1)      Alter Table:

* Add a new column linkedin\_profile to employees table to store LinkedIn URLs as varchar.

Alter table employees Add column linkedin\_profile varchar

* Change the linkedin\_profile column data type from VARCHAR to TEXT.

Alter table employees

Alter column linkedin\_profile Set Data type Text;

* Add unique, not null constraint to linkedin\_profile

UPDATE employees

SET linkedin\_profile = 'placeholder\_' || "employeeID"

WHERE linkedin\_profile IS NULL;

Above query was needed as when the column is inserted it automatically get s null values so imposing unique & not null constraint is not possible. So we change the column values of linkedin\_profile then perform other unique & not null constraint queries

Alter table employees

Alter Column linkedin\_profile SET NOT NULL;

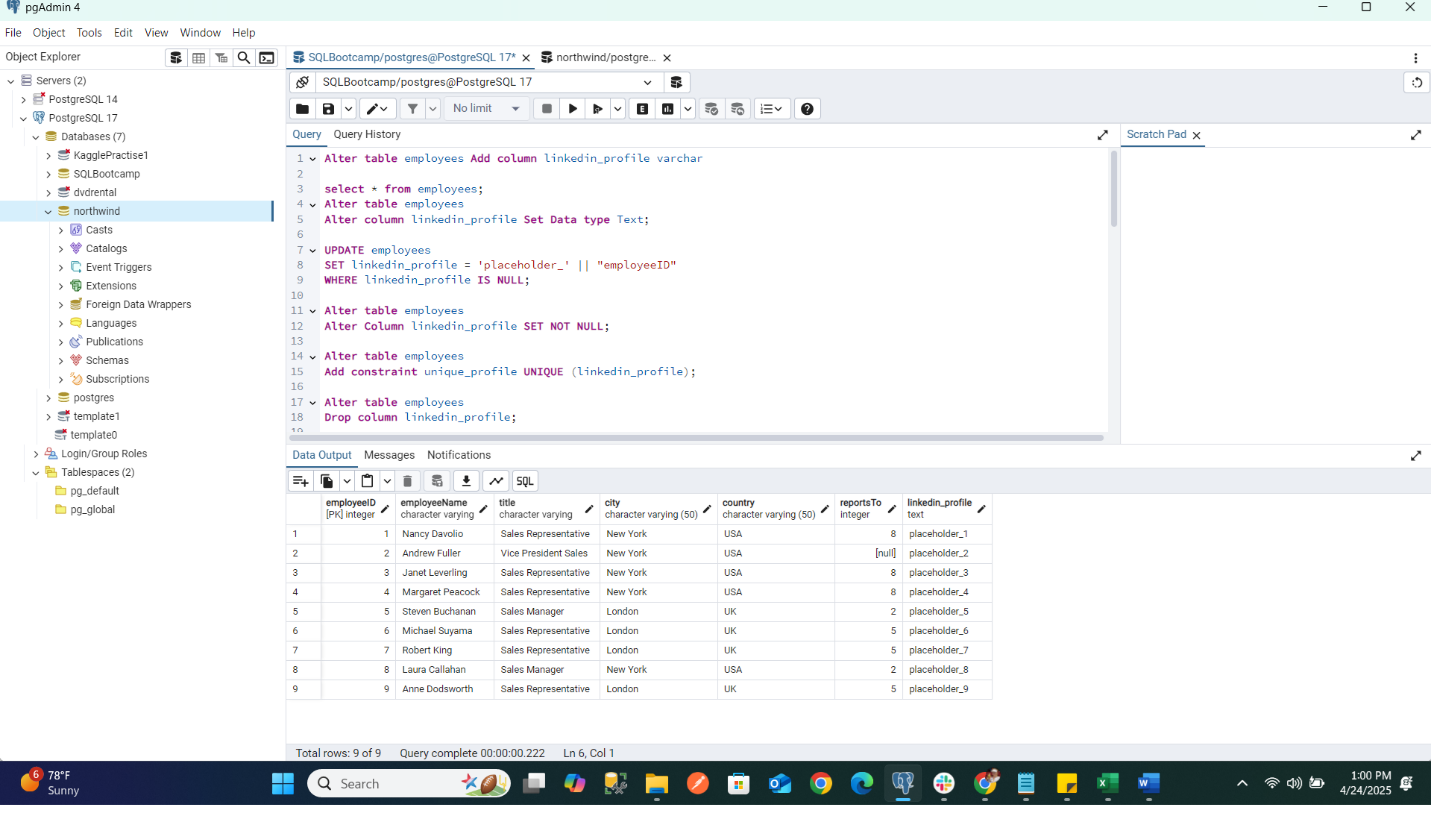
Alter table employees

Add constraint unique\_profile UNIQUE (linkedin\_profile)

* Drop column linkedin\_profile

Alter table employees

Drop column linkedin\_profile;



2)      Querying (Select)

* Retrieve the employee name and title of all employees

Select "employeeName", title from employees;

* Find all unique unit prices of products

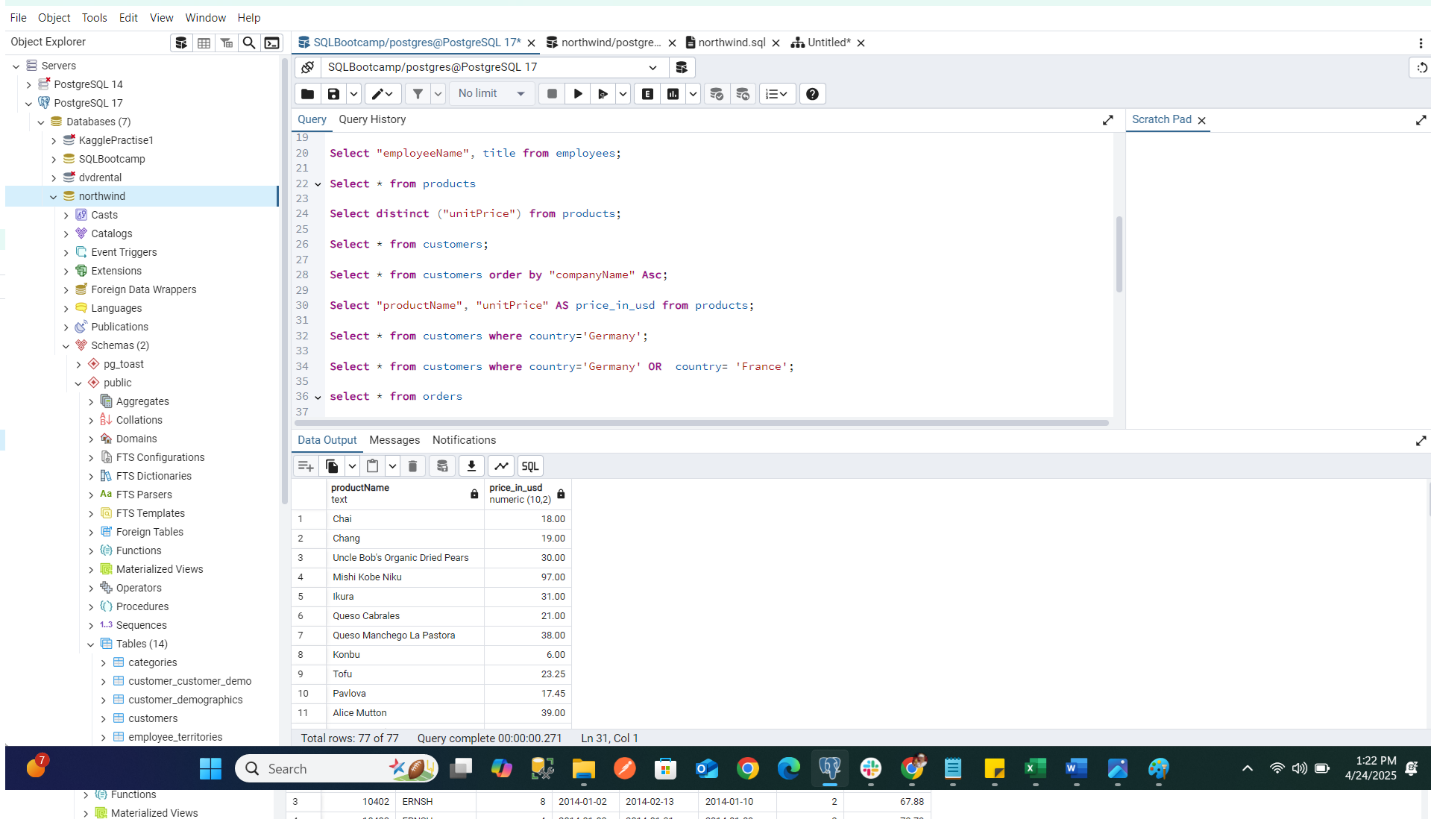
Select distinct ("unitPrice") from products;

* List all customers sorted by company name in ascending order

Select \* from customers order by “companyName”;

* Display product name and unit price, but rename the unit\_price column as price\_in\_usd

Select "productName", "unitPrice" AS price\_in\_usd from products;



3)      Filtering

* Get all customers from Germany.

Select \* from customers where country='Germany';

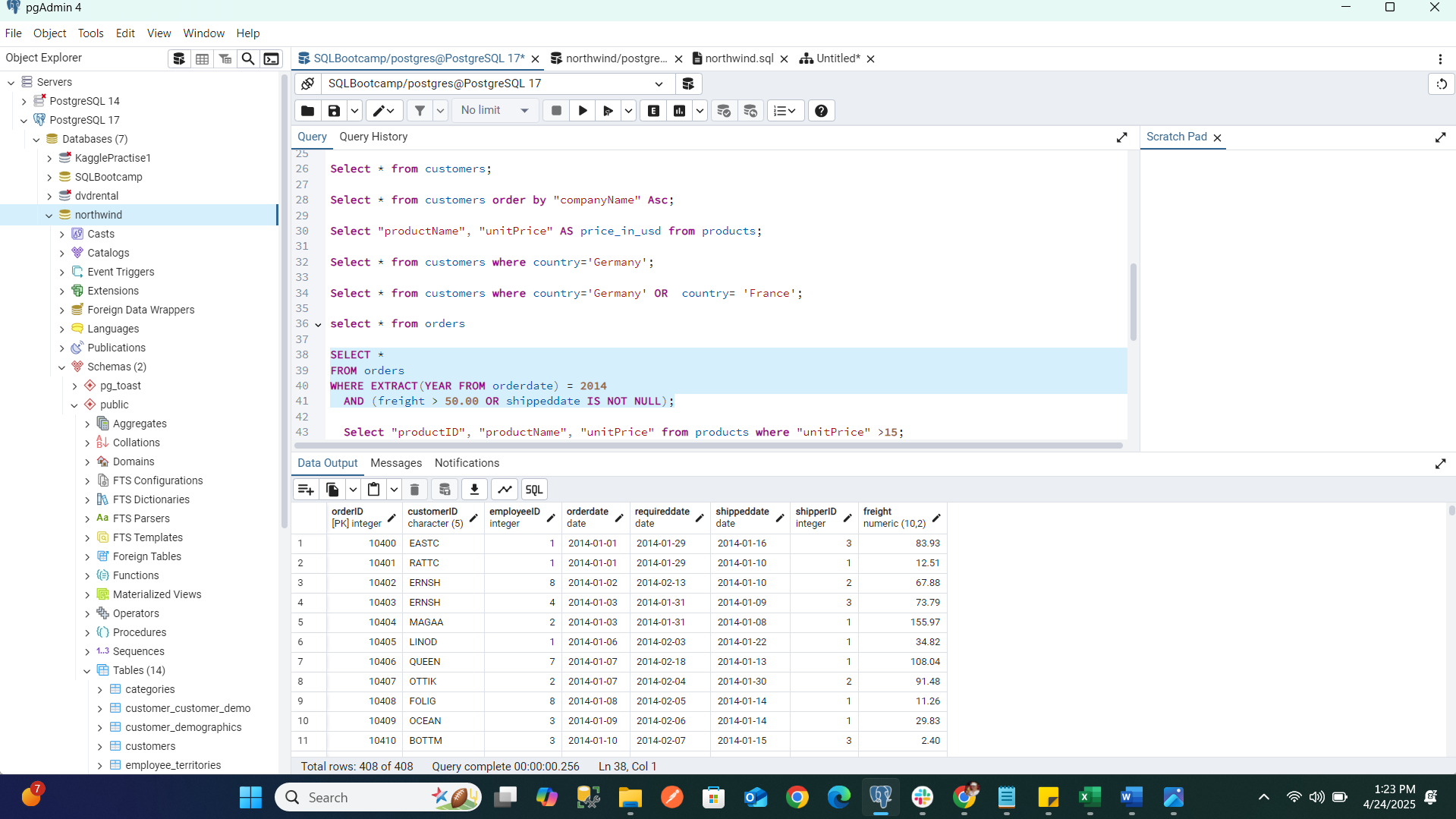
* Find all customers from France or Spain

Select \* from customers where country='Germany' OR country= 'France';

* Retrieve all orders placed in 2014(based on order\_date), and either have freight greater than 50 or the shipped date available (i.e., non-NULL)  (Hint: EXTRACT(YEAR FROM order\_date))

Select \* FROM orders where Extract(YEAR FROM orderdate) = 2014

AND (freight > 50.00 OR shippeddate IS NOT NULL);



4)      Filtering

* Retrieve the product\_id, product\_name, and unit\_price of products where the unit\_price is greater than 15.

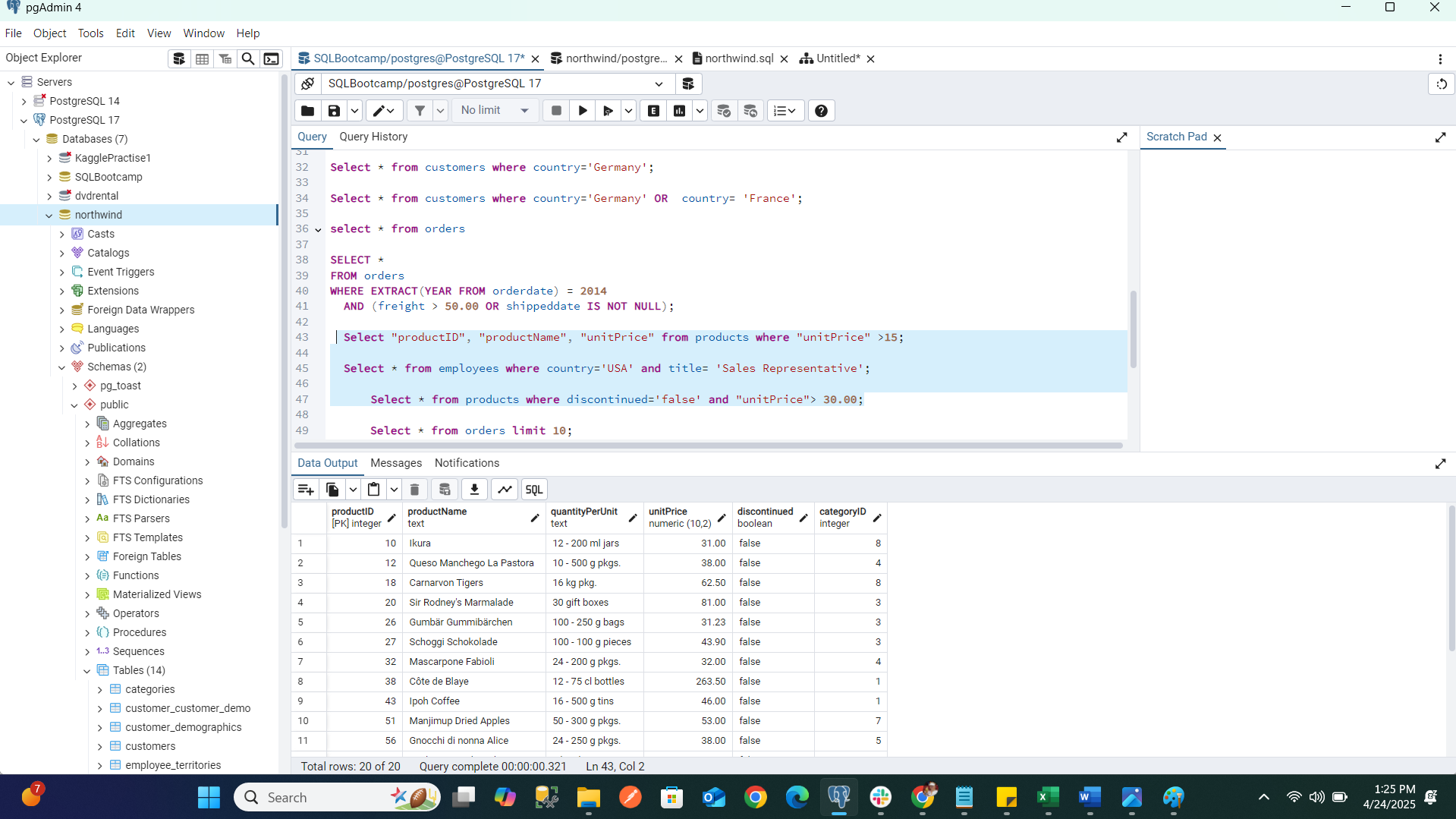
Select "productID", "productName", "unitPrice" from products where "unitPrice" >15;

List all employees who are located in the USA and have the title "Sales Representative".

Select \* from employees where country='USA' and title= 'Sales Representative';

* Retrieve all products that are not discontinued and priced greater than 30.

Select \* from products where discontinued='false' and "unitPrice"> 30.00;



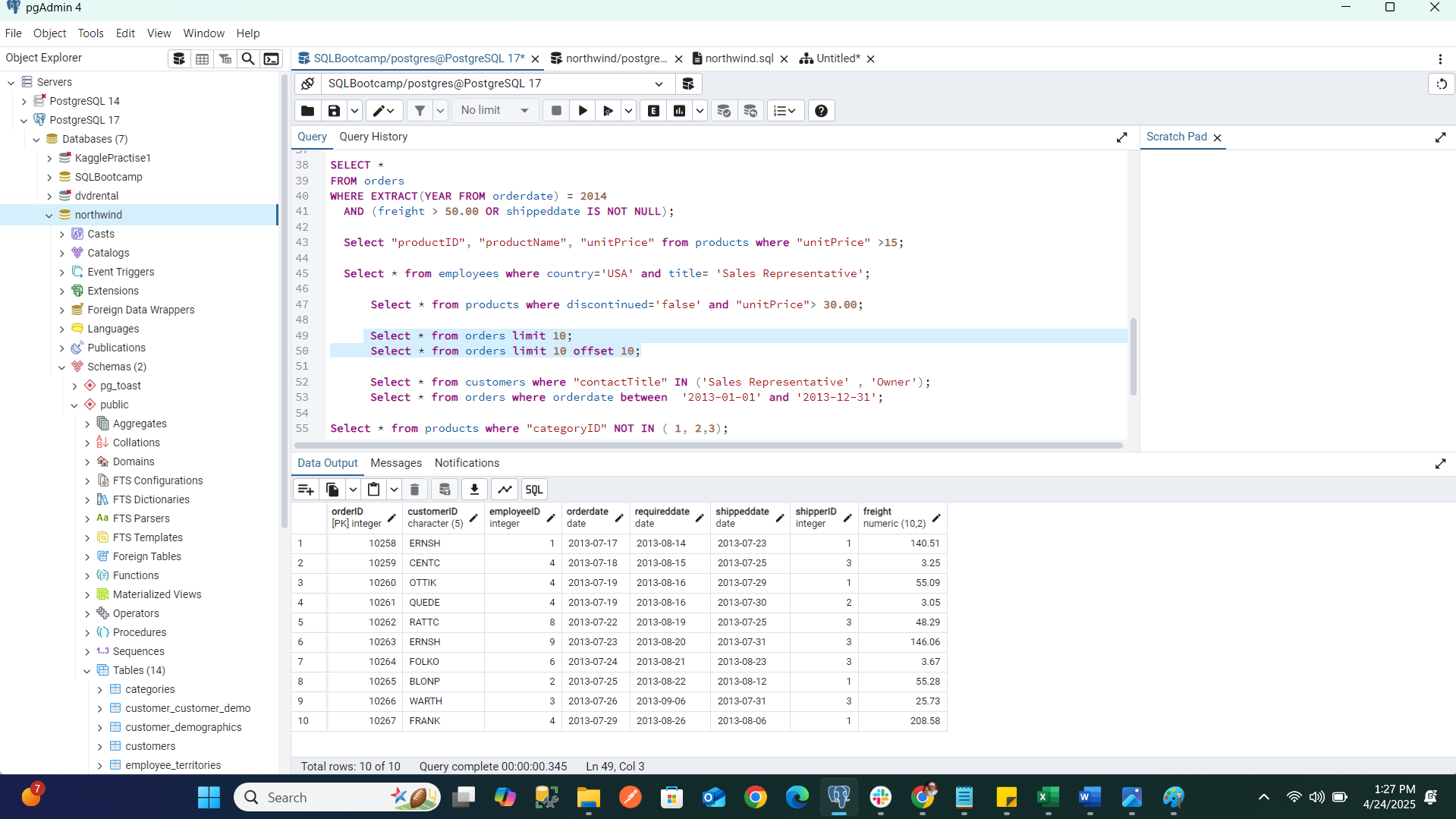
5)      LIMIT/FETCH

* Retrieve the first 10 orders from the orders table.

Select \* from orders limit 10;

* Retrieve orders starting from the 11th order, fetching 10 rows (i.e., fetch rows 11-20).

Select \* from orders limit 10 offset 10;



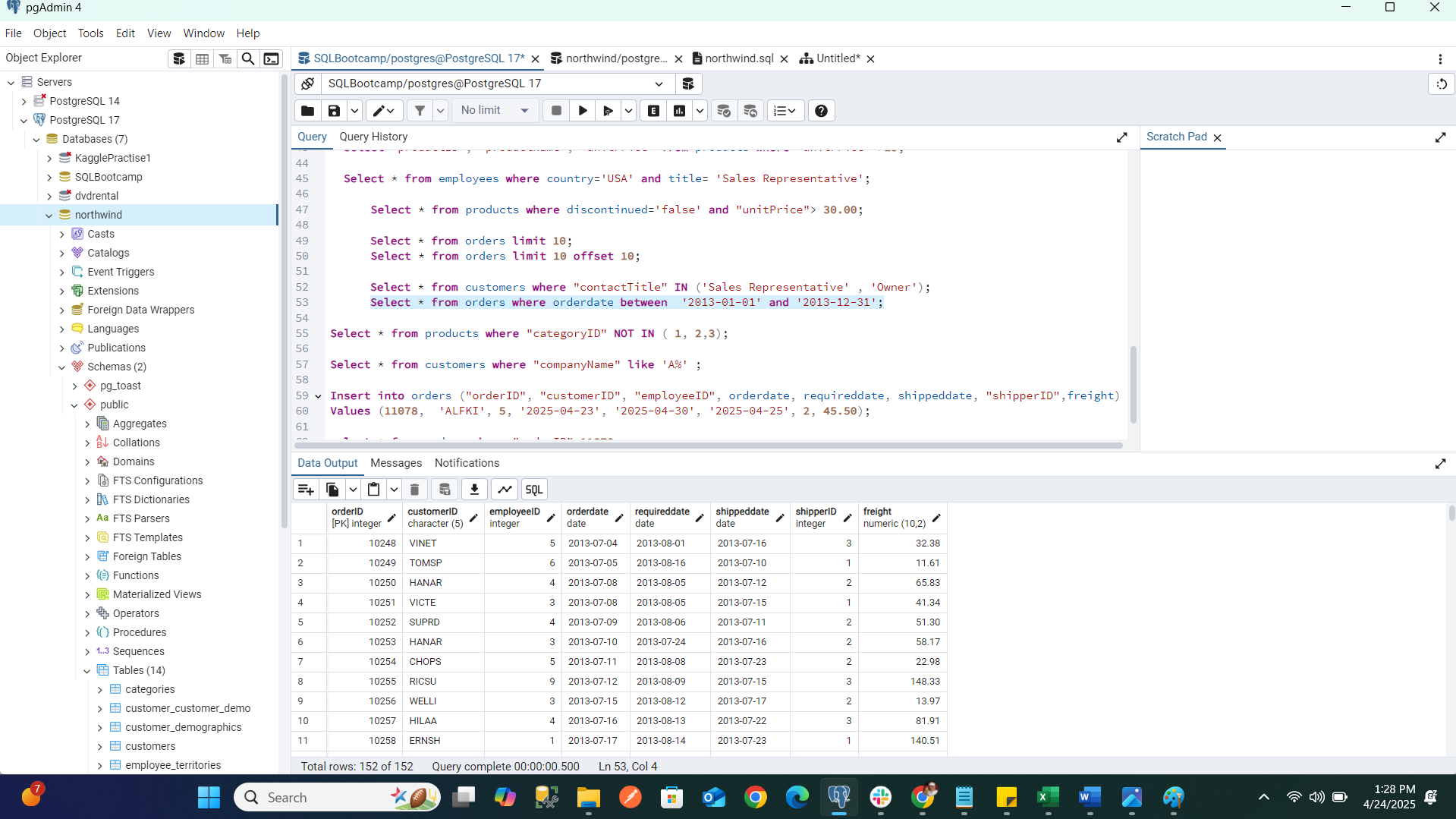
6)      Filtering (IN, BETWEEN)

* List all customers who are either Sales Representative, Owner

Select \* from customers where "contactTitle" IN ('Sales Representative' , 'Owner');

* Retrieve orders placed between January 1, 2013, and December 31, 2013.

Select \* from orders where orderdate between '2013-01-01' and '2013-12-31';



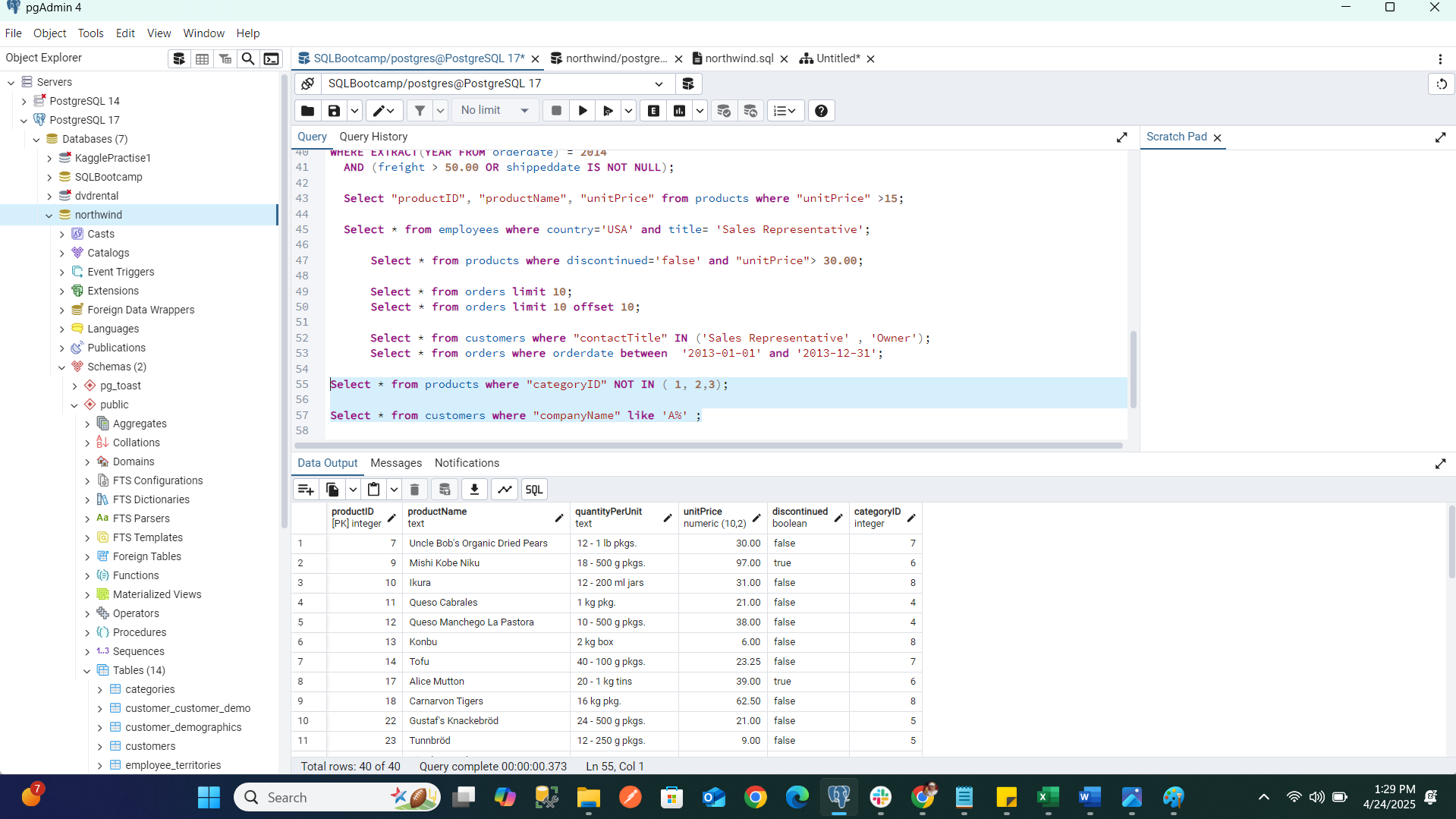
7)      Filtering

* List all products whose category\_id is not 1, 2, or 3.

Select \* from products where "categoryID" NOT IN ( 1, 2,3);

* Find customers whose company name starts with "A".

Select \* from customers where "companyName" like 'A%';



8)       INSERT into orders table:

 Task: Add a new order to the orders table with the following details:

Order ID: 11078

Customer ID: ALFKI

Employee ID: 5

Order Date: 2025-04-23

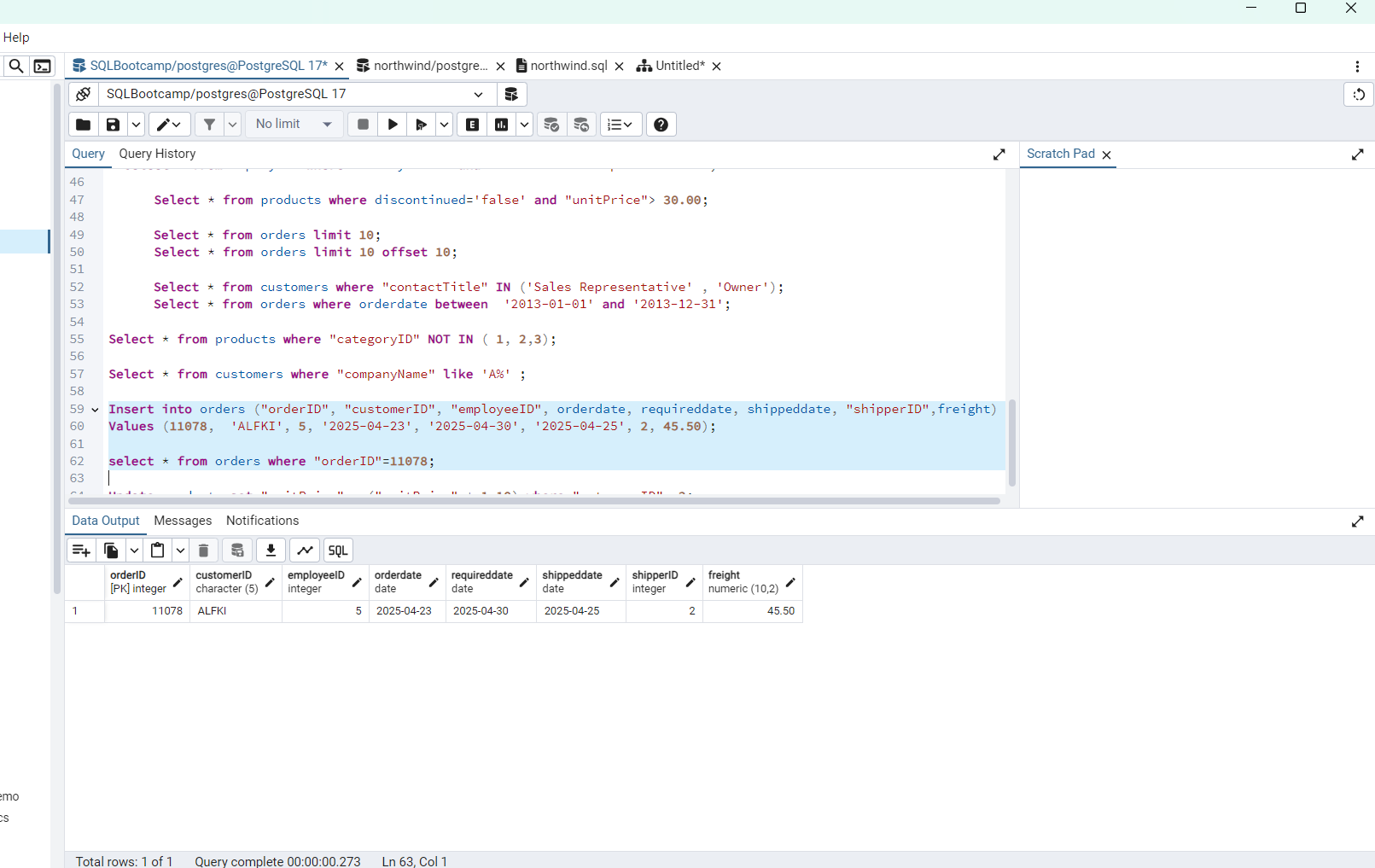
Required Date: 2025-04-30

Shipped Date: 2025-04-25

shipperID:2

Freight: 45.50

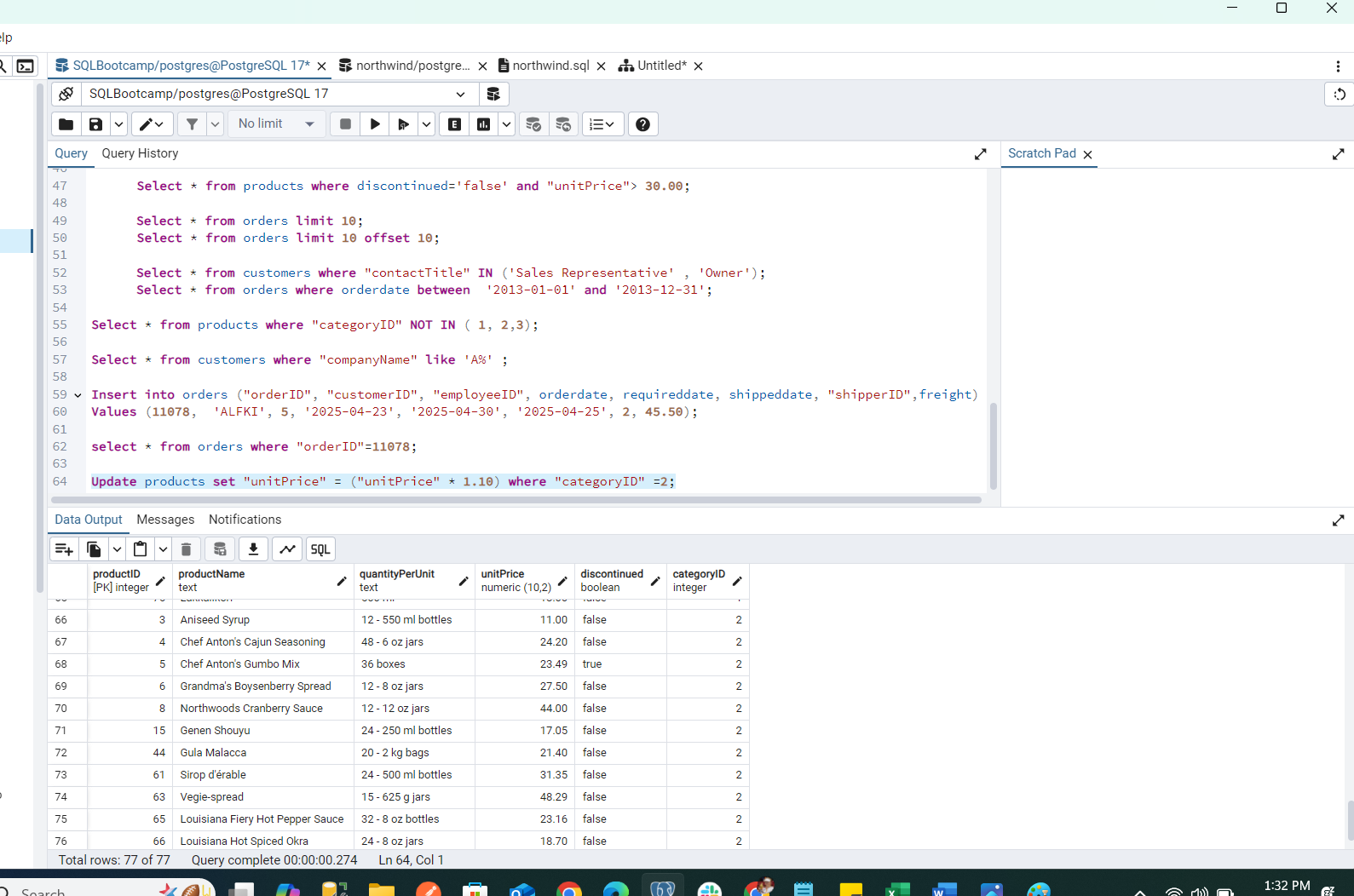
Insert into orders ("orderID", "customerID", "employeeID", orderdate, requireddate, shippeddate, "shipperID",freight) Values (11078, 'ALFKI', 5, '2025-04-23', '2025-04-30', '2025-04-25', 2, 45.50);



9)      Increase(Update)  the unit price of all products in category\_id =2 by 10%.

(HINT: unit\_price =unit\_price \* 1.10)

Update products set "unitPrice" = ("unitPrice" \* 1.10) where "categoryID" =2;



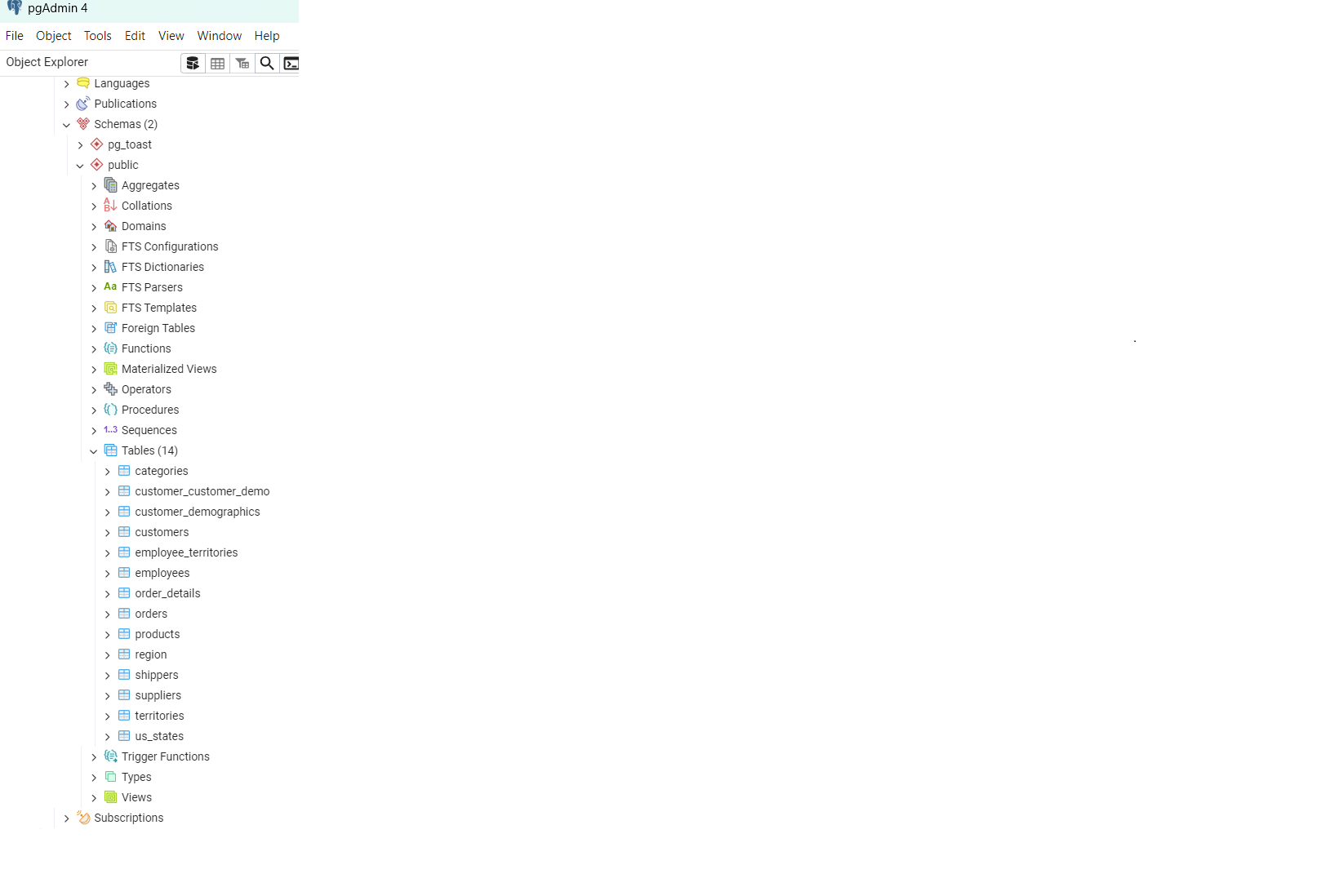
10) Sample Northwind database:

Download

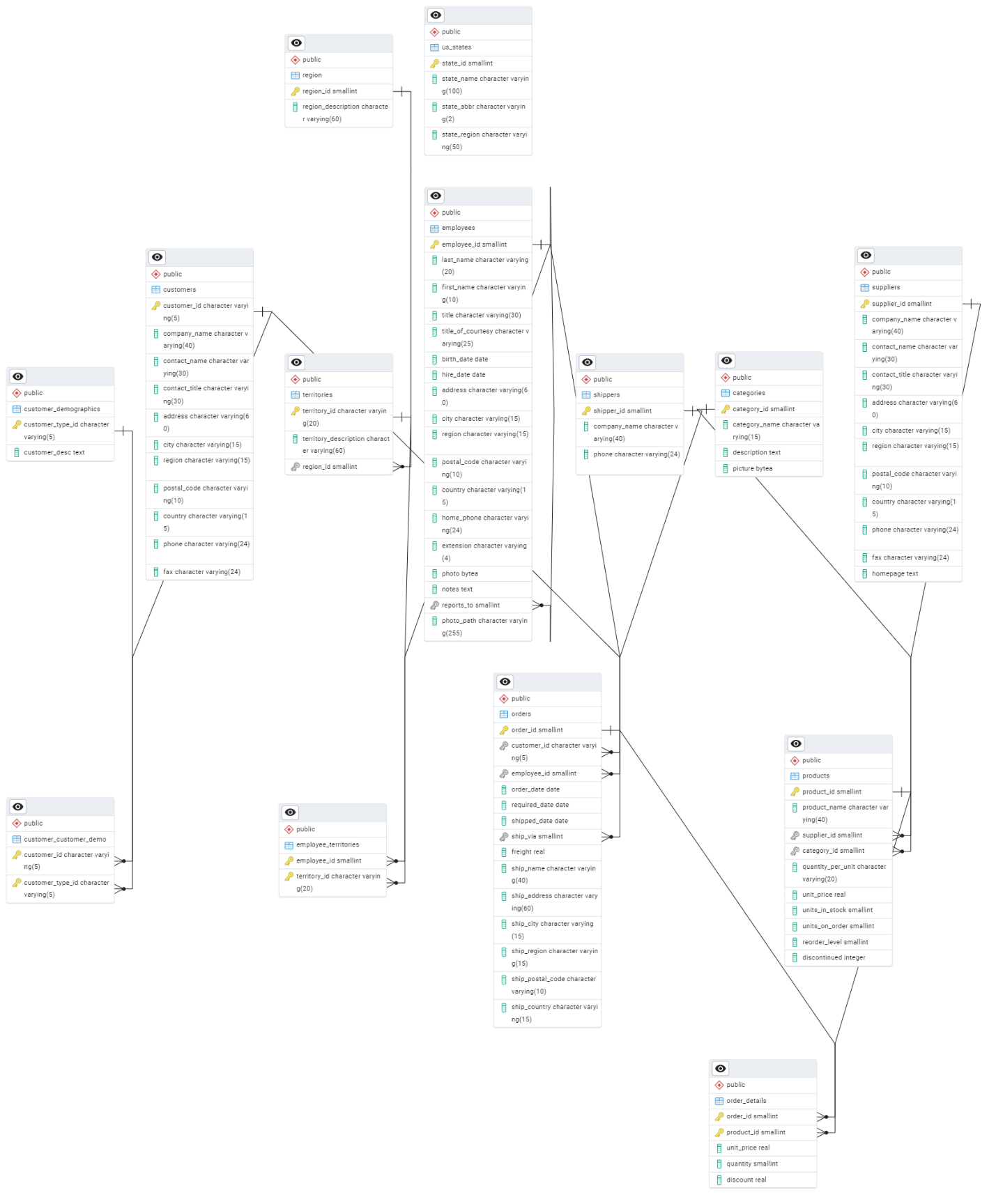
1. Download northwind.sql from below link into your local. Sign in to Git first <https://github.com/pthom/northwind_psql>
2. Manually Create the database using pgAdmin:
   1. Right-click on "Databases" → Create → Database
   2. Give name as ‘northwind’ (all small letters)
   3. Click ‘Save’

Import database:

1. Open pgAdmin and connect to your server
2. Select the database  ‘northwind’
3. Right Click-> Query tool.
4. Click the folder icon to open your northwind.sql file
5. Press F5 or click the Execute button.
6. You will see total 14 tables loaded
7. Databases → your database → Schemas → public → Tables



14 Tables imported to northwind db



ER Dig north wind Day2