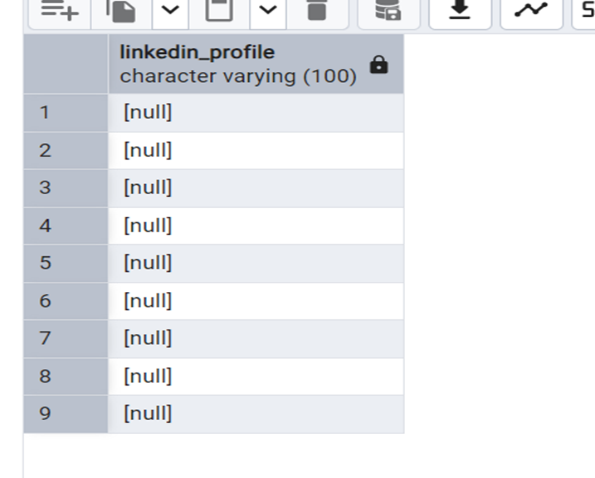
Day 2 sql Assignment

-- 1) Alter Table:

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

ALTER Table employees

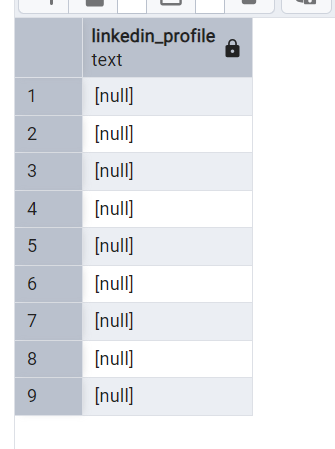
ADD column linkedin\_profile varchar(100);



-- 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

UPDATE employees

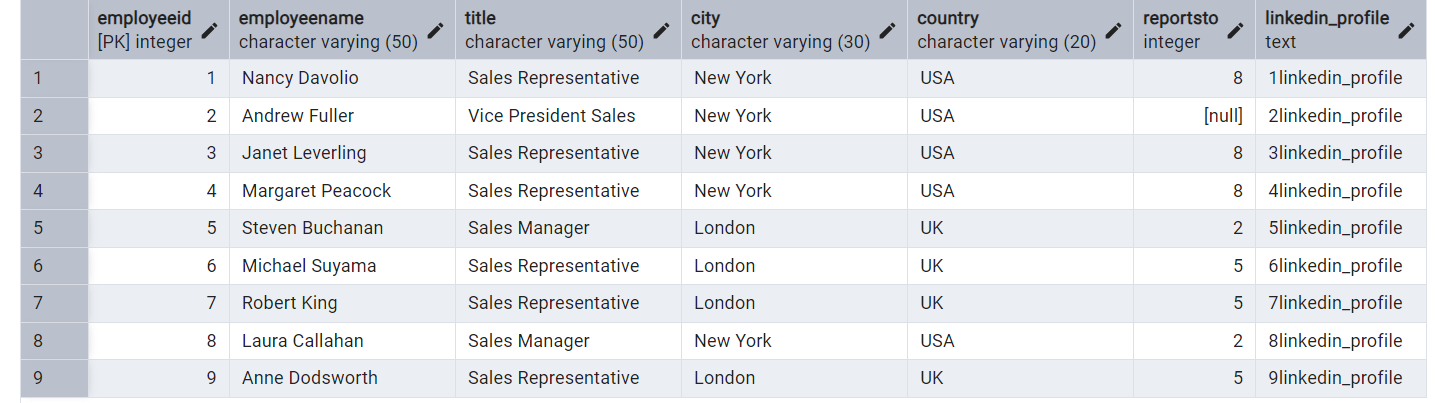
SET linkedin\_profile = employeeid || 'linkedin\_profile';

ALTER Table employees

ADD CONSTRAINT linkedin\_profile\_const UNIQUE (linkedin\_profile);

ALTER TABLE employees

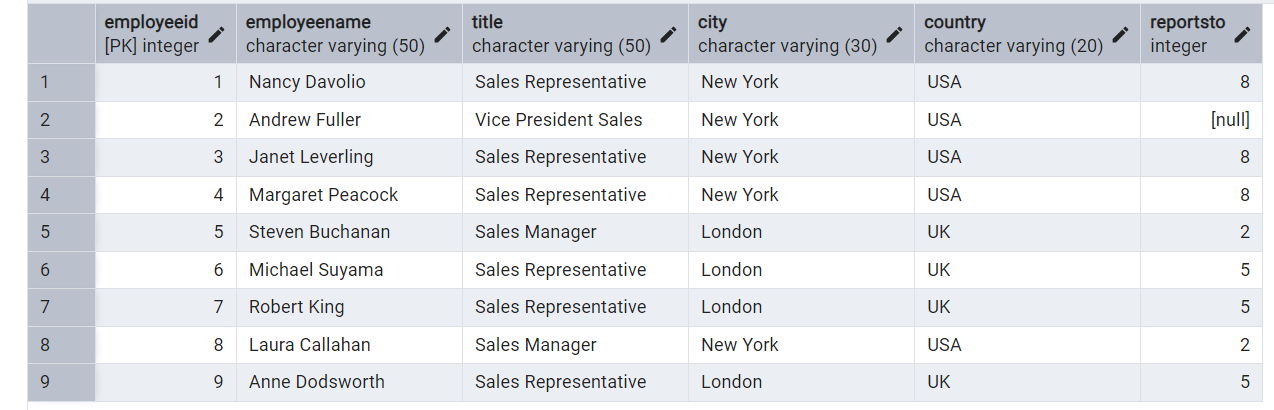
ALTER COLUMN linkedin\_profile SET NOT NULL;



--Drop column linkedin\_profile

ALTER TABLE employees

DROP COLUMN linkedin\_profile;



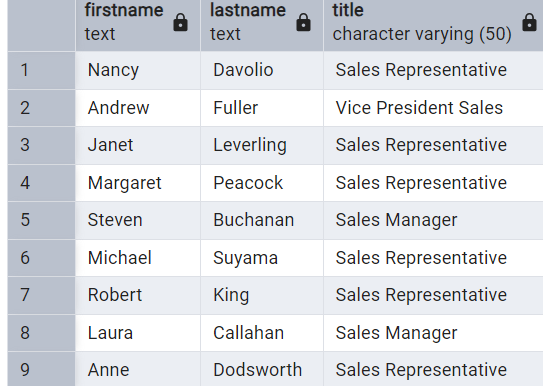
2)      Querying (Select)

* Retrieve the employee name and title of all employees

select split\_part(employeename,' ',1) as firstname ,

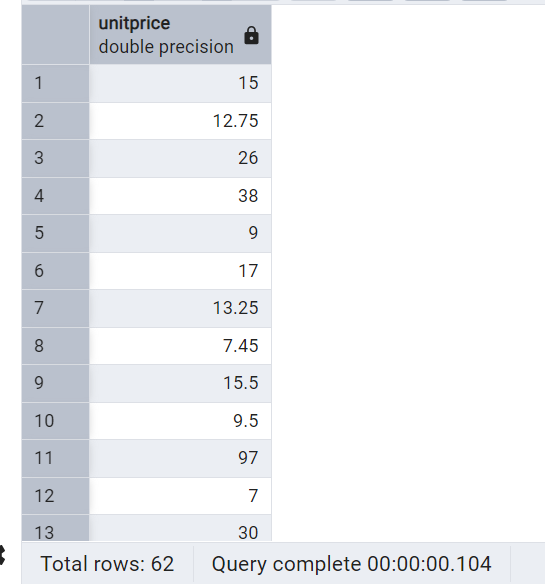
split\_part(employeename, ' ',2) as lastname ,title

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

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 In('France' , 'Spain');



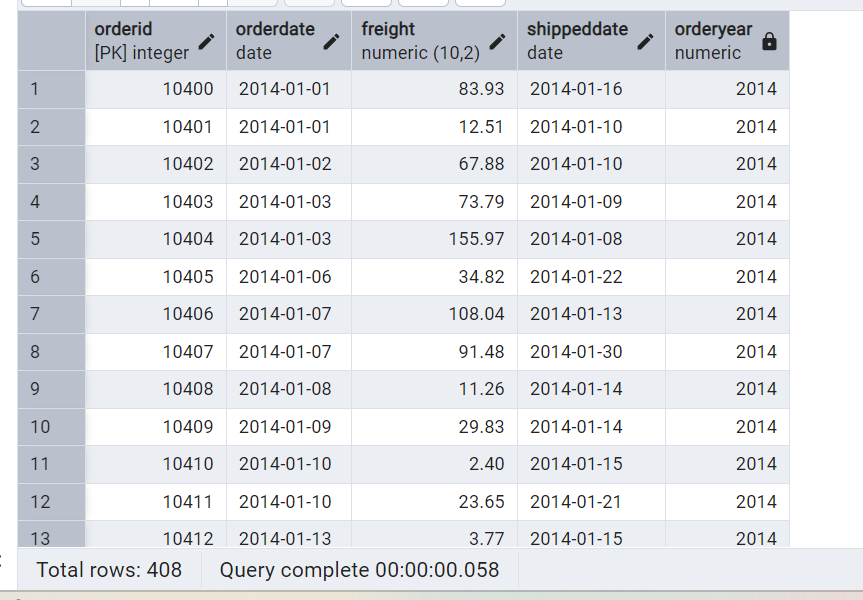
* Retrieve all orders placed in 2014(based on order\_date), and either have freight greater than 50 or the shipped date available (i.e., non-NULL)  (Hint: EXTRACT(YEAR FROM order\_date))

select orderid, orderdate,freight,shippeddate,

EXTRACT (year from orderdate) as orderYear from orders

where DATE\_PART('year',orderdate )= 2014

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,unitprice from products

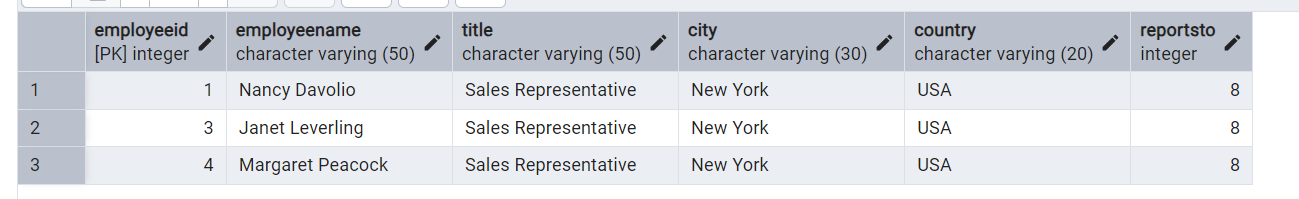
where unitprice > 15 ;



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

select \* from employees

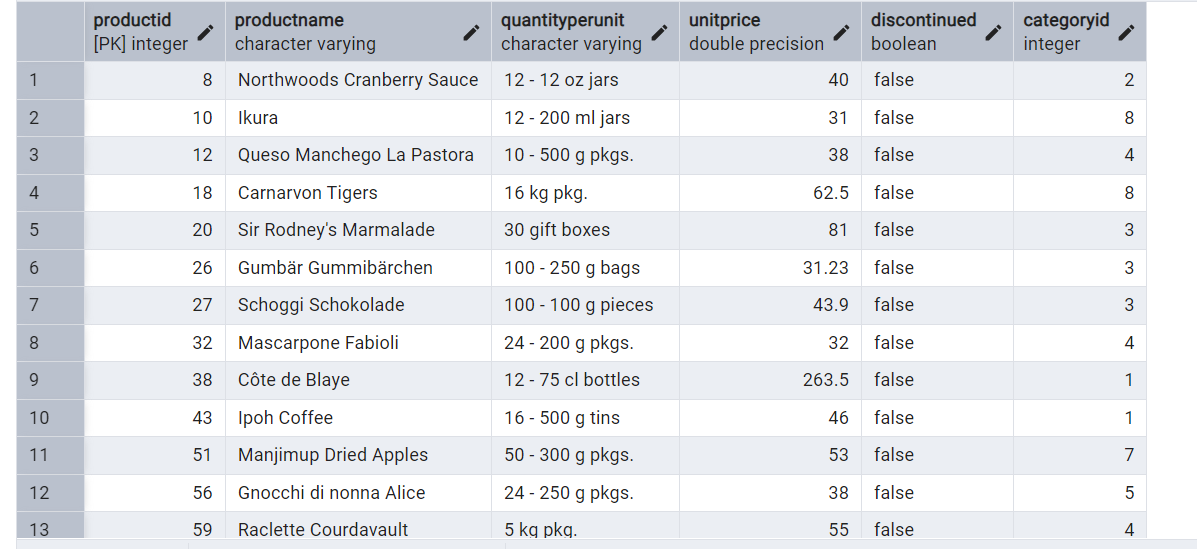
where country = 'USA' AND title = 'Sales Representative';



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

select \* from products

where discontinued = false AND unitprice > 30;

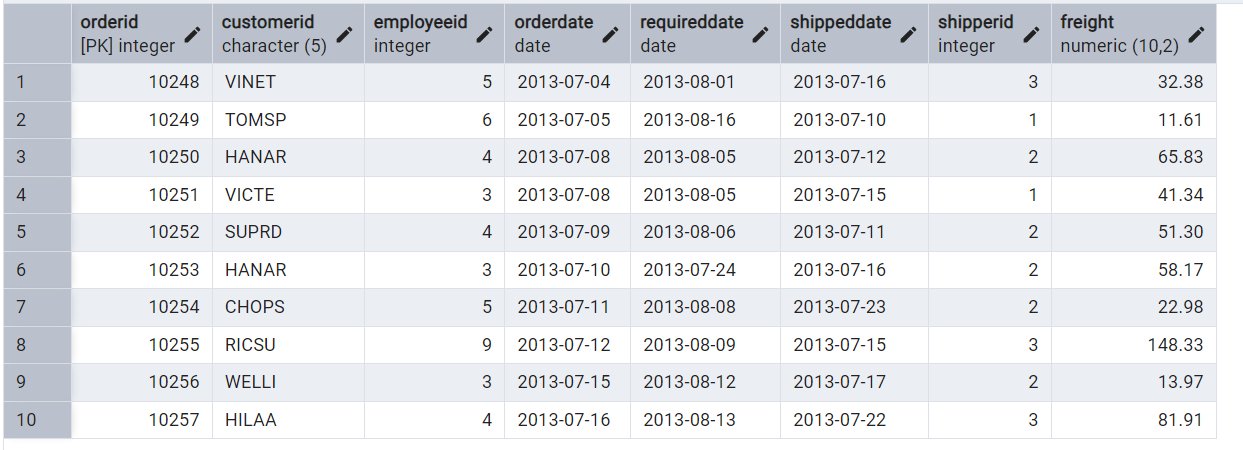


5)      LIMIT/FETCH

* Retrieve the first 10 orders from the orders table.

select \* from orders select \* from orders

LIMIT 10; (or) FETCH FIRST 10 rows only;

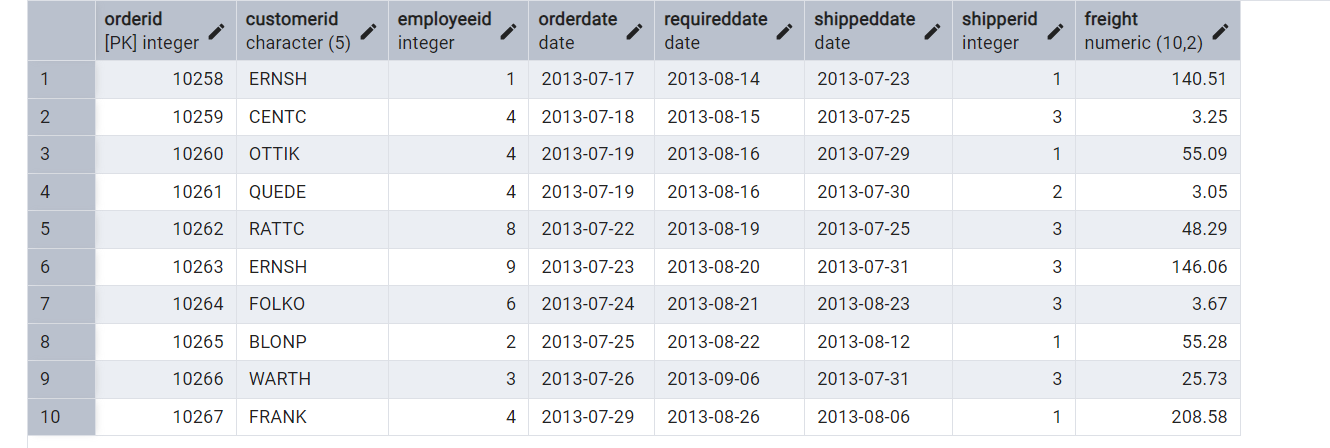


* Retrieve orders starting from the 11th order, fetching 10 rows (i.e., fetch rows 11-20).

 select \* from orders

offset 10 rows

FETCH NEXT 10 ROWS ONLY

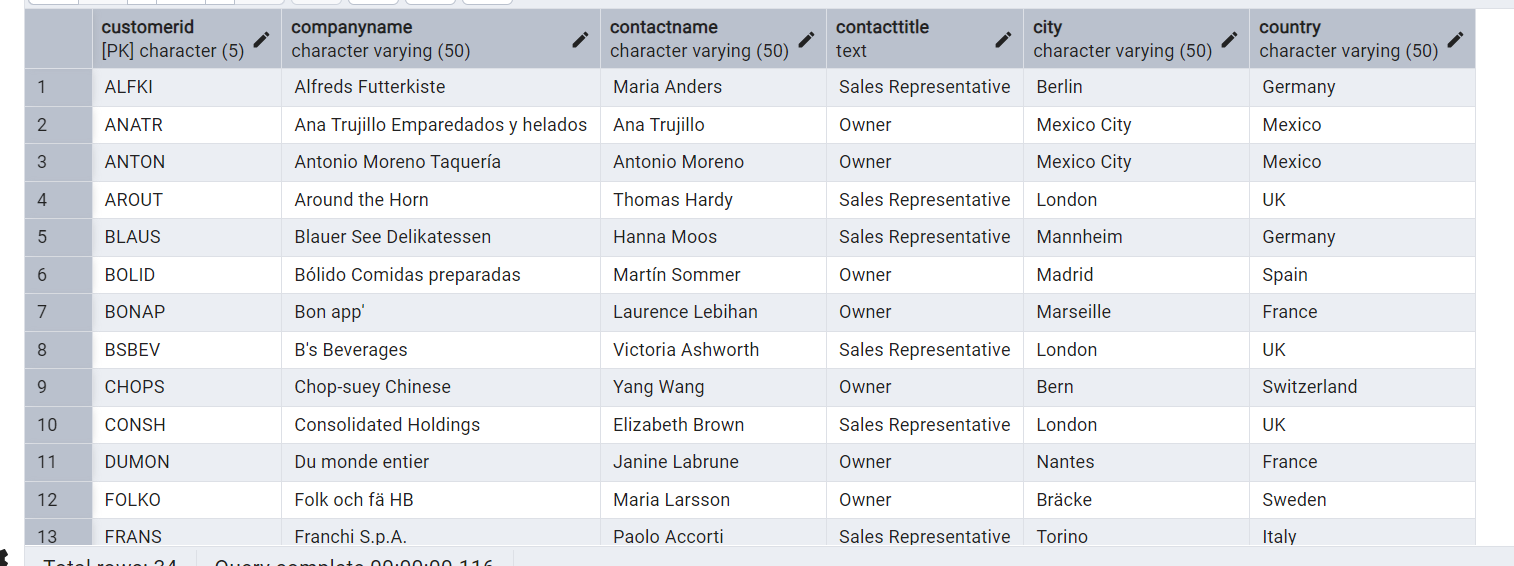


6)      Filtering (IN, BETWEEN)

* List all customers who are either Sales Representative, Owner

select \* from customers

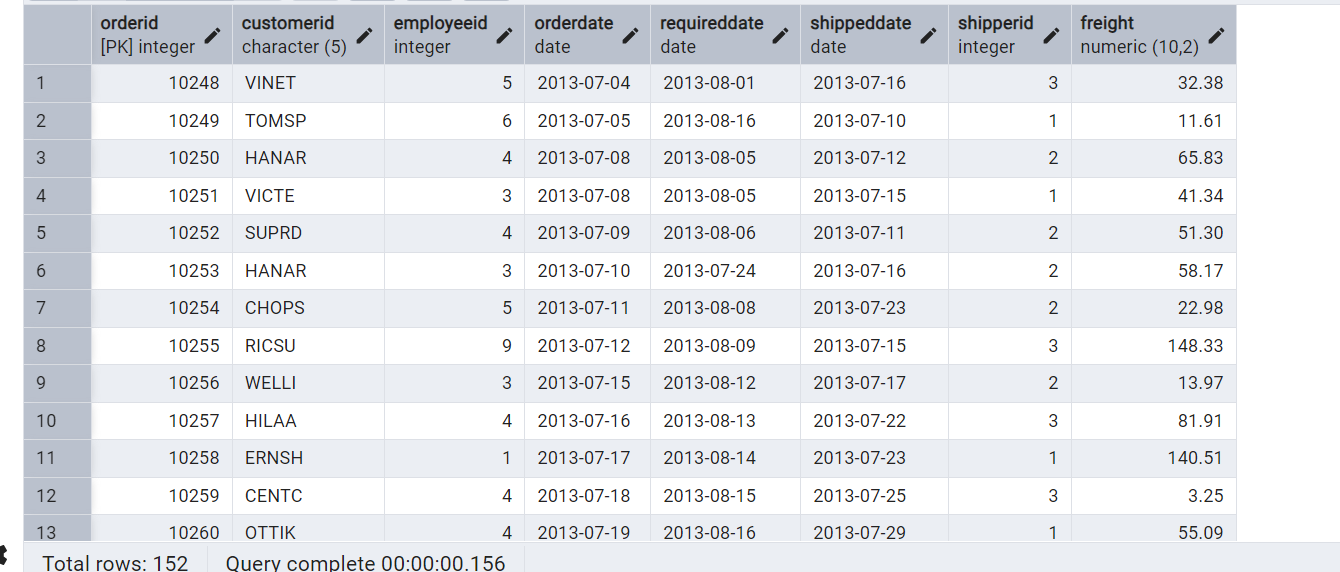
where contacttitle IN ('Sales Representative' , 'Owner');



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

select \* from orders

where orderdate BETWEEN '2013-01-01' AND '2013-12-31';

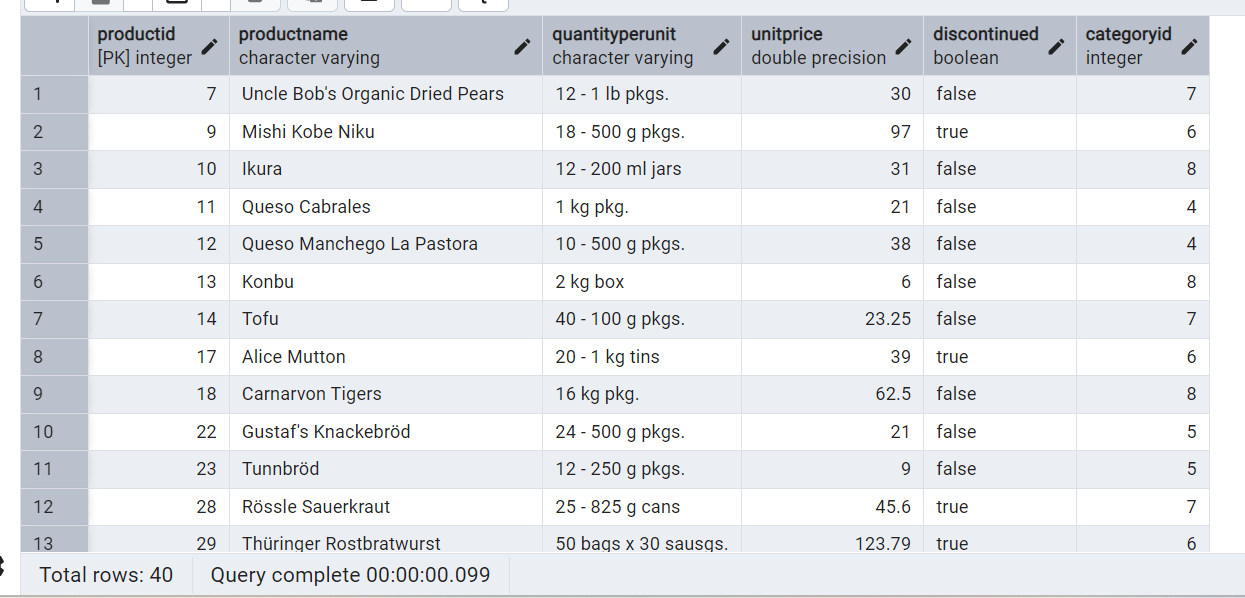


7)      Filtering

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

select \* from products

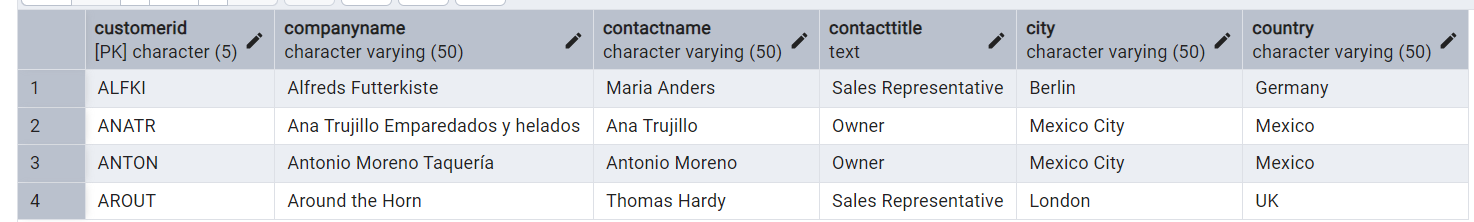
where categoryid NOT IN (1, 2, 3);



* 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

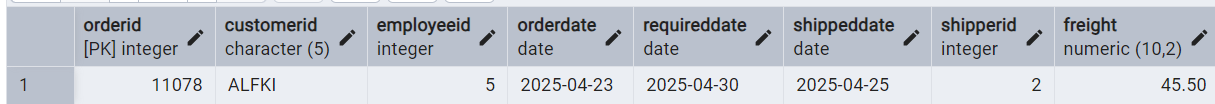
Shipped Date: 2025-04-25

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);



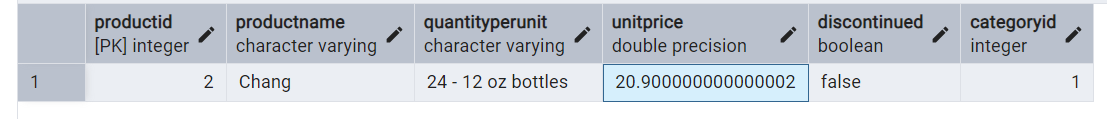
9)      Increase(Update)  the unit price of all products in category\_id =2 by 10%.

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

UPDATE products

set unitprice = unitprice \* 1.10

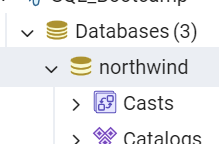
where productid = 2;



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

