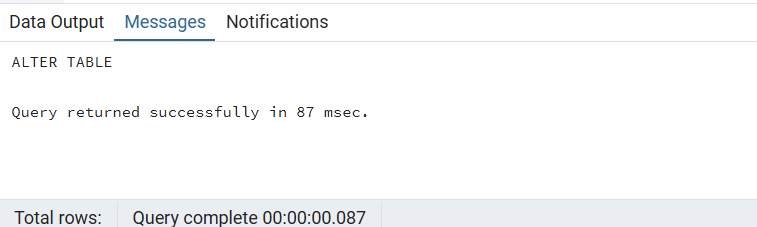
**Day2**

1) Alter Table:

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

ALTER TABLE employees

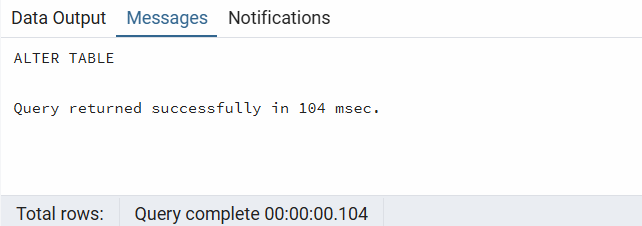
ADD COLUMN linkedin\_profile VARCHAR(100);



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

ALTER TABLE employees

ALTER COLUMN linkedin\_profile TYPE TEXT;

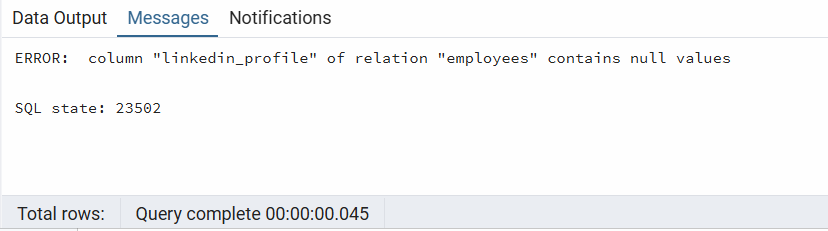


* Add unique, not null constraint to linkedin\_profile

ALTER TABLE employees

ALTER COLUMN linkedin\_profile SET NOT NULL,

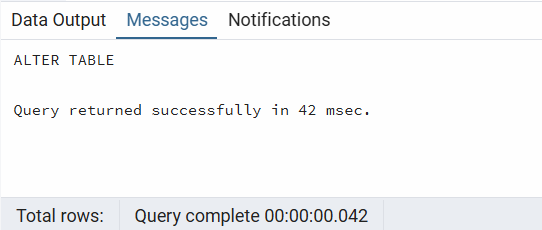
ADD CONSTRAINT linkedin\_profile\_unique UNIQUE (linkedin\_profile);



* Drop column linkedin\_profile

ALTER TABLE employees

DROP COLUMN linkedin\_profile;

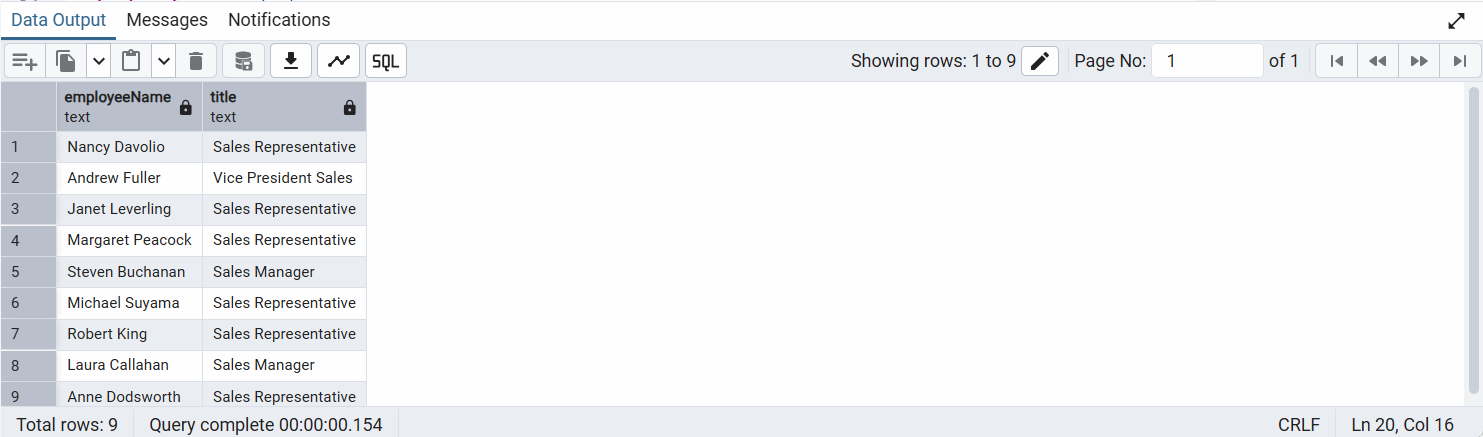


2) Querying (Select)

* Retrieve the employee name and title of all employees

SELECT "employeeName", title

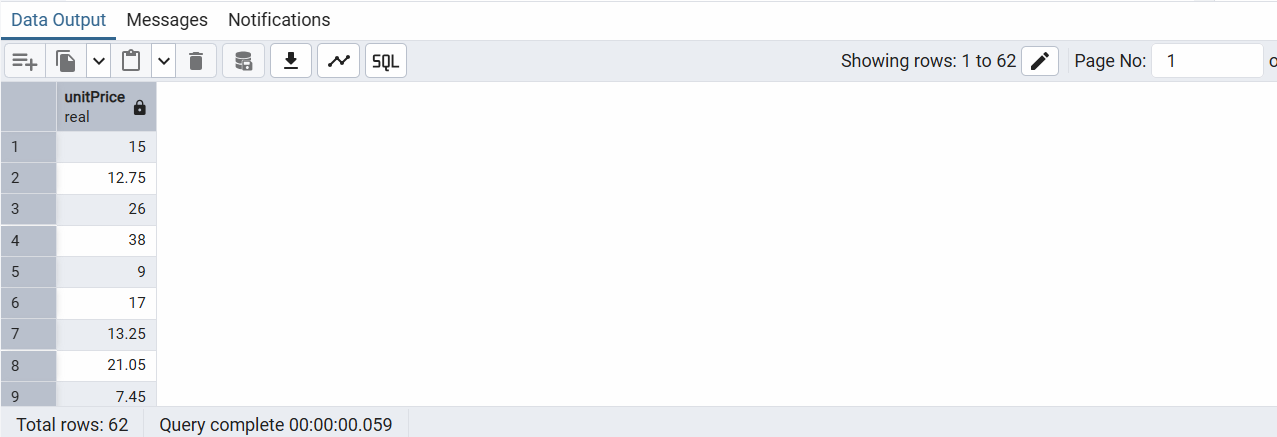
FROM employees;



* Find all unique unit prices of products

SELECT DISTINCT "unitPrice"

FROM products;

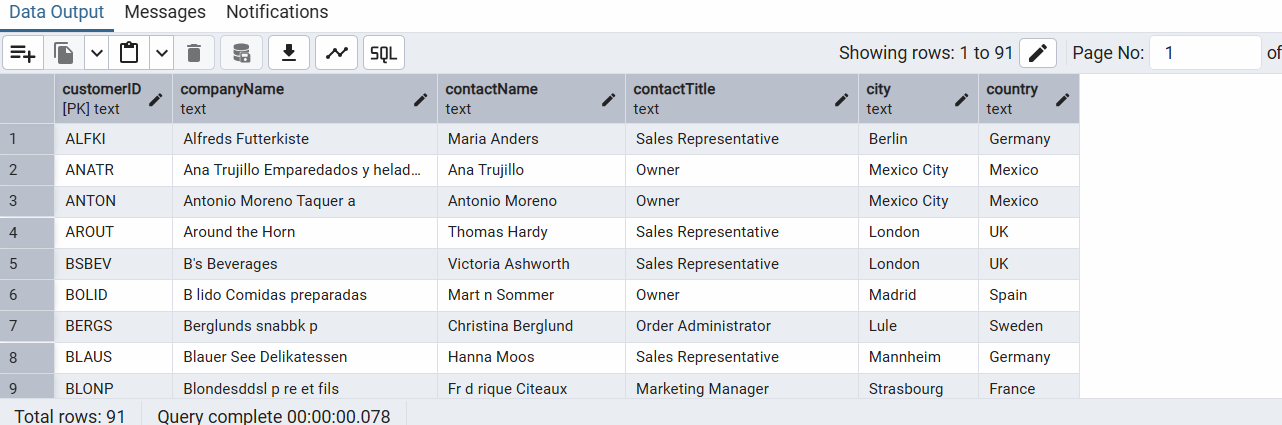


* List all customers sorted by company name in ascending order

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;



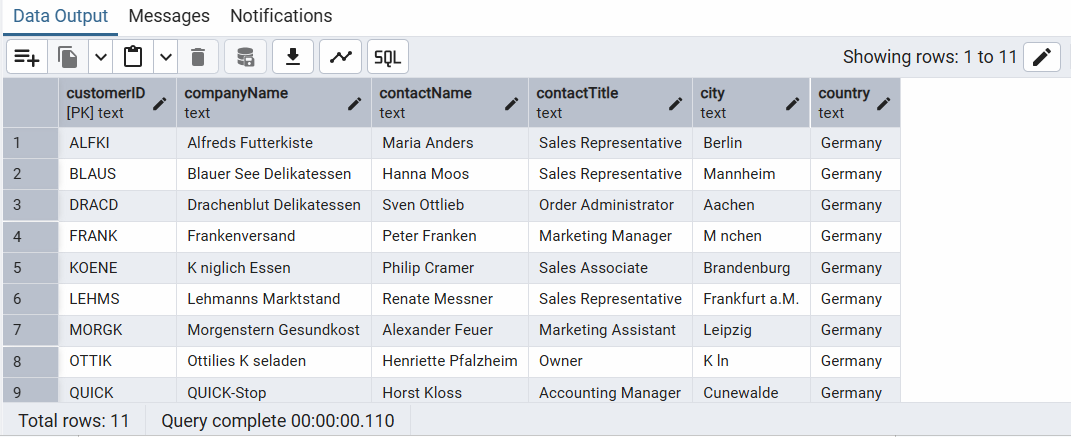
3) Filtering

* Get all customers from Germany.

SELECT \*

FROM customers

WHERE country = 'Germany';

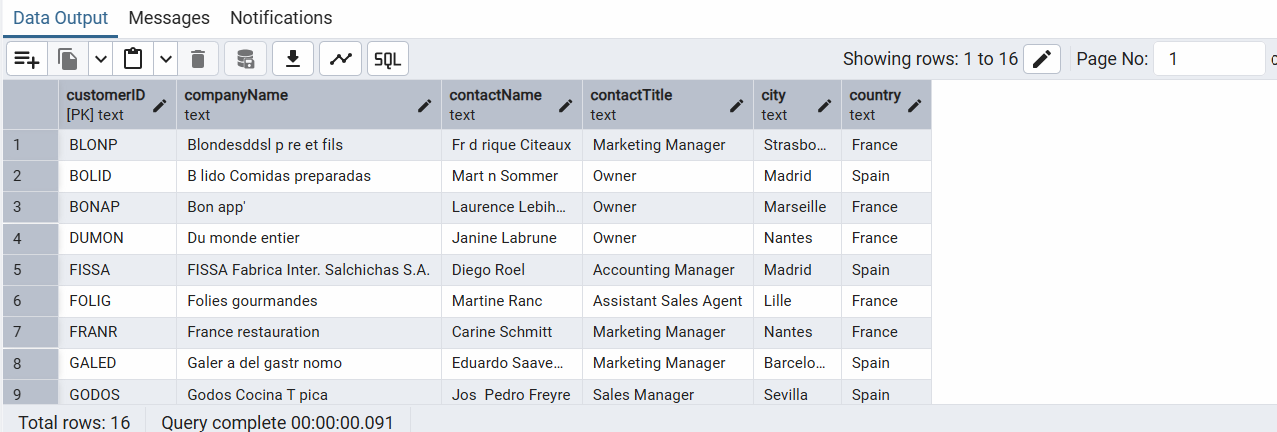


* Find all customers from France or Spain

SELECT \*

FROM customers

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



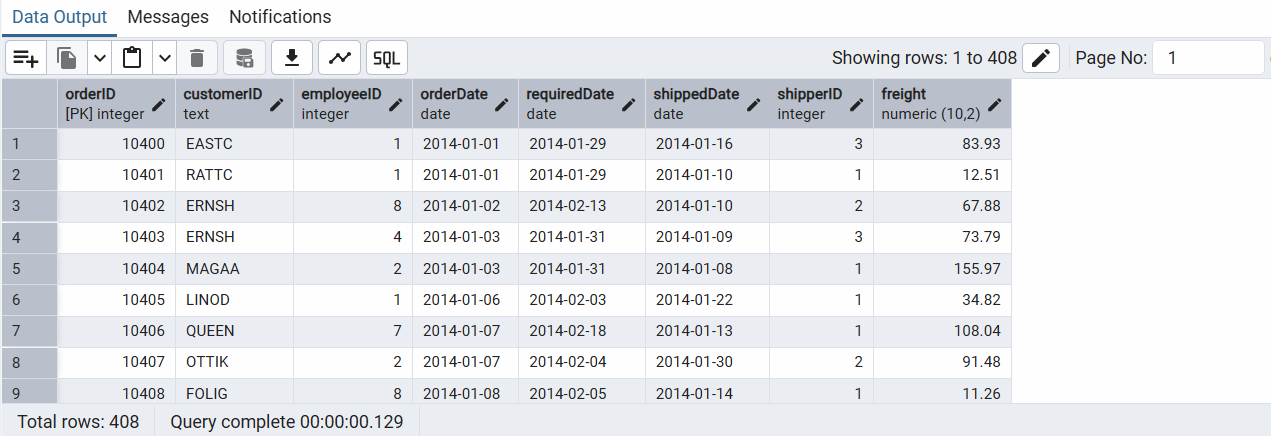
* Retrieve all orders placed in 2014(based on order\_date), and either have freight greater than 50 or the shipped date available (i.e., non-NULL) (Hint: EXTRACT(YEAR FROM order\_date))

SELECT \*

FROM orders

WHERE EXTRACT(YEAR FROM "orderDate") = 2014

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



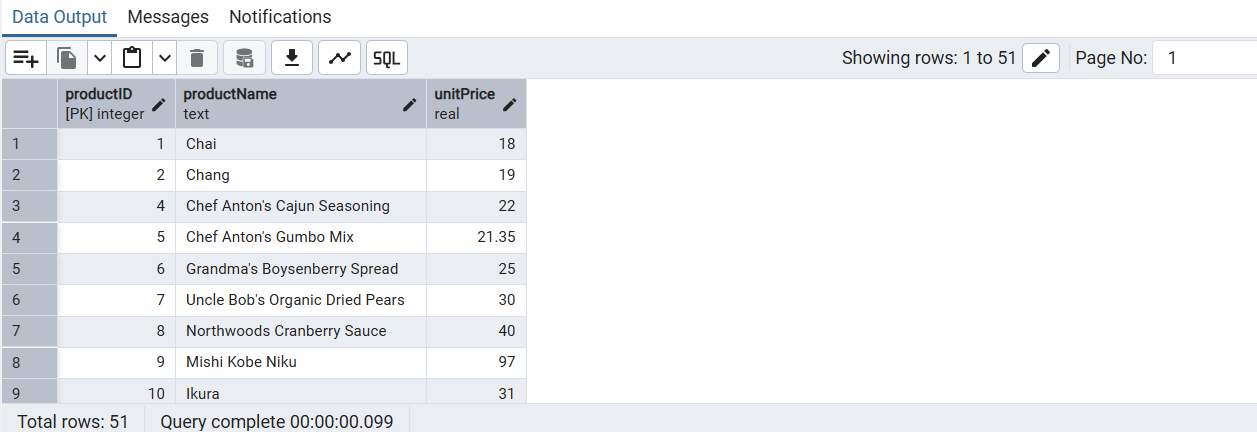
4) Filtering

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

SELECT "productID", "productName", "unitPrice"

FROM products

WHERE "unitPrice" > 15;



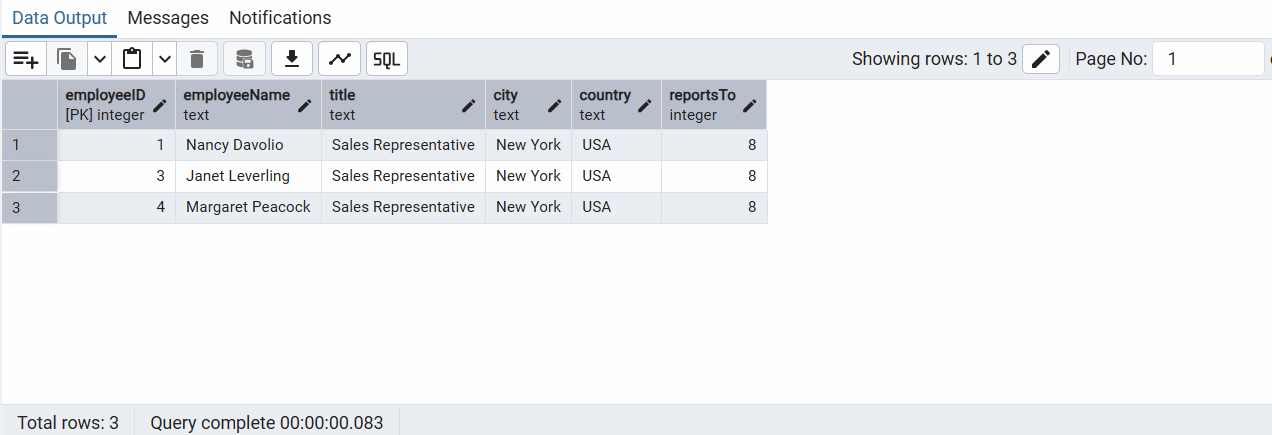
* List all employees who are located in the USA and have the title "Sales Representative".

SELECT \*

FROM employees

WHERE country = 'USA'

AND title = 'Sales Representative';



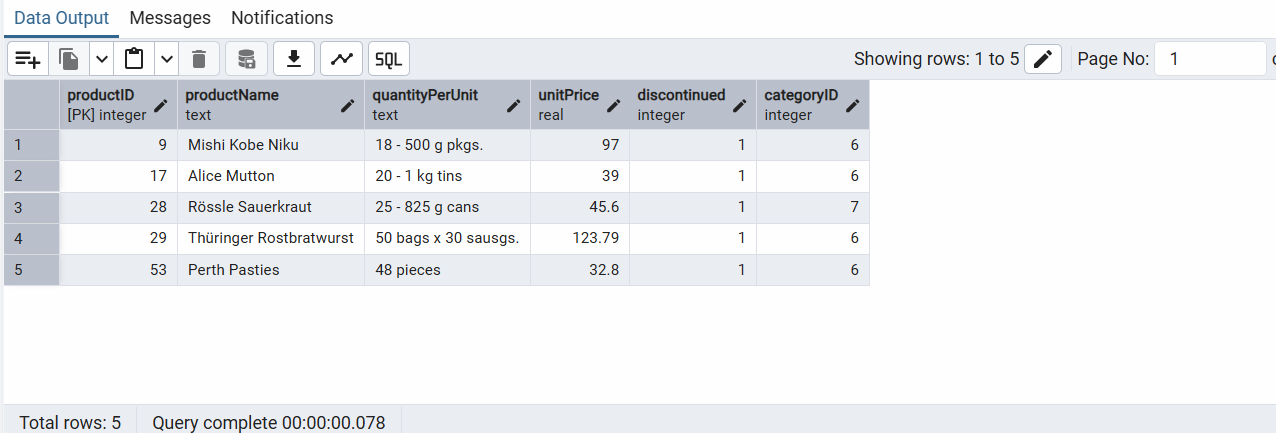
* Retrieve all products that are not discontinued and priced greater than 30.

SELECT \*

FROM products

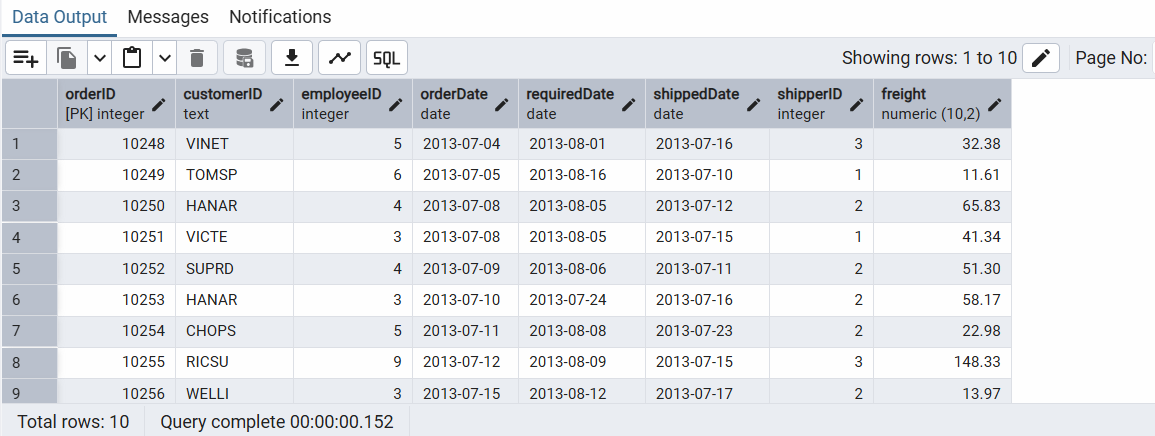
WHERE discontinued = 1

AND "unitPrice" > 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).

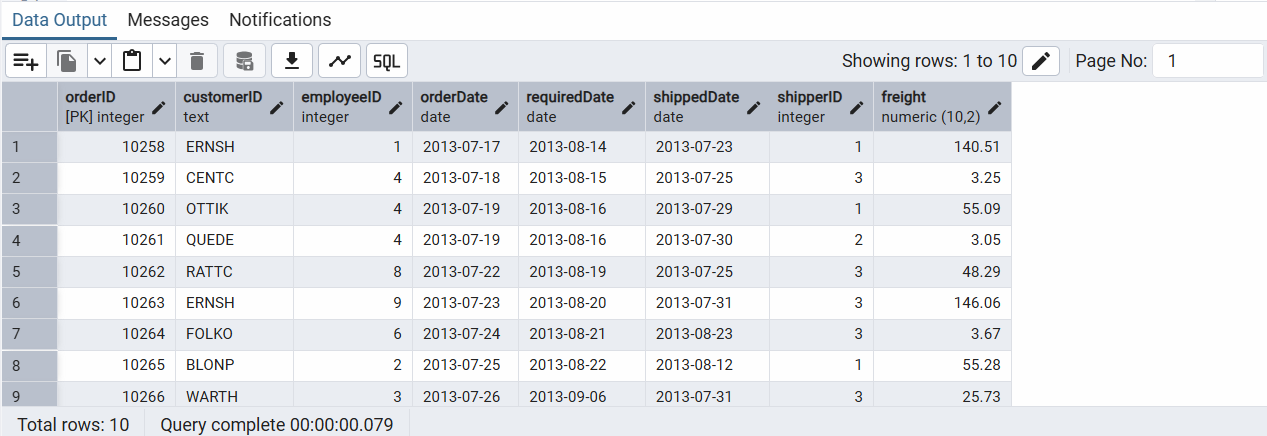
SELECT \*

FROM orders

ORDER BY "orderID"

OFFSET 10

LIMIT 10;



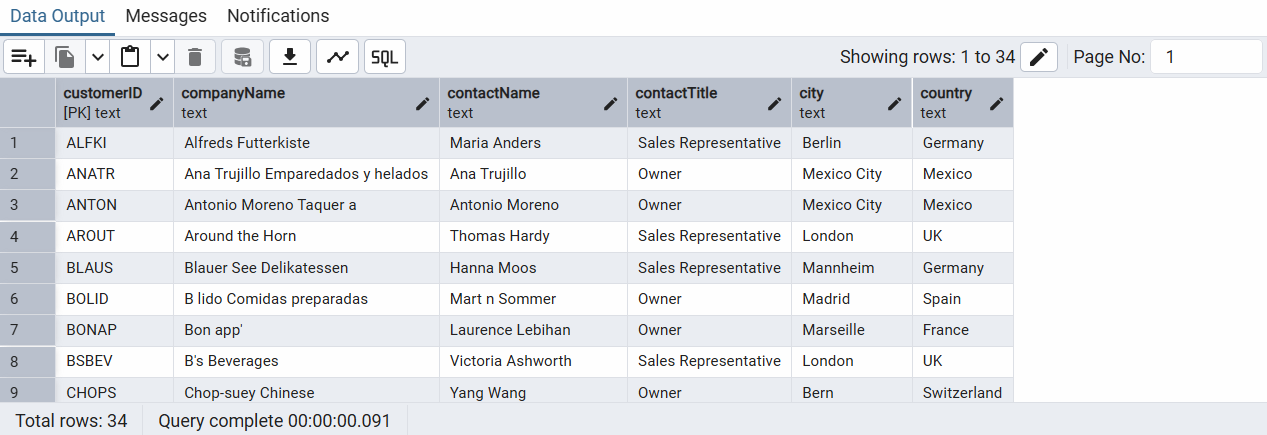
6) Filtering (IN, BETWEEN)

* List all customers who are either Sales Representative, Owner

SELECT \*

FROM customers

WHERE "contactTitle" IN ('Sales Representative', 'Owner');

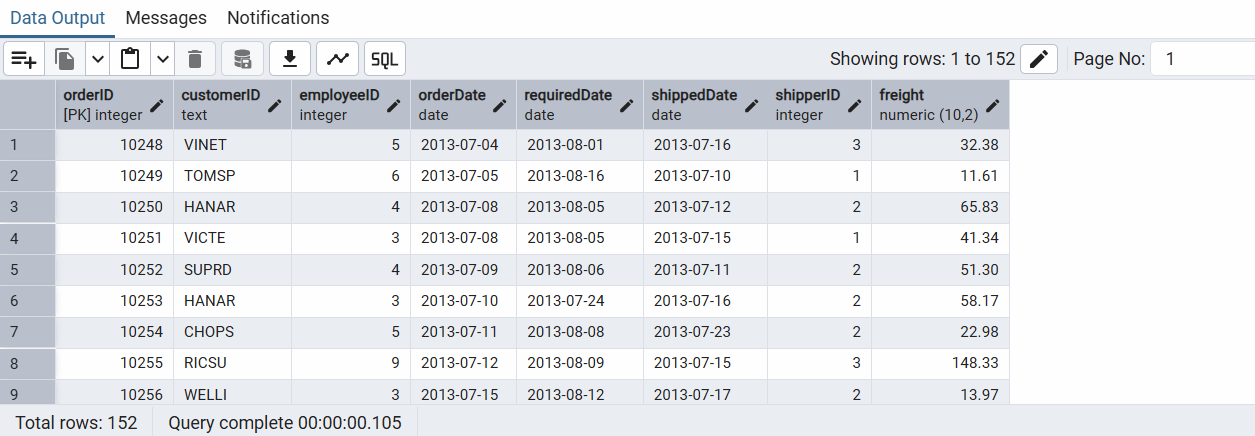


* Retrieve orders placed between January 1, 2013, and December 31, 2013.

SELECT \*

FROM orders

WHERE "orderDate" BETWEEN '2013-01-01' AND '2013-12-31'



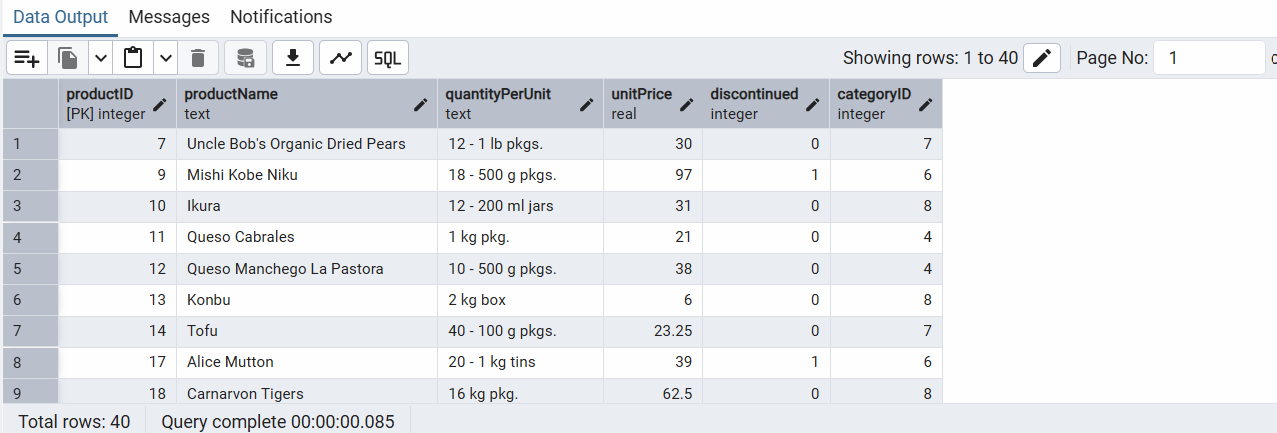
7) Filtering

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

SELECT \*

FROM products

WHERE "categoryID" NOT IN (1, 2, 3);

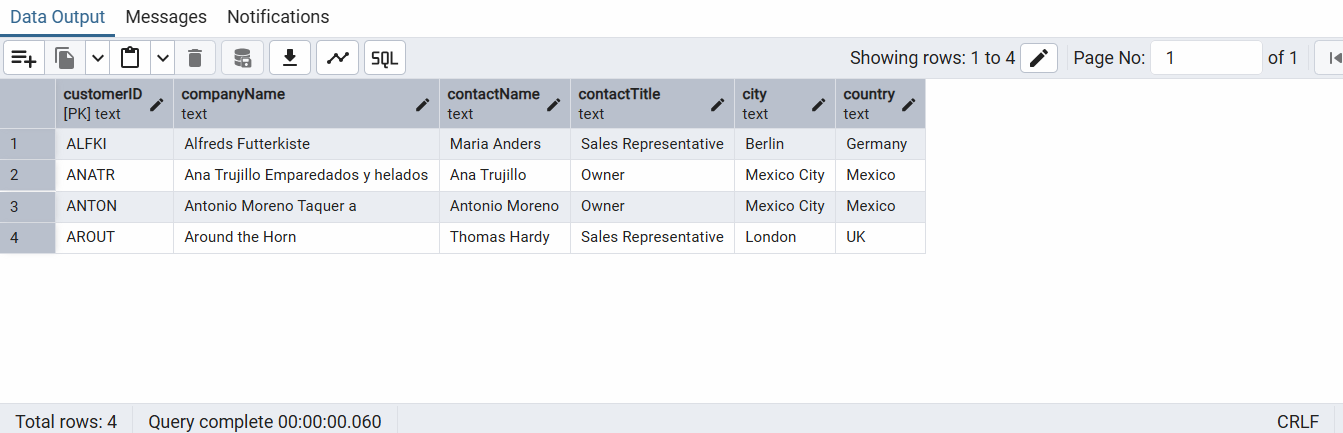


* Find customers whose company name starts with "A".

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

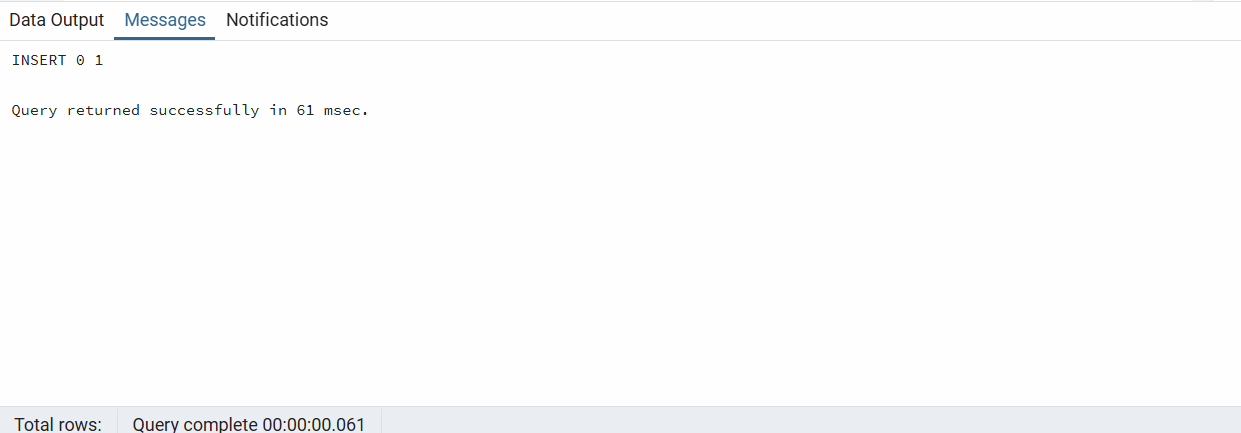
)

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 "categoryID" = 2;

