MSDS 7330

File Organization and Database Management

Mini Project 3

Name: Cameron Stewart

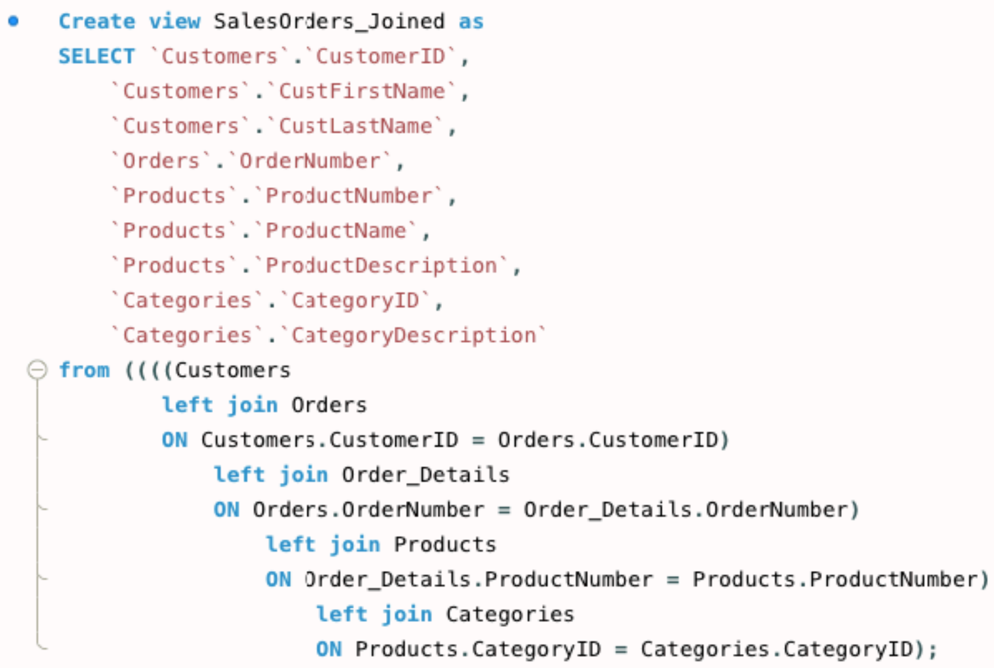
This is a mini project for MSDS 7330, File Organization and Database Management. For this assignment, turn in a single pdf file containing all of your answers. The file should be named ¡yourLastName¿MiniProject-Number.pdf. For example, the file name for my mini project 1 would be ‘RafiqiMiniProject-1.pdf’. Insert your answer pages into this file with the answer for Question 1 inserted immediately after Question 1 and before Question 2, the answer for Question 2 inserted immediately after Question 2, and the answer for Question 3 inserted immediately after Question 3. You may insert a front page containing your name and date if you do not wish to or cannot electronically add that information to the first page of this homework sheet.

Collaboration is expected and encouraged; however, each student must hand in their own homework assignment. To the greatest extent possible, answers should not be copied but, instead, should be written in your own words. Copying answers from anywhere is plagiarism, this includes copying text directly from the textbook. Do not copy answers. Always use your own words and your own code. Directly under each question list all persons with whom you collaborated and list all resources used in arriving at your answer. Resources include but are not limited to the textbook used for this course, papers read on the topic, and Google search results. Don’t forget to place your name on the first page of the pdf document.

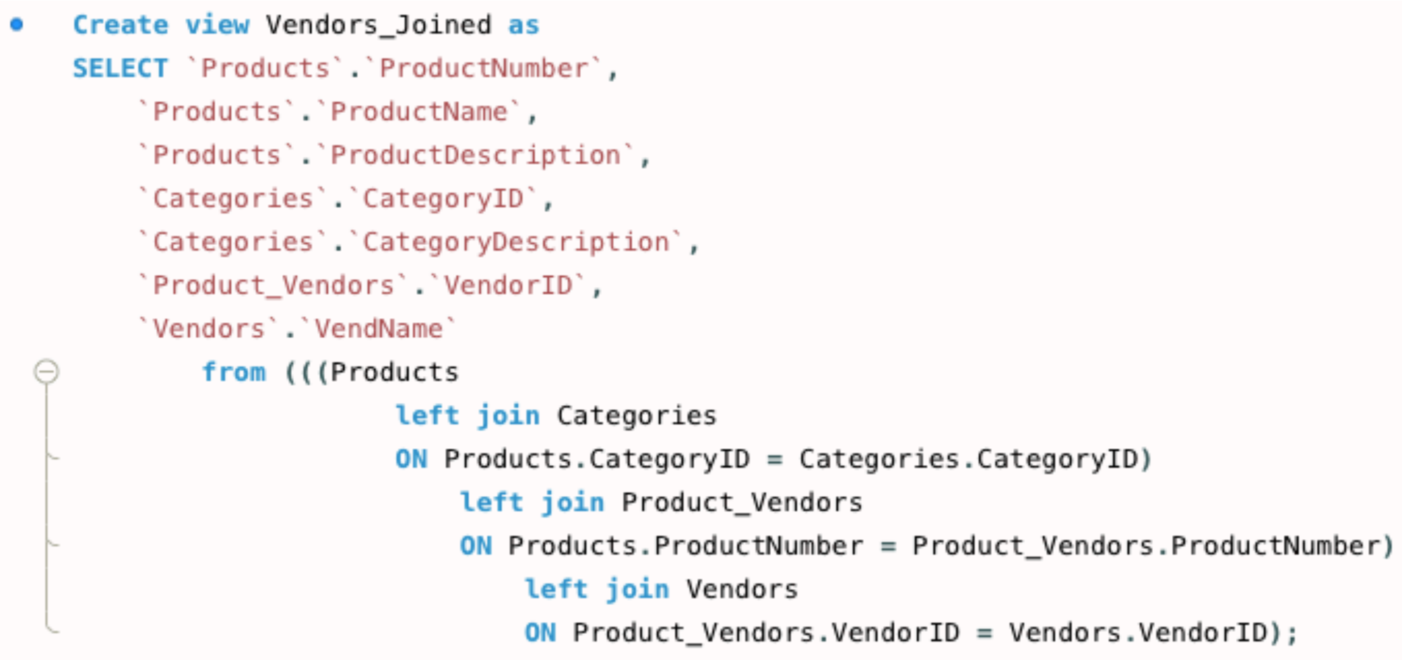
MySQL Database

Question 1: Use the Sales Order Database provided and answer the following queries using MySQL Workbench. Submit screenshots of queries along with screenshots of results. If results are longer than one page then simply provide a number of rows returned from the query. Answers for the following queries:

I created two views used in the below queries:

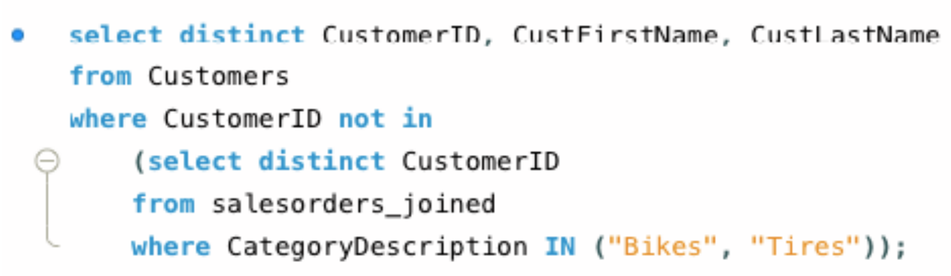


```
Create view SalesOrders_Joined as
SELECT `Customers`.`CustomerID`,
    `Customers`.`CustFirstName`,
    `Customers`.`CustLastName`,
    `Orders`.`OrderNumber`,
    `Products`.`ProductNumber`,
    `Products`.`ProductName`,
    `Products`.`ProductDescription`,
    `Categories`.`CategoryID`,
    `Categories`.`CategoryDescription`
from ((((Customers
        left join Orders
        ON Customers.CustomerID = Orders.CustomerID)
            left join Order_Details
            ON Orders.OrderNumber = Order_Details.OrderNumber)
                left join Products
                ON Order_Details.ProductNumber = Products.ProductNumber)
                    left join Categories
                    ON Products.CategoryID = Categories.CategoryID);
```

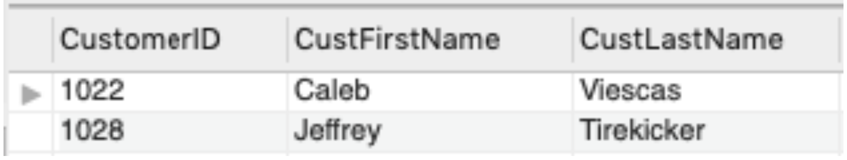


```
Create view Vendors_Joined as
SELECT `Products`.`ProductNumber`,
    `Products`.`ProductName`,
    `Products`.`ProductDescription`,
    `Categories`.`CategoryID`,
    `Categories`.`CategoryDescription`,
    `Product_Vendors`.`VendorID`,
    `Vendors`.`VendName`
        from (((Products
                    left join Categories
                    ON Products.CategoryID = Categories.CategoryID)
                        left join Product_Vendors
                        ON Products.ProductNumber = Product_Vendors.ProductNumber)
                            left join Vendors
                            ON Product_Vendors.VendorID = Vendors.VendorID);
```

1. Display the customers who have never ordered bikes or tires.

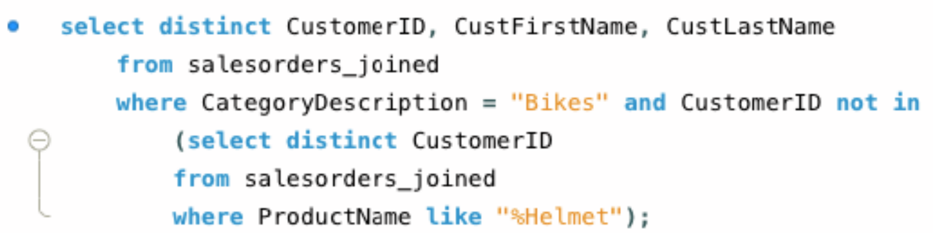


```
select distinct CustomerID, CustFirstName, CustLastName
from Customers
where CustomerID not in
    (select distinct CustomerID
    from salesorders_joined
    where CategoryDescription IN ("Bikes", "Tires"));
```

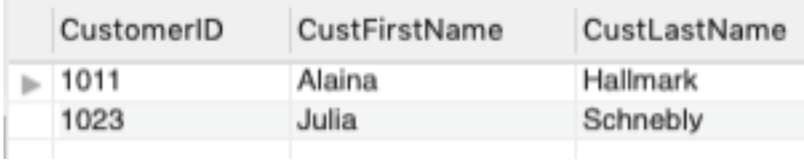


CustomerID	CustFirstName	CustLastName
1022	Caleb	Viescas
1028	Jeffrey	Tirekicker

2. List the customers who have purchased a bike but not a helmet.

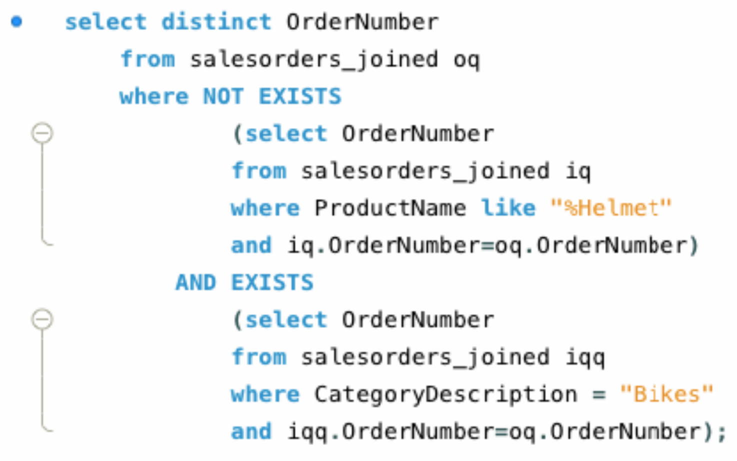


```
select distinct CustomerID, CustFirstName, CustLastName
    from salesorders_joined
    where CategoryDescription = "Bikes" and CustomerID not in
        (select distinct CustomerID
        from salesorders_joined
        where ProductName like "%Helmet");
```

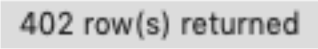


CustomerID	CustFirstName	CustLastName
1011	Alaina	Hallmark
1023	Julia	Schnebly

3. Show me the customer orders that have a bike but do not have a helmet. Hint: This might seem to be the same as problem 2 above, but it is not. Solve it using EXISTS and NOT EXISTS.

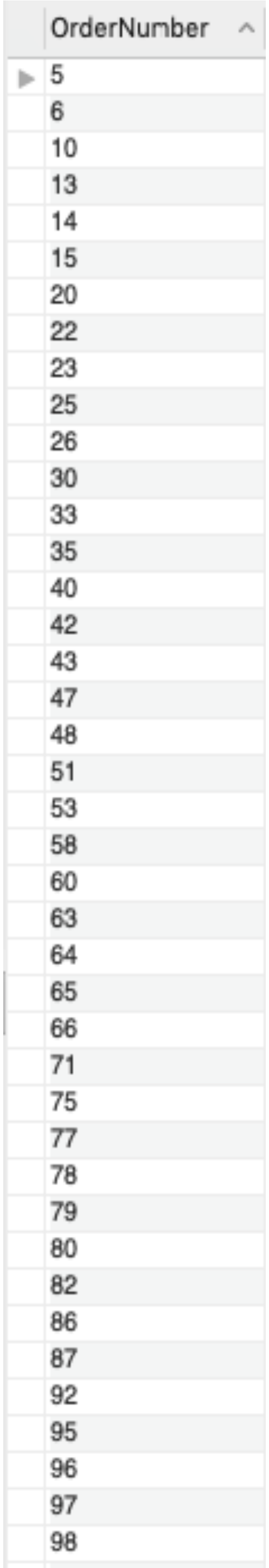


```
select distinct OrderNumber
    from salesorders_joined oq
    where NOT EXISTS
            (select OrderNumber
            from salesorders_joined iq
            where ProductName like "%Helmet"
            and iq.OrderNumber=oq.OrderNumber)
        AND EXISTS
            (select OrderNumber
            from salesorders_joined iqq
            where CategoryDescription = "Bikes"
            and iqq.OrderNumber=oq.OrderNunber);
```



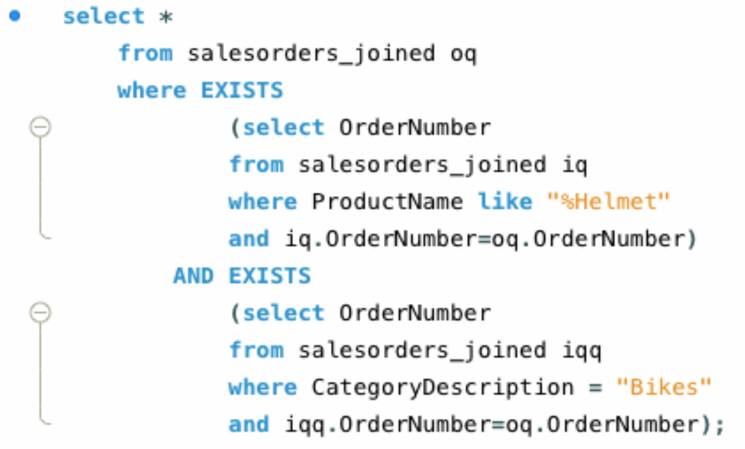
402 row(s) returned

Parital Output Below because it was too long.

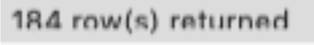


OrderNumber
5
6
10
13
14
15
20
22
23
25
26
30
33
35
40
42
43
47
48
51
53
58
60
63
64
65
66
71
75
77
78
79
80
82
86
87
92
95
96
97
98

4. Display the customers and their orders that have a bike and a helmet in the same order. Hint: Solve this problem using EXISTS

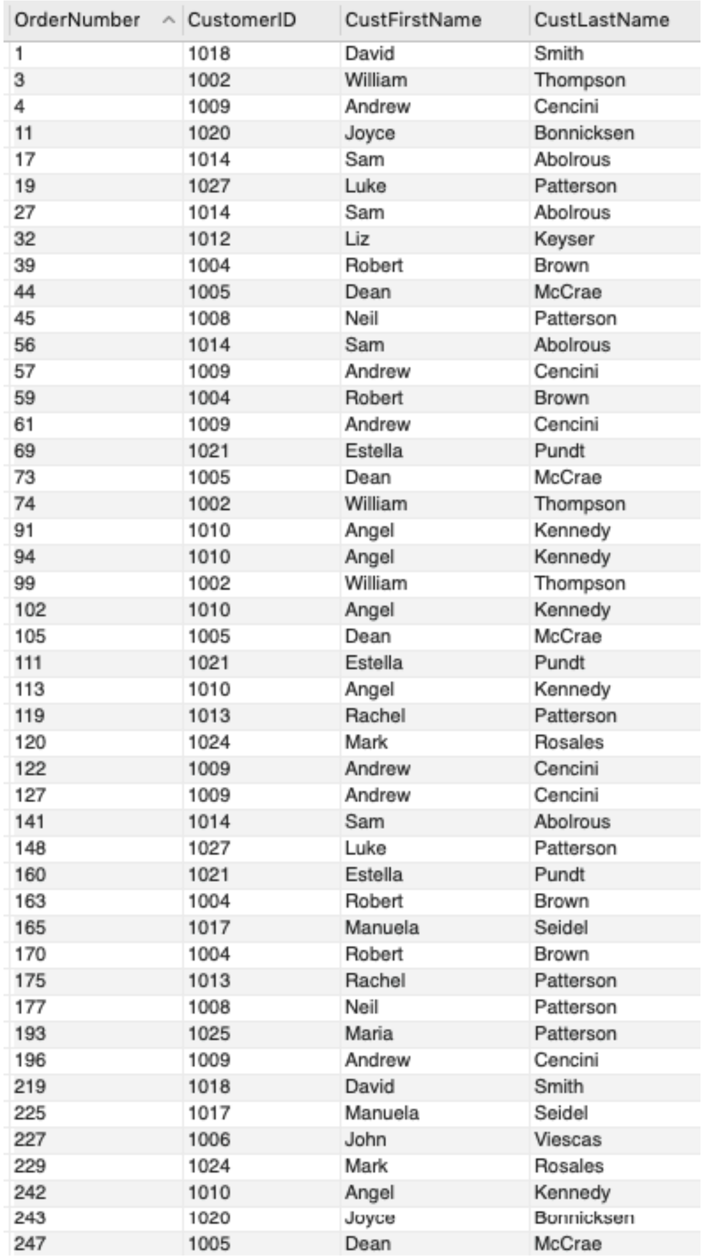


```
select *
    from salesorders_joined oq
    where EXISTS
            (select OrderNumber
            from salesorders_joined iq
            where ProductName like "%Helmet"
            and iq.OrderNumber=oq.OrderNumber)
        AND EXISTS
            (select OrderNumber
            from salesorders_joined iqq
            where CategoryDescription = "Bikes"
            and iqq.OrderNumber=oq.OrderNumber);
```



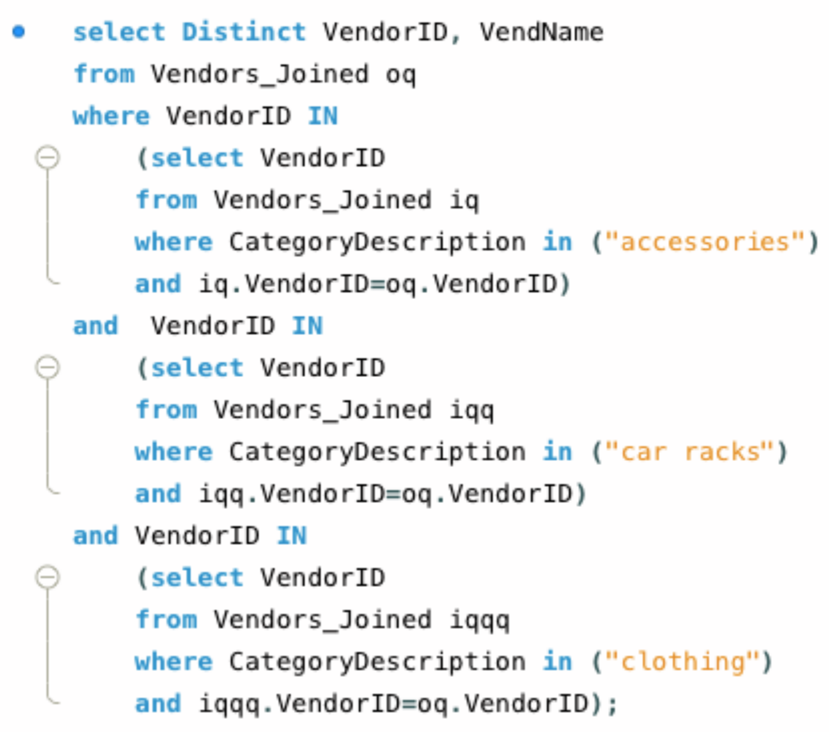
```
184 row(s) returned
```

Parital Output Below because it was too long.

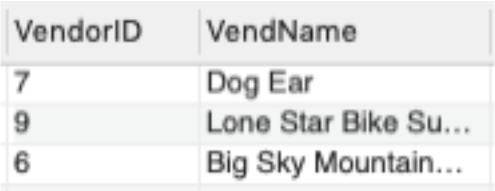


OrderNumber	CustomerID	CustFirstName	CustLastName
1	1018	David	Smith
3	1002	William	Thompson
4	1009	Andrew	Cencini
11	1020	Joyce	Bonnicksen
17	1014	Sam	Abolrous
19	1027	Luke	Patterson
27	1014	Sam	Abolrous
32	1012	Liz	Keyser
39	1004	Robert	Brown
44	1005	Dean	McCrae
45	1008	Neil	Patterson
56	1014	Sam	Abolrous
57	1009	Andrew	Cencini
59	1004	Robert	Brown
61	1009	Andrew	Cencini
69	1021	Estella	Pundt
73	1005	Dean	McCrae
74	1002	William	Thompson
91	1010	Angel	Kennedy
94	1010	Angel	Kennedy
99	1002	William	Thompson
102	1010	Angel	Kennedy
105	1005	Dean	McCrae
111	1021	Estella	Pundt
113	1010	Angel	Kennedy
119	1013	Rachel	Patterson
120	1024	Mark	Rosales
122	1009	Andrew	Cencini
127	1009	Andrew	Cencini
141	1014	Sam	Abolrous
148	1027	Luke	Patterson
160	1021	Estella	Pundt
163	1004	Robert	Brown
165	1017	Manuela	Seidel
170	1004	Robert	Brown
175	1013	Rachel	Patterson
177	1008	Neil	Patterson
193	1025	Maria	Patterson
196	1009	Andrew	Cencini
219	1018	David	Smith
225	1017	Manuela	Seidel
227	1006	John	Viescas
229	1024	Mark	Rosales
242	1010	Angel	Kennedy
243	1020	Joyce	Bonnicksen
247	1005	Dean	McCrae

5. Show the vendors who sell accessories, car racks, and clothing. Hint: Solve this problem using IN.



```
select Distinct VendorID, VendName
from Vendors_Joined oq
where VendorID IN
    (select VendorID
    from Vendors_Joined iq
    where CategoryDescription in ("accessories")
    and iq.VendorID=oq.VendorID)
and  VendorID IN
    (select VendorID
    from Vendors_Joined iqq
    where CategoryDescription in ("car racks")
    and iqq.VendorID=oq.VendorID)
and VendorID IN
    (select VendorID
    from Vendors_Joined iqqq
    where CategoryDescription in ("clothing")
    and iqqq.VendorID=oq.VendorID);
```



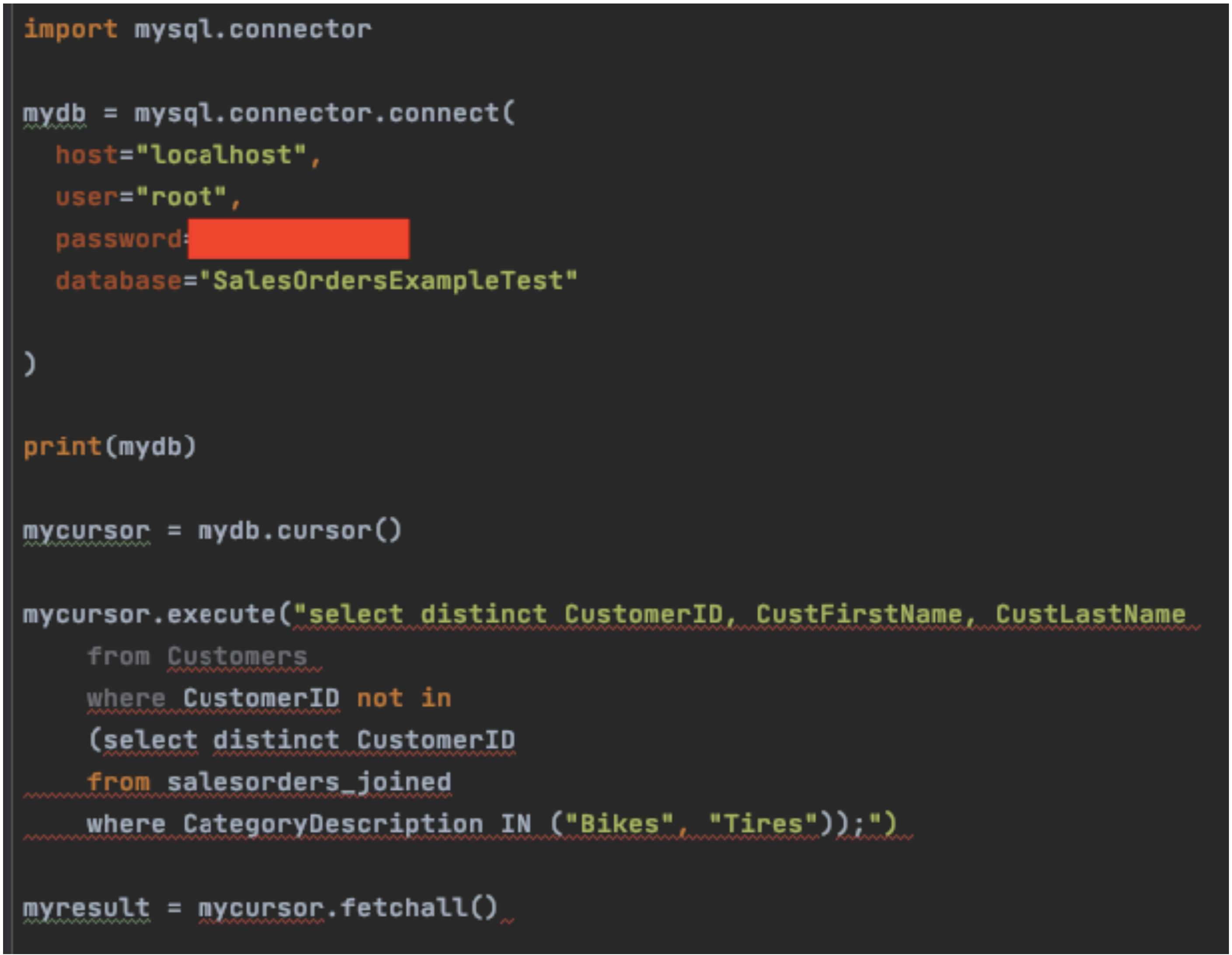
VendorID	VendName
7	Dog Ear
9	Lone Star Bike Su...
6	Big Sky Mountain...

Capture the screenshots of queries and resulting output

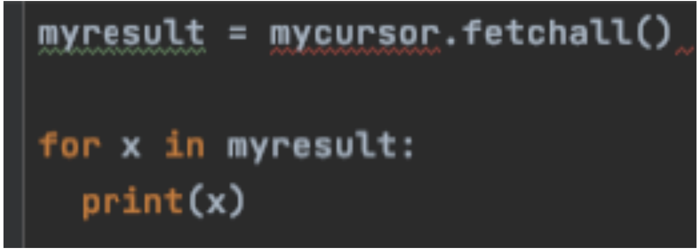
Question 2: Python – Write a Python Script that will connect to the Sales Order database and execute queries from question 1. The python script will connect to the MySQL database using MySQL connector and then you will execute the query using the cursor. To make it easier simply define the query at the beginning of the program. Submit a complete python script.

Hint:

Import MySql connector
Define server name, user name, password
connect to the database
initialize cursor, execute query

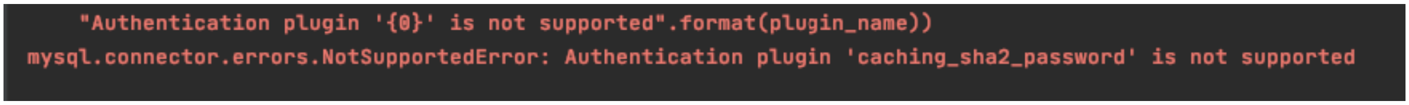


```
import mysql.connector

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password[redacted]
  database="SalesOrdersExampleTest"

)

print(mydb)

mycursor = mydb.cursor()

mycursor.execute("select distinct CustomerID, CustFirstName, CustLastName
     from Customers
     where CustomerID not in
     (select distinct CustomerID
     from salesorders_joined
     where CategoryDescription IN ("Bikes", "Tires"));")

myresult = mycursor.fetchall()
```



```
myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```

This is my first time using python. I spent 5 hours trying to resolve this error. I couldn’t get past it. I’ll keep trying.



```
    "Authentication plugin '{0}' is not supported".format(plugin_name))
mysql.connector.errors.NotSupportedError: Authentication plugin 'caching_sha2_password' is not supported
```

Resources:

https://www.w3schools.com/python/python_mysql_getstarted.asp

https://pynative.com/python-mysql-database-connection/