# SQL\_Assignment

Day3

[SQL\_Assignment 1](#_Toc196433870)

[1)      Update the categoryName From “Beverages” to "Drinks" in the categories table 1](#_Toc196433871)

[2)  Insert into shipper new record (give any values) Delete that new record from shippers table. 2](#_Toc196433872)

[3)     a. 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. 3](#_Toc196433873)

[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) 5](#_Toc196433874)

[5)      Insert the following data to Products using UPSERT: 6](#_Toc196433875)

[6)      Write a **MERGE query**: 8](#_Toc196433876)

[7)      List all orders with employee full names. (Inner join) 11](#_Toc196433877)

## 

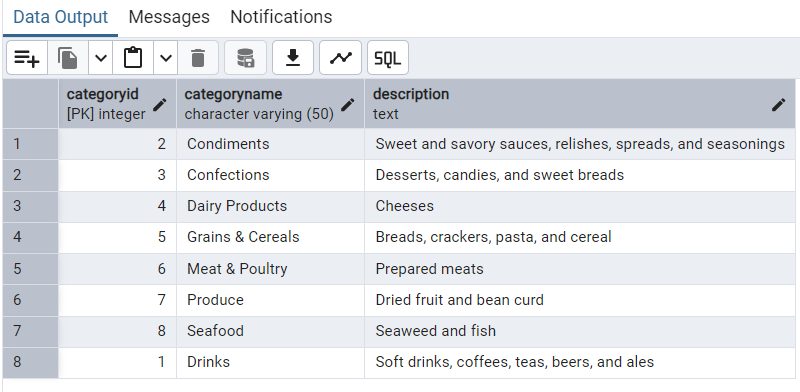
## 1)      Update the categoryName From “Beverages” to "Drinks" in the categories table

**Query:**

UPDATE categories SET categoryName = 'Drinks'

WHERE categoryName = 'Beverages';

**Screenshot:**

****

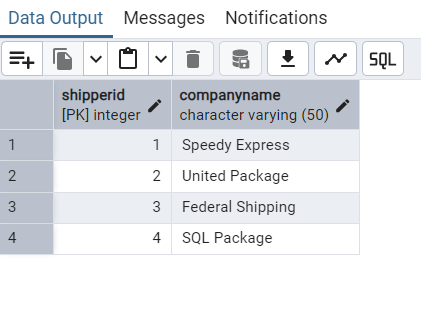
## 2)  Insert into shipper new record (give any values) Delete that new record from shippers table.

**Query:**

INSERT INTO shippers (shipperid, companyname)

VALUES (4, 'SQL Package');

**Screenshot:**

****

## 3)     a. 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.

**Query:**

ALTER TABLE products

DROP CONSTRAINT products\_categoryid\_fkey; --Drop key

ALTER TABLE products

ADD CONSTRAINT products\_categoryid\_fkey

FOREIGN KEY (categoryid)

REFERENCES categories(categoryid)

ON UPDATE CASCADE

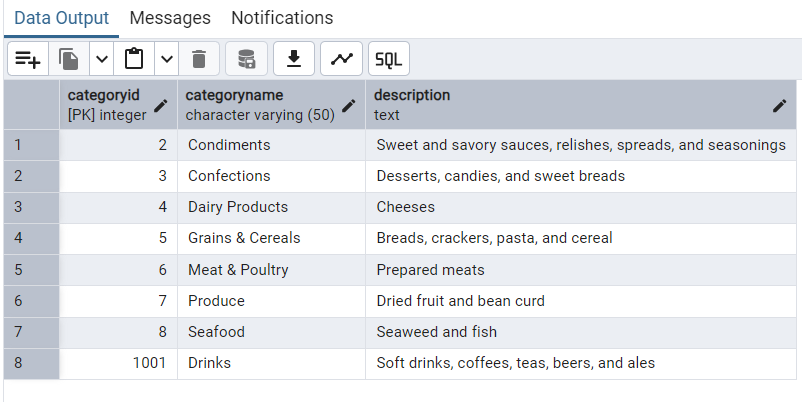
ON DELETE CASCADE; --Add key with cascade

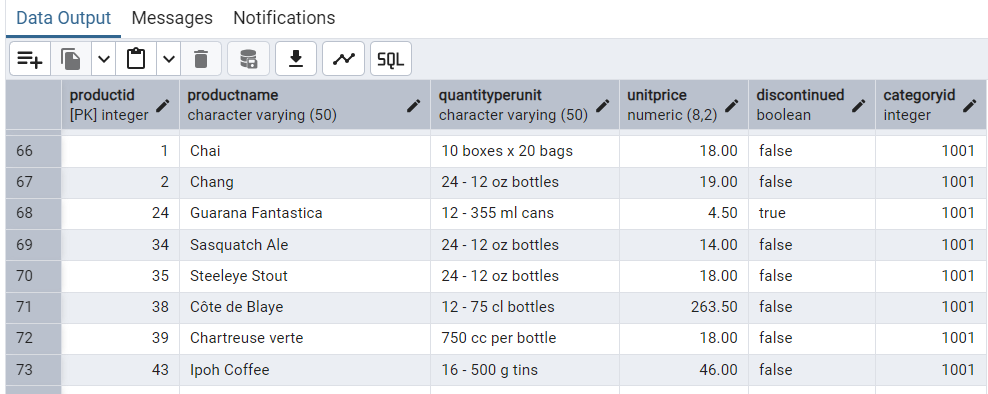
UPDATE categories

SET categoryid = 1001

WHERE categoryid = 1; --Update 1 to 1001

**Screenshot:**

****

****

 b. 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)

**Query:**

ALTER TABLE order\_details

DROP CONSTRAINT order\_details\_productid\_fkey; -- Delete key from Products

ALTER TABLE order\_details

ADD CONSTRAINT order\_details\_productid\_fkey

FOREIGN KEY (productid)

REFERENCES products(productid)

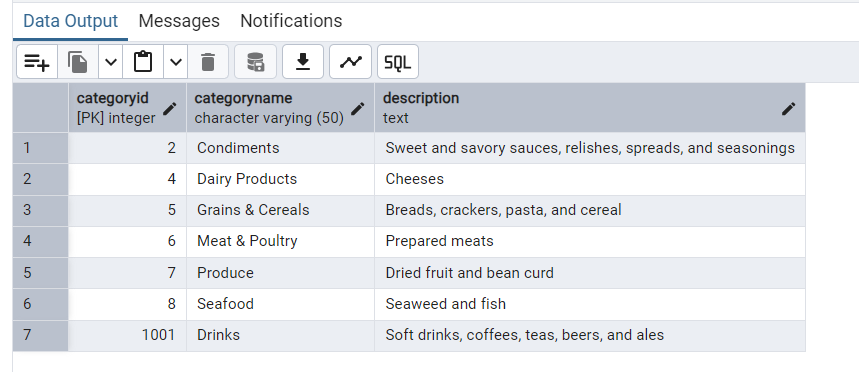
ON UPDATE CASCADE

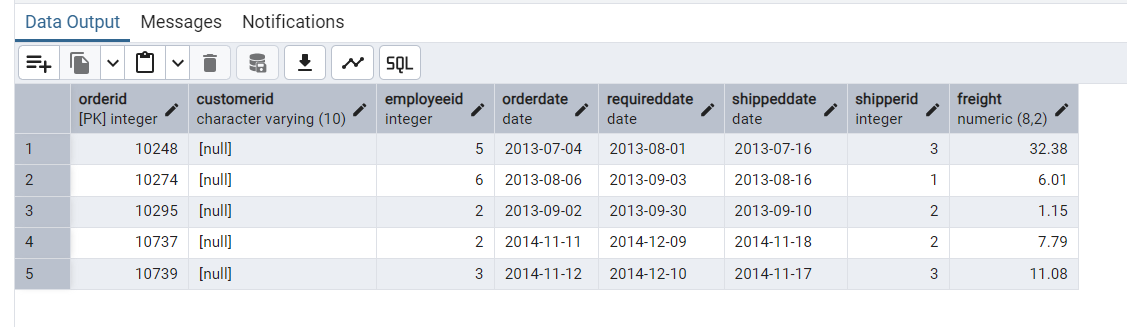
ON DELETE CASCADE; -- Add Key with cascade

DELETE FROM categories

WHERE categoryid = 3;

**Screenshot:**





## 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)

**Query:**

 ALTER TABLE orders

DROP CONSTRAINT orders\_customerid\_fkey; --Delete key from orders

ALTER TABLE orders

ADD CONSTRAINT orders\_customerid\_fkey

FOREIGN KEY (customerid)

REFERENCES customers(customerid)

ON UPDATE CASCADE

ON DELETE SET NULL; --Add Key with cascade

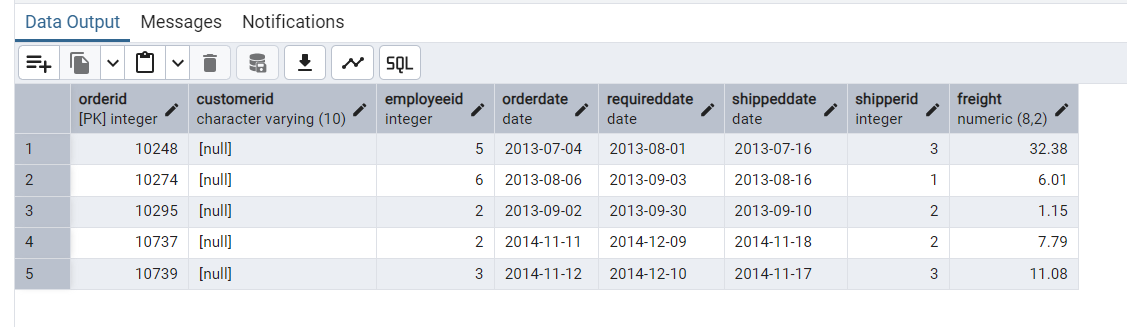
DELETE FROM customers

WHERE customerid = 'VINET';

SELECT \* FROM orders

WHERE customerid IS NULL;

**Screenshot:** 



## 5)      Insert the following data to Products using UPSERT:

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

**Query:**

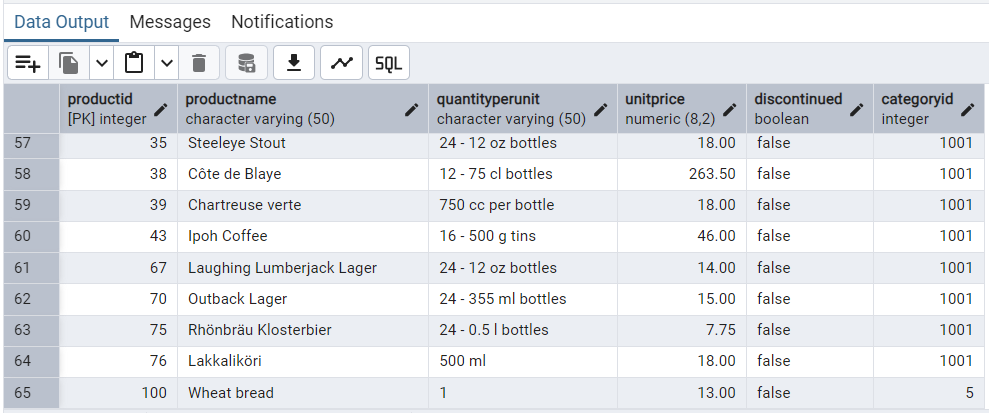
INSERT INTO products ( productid, productname, quantityperunit, unitprice, discontinued, categoryid )

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

ON CONFLICT (productid) DO UPDATE

SET quantityperunit = EXCLUDED.quantityperunit;

**Screenshot:**

****

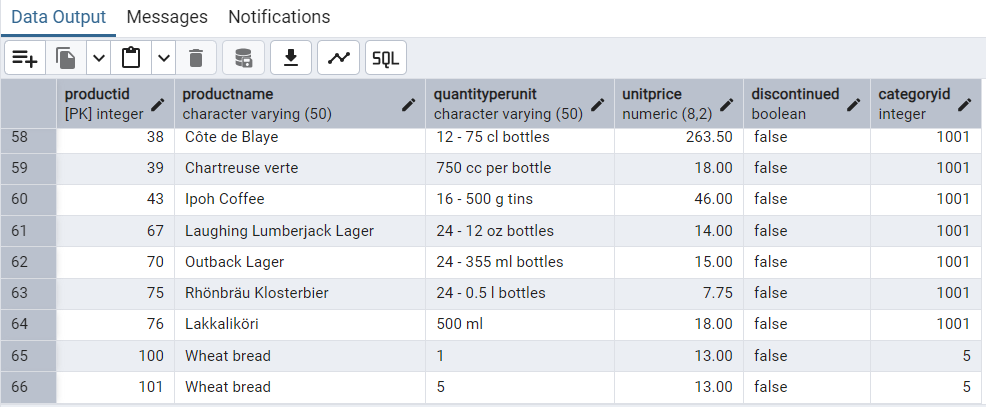
b. product\_id = 101, product\_name = White bread, quantityperunit=5 boxes,unitprice = 13, discontinued = 0, categoryID=3

**Query:**

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

VALUES (101, 'Wheat bread', 5, 13, false, 5)

**Screenshot:**

****

c. 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:**

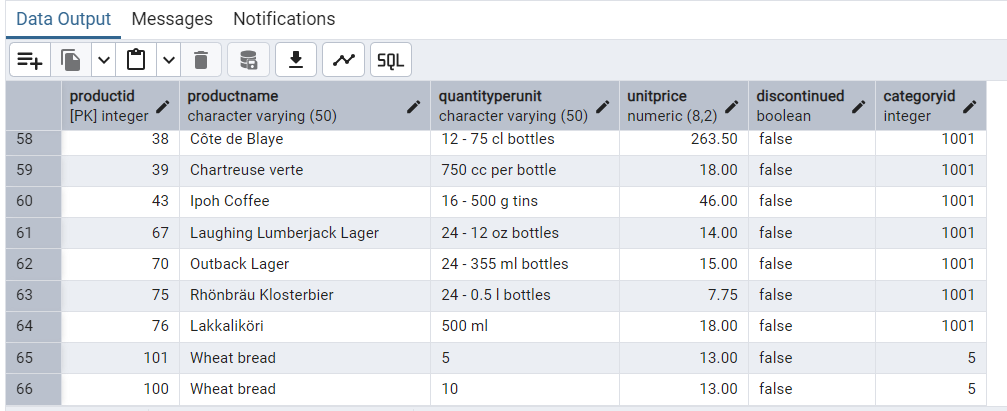
INSERT INTO products ( productid, productname, quantityperunit, unitprice, discontinued, categoryid )

VALUES (100, 'Wheat bread', 10, 13, false, 5)

ON CONFLICT (productid) DO UPDATE

SET quantityperunit = EXCLUDED.quantityperunit;

**Screenshot:**



## 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 |

**Query:**

CREATE TEMP TABLE updated\_products (

productid INT PRIMARY KEY,

productname TEXT,

quantityperunit TEXT,

unitprice NUMERIC(10, 2),

discontinued BOOLEAN,

categoryid INT

);

SELECT \* FROM updated\_products

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

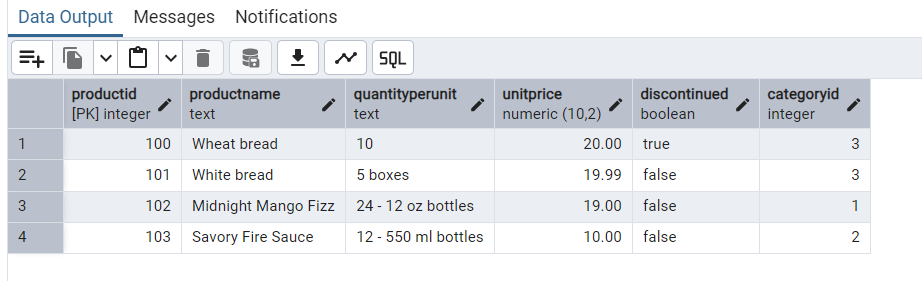
VALUES (100, 'Wheat bread', '10', 20, true, 3),

(101, 'White bread', '5 boxes', 19.99, false, 3),

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

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

**Screenshot:**



* 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.

**Query:**

MERGE INTO products p

USING updated\_products u

ON p.productid = u.productid

WHEN MATCHED AND u.discontinued = 'FALSE' THEN

UPDATE SET

unitprice = u.unitprice,

discontinued = u.discontinued

WHEN MATCHED AND u.discontinued = 'TRUE' THEN

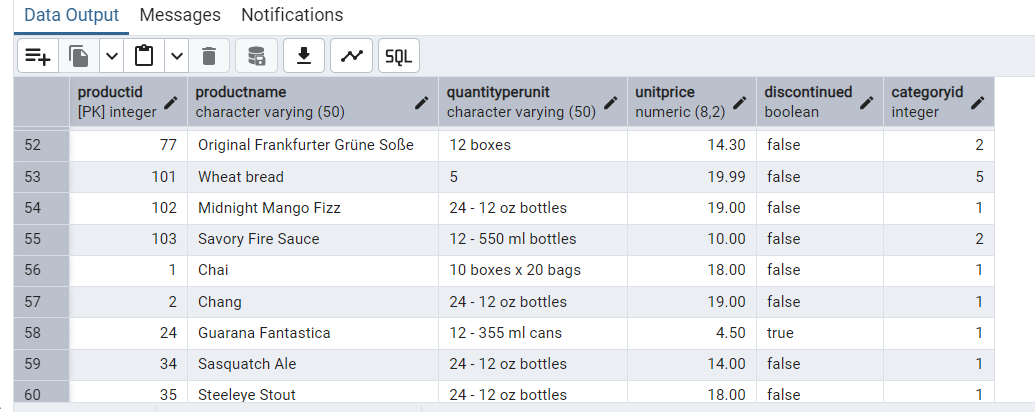
DELETE

WHEN NOT MATCHED AND u.discontinued = 'FALSE' THEN

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

VALUES (u.productid, u.productname, u.quantityperunit, u.unitprice, u.discontinued, u.categoryid);

**Screenshot:**



**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 employeefullname

FROM orders o

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

**Screenshot:**

