/\*--1)Alter Table:

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

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

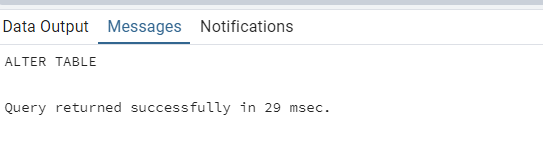
Add unique, not null constraint to linkedin\_profile

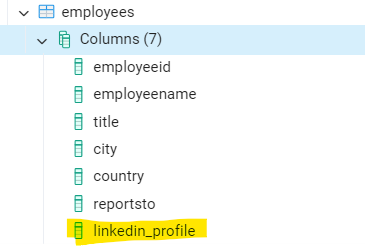
Drop column linkedin\_profile --\*/

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

ALTER TABLE employees

ADD COLUMN linkedin\_profile VARCHAR;

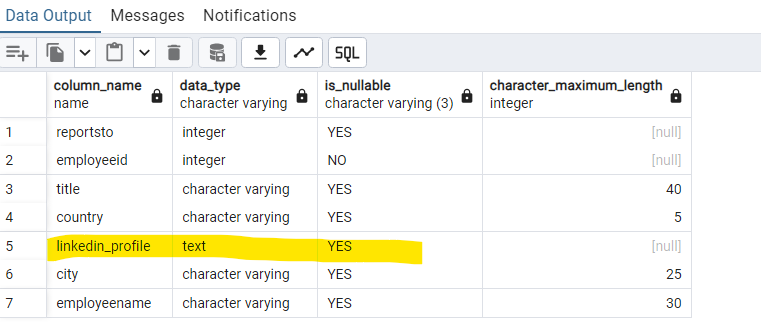




--Change the linkedin\_profile column data type from VARCHAR to TEXT.

ALTER TABLE employees

ALTER COLUMN linkedin\_profile TYPE TEXT;



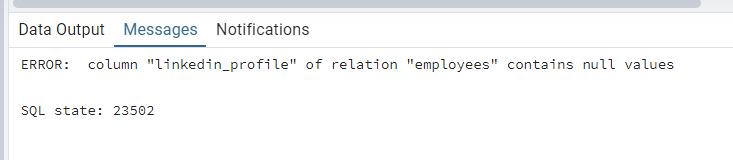
-- Add unique, not null constraint to linkedin\_profile

ALTER TABLE employees

ADD CONSTRAINT unique\_linkedin\_profile UNIQUE (linkedin\_profile);

ALTER TABLE employees

ALTER COLUMN linkedin\_profile SET NOT NULL;

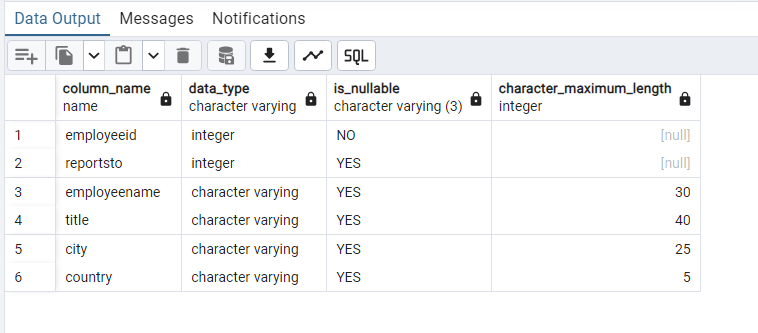


Command failed because linkedin\_profile is blank, values not inserted yet.

--Drop column linkedin\_profile

ALTER TABLE employees

DROP COLUMN linkedin\_profile;



/\*--2) Querying (Select)

Retrieve the employee name and title of all employees

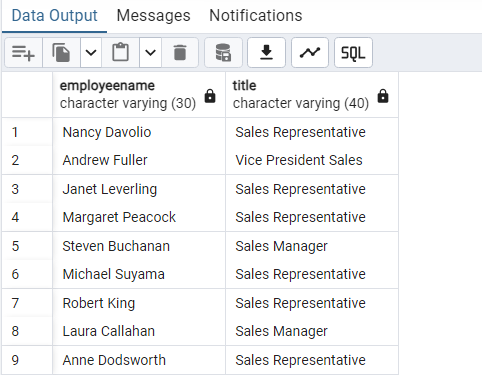
Find all unique unit prices of products

List all customers sorted by company name in ascending order

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

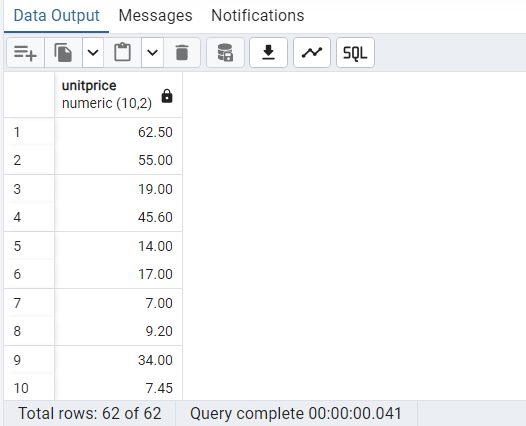
--Retrieve the employee name and title of all employees

SELECT employeename, title FROM employees;



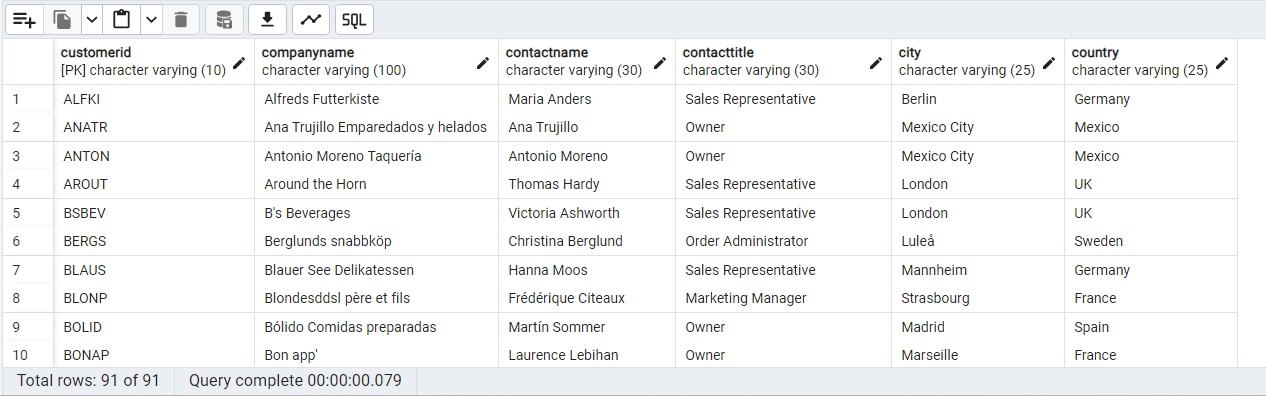
-- Find all unique unit prices of products

SELECT DISTINCT unitprice FROM products;



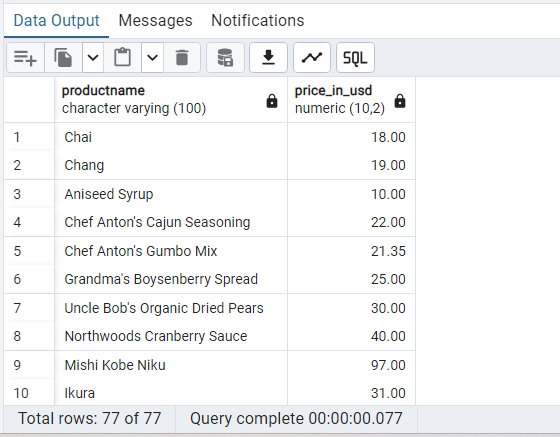
--List all customers sorted by company name in ascending order

SELECT \* FROM customers ORDER BY companyname 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.

Find all customers from France or Spain

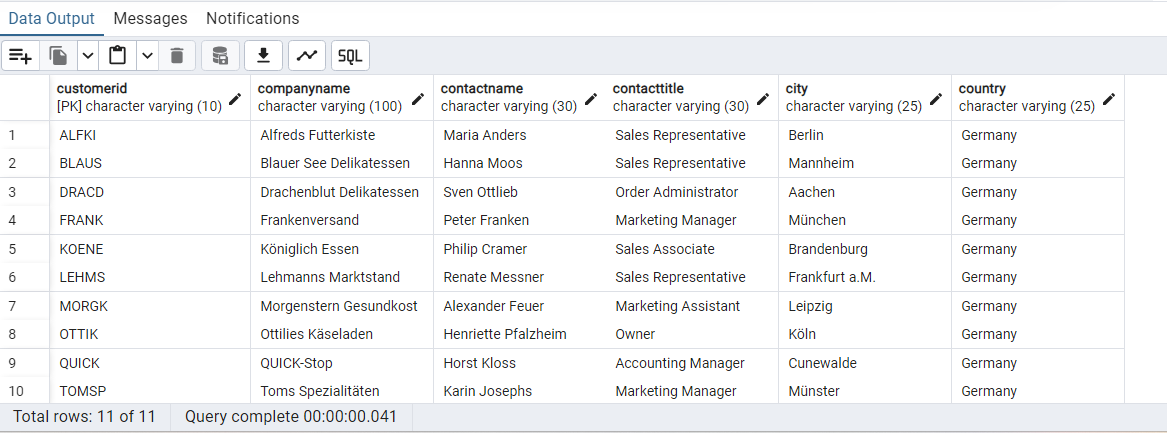
Retrieve all orders placed in 2014 (based on order\_date), and either have freight greater than 50 or the shipped date available (i.e., non-NULL)

(Hint: EXTRACT(YEAR FROM order\_date)) --\*/

--Get all customers from Germany.

SELECT \* FROM customers

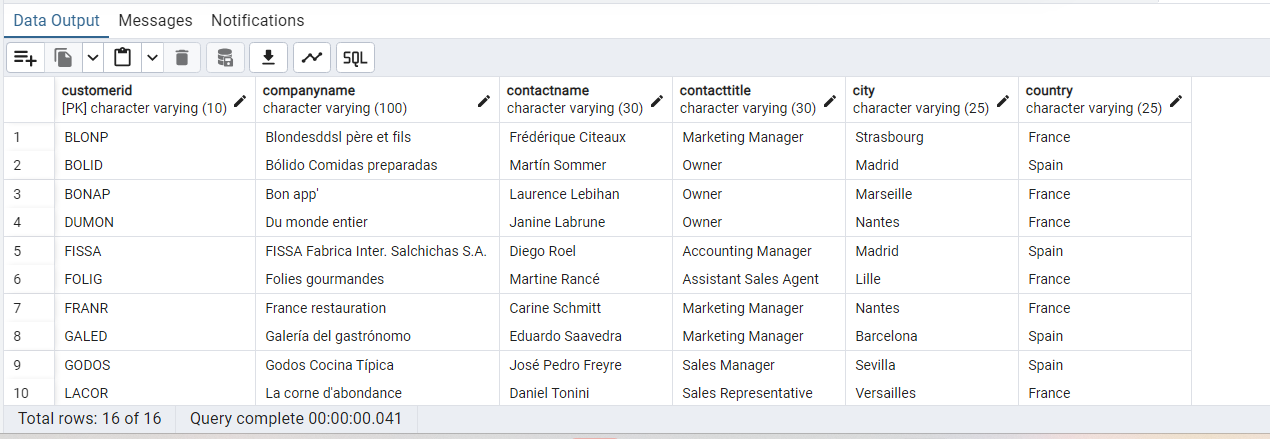
WHERE country = 'Germany';



--Find all customers from France or Spain

SELECT \* FROM customers

WHERE country IN ('France', 'Spain');

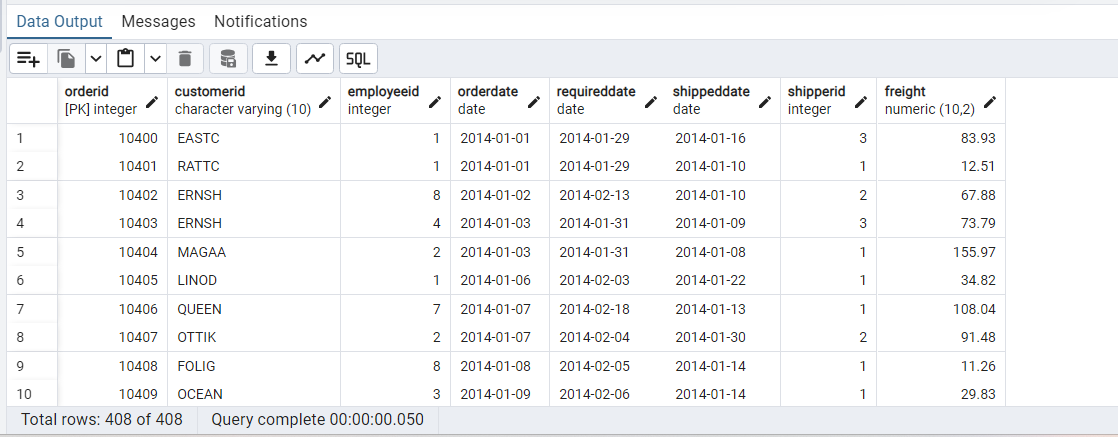


--Retrieve all orders placed in 2014 (based on order\_date), and either have freight greater than 50 or the shipped date available (i.e., non-NULL)

SELECT \* FROM orders

WHERE EXTRACT(YEAR FROM 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.

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

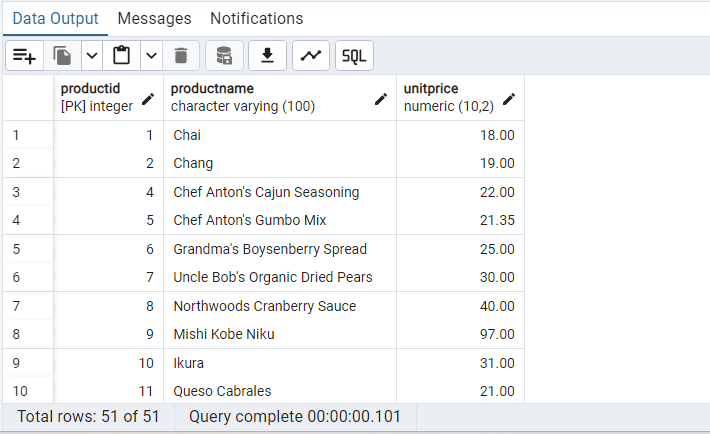
Retrieve all products that are not discontinued and priced greater than 30. \*\*/

-- 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

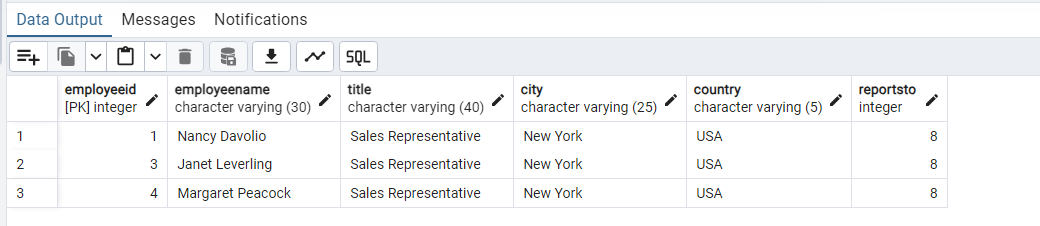
WHERE unitprice > 15;



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

SELECT \* FROM employees

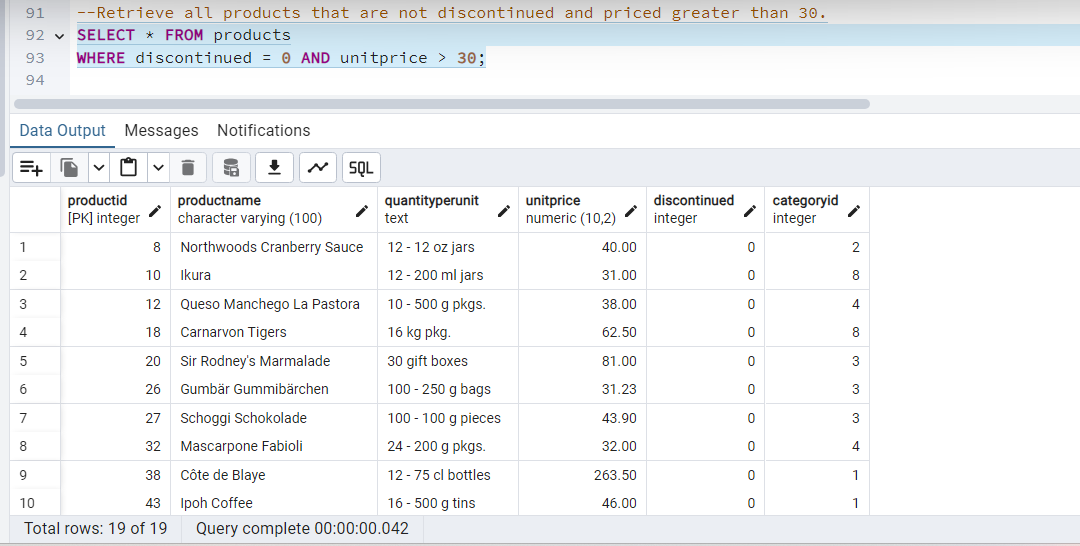
WHERE country = 'USA' AND title = 'Sales Representative';

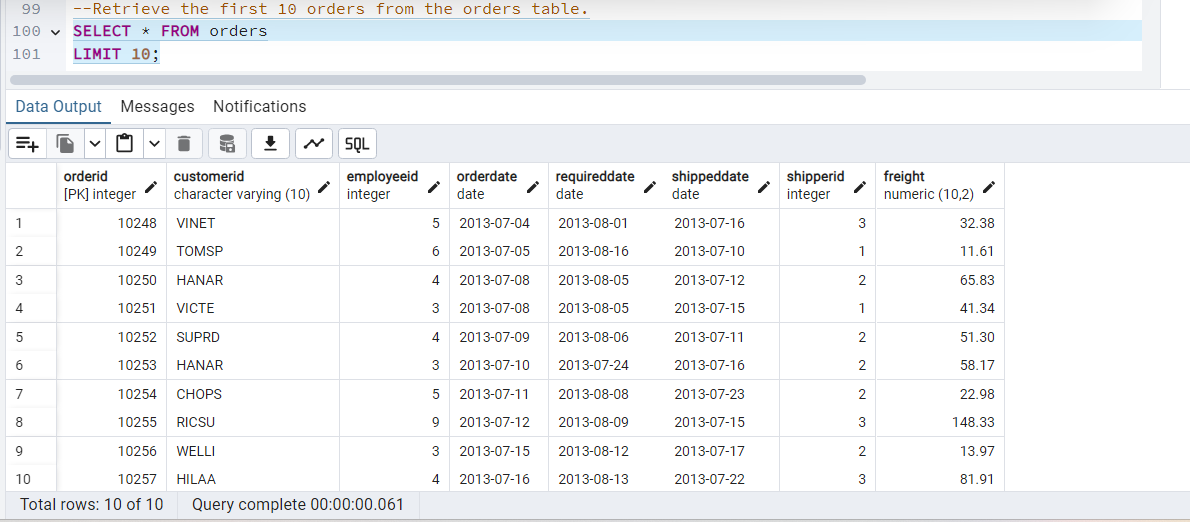


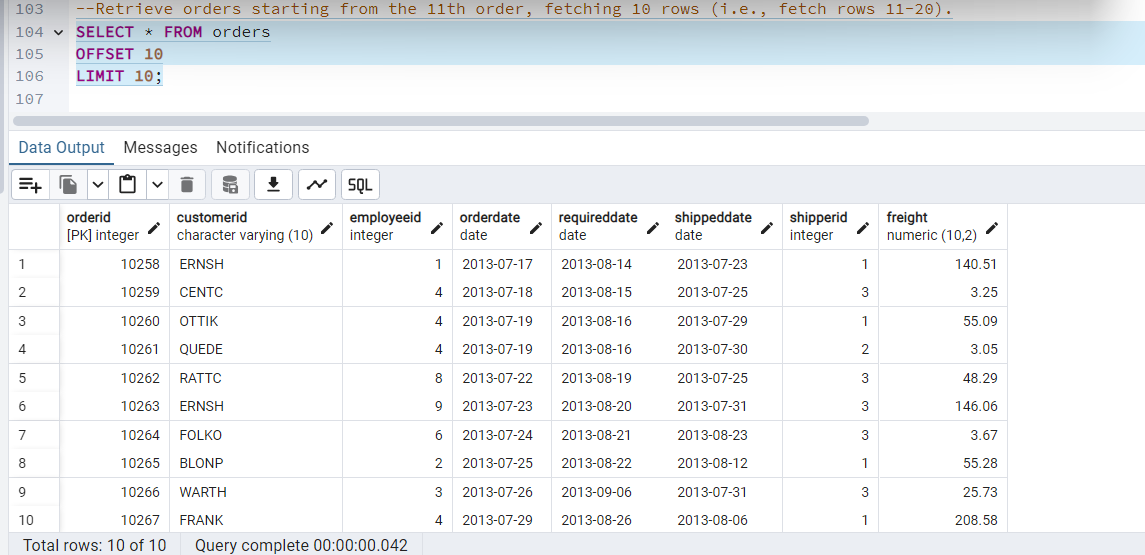
/\*-- 5) LIMIT/FETCH

Retrieve the first 10 orders from the orders table.

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



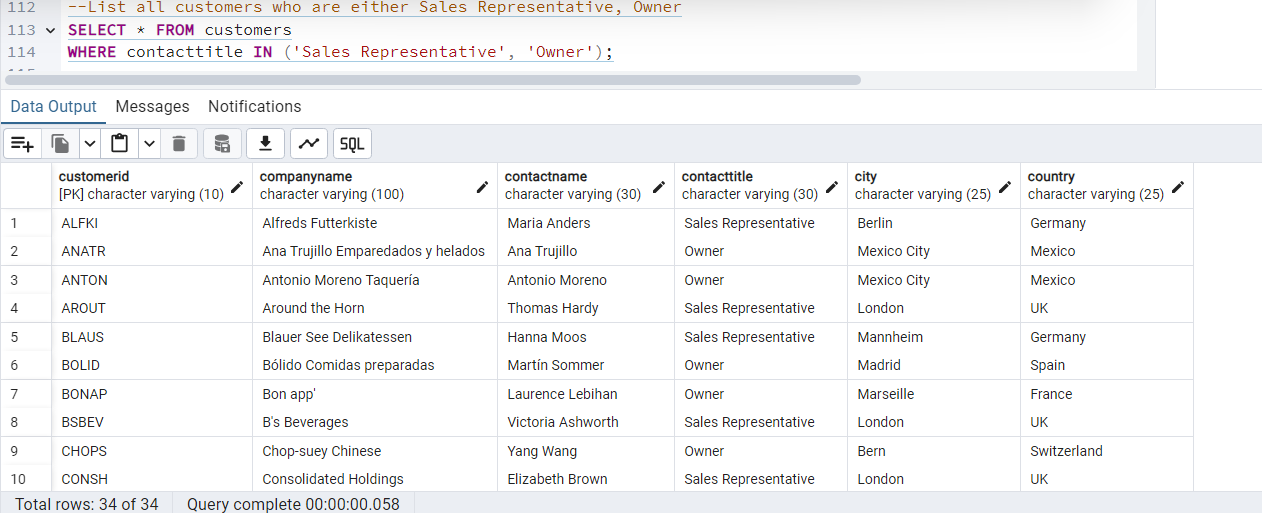


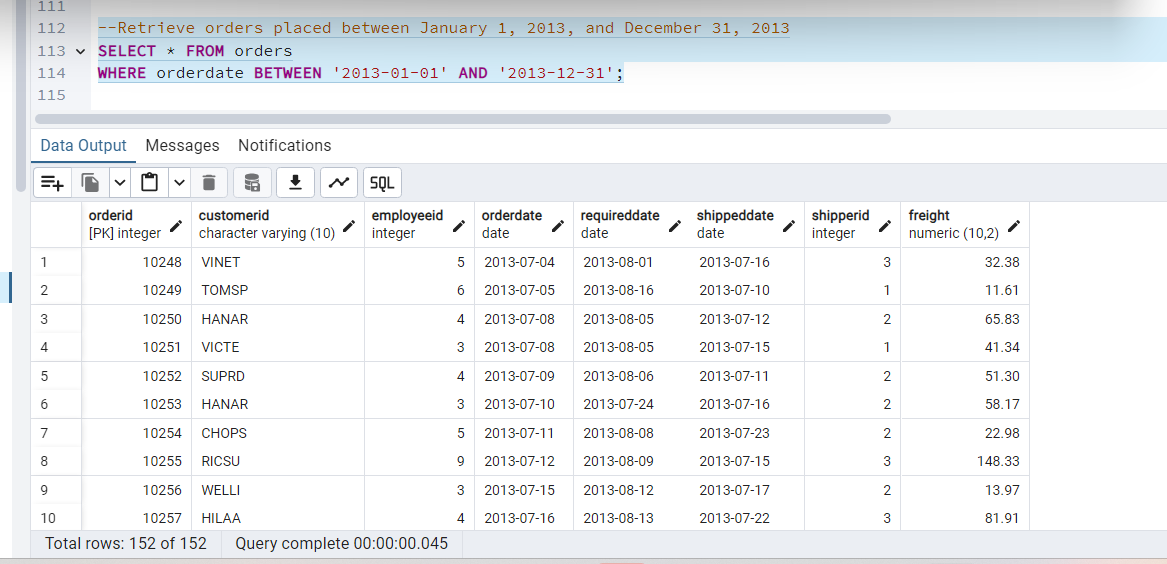


/\*-- 6) Filtering (IN, BETWEEN)

List all customers who are either Sales Representative, Owner

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





/\*-- 7) Filtering

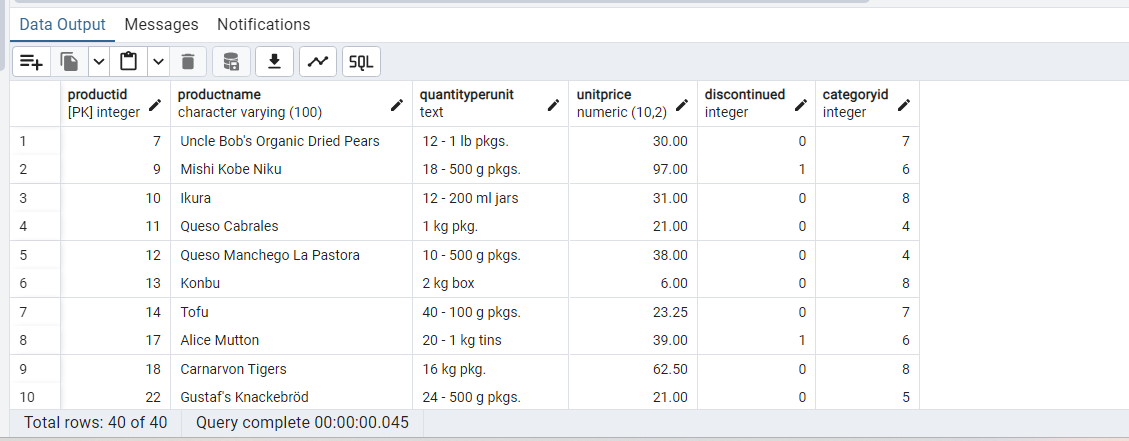
List all products whose category\_id is not 1, 2, or 3.

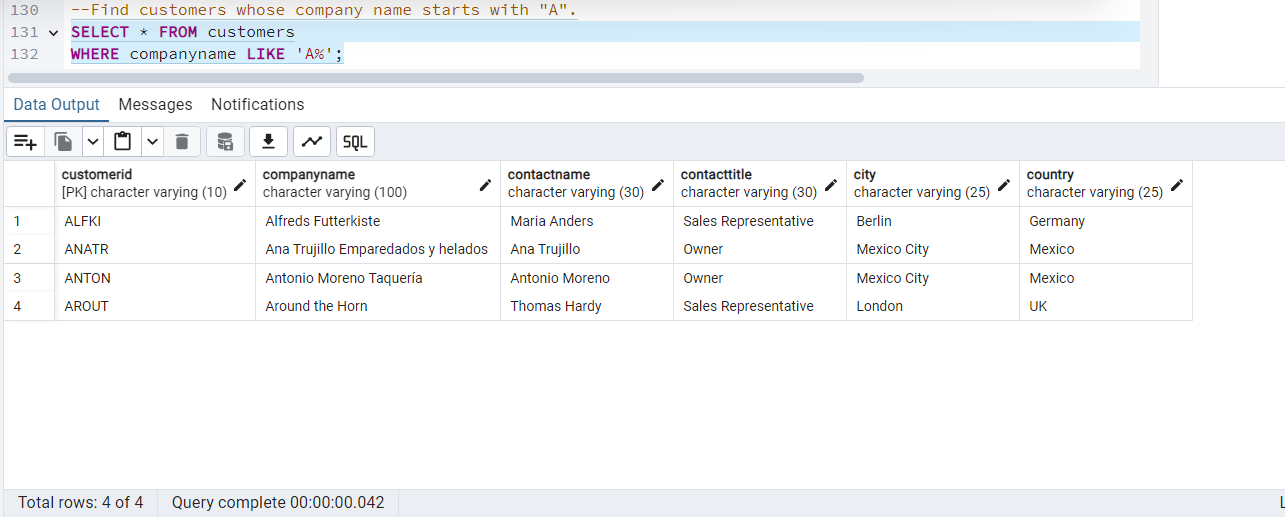
Find customers whose company name starts with "A". --\*/

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

SELECT \* FROM products

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





/\*\* 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 (

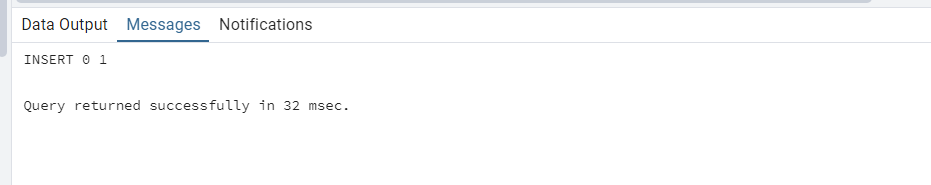
orderid, customerid, employeeid, orderdate,

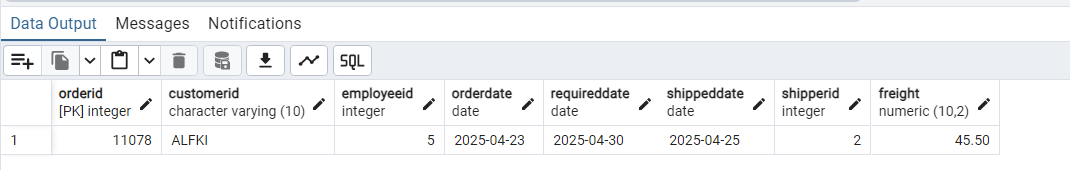
requireddate, shippeddate, shipperid, freight

)

VALUES (

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





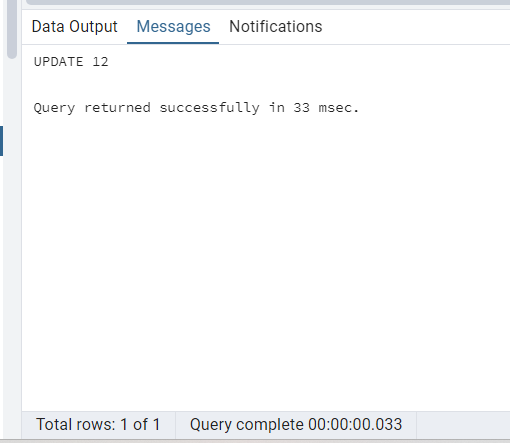
/\*\*9) Increase(Update) the unit price of all products in category\_id =2 by 10%.

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

UPDATE products

SET unitprice = unitprice \* 1.10

WHERE categoryid = 2;



/\*\*10) Sample Northwind database:

Download

Download northwind.sql from below link into your local. Sign in to Git first https://github.com/pthom/northwind\_psql

Manually Create the database using pgAdmin:

Right-click on "Databases" → Create → Database

Give name as ‘northwind’ (all small letters)

Click ‘Save’

Import database:

Open pgAdmin and connect to your server

Select the database ‘northwind’

Right Click-> Query tool.

Click the folder icon to open your northwind.sql file

Press F5 or click the Execute button.

You will see total 14 tables loaded

Databases → your database → Schemas → public → Tables \*\*/

