








Assignment 3

Day 3

USE Northwind from Kaggle:








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

```
UPDATE categories
SET categoryname = 'Drinks'
WHERE categoryname = 'Beverages';
SELECT * FROM categories;
```

Data Output Messages Notifications			
       SQL			
	categoryid [PK] integer	categoryname character varying (50)	description text
1	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
2	3	Confections	Desserts, candies, and sweet breads
3	4	Dairy Products	Cheeses
4	5	Grains & Cereals	Breads, crackers, pasta, and cereal
5	6	Meat & Poultry	Prepared meats
6	7	Produce	Dried fruit and bean curd
7	8	Seafood	Seaweed and fish
8	1	Drinks	Soft drinks, coffees, teas, beers, and ales

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

```
INSERT INTO shippers (shippersid, shippersname)
VALUES (4, 'Bulk Shipping');
SELECT * FROM shippers;
```

Data Output Messages Notifications		
      		
	shippersid [PK] integer	shippersname character varying (50)
1	1	Speedy Express
2	2	United Package
3	3	Federal Shipping
4	4	Bulk Shipping

```
DELETE FROM shippers
WHERE shippersid = 4;
```

Data Output Messages Notifications

	shippersid [PK] integer	shippersname character varying (50)
1	1	Speedy Express
2	2	United Package
3	3	Federal Shipping

- 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.
(HINT: Alter the foreign key on products(categoryID) to add ON UPDATE CASCADE, ON DELETE CASCADE)

```
17 ALTER TABLE product
18 DROP CONSTRAINT IF EXISTS product_categoryid_fkey;
19
20 ALTER TABLE product
21 ADD CONSTRAINT product_categoryid_fkey
22 FOREIGN KEY (categoryid)
23 REFERENCES categories(categoryid)
24 ON UPDATE CASCADE;
25
26 UPDATE categories
27 SET categoryid = 1001
28 WHERE categoryid = 1;
29 SELECT * FROM categories WHERE categoryid = 1001;
```

Data Output Messages Notifications

	categoryid [PK] integer	categoryname character varying (50)	description text
1	1001	Drinks	Soft drinks, coffees, teas, beers, and ales

```
30 SELECT * FROM product WHERE categoryid = 1001;
```

Data Output Messages Notifications

SQL

	productid [PK] integer	productname character varying (50)	quantityperunit text	unitprice double precision	discontinued double precision	categoryid integer
1	1	Chai	10 boxes x 20 ba...	18	0	1001
2	2	Chang	24 - 12 oz bottles	19	0	1001
3	24	Guarana Fantastica	12 - 355 ml cans	4.5	1	1001
4	34	Sasquatch Ale	24 - 12 oz bottles	14	0	1001
5	35	Steeleye Stout	24 - 12 oz bottles	18	0	1001
6	38	Côte de Blaye	12 - 75 cl bottles	263.5	0	1001
7	39	Chartreuse verte	750 cc per bottle	18	0	1001
8	43	Ipoh Coffee	16 - 500 g tins	46	0	1001
9	67	Laughing Lumberjack Lager	24 - 12 oz bottles	14	0	1001
10	70	Outback Lager	24 - 355 ml bottles	15	0	1001
11	75	Phoebus Kleisterbier	24 - 0.5 l bottles	7.75	0	1001
Total rows: 12 Query complete 00:00:00.076						

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

```
ALTER TABLE order_details
DROP CONSTRAINT IF EXISTS order_details_productid_fkey;
```

```
ALTER TABLE order_details
ADD CONSTRAINT order_details_productid_fkey
FOREIGN KEY (productid)
REFERENCES product(productid)
ON UPDATE CASCADE
ON DELETE CASCADE;
```

```
44 DELETE FROM categories
45 WHERE categoryid = 3;
46 SELECT * FROM product WHERE categoryid = 3;
```

Data Output Messages Notifications

SQL

	productid [PK] integer	productname character varying (50)	quantityperunit text	unitprice double precision	discontinued double precision	categoryid integer
--	---------------------------	---------------------------------------	-------------------------	-------------------------------	----------------------------------	-----------------------

- 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)

```
39 ALTER TABLE orders
40 DROP CONSTRAINT orders_customerid_fkey;
41
42 ALTER TABLE orders
43 ADD CONSTRAINT orders_customerid_fkey
44 FOREIGN KEY (customerid)
45 REFERENCES customers(customerid)
46 ON DELETE SET NULL;
47
48 DELETE FROM customers
49 WHERE customerid = 'VINET';
50 SELECT * FROM customers WHERE customerid = 'VINET';
```

Data Output Messages Notifications

									SQL
customerid	companyname	contactname							
[PK] character (5)	character varying (100)	character varying (100)							

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

product_id = 100, product_name = Wheat bread, quantityperunit=1,unitprice = 13, discontinued = 0, categoryID=5

```
INSERT INTO product (productid, productname, quantityperunit, unitprice, discontinued, categoryid)
VALUES (100, 'Wheat bread', 1, 13, 0, 5)
ON CONFLICT (productid)
DO UPDATE
SET productname = '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

```
INSERT INTO product (productid, productname, quantityperunit, unitprice, discontinued, categoryid)
VALUES (101, 'White bread', '5 boxes', 13, 0, 5)
ON CONFLICT (productid)
DO UPDATE
SET productname = '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

```
INSERT INTO product (productid, productname, quantityperunit, unitprice, discontinued, categoryid)
VALUES (100, 'Wheat bread', '10 boxes', 13, 0, 5)
ON CONFLICT (productid)
DO UPDATE
SET productname = 'Wheat bread',
    quantityperunit = '10 boxes',
    unitprice = 13,
    discontinued = 0,
    categoryid = 5;
```

(this should update the quantityperunit for product_id = 100)

```
105 SELECT * FROM product
106 WHERE productid IN(100, 101);
```

Data Output

Messages

Notifications

SQL

	productid [PK] integer	productname character varying (50)	quantityperunit text	unitprice double precision	discontinued double precision	categoryid integer
1	101	White bread	5 boxes	13	0	5
2	100	Wheat bread	10 boxes	13	0	5

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

- 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.

```

MERGE INTO product p
USING (
    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)
)
AS updated_products (productID ,productName, quantityPerUnit, unitPrice, discontinued, categoryID)
ON p.productid = updated_product.productID
WHEN MATCHED AND updated_products = 1 THEN
DELETE
WHEN MATCHED AND updated_products = 0 THEN
UPDATE SET
unitprice = updated_products.unitprice,
discontinued = updated_products.discontinued
WHEN NOT MATCHED AND updated_products.discontinued = 0 THEN
INSERT productid, productname, quantitytperunit, unitprice, discontinued, categoryid
VALUES (updated_products.productid,updated_products.productname, updated_products.quantityperunit, updated_products.unitprice, updated_products.discontinued, updated_products.categoryid)

```

USE NEW Northwind DB:

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

Query Query History

1 SELECT E.last_name || ' ' || E.first_name as fullname, O.order_id
2 FROM employees E
3 INNER JOIN orders O
4 ON E.employee_id = O.employee_id;

Data Output Messages Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	fullname text	order_id [PK] smallint
1	BuchananSteven	10248
2	SuyamaMichael	10249
3	PeacockMargaret	10250
4	LeverlingJanet	10251
5	PeacockMargaret	10252
6	LeverlingJanet	10253
7	BuchananSteven	10254
8	DodsworthAnne	10255
9	LeverlingJanet	10256
10	PeacockMargaret	10257
11	DavolioNancy	10258
12	PeacockMargaret	10259

Total rows: 830
Query complete 00:00:00.115