Assignment 3

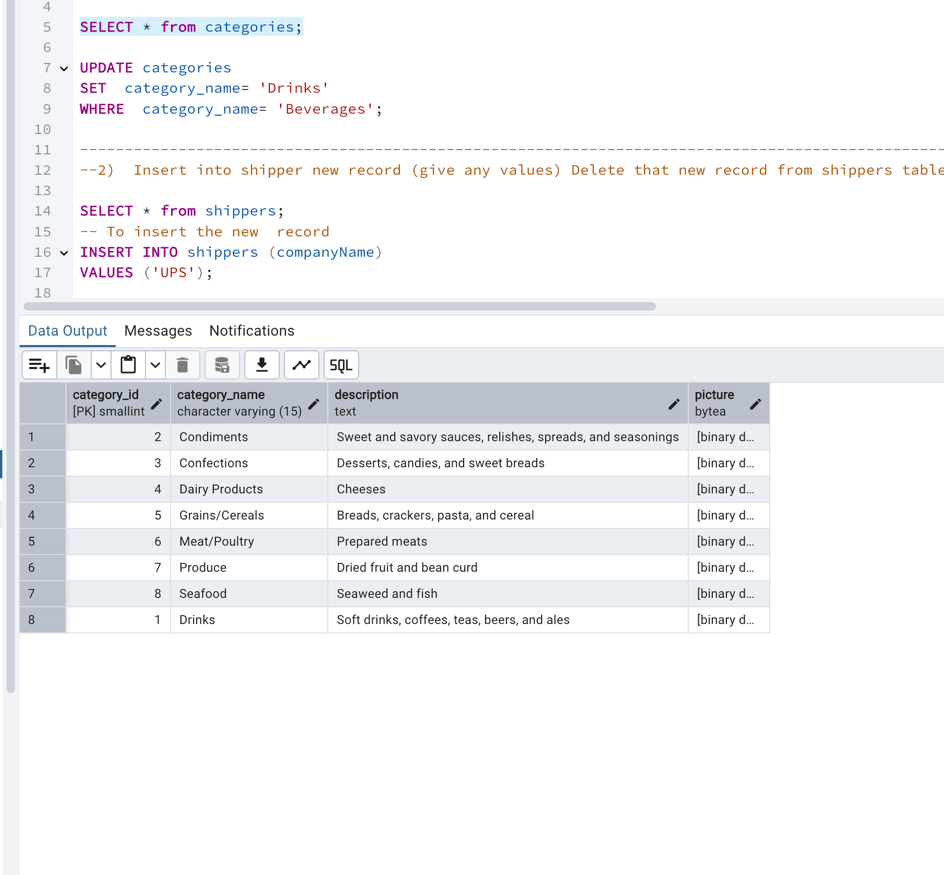
--1)Update the categoryName From “Beverages” to "Drinks" in the categories table.

SELECT \* from categories;

UPDATE categories

SET categoryname = 'Drinks'

WHERE categoryname= 'Beverages';



---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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

SELECT \* from shippers;

-- To insert the new record

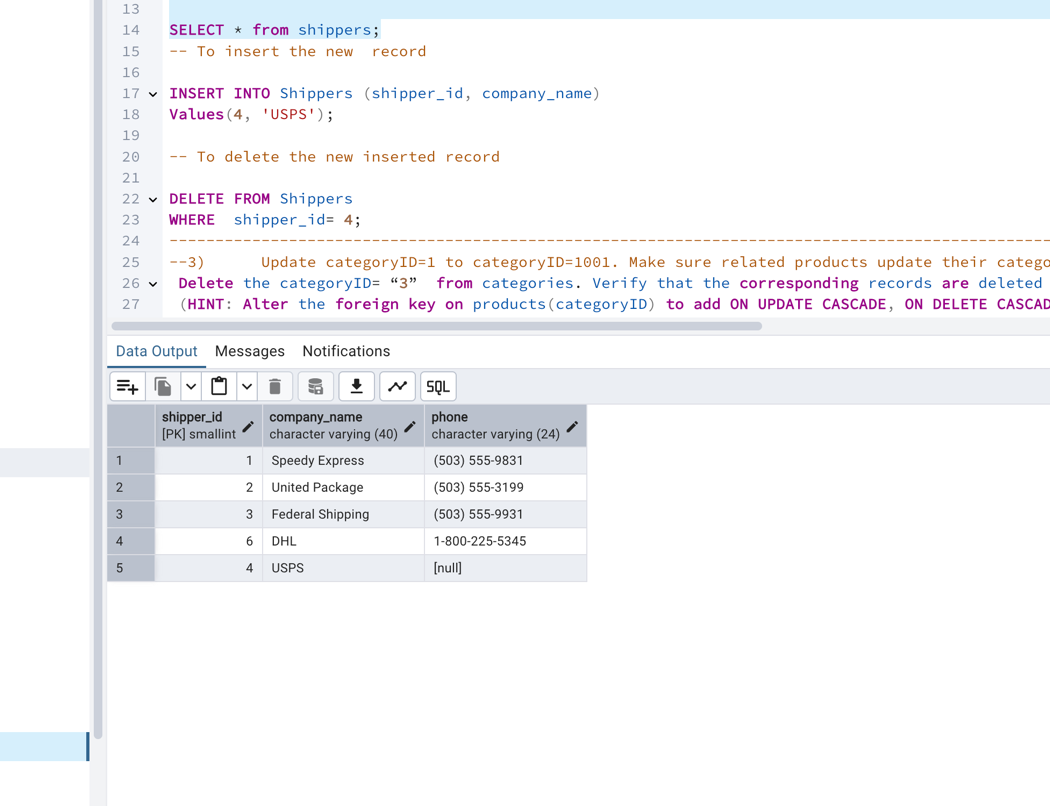
INSERT INTO Shippers (shipper\_id, company\_name)

Values(4, 'USPS');

-- To delete the new inserted record

DELETE FROM Shippers

WHERE shipper\_id= 4;



---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

--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, add ON DELETE CASCADE for order\_details(productid) )

--To find the foreign key name

SELECT CONSTRAINT\_NAME, TABLE\_NAME

FROM information\_schema.KEY\_COLUMN\_USAGE

WHERE TABLE\_NAME = 'products' AND CONSTRAINT\_SCHEMA = 'Northwind Traders';

-- To drop existing FK if any on products table

ALTER TABLE products

DROP FOREIGN KEY fk\_products\_category;

-- To add FK with ON UPDATE and ON DELETE CASCADE

ALTER TABLE products

ADD CONSTRAINT fk\_products\_category

FOREIGN KEY (category\_id)

REFERENCES categories(category\_id)

ON UPDATE CASCADE

ON DELETE CASCADE;

SELECT \* FROM products;

-- To drop existing FK if any on order\_details table

ALTER TABLE order\_details

DROP FOREIGN KEY fk\_orderdetails\_product;

-- Add FK with ON DELETE CASCADE

ALTER TABLE order\_details

ADD CONSTRAINT fk\_orderdetails\_product

FOREIGN KEY (product\_id)

REFERENCES products(product\_id)

ON DELETE CASCADE;

SELECT \* FROM order\_details;

-- To update categoryID=1 to 1001

UPDATE categories

SET category\_id = 1001

WHERE category\_id = 1;

SELECT \* FROM categories

WHERE category\_id = 1;

-- To display categories and products to show the update

SELECT \* FROM categories;

SELECT \* FROM products;

-- To delete categoryID=3 from categories

DELETE FROM categories

WHERE category\_id = 3;----

SELECT \* FROM products 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)

-- TO drop the existing foreign key (if it exists):

ALTER TABLE orders

DROP FOREIGN KEY fk\_orders\_customers;

--To add a new foreign key with ON DELETE SET NULL:

ALTER TABLE orders

ADD CONSTRAINT fk\_orders\_customers

FOREIGN KEY (customerID)

REFERENCES customers(customerID)

ON DELETE SET NULL;

-- To set the customerID to NULL in orders TABLE

ALTER TABLE orders

ALTER customerID VARCHAR(50) NULL;

-- To delete the customer ='VINET'

DELETE FROM customers

WHERE customerID = 'VINET';

SELECT \* FROM orders

WHERE customerID IS NULL;

------------------------------------------------------------------------------------------------------------------

--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 product\_id = 100

Insert OR update productid = 100

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

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

ON DUPLICATE KEY UPDATE

productname = VALUES(productname),

quantityperunit = VALUES(quantityperunit),

unitprice = VALUES(unitprice),

discontinued = VALUES(discontinued),

categoryid = VALUES(categoryid);

-- Insert productid = 101

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

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

ON DUPLICATE KEY UPDATE

productname = VALUES(productname),

quantityperunit = VALUES(quantityperunit),

unitprice = VALUES(unitprice),

discontinued = VALUES(discontinued),

categoryid = VALUES(categoryid);

-- Update product\_id = 100 with new quantityperunit

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

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

ON DUPLICATE KEY UPDATE

quantityperunit = VALUES(quantityperunit);

------------------------------------------------------------------------------------------------------------------

--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.

--To create temp TABLE updated\_products

CREATE TEMP TABLE updated\_products (

productid INT PRIMARY KEY,

productname VARCHAR(100),

quantityperunit VARCHAR(50),

unitprice DECIMAL(10, 2),

discontinued bool,

categoryid INT

);

-- To insert sample date into temp TABLE

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, 1),

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

SELECT \* FROM updated\_products;

--UPDATE matching products where discontinued = 0

UPDATE products p

SET

unitprice = u.unitprice,

discontinued = u.discontinued

FROM updated\_products u

WHERE p.productid = u.productid

AND u.discontinued = 0;

--2. DELETE matching products where discontinued = 1

DELETE FROM products p

USING updated\_products u

WHERE p.productid = u.productid

AND u.discontinued = 1;

3. INSERT new products where discontinued = 0 and not exists in products

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

SELECT u.productid, u.productname, u.quantityperunit, u.unitprice, u.discontinued, u.categoryid

FROM updated\_products u

LEFT JOIN products p ON p.productid = u.productid

WHERE p.productid IS NULL

AND u.discontinued = 0;

A screenshot of a computer

AI-generated content may be incorrect.

------------------------------------------------------------------------------------------------------------------

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

SELECT

o."orderID",

o."customerID",

o."orderDate",

e.employeeid,

e.employeename AS employeename

FROM orders o

INNER JOIN employees e ON o.employeeid = e.employeeid;

A screenshot of a computer

AI-generated content may be incorrect.