Exercise 1

Write a query that displays the three most expensive orders, **per vendor ID**, from the Purchasing.PurchaseOrderHeader table. There should ONLY be three records per Vendor ID, even if some of the total amounts due are identical. "Most expensive" is defined by the amount in the "TotalDue" field.

Include the following fields in your output:

- PurchaseOrderID
- VendorID
- OrderDate
- TaxAmt
- Freight
- TotalDue

Hints:

- You will first need to define a field that assigns a unique rank to every purchase order, within each group of like vendor IDs.
- You'll probably want to use a Window Function with PARTITION BY and ORDER BY to do this.
- The last step will be to apply the appropriate criteria to the field you created with your Window Function.

Exercise 2

Modify your query from the first problem, such that the top three purchase order **amounts** are returned, regardless of how many records are returned per Vendor Id.

In other words, if there are multiple orders with the same total due amount, all should be returned as long as the total due amount for these orders is one of the top three.

Ultimately, you should see three distinct total due amounts (i.e., the top three) for each group of like Vendor Ids. However, there could be multiple records for each of these amounts.

Hint: Think carefully about how the different ranking functions (ROW_NUMBER, RANK, and DENSE_RANK) work, and which one might be best suited to help you here.