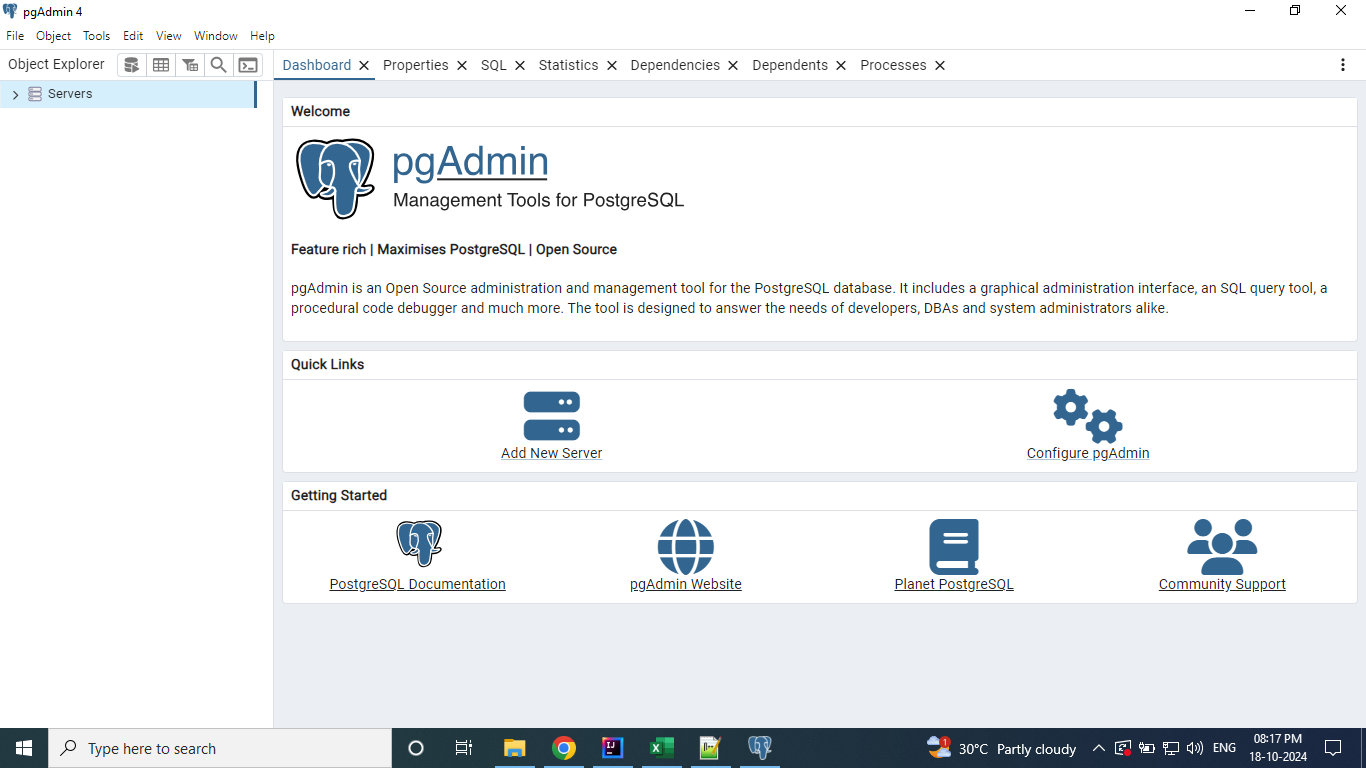
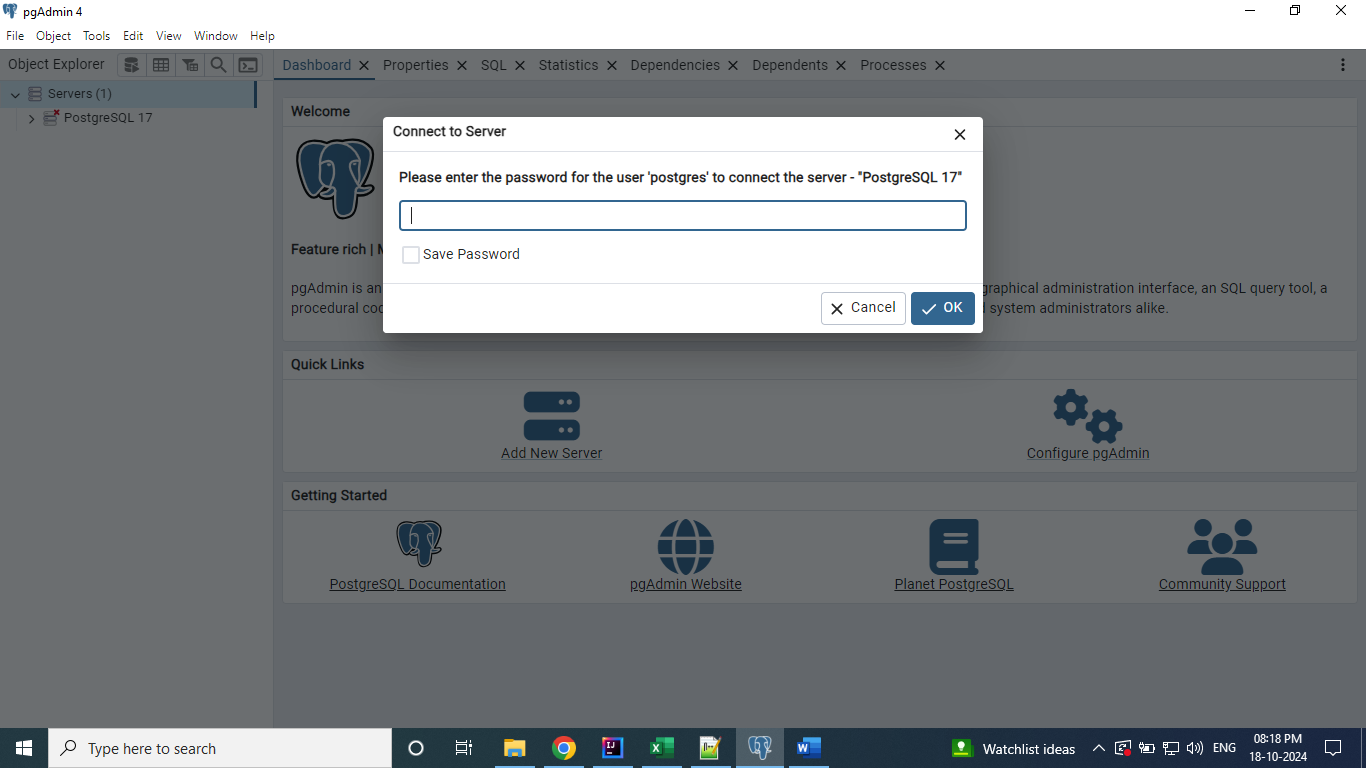
The very first look of Postgres as below.



When you click on Servers. You will be prompted for the password provided during the installation.



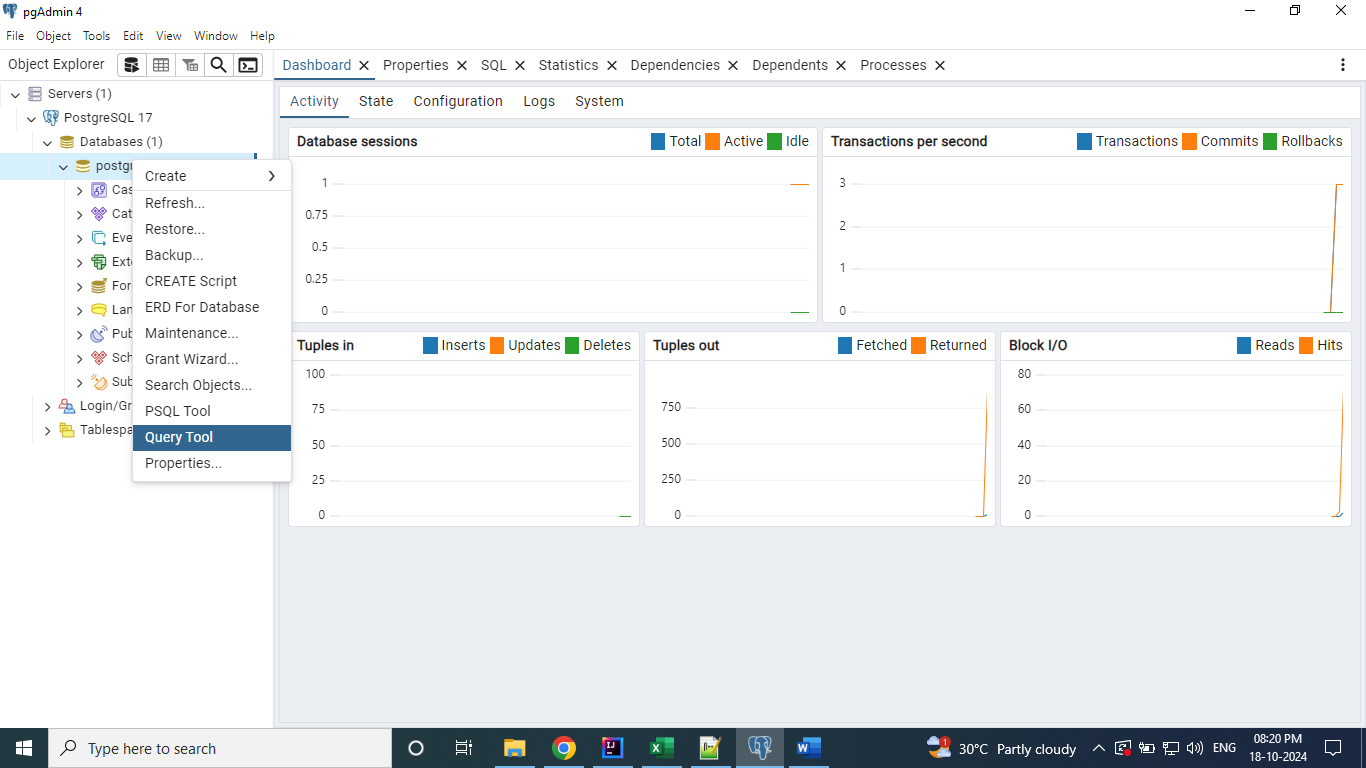
Enter the password and click Ok.



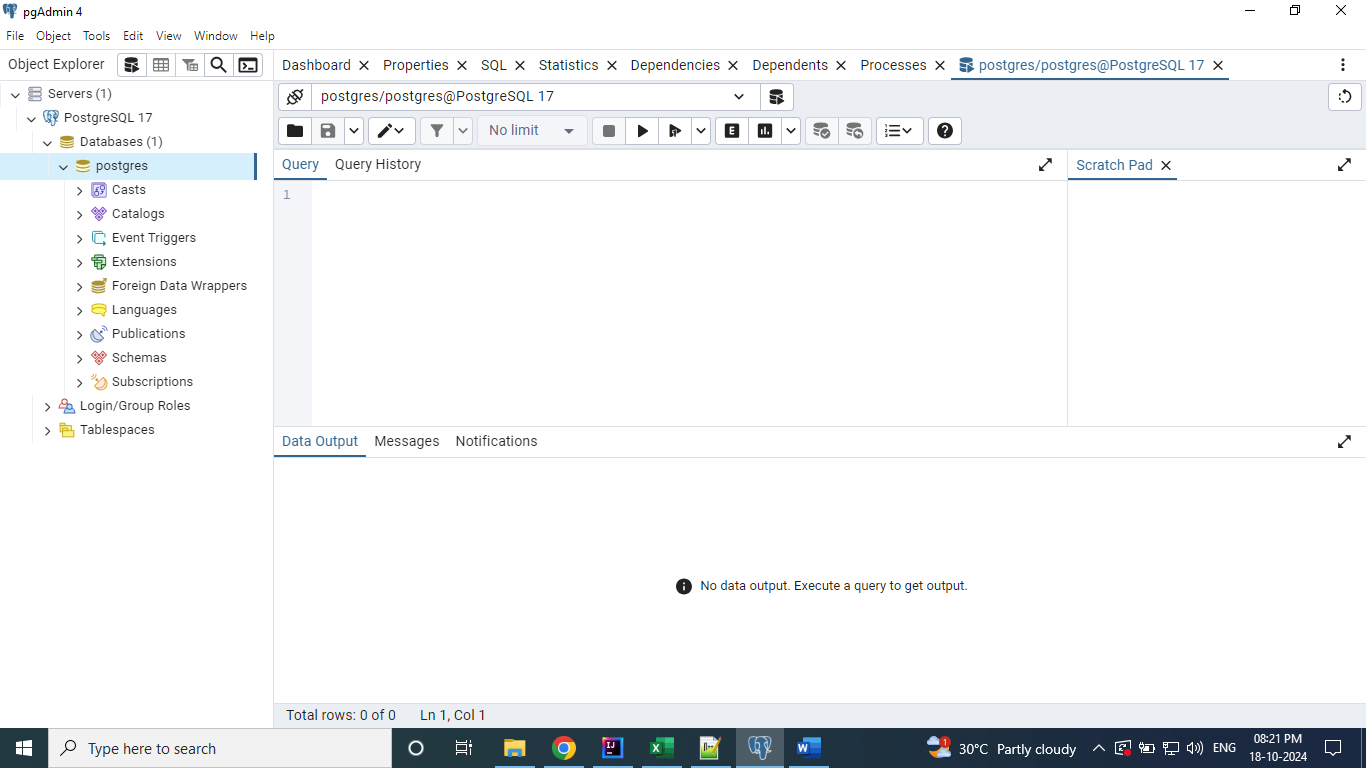
After click on Ok. The screen looks like below. It has default Database.



Right click on Database and select Query Tool.

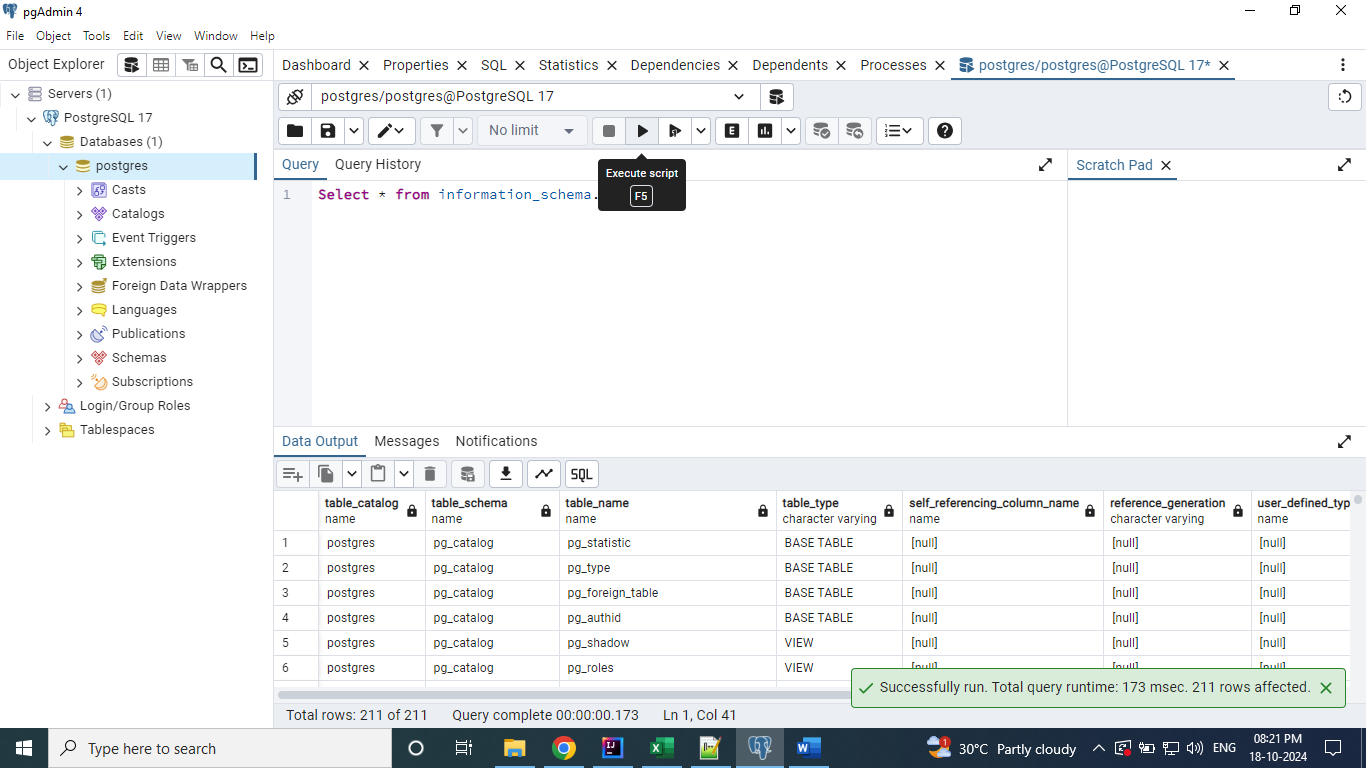


Query Tool Looks like below.



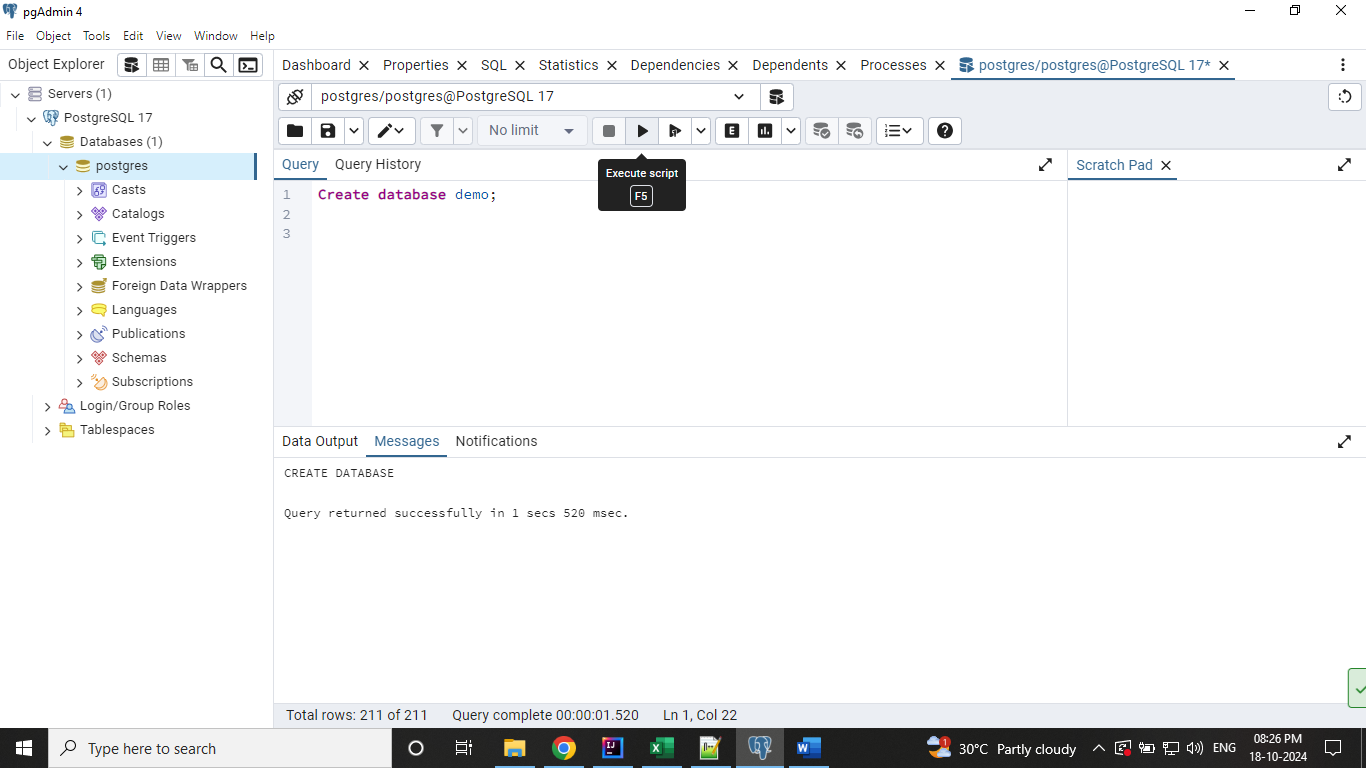
Executed the below query to know the tables details available on the default Database.

**Select \* from information\_schema.tables;**

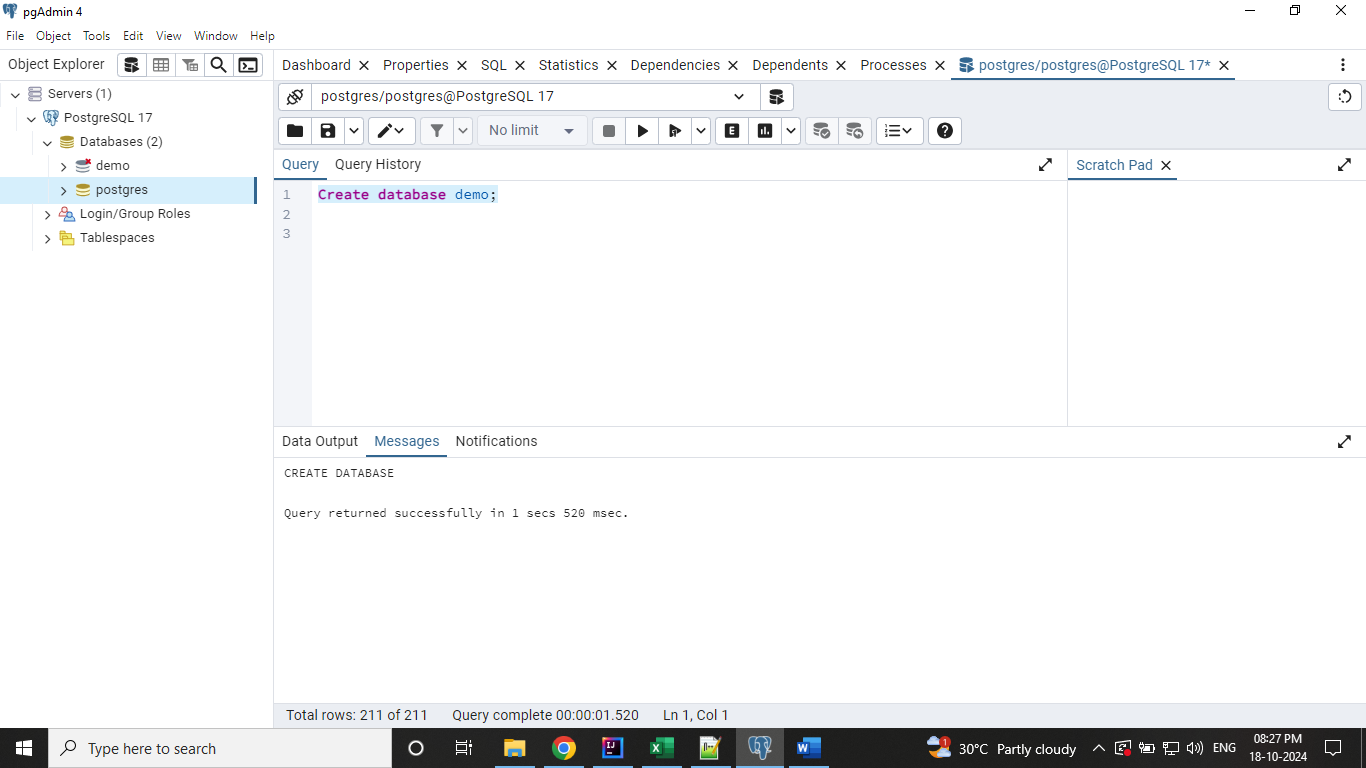


Have created the new Database with name “demo”

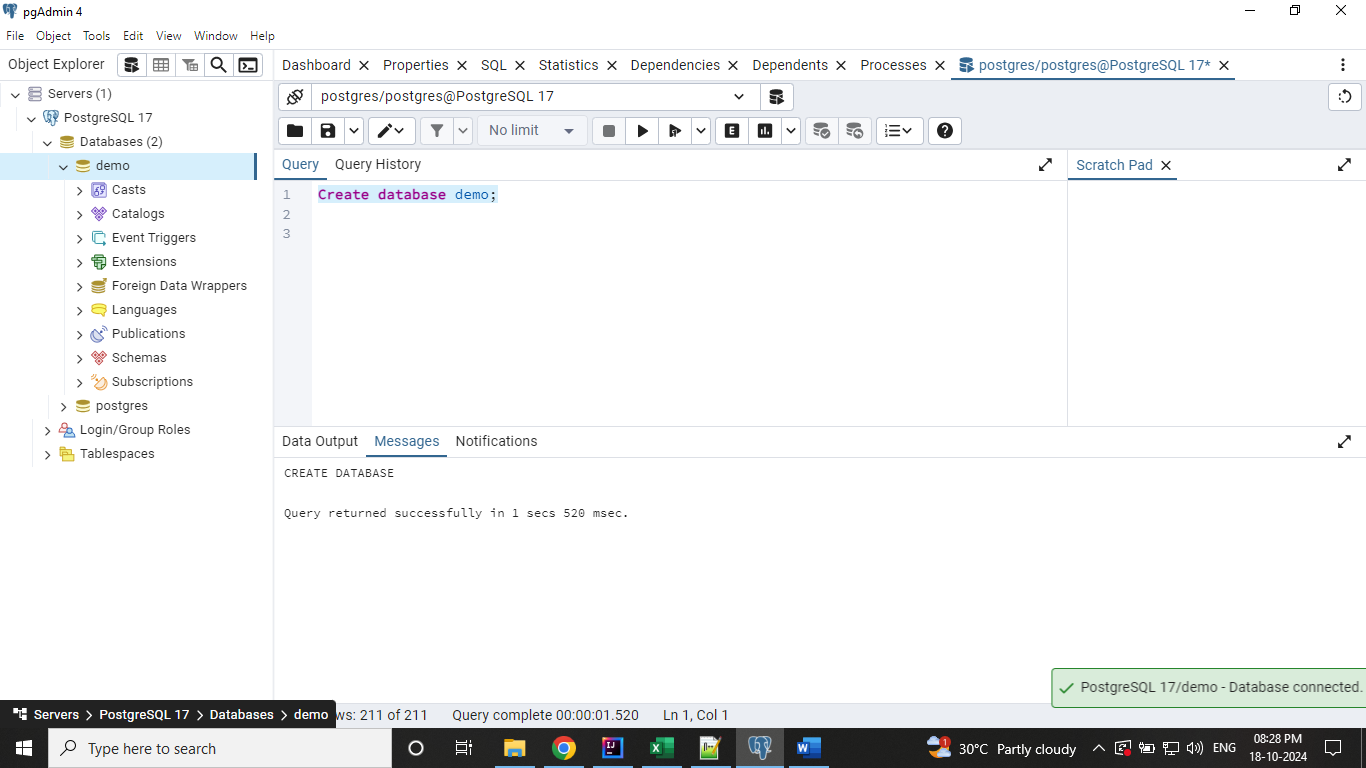
**Create database demo;**

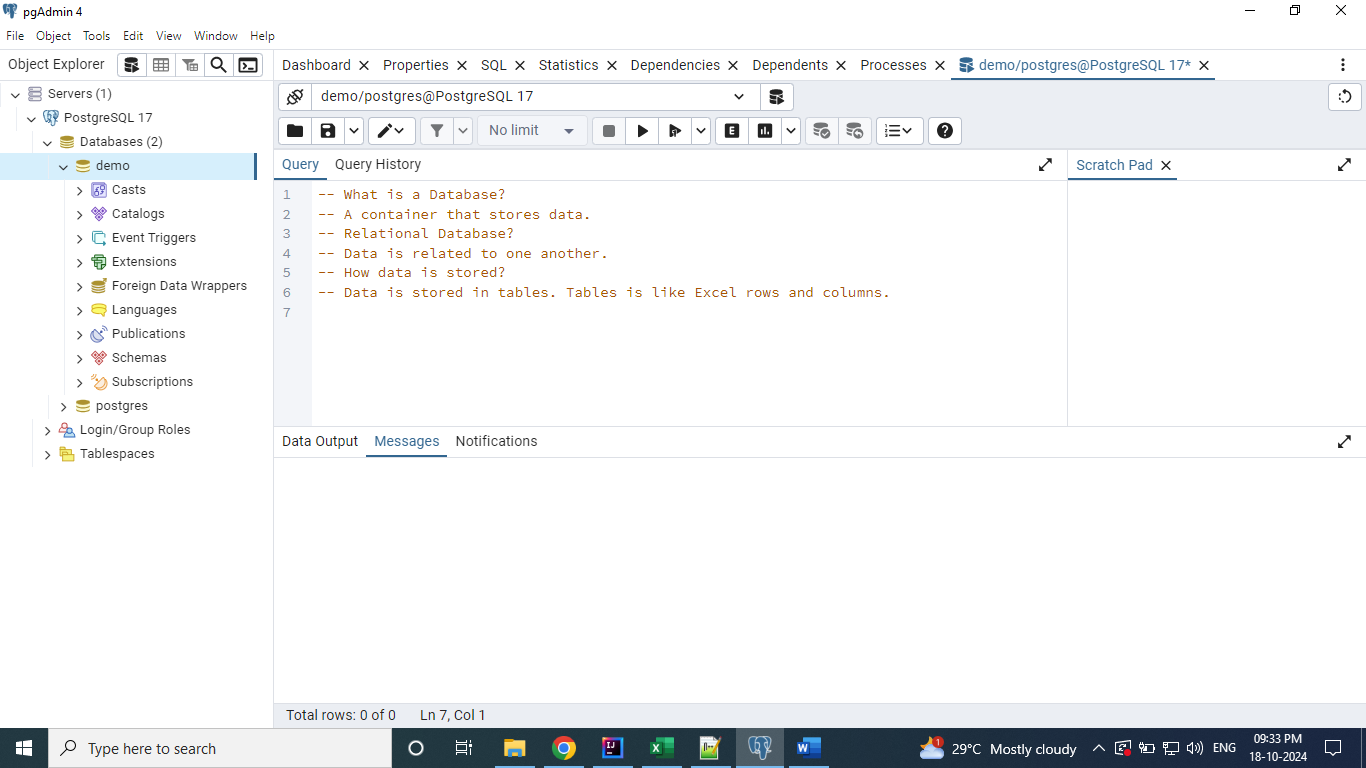


Right click on the Databases and click on Refresh to see the newly created Database.



By default, the connection to the new Database is not established. You can click on it, then the connection will be established and the ‘x’ mark will be removed.





Created a table with name “Products” on the demo database.

**create table products (**

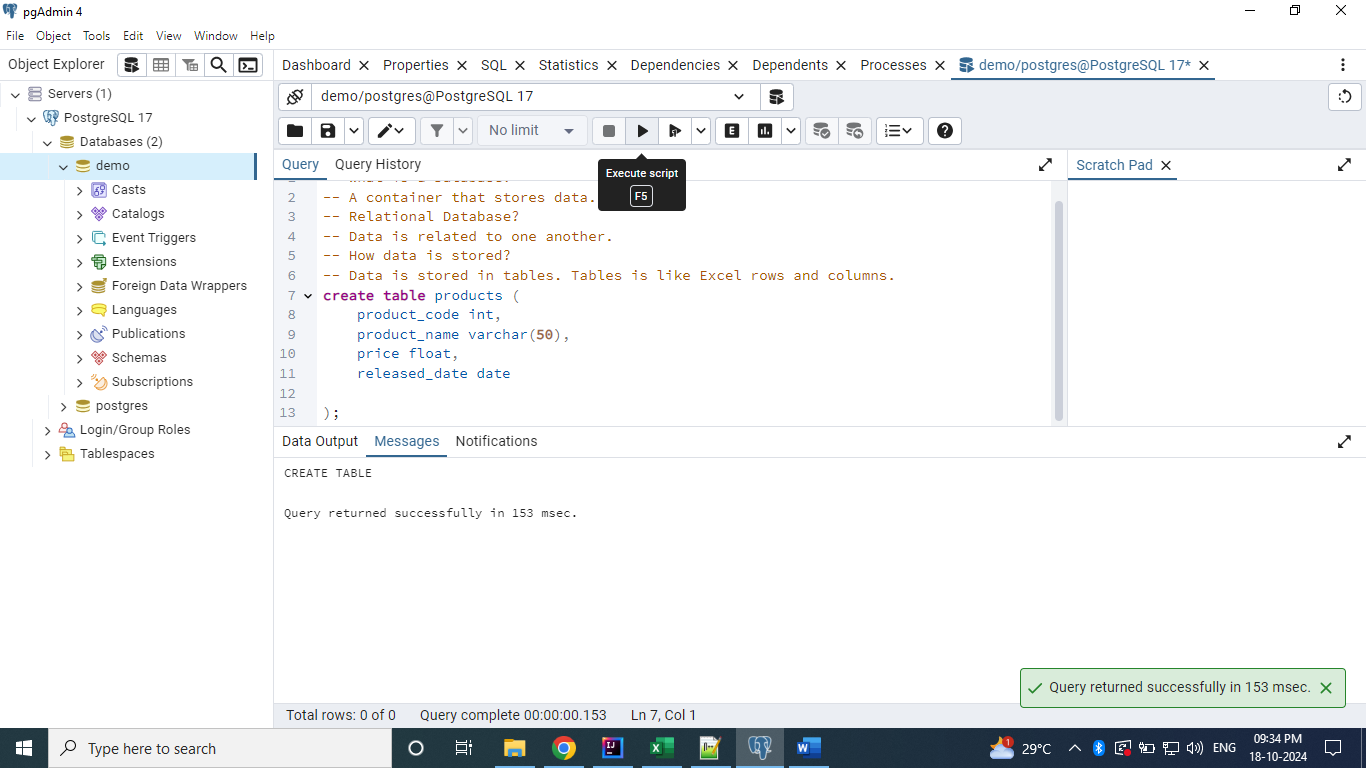
**product\_code int,**

**product\_name varchar(50),**

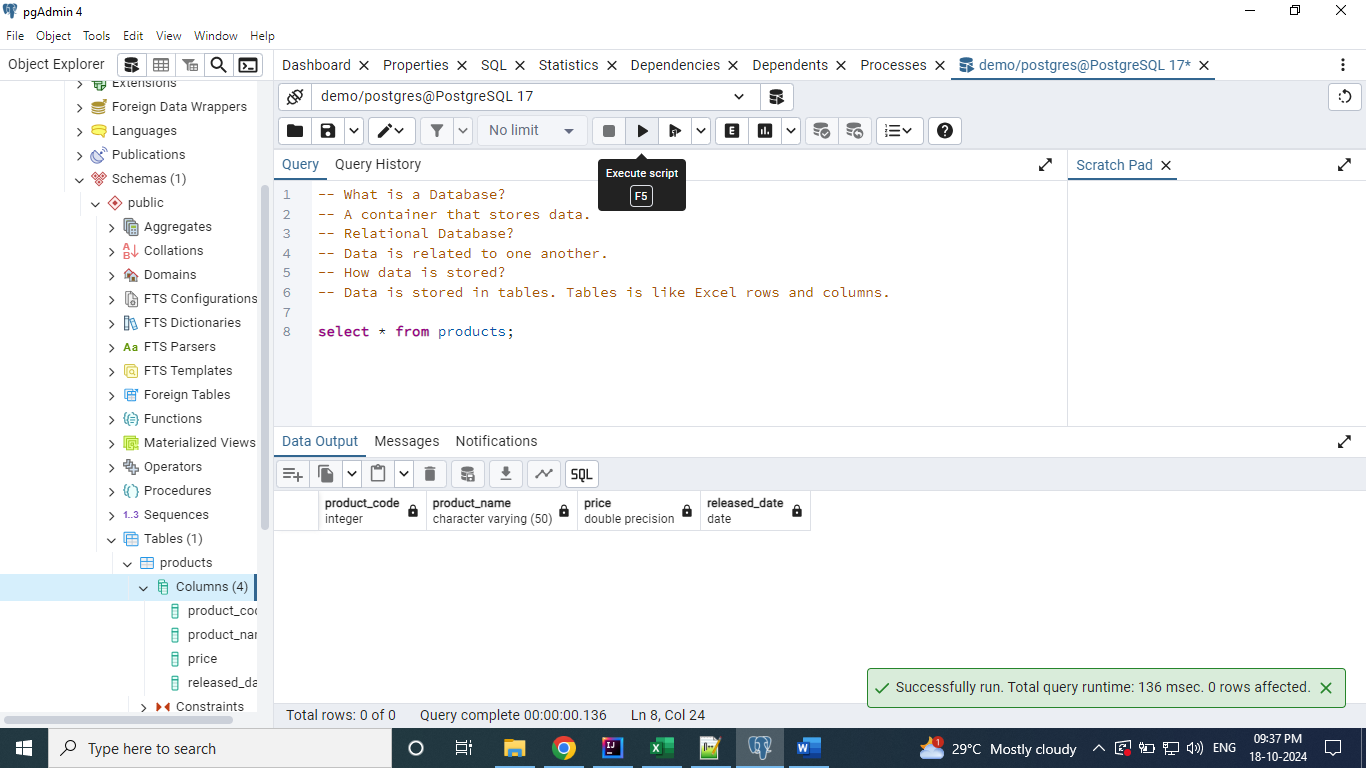
**price float,**

**released\_date date**

**);**



Query the created table.

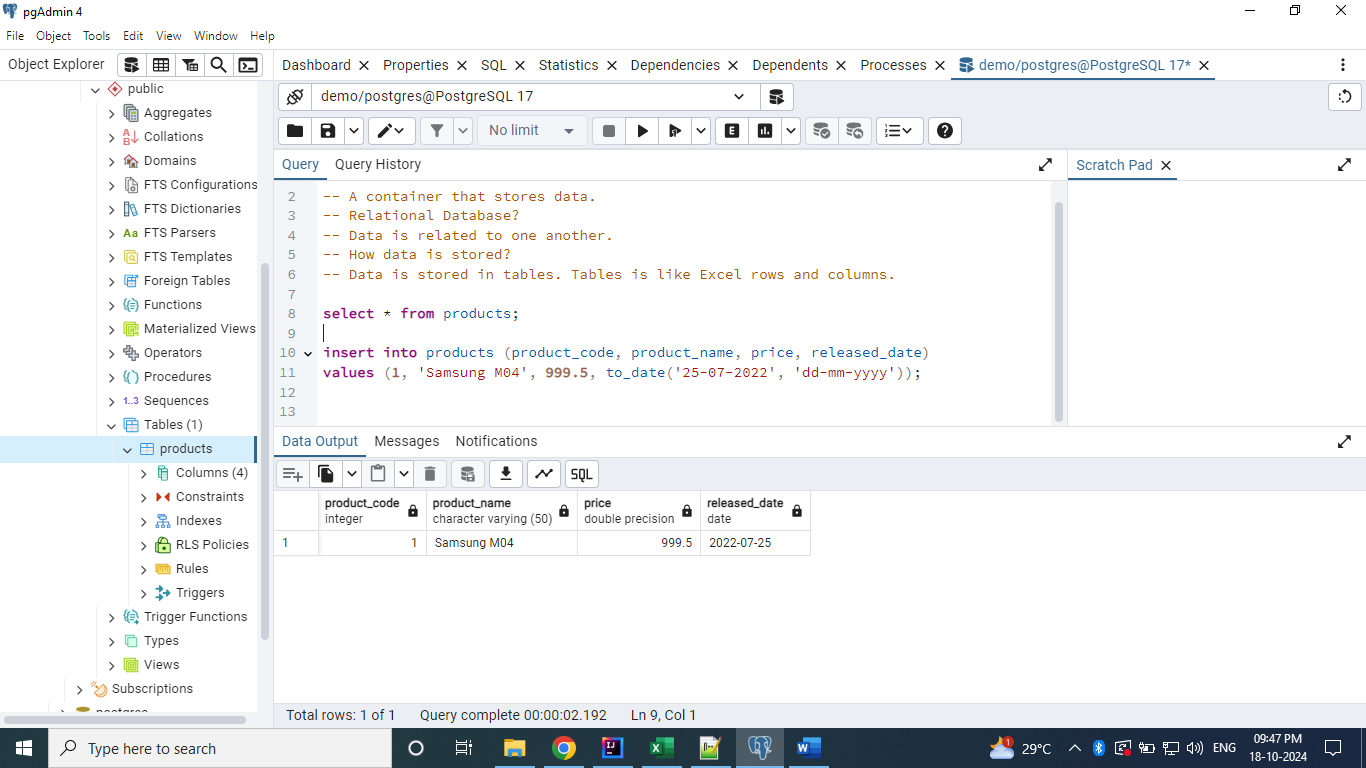


We can add the data into the table as below.

**insert into products (product\_code, product\_name, price, released\_date)**

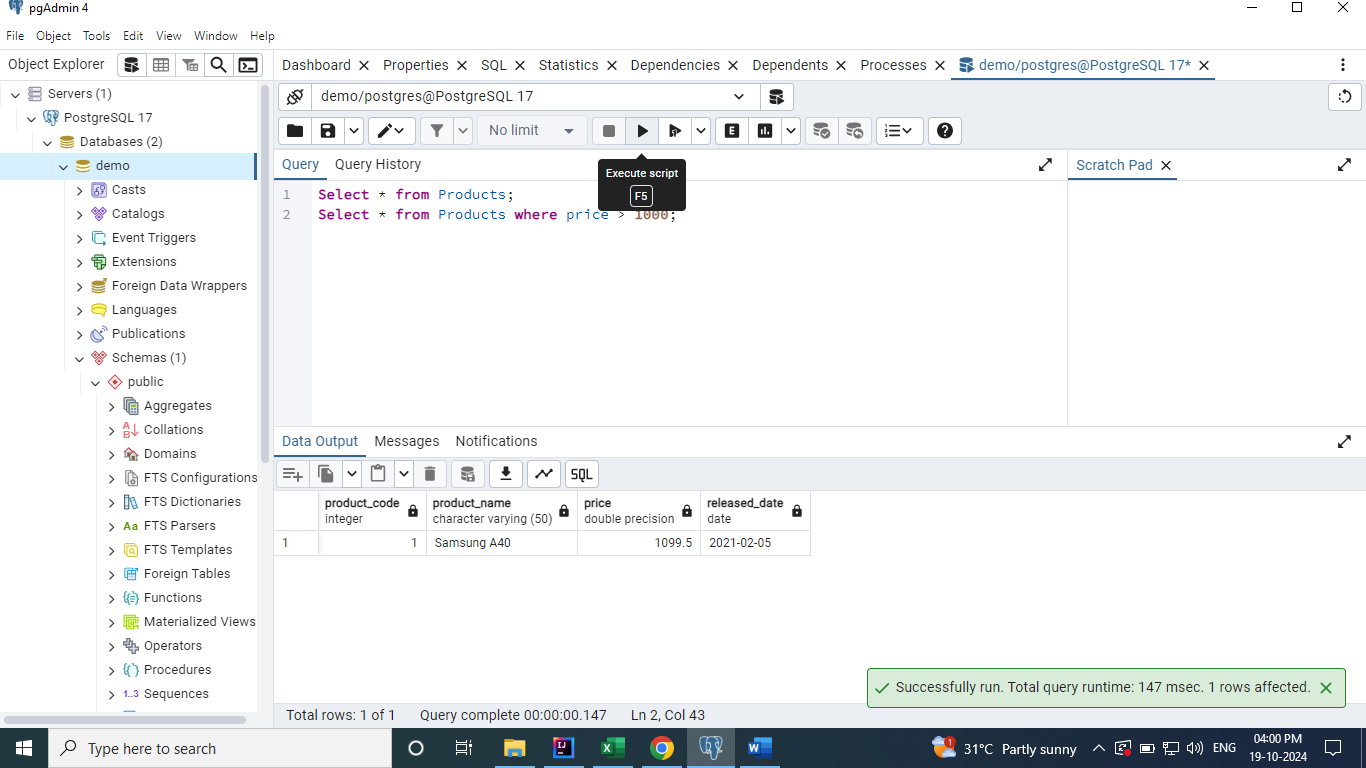
**values (1, 'Samsung M04', 999.5, to\_date('25-07-2022', 'dd-mm-yyyy'));**





Fetch the records from Product table based on price whose value is greater than 1000.

**Select \* from Products where price > 1000;**

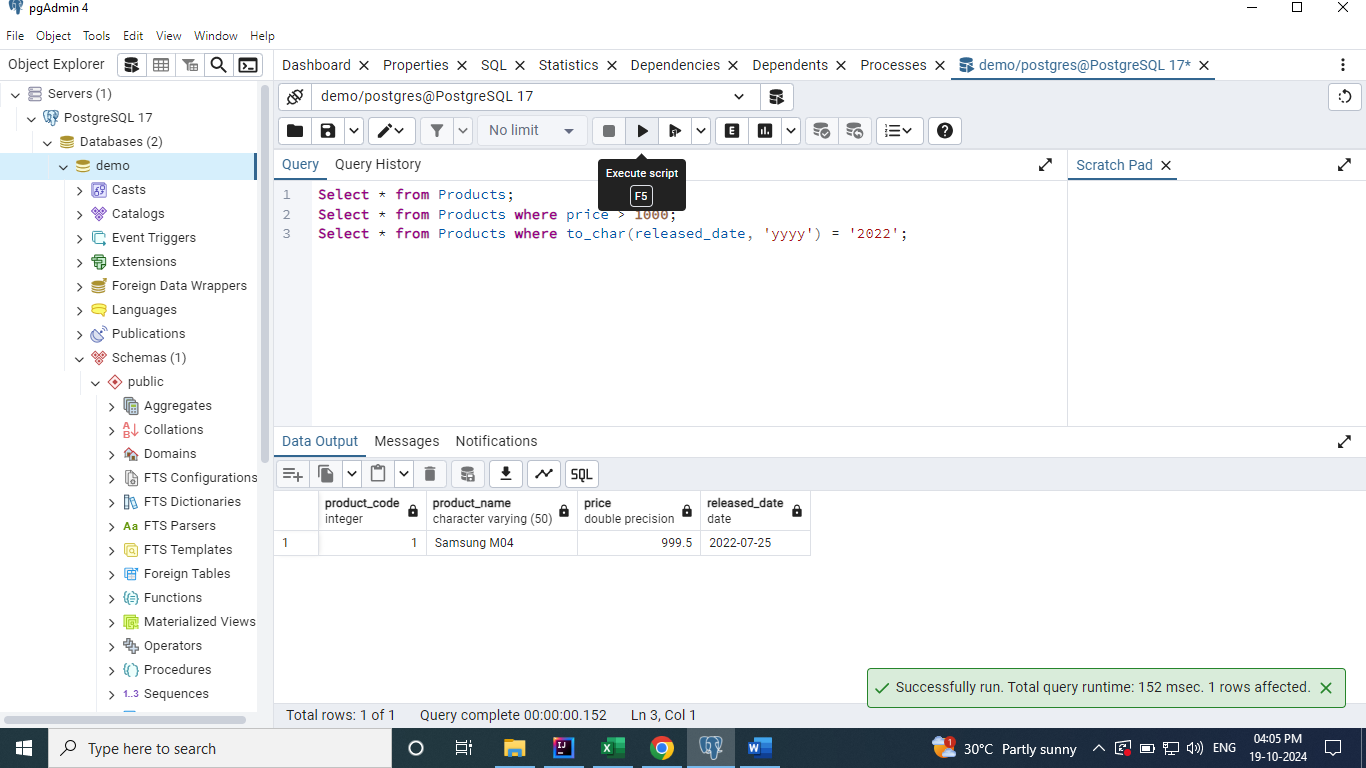


Fetch the records based on the year. Here, the released\_date column is of type ‘date’ and when

We query using the value year as ‘2022’ it will throw an error. So, we need to typecast / convert the

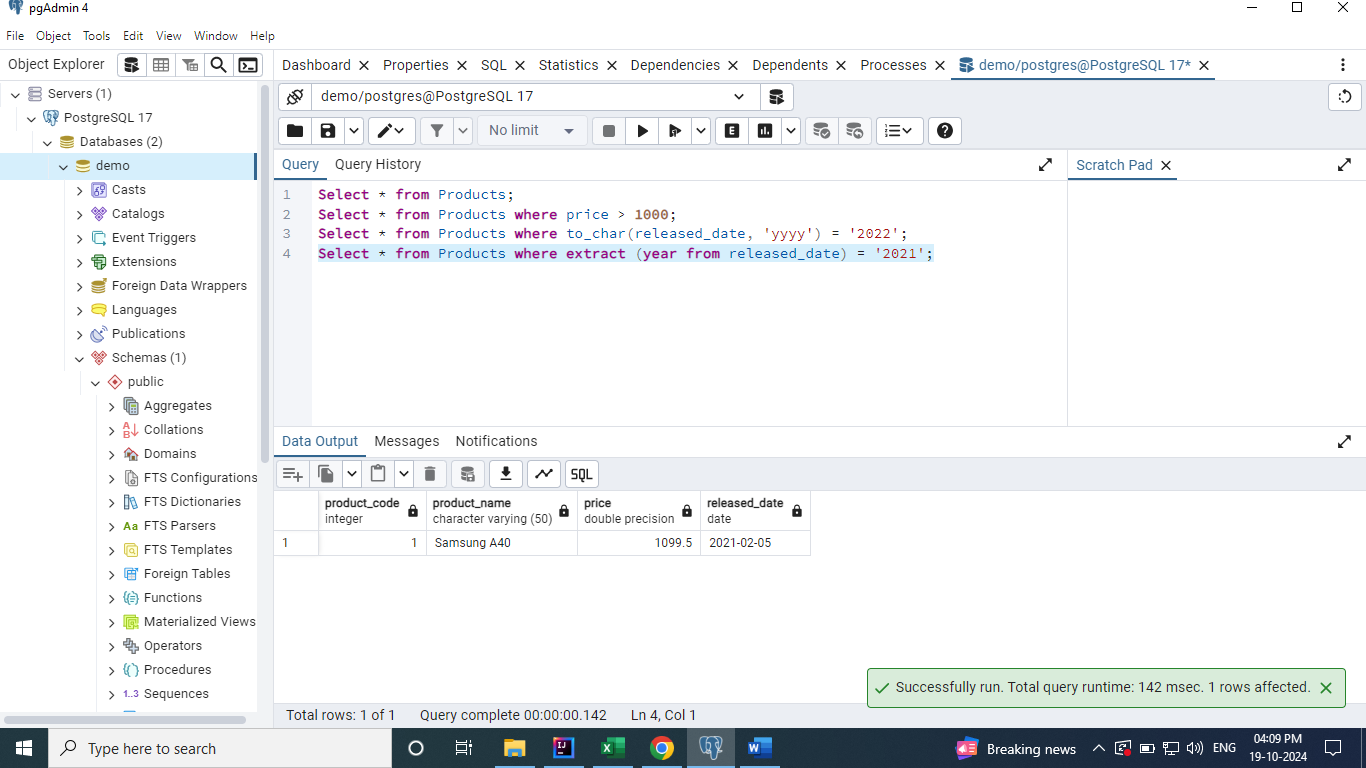
released\_date column to varchar type using the to\_char function.

**Select \* from Products where to\_char(released\_date, 'yyyy') = '2022';**



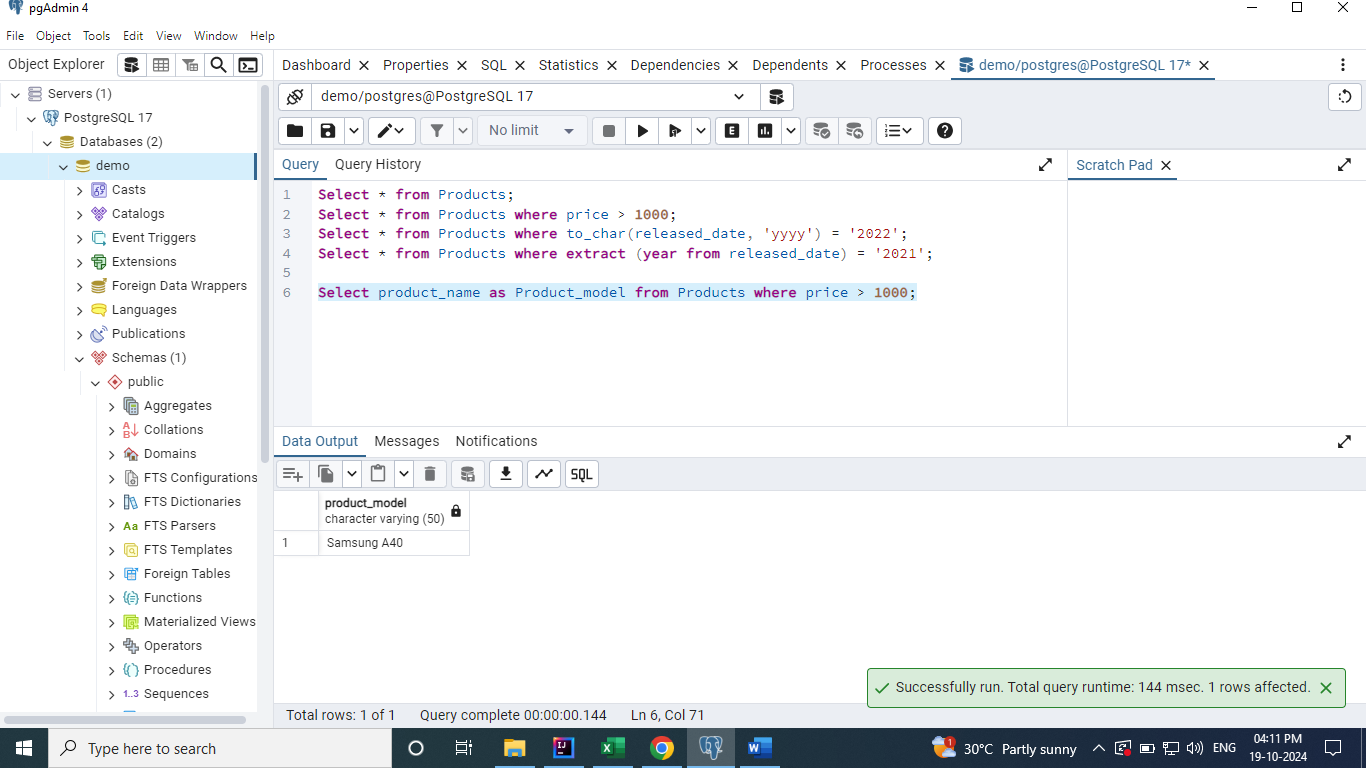
The above result can be retrieved using the Postgres SQL’s specific function called extract.

**Select \* from Products where extract (year from released\_date) = '2021';**



Using alias when retrieving the data from the Products table. ‘As’ keyword is used for alias. We can use it for both the columns and tables.

**Select product\_name as Product\_model from Products where price > 1000;**

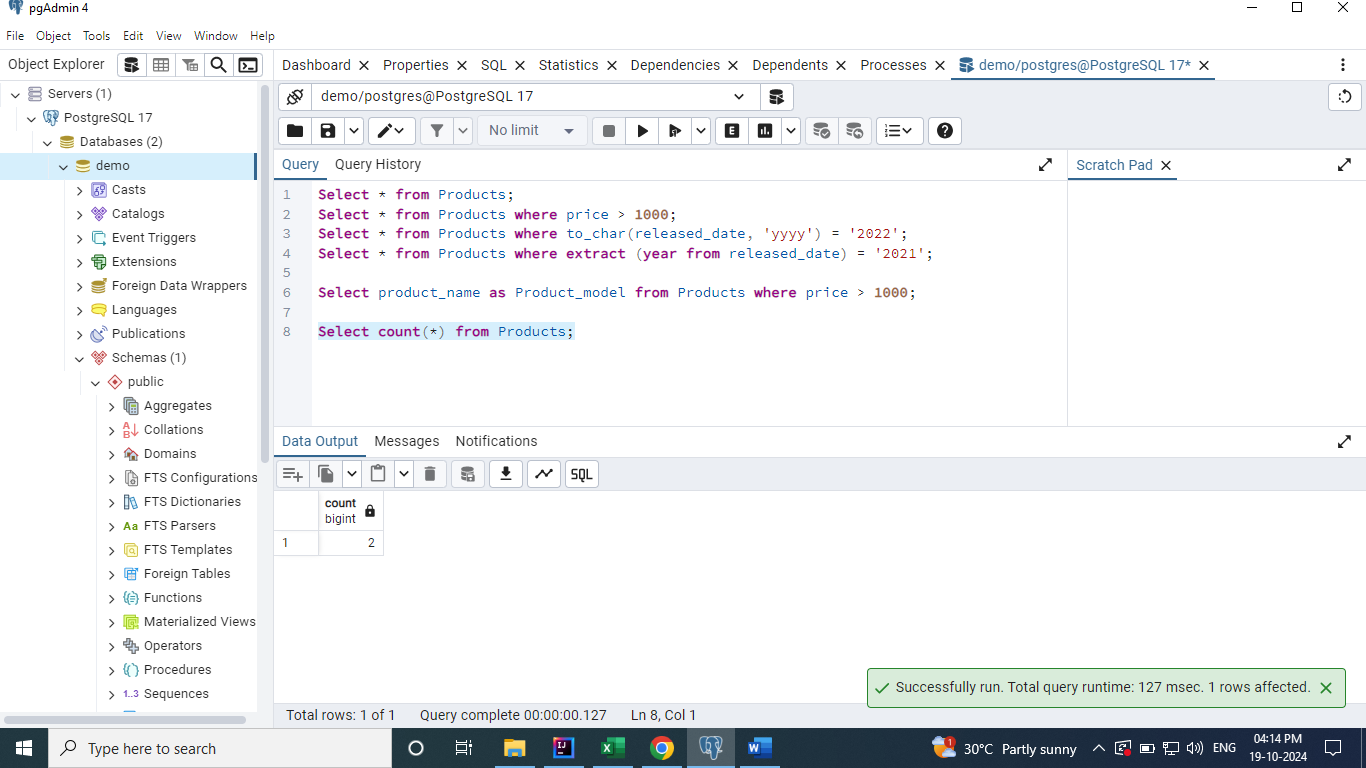


**Aggregate Functions**

Count, Sum and Avg.

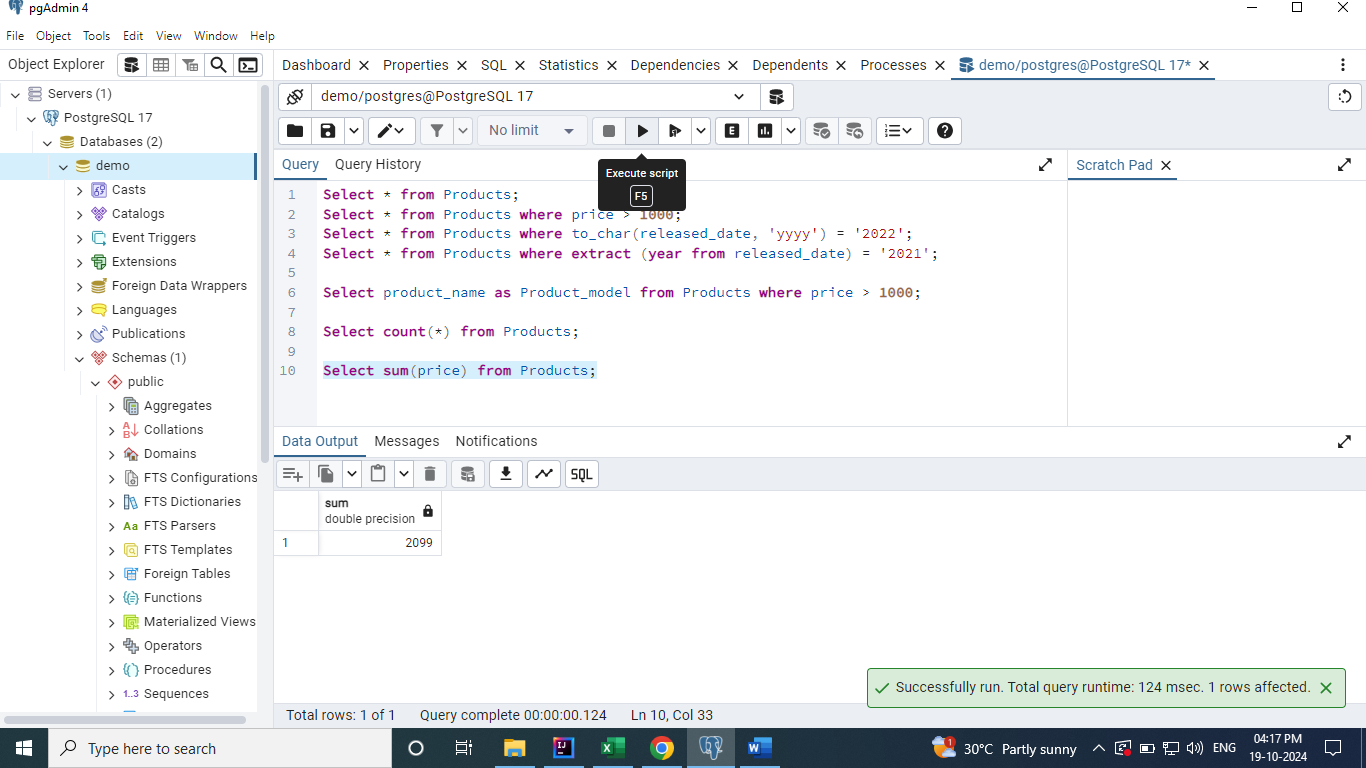
**Count –** It is used to get the total number of records present in the table.

**Select count(\*) from Products;**



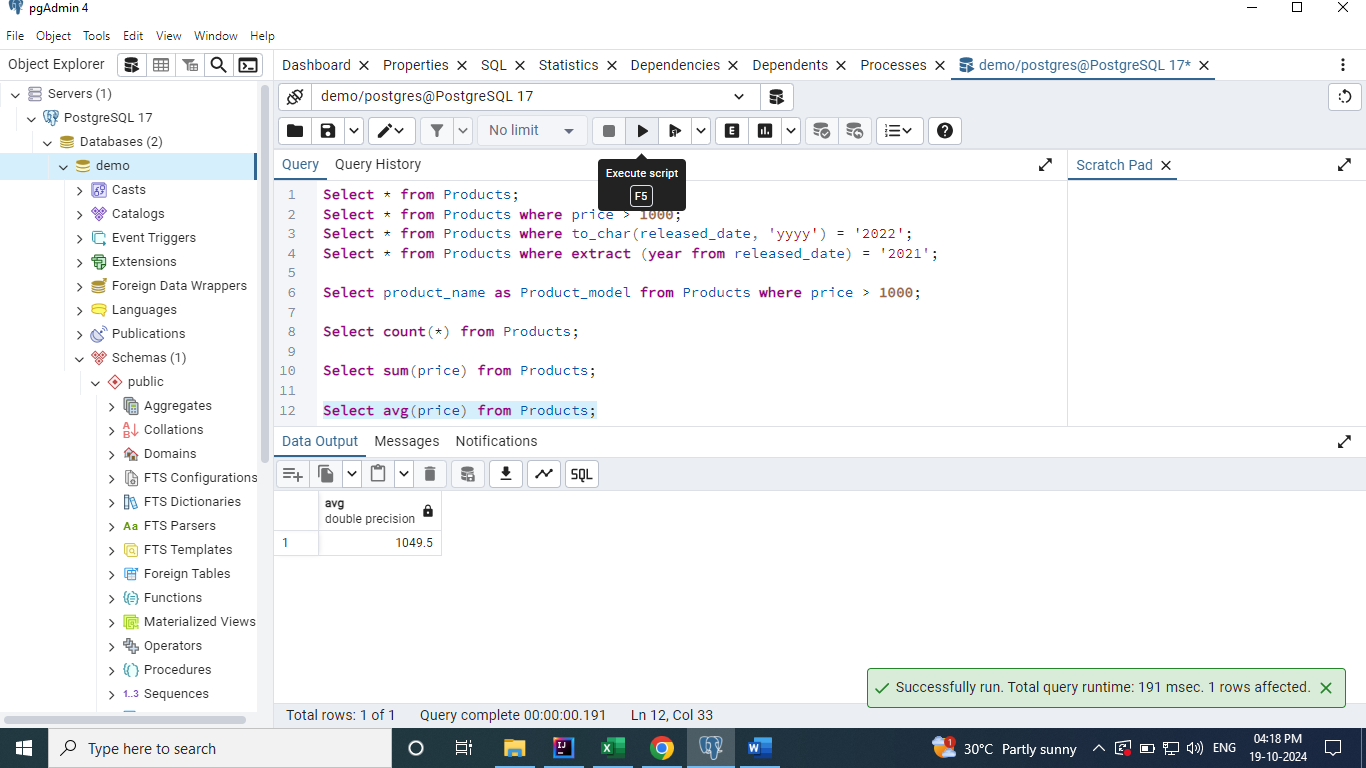
**Sum -** It is used to sum up the all the values in a particular column.

**Select sum(price) from Products;**



**Avg -** It is used to sum up the all the values in a particular column and find the average of it.

**Select avg(price) from Products;**



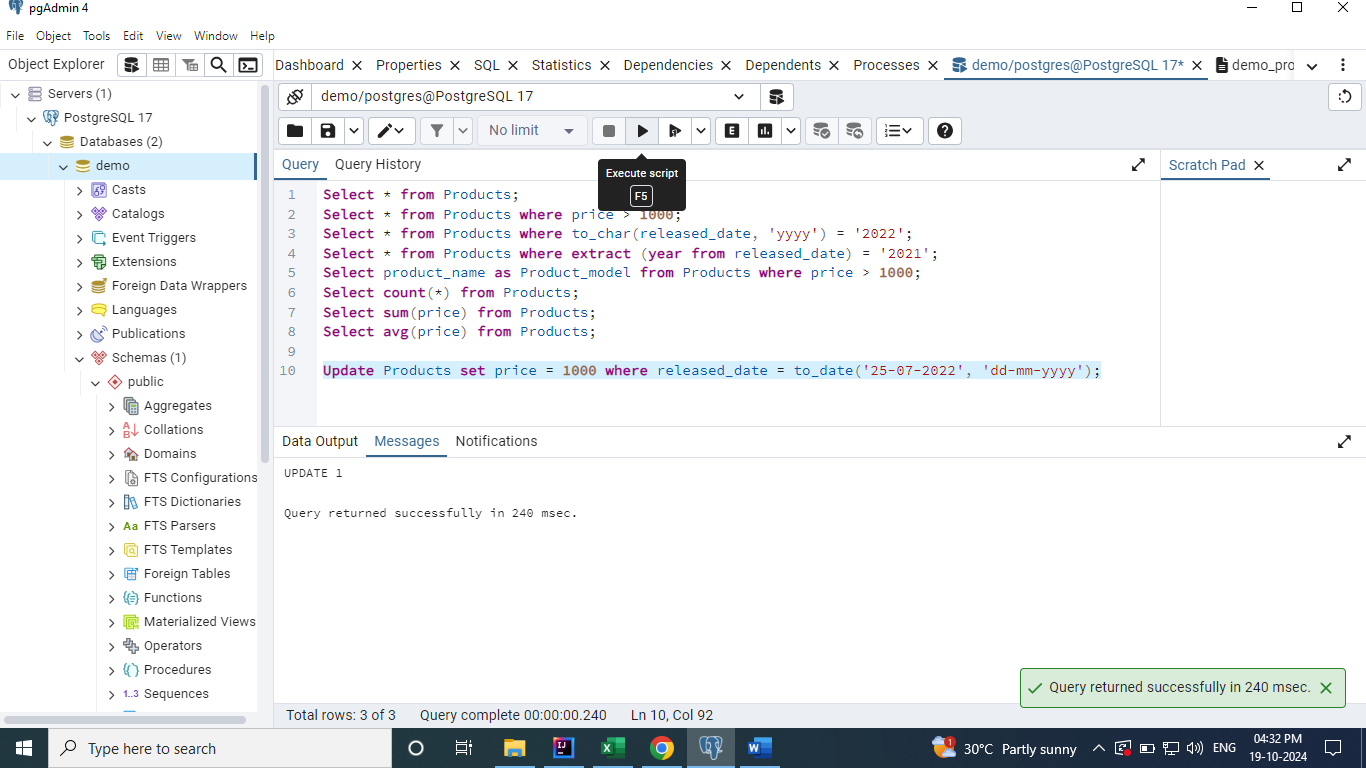
**Update Operations**

Here, I have used the update keyword to update the table records.

Updated one of the records of products table by using the where clause. If the where clause is NOT

used in the update statement, then the update operation will be applied to all the records.

**Update Products set price = 1000 where released\_date = to\_date('25-07-2022', 'dd-mm-yyyy');**



After update



**Using Like Keyword**

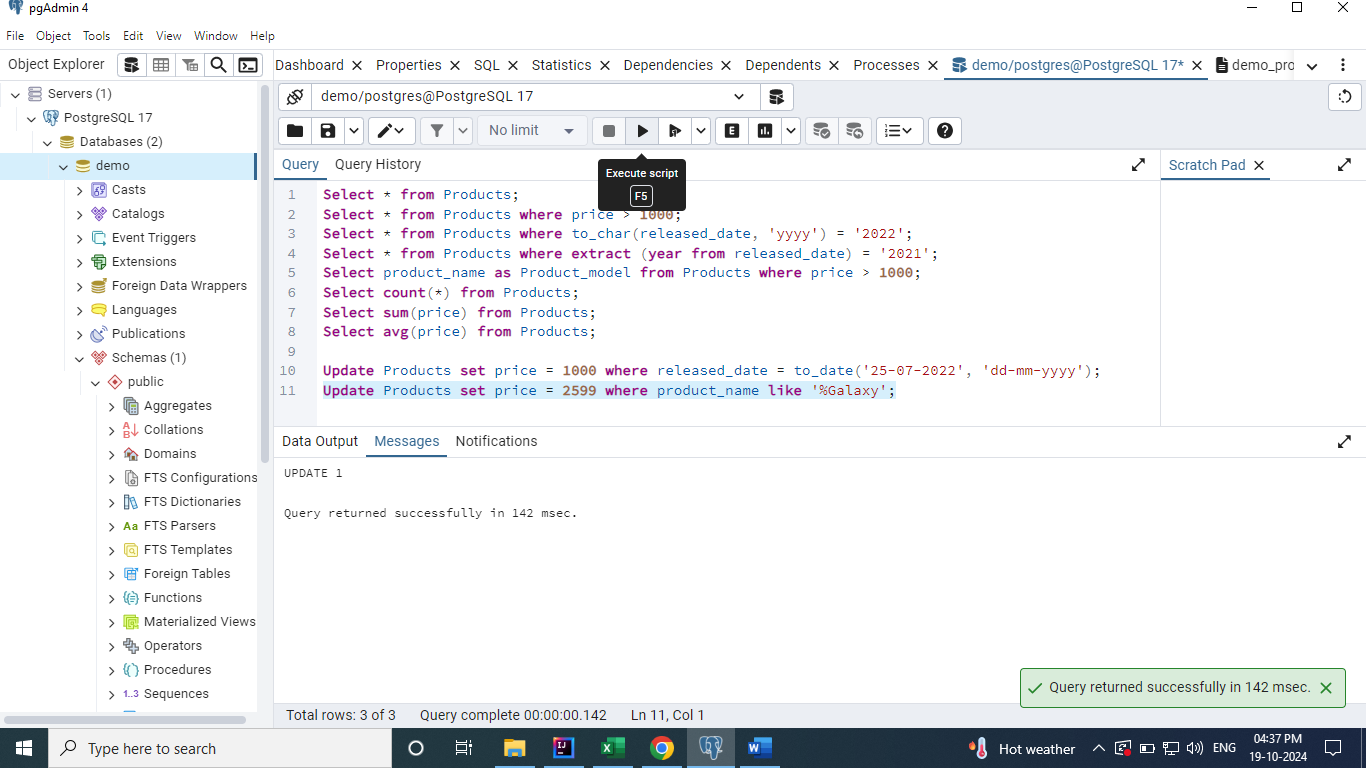
**We can use the like keyword to update the records. The like operator can be used in below ways.**

**Like value%** - Ends with value

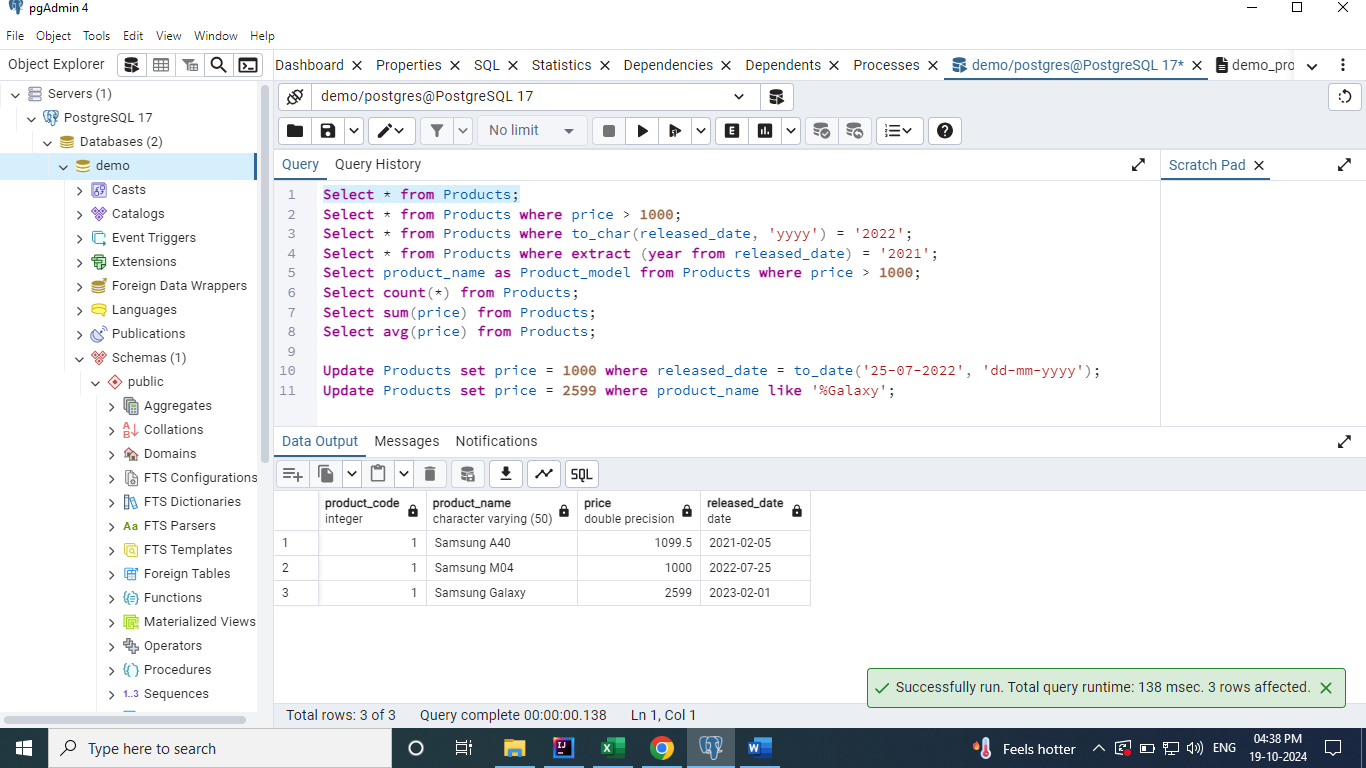
**Like %value** - Starts with value

**Like %value%** - Anywhere the value lies.

**Update Products set price = 2599 where product\_name like '%Galaxy';**

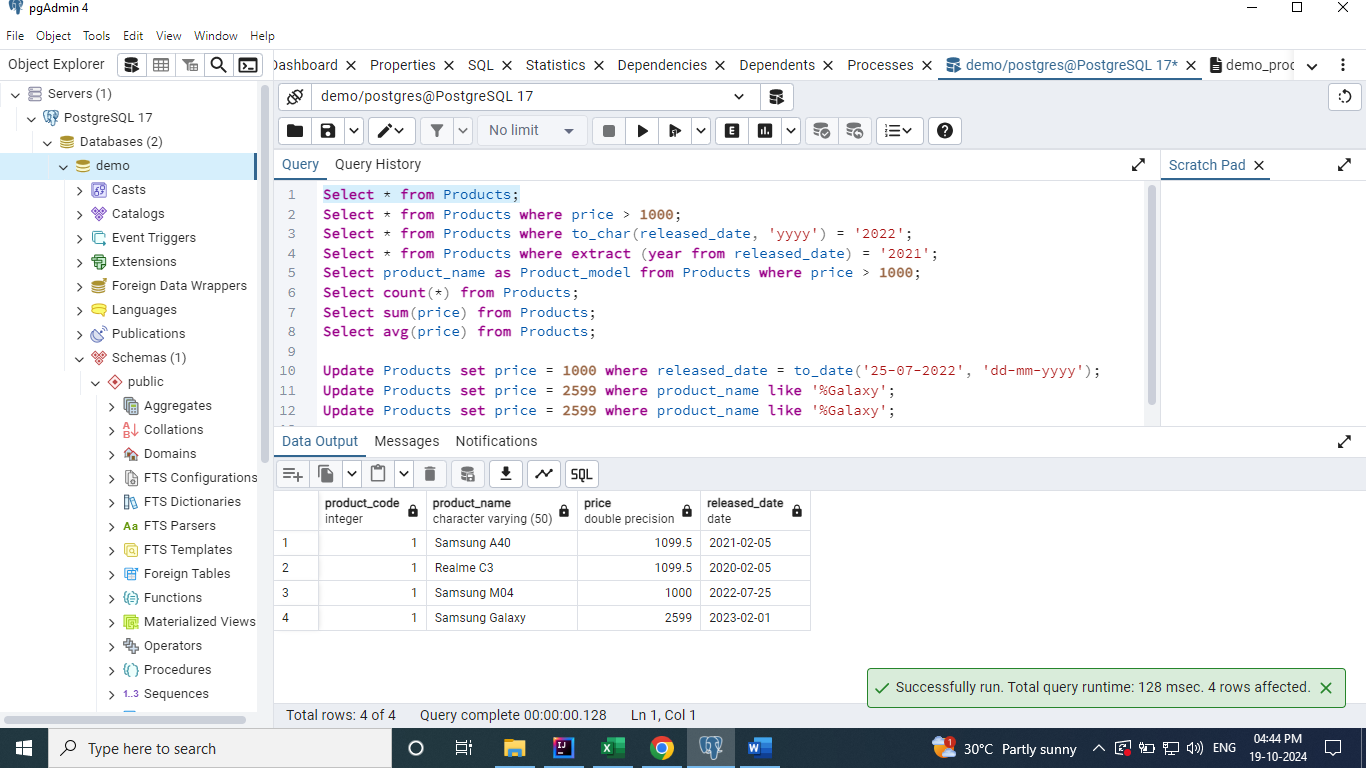


After Update



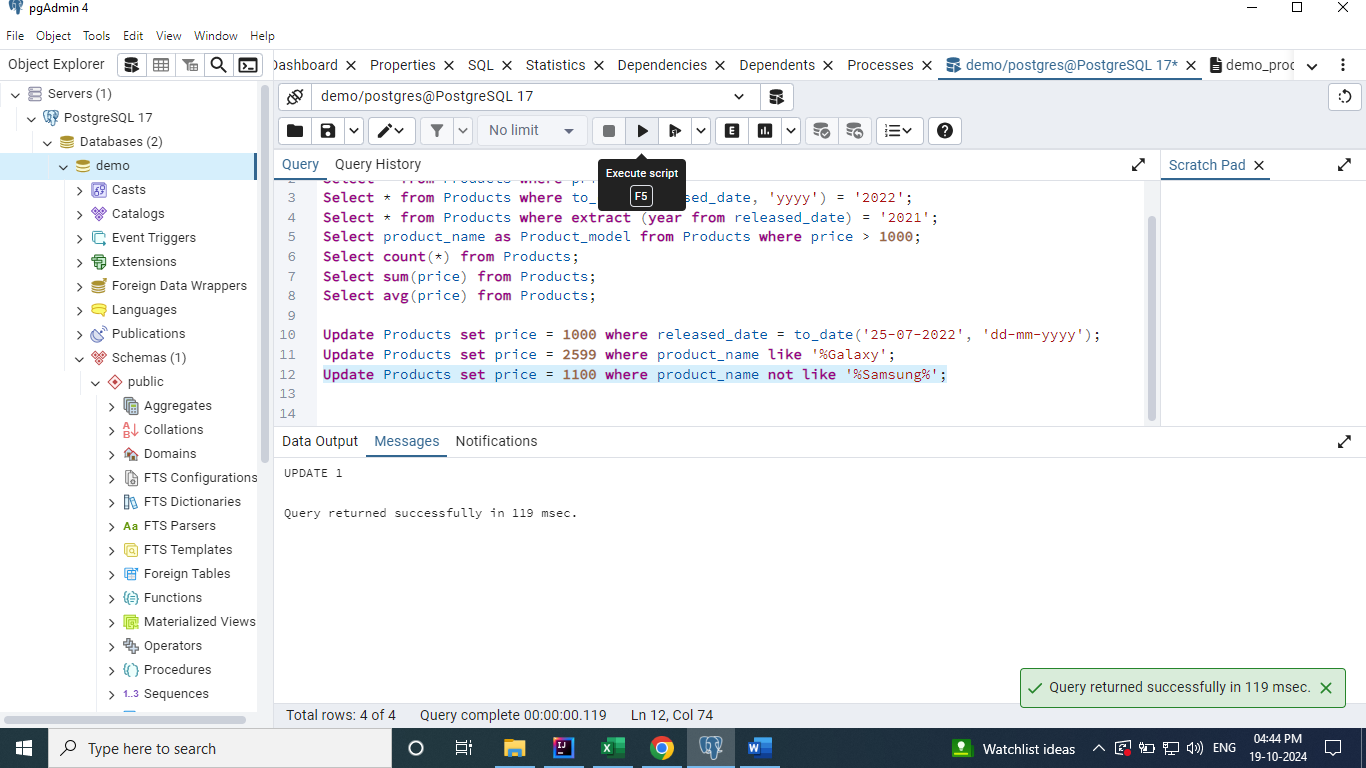
**Not Like keyword**

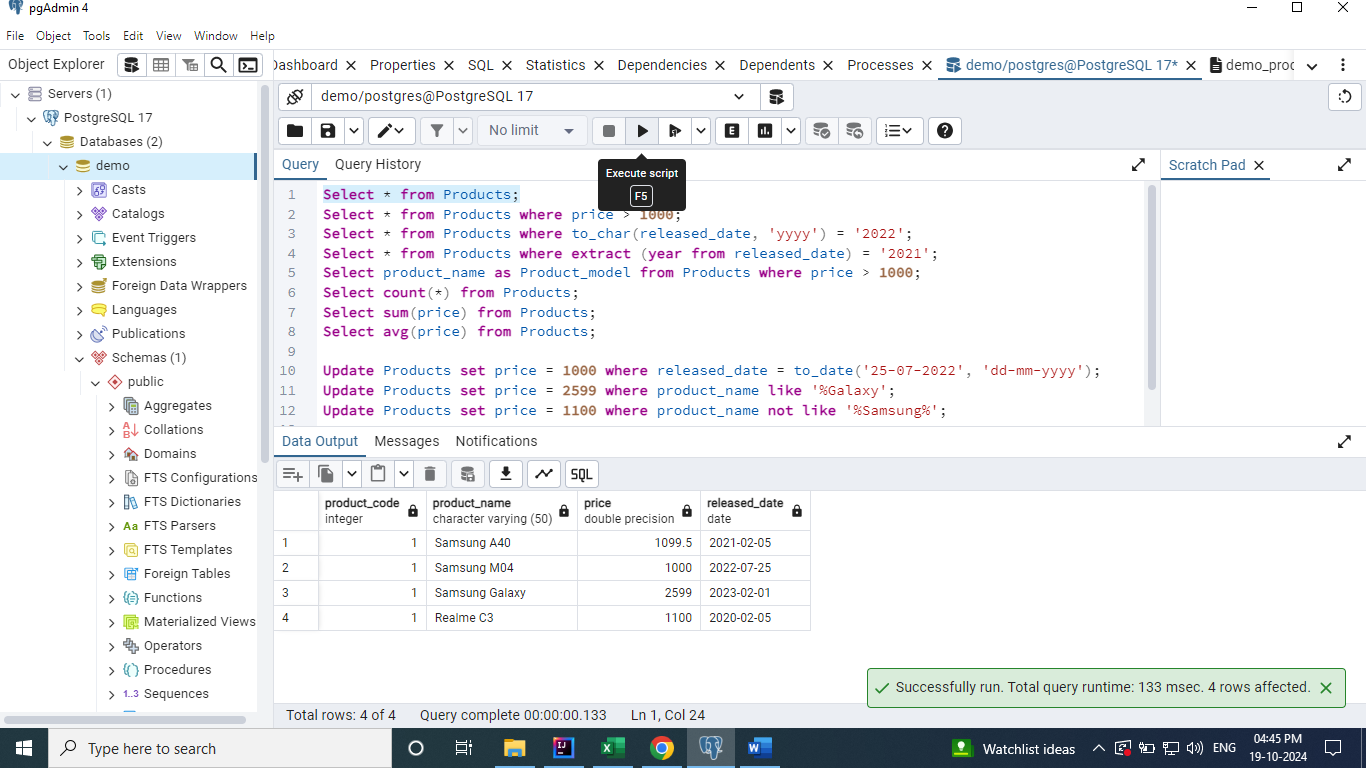
It behaves exactly the reverse of the like keyword.



The below query updates the Realme row’s price.

**Update Products set price = 1100 where product\_name not like '%Samsung%';**





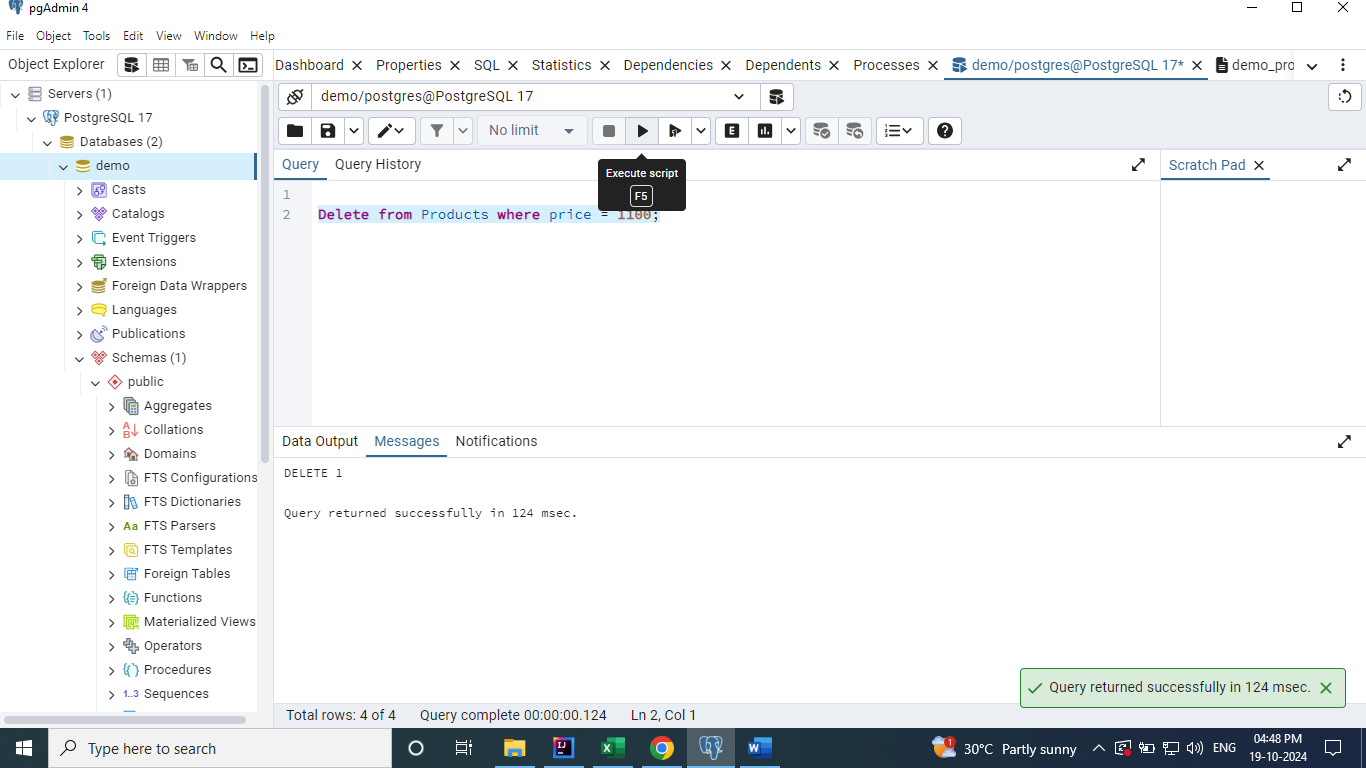
**Delete Operations**

Here, I have used the delete keyword to delete the table records.

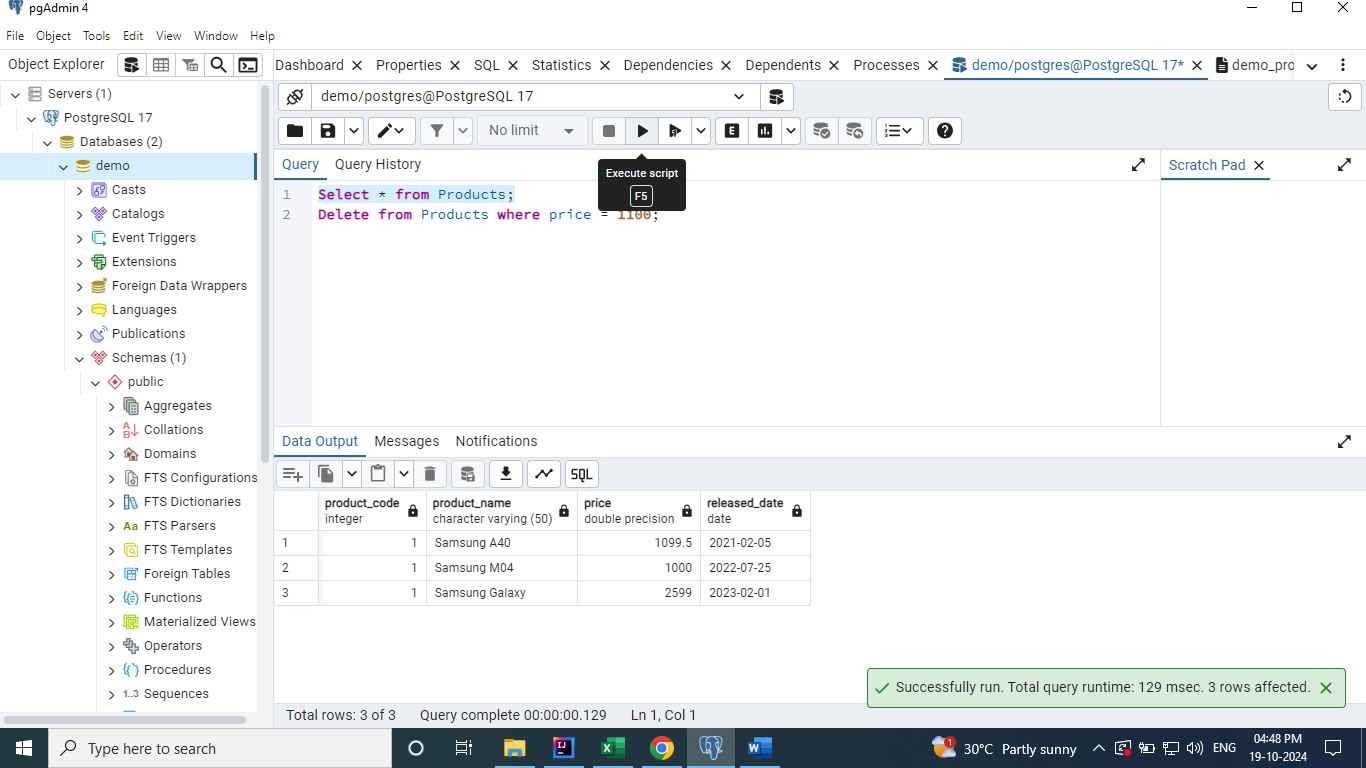
Deleted one of the records of products table by using the where clause. If the where clause is NOT

used in the delete statement, then the delete operation will be applied to all the records.

**Delete from Products where price = 1100;**

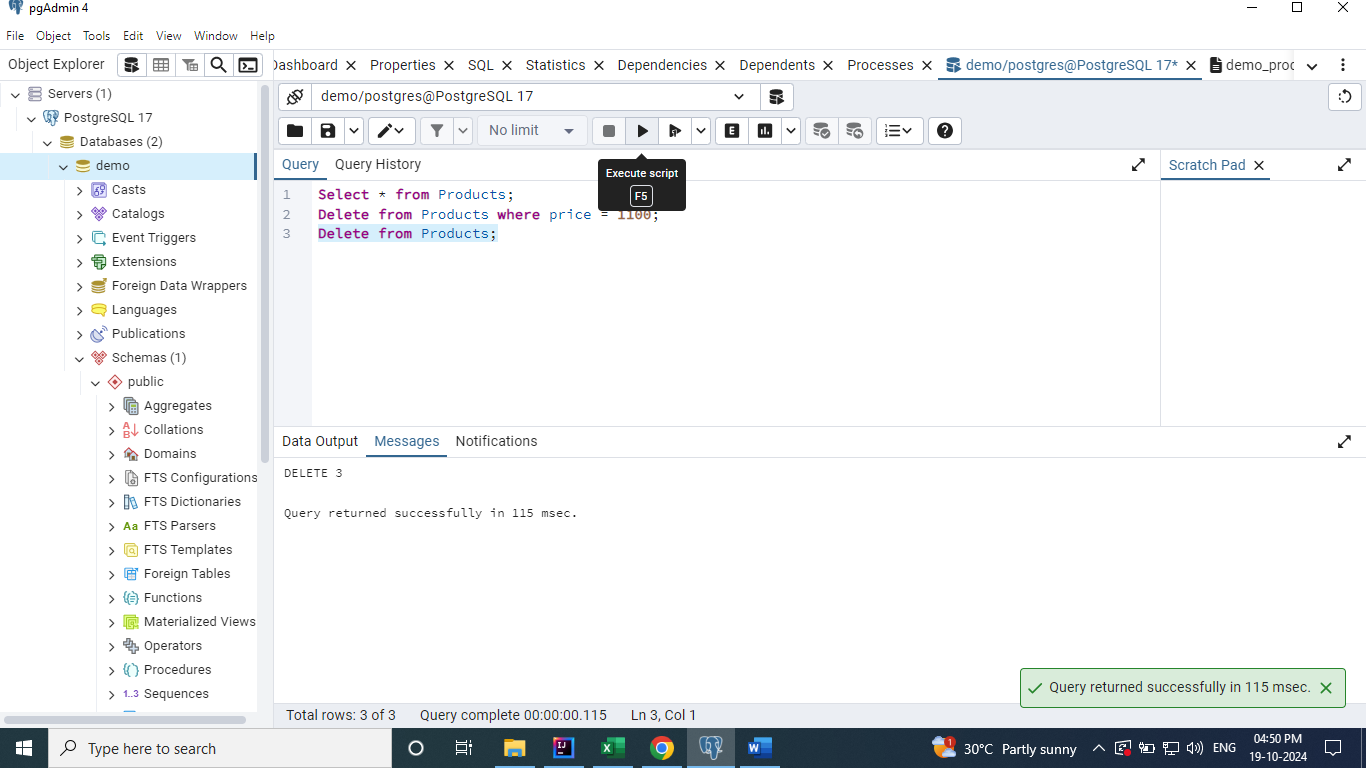


After Delete



**Delete all the records from Products**

**Delete from Products;**



**Truncate – It is used to delete all the records from the table and its very fast than the delete keyword.**

**Truncate table products;**

