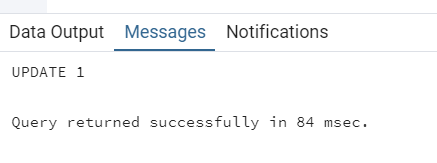
**Day3 Assignment**

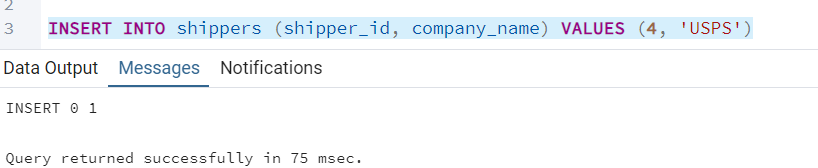
1. Update the categoryName From “Beverages” to "Drinks" in the categories table.

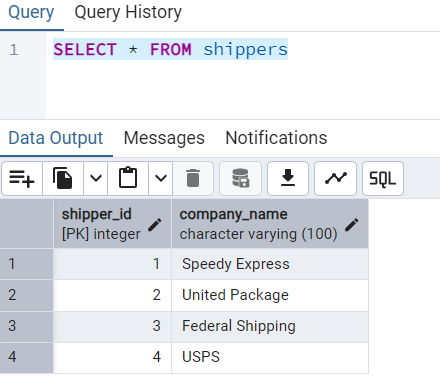
**UPDATE categories set category\_name = 'Drinks' WHERE category\_name = 'Beverages'**

****

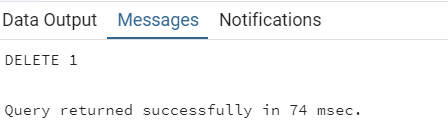
1. Insert into shipper new record (give any values) Delete that new record from shippers table

**INSERT INTO shippers (shipper\_id, company\_name) VALUES (4, 'USPS')**





**DELETE FROM shippers where shipper\_id = 4**



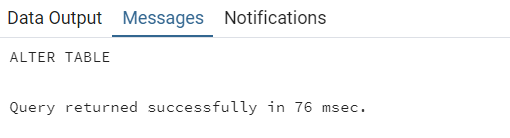
1. Update categoryID=1 to categoryID=1001. Make sure related products update their categoryID too. Display the both category and products table to show the cascade

**ALTER TABLE products**

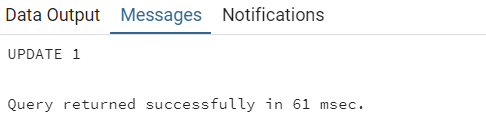
**DROP CONSTRAINT category\_id\_fk,**

**ADD CONSTRAINT category\_id\_fk FOREIGN KEY (category\_id)**

**REFERENCES categories (category\_id) ON UPDATE CASCADE ON DELETE CASCADE**

****

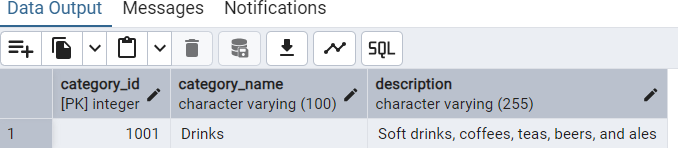
**UPDATE categories set category\_id = 1001 where category\_id = 1**

****

**SELECT \* from products where category\_id = 1001**

****

**SELECT \* from categories where category\_id = 1001**

****

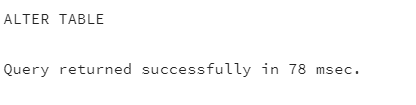
1. Delete the customer = “VINET” from customers. Corresponding customers in orders table should be set to null (HINT: Alter the foreign key on orders(customerID) to use ON DELETE SET NULL)

**ALTER TABLE orders**

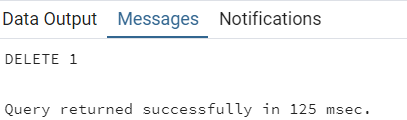
**DROP CONSTRAINT customer\_id\_fk,**

**ADD CONSTRAINT customer\_id\_fk FOREIGN KEY (customer\_id)**

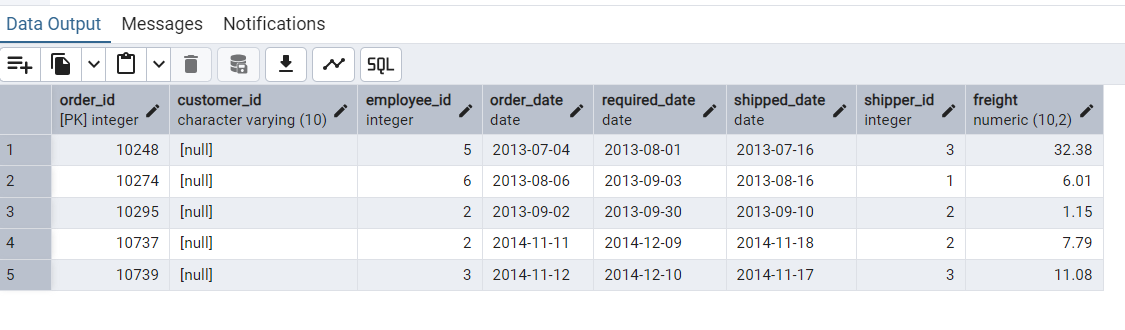
**REFERENCES customers (customer\_id) ON UPDATE CASCADE ON DELETE SET NULL**

****

**DELETE FROM customers where customer\_id = 'VINET'**

****

**SELECT \* FROM orders WHERE customer\_id IS NULL**

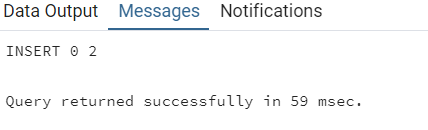
****

1. Insert the following data to Products using UPSERT

**INSERT INTO products (product\_id, product\_name, quantity\_per\_unit, unit\_price, discontinued, category\_id)**

**VALUES (100, 'Wheat bread', '1', 13, FALSE, 5),**

**(101, 'White bread', '5 boxes', 13, FALSE, 5)**

****

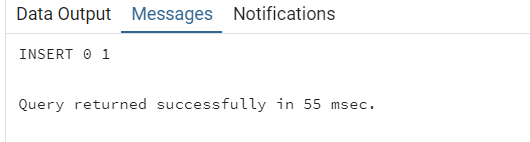
**INSERT INTO products (product\_id, product\_name, quantity\_per\_unit, unit\_price, discontinued, category\_id)**

**VALUES (100, 'White bread', '10 boxes', 13, FALSE, 5)**

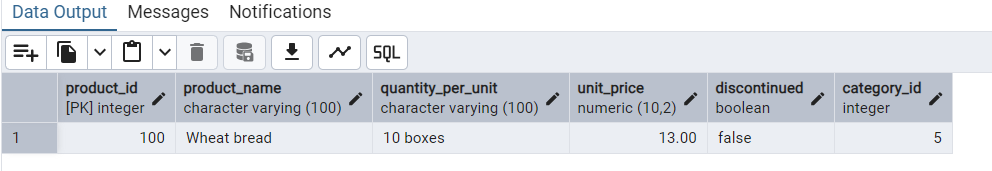
**ON CONFLICT(product\_id)**

**DO UPDATE SET**

**quantity\_per\_unit = EXCLUDED.quantity\_per\_unit**

****

**SELECT \* FROM products WHERE product\_id = 100**

****

1. **Merge Query**

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

**CREATE TEMP TABLE updated\_products(**

**product\_id INTEGER,**

**product\_name VARCHAR(100),**

**quantity\_per\_unit VARCHAR(100),**

**unit\_price NUMERIC(10,2),**

**discontinued BOOLEAN,**

**category\_id INTEGER);**

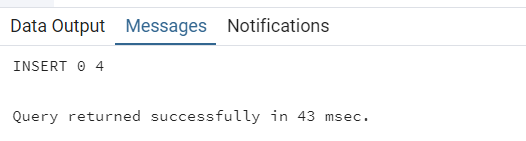
**INSERT INTO updated\_products(product\_id, product\_name, quantity\_per\_unit, unit\_price, discontinued, category\_id)**

**VALUES (100, 'Wheat bread', '10', 20, TRUE, 5),**

**(101, 'White bread', '5 boxes', 19.99, FALSE, 5),**

**(102, 'Midnight Mango Fizz', '24 - 12 oz bottles', 19, FALSE, 1),**

**(103, 'Savory Fire Sauce', '12 - 550 ml bottles', 10, FALSE, 2)**

****

**MERGE INTO products USING updated\_products ON products.product\_id = updated\_products.product\_id**

**WHEN MATCHED AND updated\_products.discontinued = FALSE THEN**

**UPDATE SET unit\_price = updated\_products.unit\_price, discontinued = updated\_products.discontinued**

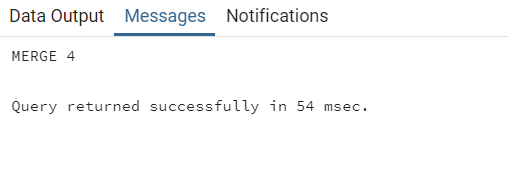
**WHEN MATCHED AND updated\_products.discontinued = TRUE THEN DELETE**

**WHEN NOT MATCHED AND updated\_products.discontinued = FALSE THEN**

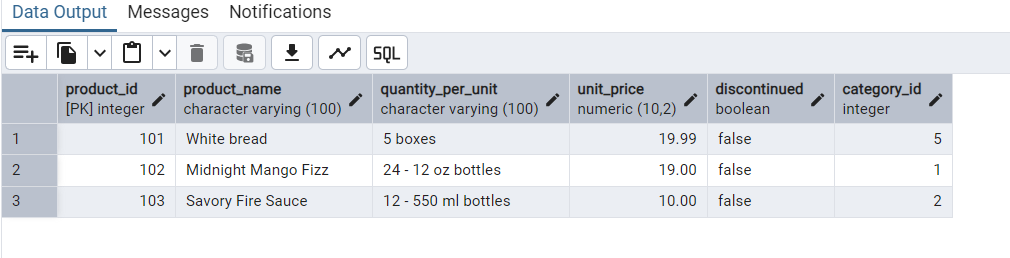
**INSERT (product\_id, product\_name, quantity\_per\_unit, unit\_price, discontinued, category\_id)**

**VALUES(updated\_products.product\_id, updated\_products.product\_name, updated\_products.quantity\_per\_unit,**

**updated\_products.unit\_price, updated\_products.discontinued, updated\_products.category\_id)**

****

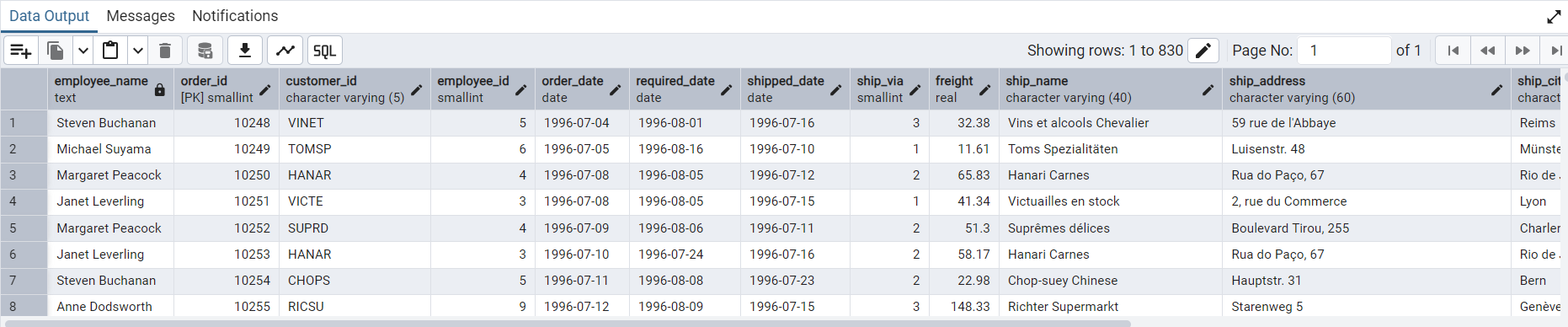
**select \* from products where product\_id IN (101, 102, 103, 100)**

****

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

**select CONCAT(e.first\_name, ' ', e.last\_name) AS employee\_name, o.\* from orders o INNER JOIN employees e**

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

****