Day 2 - Screenshot

**Day2**

1)      Alter Table:

* Add a new column linkedin\_profile to employees table to store LinkedIn URLs as varchar.

ALTER TABLE employees

ADD COLUMN linkedin\_profile VARCHAR;

A screen shot of a computer

AI-generated content may be incorrect.

----Checking table after every alteration

SELECT \* FROM EMPLOYEES

A screenshot of a computer

AI-generated content may be incorrect.

* Change the linkedin\_profile column data type from VARCHAR to TEXT.

ALTER TABLE employees

ALTER COLUMN linkedin\_profile TYPE TEXT;

A screenshot of a computer

AI-generated content may be incorrect.

----Checking table after every alteration

SELECT \* FROM EMPLOYEES

A screenshot of a computer

AI-generated content may be incorrect.

* Add unique, not null constraint to linkedin\_profile

----Adding Unique Linkedin profile

UPDATE employees

SET linkedin\_profile = 'https://linkedin.com/in/user\_' || employeeid

A screenshot of a computer

AI-generated content may be incorrect.

----Checking table after every alteration

SELECT \* FROM EMPLOYEES

A screenshot of a computer

AI-generated content may be incorrect.

ALTER TABLE employees

ALTER COLUMN linkedin\_profile SET NOT NULL,

ADD CONSTRAINT unique\_linkedin\_profile UNIQUE (linkedin\_profile);

A screenshot of a computer

AI-generated content may be incorrect.

----Checking table after every alteration

SELECT \* FROM EMPLOYEES

A screenshot of a computer

AI-generated content may be incorrect.

* Drop column linkedin\_profile

ALTER TABLE employees

DROP COLUMN linkedin\_profile;

A screenshot of a computer

AI-generated content may be incorrect.

----Checking table after every alteration

SELECT \* FROM EMPLOYEES

A screenshot of a computer

AI-generated content may be incorrect.

2)      Querying (Select)

* Retrieve the first name, last name, and title of all employees

SELECT employeename, title

FROM employees;

A screenshot of a computer

AI-generated content may be incorrect.

* Find all unique unit prices of products

SELECT DISTINCT unitprice

FROM products;

A screenshot of a computer

AI-generated content may be incorrect.

* List all customers sorted by company name in ascending order

SELECT \*

FROM customers

ORDER BY companyname ASC;

A screenshot of a computer

AI-generated content may be incorrect.

* 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;

A screenshot of a computer

AI-generated content may be incorrect.

3)      Filtering

* Get all customers from Germany.

SELECT \*

FROM customers

WHERE country = 'Germany';

A screenshot of a computer

AI-generated content may be incorrect.

* Find all customers from France or Spain

SELECT \*

FROM customers

WHERE country IN ('France', 'Spain')

ORDER BY country ASC ;

A screenshot of a computer

AI-generated content may be incorrect.

* Retrieve all orders placed in 1997 (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);

A screenshot of a computer screen

AI-generated content may be incorrect.

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 asc;

A screenshot of a computer

AI-generated content may be incorrect.

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

SELECT \*

FROM employees

WHERE country = 'USA'

AND title = 'Sales Representative';

A screenshot of a computer

AI-generated content may be incorrect.

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

SELECT \*

FROM products

WHERE discontinued = FALSE

AND unitprice > 30 order by unitprice asc;

A screenshot of a data output

AI-generated content may be incorrect.

5)      LIMIT/FETCH

* Retrieve the first 10 orders from the orders table.

SELECT \*

FROM orders

LIMIT 10;

A screenshot of a computer

AI-generated content may be incorrect.

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

SELECT \*

FROM orders

LIMIT 10 OFFSET 10;

A screenshot of a computer

AI-generated content may be incorrect.

6)      Filtering (IN, BETWEEN)

* List all customers who are either Sales Representative, Owner

SELECT \*

FROM customers

WHERE contactTitle IN ('Sales Representative', 'Owner')

order by contactTitle asc ;

A screenshot of a computer

AI-generated content may be incorrect.

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

SELECT \*

FROM orders

WHERE orderDate BETWEEN '2013-01-01' AND '2013-12-31'

order by orderdate asc ;

A screenshot of a computer

AI-generated content may be incorrect.

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 asc ;

A screenshot of a computer

AI-generated content may be incorrect.

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

SELECT \*

FROM customers

WHERE companyName LIKE 'A%'

order by companyname asc ;

A screenshot of a computer

AI-generated content may be incorrect.

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

select count(\*) from

orders where customerID = 'ALFKI'; -- there were 6 rows before insert , after it increased to 7 / cross checking

INSERT INTO orders (orderID, customerID, employeeID, orderDate, requiredDate, shippedDate, shipperID, freight)

VALUES (11078, 'ALFKI', 5, '2025-04-23', '2025-04-30', '2025-04-25', 2, 45.50);

A screenshot of a computer

AI-generated content may be incorrect.

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

order by unitPrice asc ;

UPDATE products

SET unitPrice = unitPrice \* 1.10

WHERE categoryID = 2;

A screenshot of a computer

AI-generated content may be incorrect.

10) Sample Northwind database:

Download

 Download northwind.sql from below link into your local. Sign in to Git first <https://github.com/pthom/northwind_psql>

 Manually Create the database using pgAdmin:

 Right-click on "Databases" → Create → Database

Give name as ‘northwind’ (all small letters)

Click ‘Save’

A screenshot of a computer

AI-generated content may be incorrect.

Import database:

 Open pgAdmin and connect to your server

  Select the database  ‘northwind’

  Right Click-> Query tool.

  Click the folder icon to open your northwind.sql file

 Press F5 or click the Execute button.

  You will see total 14 tables loaded

  Databases → your database → Schemas → public → Tables

