**Day2**

1) Alter Table:

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

alter table "My Schema".employees

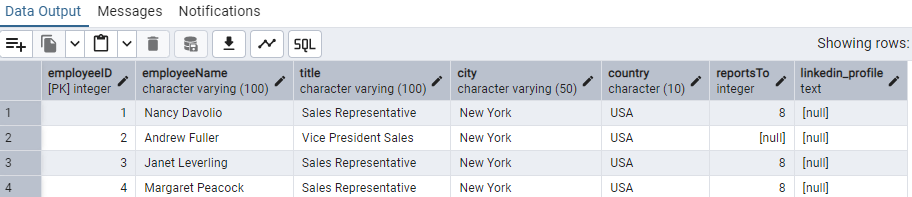
add column linkedin\_profile character varying



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

alter table employees

alter column linkedin\_profile TYPE text



* Add unique, not null constraint to linkedin\_profile

------For Not Null Constraint

update employees

set linkedin\_profile = 'Missing'

where linkedin\_profile is Null

alter table employees

alter column linkedin\_profile set not null

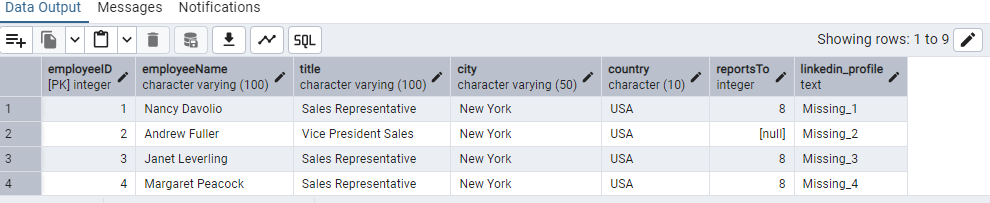
-------For unique constraint

update employees

set linkedin\_profile = 'Missing\_'|| "employeeID"

alter table employees

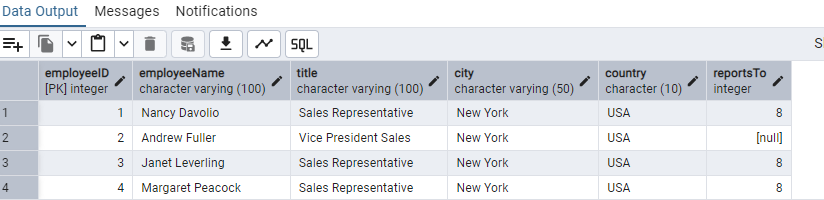
add constraint unique\_profile Unique(linkedin\_profile)



* Drop column linkedin\_profile

alter table employees

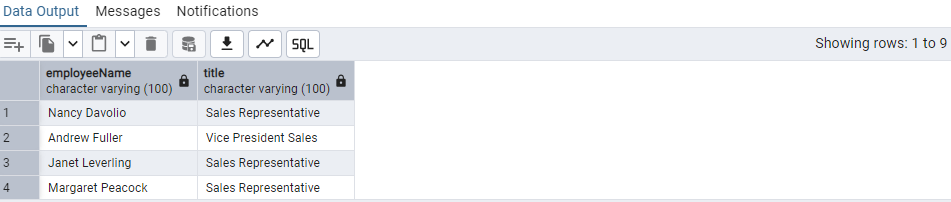
Drop column linkedin\_profile



2) Querying (Select)

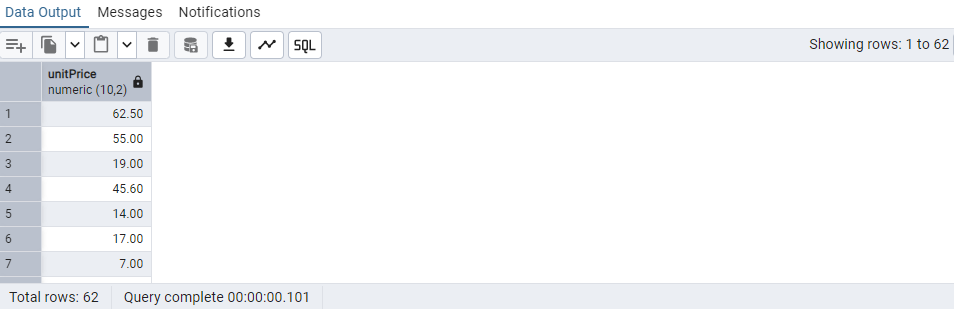
* Retrieve the first name, last name, and title of all employees

select "employeeName","title" from employees



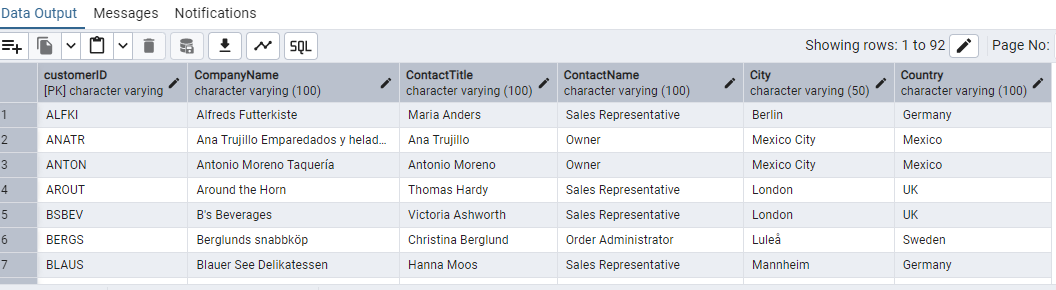
* Find all unique unit prices of products

select distinct "unitPrice" from products



* List all customers sorted by company name in ascending order

select \* from customers order by "CompanyName" ASC

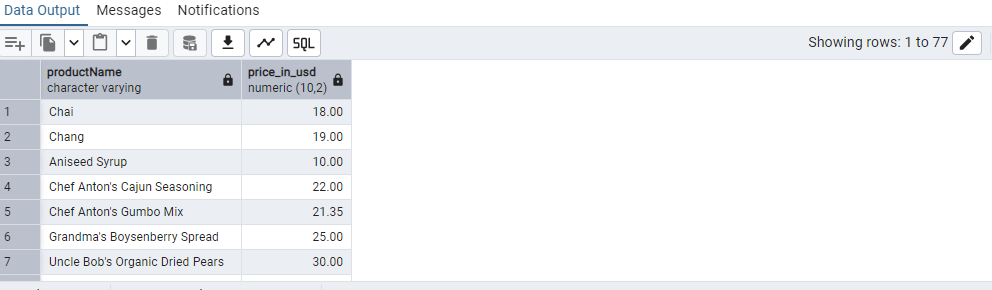


* Display product name and unit price, but rename the unit\_price column as price\_in\_usd

alter table products

rename "unitPrice" to "price\_in\_usd"

select "productName","price\_in\_usd" from products

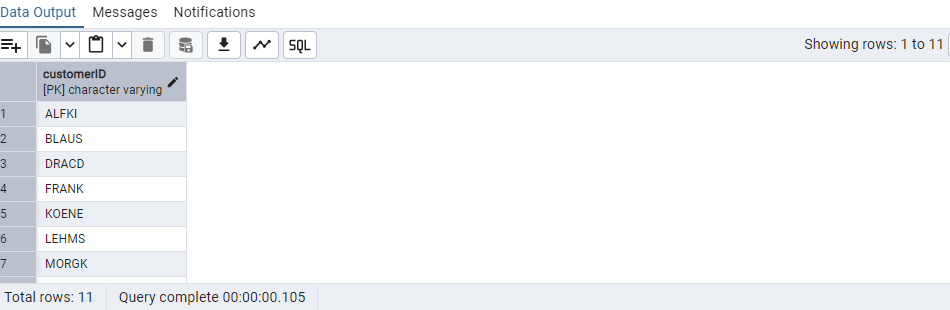


3) Filtering

* Get all customers from Germany.

select "customerID" from customers

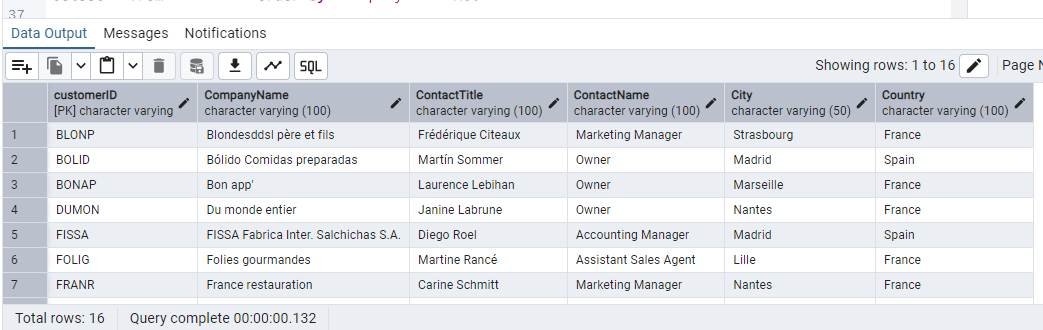
where "Country" = 'Germany'



* Find all customers from France or Spain

select \* from customers

where "Country" In ('France','Spain')

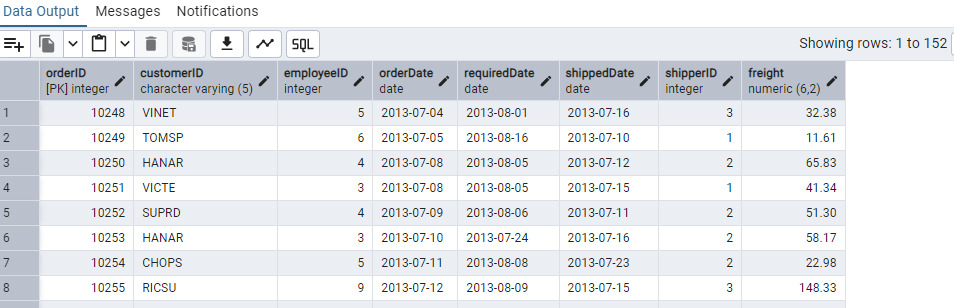


* Retrieve all orders placed in 1997 (based on order\_date), and either have freight greater than 50 or the shipped date available (i.e., non-NULL) (Hint: EXTRACT(YEAR FROM order\_date))

Select \* from orders

Where EXTRACT(YEAR FROM "orderDate") = 2013

AND ("freight" > 50 OR "shippedDate" IS NOT NULL)

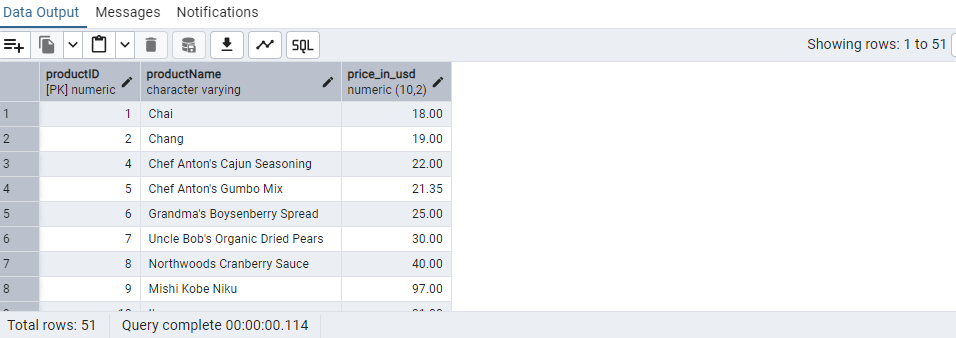


4) Filtering

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

select "productID","productName","price\_in\_usd" from products

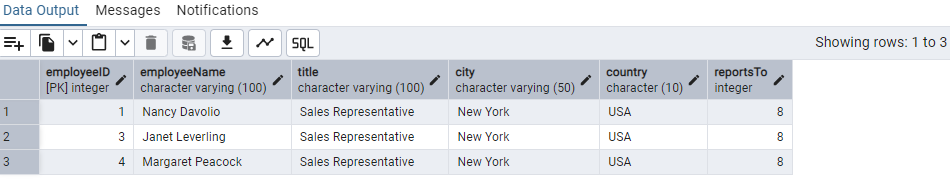
where ("price\_in\_usd">15)



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

select \* from employees

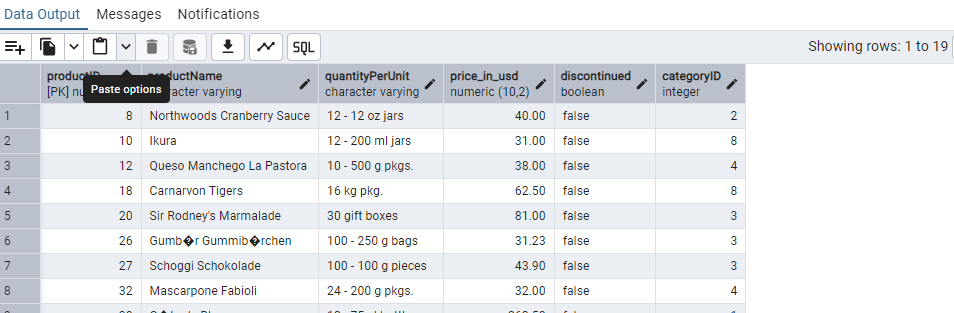
where("country"='USA' and "title"= 'Sales Representative')



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

select \* from products

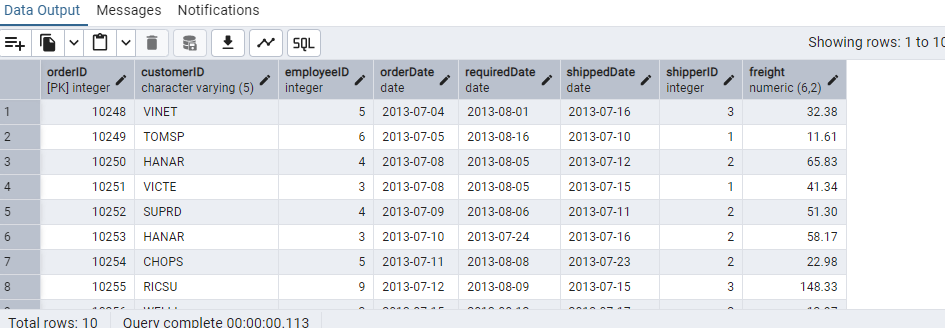
where ("discontinued" = 'false' and "price\_in\_usd" > 30)



5) LIMIT/FETCH

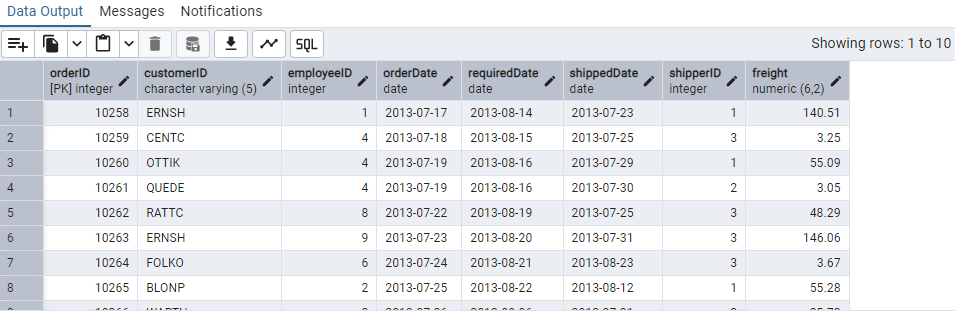
* Retrieve the first 10 orders from the orders table.

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

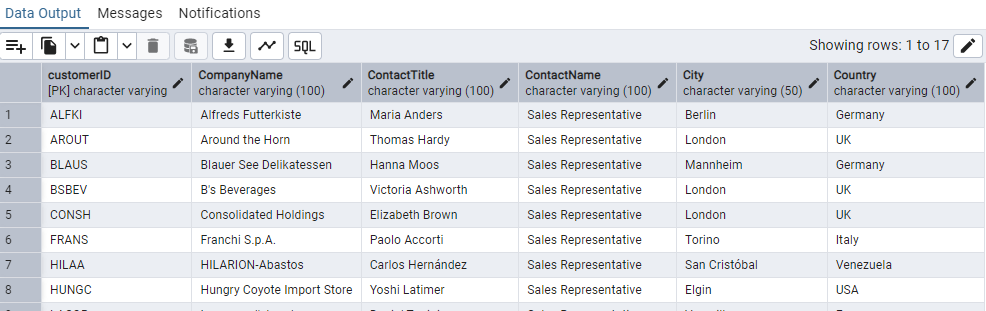


6) Filtering (IN, BETWEEN)

* List all customers who are either Sales Representative, Owner

select \* from customers

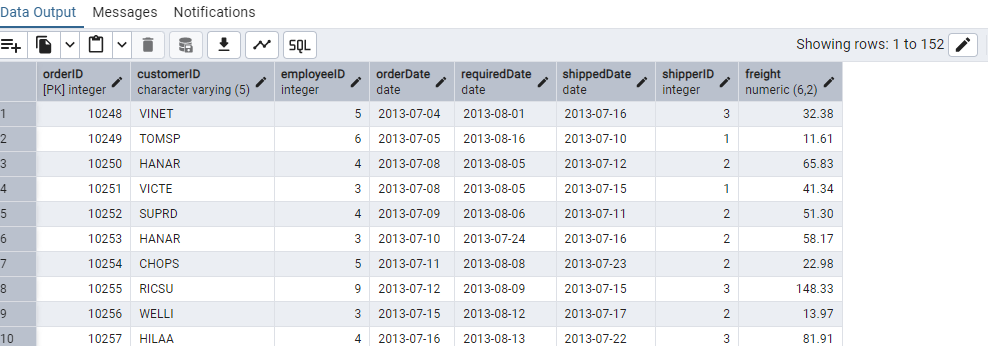
where "ContactName" in ('Sales Representative','owner')



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

select \* from orders

where "orderDate" between '2013-01-01' and '2013-12-31'

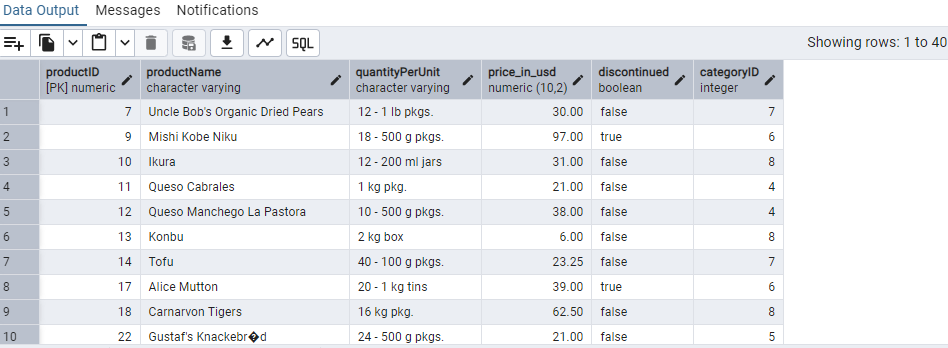


7) Filtering

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

select \* from products

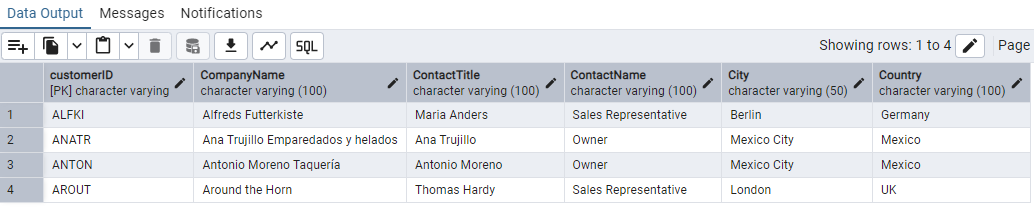
where "categoryID" not in(1,2,3)



* 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

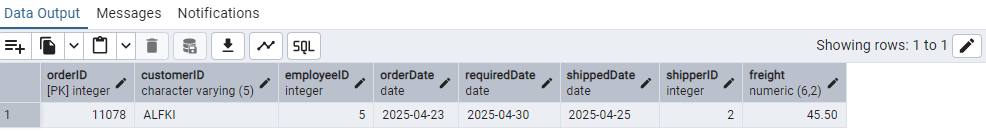
Shipped Date: 2025-04-25

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)



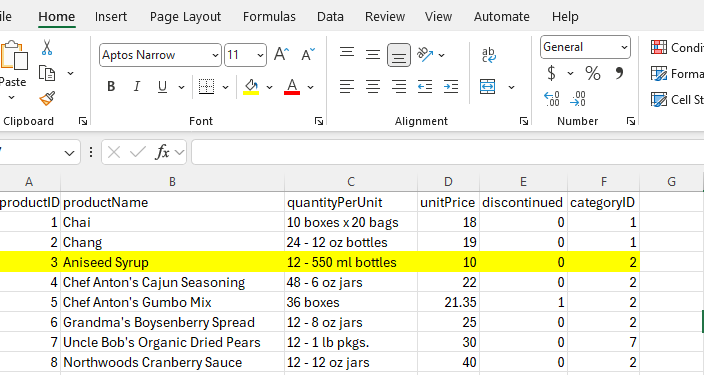
9) Increase(Update) the unit price of all products in category\_id =2 by 10%.

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

Update products

set "price\_in\_usd" = "price\_in\_usd" \* 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

A screenshot of a computer screen

AI-generated content may be incorrect.