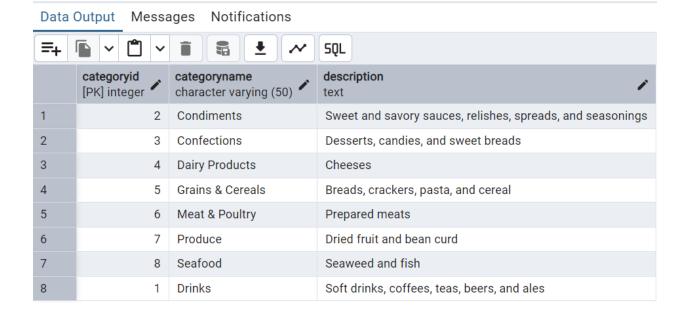
Day 3

USE Northwind from Kaggle:

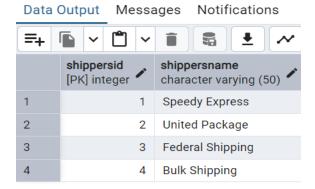
1) Update the categoryName From "Beverages" to "Drinks" in the categories table.

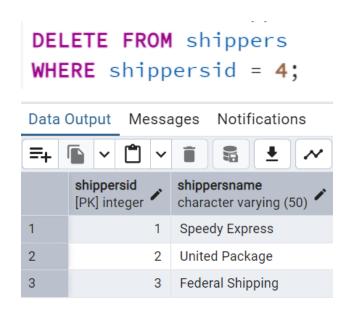
```
UPDATE categories
SET categoryname = 'Drinks'
WHERE categoryname = 'Beverages';
SELECT * FROM categories;
```



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

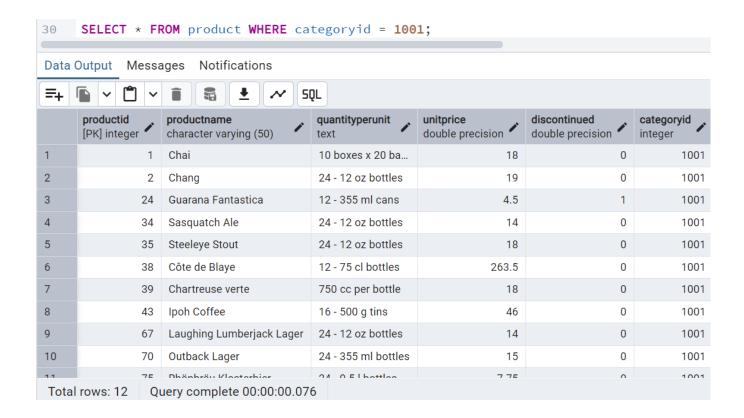
```
INSERT INTO shippers (shippersid, shippersname)
VALUES (4, 'Bulk Shipping');
SELECT * FROM shippers;
```





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)

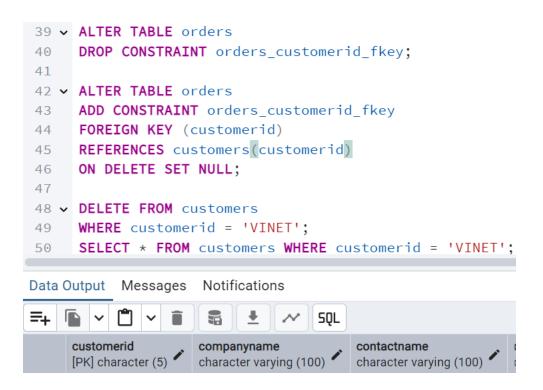
```
17 - ALTER TABLE product
18
     DROP CONSTRAINT IF EXISTS product_categoryid_fkey;
19
20 V ALTER TABLE product
     ADD CONSTRAINT product_categoryid_fkey
21
     FOREIGN KEY (categoryid)
22
23
     REFERENCES categories(categoryid)
24
     ON UPDATE CASCADE;
25
26 UPDATE categories
27
     SET categoryid = 1001
28
     WHERE categoryid = 1;
29
     SELECT * FROM categories WHERE categoryid = 1001;
Data Output
            Messages Notifications
=+
                                      5QL
                  categoryname
                                      description
     categoryid
     [PK] integer '
                  character varying (50)
                                      text
                                      Soft drinks, coffees, teas, beers, and ales
            1001
                  Drinks
```



Delete the categoryID= "3" from categories. Verify that the corresponding records are deleted automatically from products.

```
ALTER TABLE order_details
DROP CONSTRAINT IF EXISTS order_details_productid_fkey;
ALTER TABLE order details
ADD CONSTRAINT order_details_productid_fkey
FOREIGN KEY (productid)
REFERENCES product(productid)
ON UPDATE CASCADE
ON DELETE CASCADE;
44 DELETE FROM categories
45
      WHERE categoryid = 3;
46
      SELECT * FROM product WHERE categoryid = 3;
Data Output
            Messages Notifications
                                      SQL
=+
                                                                      discontinued
      productid
                  productname
                                      quantityperunit
                                                      unitprice
                                                                                      categoryid
                                                      double precision
     [PK] integer
                  character varying (50)
                                                                      double precision *
                                                                                      integer
```

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)



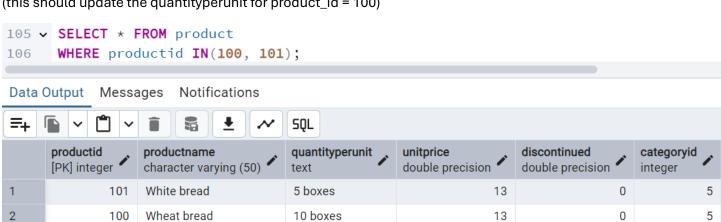
5) Insert the following data to Products using UPSERT:

```
product_id = 100, product_name = Wheat bread, quantityperunit=1,unitprice = 13, discontinued = 0, categoryID=5
INSERT INTO product (productid, productname, quantityperunit, unitprice, discontinued, categoryid)
VALUES (100, 'Wheat bread', 1, 13, 0, 5)
ON CONFLICT (productid)
DO UPDATE
SET productname = '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
INSERT INTO product (productid, productname, quantityperunit, unitprice, discontinued, categoryid)
VALUES (101, 'White bread', '5 boxes', 13, 0, 5)
ON CONFLICT (productid)
DO UPDATE
SET productname = '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

```
INSERT INTO product (productid, productname, quantityperunit, unitprice, discontinued, categoryid)
VALUES (100, 'Wheat bread', '10 boxes', 13, 0, 5)
ON CONFLICT (productid)
DO UPDATE
SET productname = 'Wheat bread',
    quantityperunit = '10 boxes',
    unitprice = 13,
    discontinued = 0,
    categoryid = 5;
```

(this should update the quantityperunit for product_id = 100)



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

- 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.

```
MERGE INTO product p
USING (
        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)
AS updated_products (productID ,productName, quantityPerUnit, unitPrice, discontinued, categoryID)
ON p.productid = updated_product.productID
WHEN MATCHED AND updated_products = 1 THEN
DELETE
WHEN MATCHED AND updated_products = 0 THEN
UPDATE SET
unitprice = updated_products.unitprice,
discontinued = updated_products.discontinued
WHEN NOT MATCHED updated_products.discontinued = 0 THEN
INSERT productid, productname, quanitytperunit, unitprice, discontinued, categoryid
VALUES (updated_products.productid,updated_products.productname, updated_products.quantityperunit, updated_pro
```

USE NEW Northwind DB:

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

