**Part II: [Total Points: 70]**

By Stephanie Boissonneault

**Notes**

• For each of the following 7 questions: Write your answer as a MySQL query statement, execute it, and take a screenshot of the results. If the query result has more than 10 records, limit the output to the top 10 before taking the screenshot. Then, paste the answer and the respective screenshot into a Word or Google document file with the question number.

**• Submission:** Submit your answer sheet for this part either in PDF or DOCX file format.

**1. [8 pts.] Retrieve all products with their corresponding category names and supplier names.**

USE electronicsdb;

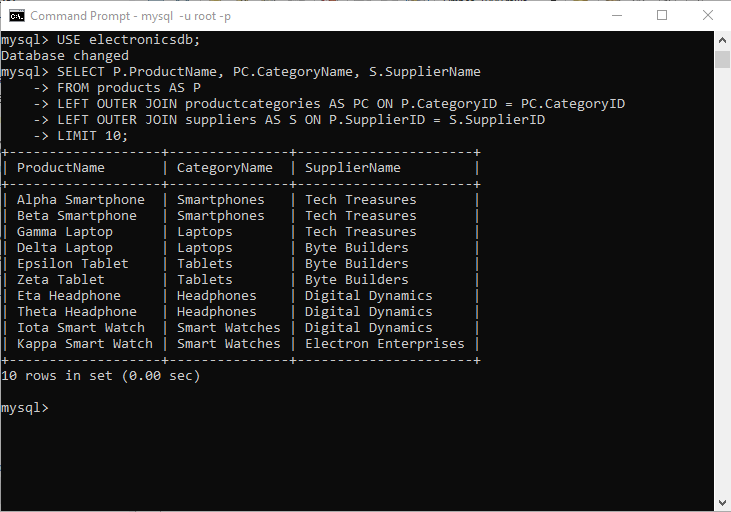
SELECT P.ProductName, PC.CategoryName, S.SupplierName

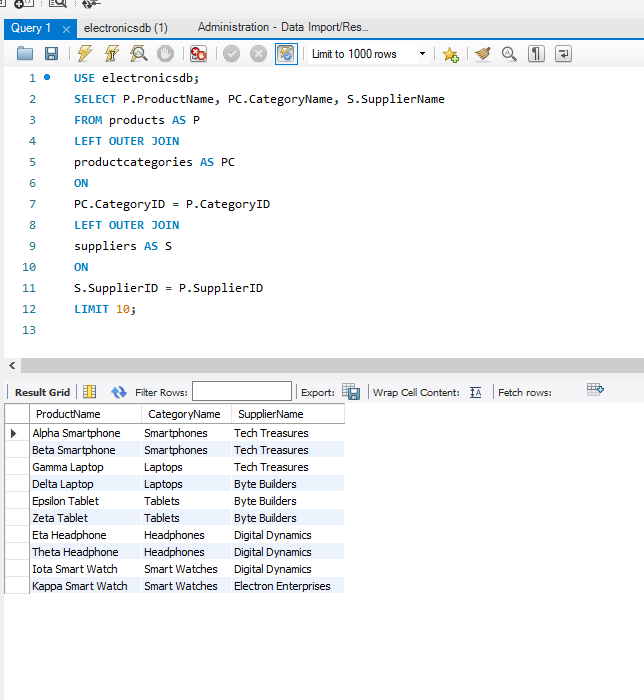
FROM products AS P

LEFT OUTER JOIN productcategories AS PC ON P.CategoryID = PC.CategoryID

LEFT OUTER JOIN suppliers AS S ON P.SupplierID = S.SupplierID

LIMIT 10;





**2. [8 pts.] Retrieve the names of all products with the same supplier as ’Alpha Smartphone’.**

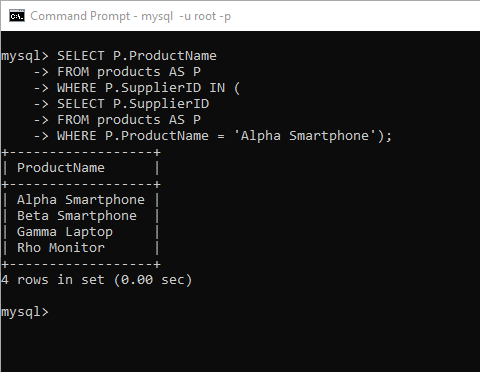
SELECT P.ProductName

FROM products AS P

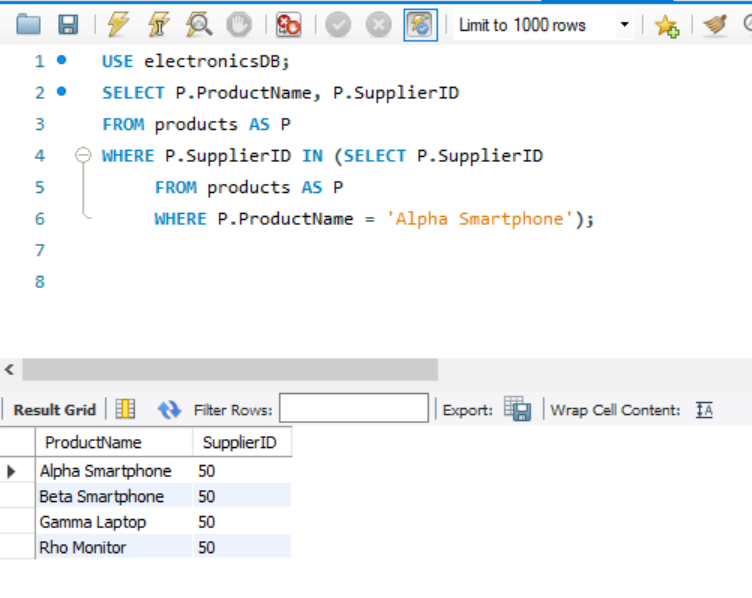
WHERE P.SupplierID IN (SELECT P.SupplierID

FROM products AS P

WHERE P.ProductName = 'Alpha Smartphone');



In the following screenshot, I have also selected to display the supplier ID to confirm that they are all the same supplier as Alpha Smartphone:

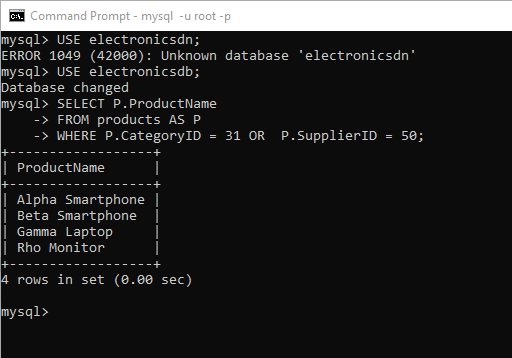


**3. [10 pts.] Retrieve all products that are either in the ’Smartphones’ category (CategoryID 31) or supplied by SupplierID 50.**

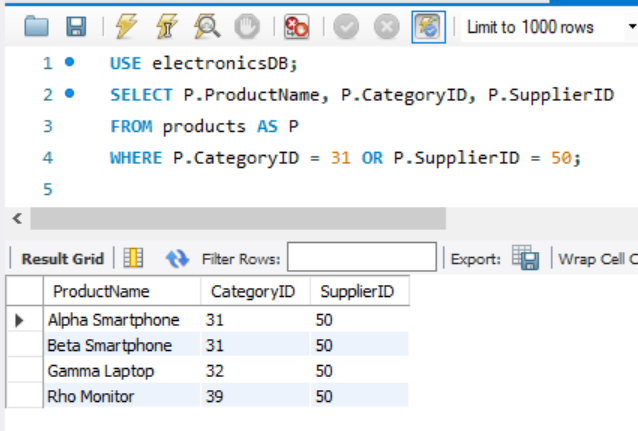
SELECT P.ProductName, P.CategoryID, P.SupplierID

FROM products AS P

WHERE P.CategoryID = 31 OR P.SupplierID = 50;



Here I have additionally selected to display the CategoryID and the SupplierID to confirm that retrievals are either (CategoryID 31) or SupplierID 50:

****

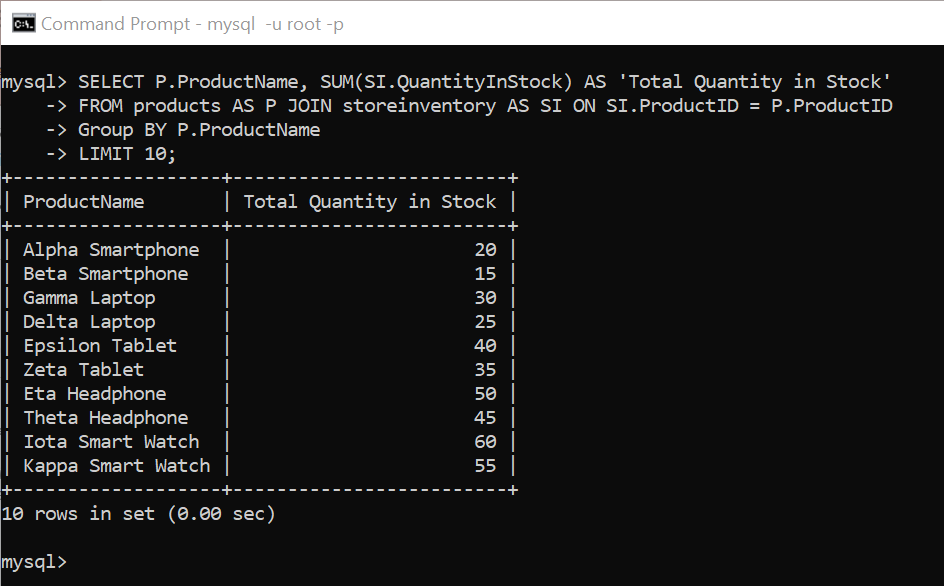
**4. [10 pts.] Find the total quantity in stock for each product.**

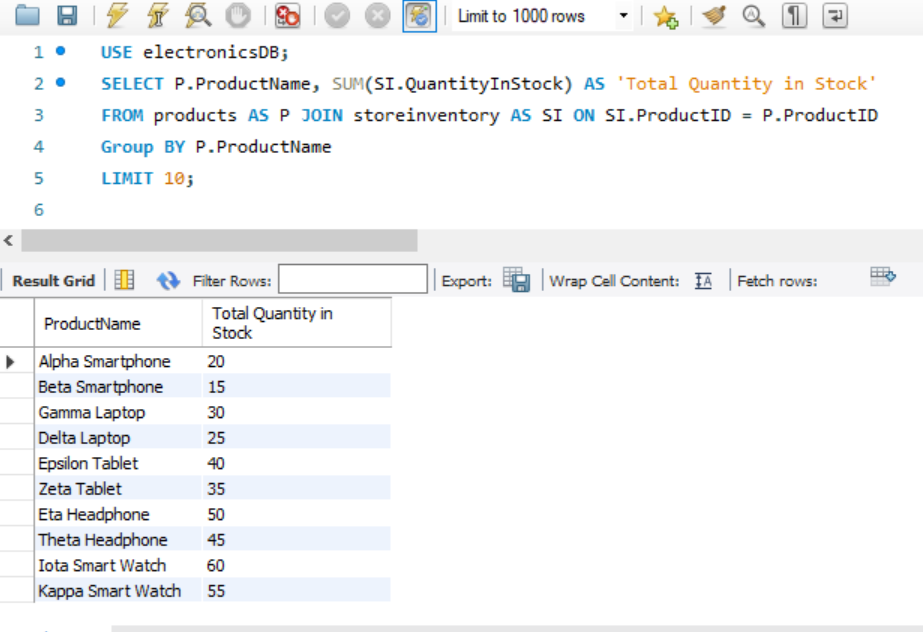
SELECT P.ProductName, SUM(SI.QuantityInStock) AS 'Total Quantity in Stock'

FROM products AS P JOIN storeinventory AS SI ON SI.ProductID = P.ProductID

Group BY P.ProductName

LIMIT 10;





**5. [10 pts.] Calculate the average stock quantity for each product category and list the categories with an average stock above 70.**

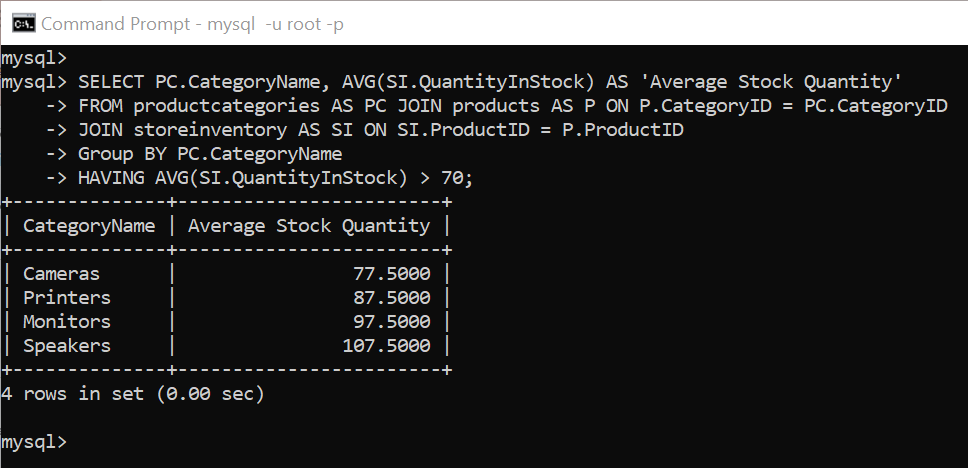
SELECT PC.CategoryName, AVG(SI.QuantityInStock) AS 'Average Stock Quantity'

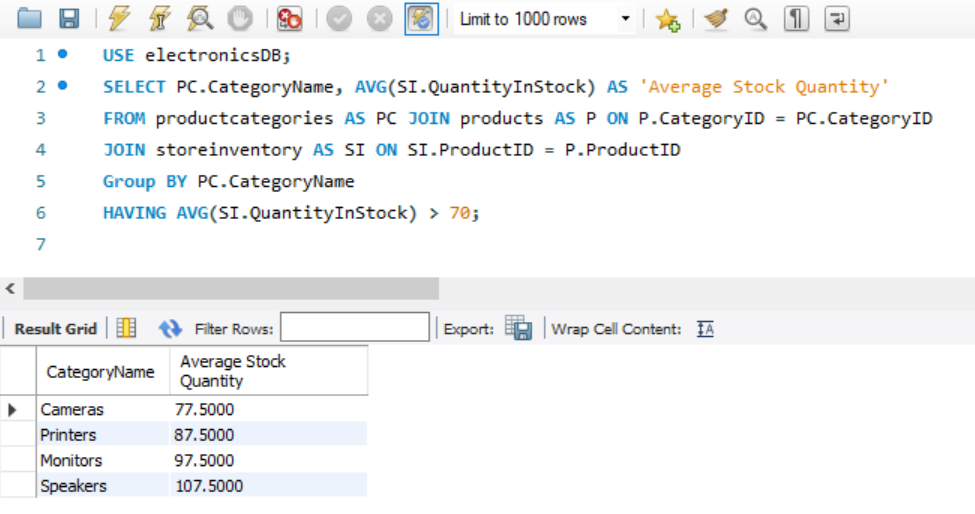
FROM productcategories AS PC JOIN products AS P ON P.CategoryID = PC.CategoryID

JOIN storeinventory AS SI ON SI.ProductID = P.ProductID

Group BY PC.CategoryName

HAVING AVG(SI.QuantityInStock) > 70;

****

****

**6. [12 pts.] Which products in the ’Cameras’ category are supplied by at least one supplier?**

SELECT P.ProductName, P.ProductID, P.SupplierID, COUNT(P.SupplierID) AS SupplierCount

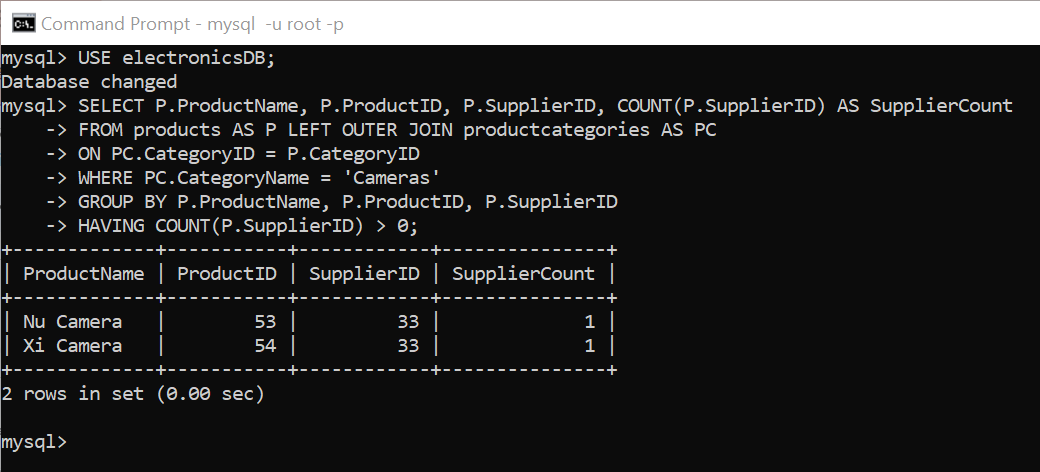
FROM products AS P LEFT OUTER JOIN productcategories AS PC

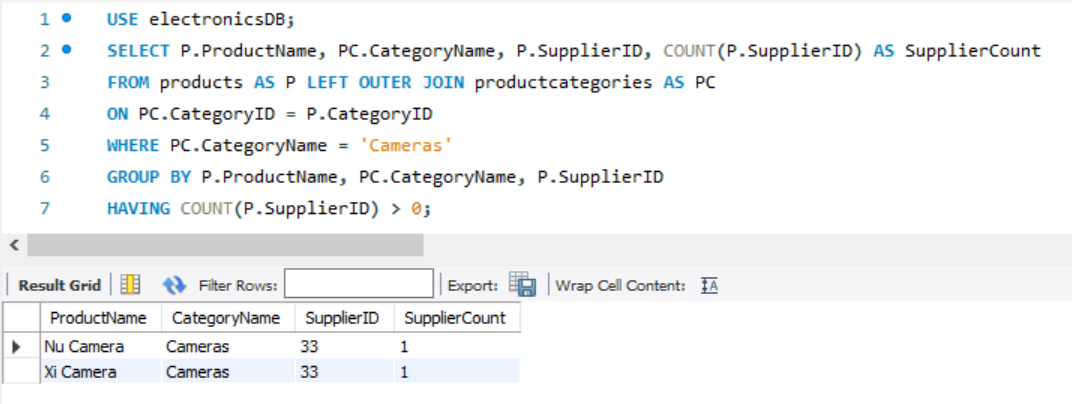
ON PC.CategoryID = P.CategoryID

WHERE PC.CategoryName = 'Cameras'

GROUP BY P.ProductName, P.ProductID, P.SupplierID

HAVING COUNT(P.SupplierID) > 0;





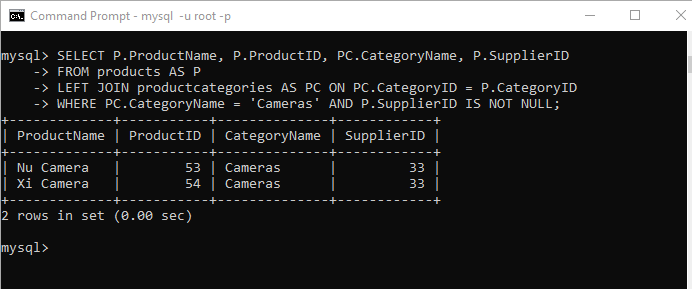
**Could also check that they are supplied by at least one supplier by checking that the SupplierID is NOT NULL (implying that there is at least one supplier):**

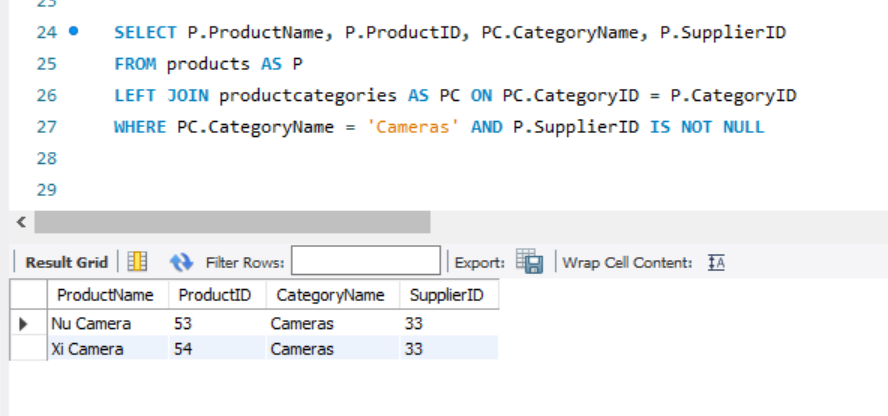
SELECT P.ProductName, P.ProductID, PC.CategoryName, P.SupplierID

FROM products AS P

LEFT JOIN productcategories AS PC ON PC.CategoryID = P.CategoryID

WHERE PC.CategoryName = 'Cameras' AND P.SupplierID IS NOT NULL;



****

**7. [12 pts.] Find products with a total quantity in stock higher than the average stock quantity of all products in their respective category.**

SELECT P.ProductName, SUM(SI.QuantityInStock) AS 'Total Quantity in Stock', P.CategoryID,

(SELECT AVG(SI2.QuantityInStock)

FROM storeinventory AS SI2 JOIN products AS P2

ON SI2.ProductID = P2.ProductID

WHERE P2.CategoryID = P.CategoryID) AS 'Average Stock Quantity'

FROM products AS P JOIN storeinventory AS SI ON SI.ProductID = P.ProductID

GROUP BY P.ProductName, P.CategoryID

HAVING SUM(SI.QuantityInStock) >

(SELECT AVG(SI2.QuantityInStock)

FROM storeinventory AS SI2 JOIN products AS P2   
 ON SI2.ProductID = P2.ProductID

WHERE P2.CategoryID = P.CategoryID);

