**Section 10**

**UNION**

Combine the result of two or more queries

SYNTAX:

Select col\_names from table

Union

Select col\_name from table

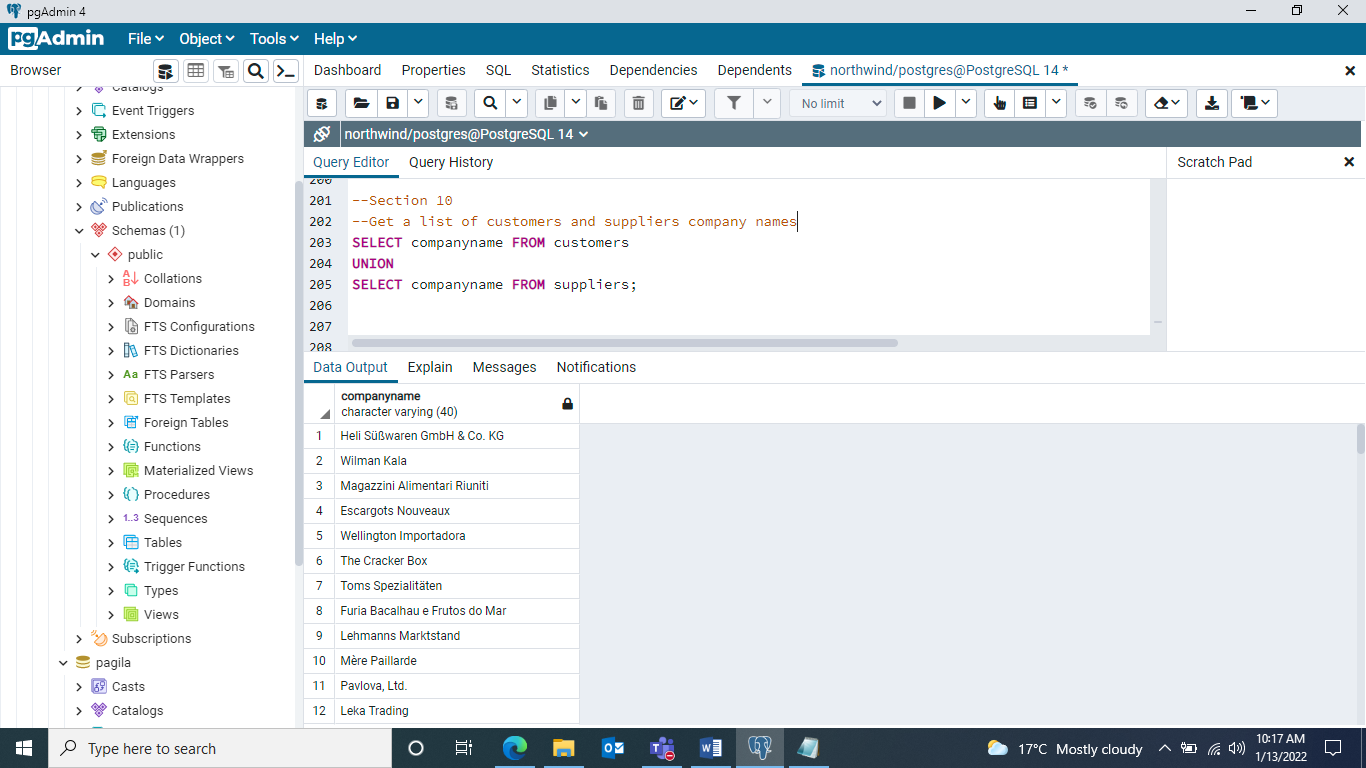
* Must have same number of columns, column types must line up
* Removes duplicates

**Practice: 1)**Get a list of customers and suppliers company names

SELECT companyname FROM customers

UNION

SELECT companyname FROM suppliers;



UNION ALL –Grab all values (get duplicate records)

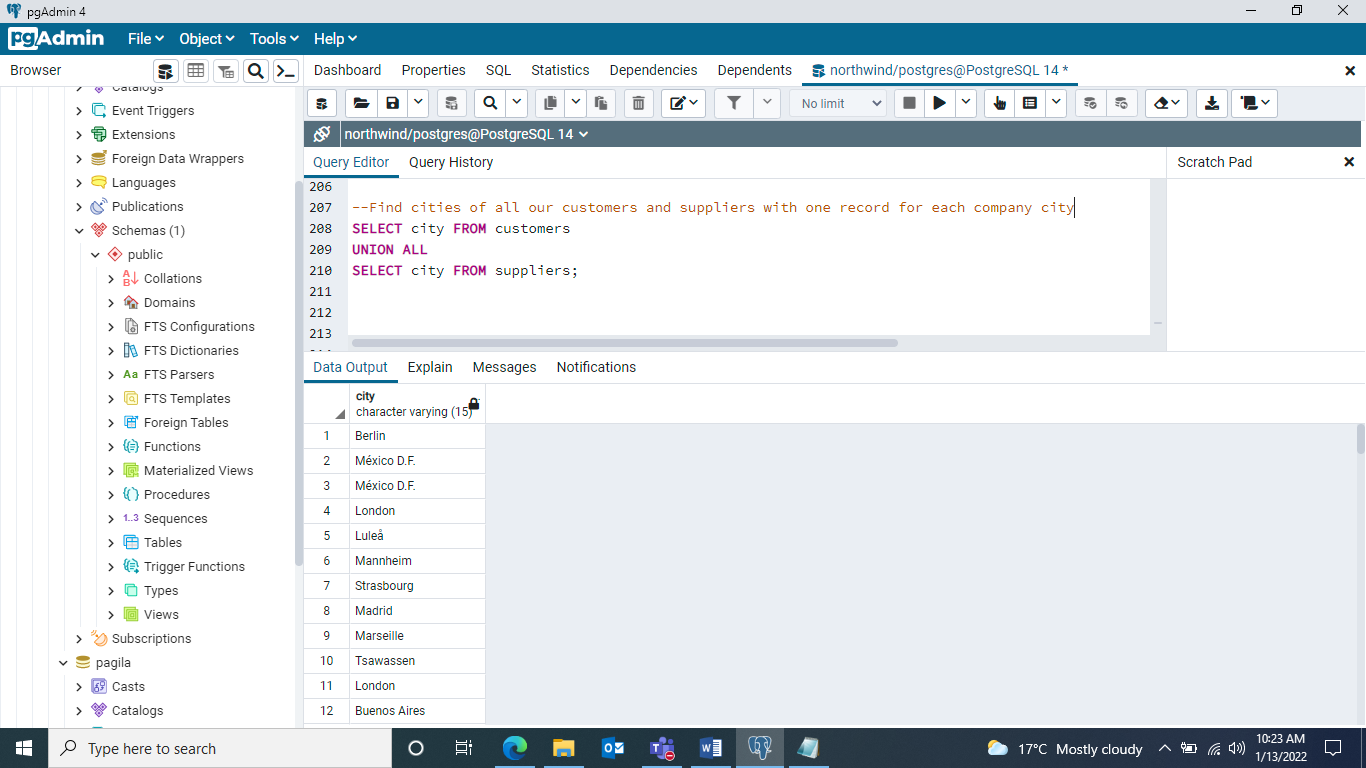
Select col\_names from table

UNION ALL

Select col\_name from table

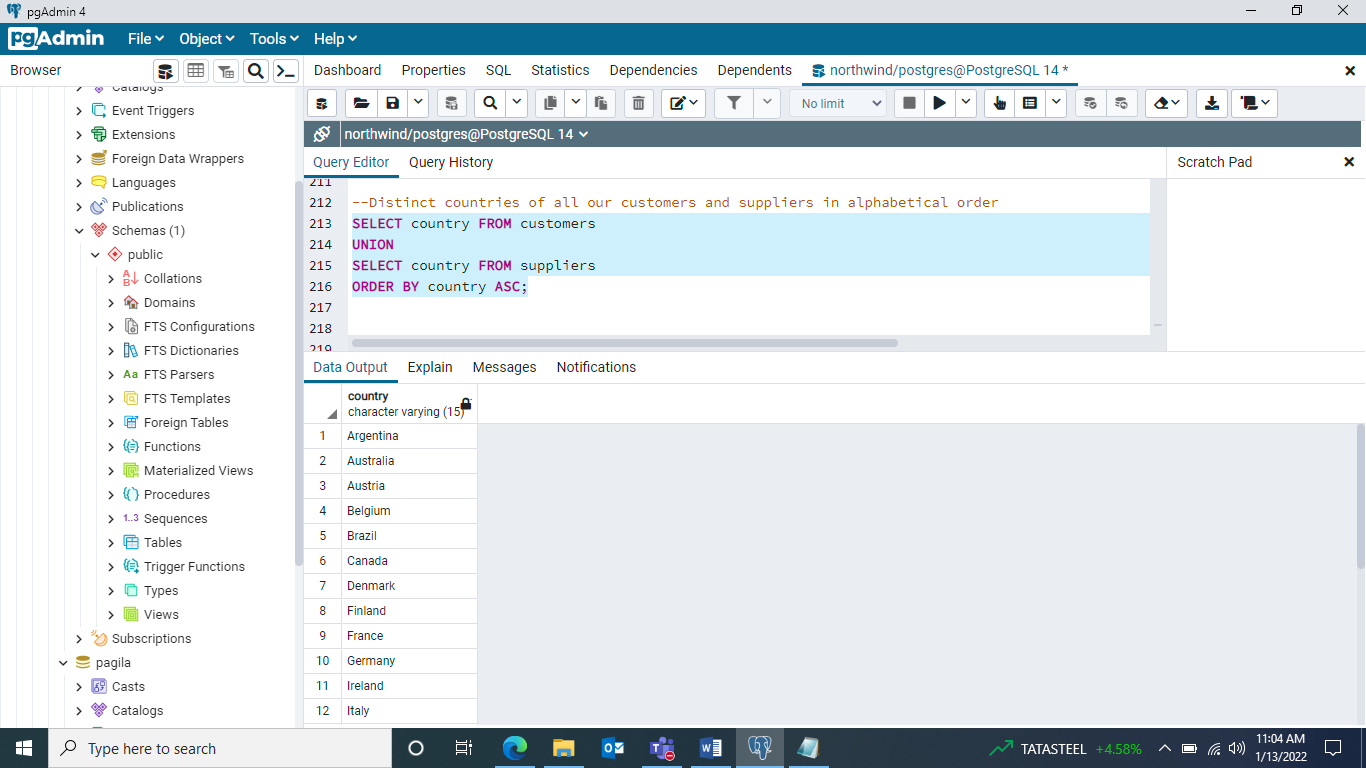
1. Find cities of all our customers and suppliers with one record for each company city

SELECT city FROM customers UNION ALL SELECT city FROM suppliers;



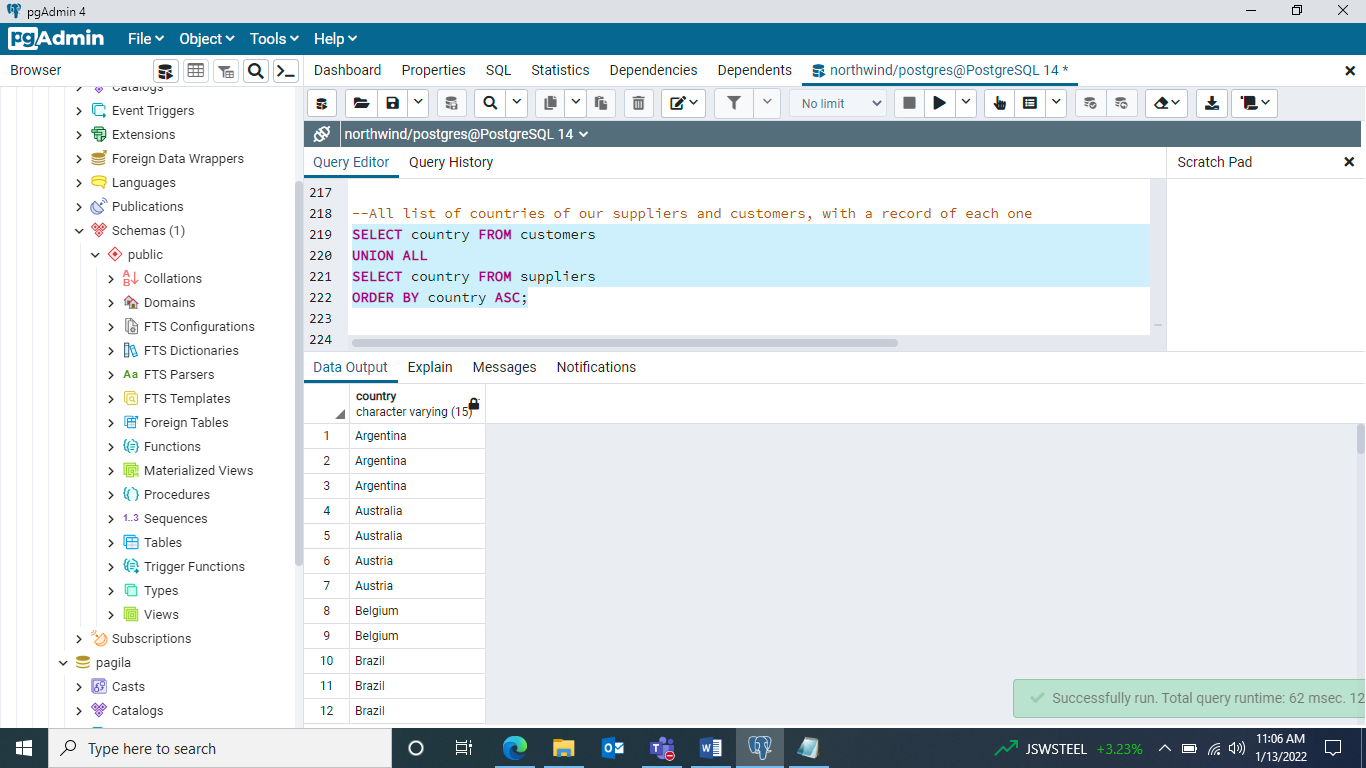
1. Distinct countries of all our customers and suppliers in alphabetical order

SELECT country FROM customers UNION SELECT country FROM suppliers ORDER BY country ASC;



1. All list of countries of our suppliers and customers, with a record of each one

SELECT country FROM customers UNION ALL SELECT country FROM suppliers ORDER BY country ASC;



**INTERSECT**

**Combining items from both sets / find items that are both queries**

Select col\_names from table

Intersect

Select col\_name from table

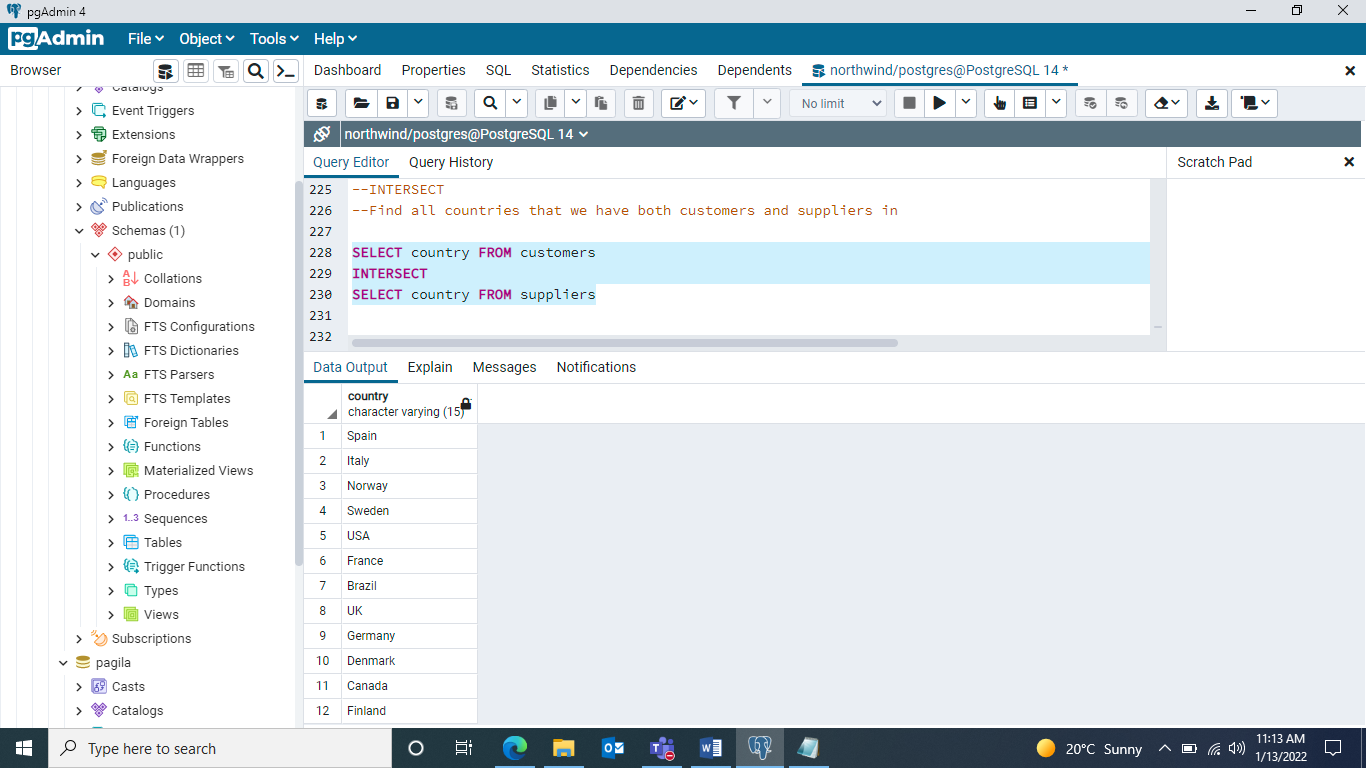
* Must have same number of columns, column types must line up
* Removes duplicates

1. Find all countries that we have both customers and suppliers in

SELECT country FROM customers

INTERSECT

SELECT country FROM suppliers



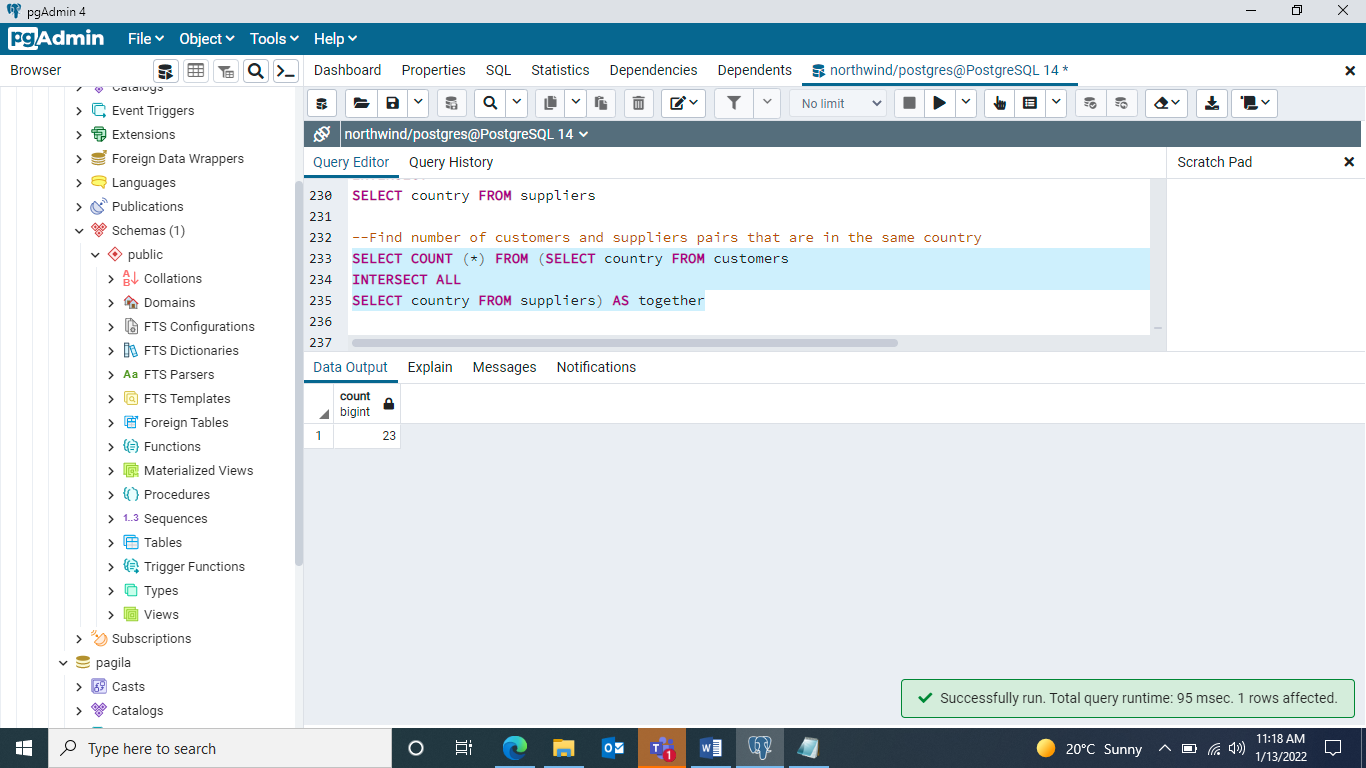
**INTERSECT ALL 🡪 it does not remove duplicates**

Select col\_names from table

Intersect all

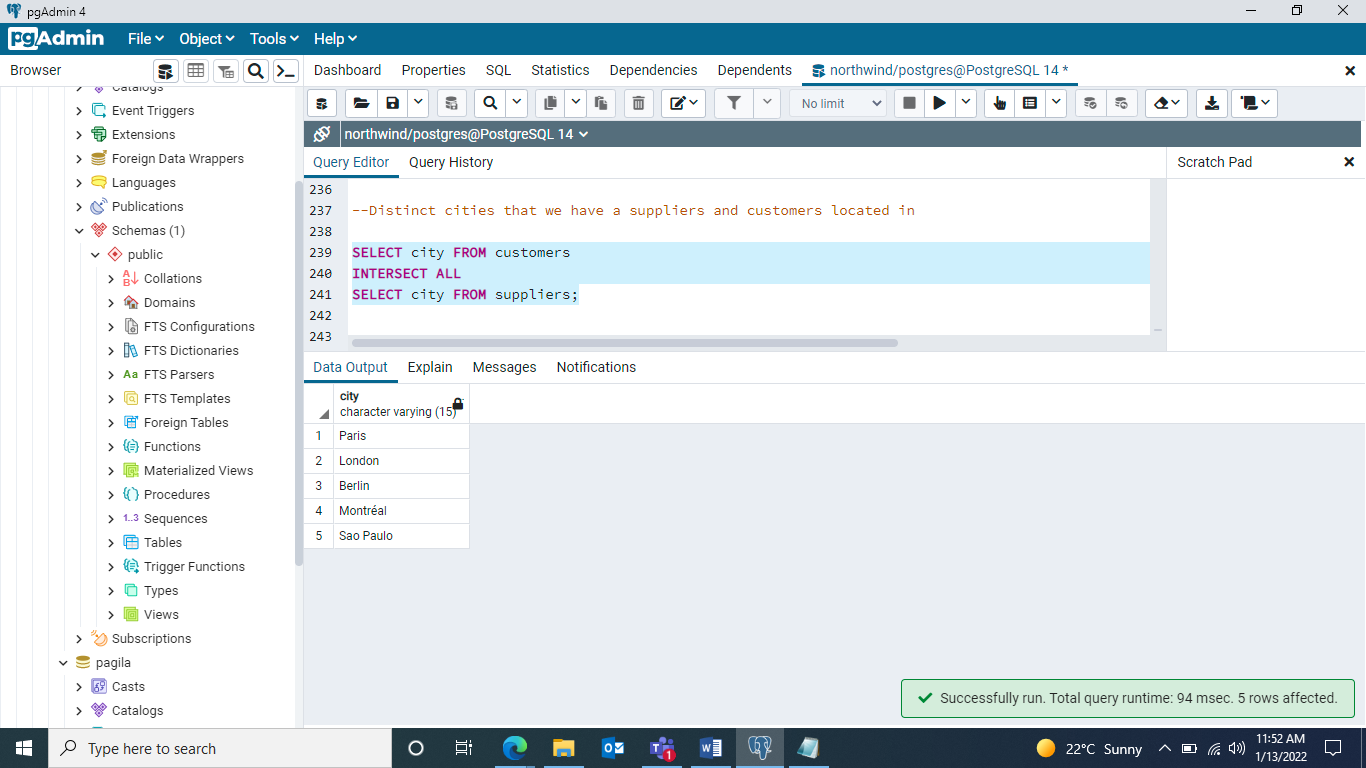
Select col\_name from table

1. Find number of customers and suppliers pairs that are in the same country



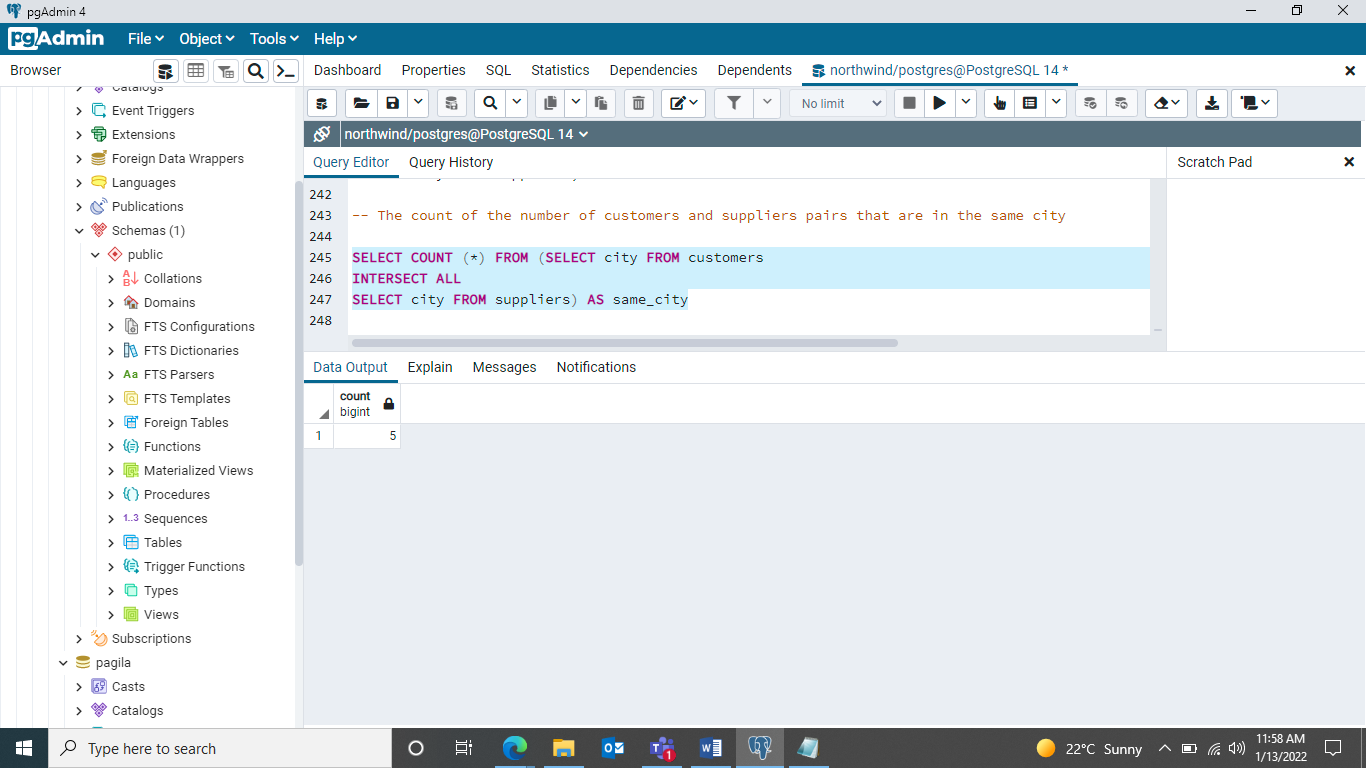
1. Distinct cities that we have a suppliers and customers located in

SELECT city FROM customers INTERSECT ALL SELECT city FROM suppliers;



1. The count of the number of customers and suppliers pairs that are in the same city

SELECT COUNT (\*) FROM (SELECT city FROM customers INTERSECT ALL SELECT city FROM suppliers) AS same\_city;



--**EXCEPT**

Find items that are the first query but not the second

Select col\_names from table

**EXCEPT**

Select col\_name from table

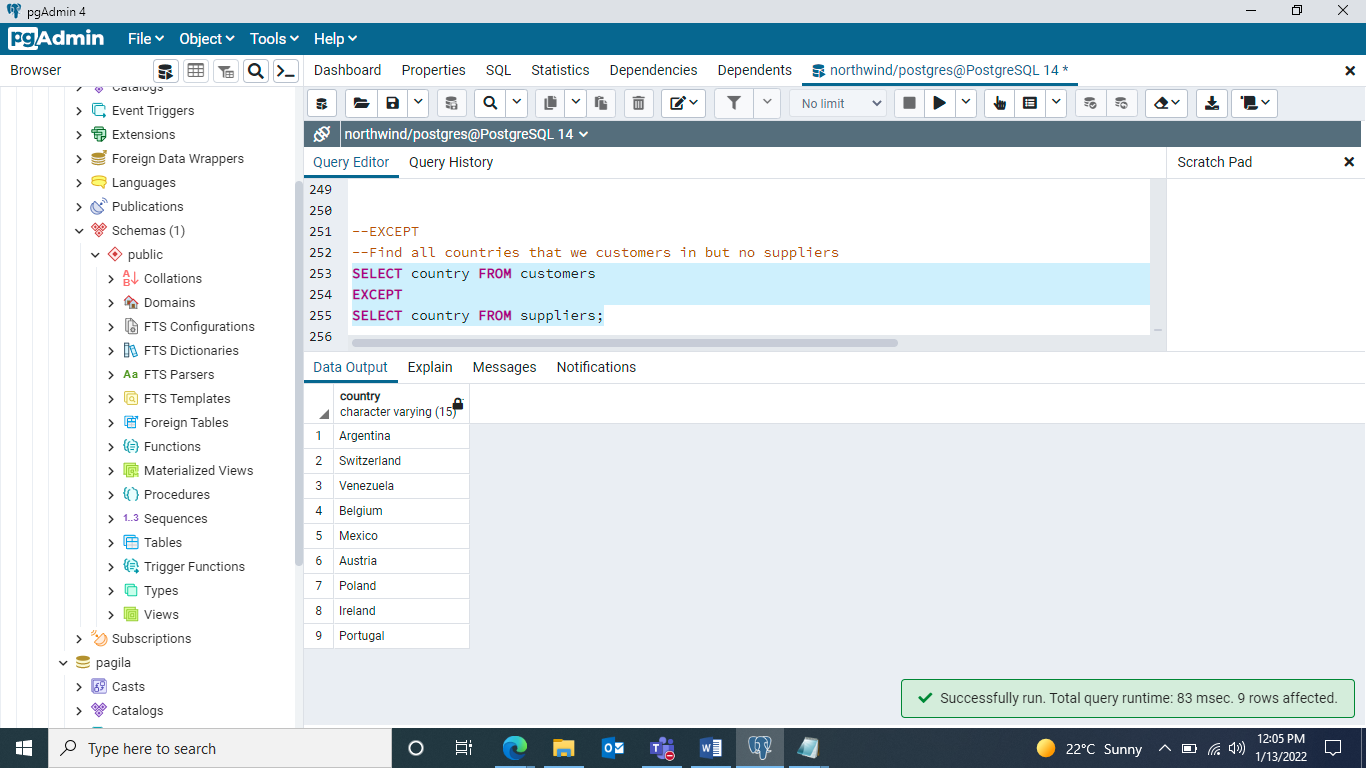
* Must have same number of columns, column types must line up
* Removes duplicates

1. Find all countries that we customers in but no suppliers

SELECT country FROM customers

EXCEPT

SELECT country FROM suppliers;



**EXCEPT** **ALL🡪GRAB ALL VALUES**

Select col\_names from table

**EXCEPT** **ALL**

Select col\_name from table

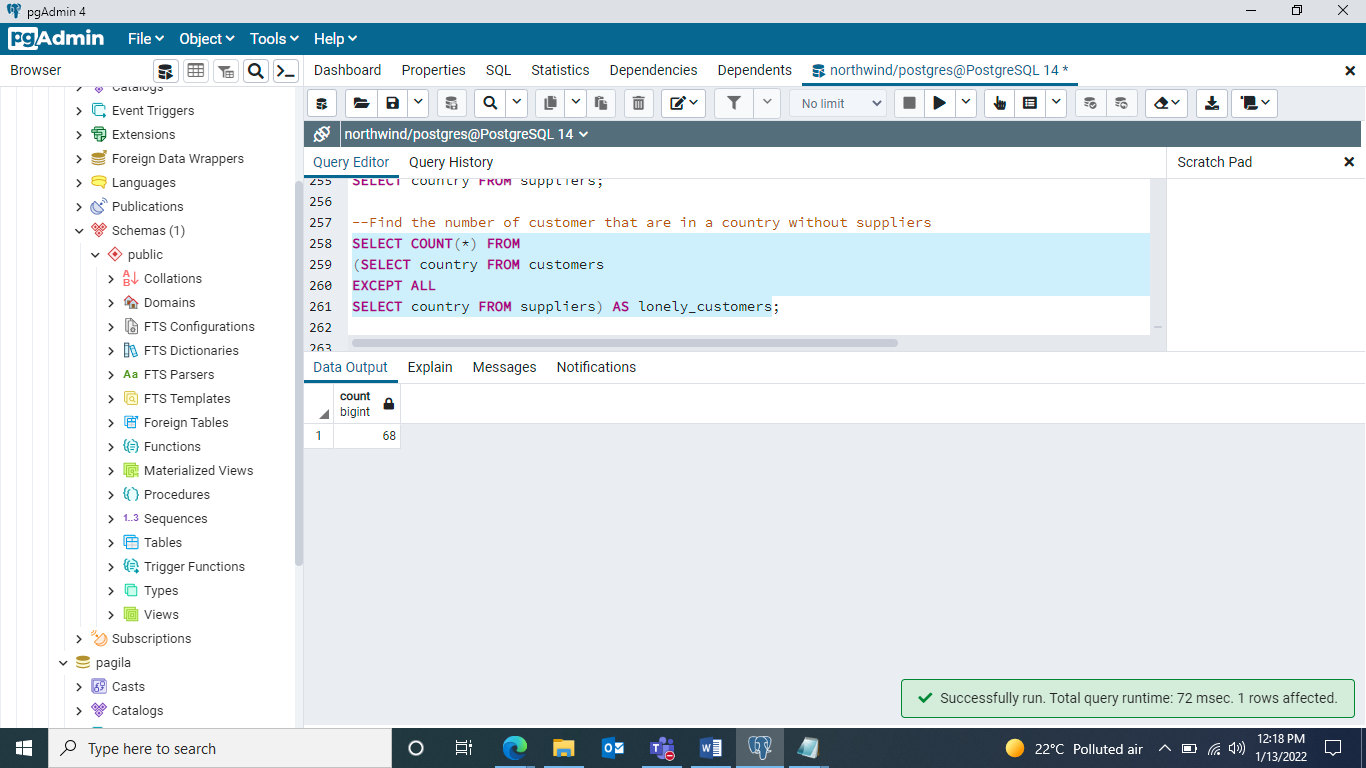
1. Find the number of customer that are in a country without suppliers

SELECT COUNT(\*) FROM

(SELECT country FROM customers

EXCEPT ALL

SELECT country FROM suppliers) AS lonely\_customers;

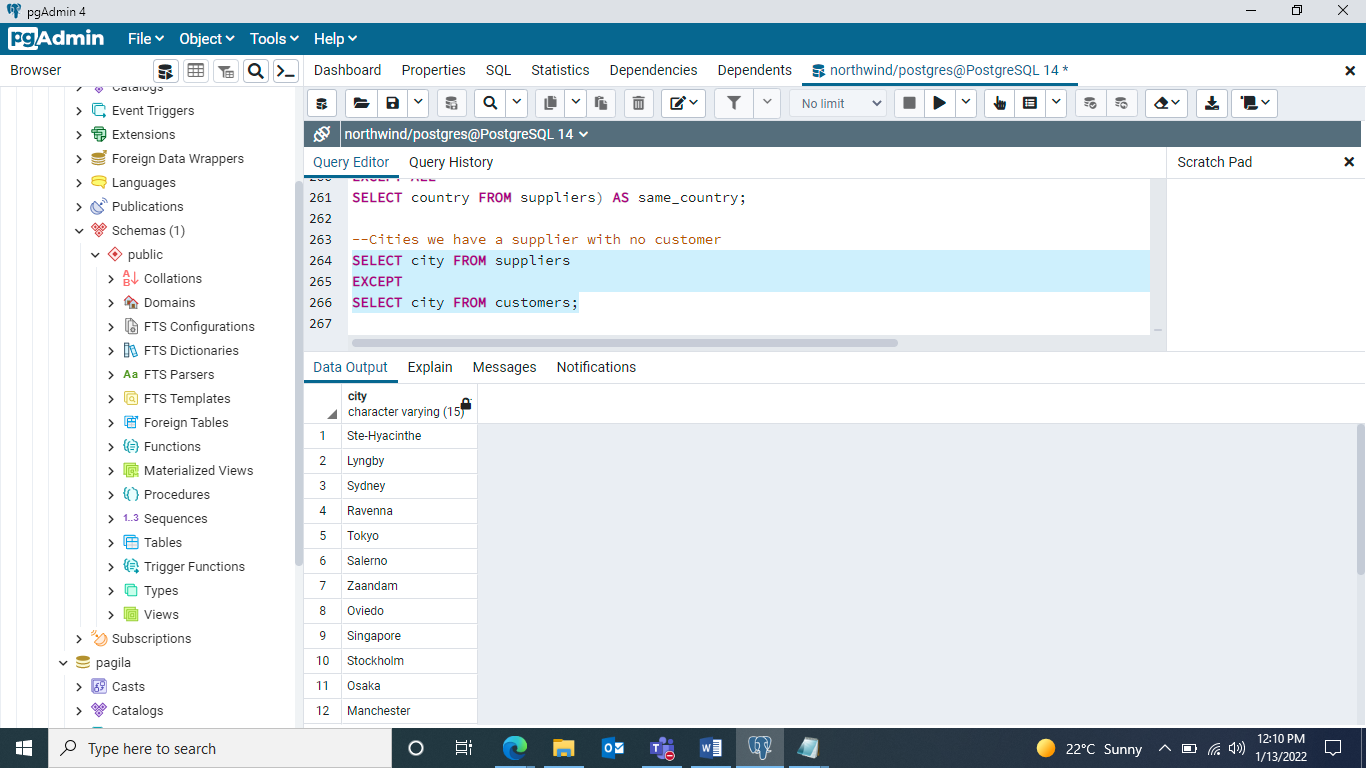


1. Cities we have a supplier with no customer

SELECT city FROM suppliers

EXCEPT

SELECT city FROM customers;



1. How many customers do we have in cities without suppliers

SELECT COUNT(\*) FROM

(SELECT city FROM customers

EXCEPT ALL

SELECT city FROM suppliers) AS lonely\_customers;

