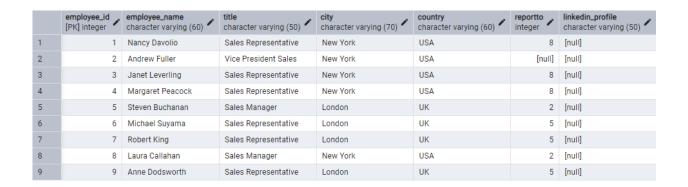
YAMUNA Day2 ASSIGNMENTS

1) Alter Table:

 Add a new column linkedin_profile to employees table to store LinkedIn URLs as varchar.

ALTER TABLE if exists employees

ADD COLUMN linkedIn_profile VARCHAR(50)



Change the linkedin_profile column data type from VARCHAR to TEXT.

ALTER TABLE employees ALTER COLUMN linkedin_profile SET DATA TYPE text

ALTER TABLE employees ALTER COLUMN linkedin_profile SET data TYPE text

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 39 msec.

Add unique, not null constraint to linkedin_profile

Alter table employees Add Constraint distinct_name UNIQUE (linkedin_profile);

Alter table employees Alter column linkedin_profile Set Not Null;

```
Alter table employees
Alter column linkedin_profile Set Not Null;

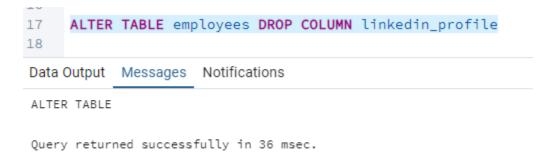
Data Output Messages Notifications

ERROR: column "linkedin_profile" of relation "employees" contains null values

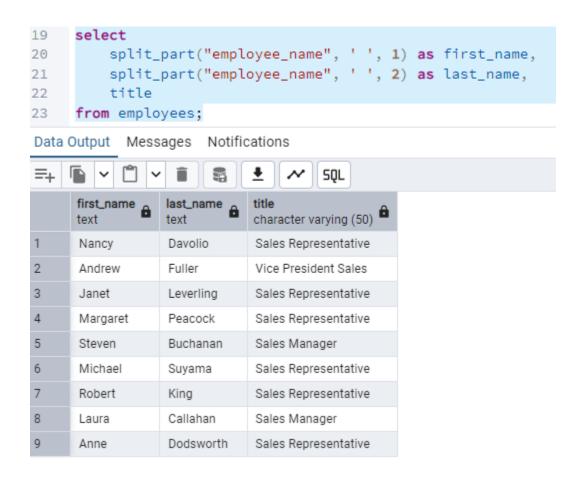
SQL state: 23502
```

Drop column linkedin_profile

ALTER TABLE employees DROP COLUMN linkedin_profile

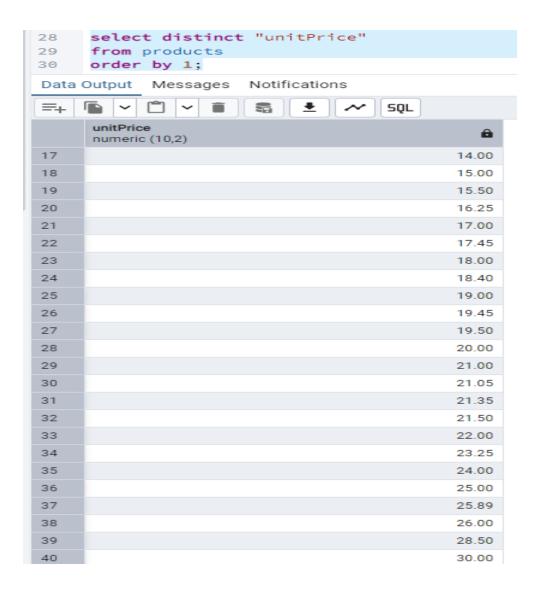


- 2) Querying (Select)
- Retrieve the employee name and title of all employees



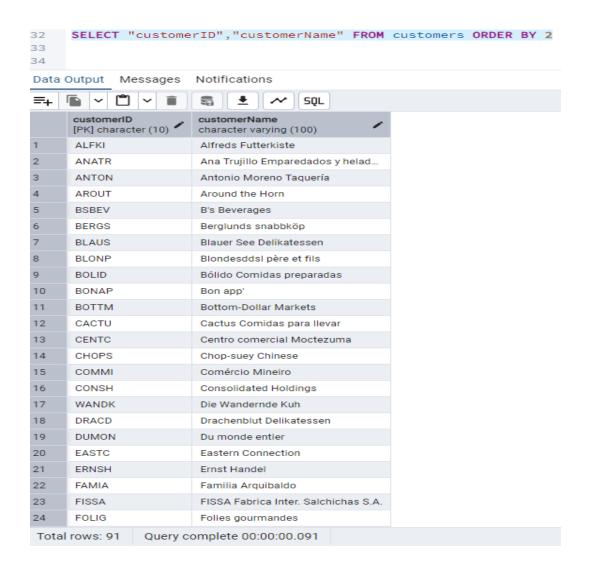
• Find all unique unit prices of products

select distinct "unitPrice" from products order by 1;



List all customers sorted by company name in ascending order

SELECT "customerID", "customerName" FROM customers ORDER BY 2



• Display product name and unit price, but rename the unit_price column as price_in_usd

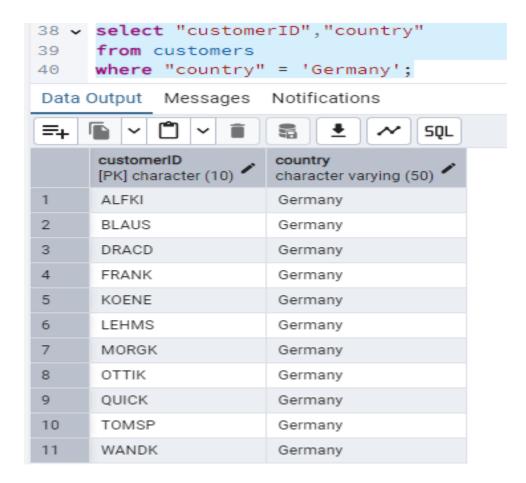
select "productName", "unitPrice" as price_in_usd from products;

34 35	<pre>select "productName", " from products;</pre>	unitPrice" as price_in_usd				
Data Output Messages Notifications						
=+		~ SQL				
	productName character varying (60)	price_in_usd numeric (10,2)				
1	Chai	18.00				
2	Chang	19.00				
3	Aniseed Syrup	10.00				
4	Chef Anton's Cajun Seasoning	22.00				
5	Chef Anton's Gumbo Mix	21.35				
6	Grandma's Boysenberry Spread	25.00				
7	Uncle Bob's Organic Dried Pears	30.00				
8	Northwoods Cranberry Sauce	40.00				
9	Mishi Kobe Niku	97.00				
10	Ikura	31.00				
11	Queso Cabrales	21.00				
12	Queso Manchego La Pastora	38.00				
13	Konbu	6.00				
14	Tofu	23.25				
15	Genen Shouyu	15.50				
16	Pavlova	17.45				
17	Alice Mutton	39.00				
18	Carnarvon Tigers	62.50				
19	Teatime Chocolate Biscuits	9.20				
20	Sir Rodney's Marmalade	81.00				
21	Sir Rodney's Scones	10.00				
22	Gustaf's Knackebröd	21.00				
23	Tunnbröd	9.00				
24	Guarana Fantastica	4.50				

3) Filtering

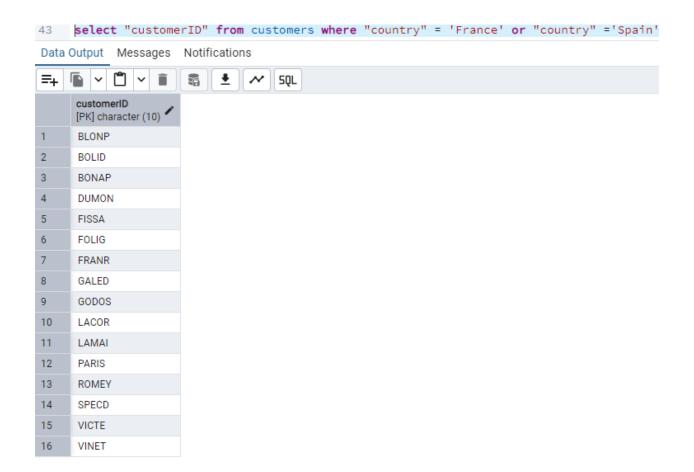
• Get all customers from Germany.

Select "customerID" from customers where "country" = 'Germany';



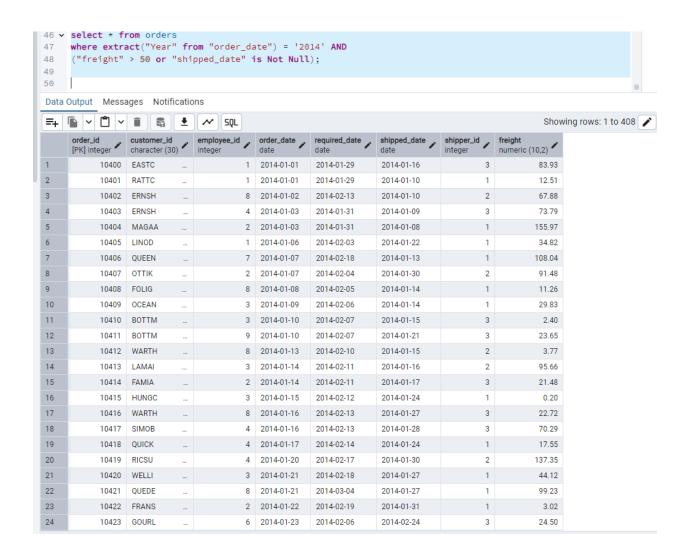
• Find all customers from France or Spain

select "customerID" from customers where "country" = 'France' or "country" = 'Spain';



• Retrieve all orders placed in 2014 (based on order_date), and either have freight greater than 50 or the shipped date available (i.e., non-NULL) (Hint: EXTRACT(YEAR FROM order_date))

select * from orders where extract("Year" from "orderDate") = '2014' and ("freight" > 50 or "shippedDate" is Not Null);



4) Filtering ● Retrieve the product_id, product_name, and unit_price of products where the unit_price is greater than 15.

select "productID", "productName", "unitPrice" from products where "unitPrice" > 15 order by "unitPrice";

```
52 v select "productio", "productName", "unitrrice"
      from products
53
      where "unitPrice" > 15
54
      order by "unitPrice";
55
Data Output Messages Notifications
=+
                                          SQL
                                                 unitPrice
      productID
                    productName
                                                 numeric (10,2)
      [PK] integer
                    character varying (60)
1
                15 Genen Shouyu
                                                           15.50
2
                    Valkoinen suklaa
                                                           16.25
                50
3
                66 Louisiana Hot Spiced Okra
                                                           17.00
4
                    Pavlova
                                                           17.45
                16
5
                39 Chartreuse verte
                                                           18.00
6
                 1
                    Chai
                                                           18.00
7
                76 Lakkaliköri
                                                           18.00
8
                35 Steeleye Stout
                                                           18.00
                40 Boston Crab Meat
9
                                                           18.40
10
                36
                    Inlagd Sill
                                                           19.00
11
                 2 Chang
                                                           19.00
12
                    Gula Malacca
                                                           19.45
                44
13
                   Ravioli Angelo
                                                           19.50
                57
```

20.00

21.00

21.00

21.05

21.35

21.50

22.00

23.25

Maxilaku

11 Oueso Cabrales

71 Flotemysost

Gustaf's Knackebröd

65 Louisiana Fiery Hot Pepper Sauce

Chef Anton's Cajun Seasoning

Chef Anton's Gumbo Mix

49

22

5

4

14 Tofu

14

15

16

17

18

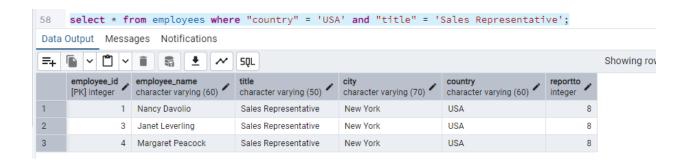
19

20

21

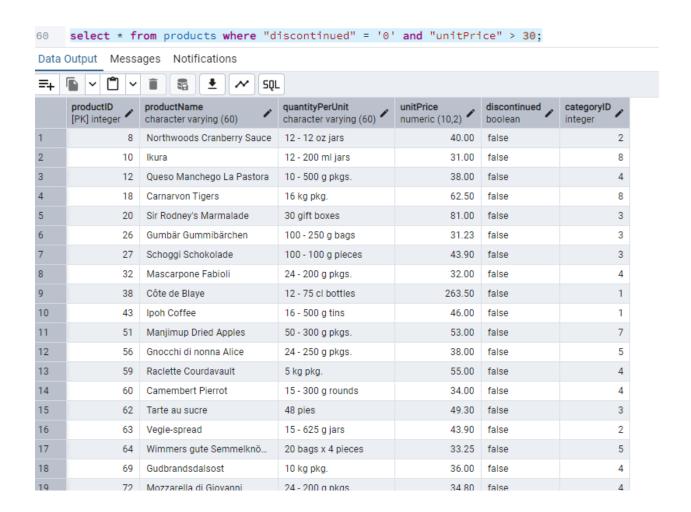
• List all employees who are in the USA and have the title "Sales Representative".

select * from employees where "country" = 'USA' and "title" = 'Sales Representative';



• Retrieve all products that are not discontinued and priced greater than 30.

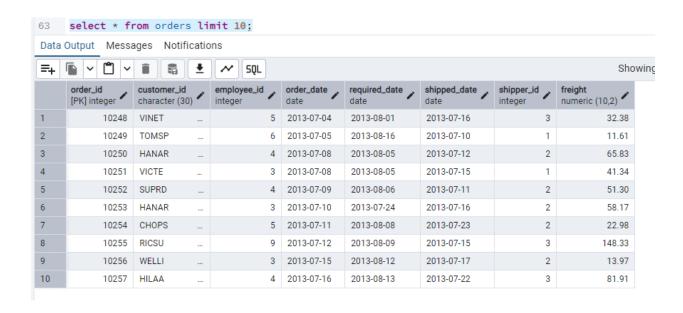
select * from products where "discontinued" = '0' and "unitPrice" > 30;



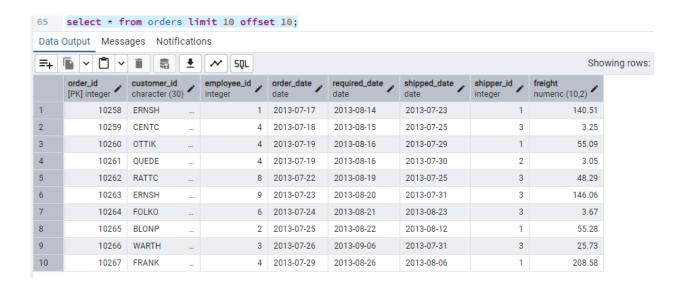
5) LIMIT/FETCH

• Retrieve the first 10 orders from the orders table.

select * from orders limit 10;

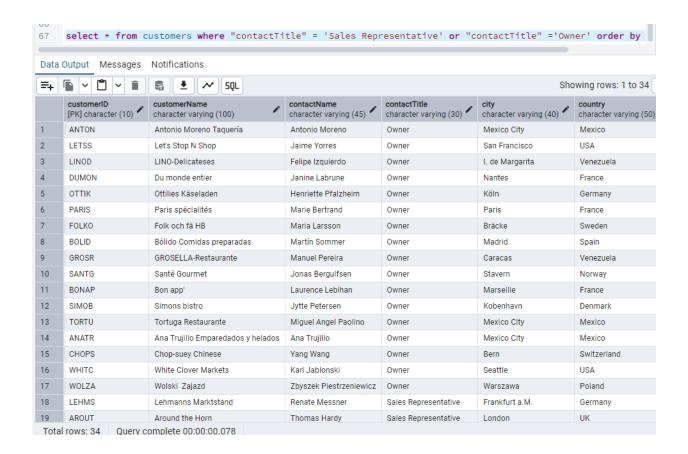


■ Retrieve orders starting from the 11th order, fetching 10 rows (i.e., fetch rows 11-20). select * from orders limit 10 offset 10;



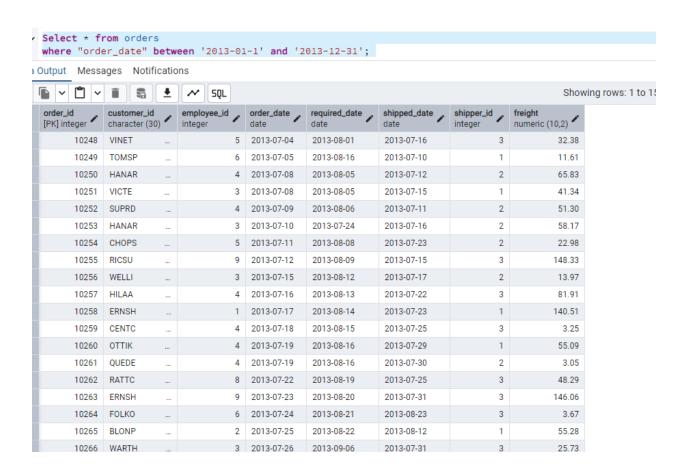
6) Filtering (IN, BETWEEN) ● List all customers who are either Sales Representative or Owner

select * from customers where "contactTitle" = 'Sales Representative' or "contactTitle" = 'Owner' order by "contactTitle";



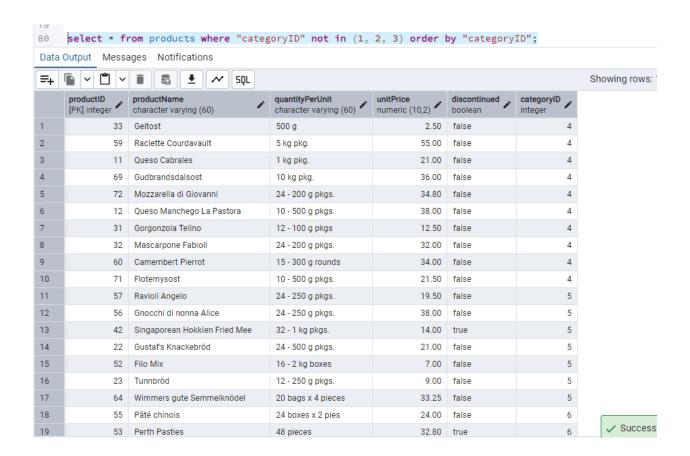
• Retrieve orders placed between January 1, 2013, and December 31, 2013.

select * from orders where "orderDate" between '2013-01-1' and '2013-12-31';



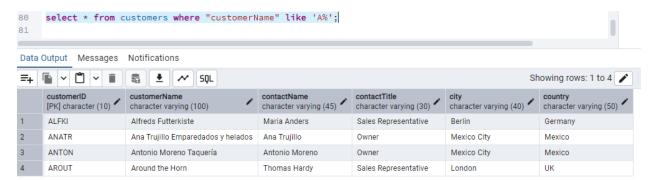
7) Filtering • List all products whose category_id is not 1, 2, or 3.

select * from products where "categoryID" not in (1, 2, 3) order by "categoryID";



• Find customers whose company name starts with "A".

select * from customers where "companyName" like 'A%';



8) INSERT into orders table: Task:

Add a new order to the orders table with the following details:

Order ID: 11078

Customer ID: ALFKI

Employee ID: 5 Order Date: 2025-04-23

Required Date: 2025-04-30

Shipped Date: 2025-04-25

shipperID:2

Freight: 45.50

Insert into orders values(11078,'ALFKI', 5, '2025-04-23', '2025-04-30','2025-04-25', 2, 45.50)

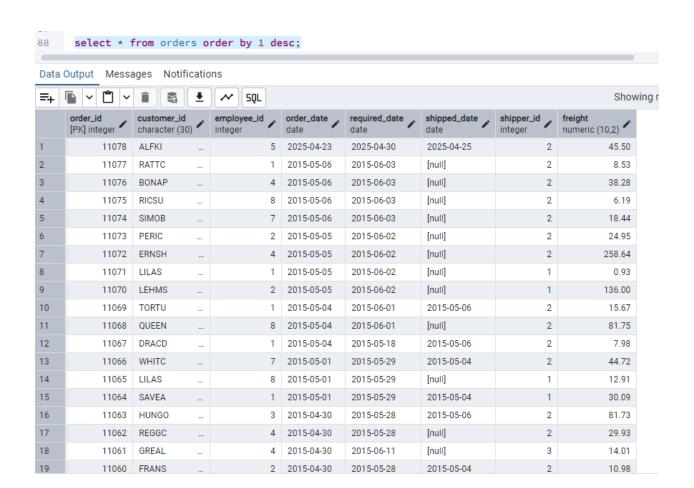
```
Insert into orders values(11078, 'ALFKI', 5, '2025-04-23', '2025-04-30', '2025-04-25', 2, 45.50)

Data Output Messages Notifications

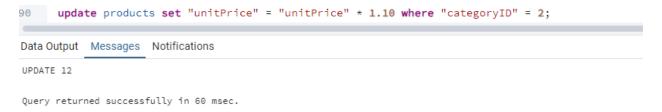
INSERT 0 1

Query returned successfully in 44 msec.
```

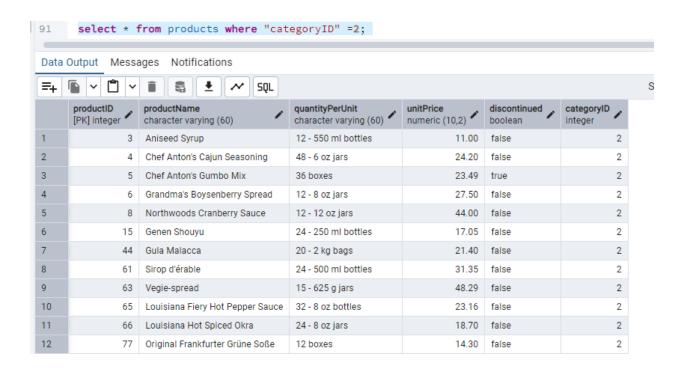
select * from orders order by 1 desc;



9) Increase(Update) the unit price of all products in category_id =2 by 10%.(HINT: unit_price = unit_price * 1.10)



select * from products where "categoryID" =2;



10) Sample Northwind database:

- v 🍔 northwind
 - > 🚱 Casts
 - > 💖 Catalogs
 - > C Event Triggers
 - > f Extensions
 - > 🥞 Foreign Data Wrappers
 - > 🤤 Languages
 - > 🖒 Publications
 - - → opublic
 - > 🖟 Aggregates
 - > 🔠 Collations
 - > 🏠 Domains
 - > 🔓 FTS Configurations
 - > TS Dictionaries
 - > Aa FTS Parsers
 - > @ FTS Templates
 - > 📑 Foreign Tables
 - > (ii) Functions
 - > @ Materialized Views
 - > 🐁 Operators
 - > (Procedures
 - > 1...3 Sequences
 - √ Imale Tables (14)
 - > == categories
 - > == customer_customer_demo
 - > == customer_demographics
 - > == customers
 - > == employee_territories
 - > ## employees
 - > == order_details
 - > III orders
 - > ## products
 - > 🛗 region
 - > 🔠 shippers
 - > == suppliers
 - > = territories
 - > 🔠 us_states