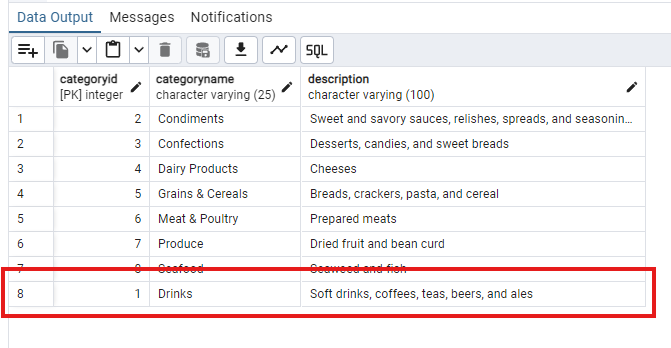
**Day 3-Assignment**

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

QUERY:

update categories set categoryname = 'Drinks' where categoryname ='Beverages';

OUTPUT:



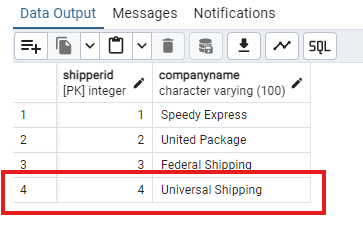
2) Insert into shipper new record (give any values) Delete that new record from shippers table.

QUERY:

insert into shippers(shipperid,companyname)

values (4,'Universal Shipping');

OUTPUT:

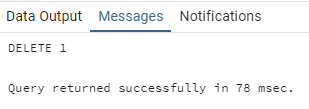


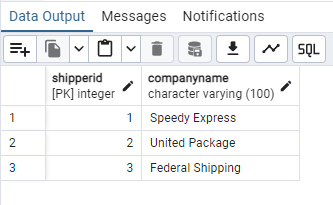
**DELETE Newly Created record:**

QUERY:

delete from shippers where shipperid=4;

OUTPUT:





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.

QUERY:

alter table products

add constraint fk\_categoryid

Foreign key (categoryid) references categories(categoryid)

on update cascade

on delete cascade;

OUTPUT: Categories table

A screenshot of a computer

AI-generated content may be incorrect.

OUTPUT Products table:

A screenshot of a computer

AI-generated content may be incorrect.

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

QUERY:

delete from categories where categoryid=3;

A screenshot of a computer

AI-generated content may be incorrect.

Query to check in Products Table:

select \* from products where categoryid=3;

OUTPUT Products Table:

A screenshot of a computer

AI-generated content may be incorrect.

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)

QUERY:

delete from customers where customerid='VINET';

OUTPUT:

select \* from customers where customerid='VINET';

A close-up of a computer screen

AI-generated content may be incorrect.

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)

QUERY:

INSERT INTO products (productid,productname,quantityperunit,unitprice,discontinued,categoryid)

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

INSERT INTO products (productid,productname,quantityperunit,unitprice,discontinued,categoryid)

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

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

ON CONFLICT (productid)

DO UPDATE SET productname = EXCLUDED.productname,

quantityperunit = EXCLUDED.quantityperunit;

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

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.

QUERY:

CREATE table products\_new(

productID INT PRIMARY KEY,

productName VARCHAR(100),

QuantityPerUnit VARCHAR(50),

unitPrice double precision,

discontinued int,

categoryID INT,

CONSTRAINT fk\_cat\_prod\_new

FOREIGN KEY(categoryID) REFERENCES categories(categoryID)

);

COPY products\_new FROM 'C:\archive\_nwt\products.csv'

WITH (FORMAT csv, HEADER true, ENCODING 'LATIN1');

CREATE temporary table updated\_products(

productID SERIAL PRIMARY KEY,

productName VARCHAR(100),

QuantityPerUnit VARCHAR(50),

unitPrice double precision,

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\_new as p

USING updated\_products as updated

ON p.productid = updated.productid

WHEN MATCHED AND updated.discontinued = 1 THEN

DELETE

WHEN MATCHED AND updated.discontinued = 0 THEN

UPDATE SET

discontinued = updated.discontinued,

unitprice = updated.unitprice

WHEN NOT MATCHED THEN

INSERT (productid,productname,quantityperunit,unitprice,discontinued,categoryid)

VALUES (updated.productid,updated.productname,updated.quantityperunit,updated.unitprice,updated.discontinued,updated.categoryid);

select \* from products\_new where productid in (100,101,102,103);

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

**USE NEW Northwind DB:**

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

QUERY:

select o.order\_id,concat(emp.first\_name,' ',emp.last\_name) as employee\_fullname from orders o inner join employees emp on o.employee\_id=emp.employee\_id;

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.