**Day 3 – Assignment - Banupriya**

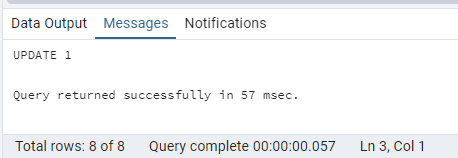
**USE Northwind from Kaggle:**

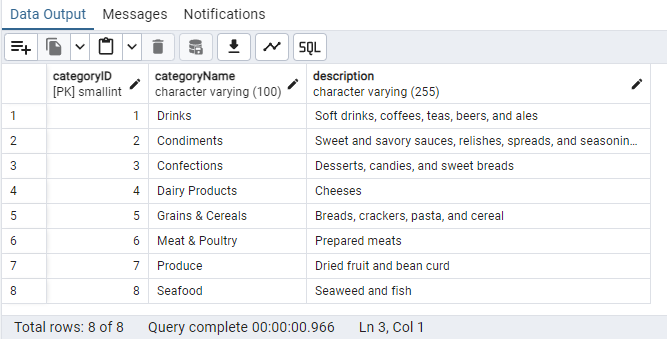
1)      Update the categoryName From “Beverages” to "Drinks" in the categories table.

**Script:**

*Update categories set "categoryName" = 'Drinks' where "categoryID" = 1;*

**Screenshot:**

****

****

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

**Script:**

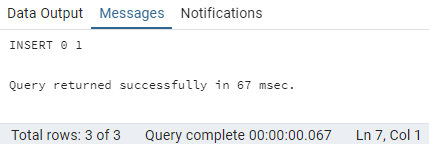
**Insert Script:**

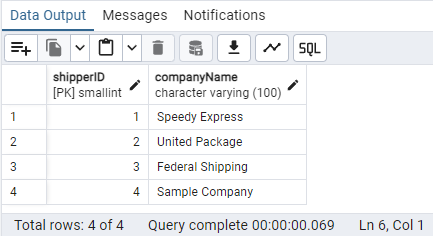
insert into shippers ("shipperID","companyName") VALUES (4,'Sample Company');

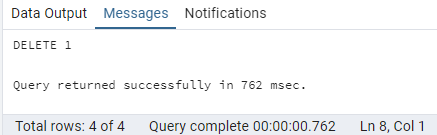
**Delete Script:**

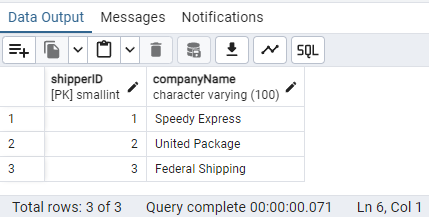
Delete from shippers where "shipperID" = 4;

**Screenshot:**









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) )

**Dropping the Constaints and recreating them:**

**Script:**

ALTER TABLE products DROP CONSTRAINT product\_categoryid\_fkey;

ALTER TABLE products ADD CONSTRAINT product\_categoryid\_fkey FOREIGN KEY ("categoryID")

REFERENCES public.categories ("categoryID")

ON UPDATE CASCADE

ON DELETE CASCADE;

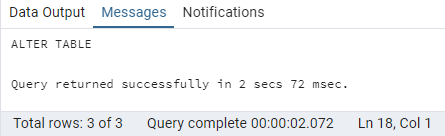
ALTER TABLE orderdetails DROP CONSTRAINT orderdetails\_productid\_fkey;

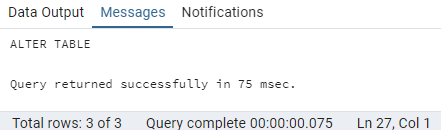
ALTER TABLE orderdetails ADD CONSTRAINT orderdetails\_productid\_fkey FOREIGN KEY ("productID")

REFERENCES public.products ("productID")

ON UPDATE CASCADE

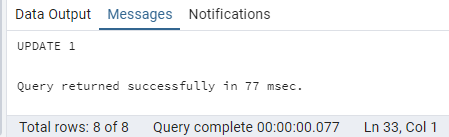
ON DELETE CASCADE;





Running the update statement:

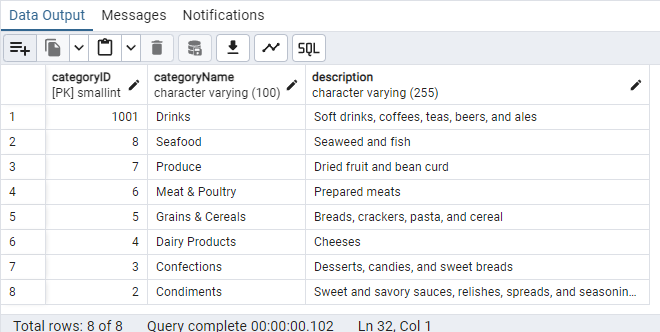
**UPDATE categories set "categoryID" = 1001 where "categoryID" = 1;**

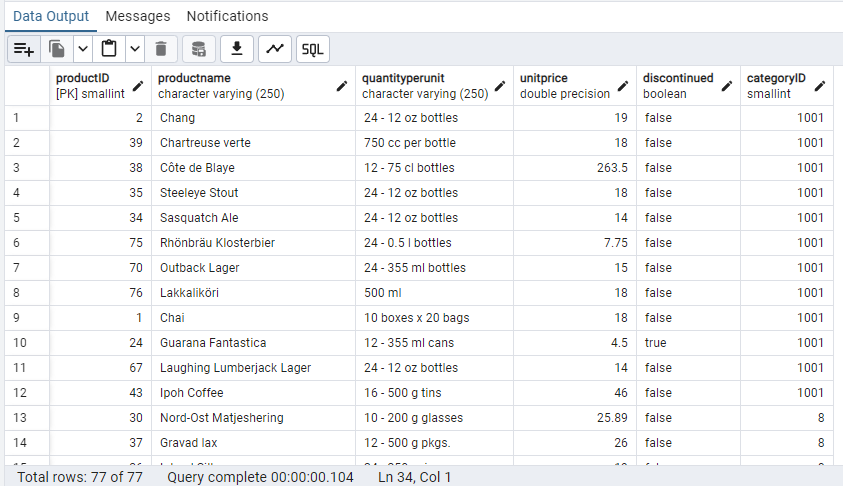


**In both the categories table and products the respective update got cascaded:**

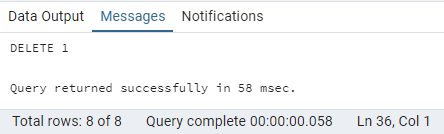
**select \* from categories order by 1 desc;**

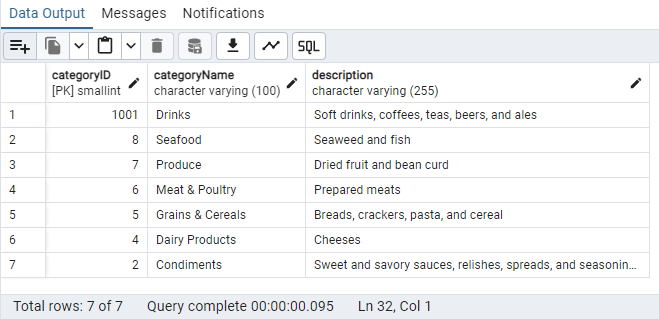
**select \* from products where "categoryID" = 3 order by "categoryID" desc**





***DELETE from categories where "categoryID" = 3;***





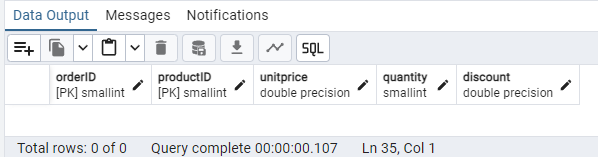
Finding in the products table to see if the categoryid = 3 is existing after it got deleted in the categories table.

***select \* from products where "categoryID" = 3;***



Finding in the orderdetails table to see if the products are existing after they got deleted in the products table.

***select \* from orderdetails where "productID" in (16,19,20,21,25,26,27,47,48,49,50,62,68);***



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)

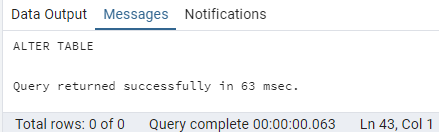
**Script:** Dropping and recreating the Constraints:

**ALTER TABLE orders DROP CONSTRAINT order\_customerid\_fkey;**

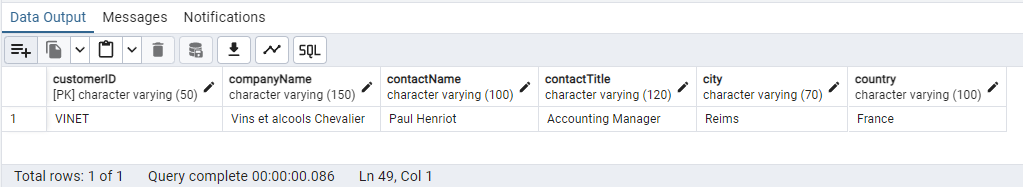
**ALTER TABLE orders ADD CONSTRAINT order\_customerid\_fkey FOREIGN KEY ("customerID")**

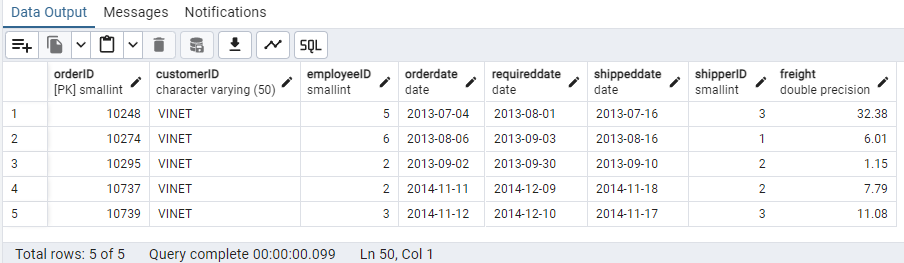
**REFERENCES public.customers ("customerID")**

**ON DELETE SET NULL;**



**Before Deleting:**

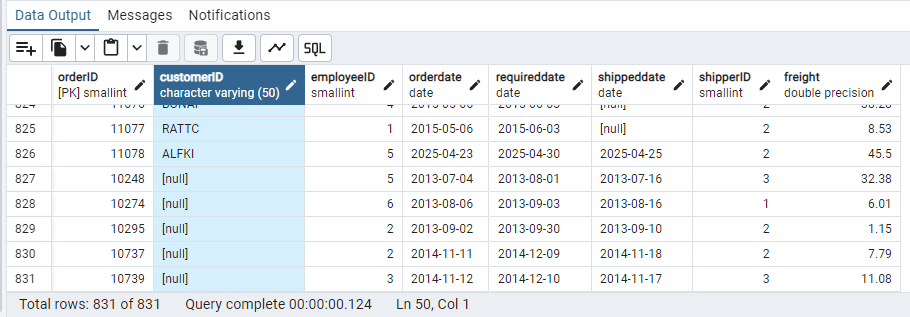
Customers Table: 

Orders Table: 

**After Deleting:**

**DELETE FROM customers where "customerID" = 'VINET';**

Customers Table: 

Orders Table: 

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)

**Script: Regular Inserts are happening**

INSERT into products("productID","productname",quantityperunit,unitprice,discontinued,"categoryID")

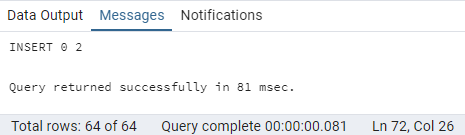
VALUES (100,'Wheat bread',1,13,false,5),

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

ON CONFLICT ("productID")

DO UPDATE

SET quantityperunit = 10;



**Script: Upsert is happening based on the conflict.**

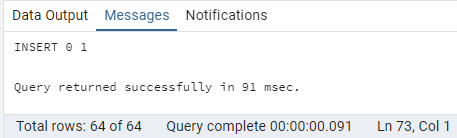
INSERT into products("productID","productname",quantityperunit,unitprice,discontinued,"categoryID")

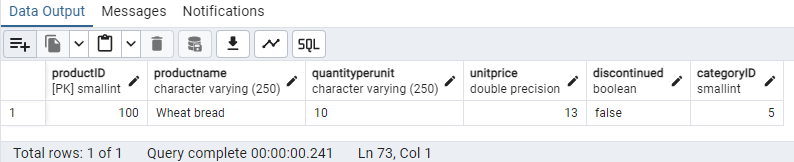
VALUES (100,'Wheat bread',10,13,false,5)

ON CONFLICT ("productID")

DO UPDATE

SET quantityperunit = EXCLUDED.quantityperunit;





6)      Write a **MERGE query**:

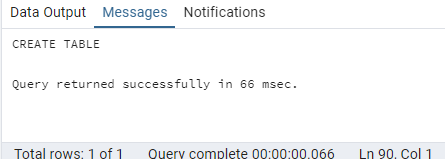
Create **temp table with name:**  ‘updated\_products’ and insert values as below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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.

**Script:**

create temp table updated\_products(productid int,productname varchar(100),quantityperunit varchar(100),unitprice real,discontinued smallint,categoryid smallint);

****

insert into updated\_products

values (100,'wheat bread',10,20,1,5),

(101,'wheat bread','5boxes',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);

**Script:**

**Creating the temp table updated\_products and inserting the values:**

create temp table updated\_products(productid int,productname varchar(100),quantityperunit varchar(100),unitprice real,discontinued boolean,categoryid smallint);

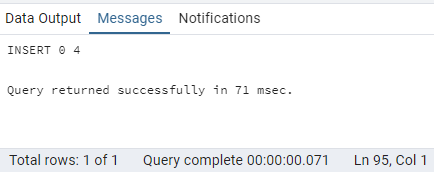
insert into updated\_products

values (100,'wheat bread',10,20,'true',5),

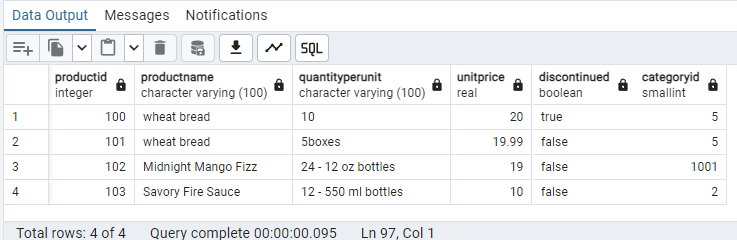
(101,'wheat bread','5boxes',19.99,'false',5),

(102,'Midnight Mango Fizz','24 - 12 oz bottles',19,'false',1001),

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

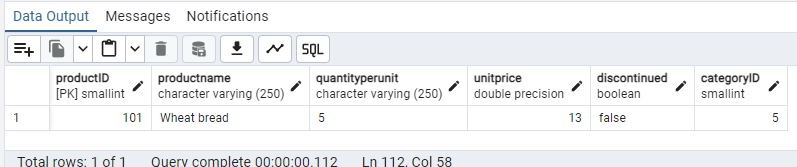
****

**select \* from updated\_products**

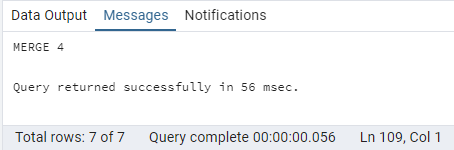
****

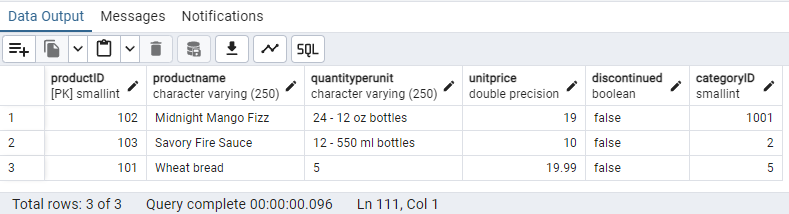
**Before running the Merge Query in products table:**

select \* from products where "productID" in (101,102,103)

****

**After running the Merge Query in products table:**





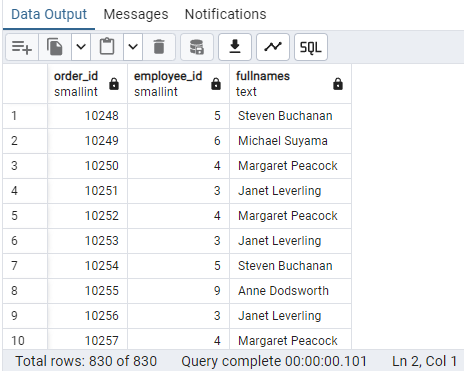
**USE NEW Northwind DB:**

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

**Script:**

select o.order\_id,e.employee\_id,e.first\_name || ' ' || e.last\_name as FullNames from orders o,employees e where e.employee\_id = o.employee\_id

**Screenshot:**

****