SQL SERVER : ASSIGNMENT 4

This assignment focuses on concepts covered in Chapters 5 and 6
Total Points: 85
(Questions 1- 4 is each worth 10 points, and Questions 5- 7 is each worth 15 points)
With reference to the MyGuitarShop database, answer the following questions. You are to upload ONE text file or a script file with scripts for all questions

1. Write a SELECT statement that returns one row for each category that has products with these columns:

   The CategoryName column from the Categories table

   The count of the products in the Products table

   The list price of the most expensive product in the Products table

   Sort the result set so the category with the most products appears first.

2. Write a SELECT statement that returns one row for each customer that has orders with these columns:

   The EmailAddress column from the Customers table

   A count of the number of orders

   The total amount for those orders (*Hint: First, subtract the discount amount from the price. Then, multiply by the quantity.*)

3. Return only those rows where the customer has more than 1 order and so it only counts and totals line items that have an ItemPrice value that's greater than 400.

4. 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 CategoryName
FROM Categories c JOIN Products p
  ON c.CategoryID = p.CategoryID
ORDER BY CategoryName
```

5. 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 ProductName and ListPrice columns for each product.

   Sort the results by the ListPrice column in descending sequence.

6. Write a SELECT statement that returns the CategoryName 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.

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 results by the ProductName column.