Day 3

USE Northwind from Kaggle:

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

```
UPDATE categories SET category_name = 'Drinks' WHERE category_name = 'Beverages';

select * from categories order by category_id;
```

Data Output Messages Notifications

	category_id [PK] smallint	category_name character varying (25)	description text
1	1	Drinks	Soft drinks, coffees, teas, beers, and ales
2	2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
3	3	Confections	Desserts, candies, and sweet breads
4	4	Dairy Products	Cheeses
5	5	Grains & Cereals	Breads, crackers, pasta, and cereal
6	6	Meat & Poultry	Prepared meats

Total rows: 8 Query complete 00:00:00.125

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

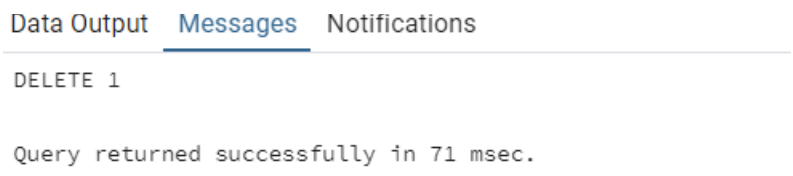
```
INSERT INTO shippers VALUES(4,'DHL');

SELECT * FROM shippers;
```

Data Output Messages Notifications

	shipper_id [PK] smallint	company_name character varying (40)
1	1	Speedy Express
2	2	United Package
3	3	Federal Shipping
4	4	DHL

Total rows: 4 Query complete 00:00:00.121

```
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)

```
ALTER TABLE products

DROP CONSTRAINT products_category_id_fkey;


ALTER TABLE products

ADD CONSTRAINT products_category_id_fkey

FOREIGN KEY (category_id)

REFERENCES categories (category_id)

ON UPDATE CASCADE

ON DELETE CASCADE;

SELECT * FROM categories WHERE Category_id = 1;
```

Data Output | Messages | Notifications

	category_id [PK] smallint	category_name character varying (25)	description text
1	1	Drinks	Soft drinks, coffees, teas, beers, and ales

Total rows: 1 | Query complete 00:00:00.190

SELECT * FROM PRODUCTS where category_id = 1;

Data Output | Messages | Notifications

	product_id [PK] smallint	product_name character varying (40)	quantity_per_unit character varying (20)	unit_price real	discontinued integer	category_id smallint
1	1	Chai	10 boxes x 20 bags	18	0	1
2	2	Chang	24 - 12 oz bottles	19	0	1
3	24	Guarana Fantastica	12 - 355 ml cans	4.5	1	1
4	34	Sasquatch Ale	24 - 12 oz bottles	14	0	1
5	35	Steeleye Stout	24 - 12 oz bottles	18	0	1
6	38	Côte de Blave	12 - 75 cl bottles	263.5	0	1

Total rows: 12 | Query complete 00:00:00.120

UPDATE categories SET category_id = 1001 where category_id = 1;

SELECT * FROM categories WHERE Category_id = 1001;

Data Output Messages Notifications

	category_id [PK] smallint	category_name character varying (25)	description text
1	1001	Drinks	Soft drinks, coffees, teas, beers, and ales

SELECT * FROM categories WHERE Category_id = 1001;

Data Output Messages Notifications

	product_id [PK] smallint	product_name character varying (40)	quantity_per_unit character varying (20)	unit_price real	discontinued integer	category_id smallint
1	1	Chai	10 boxes x 20 bags	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 Blave	12 - 75 cl bottles	263.5	0	1001

Total rows: 12 Query complete 00:00:00.154

ALTER TABLE order_details

DROP CONSTRAINT order_details_product_id_fkey;

ALTER TABLE order_details

ADD CONSTRAINT order_details_product_id_fkey

FOREIGN KEY (product_id)

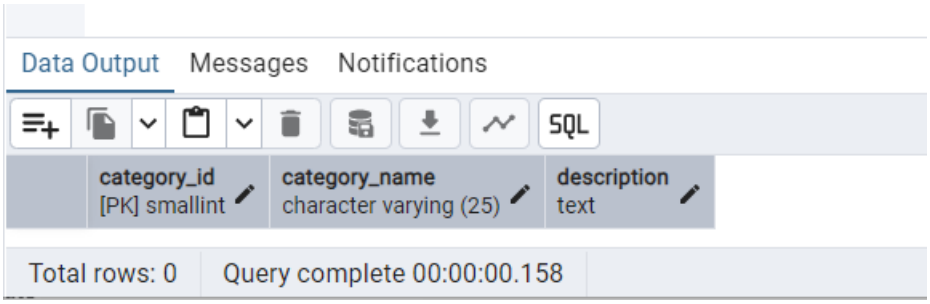
REFERENCES products (product_id)

ON UPDATE CASCADE

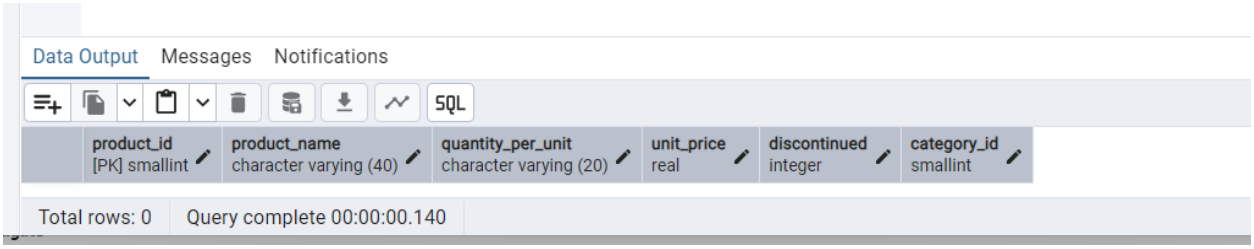
ON DELETE CASCADE;

DELETE FROM categories WHERE category_id = 3;

select * from categories where category_id =3 ;



select * from products where category_id = 3;

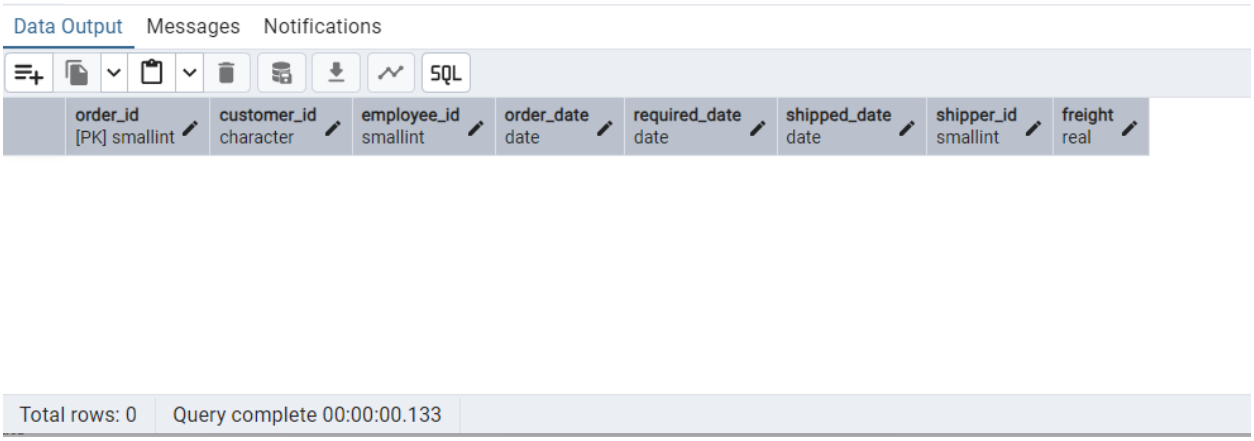


4) Delete the customer = “VINET” from customers. Corresponding customers in orders table should be set to null (HINT: Alter the foreign key on orders(customerID) to use ON DELETE SET NULL)

```
ALTER TABLE orders

DROP CONSTRAINT orders_customer_id_fkey;


ALTER TABLE orders

ADD CONSTRAINT orders_customer_id_fkey

FOREIGN KEY (customer_id)

REFERENCES customers (customer_id)

ON UPDATE CASCADE

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';



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

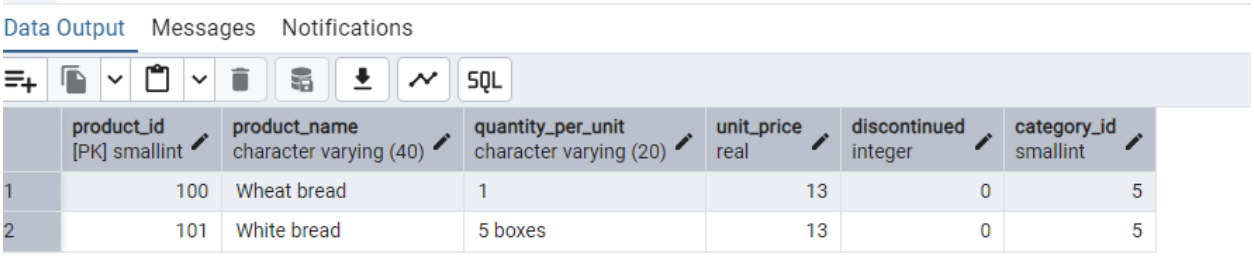
product_id = 100, product_name = Wheat bread, quantityperunit=10 boxes,unitprice = 13, discontinued = 0, categoryID=5

(this should update the quantityperunit for product_id = 100)

INSERT INTO products

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

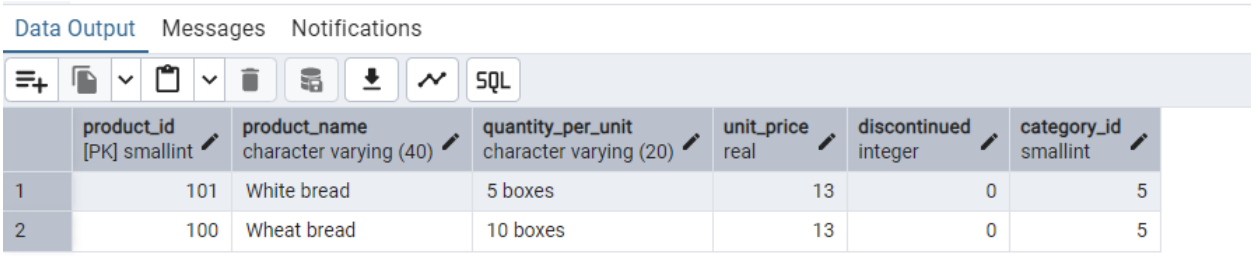
(101, 'White bread', '5 boxes', 13,0,5);



Data Output | Messages | Notifications

	product_id [PK] smallint	product_name character varying (40)	quantity_per_unit character varying (20)	unit_price real	discontinued integer	category_id smallint
1	100	Wheat bread	1	13	0	5
2	101	White bread	5 boxes	13	0	5

```
INSERT INTO products
VALUES (100, 'Wheat bread', '10 boxes', 13,0,5)
ON CONFLICT(product_id)
DO UPDATE SET
  quantity_per_unit = EXCLUDED.quantity_per_unit;

  SELECT * FROM PRODUCTS where product_id >=100;
```



Data Output | Messages | Notifications

	product_id [PK] smallint	product_name character varying (40)	quantity_per_unit character varying (20)	unit_price real	discontinued integer	category_id smallint
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.

```
CREATE TEMP TABLE updated_products
(   product_id smallint NOT NULL,
    product_name character varying(40),
    quantity_per_unit character varying(20),
    unit_price real,
    discontinued integer NOT NULL,
    category_id smallint
);

INSERT INTO updated_products
  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);



MERGE INTO products p
USING (
   SELECT
        product_id, product_name, quantity_per_unit, unit_price,discontinued,category_id
        FROM updated_products
) u
ON p.product_id = u.product_id
WHEN MATCHED AND u.discontinued = 1 THEN
   DELETE
WHEN MATCHED AND u.discontinued = 0 THEN
   UPDATE SET
      unit_price = u.unit_price
WHEN NOT MATCHED AND u.discontinued = 0 THEN
   INSERT (product_id, product_name, quantity_per_unit, unit_price,discontinued,category_id)
   VALUES (u.product_id, u.product_name, u.quantity_per_unit,
u.unit_price,u.discontinued,u.category_id);


 SELECT * FROM PRODUCTS where product_id >=100;
```

Data Output Messages Notifications

	product_id [PK] smallint	product_name character varying (40)	quantity_per_unit character varying (20)	unit_price real	discontinued integer	category_id smallint
1	101	White bread	5 boxes	19.99	0	5
2	102	Midnight Mango Fizz	24 - 12 oz bottles	19	0	1
3	103	Savory Fire Sauce	12 - 550 ml bottles	10	0	2

Total rows: 3 Query complete 00:00:00.155

USE NEW Northwind DB:

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

```
SELECT order_id, concat(e.first_name,' ', e.last_name) as employee_full_name FROM orders o
INNER JOIN employees e
on o.employee_id = e.employee_id
order by order_id;
```

Data Output Messages Notifications

	order_id [PK] smallint	employee_full_name text
1	10248	Steven Buchanan
2	10249	Michael Suyama
3	10250	Margaret Peacock
4	10251	Janet Leverling
5	10252	Margaret Peacock
6	10253	Janet Leverling
7	10254	Steven Buchanan

Total rows: 830 Query complete 00:00:00.213