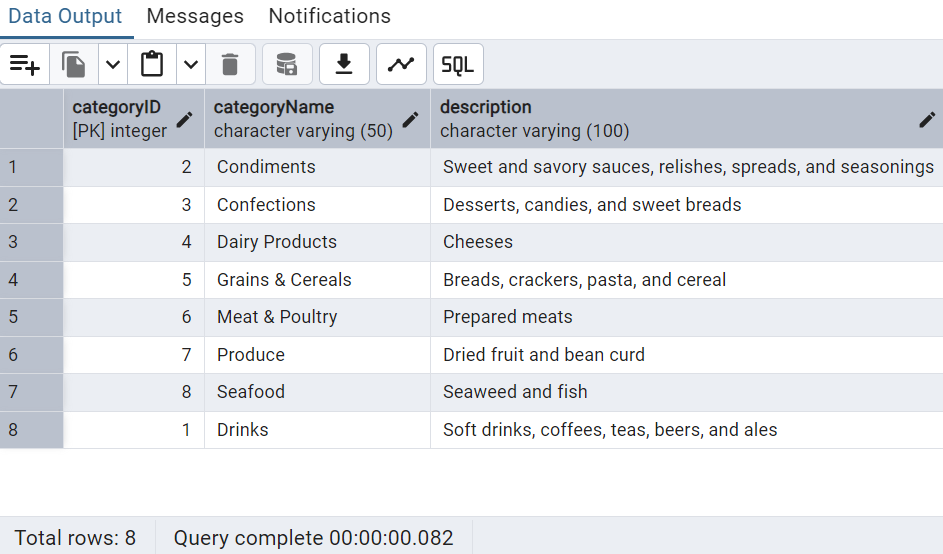
  Assignment 3

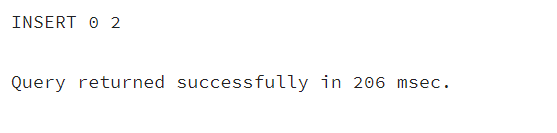
1.Update the categoryName From “Beverages” to "Drinks" in the categories table.

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 INTO shippers ("shipperID","companyName")

VALUES ('4', 'DHL'),('5', 'fedex');



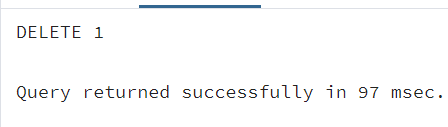
After inserting new record

select \* from shippers;



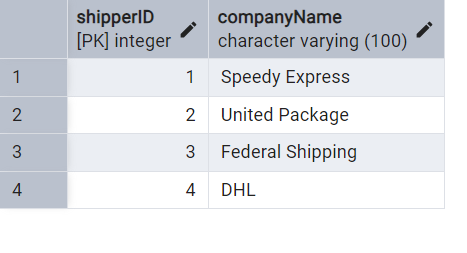
DELETE FROM shippers

WHERE "shipperID" = '5' AND "companyName" = 'fedex';



After deleting one new record

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.

 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)

ALTER TABLE products ADD CONSTRAINT "fk\_categoryID" FOREIGN KEY ("categoryID")

REFERENCES categories("categoryID")

ON UPDATE CASCADE

ON DELETE CASCADE;

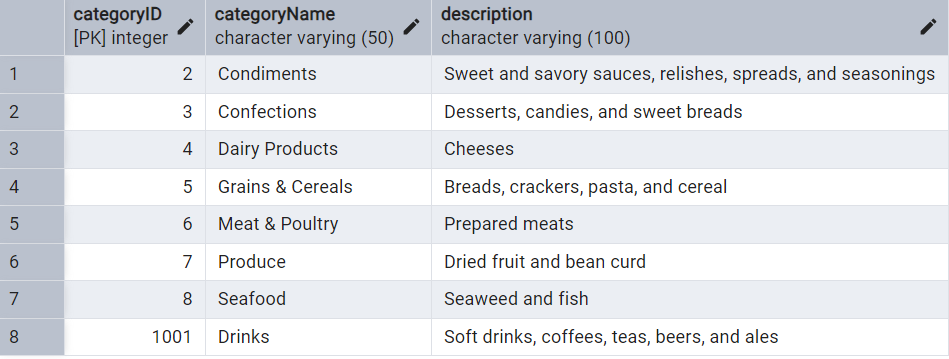
------Update categoryID=1 to categoryID=1001

UPDATE categories

SET "categoryID" = 1001

WHERE "categoryID"= 1;

select \* from categories;



select \* from products;



ALTER TABLE order\_details DROP CONSTRAINT "fk\_products";

ALTER TABLE order\_details

ADD CONSTRAINT "fk\_products"

FOREIGN KEY ("productID")

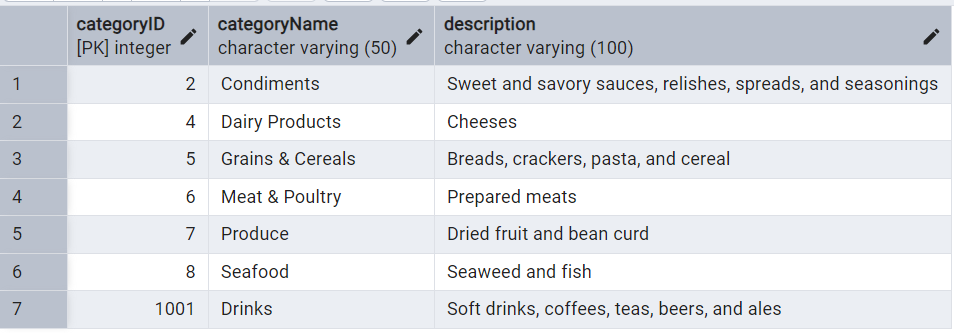
REFERENCES products("productID")

ON UPDATE CASCADE

ON DELETE CASCADE;

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)

ALTER TABLE orders

ALTER COLUMN "customerID" DROP NOT NULL;

ALTER TABLE orders DROP CONSTRAINT "fk\_customerid";

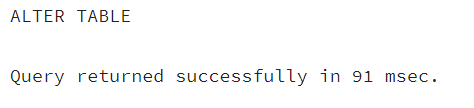
ALTER TABLE orders

ADD CONSTRAINT "fk\_customerid"

FOREIGN KEY ("customerID")

REFERENCES customers("customerID")

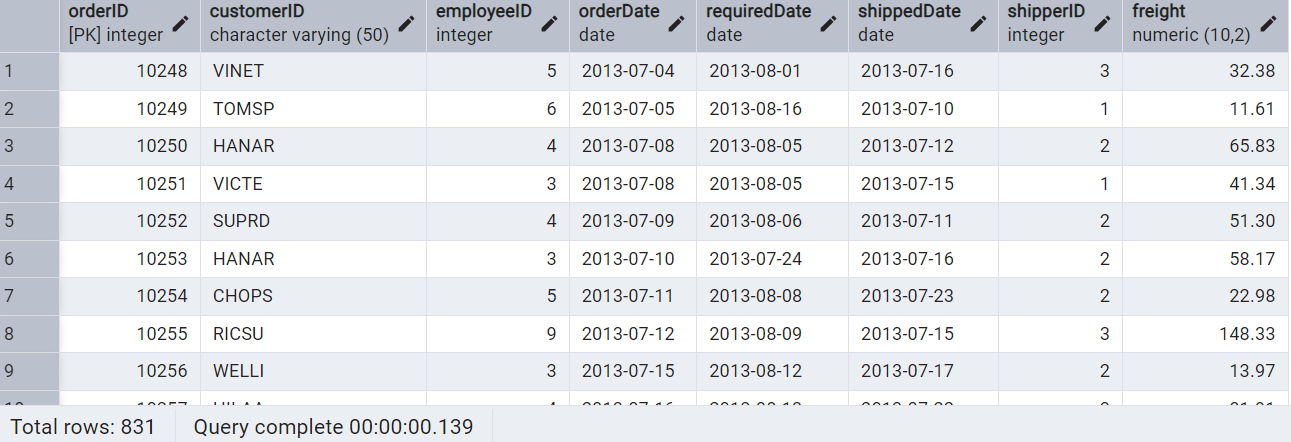
ON DELETE SET NULL;



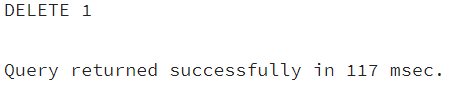
DELETE FROM customers WHERE "customerID" = 'VINET';

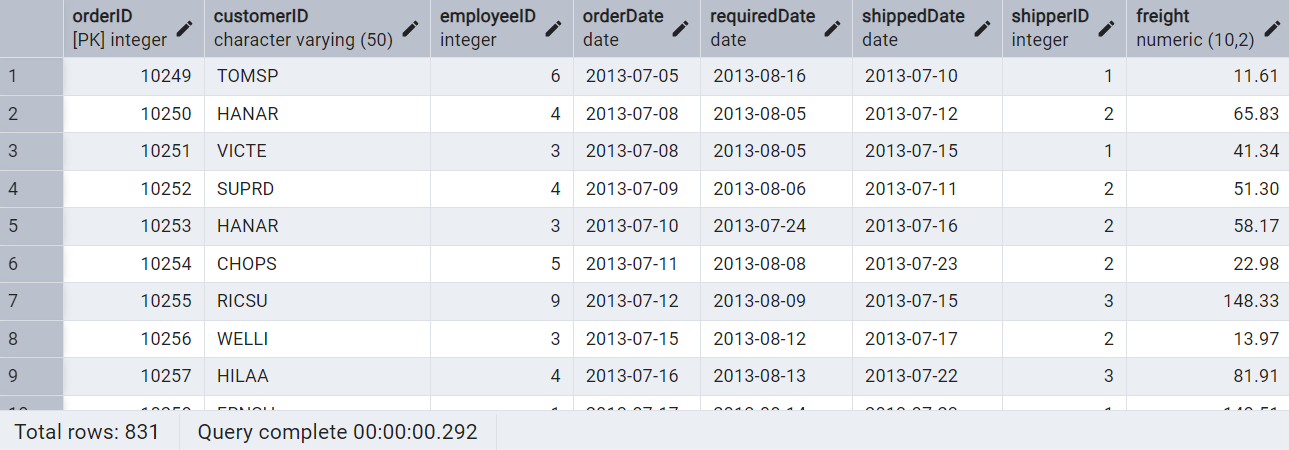
select \* from orders;

Before Deleting:



After Deleting





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

ON CONFLICT ("productID")

DO UPDATE SET

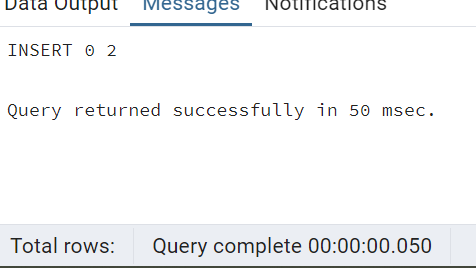
"quantityPerUnit" = EXCLUDED."quantityPerUnit",

"productName" = EXCLUDED."productName",

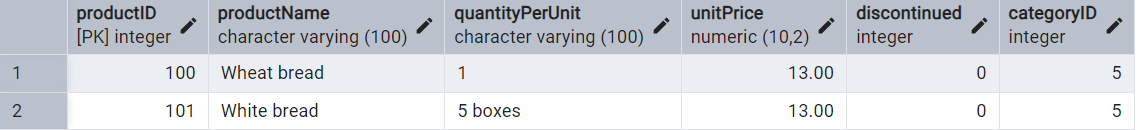
"unitPrice" = EXCLUDED."unitPrice",

"discontinued" = EXCLUDED."discontinued",

"categoryID"= EXCLUDED."categoryID";



select \* from products where "productID" IN ('100','101');



After Upsert

INSERT INTO products ("productID","productName","quantityPerUnit","unitPrice","discontinued","categoryID")

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

ON CONFLICT ("productID")

DO UPDATE SET

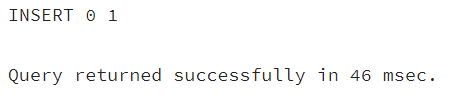
"quantityPerUnit" = EXCLUDED."quantityPerUnit",

"productName" = EXCLUDED."productName",

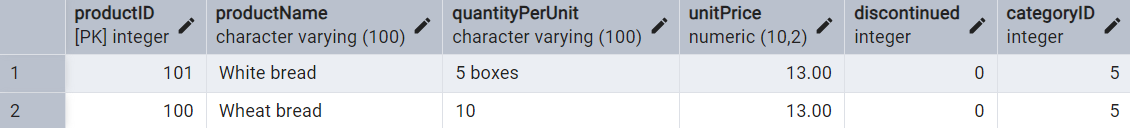
"unitPrice" = EXCLUDED."unitPrice",

"discontinued" = EXCLUDED."discontinued",

"categoryID"= EXCLUDED."categoryID";



select \* from products where "productID" IN ('100','101');



6)      Write a **MERGE query**:

--Create a temporary table and insert data

CREATE TEMP TABLE updatedProducts (

"productID" SERIAL PRIMARY KEY,

"productName" VARCHAR(100) NOT NULL,

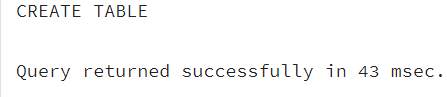
"quantityPerUnit" VARCHAR(100) NOT NULL,

"unitPrice" NUMERIC(10,2) NOT NULL,

"discontinued" INT NOT NULL,

"categoryID" INT NOT NULL

);

INSERT INTO updatedProducts ("productID", "productName", "quantityPerUnit", "unitPrice", "discontinued", "categoryID")

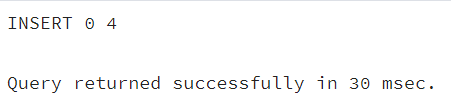
VALUES

(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" p

USING "updated\_products" u

ON p."productID" = u."productID"

-- If matching and discontinued = 0 → UPDATE

WHEN MATCHED AND u."discontinued" = 0 THEN

UPDATE SET

"unitPrice" = u."unitPrice",

"discontinued" =u."discontinued"

-- If matching and discontinued = 1 → DELETE

WHEN MATCHED AND u."discontinued" = 1 THEN

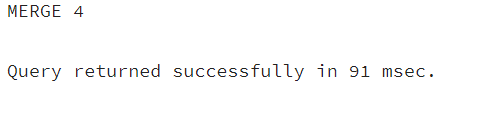
DELETE

-- If not matched and discontinued = 0 → INSERT

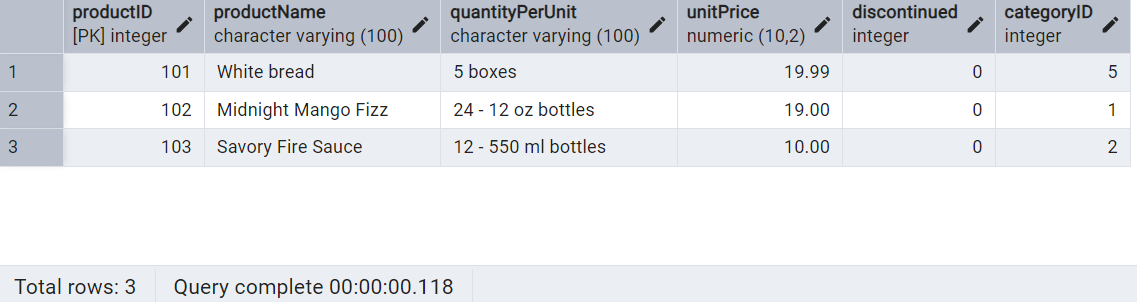
WHEN NOT MATCHED AND u."discontinued" = 0 THEN

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

VALUES (u."productID", u."productName", u."quantityPerUnit", u."unitPrice", u."discontinued", u."categoryID");



SELECT \* FROM products WHERE "productID" IN (100,101,102,103);



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

SELECT o.order\_id,

o.order\_date,

e.first\_name || ' ' || e.last\_name AS employeeFullName

FROM orders o

INNER JOIN employees e

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

