DAY2\_ASSIGNMENT

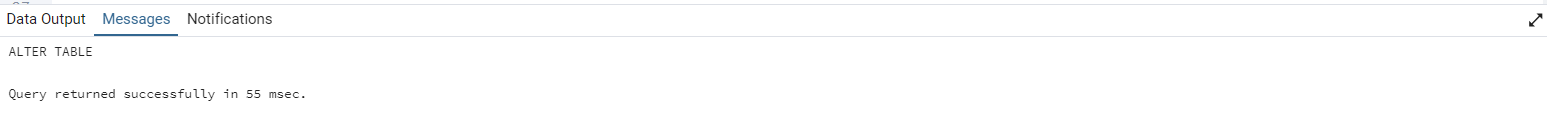
-- 1) Alter Table:

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

ALTER TABLE EMPLOYEES

ADD COLUMN LINKEDIN\_PROFILE VARCHAR(255);

### OUTPUT



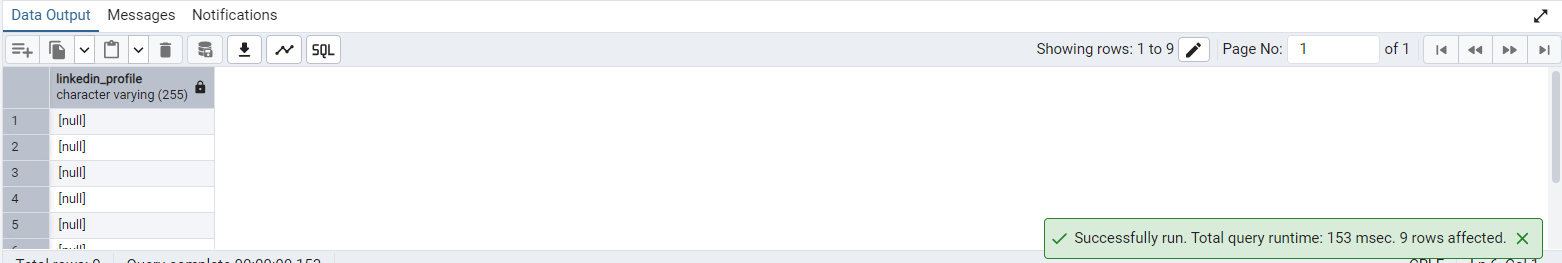
SELECT

LINKEDIN\_PROFILE

FROM

EMPLOYEES

### OUTPUT



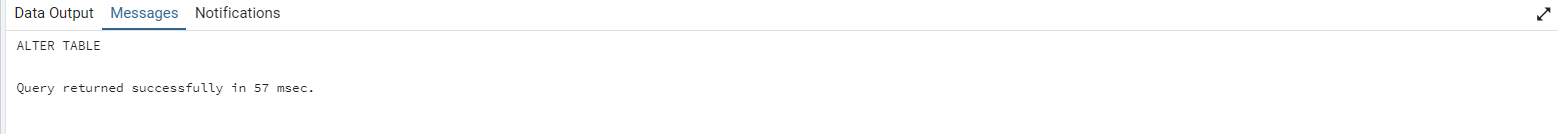
-- Change the linkedin\_profile column data type from VARCHAR to TEXT.

ALTER TABLE EMPLOYEES

ALTER COLUMN LINKEDIN\_PROFILE

SET DATA TYPE TEXT;

### OUTPUT



-- Update nulls

UPDATE EMPLOYEES

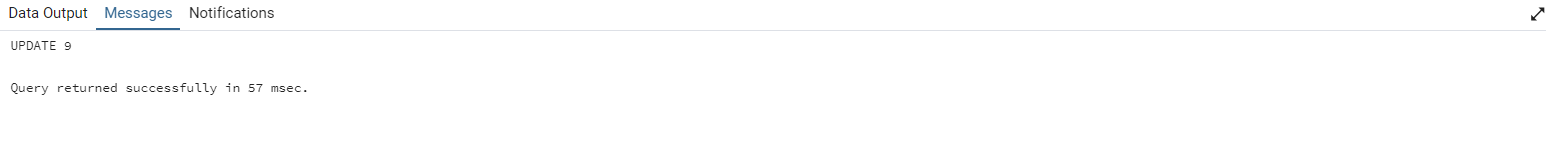
SET

LINKEDIN\_PROFILE = 'https://www.linkedin.com/in/unknown-' || EMPLOYEEID

WHERE

LINKEDIN\_PROFILE IS NULL;

### OUTPUT



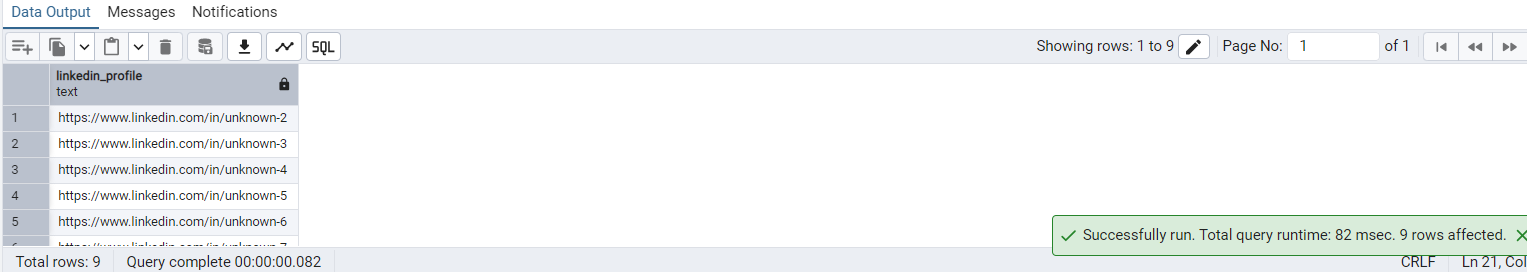
SELECT

LINKEDIN\_PROFILE

FROM

EMPLOYEES

### OUTPUT



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

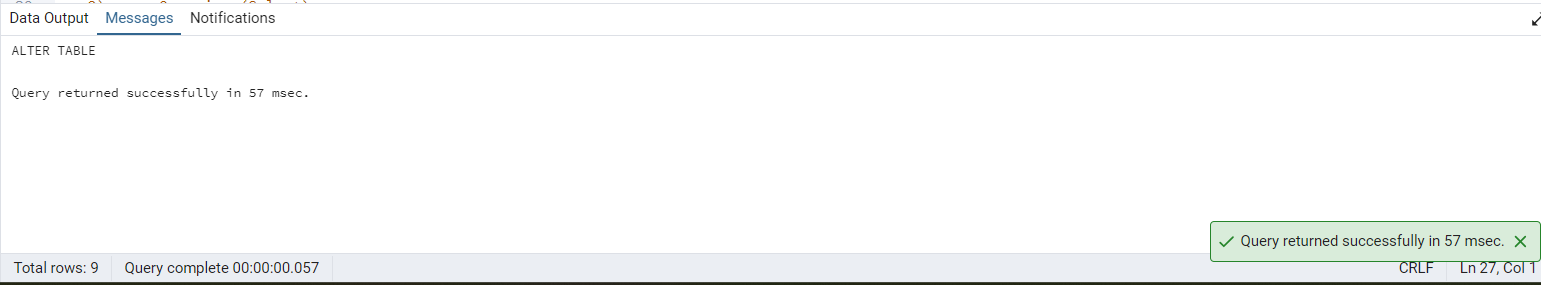
-- Add NOT NULL constraint

ALTER TABLE EMPLOYEES

ALTER COLUMN LINKEDIN\_PROFILE

SET NOT NULL;

### OUTPUT

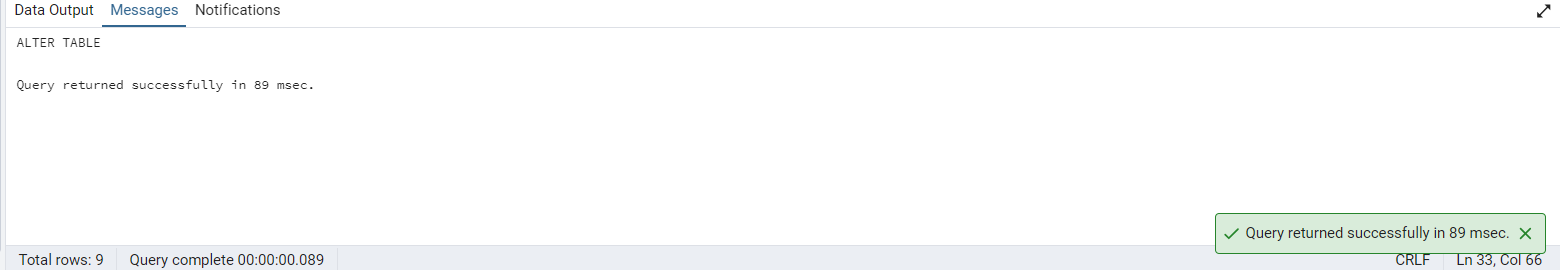


--Add Unique

ALTER TABLE EMPLOYEES

ADD CONSTRAINT UNIQUE\_LINKEDIN\_PROFILE UNIQUE (LINKEDIN\_PROFILE);

### OUTPUT

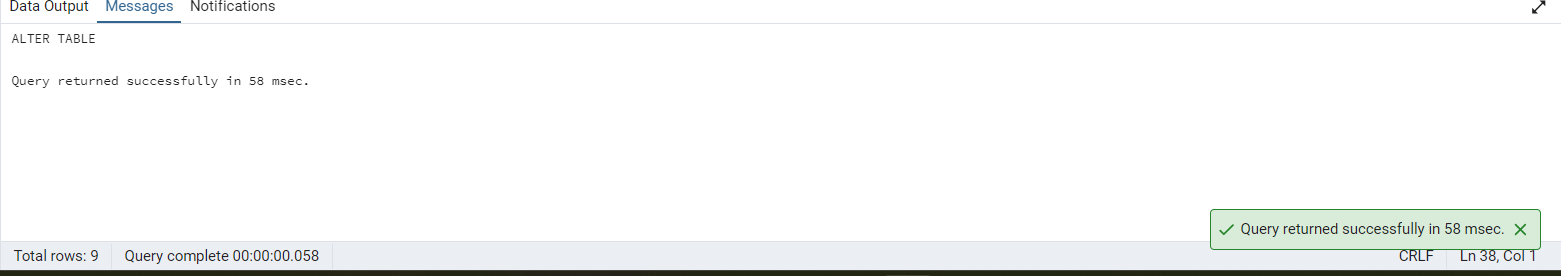


-- Drop column linkedin\_profile

ALTER TABLE EMPLOYEES

DROP COLUMN LINKEDIN\_PROFILE;

### OUTPUT



-- 2) Querying (Select)

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

SELECT

\*

FROM

PRODUCTS

SELECT

EMPLOYEENAME,

SPLIT\_PART(EMPLOYEENAME, ' ', 1) AS FIRST\_NAME,

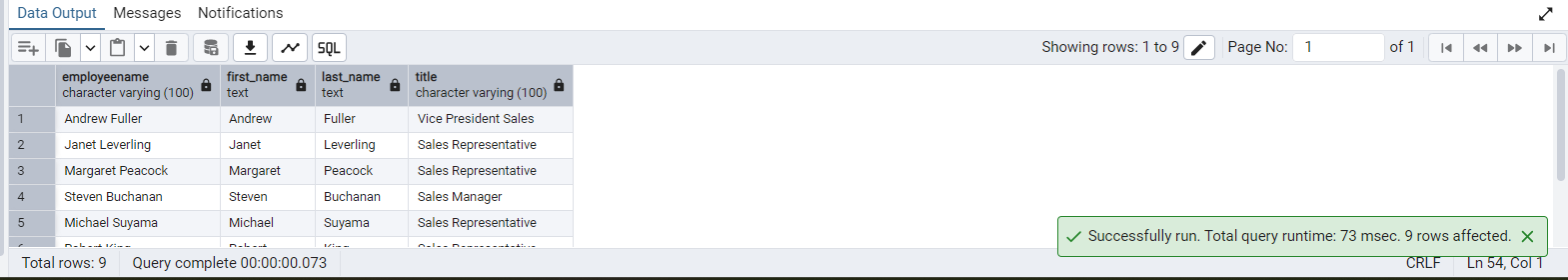
SPLIT\_PART(EMPLOYEENAME, ' ', 2) AS LAST\_NAME,

TITLE

FROM

EMPLOYEES;

### OUTPUT



-- Find all unique unit prices of products

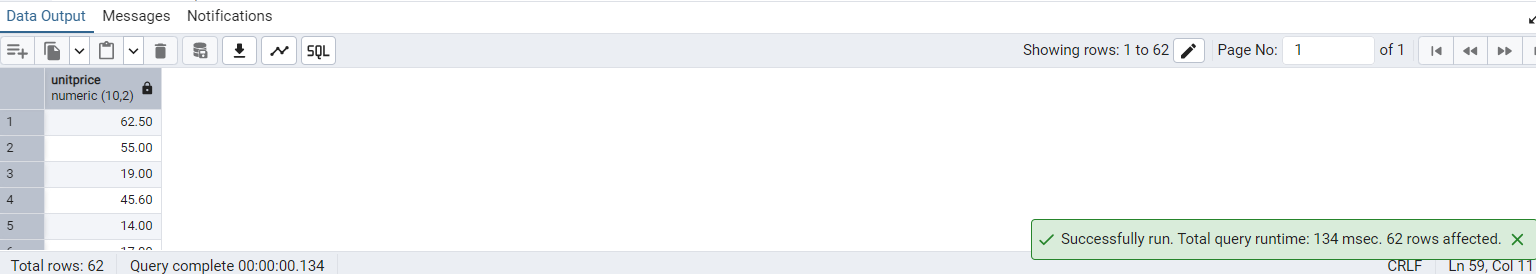
SELECT DISTINCT

UNITPRICE

FROM

PRODUCTS;

### OUTPUT



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

SELECT

CUSTOMER\_ID,

COMPANY\_NAME

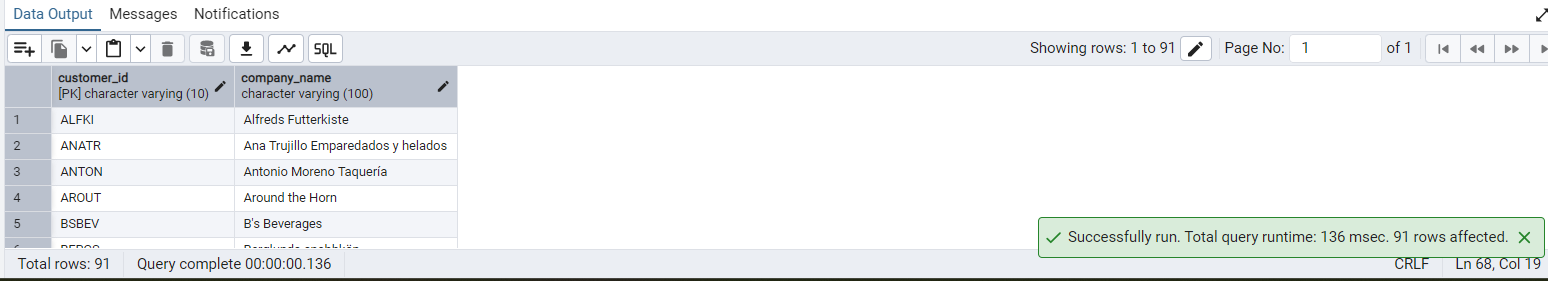
FROM

CUSTOMERS

ORDER BY

COMPANY\_NAME ASC;

### OUTPUT



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

SELECT

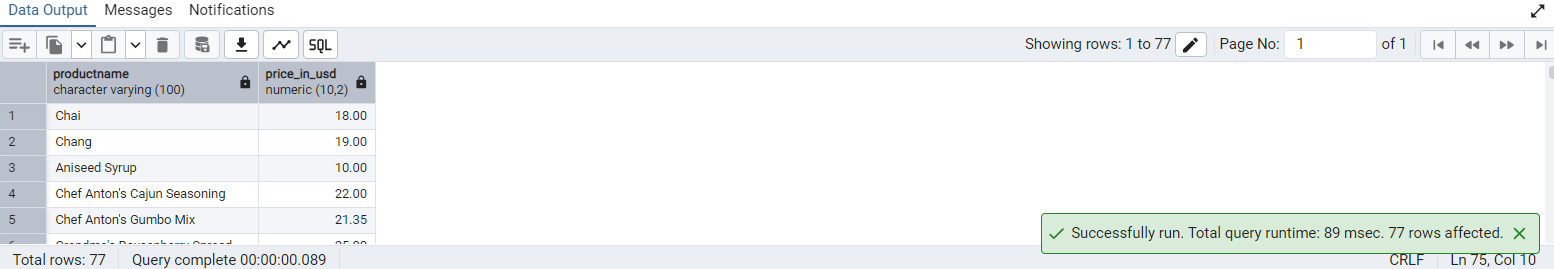
PRODUCTNAME,

UNITPRICE AS PRICE\_IN\_USD

FROM

PRODUCTS

### OUTPUT



-- 3) Filtering

-- Get all customers from Germany.

SELECT

CUSTOMER\_ID,

CONTACT\_NAME

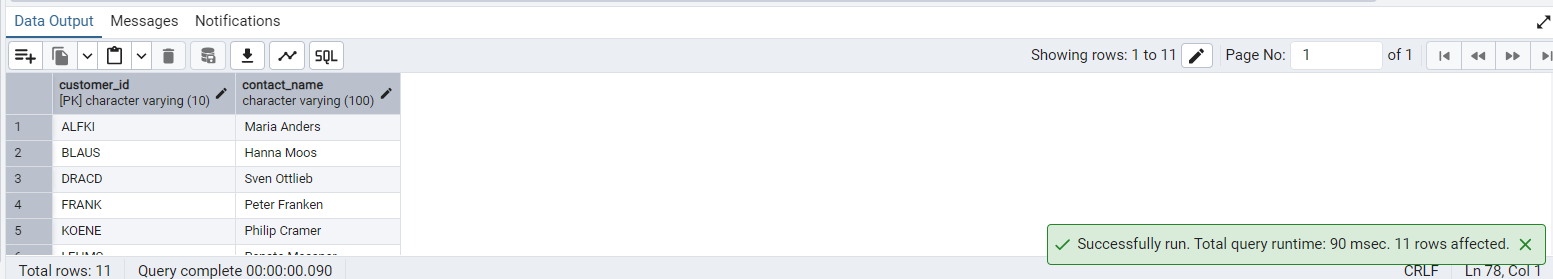
FROM

CUSTOMERS

WHERE

COUNTRY = 'Germany';

### OUTPUT



-- Find all customers from France or Spain

SELECT

CUSTOMER\_ID,

CONTACT\_NAME,

COUNTRY

FROM

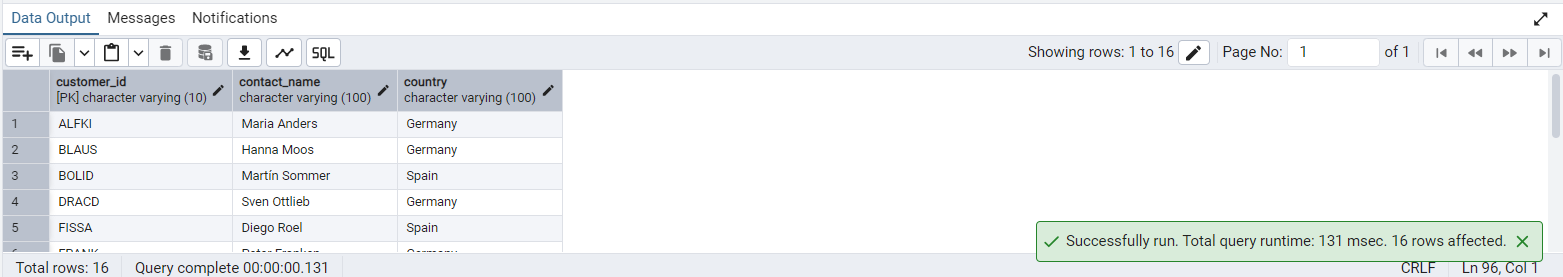
CUSTOMERS

WHERE

COUNTRY = 'Germany'

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))

SELECT

\*

FROM

ORDERS

WHERE

EXTRACT(

YEAR

FROM

ORDERDATE

) = 1997

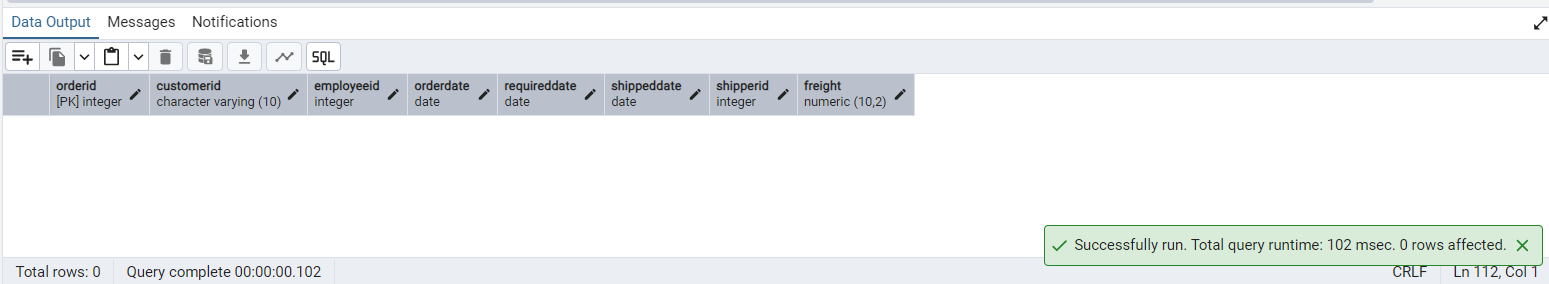
AND (

FREIGHT > 50

OR SHIPPEDDATE IS NOT NULL

);

### OUTPUT



-- 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

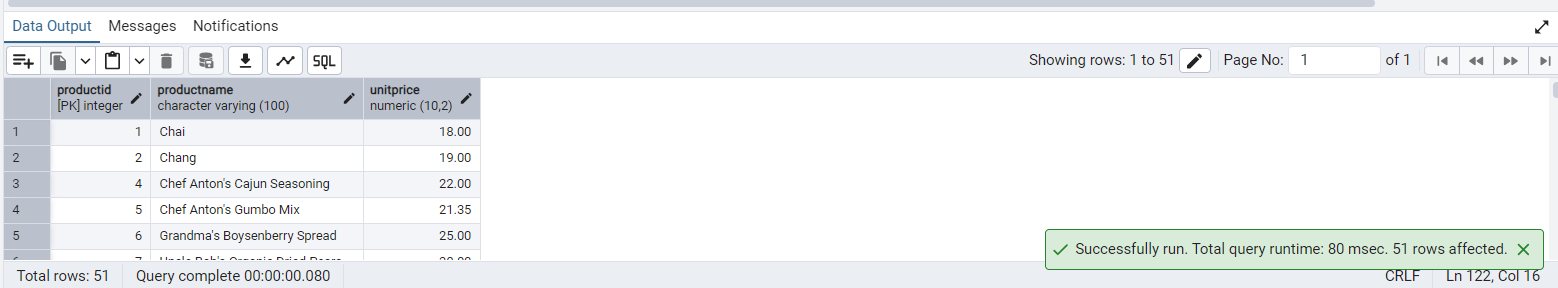
FROM

PRODUCTS

WHERE

UNITPRICE > 15

### OUTPUT



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

SELECT

\*

FROM

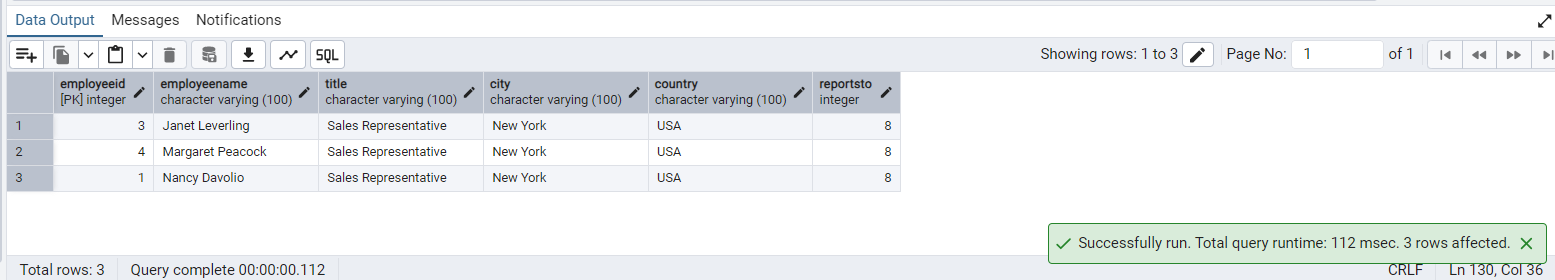
EMPLOYEES

WHERE

COUNTRY = 'USA'

AND TITLE = 'Sales Representative'

### OUTPUT



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

SELECT

\*

FROM

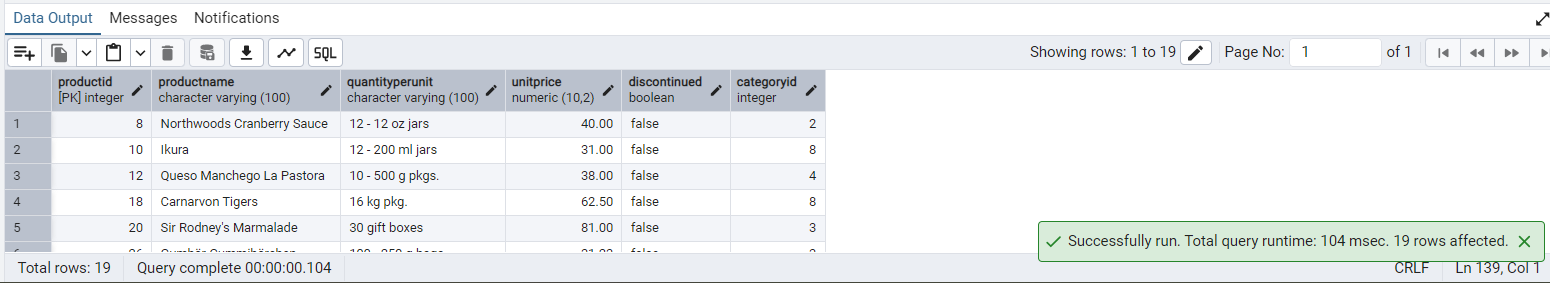
PRODUCTS

WHERE

DISCONTINUED = FALSE

AND UNITPRICE > 30;

### OUTPUT



-- 5) LIMIT/FETCH

-- Retrieve the first 10 orders from the orders table.

SELECT

\*

FROM

ORDERS

LIMIT

10;

--OR--

SELECT

\*

FROM

ORDERS

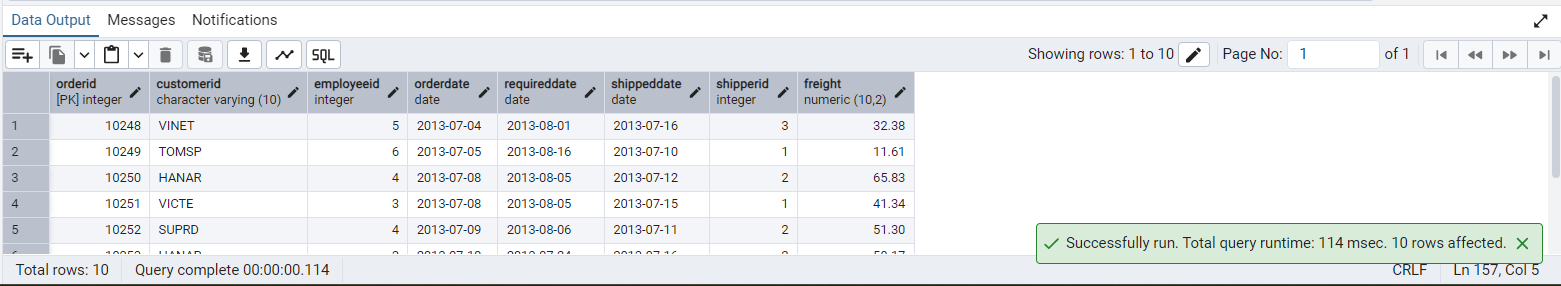
ORDER BY

ORDERDATE

LIMIT

10;

### OUTPUT



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

SELECT

\*

FROM

ORDERS

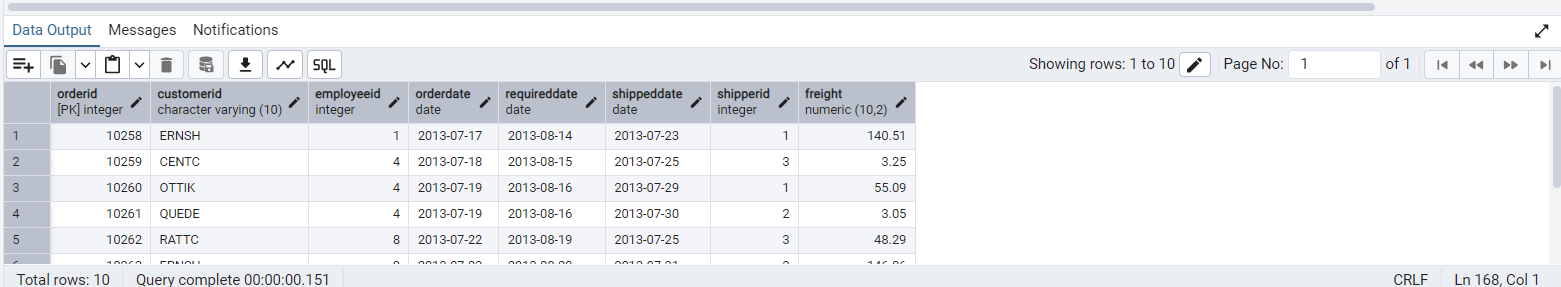
OFFSET

10

LIMIT

10;

### OUTPUT



-- 6) Filtering (IN, BETWEEN)

-- List all customers who are either Sales Representative, Owner

SELECT

\*

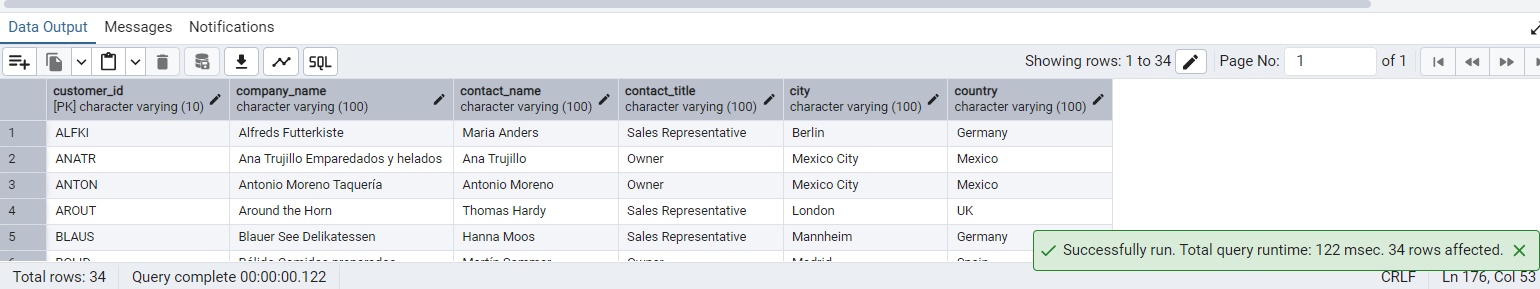
FROM

CUSTOMERS

WHERE

CONTACT\_TITLE IN ('Sales Representative', 'Owner');

### OUTPUT



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

SELECT

\*

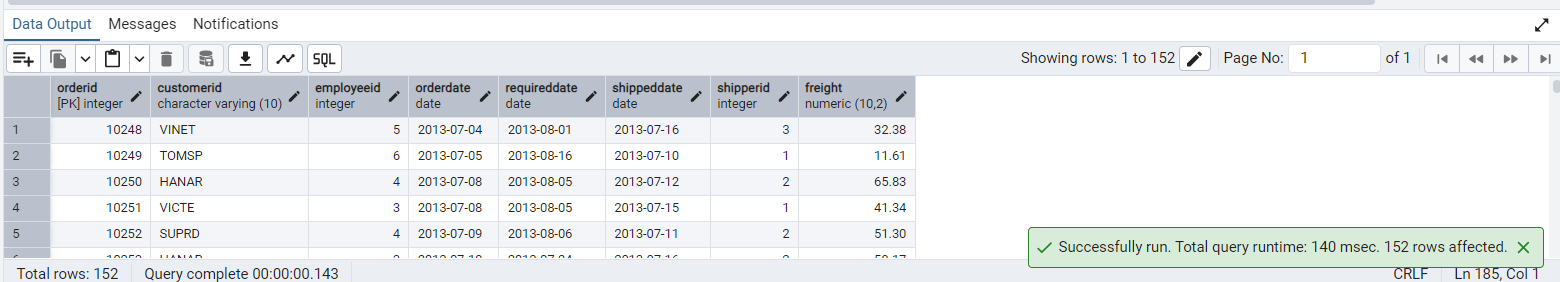
FROM

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.

SELECT

\*

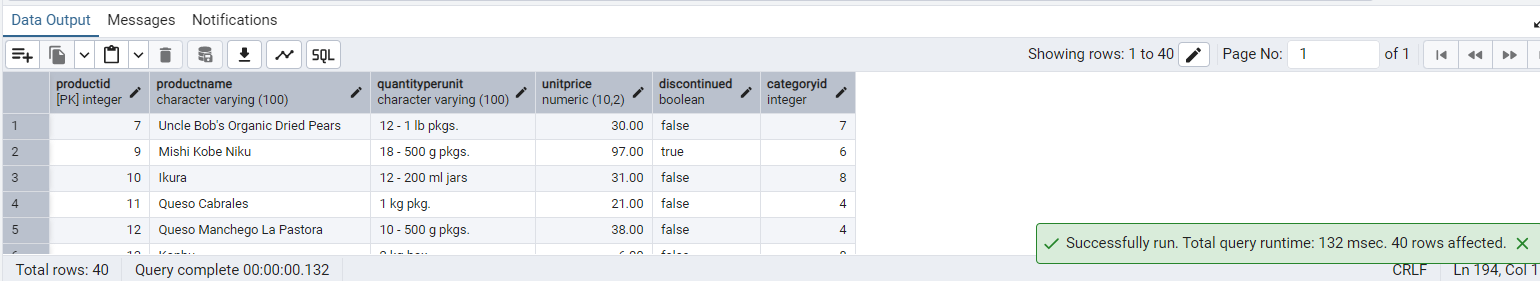
FROM

PRODUCTS

WHERE

CATEGORYID NOT IN (1, 2, 3);

### OUTPUT



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

SELECT

\*

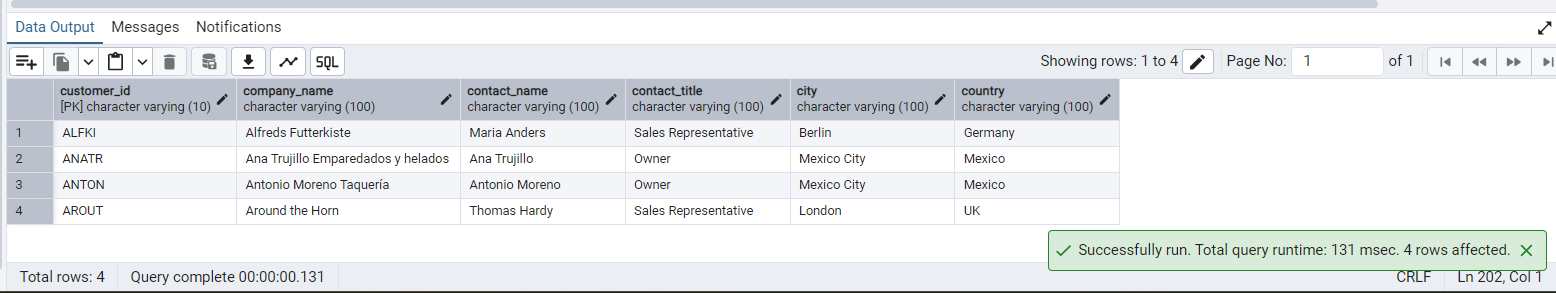
FROM

CUSTOMERS

WHERE

COMPANY\_NAME 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

INSERT INTO

ORDERS (

ORDERID,

CUSTOMERID,

EMPLOYEEID,

ORDERDATE,

REQUIREDDATE,

SHIPPEDDATE,

SHIPPERID,

FREIGHT

)

VALUES

(

11078,

'ALFKI',

5,

'2025-04-23',

'2025-04-30',

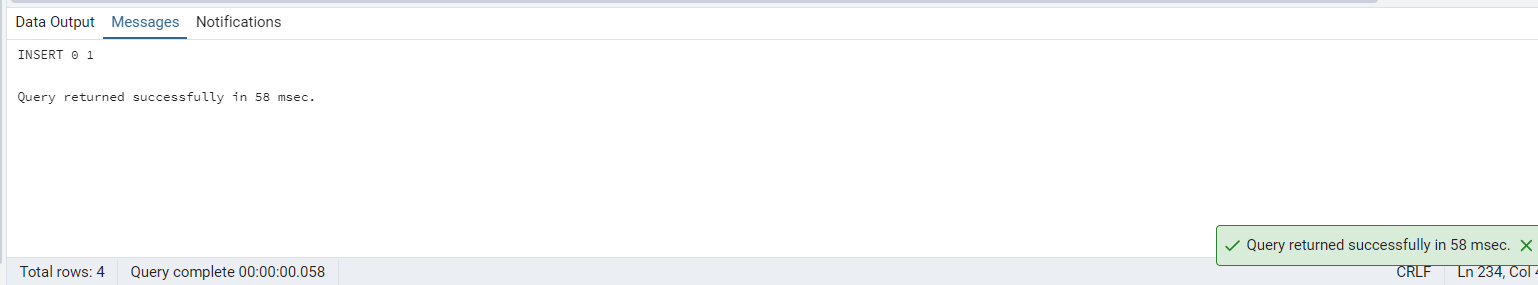
'2025-04-25',

2,

45.50

);

### OUTPUT



SELECT

\*

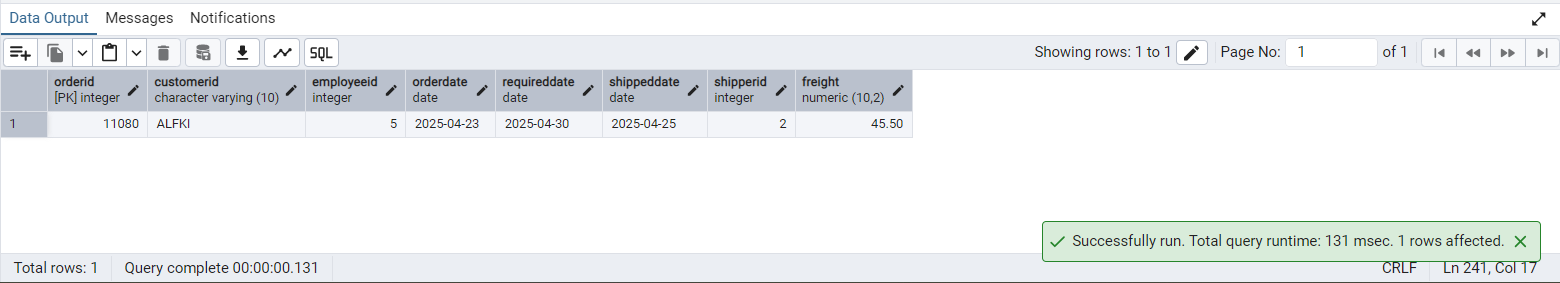
FROM

ORDERS

WHERE

ORDERID = 11078

### OUTPUT



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

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

UPDATE PRODUCTS

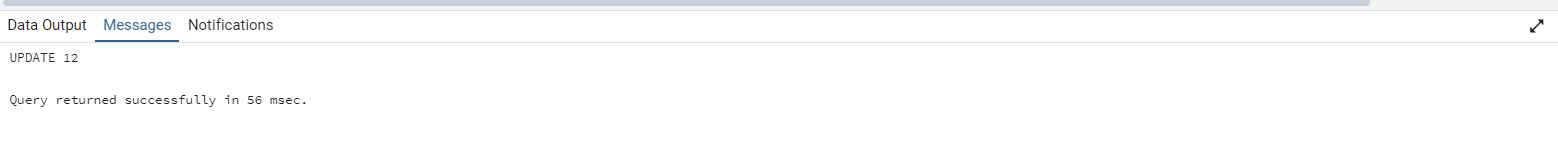
SET

UNITPRICE = UNITPRICE \* 1.10

WHERE

CATEGORYID = 2;

### OUTPUT



SELECT

UNITPRICE

FROM

PRODUCTS

WHERE

CATEGORYID =2

### OUTPUT

