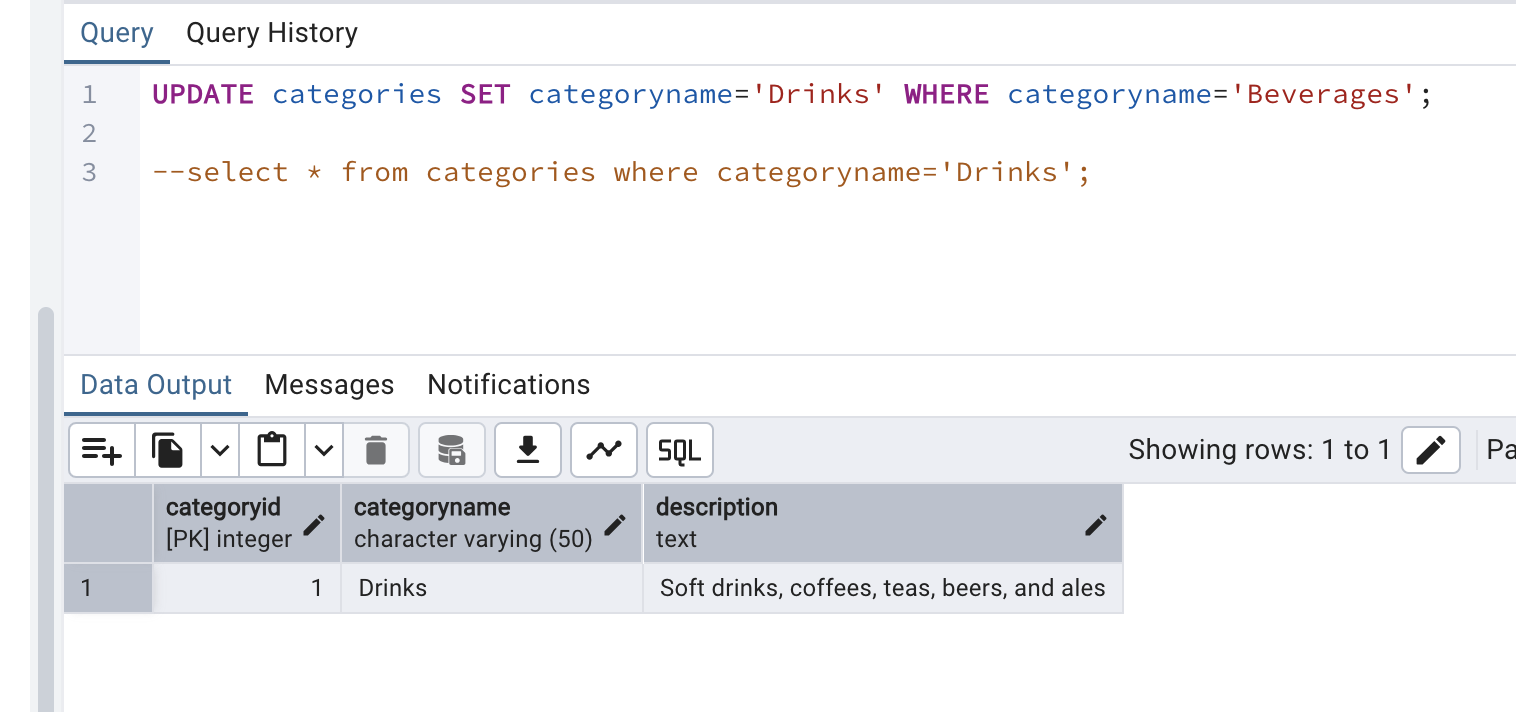
**Day 3**

**USE Northwind from Kaggle:**

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

UPDATE categories SET categoryname='Drinks' WHERE categoryname='Beverages';



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

--Insert new record

INSERT INTO shippers (shipperid, companyname)

VALUES (4,'CCC');

A screenshot of a computer

AI-generated content may be incorrect.

--Delete new record from shippers

DELETE FROM shippers WHERE shipperid=4 ;

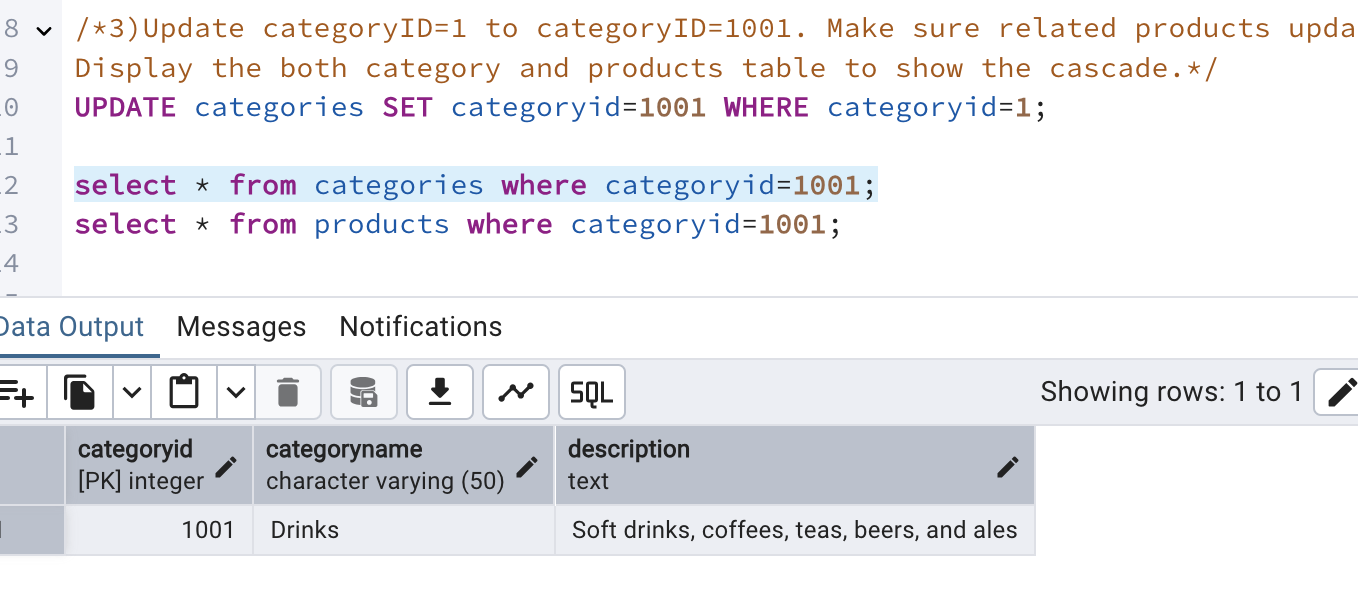
A screenshot of a computer

AI-generated content may be incorrect.

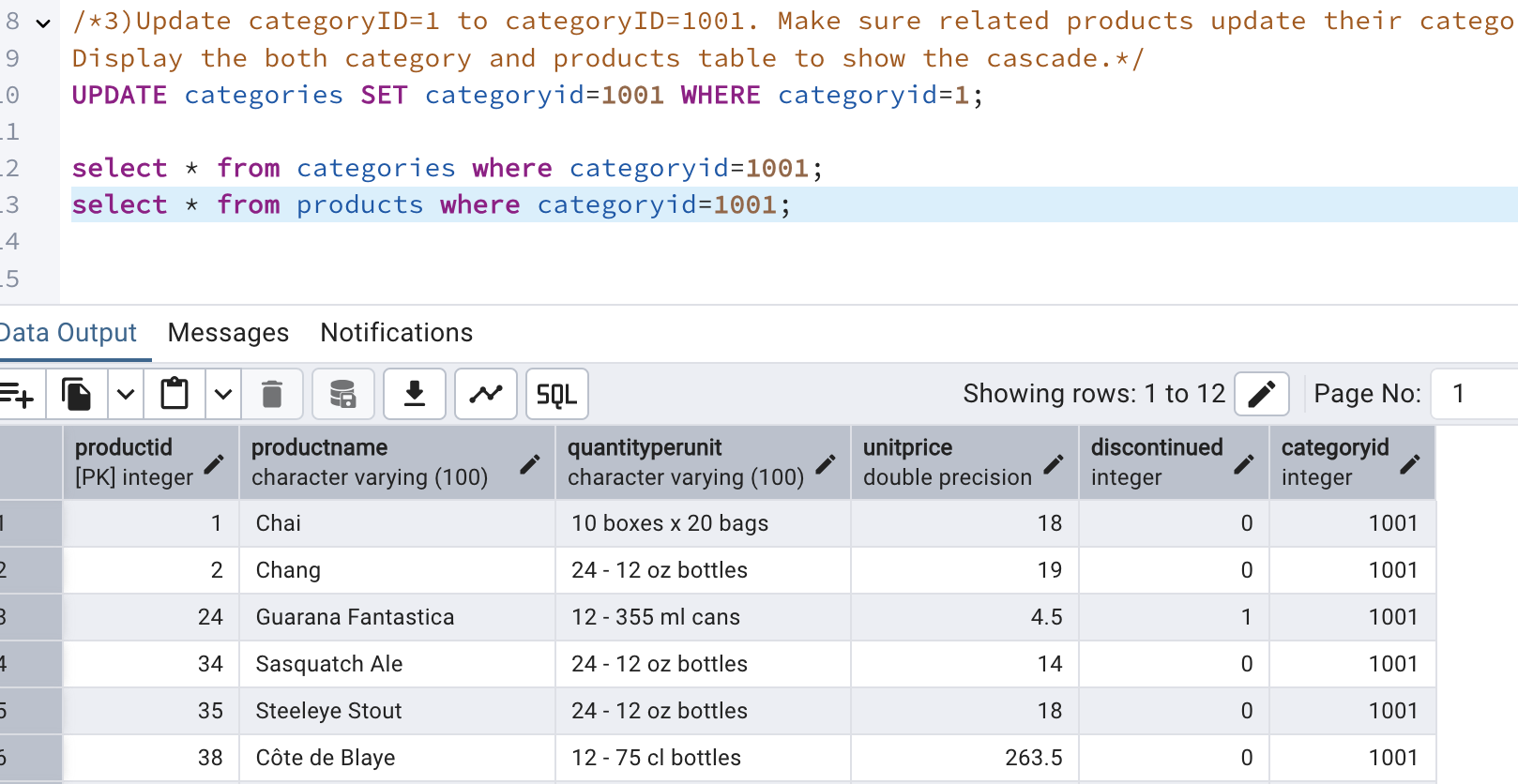
3)      Update categoryID=1 to categoryID=1001. Make sure related products update their categoryID too. Display the both category and products table to show the cascade.

UPDATE categories SET categoryid=1001 WHERE categoryid=1;

select \* from categories where categoryid=1001;



select \* from products where categoryid=1001;



 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)

DELETE FROM categories WHERE categoryid=3;

/\*ALTER TABLE order\_details

DROP CONSTRAINT order\_details\_productid\_fkey;

ALTER TABLE order\_details

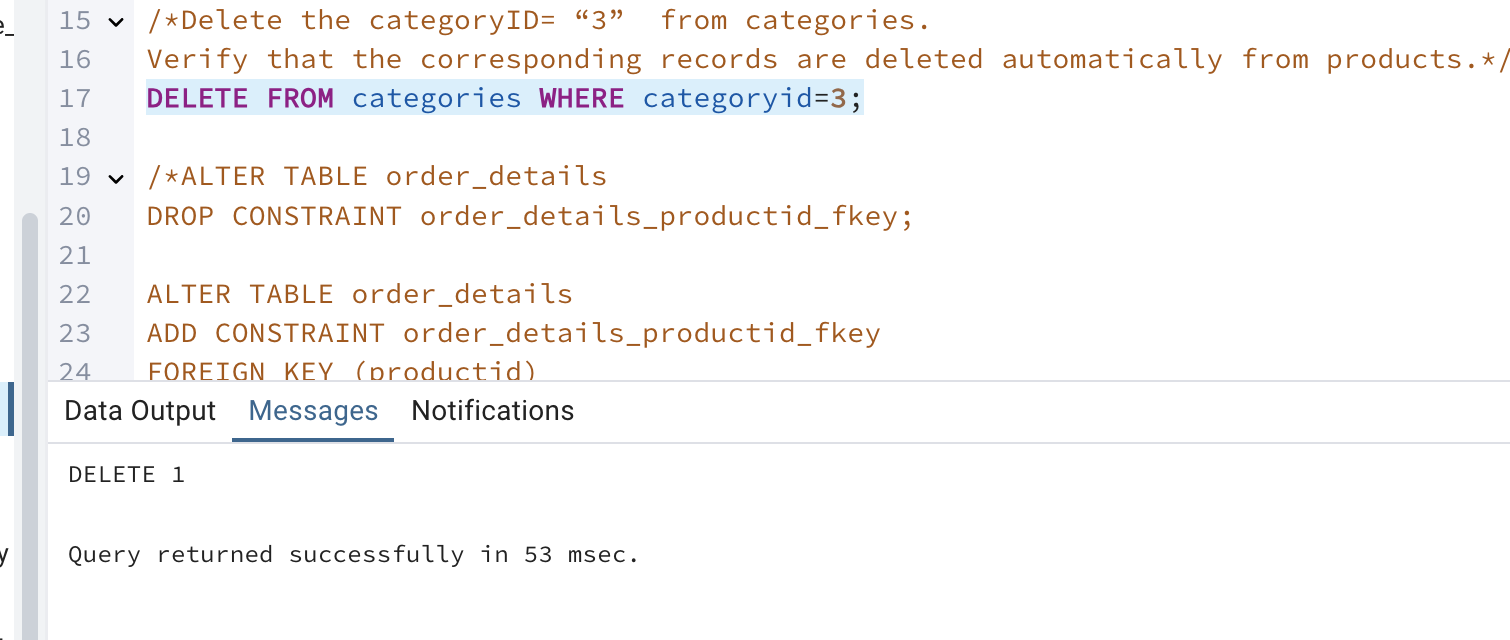
ADD CONSTRAINT order\_details\_productid\_fkey

FOREIGN KEY (productid)

REFERENCES products(productid)

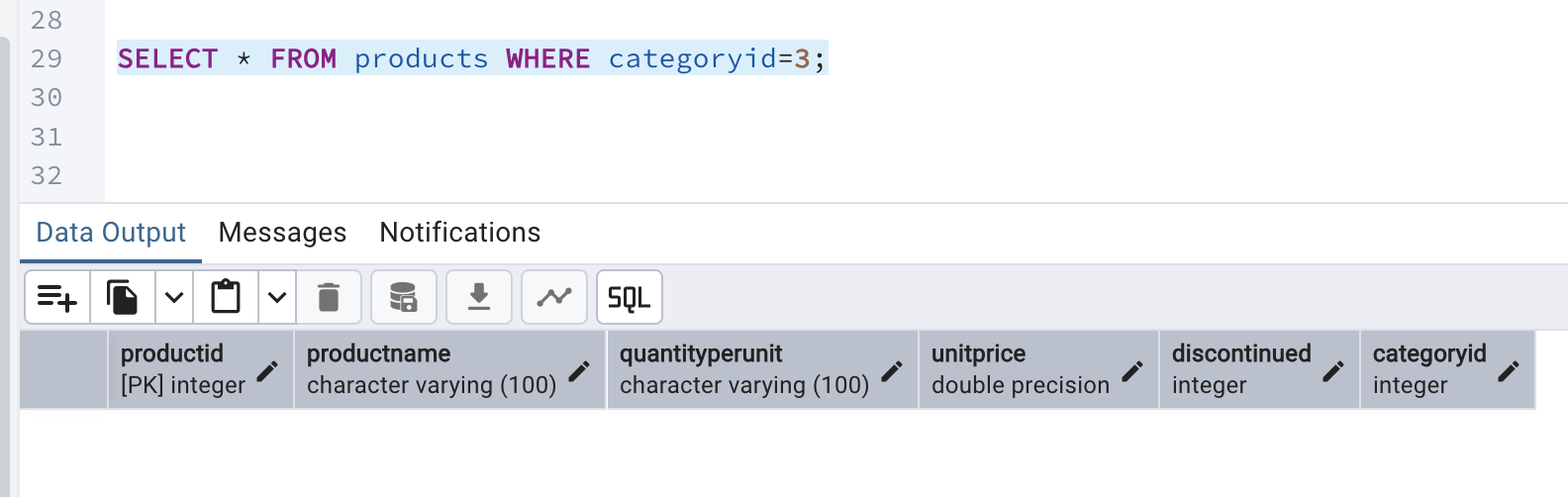
ON UPDATE CASCADE

ON DELETE CASCADE;\*/



Verify that the corresponding records are deleted automatically from products.

SELECT \* FROM products WHERE categoryid=3;



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 customers where customerid='VINET' ;

--select \* from orders where customerid='VINET' ;

ALTER TABLE orders ALTER COLUMN customerid DROP NOT NULL;

ALTER TABLE orders

DROP CONSTRAINT fk\_customers;

ALTER TABLE orders

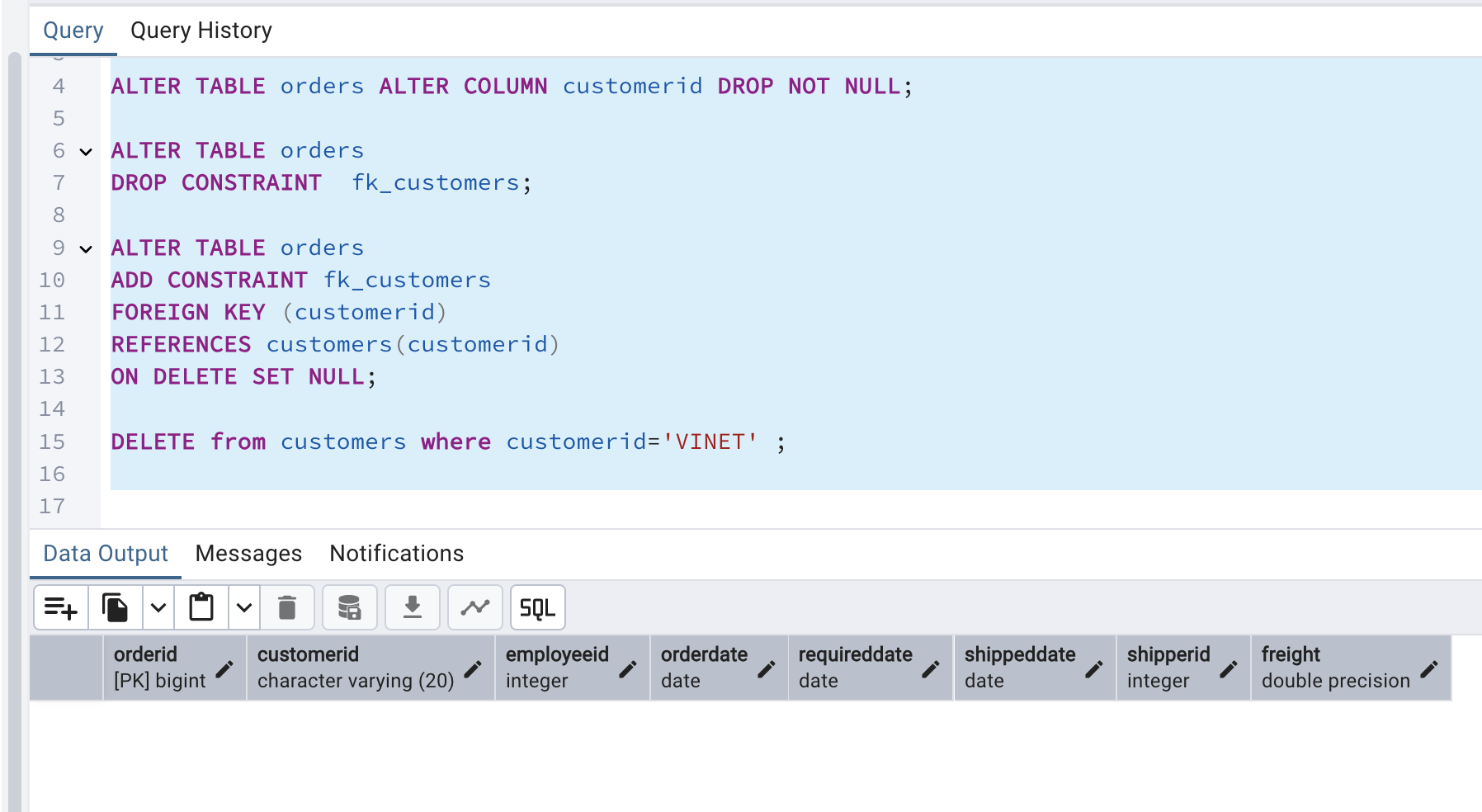
ADD CONSTRAINT fk\_customers

FOREIGN KEY (customerid)

REFERENCES customers(customerid)

ON DELETE SET NULL;

DELETE from customers where customerid='VINET' ;



5)      Insert the following data to Products using UPSERT:

product\_id = 100, product\_name = Wheat bread, quantityperunit=1,unitprice = 13, discontinued = 0, categoryID=5

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

product\_id = 100, product\_name = Wheat bread, quantityperunit=10 boxes,unitprice = 13, discontinued = 0, categoryID=5

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

INSERT INTO products (productid,productname,quantityperunit,unitprice,discontinued,categoryID)

VALUES

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

ON CONFLICT (productid)

DO UPDATE SET

quantityperunit = EXCLUDED.quantityperunit;

INSERT INTO products (productid,productname,quantityperunit,unitprice,discontinued,categoryID)

VALUES

(101,'White bread','5 boxes',13,0,5)

ON CONFLICT (productid)

DO UPDATE SET

quantityperunit = EXCLUDED.quantityperunit;

INSERT INTO products (productid,productname,quantityperunit,unitprice,discontinued,categoryID)

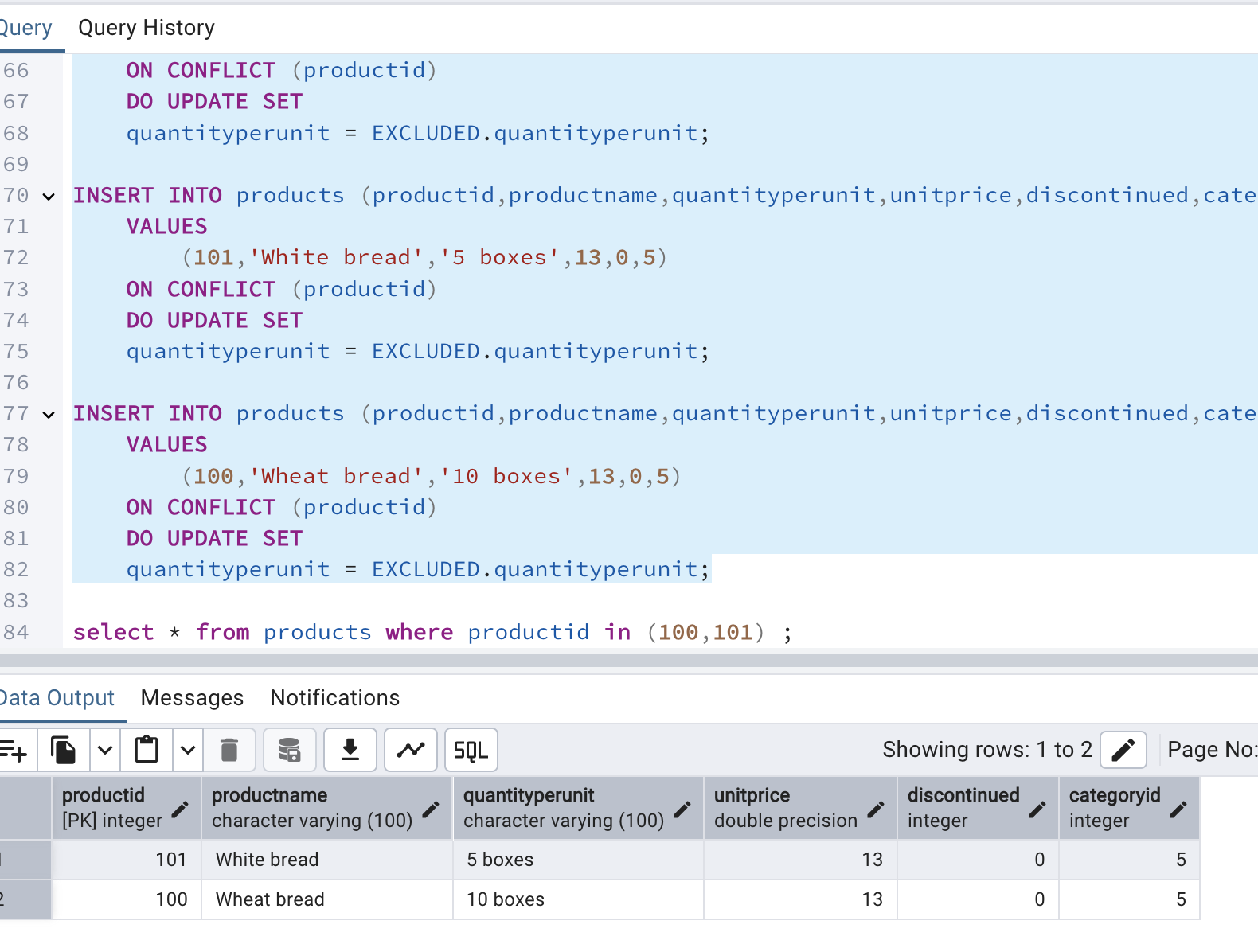
VALUES

(100,'Wheat bread','10 boxes',13,0,5)

ON CONFLICT (productid)

DO UPDATE SET

quantityperunit = EXCLUDED.quantityperunit;



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 | 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 |

* 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.

MERGE INTO products p

USING (

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)

)AS updated\_products (productid,productname,quantityperunit,unitprice,discontinued,categoryid)

ON p.productid=updated\_products.productid

WHEN MATCHED AND updated\_products.discontinued=1 THEN

DELETE

WHEN MATCHED AND updated\_products.discontinued=0 THEN

UPDATE SET

unitprice=updated\_products.unitprice,

discontinued=updated\_products.discontinued

WHEN NOT MATCHED AND updated\_products.discontinued =0 THEN

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

VALUES

(updated\_products.productid,updated\_products.productname,updated\_products.quantityperunit,updated\_products.unitprice,updated\_products.discontinued,updated\_products.categoryid);

A screenshot of a computer

AI-generated content may be incorrect.

NOTE: This error is because I am using the version pgadmin-14 that doesn’t support MERGE function.