**Day 3**

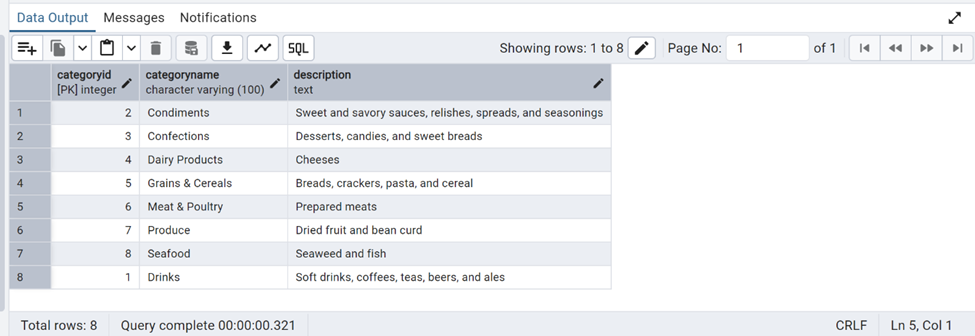
**USE Northwind from Kaggle:**

1) Update the categoryName From “Beverages” to "Drinks" in the categories table.

update categories

set categoryName = 'Beverages'

where categoryName = 'Drinks';



2) Insert into shipper new record (give any values) Delete that new record from shippers table.

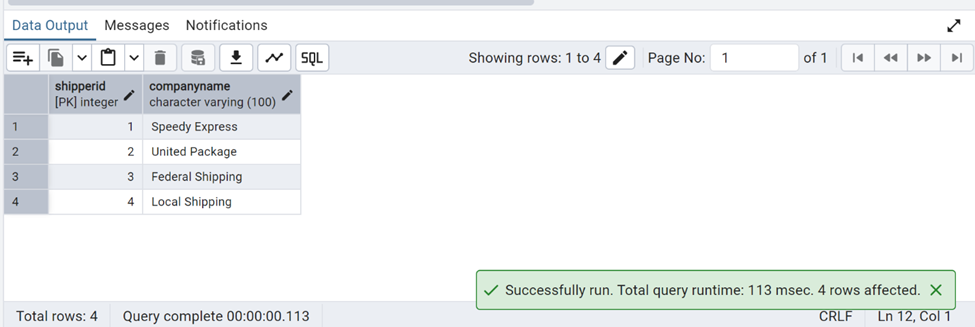
insert into shippers ("shipperID", "companyName")

values(4,'local shipping');

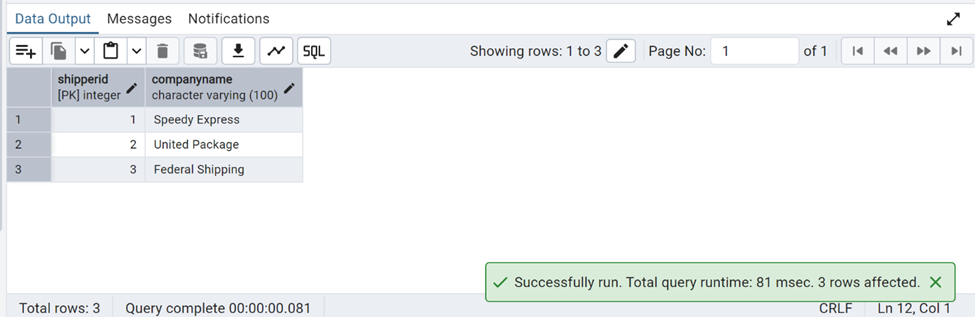
select \* from shippers;

delete from shippers

where "companyName "='Local shipping';



After Deletion:

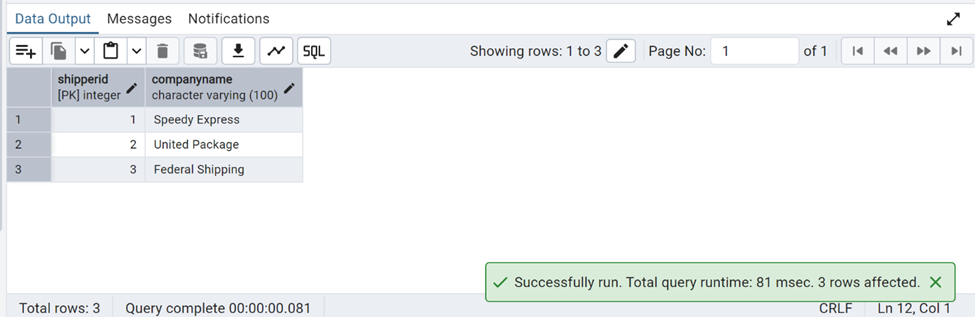


3) Update categoryID=1 to categoryID=1001. Make sure related products update their categoryID too. Display the both category and products table to show the cascade.

Delete the categoryID= “3” from categories. Verify that the corresponding records are deleted automatically from products.

(HINT: Alter the foreign key on products(categoryID) to add ON UPDATE CASCADE, ON DELETE CASCADE)

Select \* from categories where categoryID =1;



alter table products

drop constraint if exists

fk\_products\_category;



alter table products

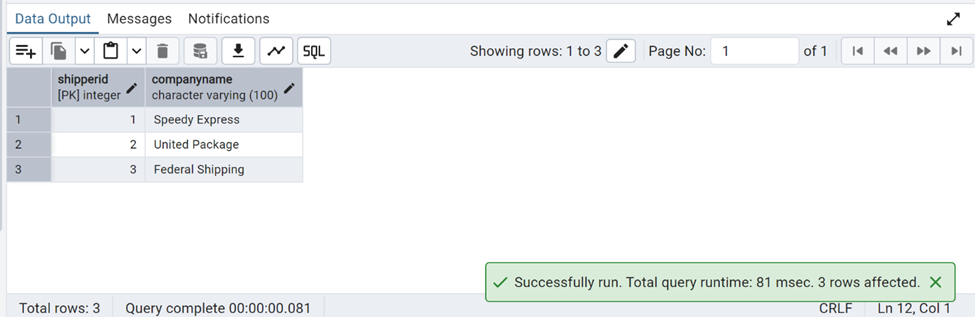
add constraint fk\_products\_category

foreign key ("categoryID")

references categories("categoryID")

on update cascade

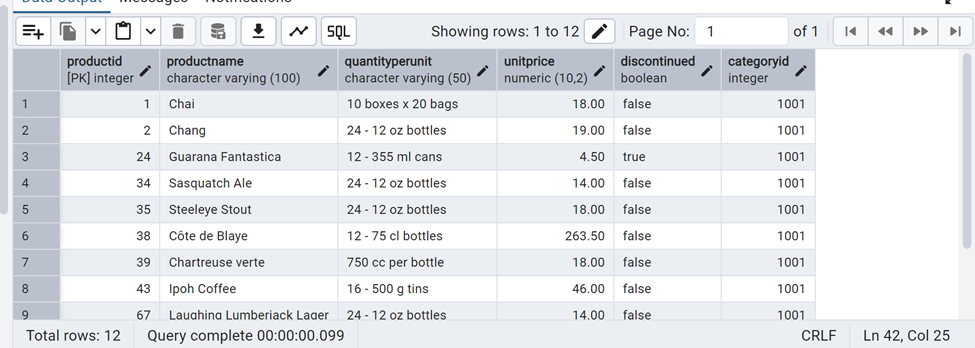
on delete cascade;



update categories

set "categoryID"=1001

where "categoryID"=1



4) Delete the customer = “VINET” from customers. Corresponding customers in orders table should be set to null (HINT: Alter the foreign key on orders(customerID) to use ON DELETE SET NULL)

select \*from orders;

ALTER TABLE orders

DROP CONSTRAINT orders\_customerid\_fkey;

ALTER TABLE orders

ADD CONSTRAINT fk\_orders\_customerid

FOREIGN KEY (customerID)

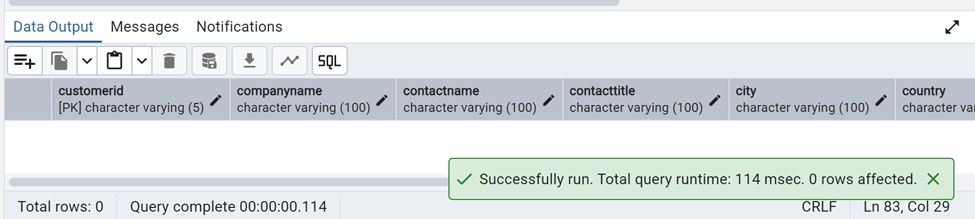
REFERENCES customers(customerID)

ON DELETE SET NULL;

delete from customers where customerid='VINET';

select \* from customers where customerid='VINET';

select \* from orders where customerid is null;



5) Insert the following data to Products using UPSERT:

product\_id = 100, product\_name = Wheat bread, quantityperunit=1,unitprice = 13, discontinued = 0, categoryID=3

product\_id = 101, product\_name = White bread, quantityperunit=5 boxes,unitprice = 13, discontinued = 0, categoryID=3

product\_id = 100, product\_name = Wheat bread, quantityperunit=10 boxes,unitprice = 13, discontinued = 0, categoryID=3

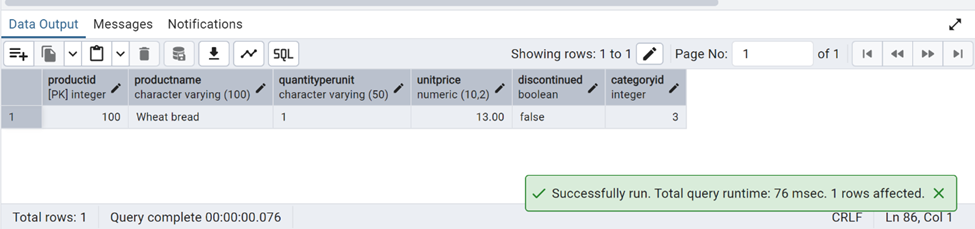
(this should update the quantityperunit for product\_id = 100)

Query:

select \* from products where productid in (100,101);

INSERT INTO Products (productid,productname,quantityperunit,unitprice,discontinued,categoryid)

VALUES (100, 'Wheat bread',1,13,0,5)



6) Write a **MERGE query**:

Create **temp table with name:**  ‘updated\_products’ and insert values as below:

| productID | productName | quantityPerUnit | unitPrice | discontinued | categoryID |
| --- | --- | --- | --- | --- | --- |
| 100 | Wheat bread | 10 | 20 | 1 | 3 |
| 101 | White bread | 5 boxes | 19.99 | 0 | 3 |
| 102 | Midnight Mango Fizz | 24 - 12 oz bottles | 19 | 0 | 1 |
| 103 | Savory Fire Sauce | 12 - 550 ml bottles | 10 | 0 | 2 |

* Update the price and discontinued status for from below table ‘updated\_products’ only if there are matching products and updated\_products .discontinued =0
* If there are matching products and updated\_products .discontinued =1 then delete

* Insert any new products from updated\_products that don’t exist in products only if updated\_products .discontinued =0.

Answer:

**CREATE TEMP TABLE updated\_products**

**(productid integer,**

**productname text,**

**quantityperunit text,**

**unitprice real,**

**discontinued integer,**

**categoryid integer**

**);**

**INSERT INTO updated\_products (productID, productName, quantityPerUnit, unitPrice, discontinued, categoryID)**

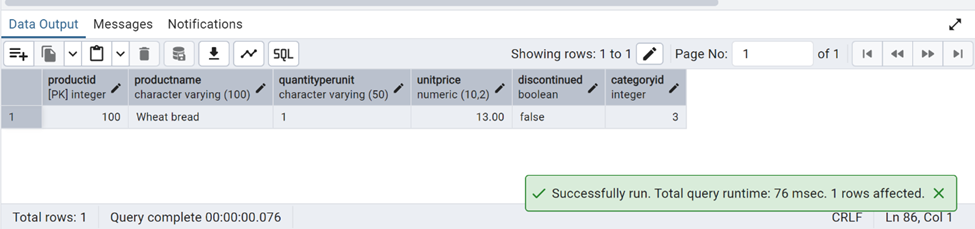
**VALUES**

**(100, 'Wheat bread', '10', 20, 1, 5),**

**(101, 'White bread', '5 boxes', 19.99, 0, 5),**

**(102, 'Midnight Mango Fizz', '24 - 12 oz bottles', 19, 0, 1),**

**(103, 'Savory Fire Sauce', '12 - 550 ml bottles', 10, 0, 2);**



**MERGE INTO products AS p**

**USING updated\_products AS up**

**ON p.productID = up.productID**

**WHEN MATCHED AND up.discontinued = 0 THEN**

**UPDATE SET**

**unitPrice = up.unitPrice,**

**discontinued = up.discontinued**

**WHEN MATCHED AND up.discontinued = 1 THEN**

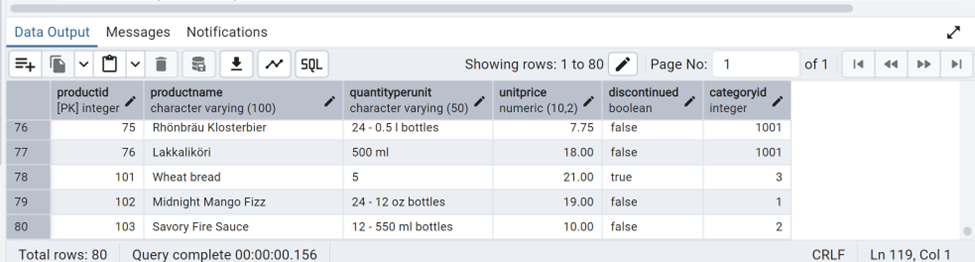
**DELETE**

**WHEN NOT MATCHED AND up.discontinued = 0 THEN**

**INSERT (productID, productName, quantityPerUnit, unitPrice, discontinued, categoryID)**

**VALUES (up.productID, up.productName, up.quantityPerUnit, up.unitPrice, up.discontinued, up.categoryID);**

**select \* from products;**

****

**USE NEW Northwind DB:**

7) List all orders with employee full names. (Inner join)

Query:

SELECT o.order\_id, (e.first\_name||' ' ||e.last\_name) AS full\_name

FROM orders o

INNER JOIN employees e ON o.employee\_id = e.employee\_id;

Screenshot:

