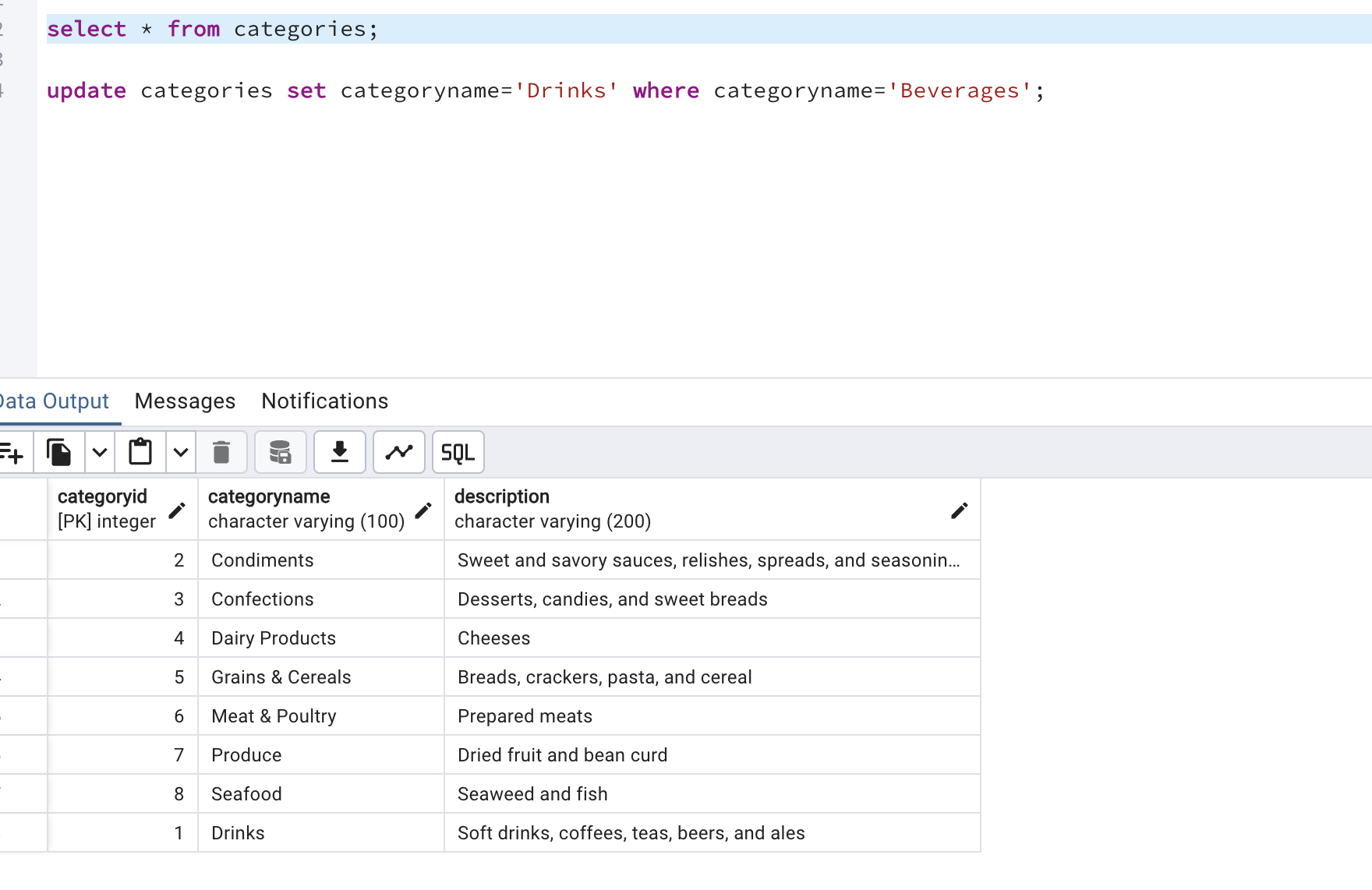
1) Update the categoryName From “Beverages” to "Drinks" in the categories table.

select \* from categories;

update categories set categoryname='Drinks' where categoryname='Beverages';





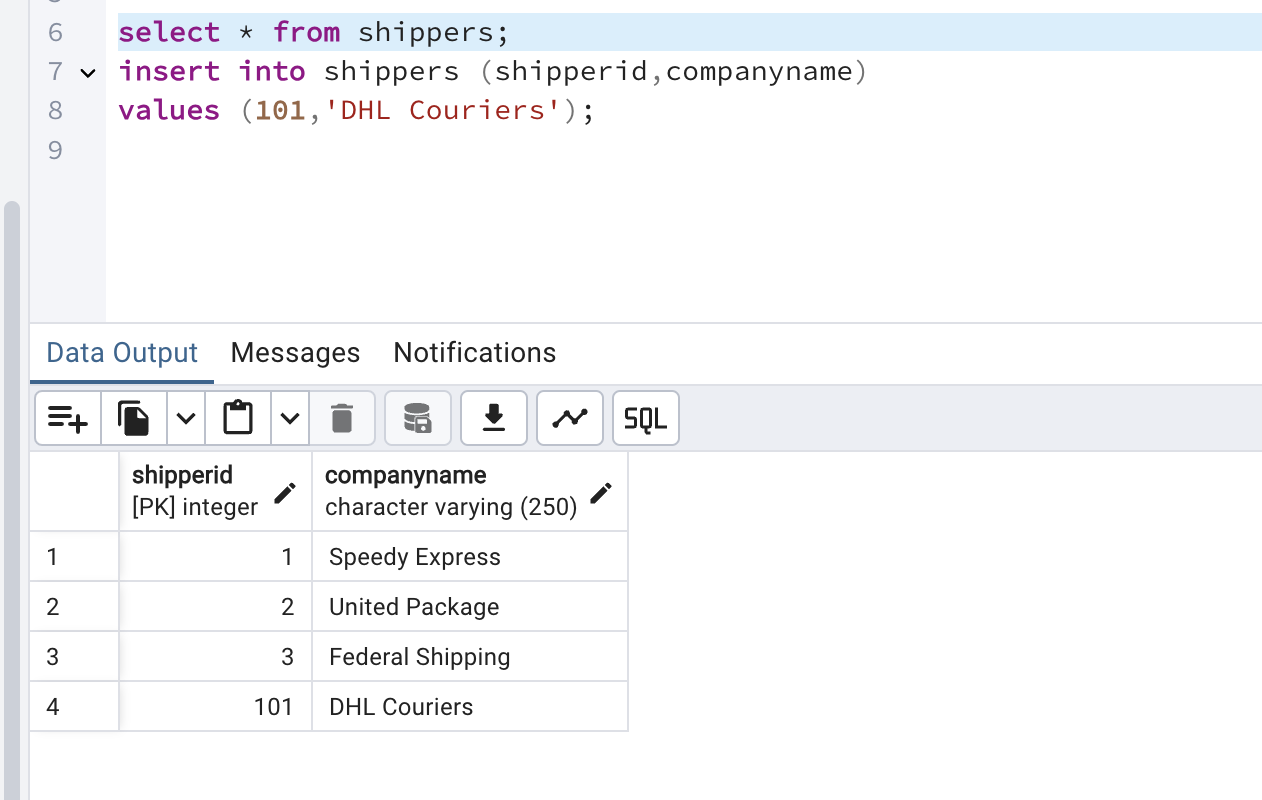
2) Insert into shipper new record (give any values) Delete that new record from shippers table.

select \* from shippers;

insert into shippers (shipperid,companyname)

values (101,'DHL Couriers');

delete from shippers where shipperid=101;





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, add ON DELETE CASCADE for order\_details(productid) )

alter table products

add constraint fk\_products\_categoryID

foreign key (categoryid)

references categories(categoryid)

on update cascade

on delete cascade;

ALTER TABLE products

DROP CONSTRAINT fk\_categoryID;

update categories

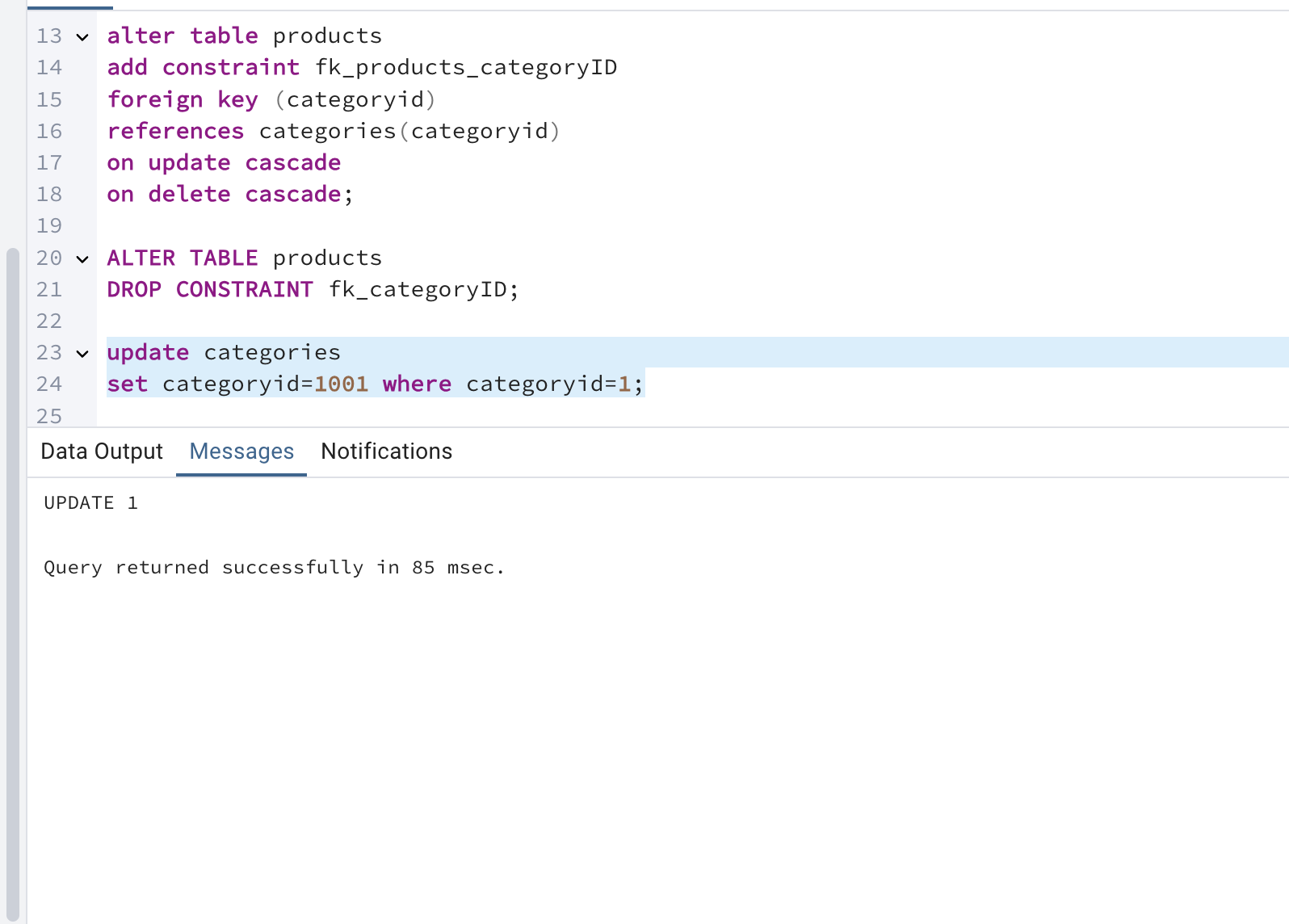
set categoryid=1001 where categoryid=1;

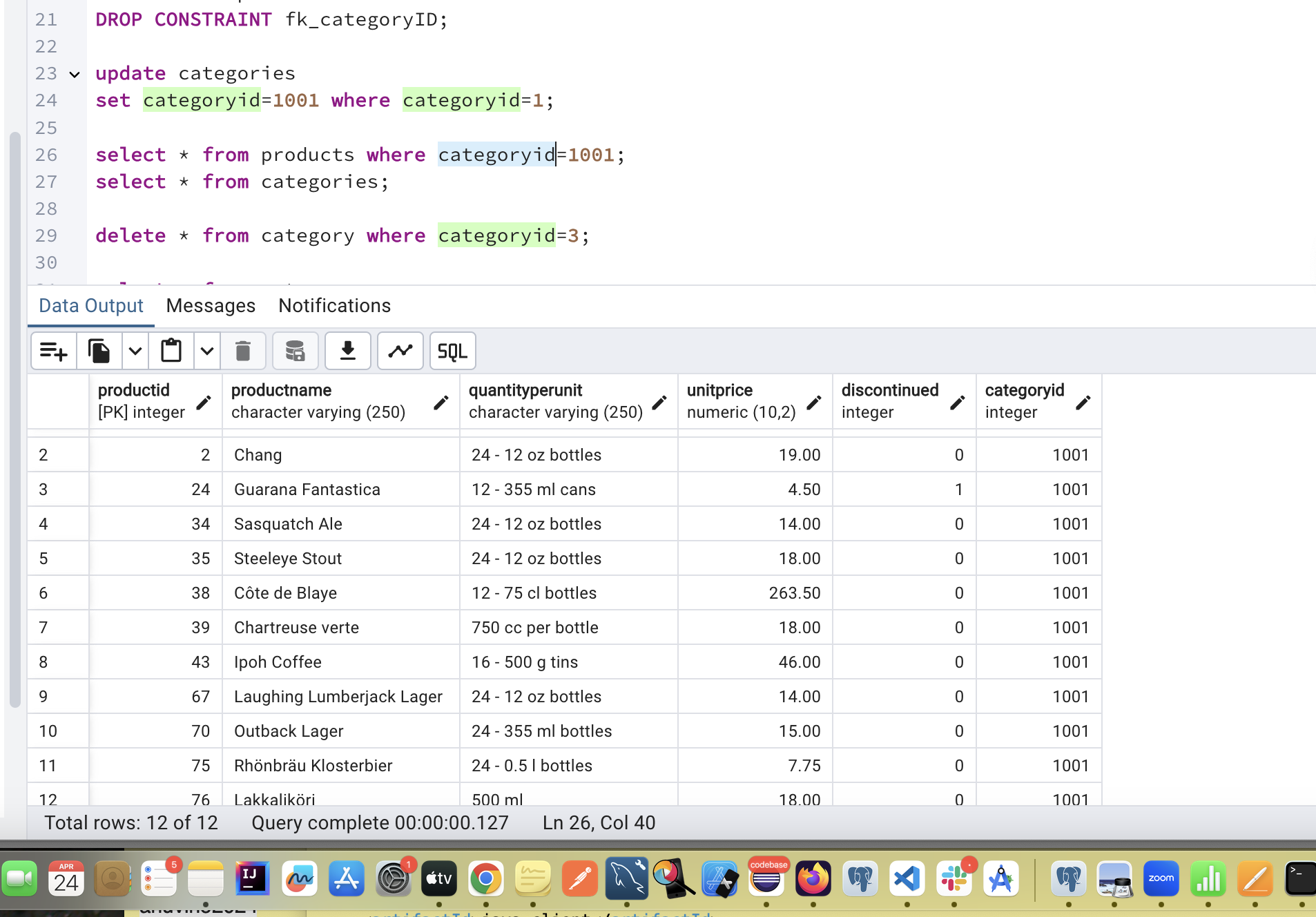
select \* from products where categoryid=1001;

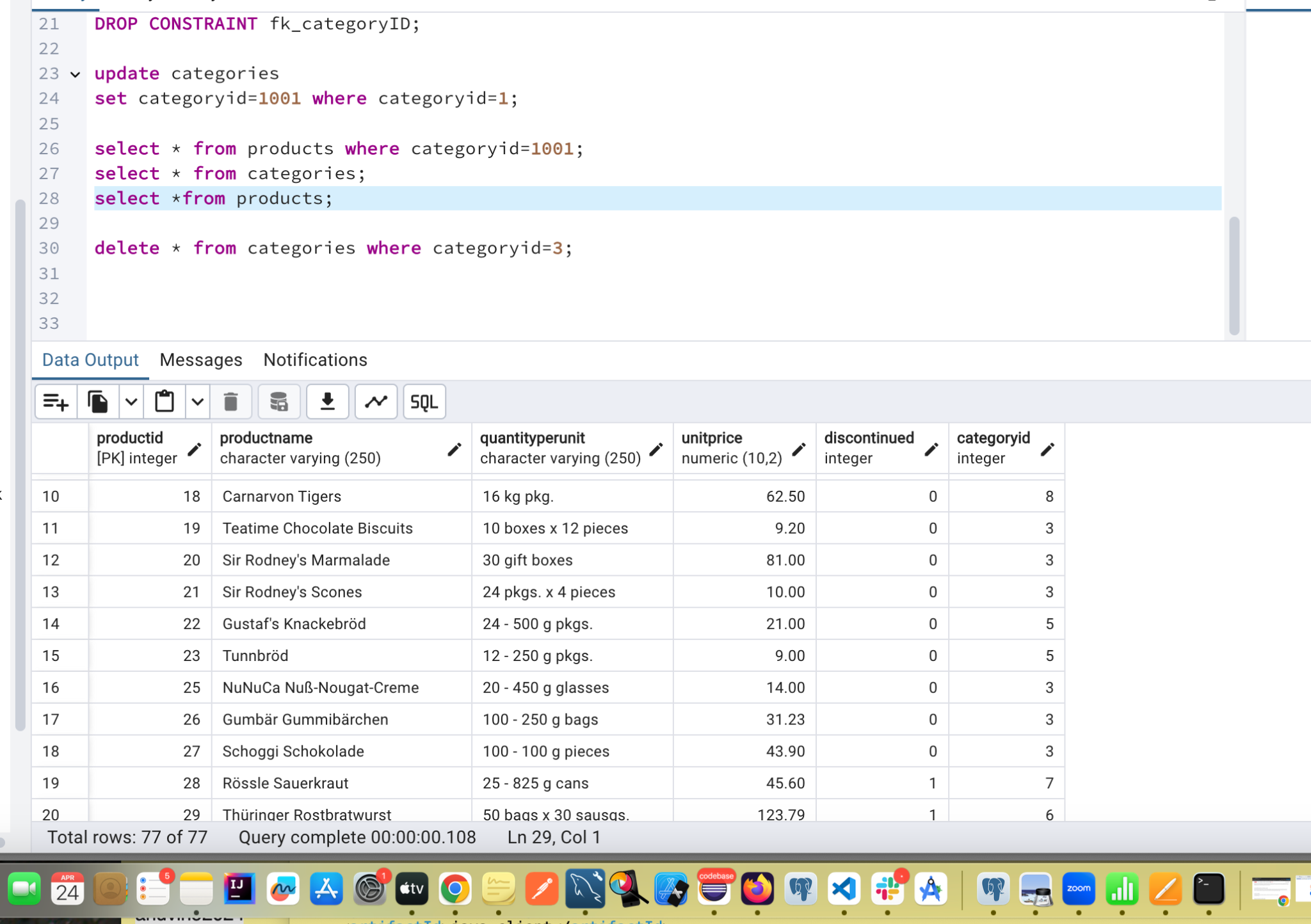
select \* from categories;

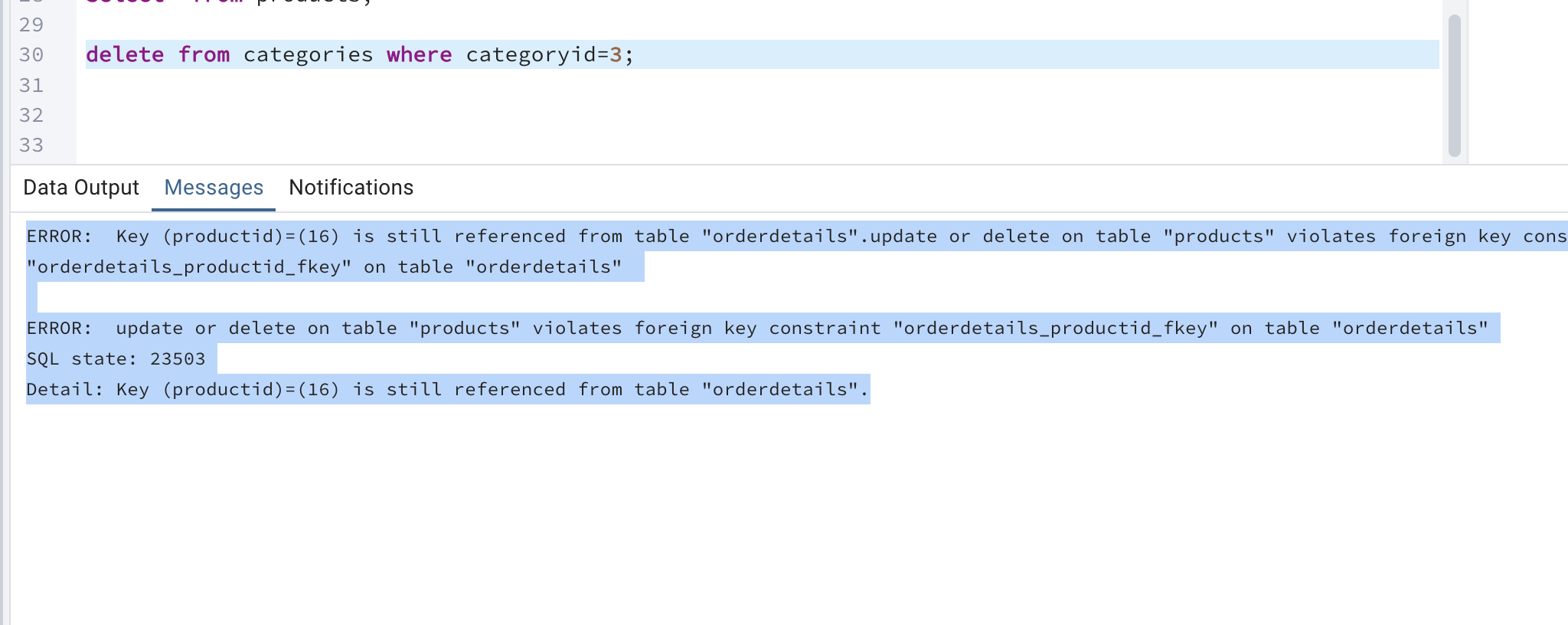
select \*from products;

delete from categories where categoryid=3;









So lets add constraint to fk in order details table and hen delete the category=3;

alter table orderdetails

add constraint orderd\_productid\_fkey

foreign key (productid)

references products(productid)

on update cascade

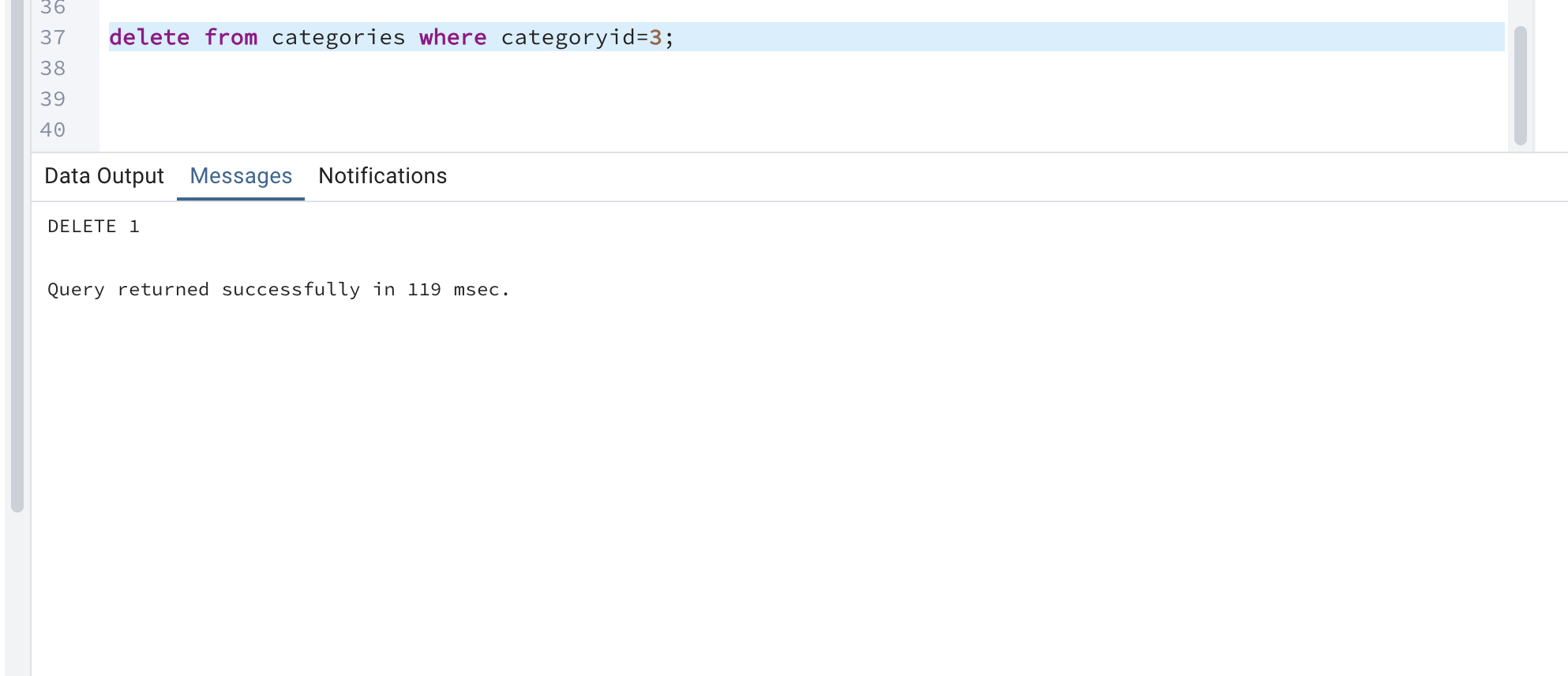
on delete cascade;

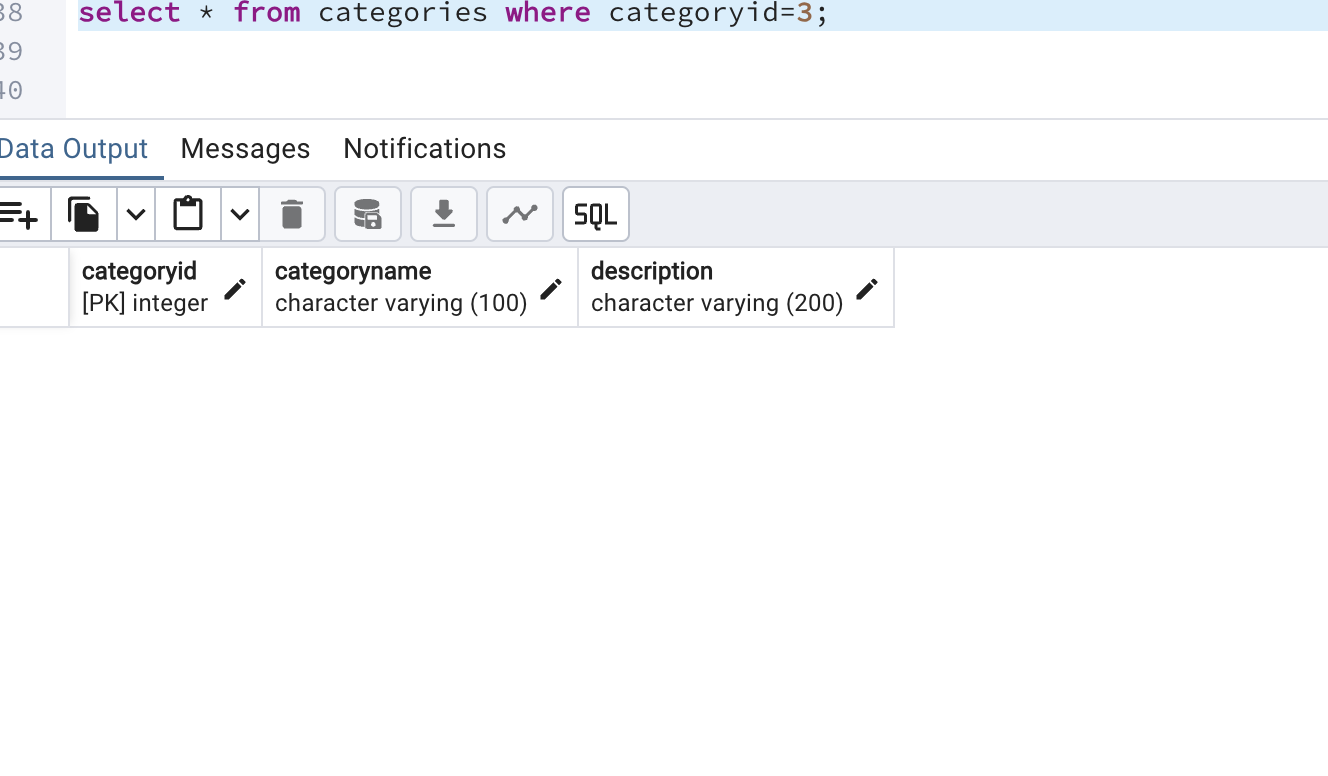
ALTER TABLE orderdetails

DROP CONSTRAINT orderdetails\_productid\_fkey;

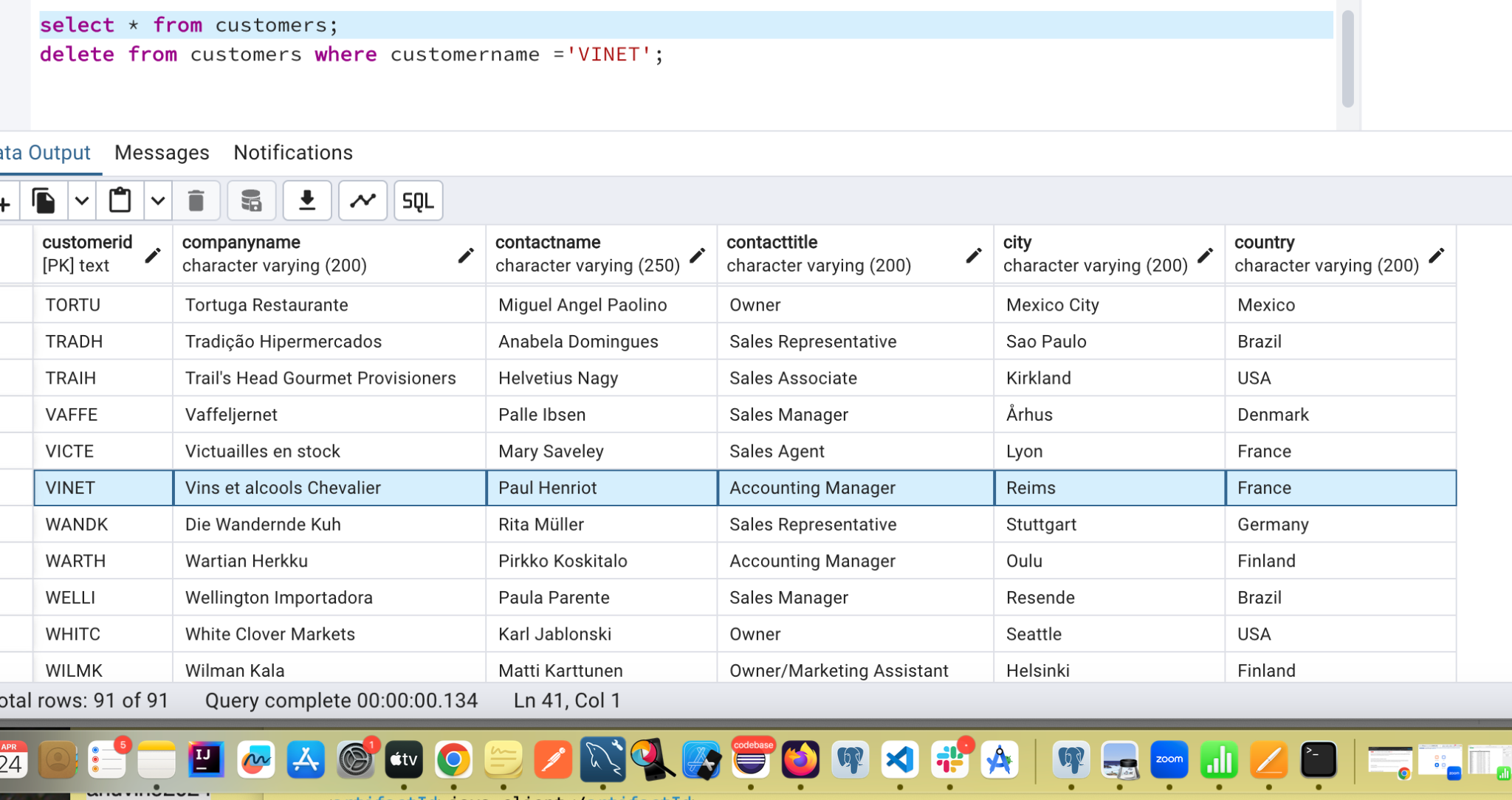
delete from categories where categoryid=3;

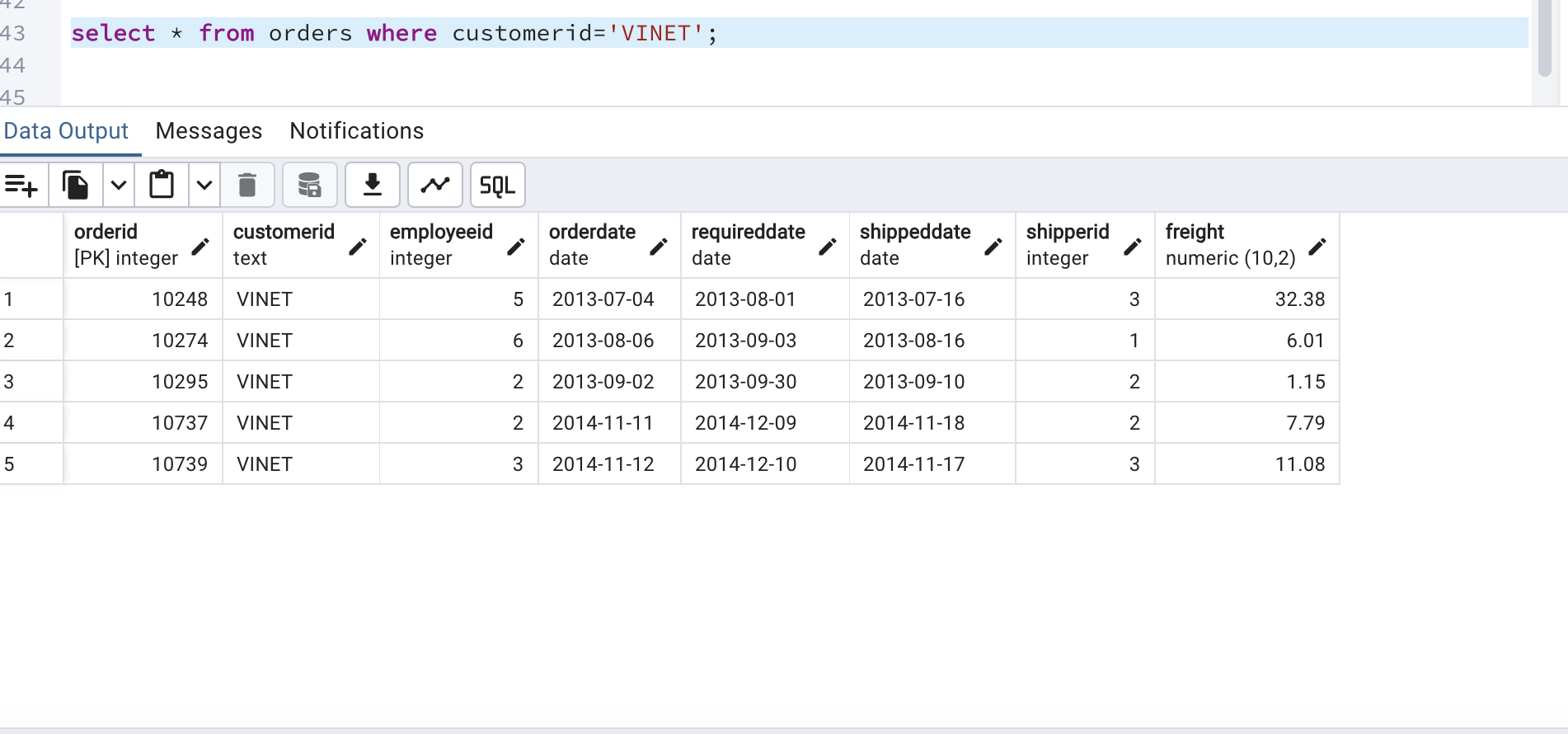
select \* from categories where categoryid=3;





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';;

delete from customers where customerid ='VINET';

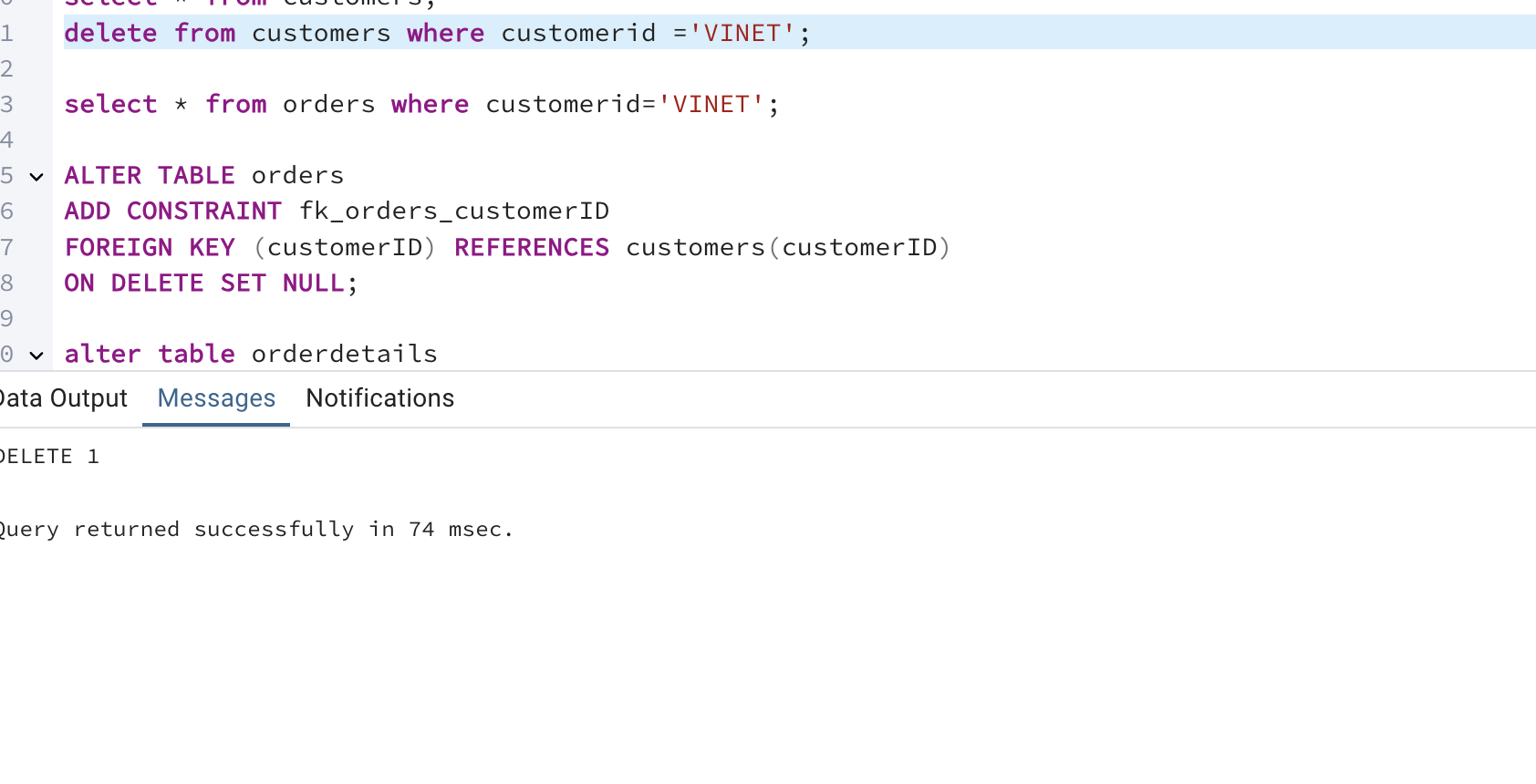
select \* from orders where customerid='VINET';

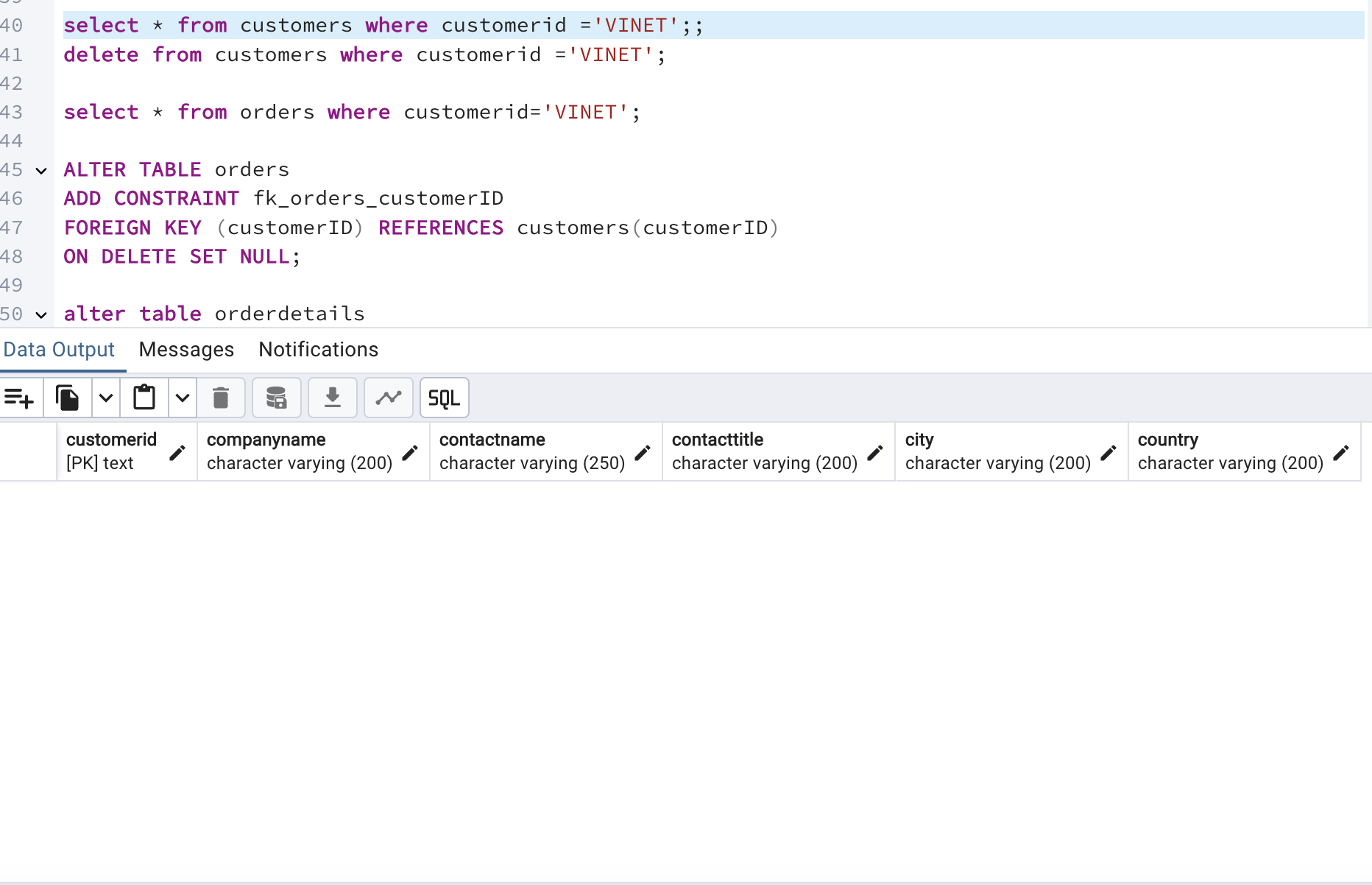
ALTER TABLE orders

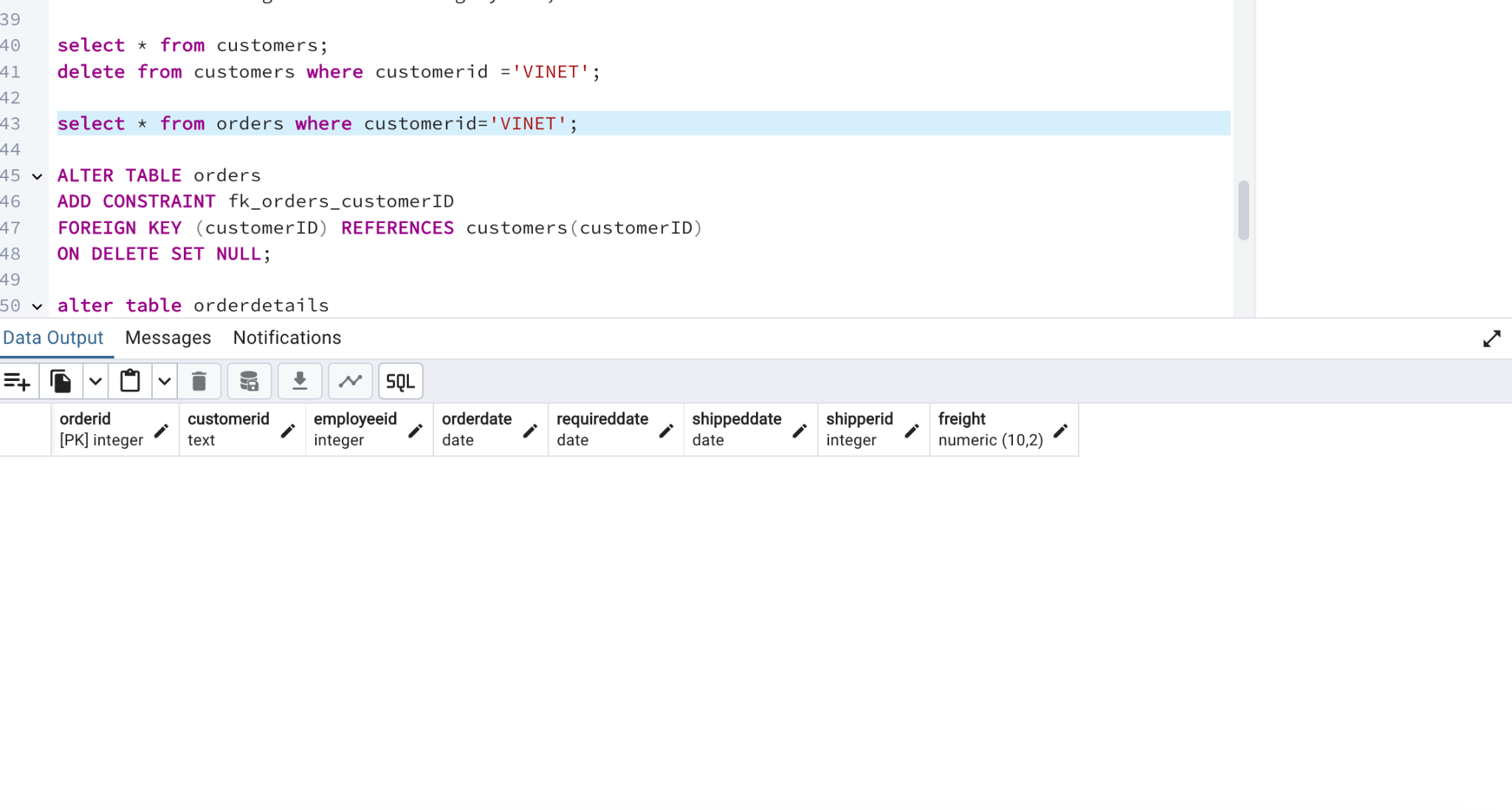
ADD CONSTRAINT fk\_orders\_customerID

FOREIGN KEY (customerID) REFERENCES customers(customerID)

ON DELETE SET NULL;







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)

select \* from products where productid=100 ;

insert into products (productid,productname,quantityperunit,unitprice,discontinued,categoryid)

values(100,'WheatBread','1',13,0,5)

on conflict (productid)

do update

set productname=excluded.productname,

quantityperunit=excluded.quantityperunit,

unitprice=excluded.unitprice,

discontinued=excluded.discontinued,

categoryid=excluded.categoryid;

insert into products (productid,productname,quantityperunit,unitprice,discontinued,categoryid)

values(101,'WheatBread','5 boxes',13,0,5)

on conflict (productid)

do update

set productname=excluded.productname,

quantityperunit=excluded.quantityperunit,

unitprice=excluded.unitprice,

discontinued=excluded.discontinued,

categoryid=excluded.categoryid;

select \* from products where productid=101 ;

insert into products (productid,productname,quantityperunit,unitprice,discontinued,categoryid)

values(100,'WheatBread','10 boxes',13,0,5)

on conflict (productid)

do update

set productname=excluded.productname,

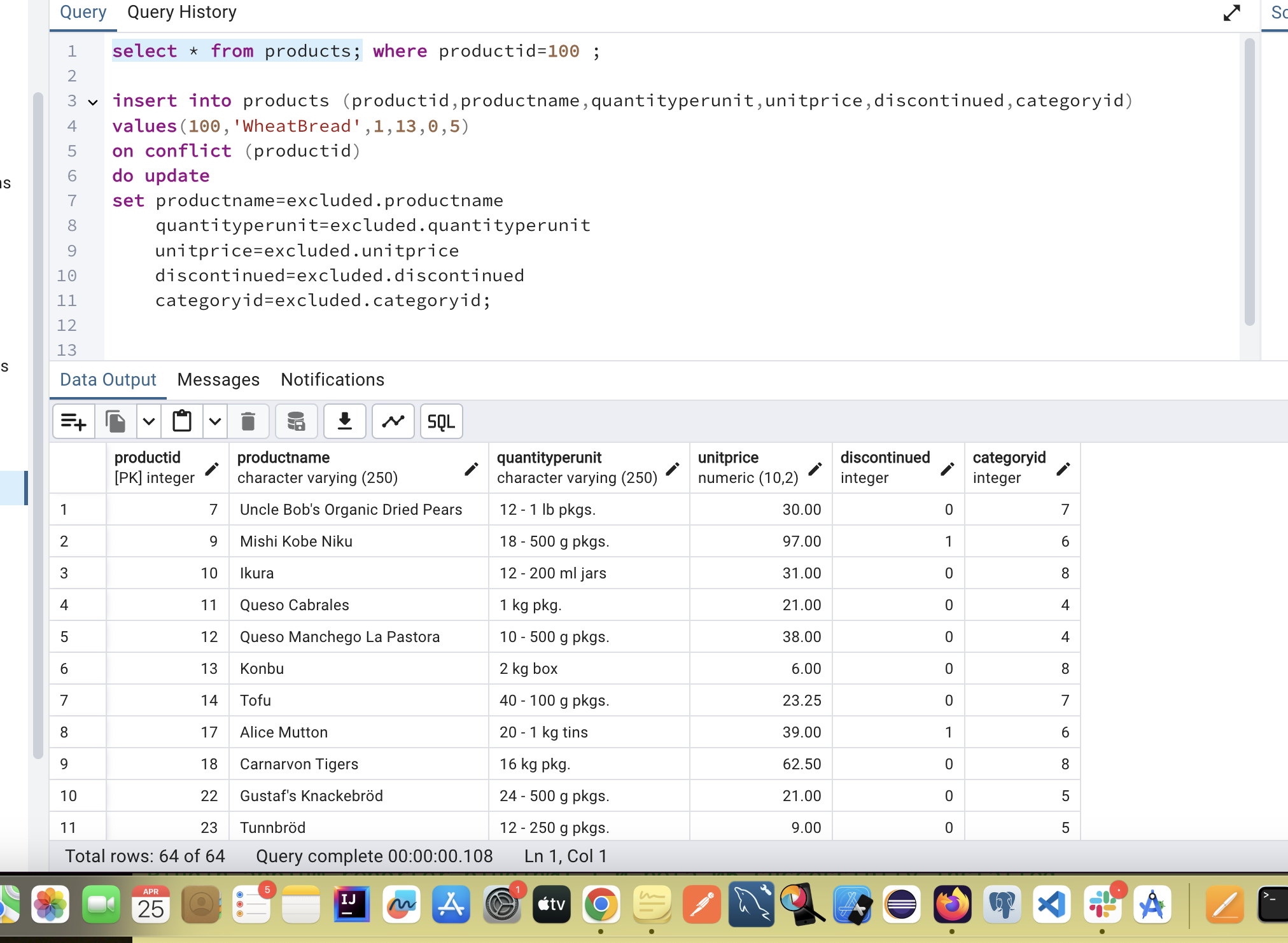
quantityperunit=excluded.quantityperunit,

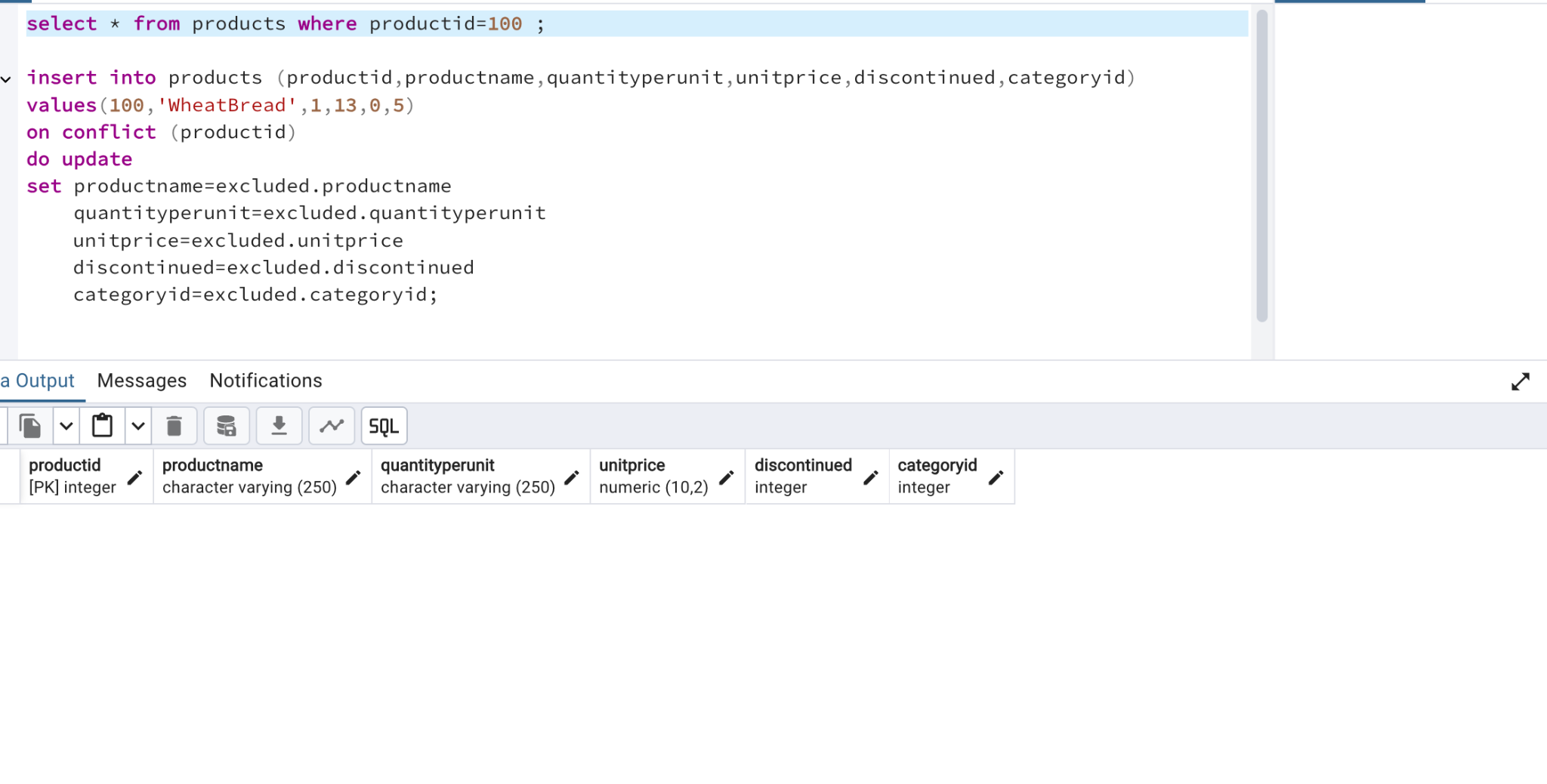
unitprice=excluded.unitprice,

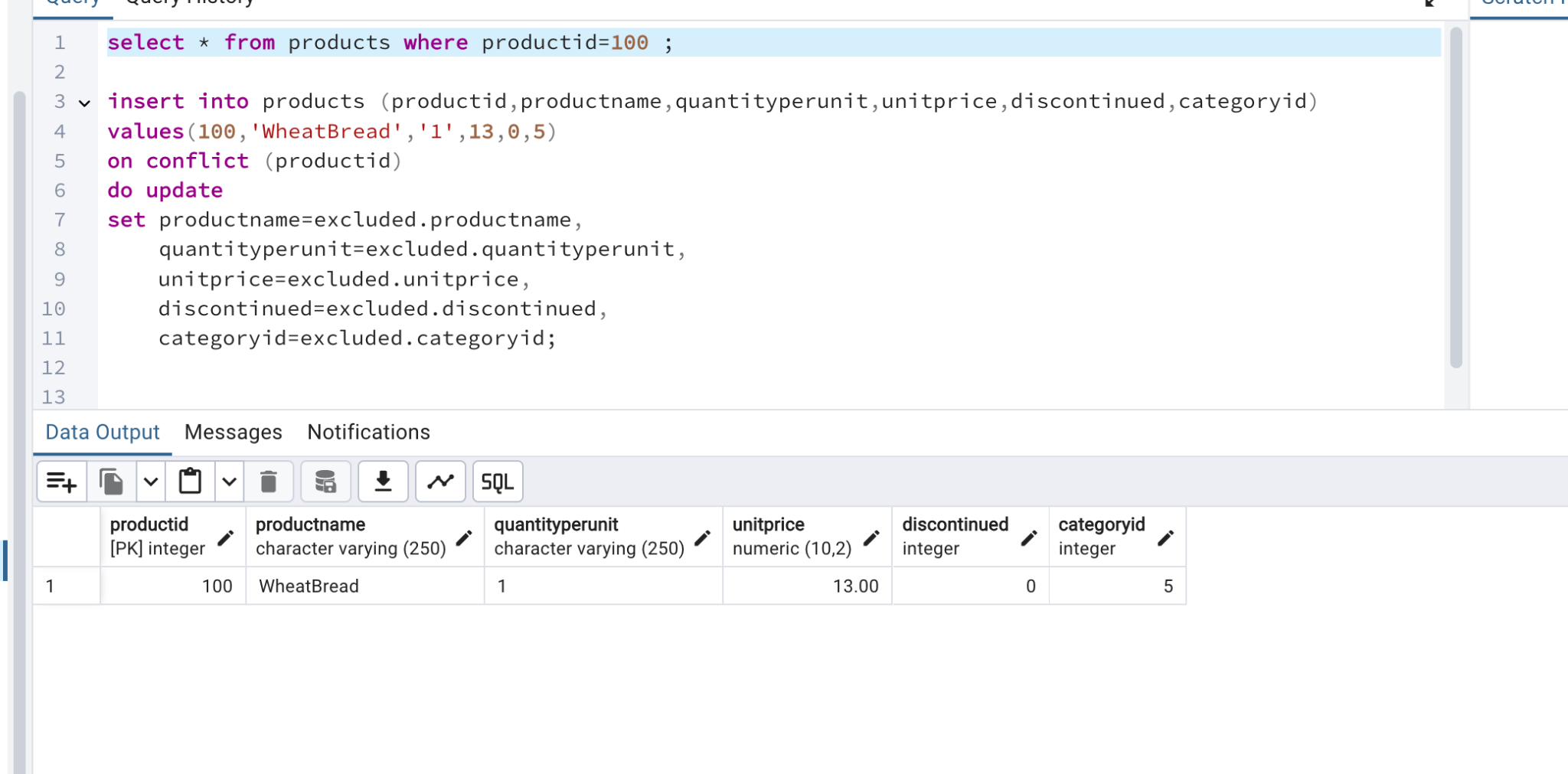
discontinued=excluded.discontinued,

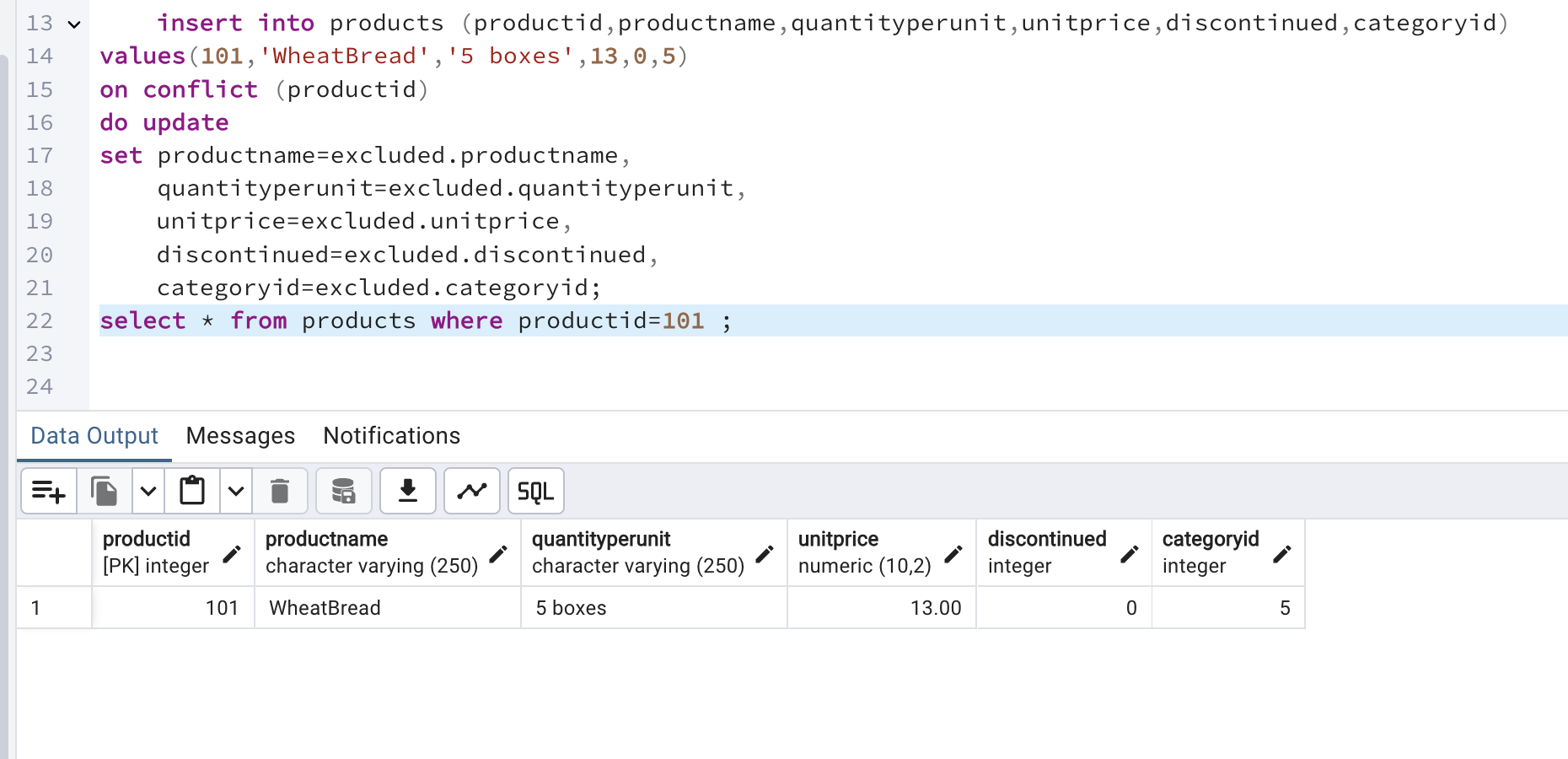
categoryid=excluded.categoryid;

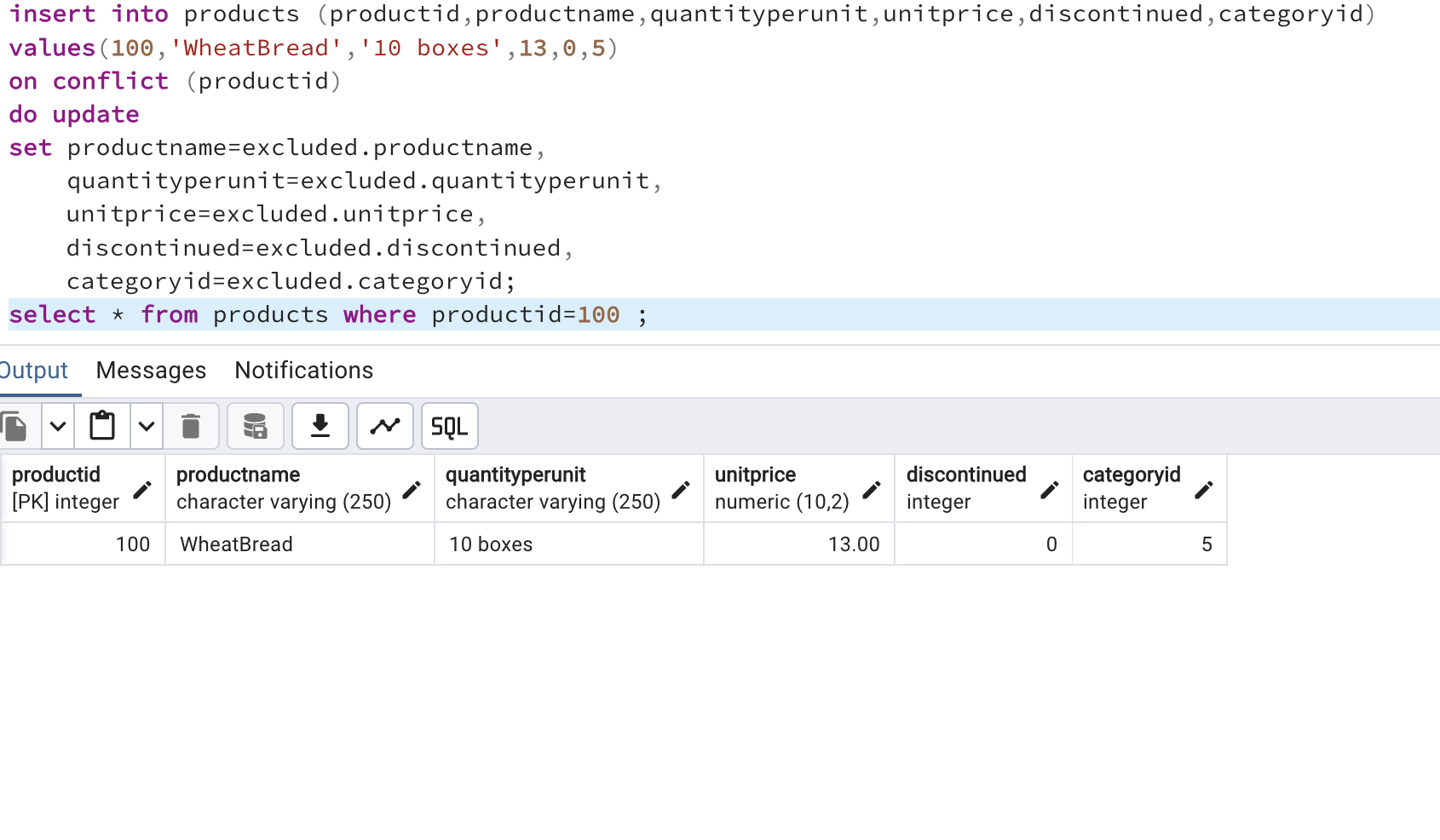
select \* from products where productid=100 ;











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

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

merge into products p

using (

values(100,'WheatBread','10',20,1,5),

(101,'WhiteBread','5 boxes',19.99,0,5),

(102,'Midnight Mango Fizz','24 - 12 oz bottles',19,0,2),

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

as updatedproducts(productid,productname,quantityperunit,unitprice,discontinued,categoryid)

on p.productid=updatedproducts.productid

When matched and updatedproducts.discontinued=0 then

update set

unitprice =updatedproducts.unitprice,

discontinued=updatedproducts.discontinued

When not matched and updatedproducts.discontinued=0 then

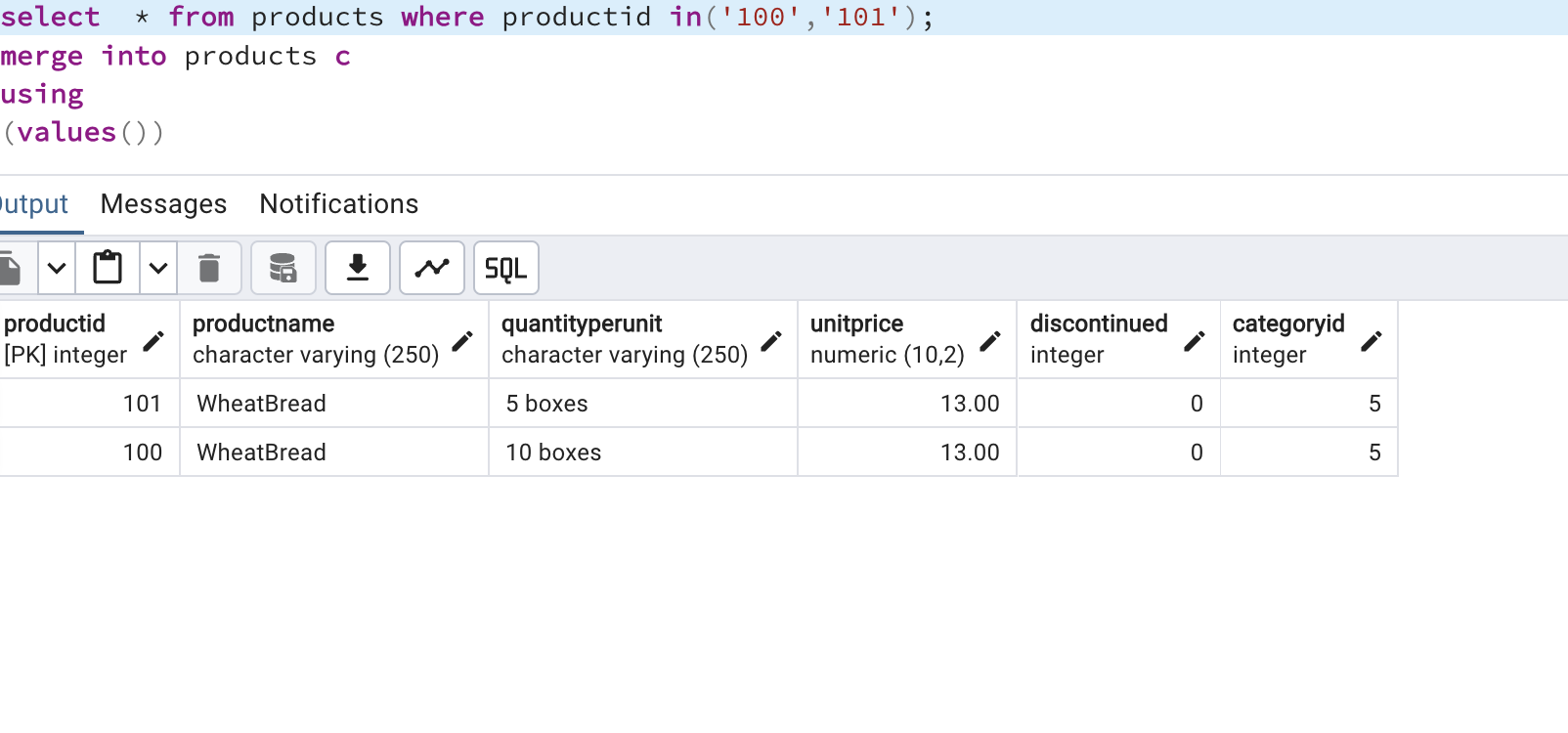
insert (productid,productname,quantityperunit,unitprice,discontinued,categoryid)

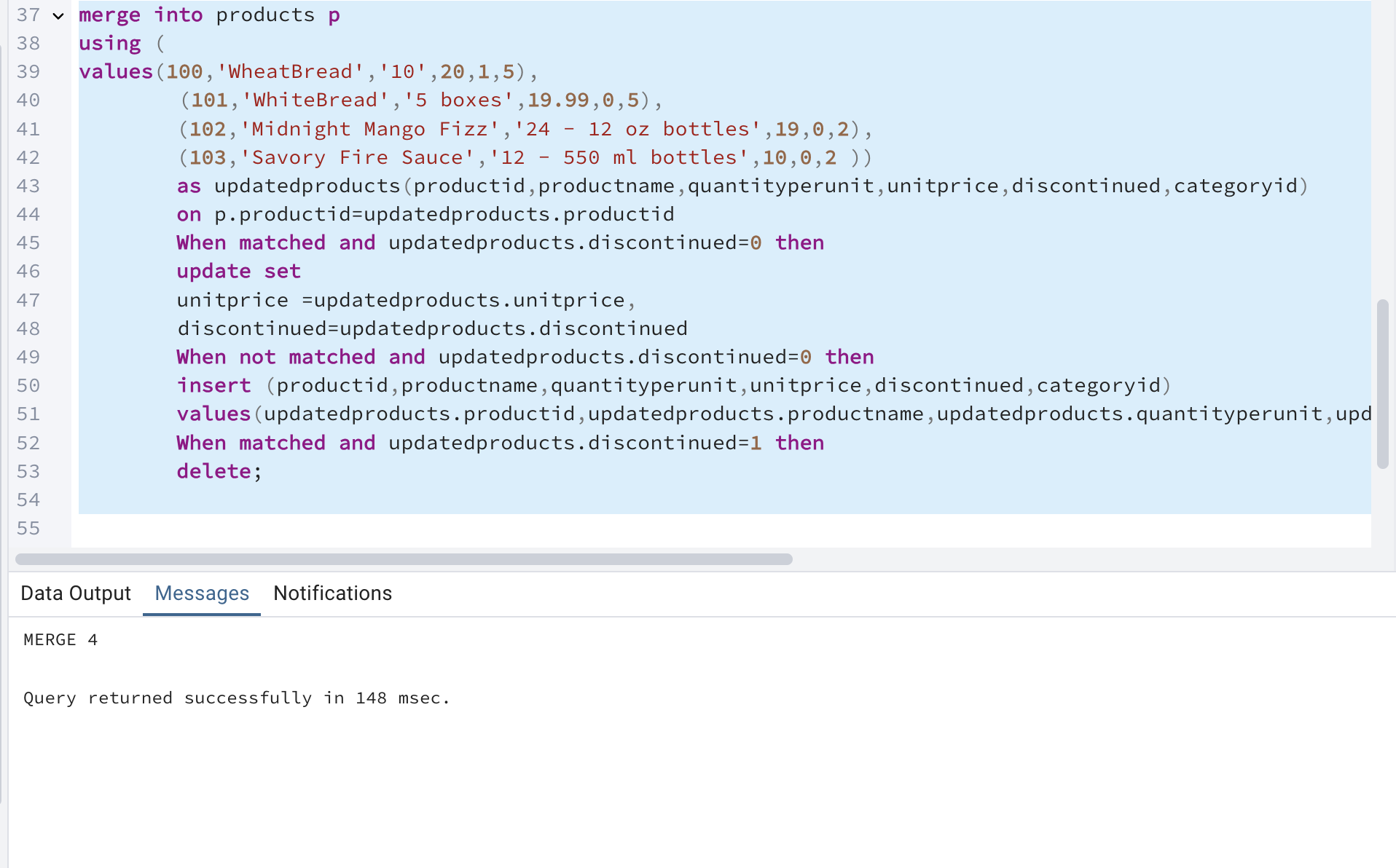
values(updatedproducts.productid,updatedproducts.productname,updatedproducts.quantityperunit,updatedproducts.unitprice,updatedproducts.discontinued,updatedproducts.categoryid)

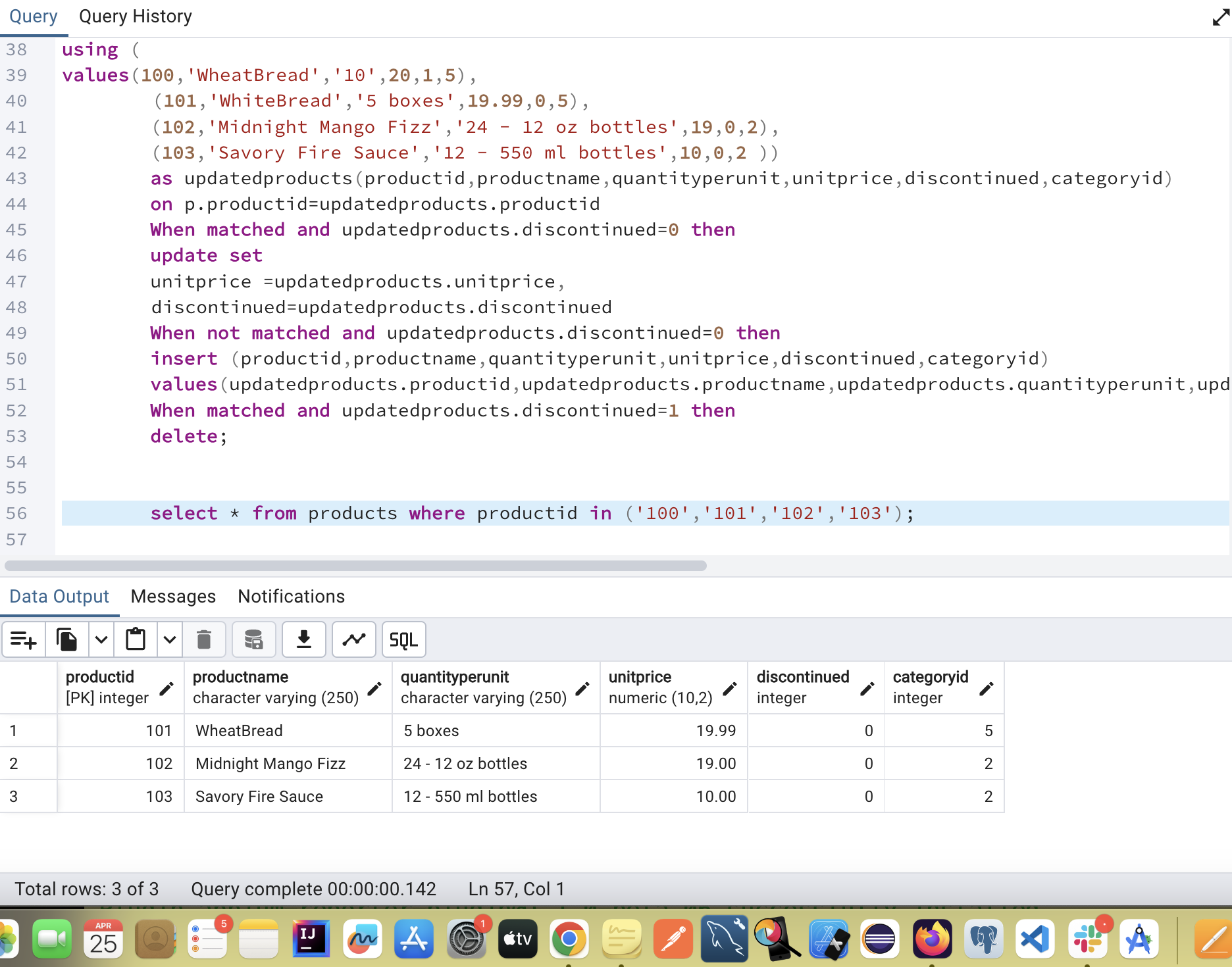
When matched and updatedproducts.discontinued=1 then

delete;

select \* from products where productid in ('100','101','102','103');

****

****

****

**USE NEW Northwind DB:**

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

Select o.employee\_id ,o.order\_id,o.order\_date,o.customer\_id,

e.first\_name || ''||e.last\_name as employeefullname from orders o

inner join employees e

on o.employee\_id=e.employee\_id;

