Day 3 Assignment

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

select \* from categories

update categories

set categoryname = 'Drinks'

where categoryname = '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.

select \* from shippers

-- Insert into shipper new record (give any values)

insert into shippers(shipperid,companyname)

values (5,'Numpyninja')

A screenshot of a computer

AI-generated content may be incorrect.

-- Delete that new record from shippers table.

delete from shippers

where shipperid = 5

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)

select \* from categories

select \* from products

ALTER TABLE products

DROP CONSTRAINT products\_categoryid\_fkey;

alter table products

add constraint fk\_category

foreign key(categoryID)

references categories(categoryID)

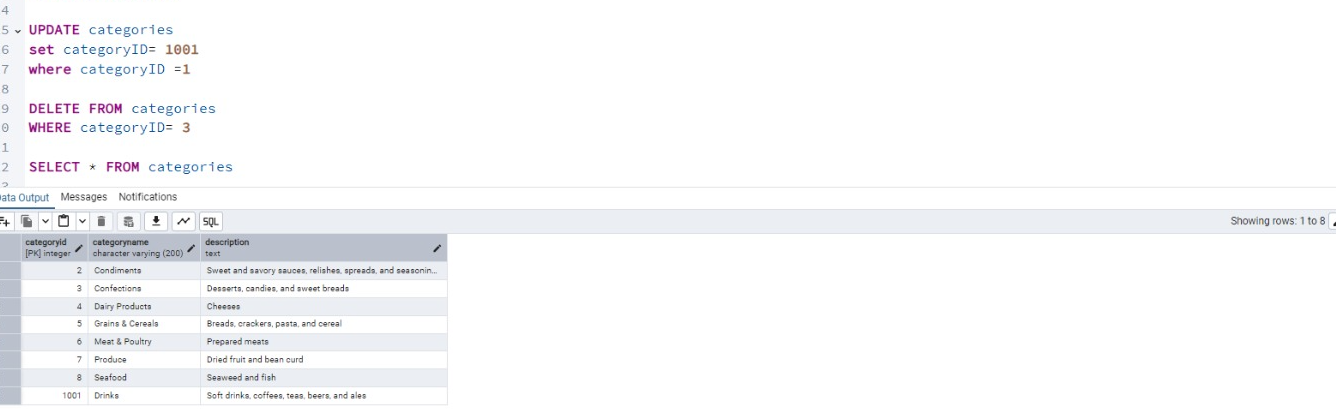
ON DELETE CASCADE

ON UPDATE CASCADE;

-- Update categoryID=1 to categoryID=1001

update categories

set categoryID = 1001



----- we have drop foreign key of order-details -------

------ because we three tables interconnected-------

ALTER TABLE order\_details

DROP CONSTRAINT order\_details\_productid\_fkey;

alter table order\_details

add constraint pk\_category

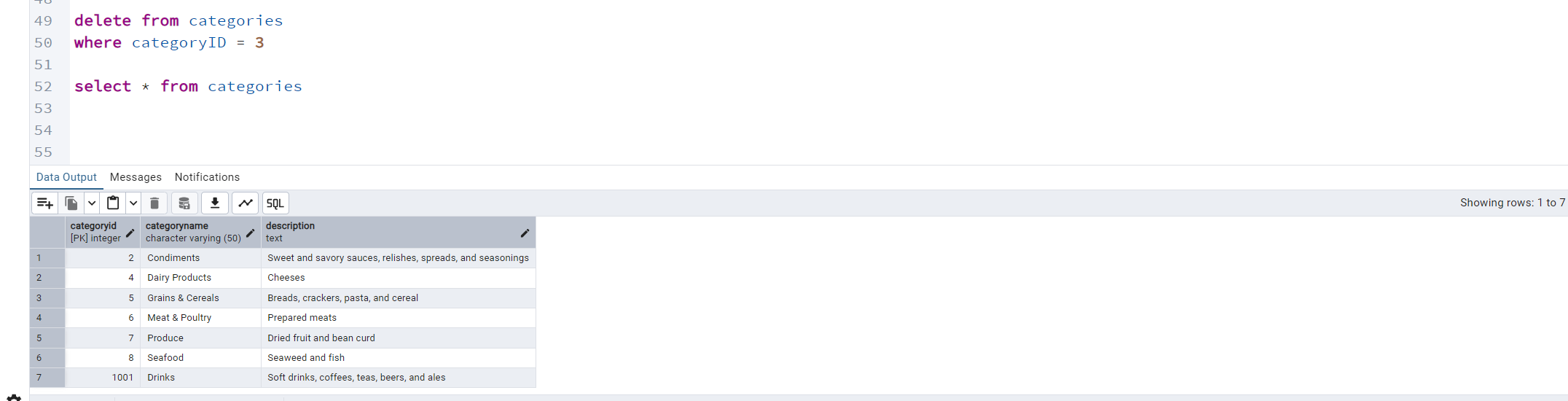
foreign key(productid)

references products(productid)

ON DELETE CASCADE

delete from categories

where categoryID = 3



select \* from categories

-- 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\_customerid\_fkey;

alter table orders

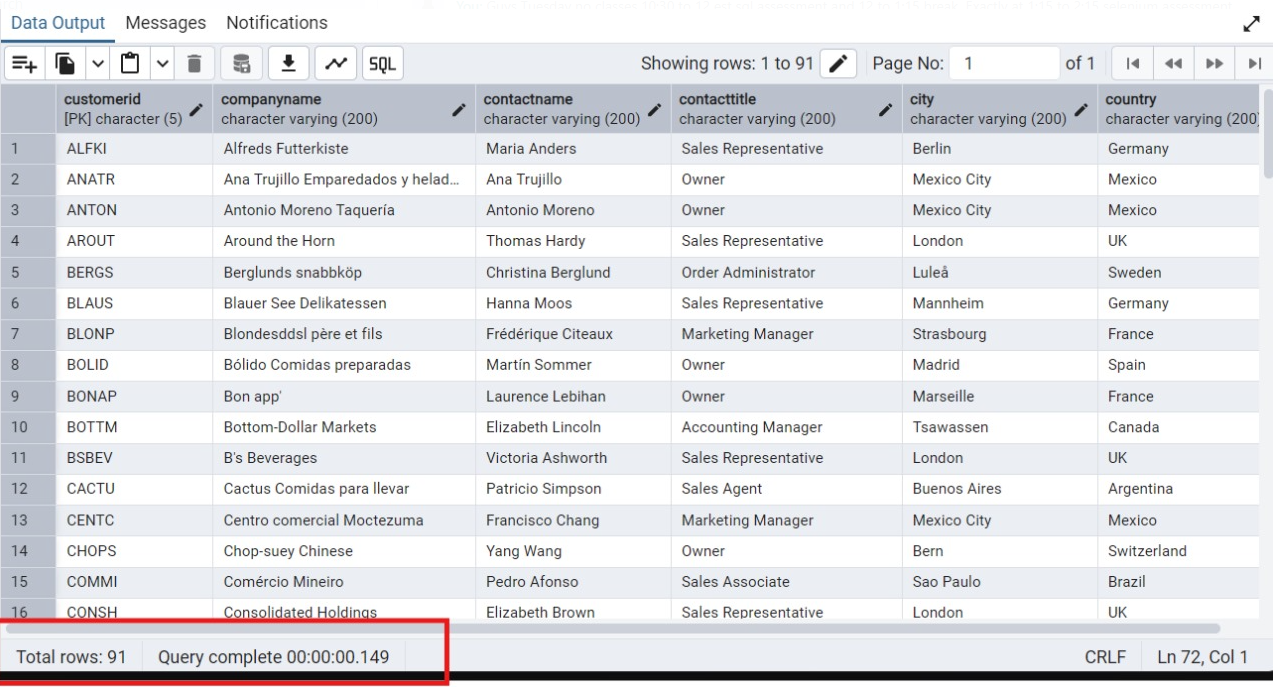
add constraint order\_foreignkey

foreign key (customerid)

references customers(customerid)

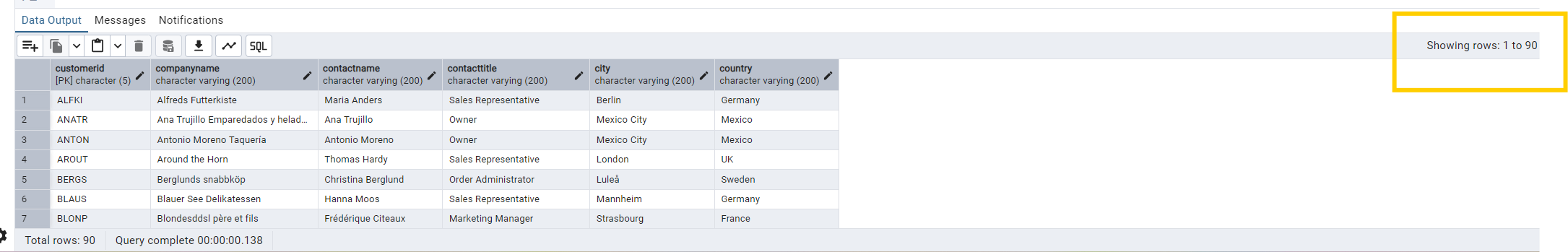
on delete set null

select \* from customers



delete from customers

where customerid = '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(productId,productname,quantityPerUnit,unitPrice,discontinued,categoryid)

values

(101,'White bread','5 boxes',13,'false',5),

(100,'Wheat bread','10 boxes',13,'false',5)

ON CONFLICT (productId)

DO UPDATE SET

productId = EXCLUDED.productId,

productname = EXCLUDED.productname,

quantityPerUnit = EXCLUDED.quantityPerUnit,

unitPrice = EXCLUDED.unitPrice,

discontinued = EXCLUDED.discontinued,

categoryid = EXCLUDED.categoryid;

select \* from products

A screenshot of a computer

AI-generated content may be incorrect.

---------you can insert unique ids once like above-----

--------- if we have duplicates like 2 hundreds the above one is updating to the below one---

INSERT INTO products(productId,productname,quantityPerUnit,unitPrice,discontinued,categoryid)

values(100,'Wheat bread','1',13,false,5),

ON CONFLICT (productId)

DO UPDATE SET

productId = EXCLUDED.productId,

productname = EXCLUDED.productname,

quantityPerUnit = EXCLUDED.quantityPerUnit,

unitPrice = EXCLUDED.unitPrice,

discontinued = EXCLUDED.discontinued,

categoryid = EXCLUDED.categoryid;

A screenshot of a computer

AI-generated content may be incorrect.

100 id got updated

-- 6) Write a MERGE query:

-- Create temp table with name: ‘updated\_products’ and insert values as below:

create temp table updated\_products(

productid int primary key,

productname varchar(50),

quantityPerUnit text,

unitprice decimal(10,2),

discontinued boolean,

categoryid int

)

select \* from updated\_products

insert into updated\_products

values(100,'Wheat bread','10',20,'true',5),

(101,'White bread','5 boxes',19.99,'false',5 ),

(102,'Midnight Mango Fizz','24-12 oz bottles',19,'false',1),

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

A screenshot of a computer

AI-generated content may be incorrect.

-- Update the price and discontinued status for from below table ‘updated\_products’ only if there are matching products and updated\_products .discontinued =0

------ to update we have to break relationship with connecting tables just dropping the foreign key relation------

Alter table products

drop constraint fk\_category

select \* from products

merge into products as p

using updated\_products u

ON p.productid = u.productid

when matched and p.discontinued='false' THEN

update set

unitprice =u.unitprice,

discontinued =u.discontinued

when not matched then

INSERT(productId,productname,quantityPerUnit,unitPrice,discontinued,categoryid)

VALUES(u.productId,u.productname,u.quantityPerUnit,u.unitPrice,u.discontinued,u.categoryid)

A screenshot of a computer

AI-generated content may be incorrect.

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

merge into products as p

using updated\_products u

ON p.productid = u.productid

when matched and u.discontinued='true' THEN

delete

select \* from products

select \* from updated\_products

A screenshot of a computer

AI-generated content may be incorrect.

100 row got deleted

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

insert into updated\_products

values(104,'dosa','8',202,'true',7),

(105,'chapathi','2 available',2.99,'false',8 )

select \* from updated\_products

MERGE INTO products p

USING updated\_products u

ON p.productid = u.productid

WHEN NOT MATCHED AND u.discontinued = 'false' THEN

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

VALUES (u.productid, u.productname, u.quantityperunit, u.unitprice, u.discontinued, u.categoryid);

select \* from products

A screenshot of a computer

AI-generated content may be incorrect.

104 value will not be added

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

select \* from orders

select \* from employees

select order\_id,first\_name || ' ' || last\_name as fullname from orders

inner join employees on

orders.employee\_id = employees.employee\_id

