SQL SERVER : ASSIGNMENT 3

This assignment focuses on concepts covered in Chapters 3 and 4

Total Points: 70

(Each question is worth 10 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 four columns from the Products table: ProductCode, ProductName, ListPrice, and DiscountPercent. The result set should be sorted by list price in descending sequence.

2. Write a SELECT statement that returns these column names and data from the Products table:

ProductName	The ProductName column
ListPrice	The ListPrice column
DateAdded	The DateAdded column

   Return only the rows with a list price that's greater than 500 and less than 2000. Sort the result set in descending sequence by the DateAdded column.

3. Write a SELECT statement that returns these columns from the Orders table:

OrderID	The OrderID column
OrderDate	The OrderDate column
ShipDate	The ShipDate column

   Return only the rows where the ShipDate column contains a null value.

4. Write a SELECT statement that joins the Customers table to the Addresses table and returns these columns: FirstName, LastName, Line1, City, State, ZipCode.

   Return one row for each customer, but only return addresses that are the shipping address for a customer.

5. Write a SELECT statement that returns the ProductName and ListPrice columns from the Products table. Return one row for each product that has the same list price as another product. Sort the result set by ProductName.

   (Hint: Use a self-join to check that the ProductID columns aren't equal but the ListPrice column is equal.)

6. Write a SELECT statement that returns these two columns:

CategoryName	The CategoryName column from the Categories table
ProductID	The ProductID column from the Products table

   Return one row for each category that has never been used. *(Hint: Use an outer join and only return rows where the ProductID column contains a null value.)*

7. Use the UNION operator to generate a result set consisting of three columns from the Orders table:

ShipStatus	A calculated column that contains a value of SHIPPED or NOT SHIPPED
OrderID	The OrderID column
OrderDate	The OrderDate column

   If the order has a value in the ShipDate column, the ShipStatus column should contain a value of SHIPPED. Otherwise, it should contain a value of NOT SHIPPED.

   Sort the result set by OrderDate.