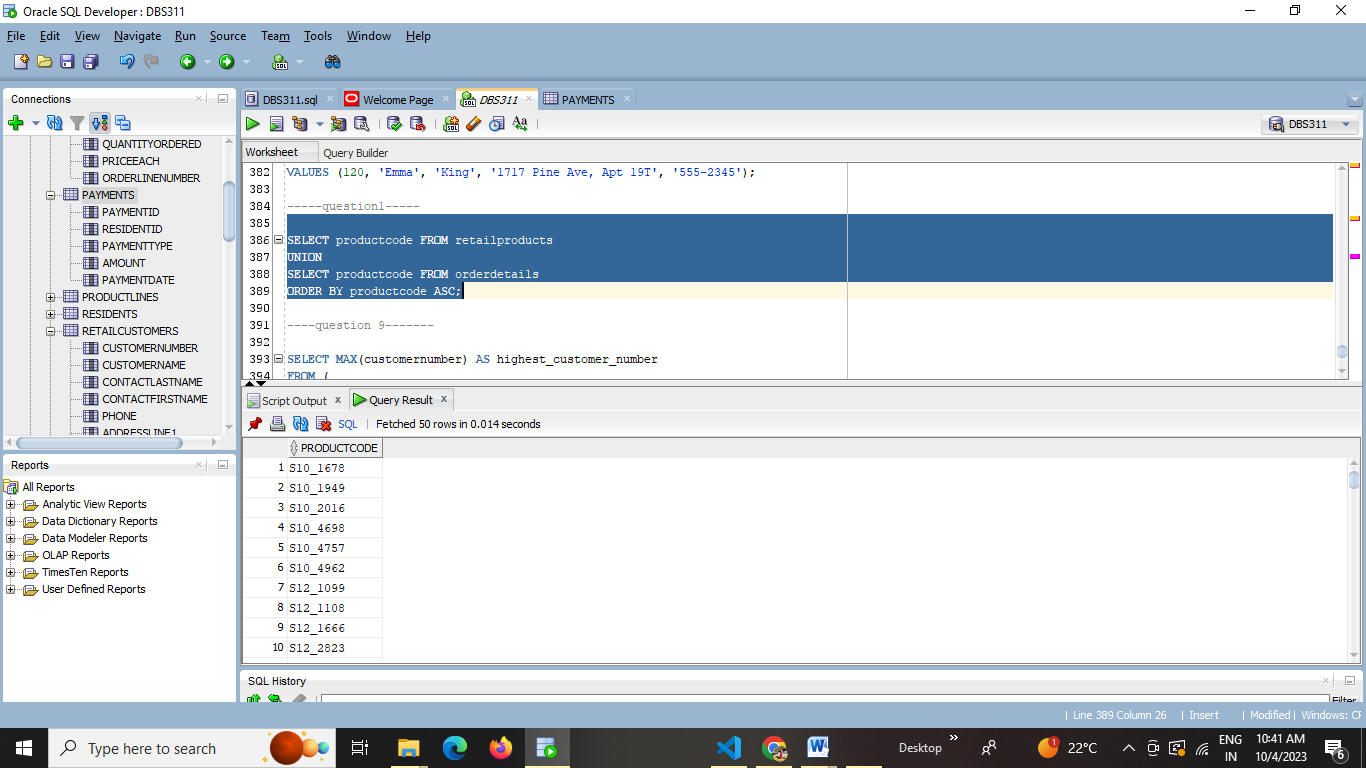
Write a SQL query using SET operators to display all the product codes of retailproducts and orderdetails and sort them in ascending order.

**SELECT productcode FROM retailproducts**

**UNION**

**SELECT productcode FROM orderdetails**

**ORDER BY productcode ASC;**



**QUESTION 2**

Write a SQL query using SET operators to display all the product codes that are not duplicates of retailproducts and orderdetails and sort them in ascending order.

**SELECT productcode**

**FROM retailproducts**

**WHERE productcode NOT IN (**

**SELECT productcode FROM orderdetails**

**)**

**UNION**

**SELECT productcode**

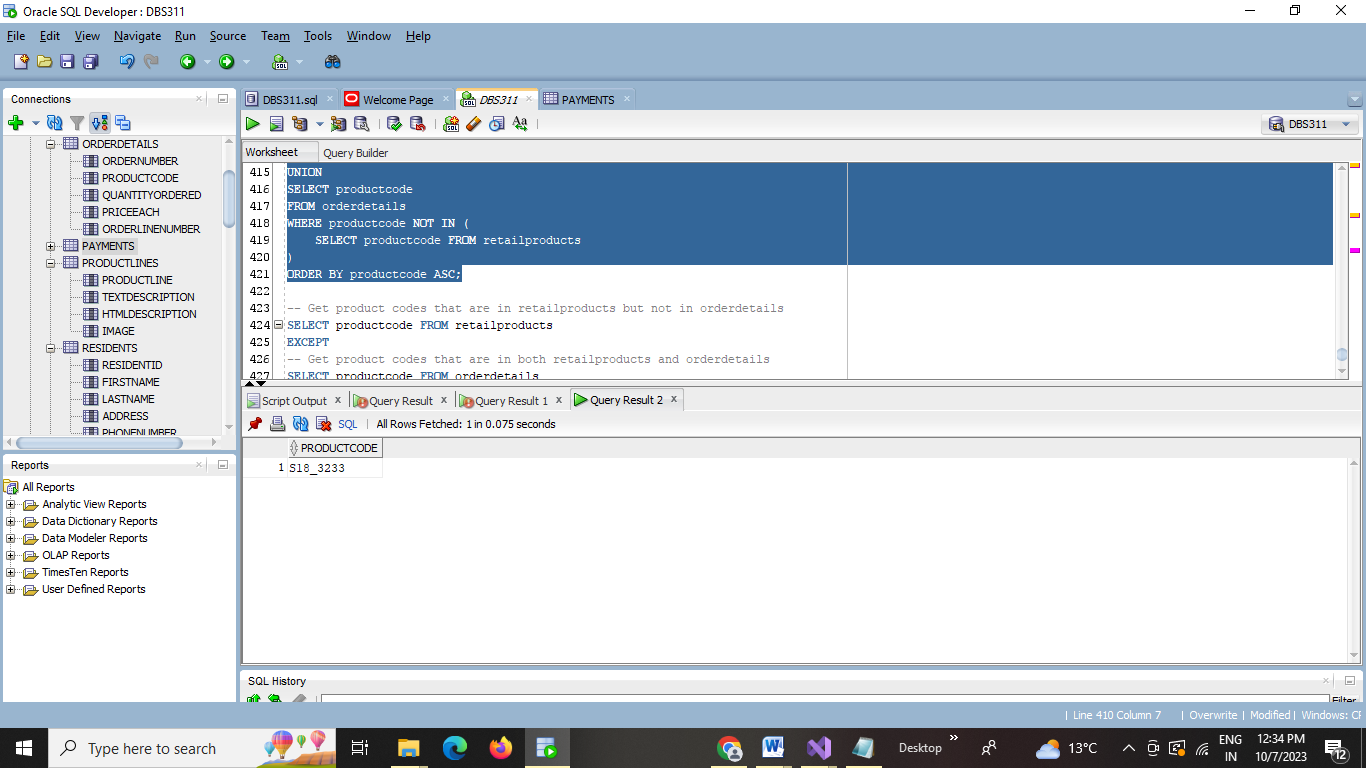
**FROM orderdetails**

**WHERE productcode NOT IN (**

**SELECT productcode FROM retailproducts**

**)**

**ORDER BY productcode ASC;**



**QUESTION 3**

Write a SQL query using SET operators to display only the customer numbers that are in retailcustomers but not in retailpayments and sort them in ascending order.

**SELECT customernumber**

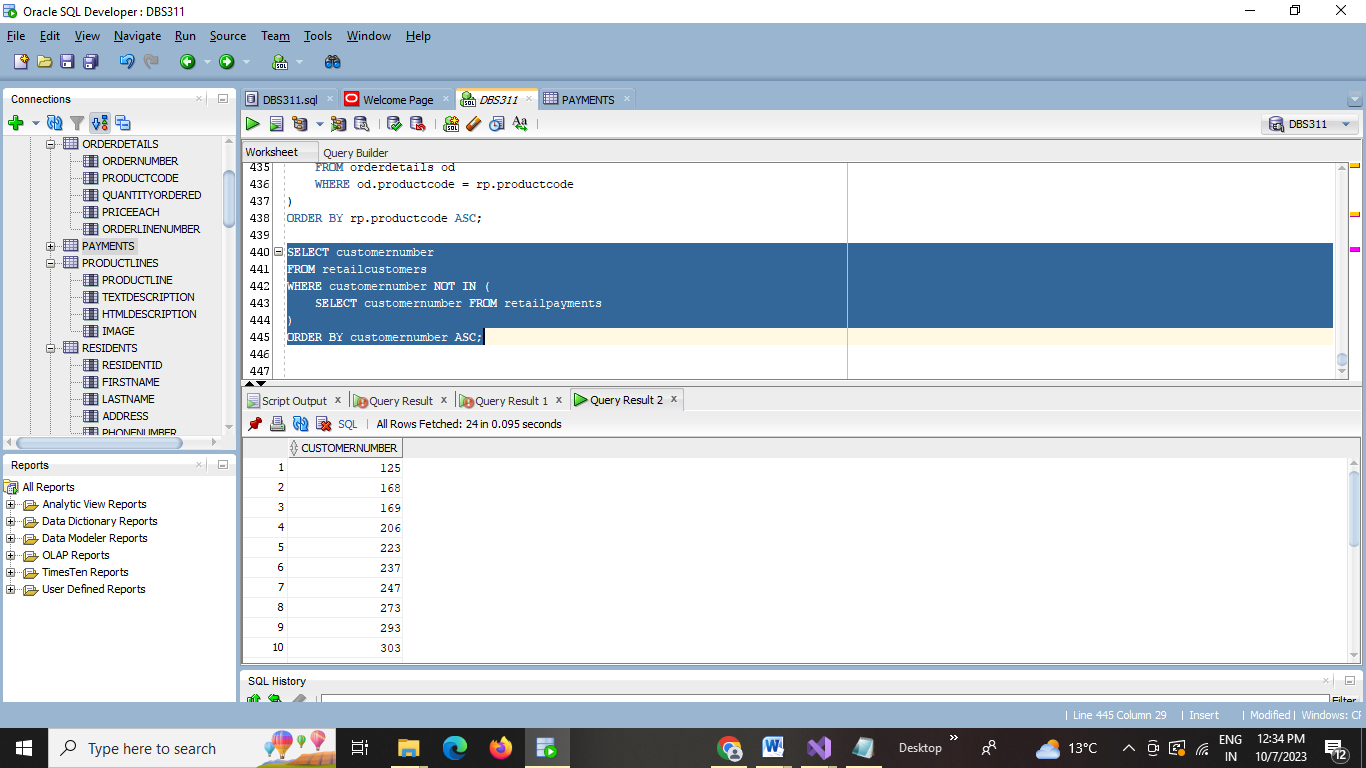
**FROM retailcustomers**

**WHERE customernumber NOT IN (**

**SELECT customernumber FROM retailpayments**

**)**

**ORDER BY customernumber ASC;**



**QUESTION 4**

Write a SQL query using SET operators to display only the customer numbers that are in retailpayments but not in retailcustomers and sort them in ascending order.

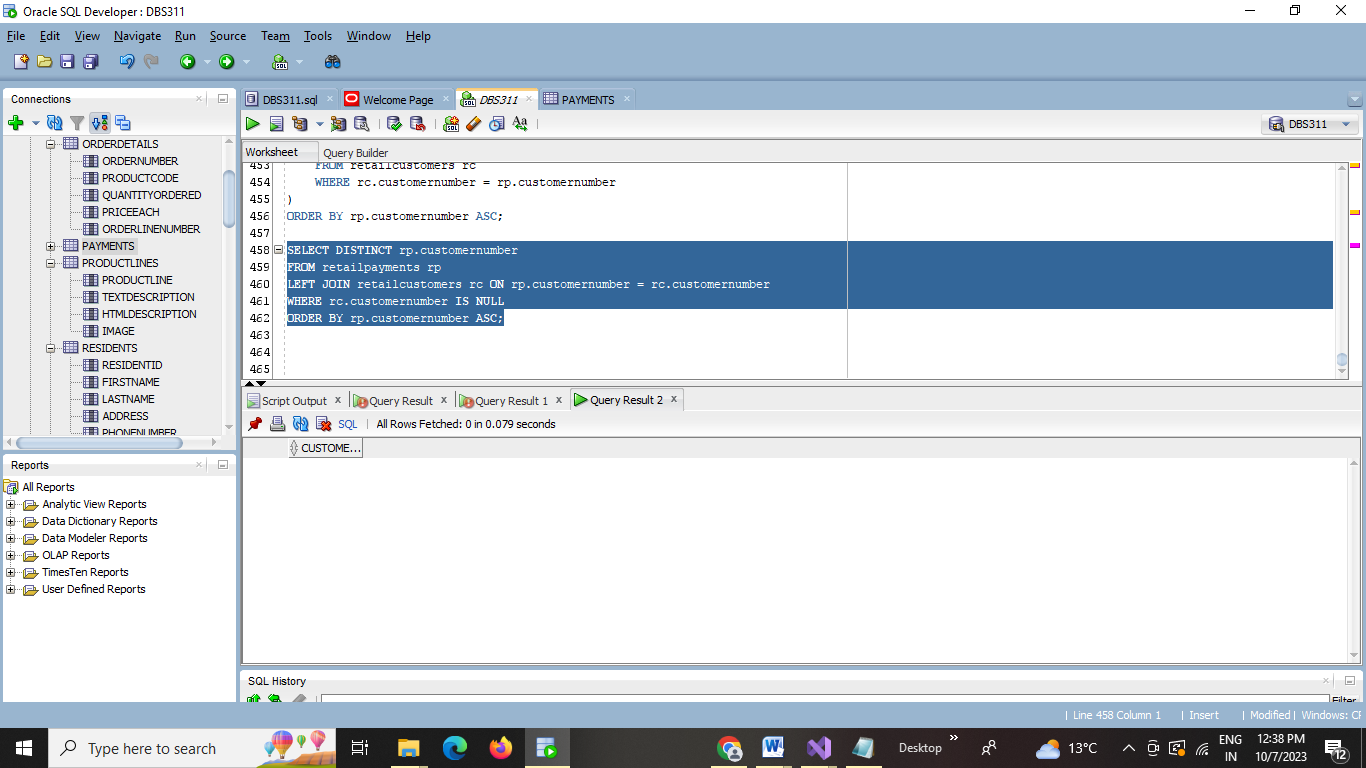
**SELECT DISTINCT rp.customernumber**

**FROM retailpayments rp**

**LEFT JOIN retailcustomers rc ON rp.customernumber = rc.customernumber**

**WHERE rc.customernumber IS NULL**

**ORDER BY rp.customernumber ASC;**



**QUESTION 5**

Write a SQL query using SET operators to display only the customer numbers that are common both retailpayments and in retailcustomers and sort them in ascending order.

**SELECT DISTINCT customernumber**

**FROM (**

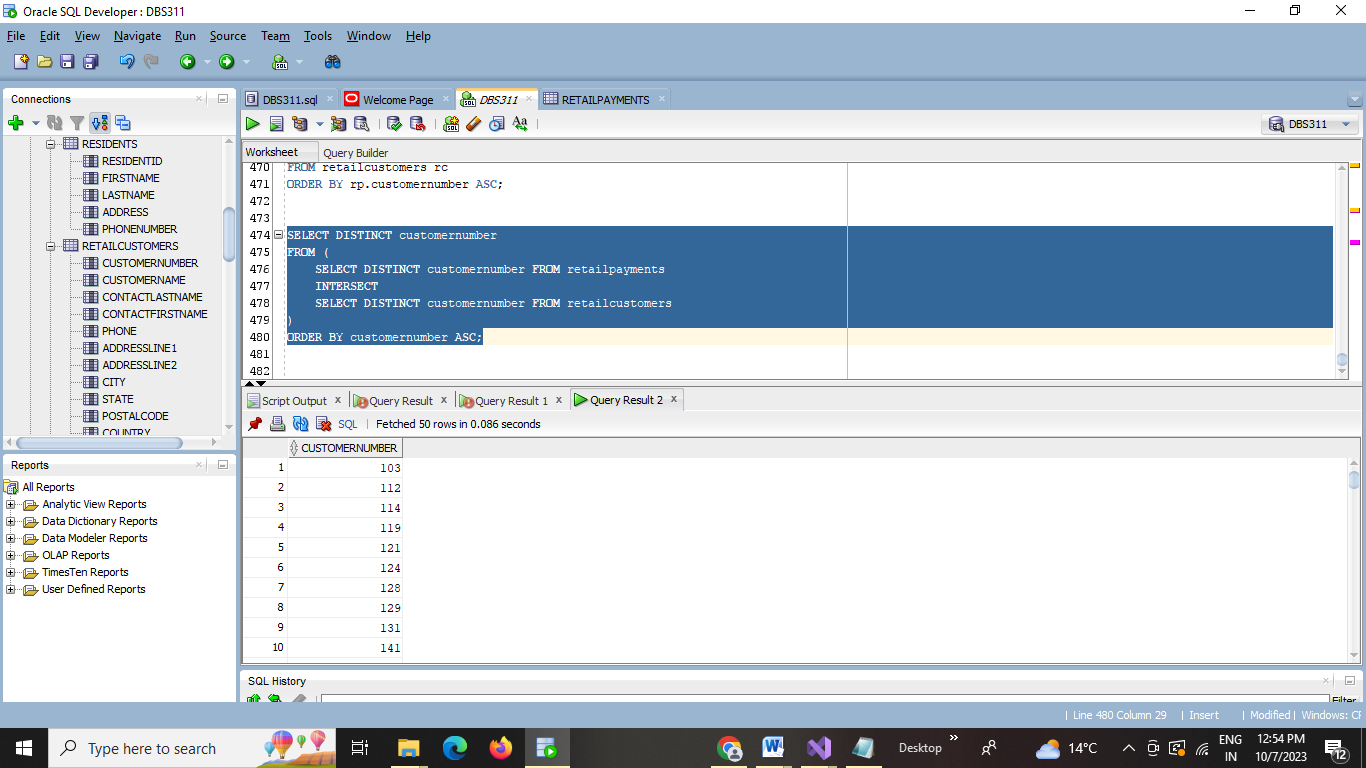
**SELECT DISTINCT customernumber FROM retailpayments**

**INTERSECT**

**SELECT DISTINCT customernumber FROM retailcustomers**

**)**

**ORDER BY customernumber ASC;**



## QUESTION 6

Write a SQL query using SET operators to display all but without duplicates of customer number and amount from retailpayments and retailcustomers and sort them in ascending order. Use to\_number(null) and to\_char(null) if a integer or character value is not in a table

**SELECT customernumber,**

**to\_number(NULL) as amount**

**FROM retailcustomers**

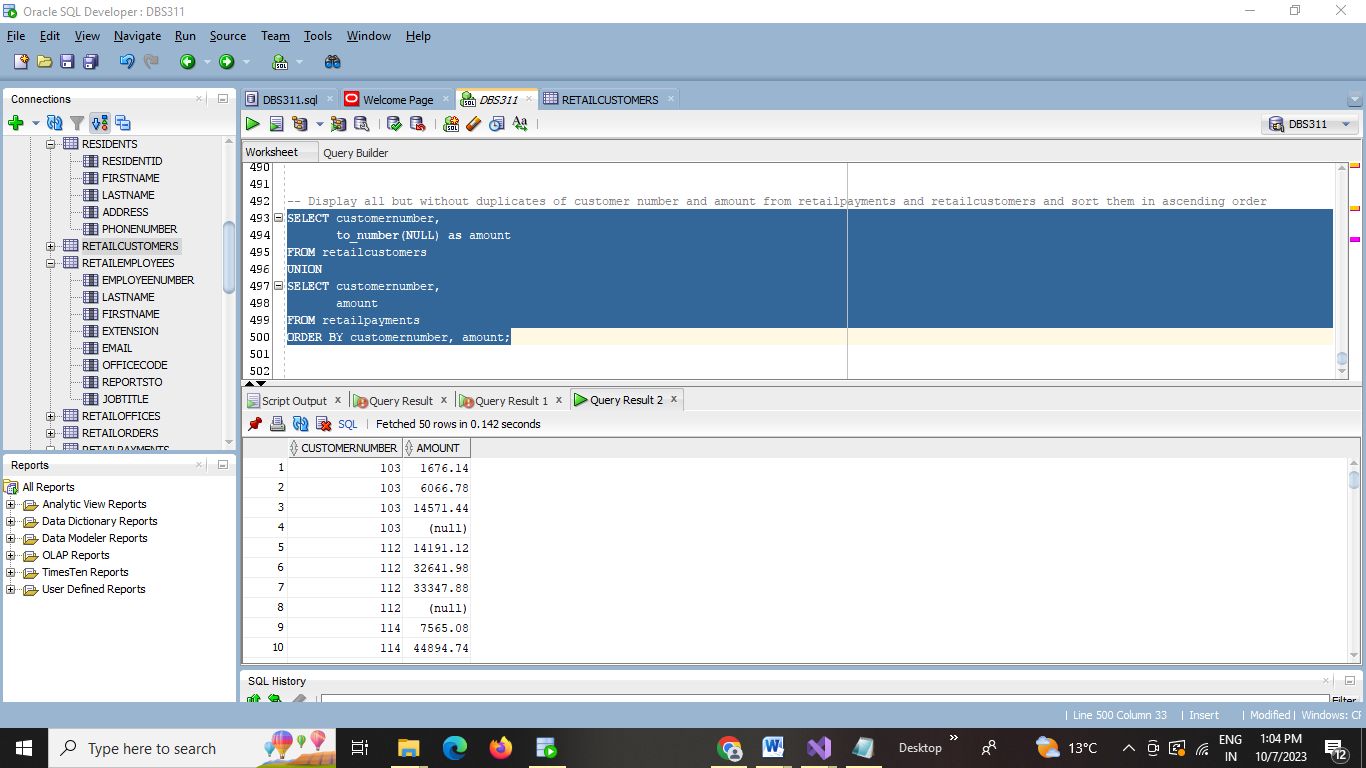
**UNION**

**SELECT customernumber,**

**amount**

**FROM retailpayments**

**ORDER BY customernumber, amount;**



## QUESTION 7

Write a SQL query using SET operators to display all but without duplicates of product code and product names from retailproducts and orderdetails and sort them in descending order. Use to\_number(null) and to\_char(null) if a integer or character value is not in a table

**SELECT productcode, productname**

**FROM retailproducts**

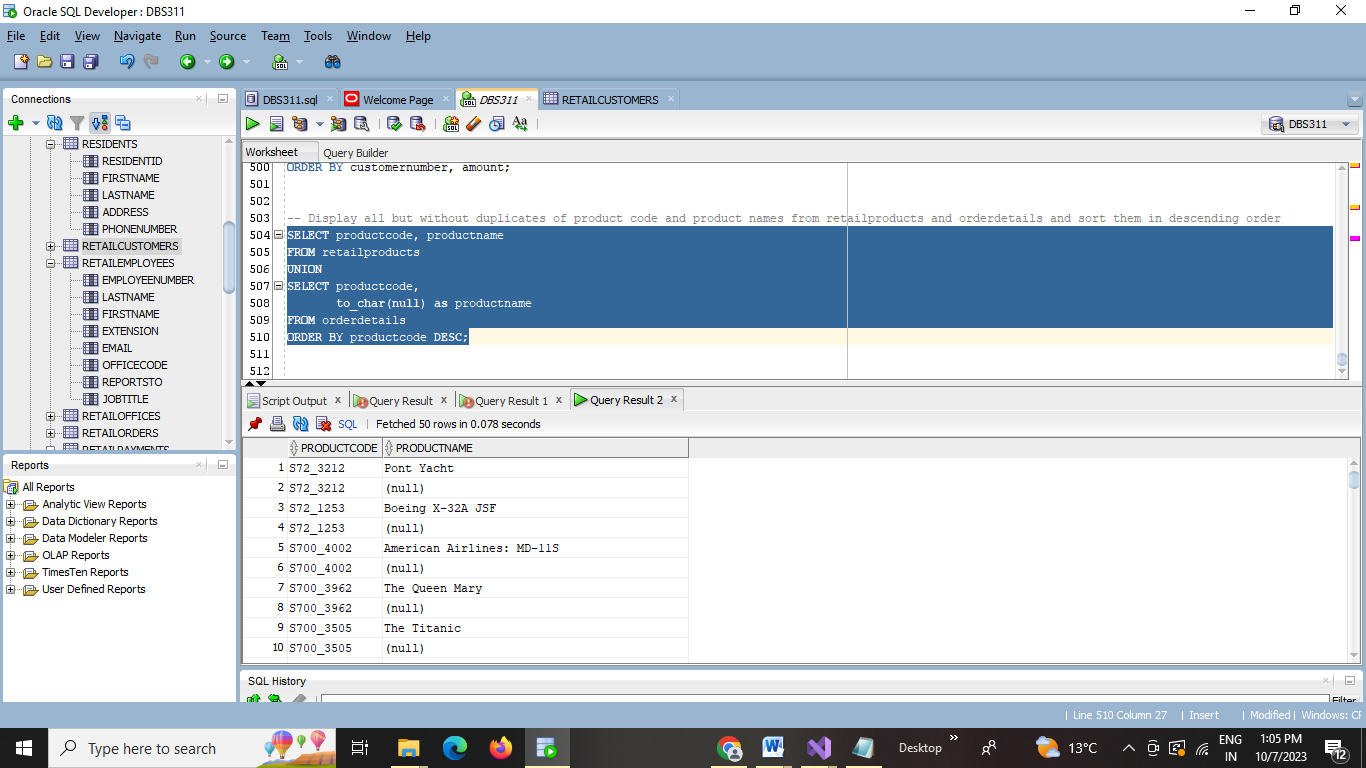
**UNION**

**SELECT productcode,**

**to\_char(null) as productname**

**FROM orderdetails**

**ORDER BY productcode DESC;**



## QUESTION 8

Write a SQL query using SET operators to display all but without duplicates of officecode, city and employee’s firstname from retailoffices and retailemployees and sort them in descending order. Use the strings ‘No\_City’ and ‘No\_Firstname’ if a integer or character value is not in a table

**SELECT officecode,**

**COALESCE(to\_char(null), 'No\_City') as city,**

**COALESCE(to\_char(null), 'No\_Firstname') as firstname**

**FROM retailOffices**

**UNION**

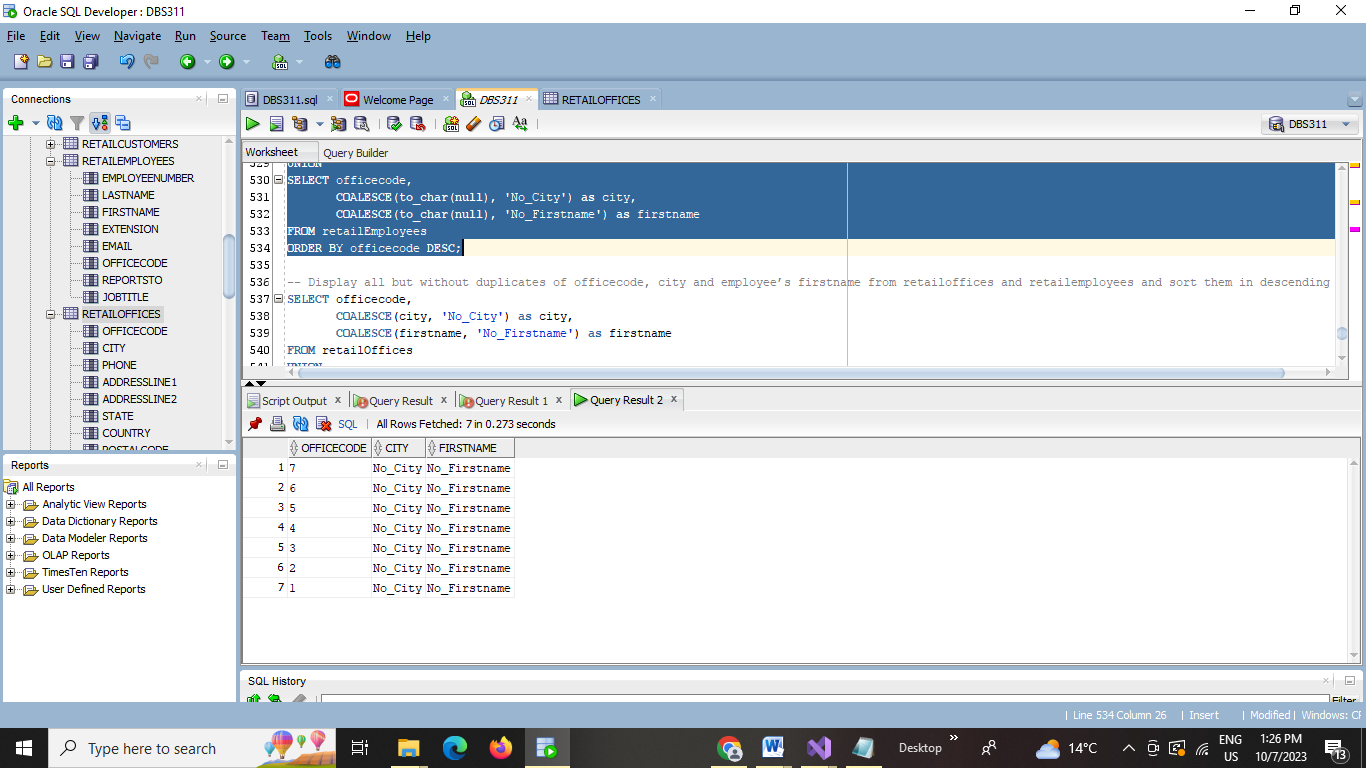
**SELECT officecode,**

**COALESCE(to\_char(null), 'No\_City') as city,**

**COALESCE(to\_char(null), 'No\_Firstname') as firstname**

**FROM retailEmployees**

**ORDER BY officecode DESC;**



## QUESTION 9

Write a SQL query using SET operators and subqueries with group functions to display the lowest and highest customer number from all the customer numbers without duplicates from the tables retailpayments, retailcustomers and retailorders and sort them in descending order.

**SELECT MAX(customernumber) AS highest\_customer\_number**

**, MIN(customernumber) AS lowest\_customer\_number**

**FROM (**

**SELECT customernumber FROM retailpayments**

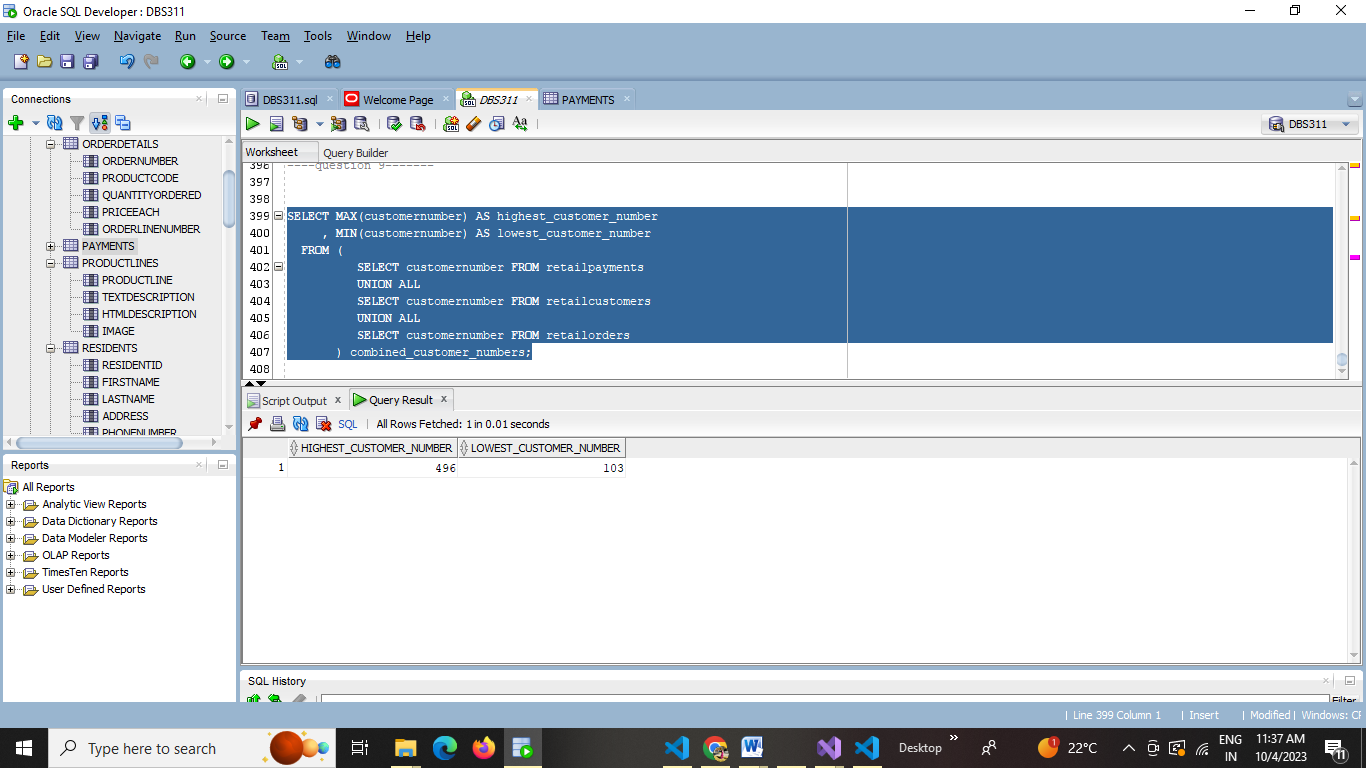
**UNION ALL**

**SELECT customernumber FROM retailcustomers**

**UNION ALL**

**SELECT customernumber FROM retailorders**

**) combined\_customer\_numbers;**



## QUESTION 10

Write a SQL query using SET operators and subqueries with group functions to display the total count from displaying all the customer number, order number , customer name, order date, amount from the tables retailpayments, retailcustomers and retailorders and sort them in descending order. Use the string ‘NULL’ and 0 if a integer or character value is not in a table

**SELECT**

**to\_number(null) as customer\_number,**

**COALESCE(ro.ordernumber, to\_number(null)) as order\_number,**

**COALESCE(rc.customername, 'NULL') as customer\_name,**

**COALESCE(to\_char(ro.orderdate, 'YYYY-MM-DD'), 'NULL') as order\_date,**

**COALESCE(rp.amount, 0) as amount**

**FROM**

**(SELECT COUNT(\*) as total\_count FROM retailpayments) dummy**

**LEFT JOIN retailpayments rp ON 1=1**

**LEFT JOIN retailcustomers rc ON to\_number(null) = rc.customernumber**

**LEFT JOIN retailorders ro ON to\_number(null) = ro.ordernumber**

**ORDER BY total\_count DESC;**

