

Exercise 1

Create a query with the following columns (feel free to borrow your code from Exercise 1 of the PARTITION BY exercises):

- "Name" from the Production.Product table, which can be aliased as "ProductName"
- "ListPrice" from the Production.Product table
- "Name" from the Production.ProductSubcategory table, which can be aliased as "ProductSubcategory"*
- "Name" from the Production.Category table, which can be aliased as "ProductCategory"**

**Join Production.ProductSubcategory to Production.Product on "ProductSubcategoryID"*

***Join Production.ProductCategory to ProductSubcategory on "ProductCategoryID"*

All the tables can be inner joined, and you do not need to apply any criteria.

Exercise 2

Enhance your query from Exercise 1 by adding a derived column called

"Price Rank " that ranks all records in the dataset by ListPrice, in descending order. That is to say, the product with the most expensive price should have a rank of 1, and the product with the least expensive price should have a rank equal to the number of records in the dataset.

Exercise 3

Enhance your query from Exercise 2 by adding a derived column called

"Category Price Rank" that ranks all products by ListPrice – *within each category* - in descending order. In other words, every product within a given category should be ranked relative to other products in the same category.

Exercise 4

Enhance your query from Exercise 3 by adding a derived column called

"Top 5 Price In Category" that returns the string "Yes" if a product has one of the top 5 list prices in its product category, and "No" if it does not. You can try incorporating your logic from Exercise 3 into a CASE statement to make this work.