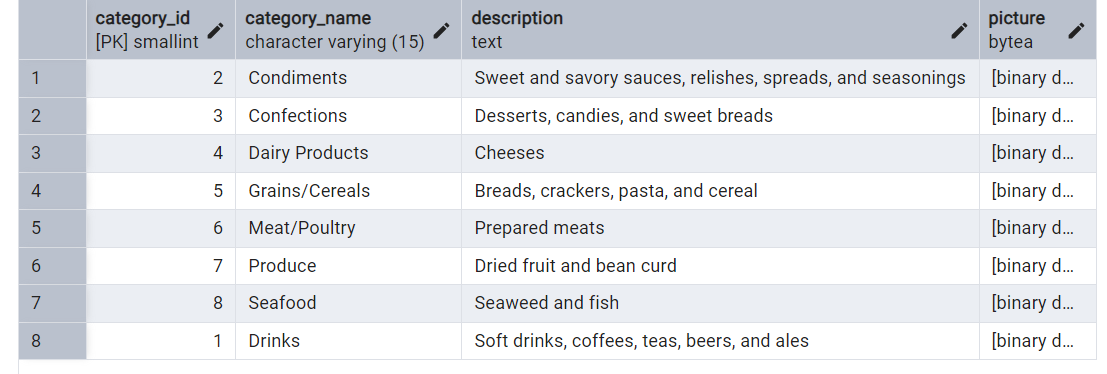
1)      Update the categoryName From “Beverages” to "Drinks" in the categories table.

UPDATE categories

set category\_name = 'Drinks'

where category\_name = 'Beverages';



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

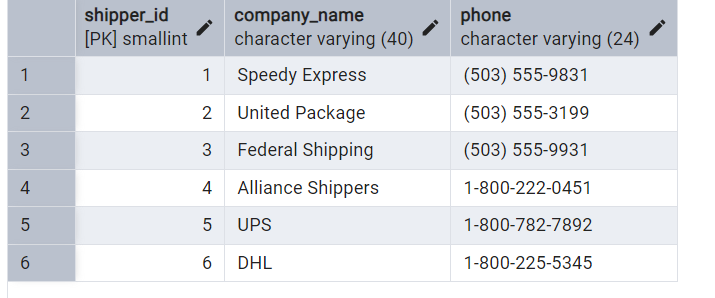
INSERT INTO shippers(shipper\_id,company\_name, phone)

VALUES(9,'USPS', 5232352345);



DELETE FROM shippers

where shipper\_id = 9;



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.

* ALTER TABLE products

DROP CONSTRAINT fk\_products\_categories

* ALTER TABLE products

ADD CONSTRAINT fk\_products\_categories

FOREIGN KEY(category\_id) REFERENCES categories(category\_id)

ON UPDATE CASCADE

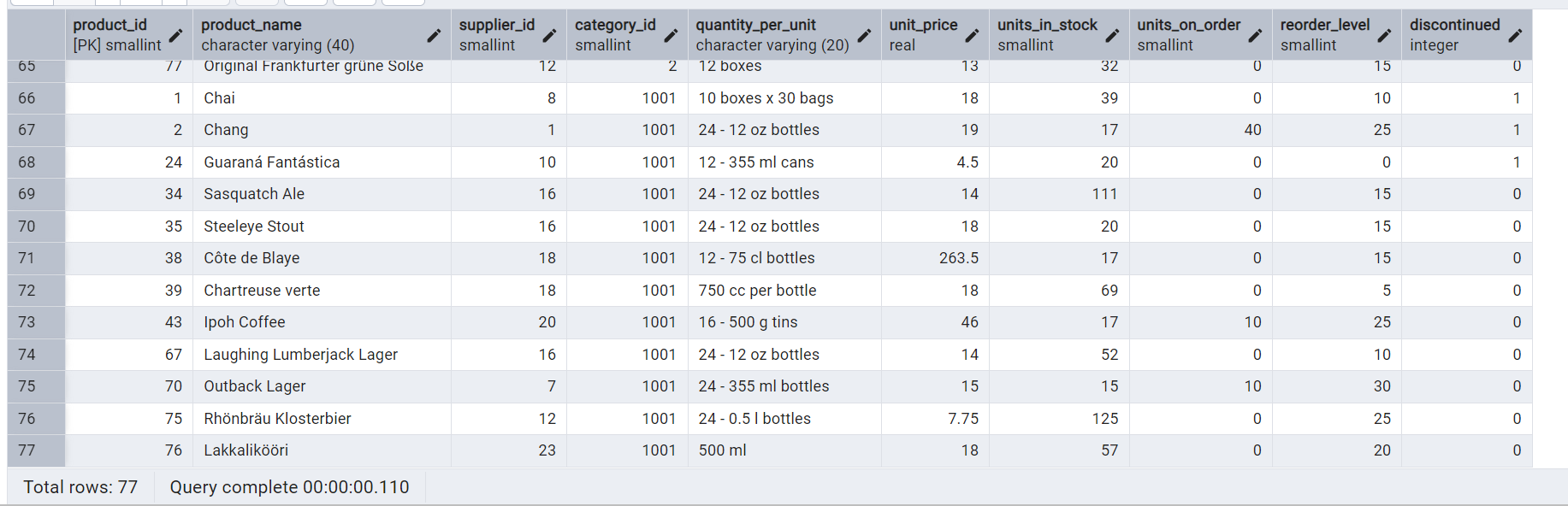
ON DELETE CASCADE;

* UPDATE categories

SET category\_id = 1001

where category\_id = 1;





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

* ALTER TABLE order\_details

DROP CONSTRAINT IF EXISTS fk\_order\_details\_products

* ALTER TABLE order\_details

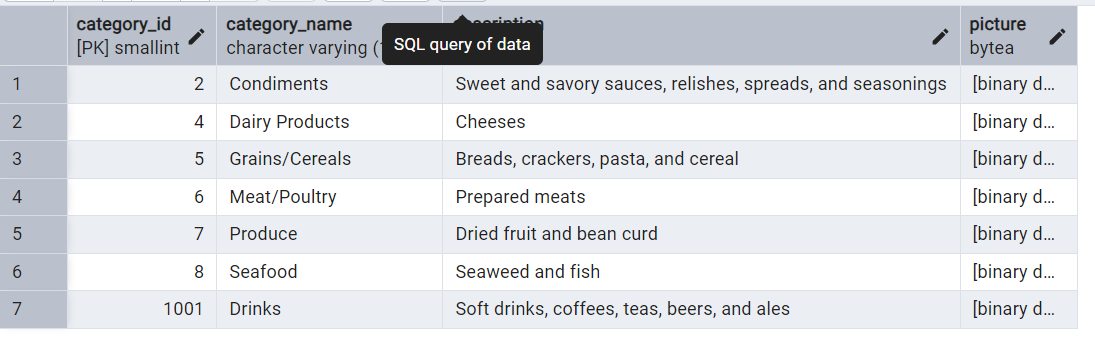
ADD CONSTRAINT fk\_order\_details\_products

FOREIGN KEY(product\_id) REFERENCES products(product\_id)

ON DELETE CASCADE;

* DELETE from categories

where category\_id = 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

DROP CONSTRAINT IF EXISTS fk\_orders\_customers

* ALTER TABLE orders

ADD CONSTRAINT fk\_orders\_customers

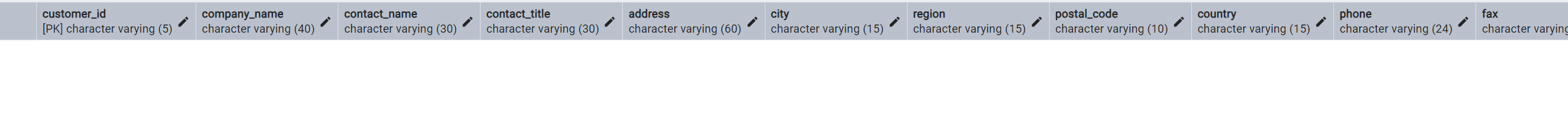
FOREIGN KEY(customer\_id) REFERENCES customers(customer\_id)

ON UPDATE CASCADE

ON DELETE SET NULL;

* DELETE from customers

where customer\_id = 'VINET';



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(product\_id, product\_name, quantity\_per\_unit, unit\_price, discontinued, category\_ID)

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

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

(100, 'Wheat bread', '10 boxes', 13, 0, 5)

ON CONFLICT(product\_id)

DO UPDATE

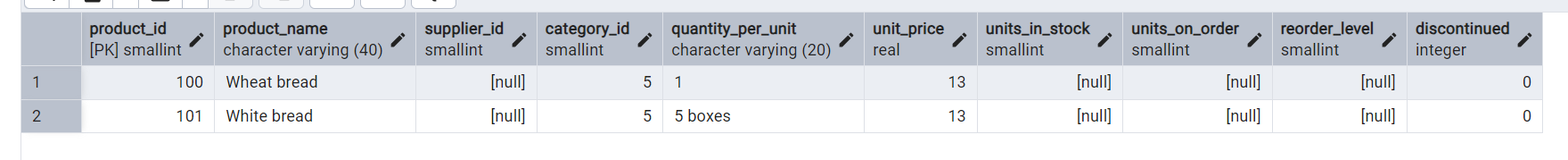
SET product\_name = EXCLUDED.product\_name,

quantity\_per\_unit = EXCLUDED.quantity\_per\_unit,

unit\_price = EXCLUDED.unit\_price,

discontinued = EXCLUDED.discontinued,

category\_ID = EXCLUDED.category\_ID;



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.

MERGE INTO products p

USING updated\_product up

ON p.product\_id = up.productid

WHEN MATCHED AND up.discontinued = 1 THEN DELETE

WHEN MATCHED AND up.discontinued = 0 THEN UPDATE

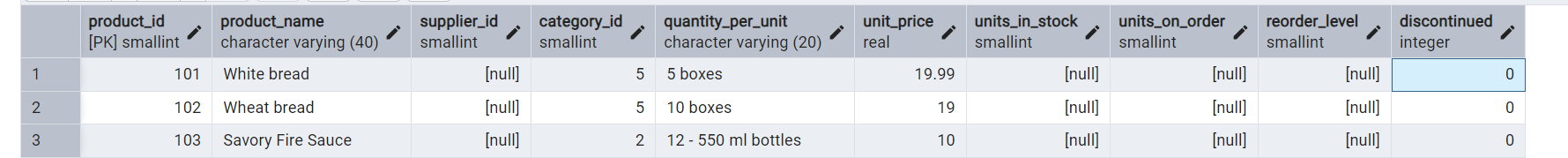
SET unit\_price = up.unitprice,

discontinued = up.discontinued

WHEN NOT MATCHED AND up.discontinued = 0 THEN

INSERT (product\_id, product\_name, quantity\_per\_unit, unit\_price, discontinued, category\_ID)

VALUES(up.productid, up.productname, up.quantityperunit, up.unitprice, up.discontinued, up.categoryID)



https://d.docs.live.net/6e9feecb47d52786/Desktop/Sumathi\_DAY3\_SQL\_assignment1.docx

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

select order\_id, (employees.first\_name|| ' ' || employees.last\_name) as employee\_fullName from orders

INNER JOIN employees ON employees.employee\_id = orders.employee\_id;

