

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;

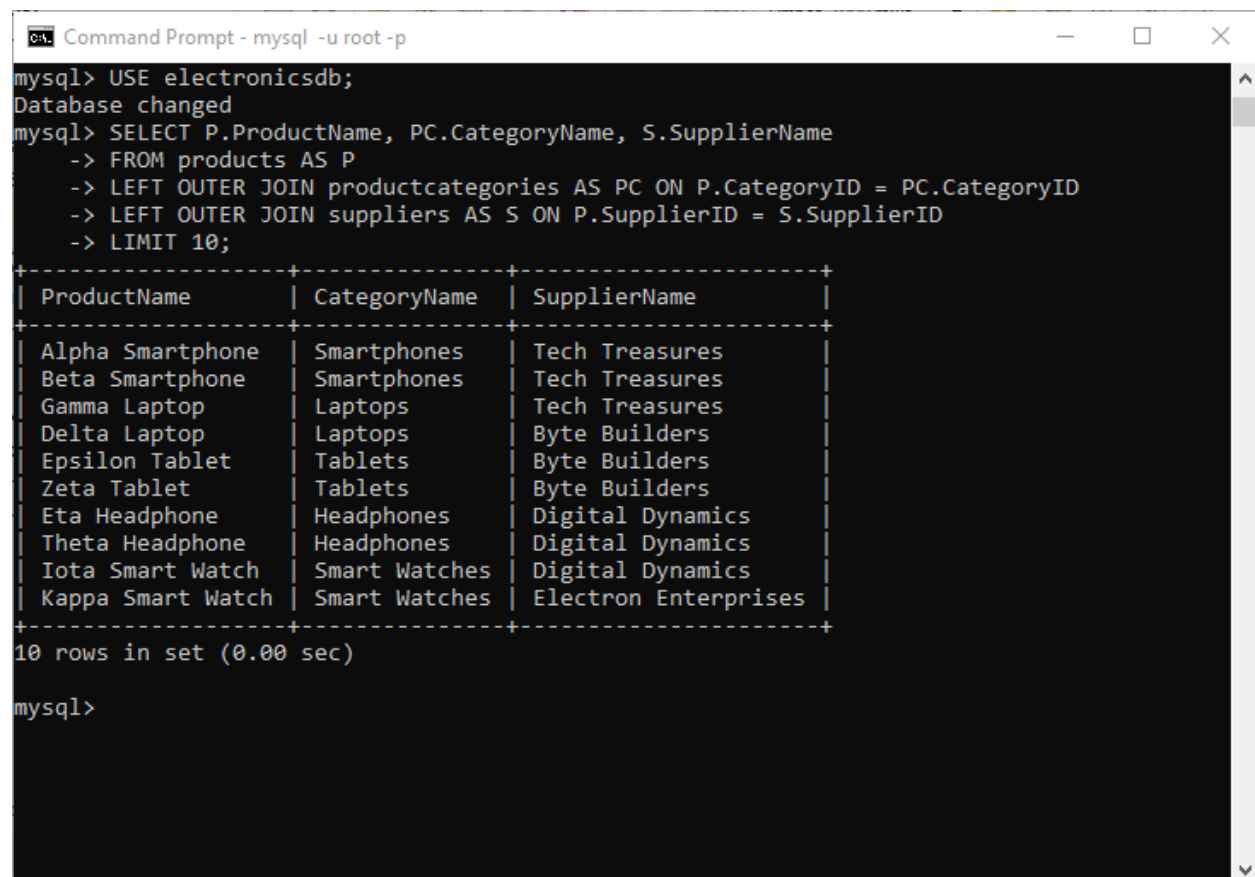
SELECT P.ProductName, PC.CategoryName, S.SupplierName

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;



```
mysql> USE electronicsdb;
Database changed
mysql> 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;
```

ProductName	CategoryName	SupplierName
Alpha Smartphone	Smartphones	Tech Treasures
Beta Smartphone	Smartphones	Tech Treasures
Gamma Laptop	Laptops	Tech Treasures
Delta Laptop	Laptops	Byte Builders
Epsilon Tablet	Tablets	Byte Builders
Zeta Tablet	Tablets	Byte Builders
Eta Headphone	Headphones	Digital Dynamics
Theta Headphone	Headphones	Digital Dynamics
Iota Smart Watch	Smart Watches	Digital Dynamics
Kappa Smart Watch	Smart Watches	Electron Enterprises

```
10 rows in set (0.00 sec)

mysql>
```

Query 1 x electronicsdb (1) Administration - Data Import/Res...

Limit to 1000 rows

```

1 • USE electronicsdb;
2 SELECT P.ProductName, PC.CategoryName, S.SupplierName
3 FROM products AS P
4 LEFT OUTER JOIN
5 productcategories AS PC
6 ON
7 PC.CategoryID = P.CategoryID
8 LEFT OUTER JOIN
9 suppliers AS S
10 ON
11 S.SupplierID = P.SupplierID
12 LIMIT 10;
13

```

Result Grid

	ProductName	CategoryName	SupplierName
▶	Alpha Smartphone	Smartphones	Tech Treasures
	Beta Smartphone	Smartphones	Tech Treasures
	Gamma Laptop	Laptops	Tech Treasures
	Delta Laptop	Laptops	Byte Builders
	Epsilon Tablet	Tablets	Byte Builders
	Zeta Tablet	Tablets	Byte Builders
	Eta Headphone	Headphones	Digital Dynamics
	Theta Headphone	Headphones	Digital Dynamics
	Iota Smart Watch	Smart Watches	Digital Dynamics
	Kappa Smart Watch	Smart Watches	Electron Enterprises

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');

```

```
Command Prompt - mysql -u root -p

mysql> SELECT P.ProductName
-> FROM products AS P
-> WHERE P.SupplierID IN (
-> SELECT P.SupplierID
-> FROM products AS P
-> WHERE P.ProductName = 'Alpha Smartphone');
+-----+
| ProductName |
+-----+
| Alpha Smartphone |
| Beta Smartphone |
| Gamma Laptop   |
| Rho Monitor    |
+-----+
4 rows in set (0.00 sec)

mysql>
```

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:

The screenshot shows a database client interface with a SQL query editor and a results grid. The query is as follows:

```
1 • USE electronicsDB;
2 • SELECT P.ProductName, P.SupplierID
3 FROM products AS P
4 WHERE P.SupplierID IN (SELECT P.SupplierID
5 FROM products AS P
6 WHERE P.ProductName = 'Alpha Smartphone');
7
8
```

The results grid displays the following data:

ProductName	SupplierID
Alpha Smartphone	50
Beta Smartphone	50
Gamma Laptop	50
Rho Monitor	50

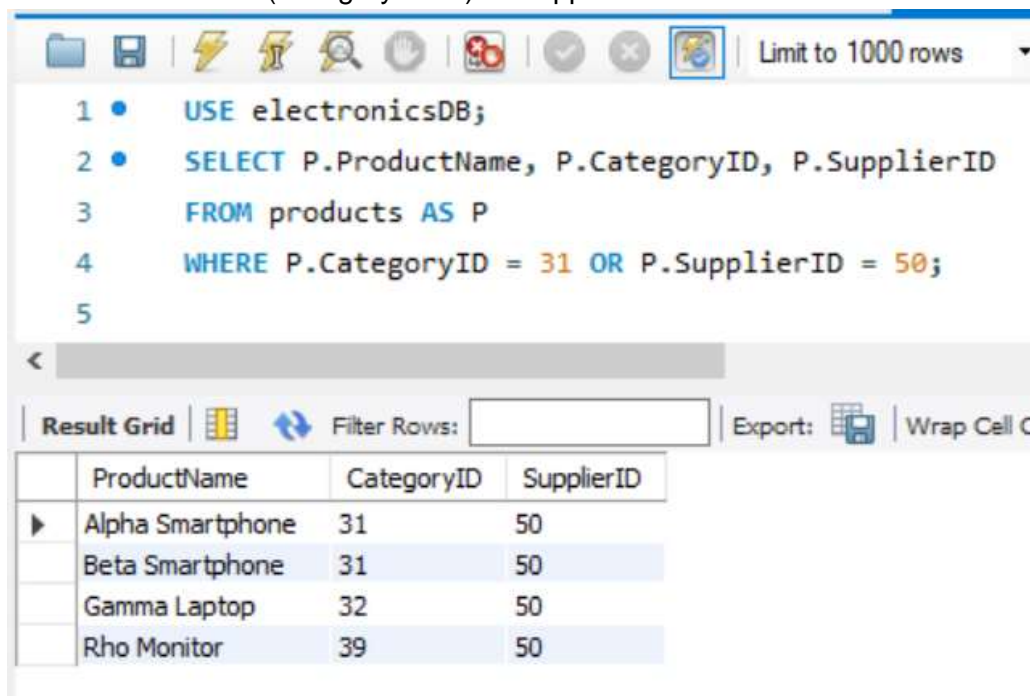
3. [10 pts.] Retrieve all products that are either in the 'Smartphones' category (CategoryID 31) or supplied by SupplierID 50.

```
SELECT P.ProductName, P.CategoryID, P.SupplierID
FROM products AS P
WHERE P.CategoryID = 31 OR P.SupplierID = 50;
```

```
Command Prompt - mysql -u root -p
mysql> USE electronicsdn;
ERROR 1049 (42000): Unknown database 'electronicsdn'
mysql> USE electronicsdb;
Database changed
mysql> SELECT P.ProductName
-> FROM products AS P
-> WHERE P.CategoryID = 31 OR P.SupplierID = 50;
+-----+
| ProductName |
+-----+
| Alpha Smartphone |
| Beta Smartphone |
| Gamma Laptop |
| Rho Monitor |
+-----+
4 rows in set (0.00 sec)

mysql>
```

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



The screenshot shows a database client interface with a toolbar at the top containing icons for file operations, search, and execution. Below the toolbar, the SQL query is entered in a text area:

```
1 • USE electronicsDB;
2 • SELECT P.ProductName, P.CategoryID, P.SupplierID
3 FROM products AS P
4 WHERE P.CategoryID = 31 OR P.SupplierID = 50;
5
```

Below the query, there is a "Result Grid" section with a table displaying the results. The table has four columns: ProductName, CategoryID, and SupplierID. The first row is highlighted with a blue arrow icon in the first column.

	ProductName	CategoryID	SupplierID
▶	Alpha Smartphone	31	50
	Beta Smartphone	31	50
	Gamma Laptop	32	50
	Rho Monitor	39	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;
```

```
C:\> Command Prompt - mysql -u root -p

mysql> 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;

+-----+-----+
| ProductName          | Total Quantity in Stock |
+-----+-----+
| Alpha Smartphone     | 20                       |
| Beta Smartphone       | 15                       |
| Gamma Laptop         | 30                       |
| Delta Laptop         | 25                       |
| Epsilon Tablet       | 40                       |
| Zeta Tablet          | 35                       |
| Eta Headphone        | 50                       |
| Theta Headphone      | 45                       |
| Iota Smart Watch     | 60                       |
| Kappa Smart Watch    | 55                       |
+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

Limit to 1000 rows

```

1 • USE electronicsDB;
2 • SELECT P.ProductName, SUM(SI.QuantityInStock) AS 'Total Quantity in Stock'
3   FROM products AS P JOIN storeinventory AS SI ON SI.ProductID = P.ProductID
4   Group BY P.ProductName
5   LIMIT 10;
6

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

ProductName	Total Quantity in Stock
Alpha Smartphone	20
Beta Smartphone	15
Gamma Laptop	30
Delta Laptop	25
Epsilon Tablet	40
Zeta Tablet	35
Eta Headphone	50
Theta Headphone	45
Iota Smart Watch	60
Kappa Smart Watch	55

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;

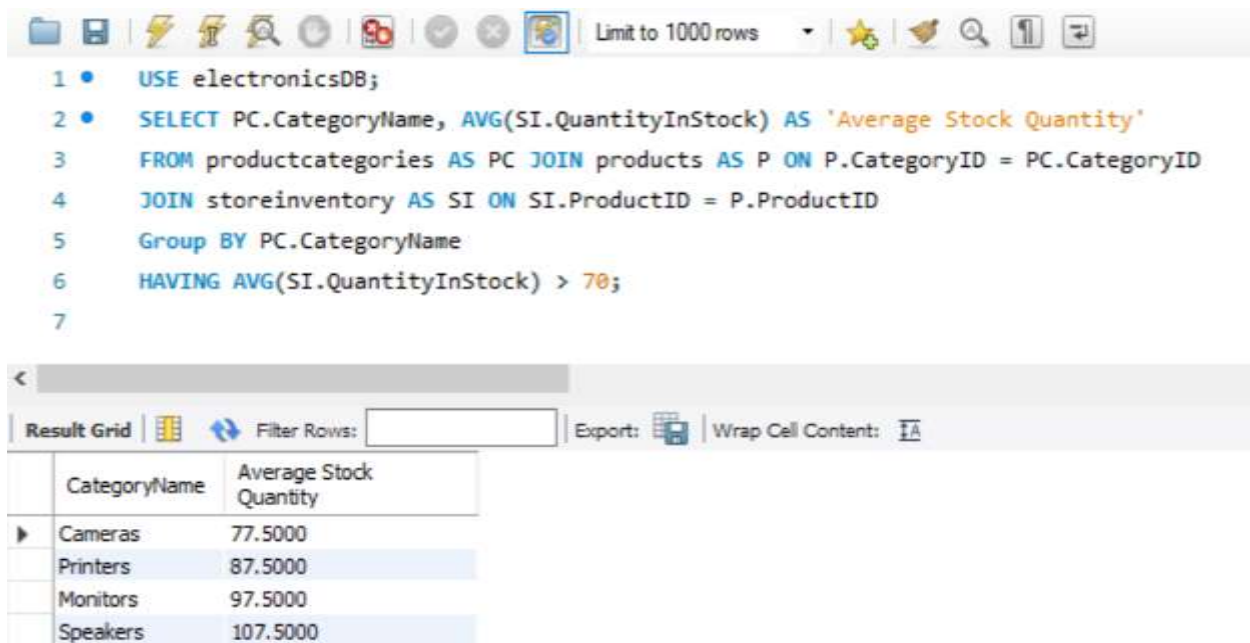
```

```

C:\> Command Prompt - mysql -u root -p
mysql>
mysql> 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;
+-----+-----+
| CategoryName | Average Stock Quantity |
+-----+-----+
| Cameras      | 77.5000                |
| Printers     | 87.5000                |
| Monitors     | 97.5000                |
| Speakers     | 107.5000               |
+-----+-----+
4 rows in set (0.00 sec)

mysql>

```



The screenshot shows a SQL IDE with a query editor and a results grid. The query is as follows:

```

1 • USE electronicsDB;
2 • SELECT PC.CategoryName, AVG(SI.QuantityInStock) AS 'Average Stock Quantity'
3   FROM productcategories AS PC JOIN products AS P ON P.CategoryID = PC.CategoryID
4   JOIN storeinventory AS SI ON SI.ProductID = P.ProductID
5   Group BY PC.CategoryName
6   HAVING AVG(SI.QuantityInStock) > 70;
7

```

The results grid displays the following data:

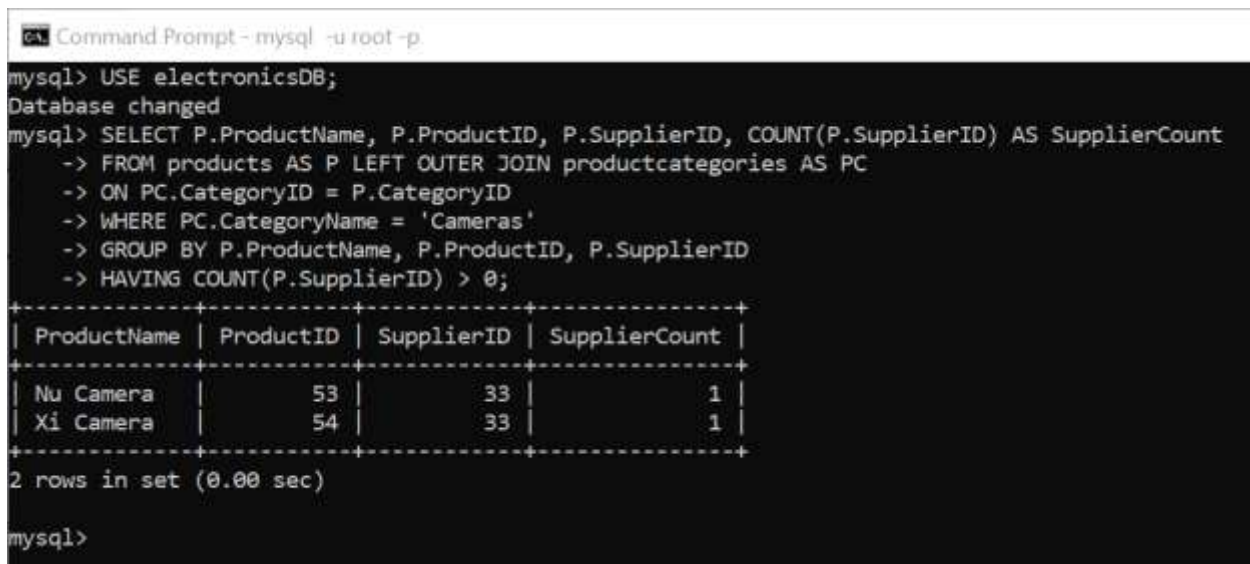
CategoryName	Average Stock Quantity
Cameras	77.5000
Printers	87.5000
Monitors	97.5000
Speakers	107.5000

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;

```



The screenshot shows a MySQL command prompt with the following commands and output:

```

mysql> USE electronicsDB;
Database changed
mysql> 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;
+-----+-----+-----+-----+
| ProductName | ProductID | SupplierID | SupplierCount |
+-----+-----+-----+-----+
| Nu Camera   | 53        | 33         | 1              |
| Xi Camera   | 54        | 33         | 1              |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>

```



```

1 • USE electronicsDB;
2 • SELECT P.ProductName, PC.CategoryName, P.SupplierID, COUNT(P.SupplierID) AS SupplierCount
3 FROM products AS P LEFT OUTER JOIN productcategories AS PC
4 ON PC.CategoryID = P.CategoryID
5 WHERE PC.CategoryName = 'Cameras'
6 GROUP BY P.ProductName, PC.CategoryName, P.SupplierID
7 HAVING COUNT(P.SupplierID) > 0;

```

<	Result Grid	Filter Rows:	Export:	Wrap Cell Content:
	ProductName	CategoryName	SupplierID	SupplierCount
▶	Nu Camera	Cameras	33	1
	Xi Camera	Cameras	33	1

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;

```

```

CA: Command Prompt - mysql -u root -p

mysql> 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;
+-----+-----+-----+-----+
| ProductName | ProductID | CategoryName | SupplierID |
+-----+-----+-----+-----+
| Nu Camera   | 53        | Cameras      | 33         |
| Xi Camera   | 54        | Cameras      | 33         |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>

```



```

24 • SELECT P.ProductName, P.ProductID, PC.CategoryName, P.SupplierID
25 FROM products AS P
26 LEFT JOIN productcategories AS PC ON PC.CategoryID = P.CategoryID
27 WHERE PC.CategoryName = 'Cameras' AND P.SupplierID IS NOT NULL
28
29

```

<div> <div>Result Grid</div> <div>Filter Rows: <input type="text"/></div> <div>Export: </div> <div>Wrap Cell Content: </div> </div>				
	ProductName	ProductID	CategoryName	SupplierID
▶	Nu Camera	53	Cameras	33
	Xi Camera	54	Cameras	33

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);

```

Command Prompt - mysql -u root -p

```
mysql> 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);
```

ProductName	Total Quantity in Stock	CategoryID	Average Stock Quantity
Alpha Smartphone	20	31	17.5000
Gamma Laptop	30	32	27.5000
Epsilon Tablet	40	33	37.5000
Eta Headphone	50	34	47.5000
Iota Smart Watch	60	35	57.5000
Lambda Console	70	36	67.5000
Nu Camera	80	37	77.5000
Omicron Printer	90	38	87.5000
Rho Monitor	100	39	97.5000
Tau Speaker	110	40	107.5000

10 rows in set (0.00 sec)

mysql>

```
2 • SELECT P.ProductName, SUM(SI.QuantityInStock) AS 'Total Quantity in Stock', P.CategoryID,  
3 (SELECT AVG(SI2.QuantityInStock)  
4 FROM storeinventory AS SI2 JOIN products AS P2 ON SI2.ProductID = P2.ProductID  
5 WHERE P2.CategoryID = P.CategoryID) AS 'Average Stock Quantity'  
6 FROM products AS P JOIN storeinventory AS SI ON SI.ProductID = P.ProductID  
7 GROUP BY P.ProductName, P.CategoryID  
8 HAVING SUM(SI.QuantityInStock) >  
9 (SELECT AVG(SI2.QuantityInStock)  
10 FROM storeinventory AS SI2 JOIN products AS P2 ON SI2.ProductID = P2.ProductID  
11 WHERE P2.CategoryID = P.CategoryID);
```

Result Grid	Filter Rows	Export	Wrap Cell Content
ProductName	Total Quantity in Stock	CategoryID	Average Stock Quantity
Alpha Smartphone	20	31	17.5000
Gamma Laptop	30	32	27.5000
Epsilon Tablet	40	33	37.5000
Eta Headphone	50	34	47.5000
Iota Smart Watch	60	35	57.5000
Lambda Console	70	36	67.5000
Nu Camera	80	37	77.5000
Omicron Printer	90	38	87.5000
Rho Monitor	100	39	97.5000
Tau Speaker	110	40	107.5000