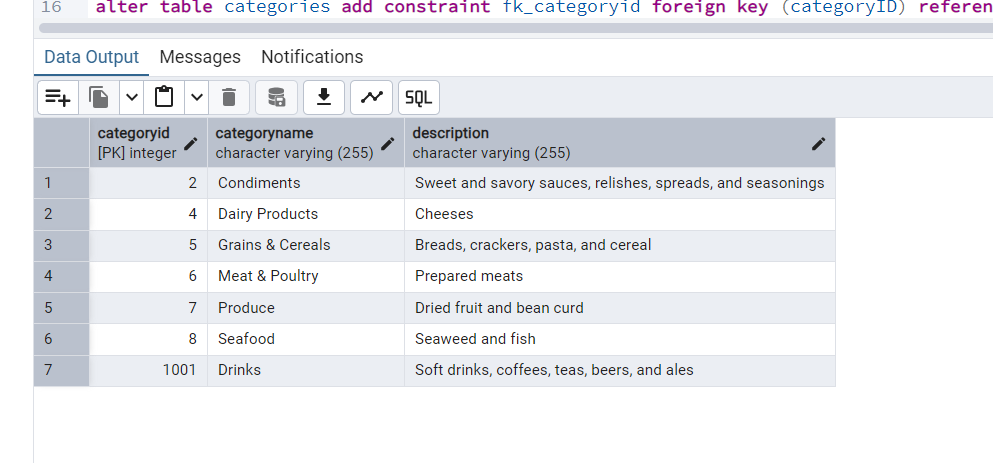
**Q1) Update the categoryName From “Beverages” to "Drinks" in the categories table.**

Query:

update categories SET categoryname= Replace(categoryname, 'Beverages','Drinks')

Output:



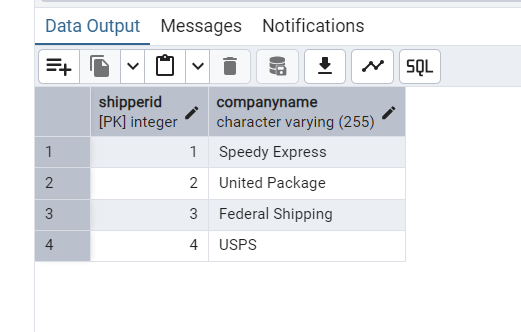
**Q2) Insert into shipper new record (give any values) Delete that new record from shippers table.**

Query:

select \* from shippers

insert into shippers (shipperid, companyname) values (4,'USPS')

Output:



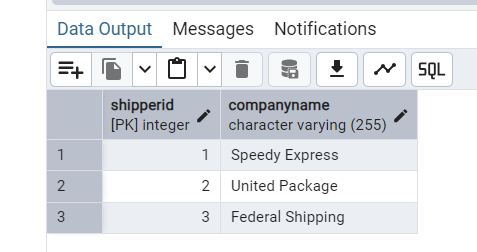
**Q2) delete the record**

Query:

delete from shippers where shipperid=4

select \* from shippers

Output:



**Q3) 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 categories drop constraint fk\_categoryid

---add constraint and cascade

alter table products add constraint fk\_product\_categories foreign key(categoryid) references categories(categoryid)

on update cascade

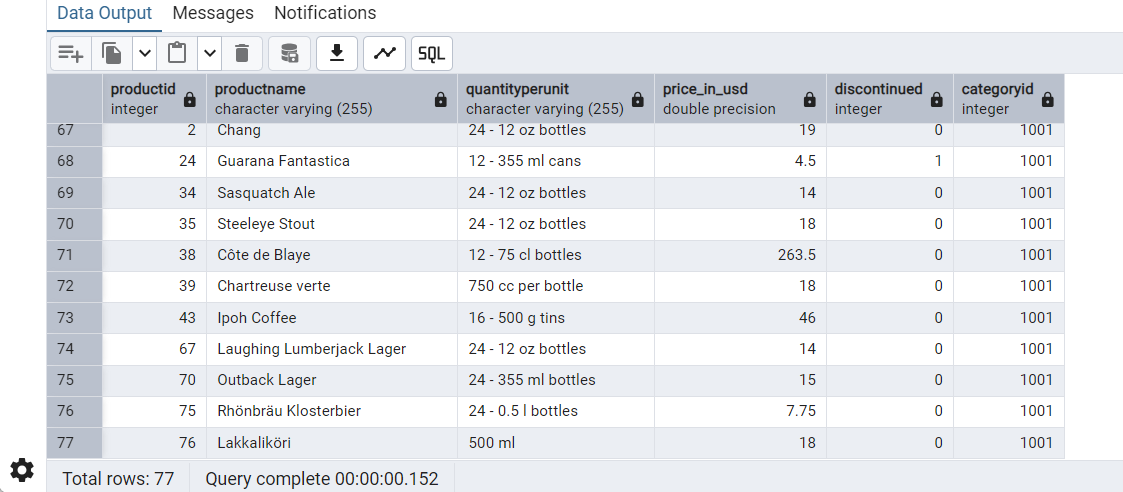
on delete cascade

----Update categoryID=1 to categoryID=1001

update categories set categoryid = 1001 where categoryid=1

Output:

Product Table:



**Categories Table:**



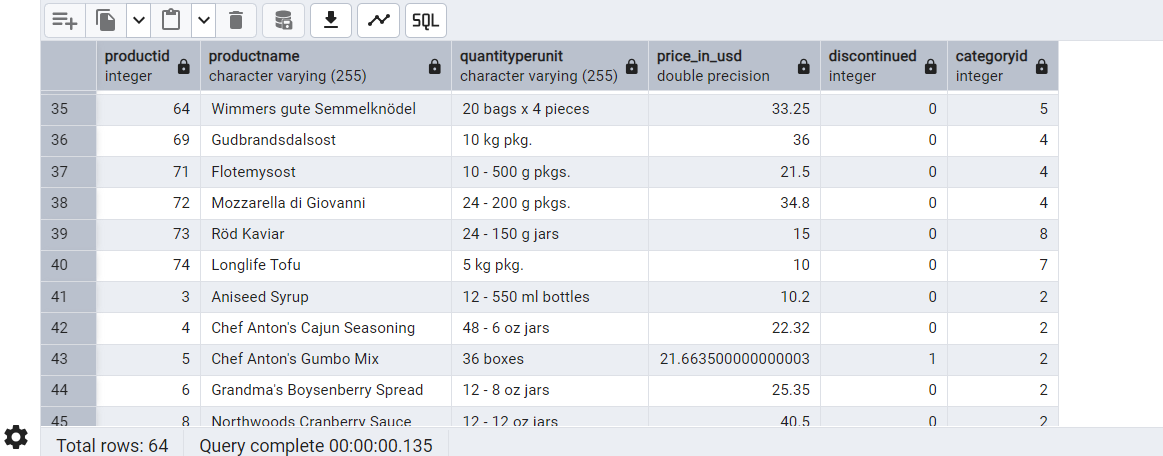
**Delete the categoryID= “3” from categories. Verify that the corresponding records are deleted automatically from products.**

Query:

delete from categories where categoryid=3

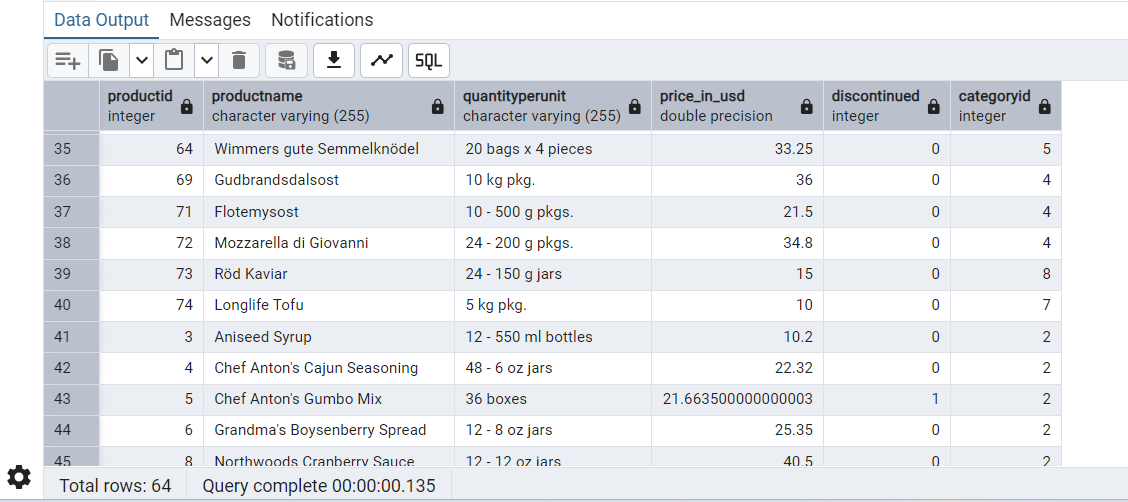
select \* from products

Output:



select \* from categories

Output:



**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:

--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 add constraint fk\_orders\_customer foreign key(customerid) references customers(customerid) on cascade

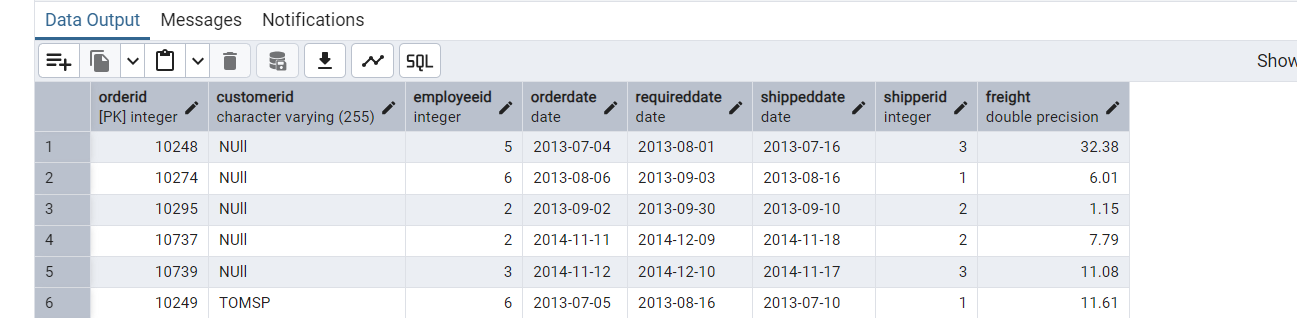
on delete SET NULL

--- delete the custoemr = 'VINET'

delete from customers where customerid = 'VINET'

select \* from orders where customerid='VINET'

Output:



**Q5) Insert the following data to Products using UPSERT:**

Query:

insert into products(productid,productname,quantityperunit,price\_in\_usd,discontinued,categoryid) values (100,'Wheat bread',1,13,0,3), (101,'White bread',5,13,0,3)

on conflict(productid)

Do update

set productname = EXCLUDED.productname,

quantityperunit = EXCLUDED.quantityperunit,

price\_in\_usd=EXCLUDED.price\_in\_usd,

discontinued=EXCLUDED.discontinued,

categoryid= ExCLUDED.categoryid

insert into products(productid,productname,quantityperunit,price\_in\_usd,discontinued,categoryid) values (100,'Wheat Bread',10,13,0,3)

on conflict(productid)

Do update

set productname = EXCLUDED.productname,

quantityperunit = EXCLUDED.quantityperunit,

price\_in\_usd=EXCLUDED.price\_in\_usd,

discontinued=EXCLUDED.discontinued,

categoryid= ExCLUDED.categoryid

Output:



**Q6) Write a MERGE query:**

Query:

Merge into products 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,price\_in\_usd,discontinued,categoryid)

on p.productid = updated\_products.productid

when matched and updated\_products.discontinued =0 then

update set

price\_in\_usd = updated\_products.price\_in\_usd,

discontinued = updated\_products.discontinued

when matched and updated\_products.discontinued =1 then

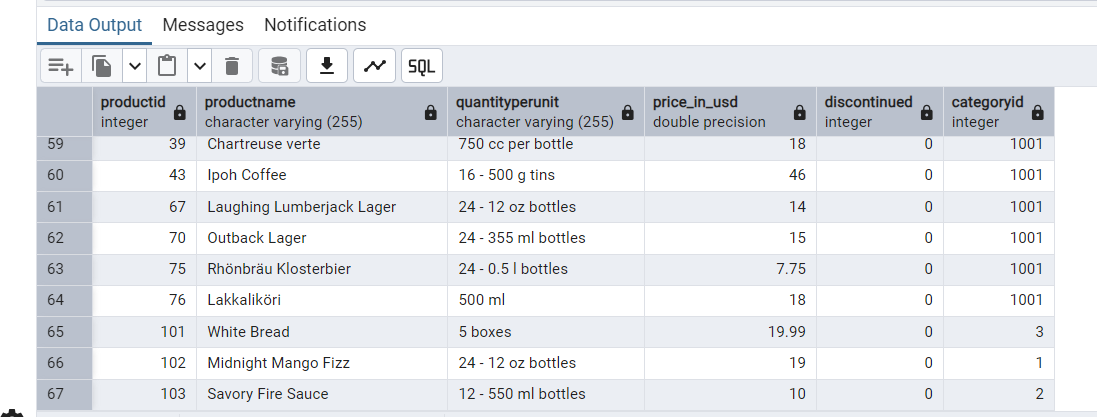
delete

when not matched and updated\_products.discontinued =0 then

insert(productid,productname,quantityperunit,price\_in\_usd,discontinued,categoryid)

values(updated\_products.productid,updated\_products.productname,updated\_products.quantityperunit,updated\_products.price\_in\_usd,updated\_products.discontinued,updated\_products.categoryid)

Output:



**Q7) List all orders with employee full names. (Inner join)**

Query:

select first\_name,last\_name,order\_id,ship\_name from employees inner join orders on employees.employee\_id = orders.employee\_id

Output:

