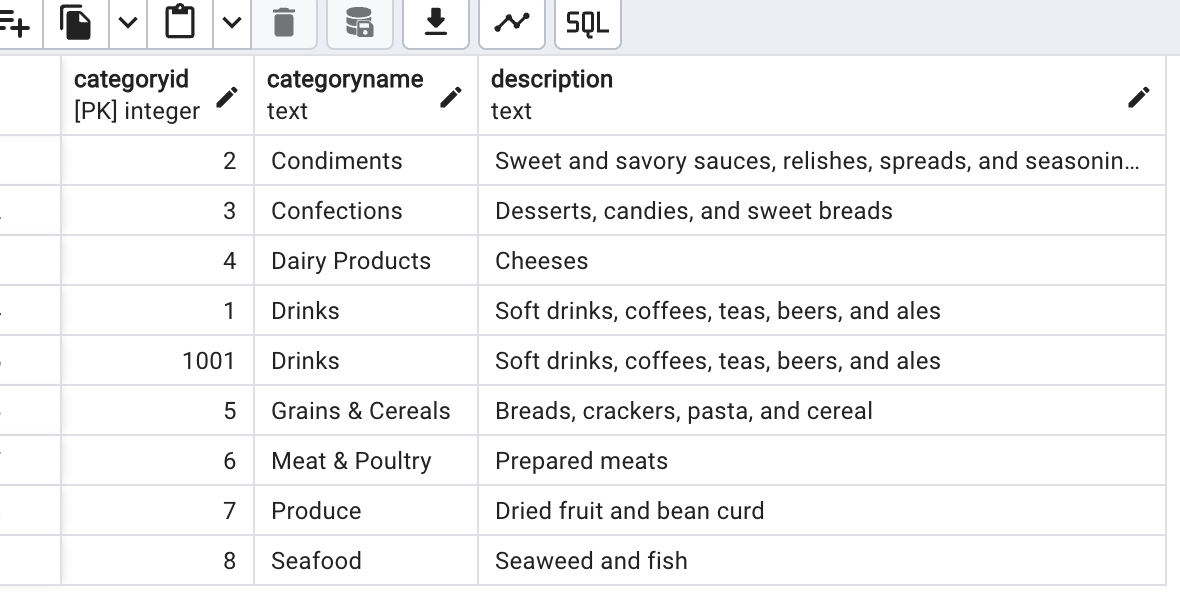
**1) Update the categoryName From “Beverages” to "Drinks" in the categories table.**

SELECT \* FROM categories ORDER BY categoryname;

UPDATE categories

SET categoryname = 'Drinks'

WHERE categoryname = 'Beverages';

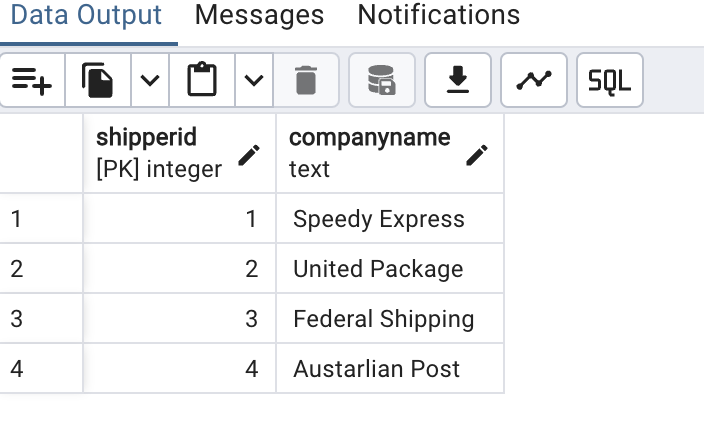


**2) Insert into shipper new record (give any values)**

INSERT INTO shippers

VALUES(4,'Austarlian Post');

SELECT \* FROM shippers;

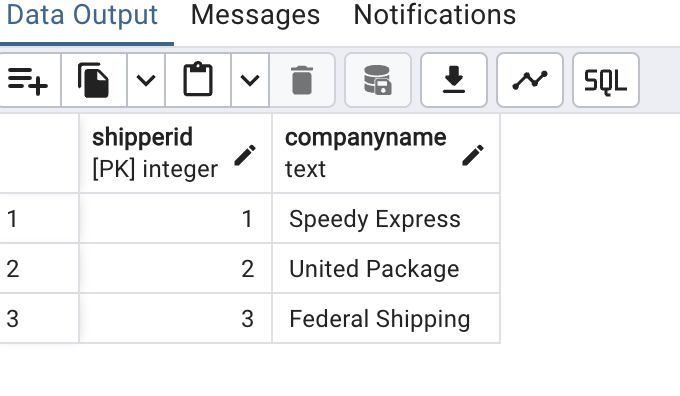


**Delete that new record from shippers table.**

DELETE FROM shippers

WHERE companyname='Austarlian Post';

SELECT \* FROM shippers;



**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.**

**(HINT: Alter the foreign key on products(categoryID)**

**to add ON UPDATE CASCADE, ON DELETE CASCADE)**

ALTER TABLE products

DROP CONSTRAINT products\_categoryid\_fkey;

ALTER TABLE products

ADD CONSTRAINT products\_categoryid\_fkey

FOREIGN KEY (categoryid)

REFERENCES categories(categoryid)

ON UPDATE CASCADE

ON DELETE CASCADE;

UPDATE categories

SET categoryid = 1001

WHERE categoryid = 1;

**Delete the categoryID= “3” from categories.**

**Verify that the corresponding records are deleted automatically from products.**

DELETE FROM categories

WHERE categoryid = 3 ;

SELECT \* FROM products WHERE categoryid = 3 ;

ALTER TABLE order\_details

DROP CONSTRAINT order\_details\_productid\_fkey;

ALTER TABLE order\_details

ADD CONSTRAINT fk\_orderdetails\_productid

FOREIGN KEY (productid)

REFERENCES products(productid)

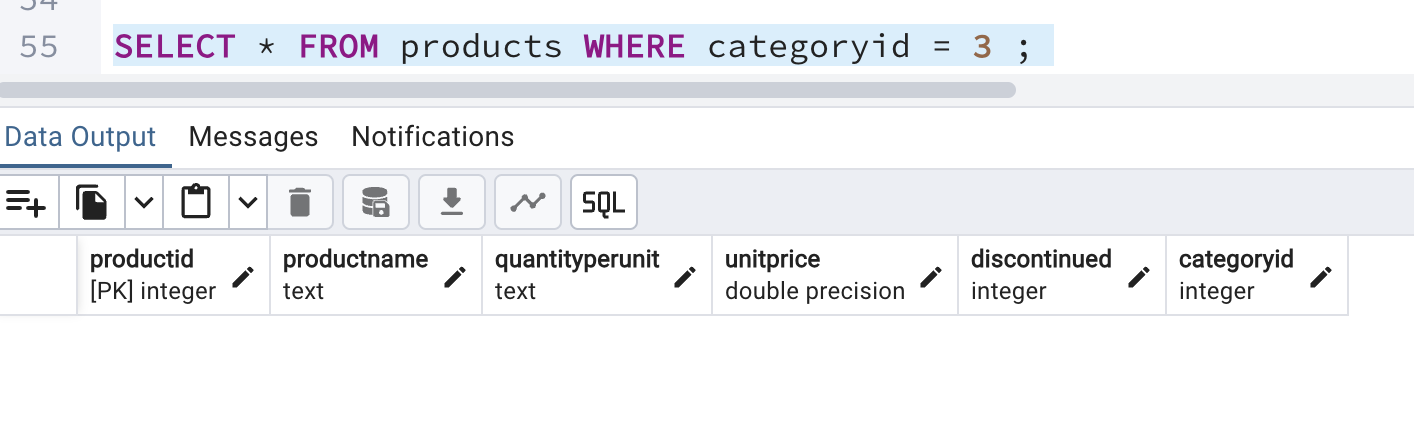
ON DELETE CASCADE;

ALTER TABLE order\_details

DROP CONSTRAINT fk\_orderdetails\_productid;

DELETE 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';

SELECT \* FROM orders WHERE customerid= 'VINET';

ALTER TABLE orders

DROP CONSTRAINT orders\_customerid\_fkey;

ALTER TABLE orders

ADD CONSTRAINT orders\_customerid\_fkey

FOREIGN KEY (customerid)

REFERENCES customers(customerid)

ON DELETE SET NULL;

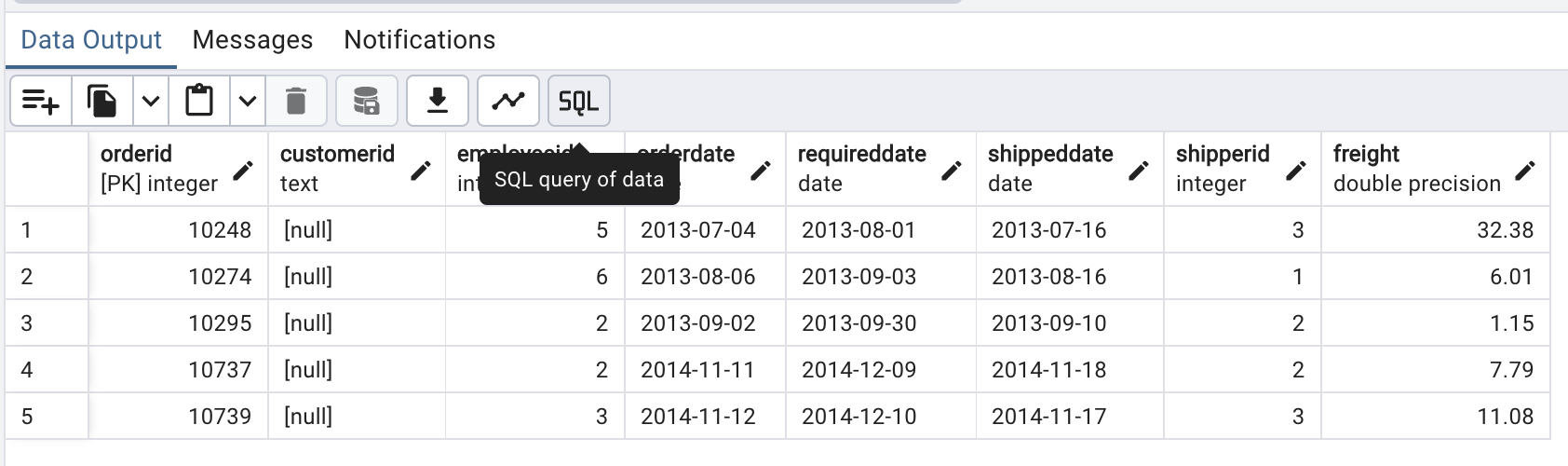
ALTER TABLE orders

ALTER COLUMN customerid DROP NOT NULL;

DELETE FROM customers

WHERE customerid = 'VINET';

SELECT \* FROM orders WHERE customerid= 'VINET' or customerid IS NULL;



**5)Insert the following data into 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 quantity per unit for product\_id = 100)**

ALTER TABLE products

RENAME COLUMN price\_in\_usd TO unitprice;

SELECT \* FROM products order by productid ;

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

VALUES (100,'Wheat bread','1 box',13,0,3),

(101,'Wheat bread','5 boxes',13,0,3)

SELECT \* FROM products WHERE productid = 100 ;

INSERT INTO categories VALUES (3,'Confections','Desserts, candies, and sweet breads')

SELECT \* FROM categories order by categoryid;

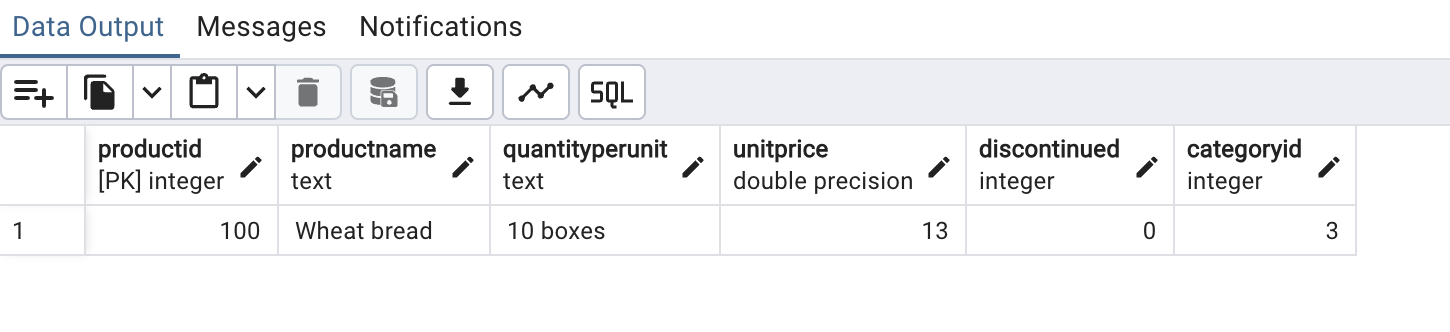
INSERT INTO products(productid, productname, quantityperunit, unitprice, discontinued, categoryid)

VALUES (100,'Wheat bread','10 boxes',13,0,3)

ON CONFLICT (productid)

DO UPDATE SET quantityperunit = EXCLUDED.quantityperunit;

SELECT \* FROM products WHERE productid = 100 ;



**6) Write a MERGE query:**

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

**Update the price and discontinued status for from below table ‘updated\_products’**

**only if there are matching products and updated\_products .discontinued =0**

**Insert any new products from updated\_products that don’t exist in products**

**only if updated\_products .discontinued =0.**

CREATE TEMPORARY TABLE updated\_products(

productID integer PRIMARY KEY,

productName text NOT NULL,

quantityPerUnit text NOT NULL,

unitPrice float NOT NULL,

discontinued integer NOT NULL,

categoryID integer NOT NULL);

SELECT \* FROM updated\_products;

INSERT INTO categories VALUES (1,'Beverages','Soft drinks, coffees, teas, beers, and ales')

INSERT INTO updated\_products

VALUES (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)

MERGE INTO products

USING updated\_products

ON products.productid = updated\_products.productid

WHEN MATCHED AND updated\_products .discontinued = 0 THEN

UPDATE SET unitPrice = COALESCE(updated\_products.unitprice, products.unitprice),

discontinued = updated\_products.discontinued

WHEN MATCHED AND updated\_products .discontinued = 1 THEN

DELETE

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)

RETURNING

merge\_action() as action,

products.productid,

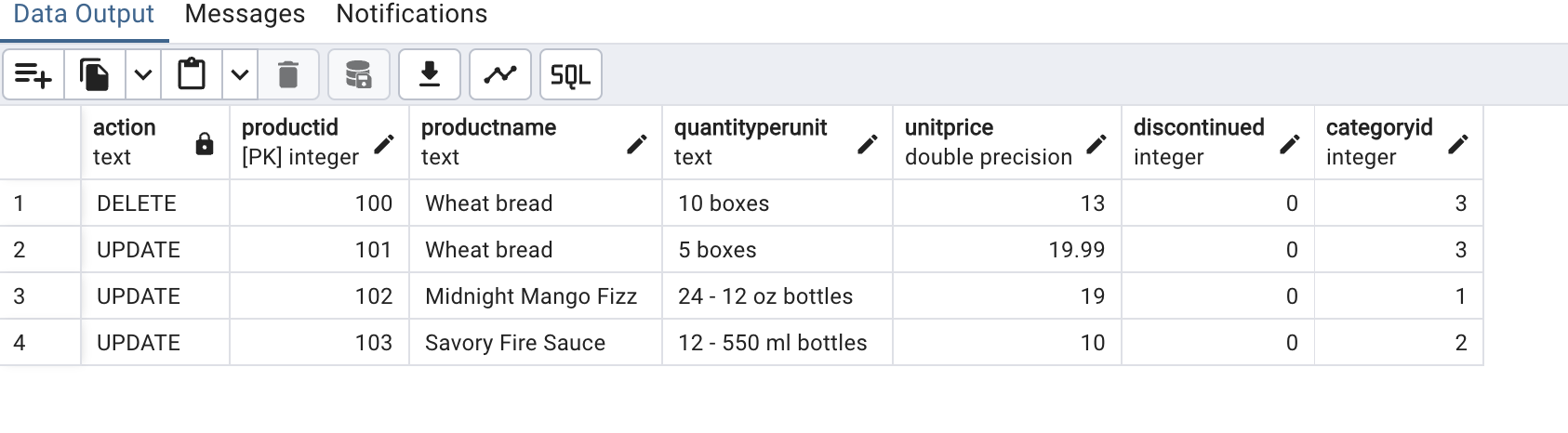
products.productName,

products.quantityPerUnit,

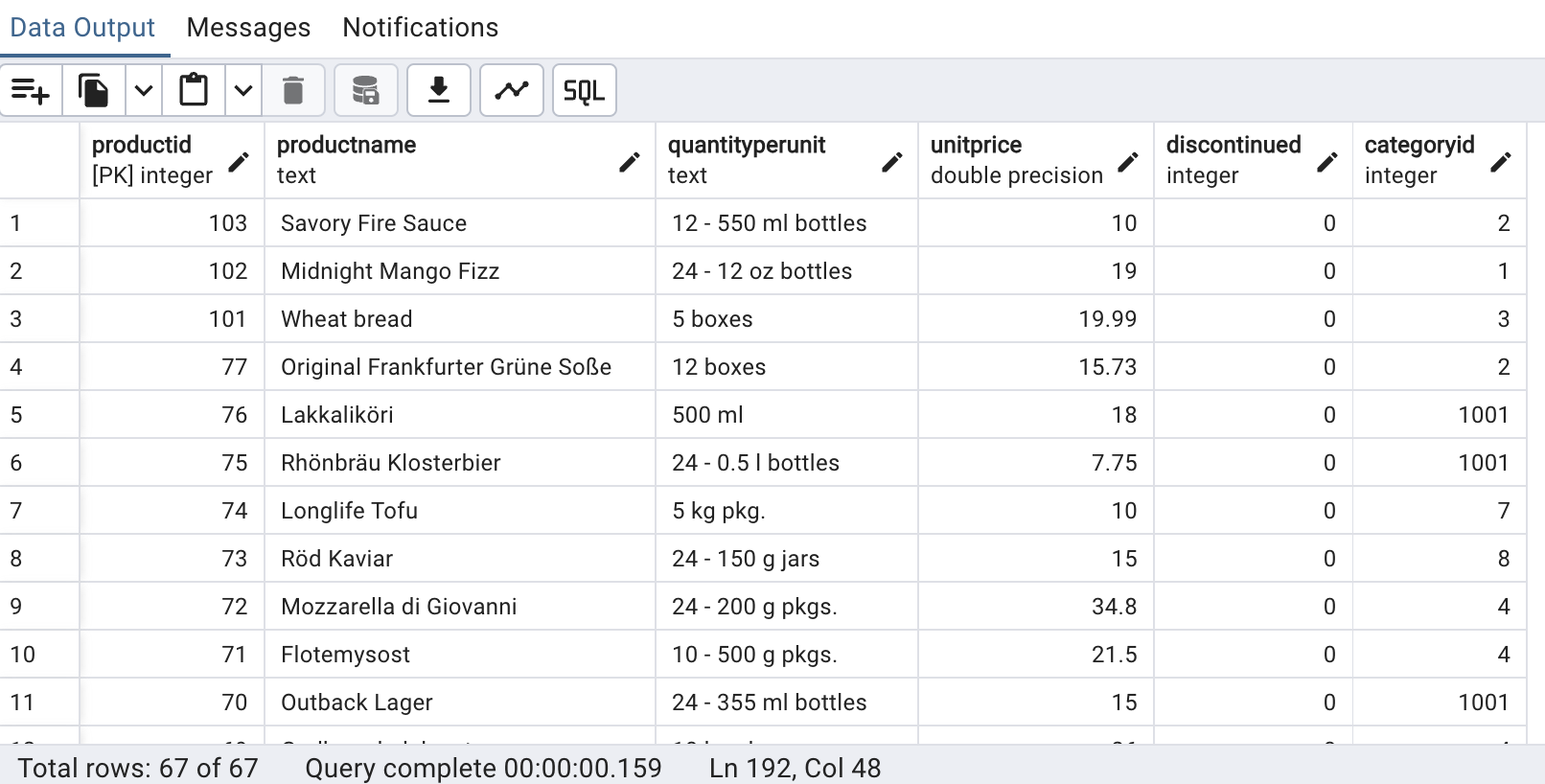
products.unitPrice,

products.discontinued,

products.categoryID;



Select \* from products order by productID desc;



**7) List all orders with employees' full names. (Inner join)**

SELECT employees.first\_name ||' '||employees.last\_name AS employee\_full\_name,

orders.order\_id as order\_number,

order\_date

FROM orders

INNER JOIN employees

ON orders.employee\_id = employees.employee\_id

ORDER BY employee\_full\_name;

