**Day2-Assignment**

1) Alter Table:

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

**QUERY:**

ALTER TABLE employees

ADD column linkedin\_profile VARCHAR;

**OUTPUT:**

A screenshot of a message

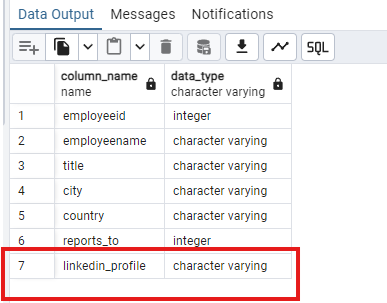
AI-generated content may be incorrect.

To Verify the new column:

select column\_name,data\_type from information\_schema.columns

where table\_schema = 'public' and table\_name='employees';

**OUTPUT:**



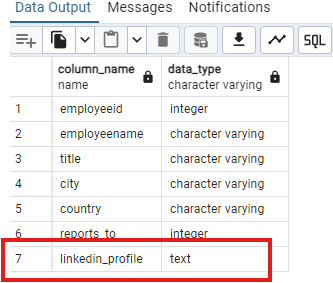
* Change the linkedin\_profile column data type from VARCHAR to TEXT.

**QUERY:**

alter table employees

alter column linkedin\_profile type TEXT;

**OUTPUT:**

****

* Add unique, not null constraint to linkedin\_profile

**QUERY:**

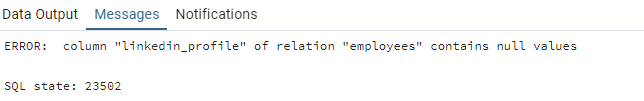
alter table employees

add constraint uk\_linkedin UNIQUE(linkedin\_profile);

alter table employees

alter column linkedin\_profile SET NOT NULL;

OUTPUT:



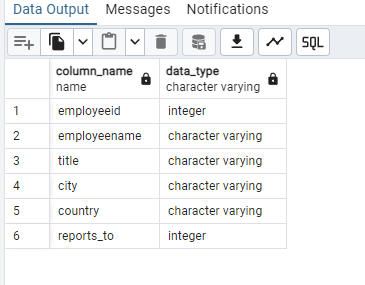
* Drop column linkedin\_profile

**QUERY:**

alter table employees

drop column linkedin\_profile;

**OUTPUT:**



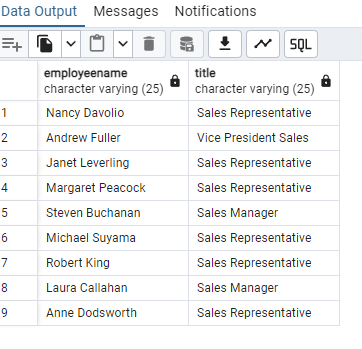
2) Querying (Select)

* Retrieve the employee name and title of all employees

**QUERY:**

select employeename,title from employees;

**OUTPUT:**

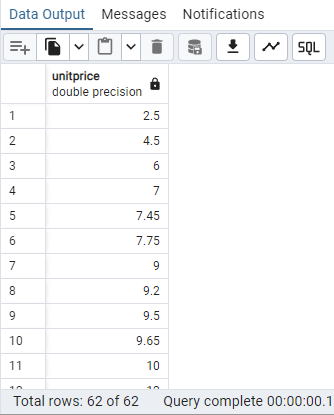


* Find all unique unit prices of products

QUERY:

select DISTINCT unitprice from products order by unitprice ASC;

OUTPUT:



* List all customers sorted by company name in ascending order

QUERY:

select \* from customers order by companyname ASC;

OUTPUT:

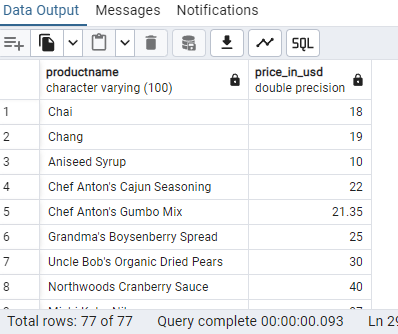


* Display product name and unit price, but rename the unit\_price column as price\_in\_usd

QUERY:

select productname,unitprice as "price\_in\_usd" from products;

OUTPUT:



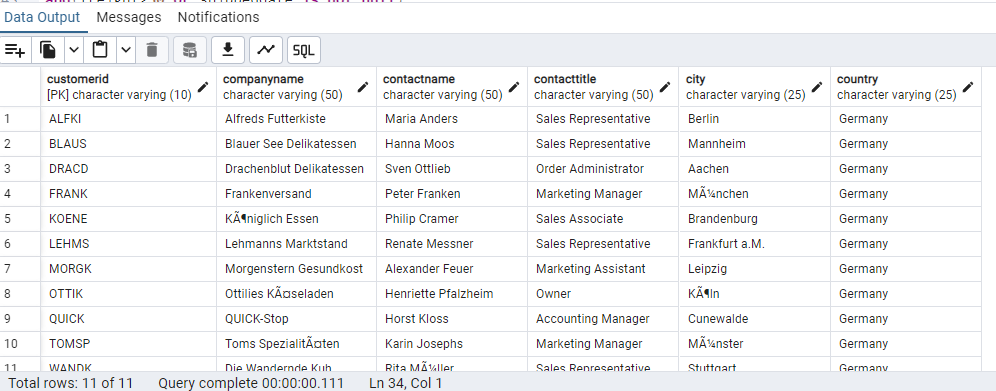
3) Filtering

* Get all customers from Germany.

QUERY:

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

OUTPUT:



* Find all customers from France or Spain

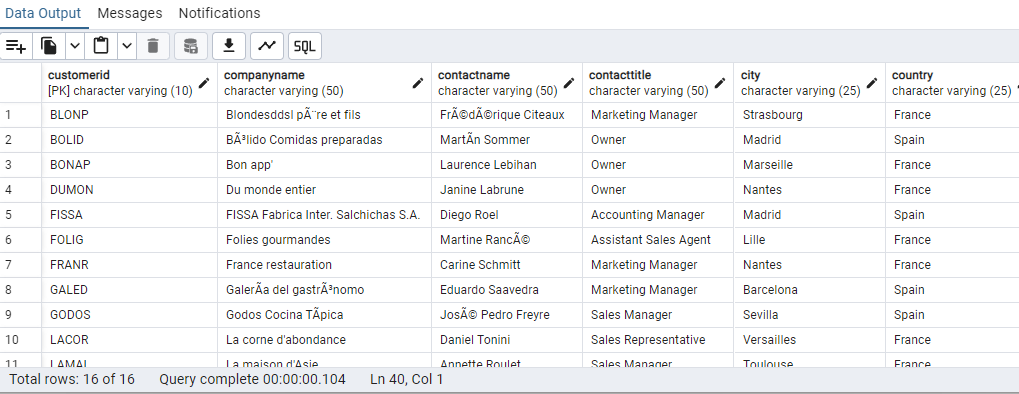
QUERY:

select \* from customers where country in ('France','Spain');

--(OR)

select \* from customers where country='France' or country='Spain';

OUTPUT:



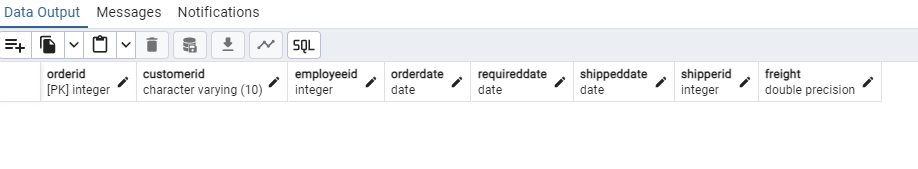
* Retrieve all orders placed in 1997 (based on order\_date), and either have freight greater than 50 or the shipped date available (i.e., non-NULL) (Hint: EXTRACT(YEAR FROM order\_date))

QUERY:

select \* from orders where extract(year from orderdate)=1997

and(freight>50 or shippeddate is not null);

OUTPUT:



There are no records with order date in 1997. So changed the year to 2014 in the same query

QUERY:

select \* from orders where extract(year from orderdate)=2014

and(freight>50 or shippeddate is not null);

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

4) Filtering

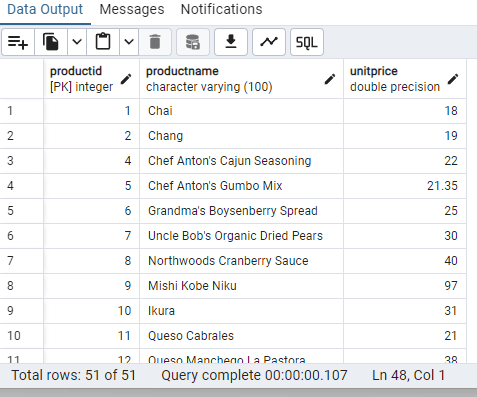
* Retrieve the product\_id, product\_name, and unit\_price of products where the unit\_price is greater than 15.

QUERY:

select productid,productname,unitprice from products

where unitprice>15;

OUTPUT:

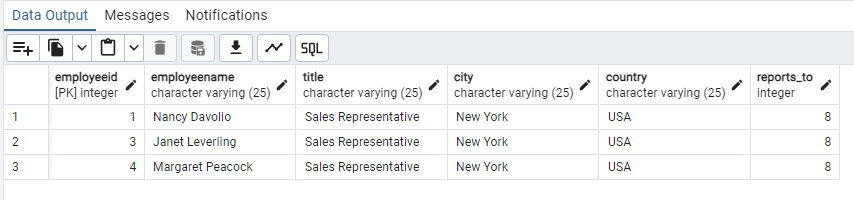


* List all employees who are located in the USA and have the title "Sales Representative".

QUERY:

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

OUTPUT:

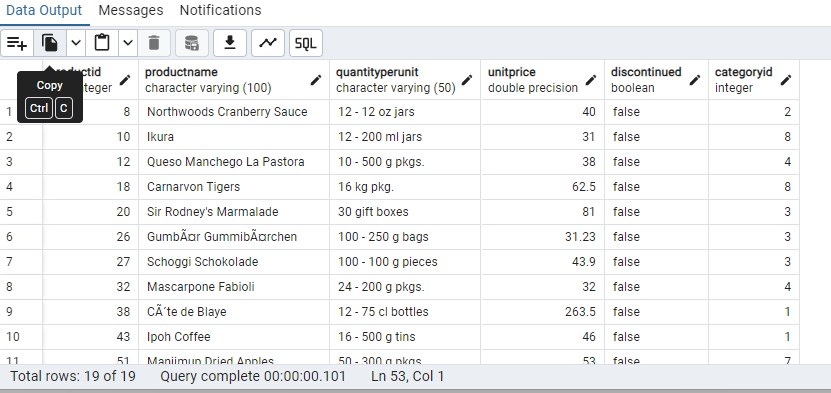


* Retrieve all products that are not discontinued and priced greater than 30.

QUERY:

select \* from products where discontinued = 'false' and unitprice>30;

OUTPUT:



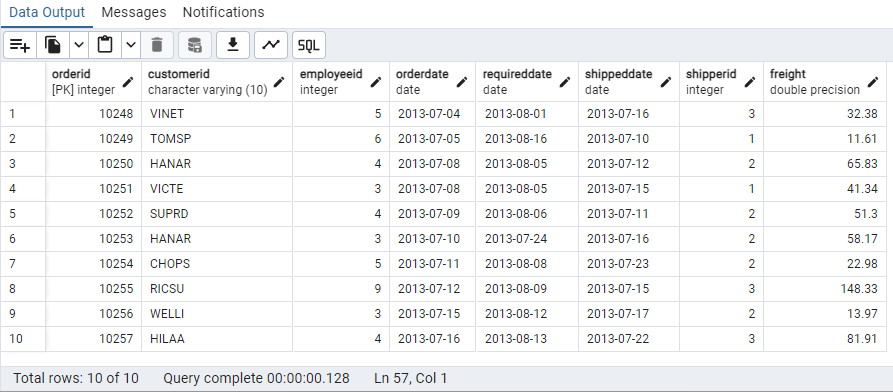
5) LIMIT/FETCH

* Retrieve the first 10 orders from the orders table.

QUERY:

select \* from orders LIMIT 10;

OUTPUT:



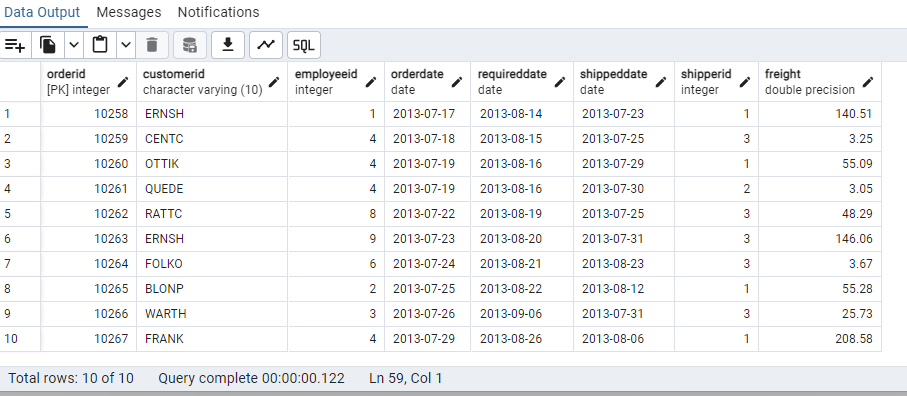
* Retrieve orders starting from the 11th order, fetching 10 rows (i.e., fetch rows 11-20).

QUERY:

select \* from orders offset 10 rows

fetch next 10 rows only;

OUTPUT:



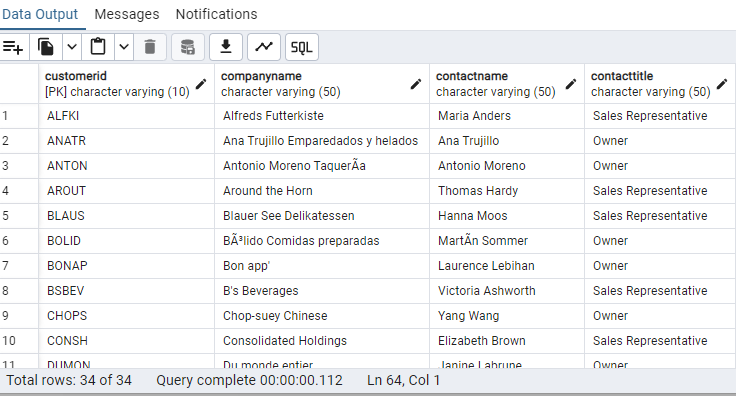
6) Filtering (IN, BETWEEN)

* List all customers who are either Sales Representative, Owner;

QUERY:

select \* from customers where contacttitle in ('Sales Representative','Owner');

OUTPUT:

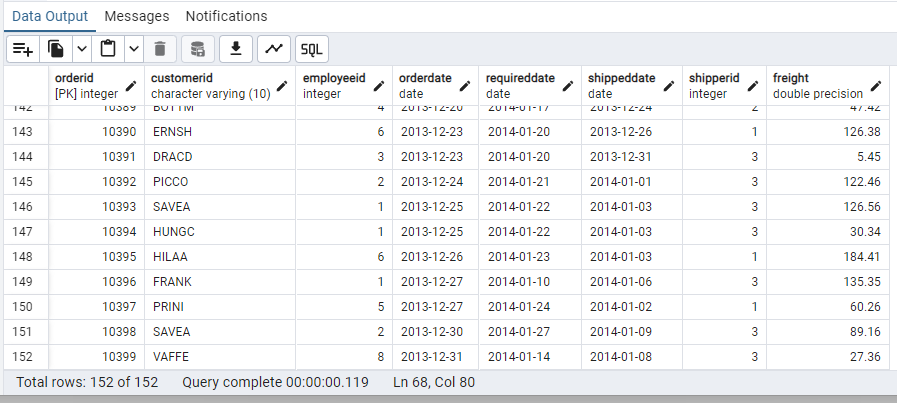


* Retrieve orders placed between January 1, 2013, and December 31, 2013.

QUERY:

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

OUTPUT:



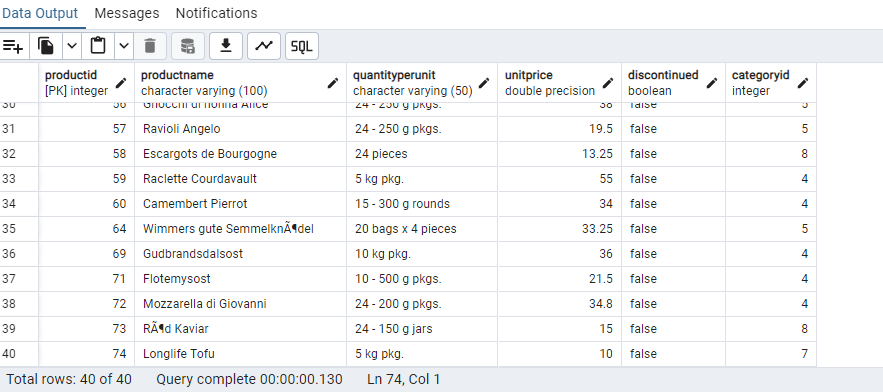
7) Filtering

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

QUERY:

select \* from products where categoryid not in (1,2,3);

OUTPUT:

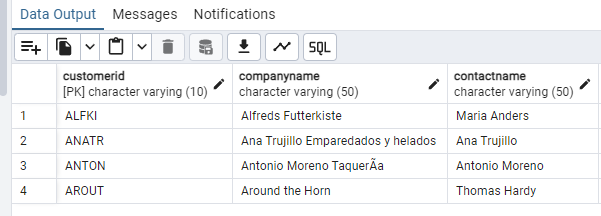


* Find customers whose company name starts with "A".

QUERY:

select \* from customers where companyname like 'A%';

OUTPUT:



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

QUERY:

insert into orders (orderid,customerid,employeeid,orderdate,requireddate,shippeddate,shipperid,freight) values

(11078,'ALFKI',5,'2025-05-23','2025-04-30','2025-04-25',2, 45.50);

OUTPUT:

A screenshot of a message

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

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

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

QUERY:

update products set unitprice = unitprice \* 1.10 where categoryid=2;

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

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’

A screenshot of a phone

AI-generated content may be incorrect.

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

A screenshot of a computer

AI-generated content may be incorrect.