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

```
UPDATE categories
SET categoryname = 'Drinks'
WHERE categoryname = 'Beverages';

Data Output Messages Notifications

UPDATE 1

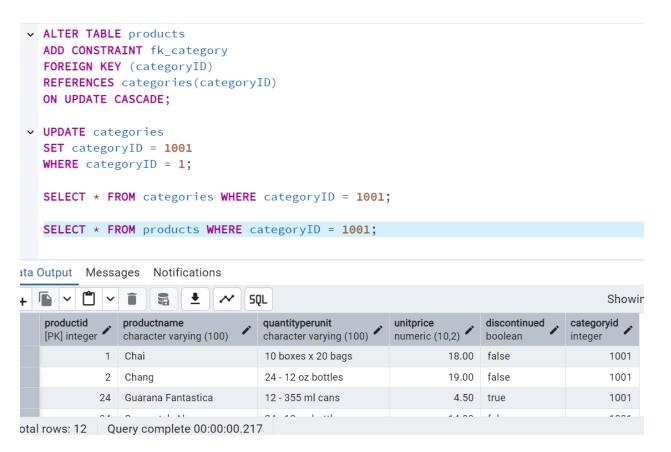
Query returned successfully in 145 msec.
```

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

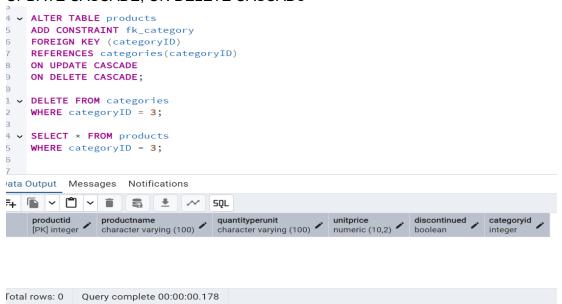
```
5 ▼ INSERT INTO shippers (ShipperID, companyName)
  VALUES (4 , 'UnitedPostal');
7 ▼ DELETE FROM shippers
    WHERE ShipperID = 'Speedy Express' AND companyName = 'UnitedPostal';
9
Data Output Messages Notifications
INSERT 0 1
Query returned successfully in 140 msec.
values (4 , 'Uniteurostat'),
7 ▼ DELETE FROM shippers
     WHERE ShipperID = 4 AND companyName = 'UnitedPostal';
9
                        Notifications
Data Output Messages
DELETE 1
Query returned successfully in 118 msec.
```

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.

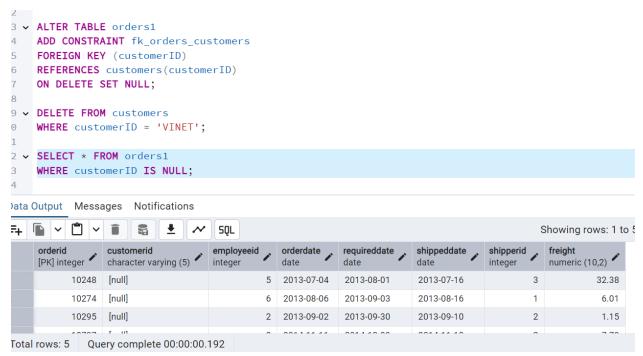
```
13 ▼ ALTER TABLE products
      ADD CONSTRAINT fk_category
14
      FOREIGN KEY (categoryID)
15
      REFERENCES categories(categoryID)
16
      ON UPDATE CASCADE;
17
18
19 ∨ UPDATE categories
      SET categoryID = 1001
20
      WHERE categoryID = 1;
21
22
23
      SELECT * FROM categories WHERE categoryID = 1001;
24
      SELECT * FROM products WHERE categoryID = 1001;
25
26
Data Output Messages Notifications
                                       SQL
=+
      categoryid
                   categoryname
                                       description
      [PK] integer
                   character varying (60)
                                       character varying (260)
                   Drinks
                                       Soft drinks, coffees, teas, beers, and ales
1
             1001
```



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



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:

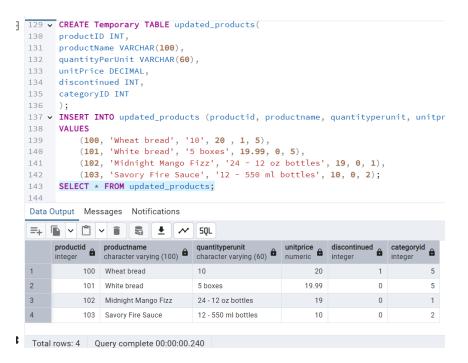
product_id = 100, product_name = Wheat bread, quantityperunit=1,unitprice = 13, discontinued = 0, categoryID=5

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

```
91 v INSERT INTO products (productid, productname, quantityperunit, unitprice, discontinued, categoryID)
92 VALUES
         (100, 'Wheat bread', '1', 13, FALSE, 5)
93
94 ON CONFLICT (productid)
95 DO UPDATE
96 SET
       productname = EXCLUDED.productname,
97
        quantityperunit = EXCLUDED.quantityperunit,
98
99
         unitprice = EXCLUDED.unitprice,
100
         discontinued = EXCLUDED.discontinued,
101
        categoryID = EXCLUDED.categoryID;
102
103 -- Second insert (productid = 101)
104 • INSERT INTO products (productid, productname, quantityperunit, unitprice, discontinued, categoryID)
105 VALUES
106
         (101, 'White bread', '5 boxes', 13, FALSE, 5)
     ON CONFLICT (productid)
    DO UPDATE
108
    SET
109
110
         productname = EXCLUDED.productname,
Data Output Messages Notifications
INSERT 0 1
Query returned successfully in 107 msec.
```

6) Write a MERGE query: Create temp table with name: 'updated products' and insert



 Update the price and discontinued status for from below table 'updated_products' only if there are matching products and updated products discontinued =0

```
7 ∨ UPDATE products
8
    SET
9
        unitprice = u.unitprice,
9
        discontinued = (u.discontinued = 0) ::boolean
    FROM updated_products u
2
    WHERE products.productid = u.productid
      AND u.discontinued = 0;
3
4
5
6
ta Output Messages Notifications
DATE 1
ery returned successfully in 230 msec.
```

If there are matching products and updated products .discontinued =1 then delete

```
156 ▼ DELETE FROM products
157 WHERE productid IN (
         SELECT u.productid
         FROM updated_products u
          WHERE u.discontinued = 1
160
          AND products.productid = u.productid
161
162
      );
163
Data Output Messages Notifications
DELETE 1
Query returned successfully in 72 msec.
```

Insert any new products from updated products that don't exist in products only if updated products .discontinued =0.

```
205 • INSERT INTO products (productid, productname, quantityperunit, unitprice, discontinued, categoryID)
206 SELECT u.productid, u.productname, u.quantityperunit, u.unitprice, (u.discontinued = 0)::boolean, u.categoryID
207
      FROM updated_products u
208
     WHERE u.discontinued = 0
209
      AND NOT EXISTS (
210
            SELECT 1 FROM products p WHERE p.productid = u.productid
       );
Data Output Messages Notifications
INSERT 0 2
Query returned successfully in 123 msec.
```

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

