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

Update public.categories

Set category\_name ='Drinks'

Where category\_name = 'Beverages';

A screenshot of a computer

AI-generated content may be incorrect.

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

Insert Into public.shippers

Values (7, 'Welfare', '123-589-2578');

A screenshot of a computer

AI-generated content may be incorrect.

Delete From public.shippers

Where shipper\_id=7;

A screenshot of a computer

AI-generated content may be incorrect.

--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.

--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)

**Altering Table**

Alter Table public.products

Add Constraint fk\_products\_categories

Foreign Key (category\_id)

References categories(category\_id)

On Update Cascade

On Delete Cascade;

**Category\_id in category table before update:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Category\_id in products table before update:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Category\_id in category table after update:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Category\_id in products table after update:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Delete From public.categories**

**Where category\_id=1001;**

**Categories Table output after delete**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Product Table output after delete**

**A close-up of a website

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)

Alter Table public.orders

Drop Constraint If Exists fk\_orders\_customers;

Alter Table public.orders

Add Constraint fk\_orders\_customers

Foreign Key(customer\_id)

References customers(customer\_id)

On Delete Set NULL;

Delete From public.customers

Where customer\_id ='VINET';

Select \* from public.customers

Where contact\_name ='VINET';

A close-up of a computer screen

AI-generated content may be incorrect.

--5)Insert the following data to Products using UPSERT:

1)--product\_id = 100, product\_name = Wheat bread, quantityperunit=1,unitprice = 13, discontinued = 0, categoryID=3

2)--product\_id = 101, product\_name = White bread, quantityperunit=5 boxes,unitprice = 13, discontinued = 0, categoryID=3

3)--product\_id = 100, product\_name = Wheat bread, quantityperunit=10 boxes,unitprice = 13, discontinued = 0, categoryID=3

--(this should update the quantityperunit for product\_id = 100)

1)

Insert Into products (product\_id,product\_name,quantity\_per\_unit,unit\_price,discontinued,category\_id)

Values (100,'Wheat bread','1',13,0,3)

On Conflict (product\_id)

Do Update

Set quantity\_per\_unit = EXCLUDED.quantity\_per\_unit;

A screenshot of a computer

AI-generated content may be incorrect.

2)

Insert Into products(product\_id,product\_name,quantity\_per\_unit,unit\_price,discontinued,category\_id)

Values (101,'White bread','5 boxes',13,0,3)

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;

A screenshot of a computer

AI-generated content may be incorrect.

3)

Insert Into products (product\_id,product\_name,quantity\_per\_unit,unit\_price,discontinued,category\_id)

Values (100,'Wheat bread','10',13,0,3)

On Conflict (product\_id)

Do Update

Set quantity\_per\_unit = EXCLUDED.quantity\_per\_unit;

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 3

101 White bread 5 boxes 19.99 0 3

102 Midnight Mango Fizz 24 - 12 oz bottles 19 0 1

103 Savory Fire Sauce 12 - 550 ml bottles 10 0 2

Create Temp Table updated\_products (

productID Int,

productName Varchar(255),

quantityPerUnit Varchar(255),

unitPrice Numeric(10,2),

discontinued Int,

categoryID Int );

Insert Into updated\_products (productID, productName, quantityPerUnit, unitPrice, discontinued, categoryID)

Values

(100, 'Wheat bread', '10', 20, 1, 3),

(101, 'White bread', '5 boxes', 19.99, 0, 3),

(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 from above table ‘updated\_products’ only if there are matching products and updated\_products .discontinued =0

Update public.products

Set

unit\_price = updated\_products.unitPrice,

discontinued = updated\_products.discontinued

From updated\_products

Where Products.product\_name= updated\_products.productName

And updated\_products.discontinued = 0;

A screenshot of a computer screen

AI-generated content may be incorrect.

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

Delete From Products

Using updated\_products

Where Products.product\_id = updated\_products.productID

And updated\_products.discontinued = 1;

A screenshot of a message

AI-generated content may be incorrect.

--Insert any new products from updated\_products that don’t exist in products only if updated\_products .discontinued =0.

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

SELECT productID, productName, quantityPerUnit, unitPrice, discontinued, categoryID

FROM updated\_products

WHERE updated\_products.discontinued = 0

On Conflict (product\_id) Do Nothing;

Alter Table public.products

Drop Constraint If Exists fk\_products\_categories;

Alter Table products

Add Constraint fk\_products\_categories

Foreign Key(category\_id)

References categories(category\_id)

On Update Cascade;

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

Select emp.first\_name||' '||emp.last\_name As fullName, ord.\*

From public.employees emp

Inner Join public.orders ord

On emp.employee\_id=ord.employee\_id

A screenshot of a computer

AI-generated content may be incorrect.