**Assignment Chapter 7**

**Question 1:**

Write a SELECT statement that returns the same result set as this SELECT statement, but don’t use a join. Instead, use a subquery in a WHERE clause that uses the IN keyword.

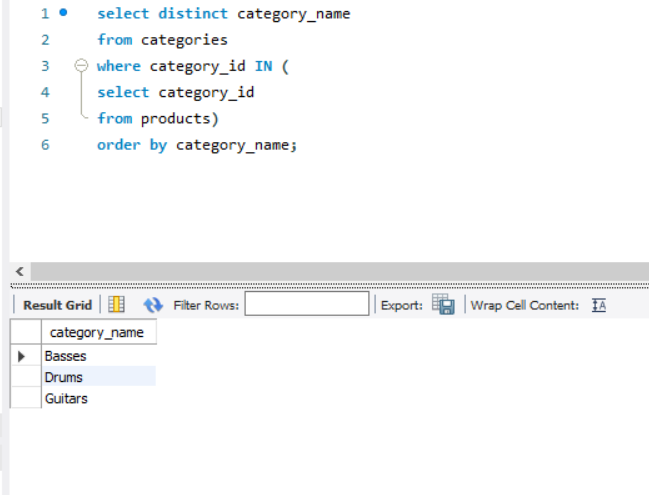
SELECT DISTINCT category\_name

FROM categories c

JOIN products p

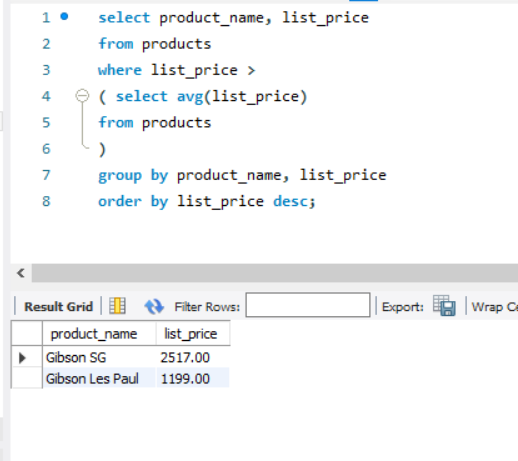
ON c.category\_id = p.category\_id

ORDER BY category\_name



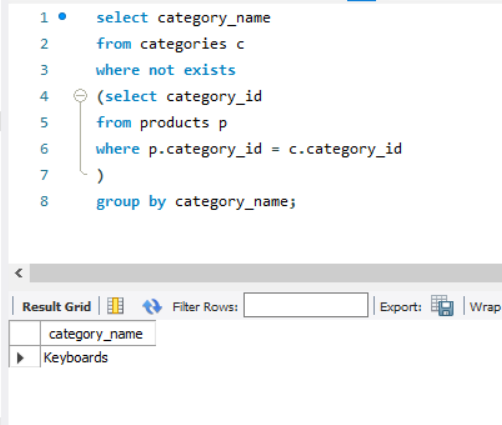
**Question 2:**

Write a SELECT statement that answers this question: Which products have a list price that’s greater than the average list price for all products? Return the product\_name and list\_price columns for each product. Sort the result set by the list\_price column in descending sequence.



**Question 3:**

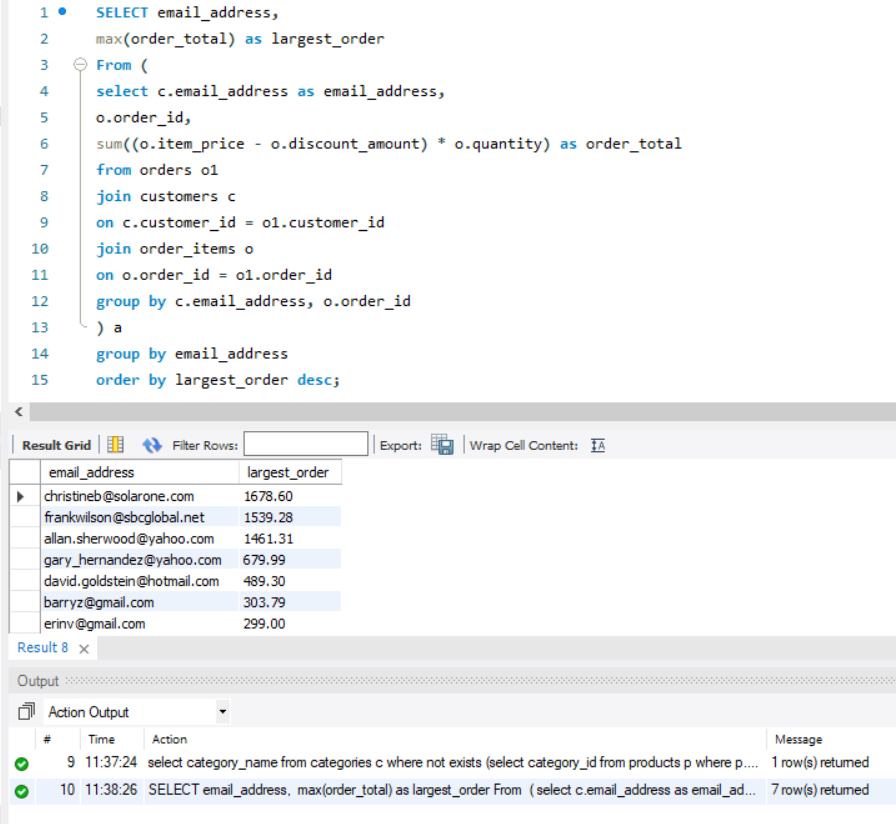
Write a SELECT statement that returns the category\_name column from the Categories table. Return one row for each category that has never been assigned to any product in the Products table. To do that, use a subquery introduced with the NOT EXISTS operator.



**Question 4:**

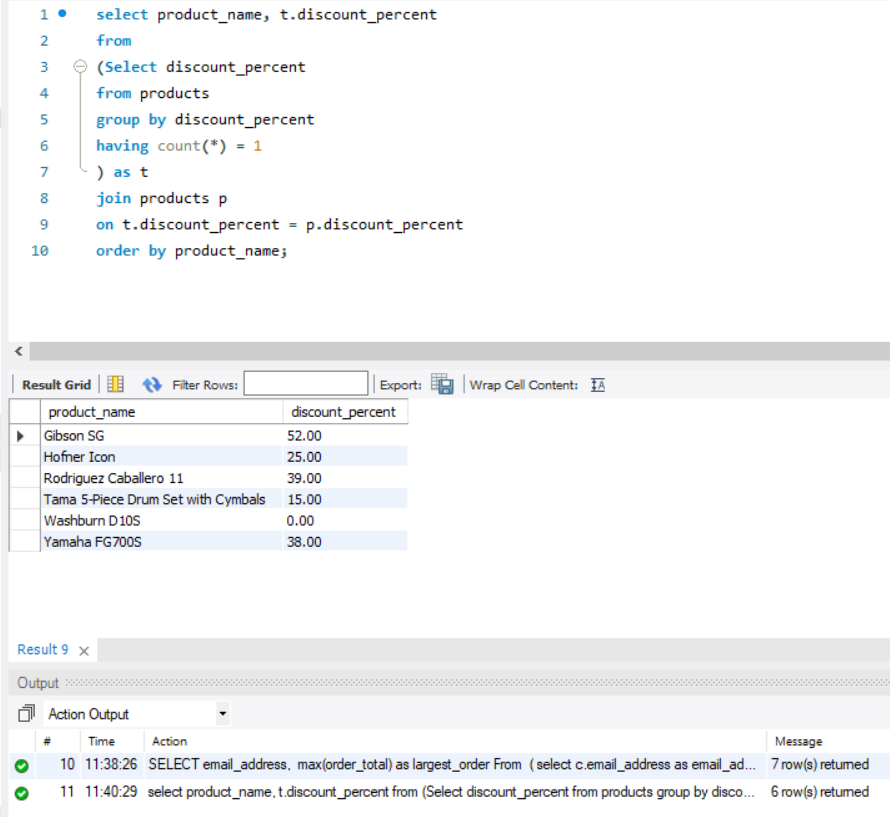
Write a SELECT statement that returns three columns: email\_address, order\_id, and the order total for each customer. To do this, you can group the result set by the email\_address and order\_id columns. In addition, you must calculate the order total from the columns in the Order\_Items table.

Write a second SELECT statement that uses the first SELECT statement in its FROM clause. The main query should return two columns: the customer’s email address and the largest order for that customer. To do this, you can group the result set by the email\_address. Sort the result set by the largest order in descending sequence.



**Question 5:**

Write a SELECT statement that returns the name and discount percent of each product that has a unique discount percent. In other words, don’t include products that have the same discount percent as another product. Sort the result set by the product\_name column.



**Question 6:**

Use a correlated subquery to return one row per customer, representing the customer’s oldest order (the one with the earliest date). Each row should include these three columns: email\_address, order\_id, and order\_date. Sort the result set by the order\_date and order\_id columns.

