Day 3 SQL bootcamp assignment

/\*q1. Update the categoryName From “Beverages” to "Drinks" in the categories table.\*/

update categories set category\_name ='Drinks' where category\_name = 'Beverages';

select \* from categories;

A screenshot of a computer

AI-generated content may be incorrect.

/\* q2. Insert into shipper new record (give any values) Delete that new record from shippers table.\*/

select \* from shippers;

insert into shippers (shipper\_id, company\_name, phone) values(7, 'USPS', 1234567890);

delete from shippers where shipper\_id=7;

A screenshot of a computer

AI-generated content may be incorrect.

/\*q3. 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)\*/

alter table products drop constraint if exists fk\_products\_categoryid\_fkey;

ALTER TABLE products

DROP CONSTRAINT fk\_products\_categories;

alter table products

add constraint products\_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 categories;

A screenshot of a computer

AI-generated content may be incorrect.

/\*q4. 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 orders\_customerid\_fkey;

ALTER TABLE orders

DROP CONSTRAINT fk\_orders\_customers;

alter table orders

add constraint orders\_customerid\_fkey

foreign key (customer\_id)

references customers(customer\_id)

on delete set null

delete from customers

where customer\_id='VINET'

select \* from customers where customer\_id='VINET';

select \* from orders where customer\_id='VINET';

A screenshot of a computer

AI-generated content may be incorrect.

/\*Q5. 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',5,13,0,5)

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;

insert into products (product\_id, product\_name, quantity\_per\_unit,unit\_price, discontinued, category\_id)

values (100,'White bread',10,13, 0,5)

on conflict (product\_id)

do update

set product\_name = excluded. product\_name,

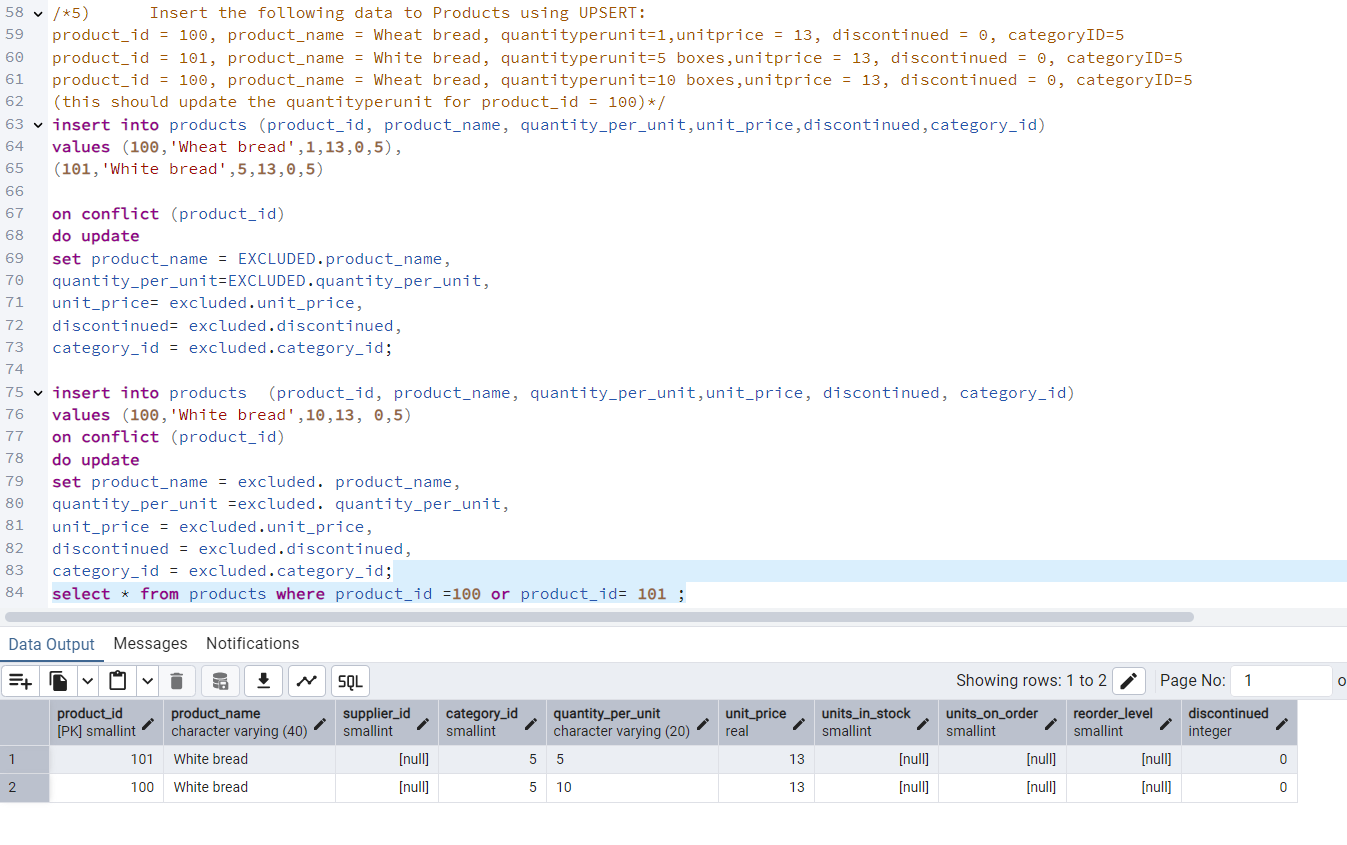
quantity\_per\_unit =excluded. quantity\_per\_unit,

unit\_price = excluded.unit\_price,

discontinued = excluded.discontinued,

category\_id = excluded.category\_id;

select \* from products where product\_id =100 or product\_id= 101 ;



Q6.       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.

CREATE TEMP TABLE updated\_products (

productID INT PRIMARY KEY,

productName VARCHAR(50),

quantityPerUnit VARCHAR(50),

unitPrice NUMERIC(8,2),

discontinued int,

categoryID INT

);

INSERT INTO updated\_products (productID, productName, quantityPerUnit, unitPrice, discontinued, categoryID)

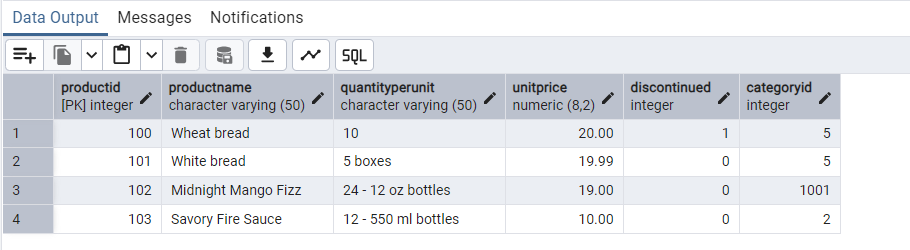
VALUES

(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, 1001),

(103, 'Savory Fire Sauce', '12 - 550 ml bottles', 10, 0, 2);



MERGE INTO products AS actual

USING updated\_products AS temp

ON actual.product\_id = temp.productID

WHEN MATCHED AND temp.discontinued = 0 THEN

UPDATE SET

unit\_price = temp.unitPrice,

discontinued = temp.discontinued

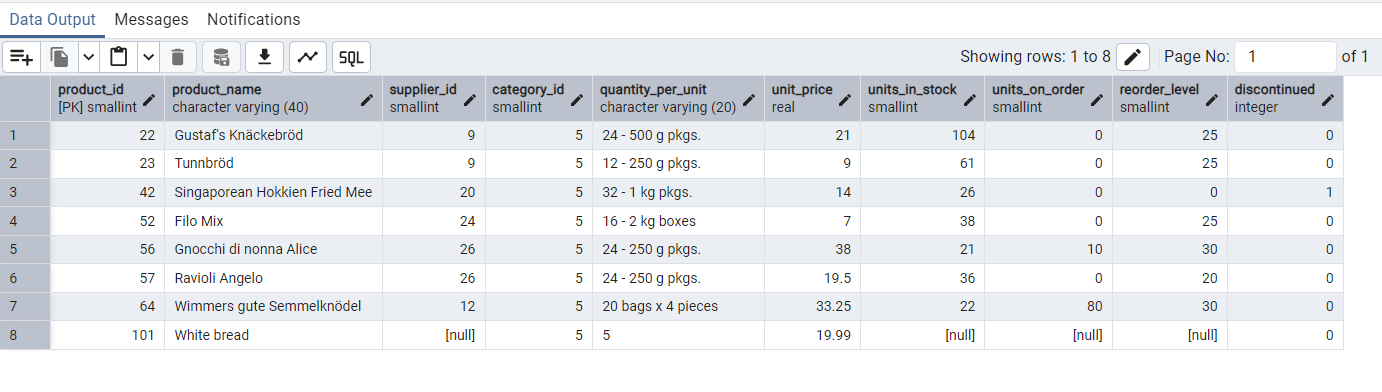
WHEN MATCHED AND temp.discontinued = 1 THEN

DELETE WHEN NOT MATCHED AND temp.discontinued = 0 THEN

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

VALUES (temp.productID, temp.productName, temp.quantityPerUnit, temp.unitPrice, temp.discontinued, temp.categoryID);

Select \* from products WHERE category\_id = 5;



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

SELECT o.order\_id, e.first\_name || ' ' || e.last\_name AS employeefullname

FROM orders o

INNER JOIN employees e ON o.employee\_id = e.employee\_id;

