Exercise 1: City List

In this exercise, you will create a query to return a City list of both Customers and Suppliers.

Task 1: Customer Cities

1. Write a query to return the City column from dbo.Customers.
2. Order the Cities in ascending order.
3. Execute the query by hitting the <F5> key.
4. The query should return 91 rows, make a note of how many times London appears in the result set.

Task 2: Supplier Cities

1. Write a query to return the City column from dbo.Suppliers.
2. Order the Cities in ascending order.
3. Execute the query by hitting the <F5> key.
4. The query should return 29 rows, make a note of how many times London appears in the result set.

Task 3: Complete list of Cities

1. Create a new query and save it with a filename of "AllCities.sql".
2. Using the queries from Tasks 1 and 2, create a query to combine the two data sets with UNION ALL. Please note you can only have one ORDER BY clause when combining the two queries.
3. Execute the query by hitting the <F5> key.
4. Test the query:
    - How many rows were returned?
    - Are there duplicate rows returned?
5. Alter the query to now use UNION.
6. Test the query:
    - How many rows were returned?
    - Are there duplicate rows returned?

Exercise 2: City Overlap

In this exercise you will find cities where there are both Customers and Suppliers using the intersection Method

Task 1: Create a query showing the Overlap

1. Modify the existing “AllCities.sql” to show use cities where we have both Customers and Suppliers.
2. Execute the query by hitting the <F5> key.
3. Test the query. This should return 5 rows.

Exercise 3: Country Disjoint

In this exercise you will find Countries in the Customers table and not in the Suppliers table and vice versa.

Task 1: All Countries

1. Create a new query and save it with a filename of “Countries.sql”.
2. Write a query to return the Country column from the dbo.Customers table.
3. Use the UNION function to select the Country column from both dbo.Customers and dbo.Suppliers.
4. Order by Country ascending.
5. Execute the query by hitting the <F5> key.
6. Test the query. This query should return 25 rows.

Task 2: Countries that contain both Customers and Suppliers

1. Edit the existing “Countries.sql” query.
2. Modify the query to use INTERSECT instead of UNION.
3. Execute the query by hitting the <F5> key.
4. Test the query. This should return 12 rows.

Task 3: Countries with Customers and not Suppliers

1. Edit the existing “Countries.sql” query.
2. Modify the query to use EXCEPT instead of UNION
3. Execute the query by hitting the <F5> key.
4. Test the query:
   a. How many rows were returned?
   b. 25 countries minus 12 intersecting countries is?

Task 4: Countries with Suppliers and not Customers.

1. Edit the existing “Countries.sql” query.
2. Modify the query by swapping the running order of the SELECT Statements. SELECT supplier first.
3. Test the query:
   a. How many rows are returned?

b. Does the number from Task 3.4.a + 12 = 25 records?