1) Alter Table:

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

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

Add unique, not null constraint to linkedin\_profile

Drop column linkedin\_profile

--To add new column linkedin\_profile as VARCHAR

ALTER TABLE employees

ADD COLUMN linkedin\_profile VARCHAR(200);

Select \* From employees;

-- To change linkedin\_profile column data type from VARCHAR to TEXT

ALTER TABLE employees

ALTER COLUMN linkedin\_profile TYPE TEXT;

Select \* From employees;

-- to add unique, not null constraint to linkedin\_profile

ALTER TABLE employees

ALTER COLUMN linkedin\_profile SET NOT NULL;

ALTER TABLE employees

ADD CONSTRAINT linkedin\_profile\_unique UNIQUE (linkedin\_profile);

Select \* From employees;

-- To drop column linkedin\_profile from employees table

ALTER TABLE employees

DROP COLUMN linkedin\_profile;

Select \* From employees;

-----------------------------------------------------------------------------------------------------------------

2) Querying (Select)

Retrieve the employee name and title of all employees

Find all unique unit prices of products

List all customers sorted by company name in ascending order

Display product name and unit price, but rename the unit\_price column as price\_in\_usd

- 1. Retrieve the employee name and title of all employees

SELECT

employeename,

title

FROM employees;

-- 2. Find all unique unit prices of products

SELECT DISTINCT

unitprice

FROM products;

-- 3. List all customers sorted by company name in ascending order

SELECT

\*

FROM customers

ORDER BY companyname ASC;

-- 4. 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;

-----------------------------------------------------------------------------------------------------------------

3) Filtering

Get all customers from Germany.

Find all customers from France or 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))

-- 1. Get all customers from Germany

SELECT \*

FROM customers

WHERE country = 'Germany';

-- 2. Find all customers from France or Spain

SELECT \*

FROM customers

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

-- 3. Retrieve all orders placed in 2014 with freight > 50 or shipped\_date not null

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.

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

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

-- 1. 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;

-- 2. 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';

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

SELECT

productid,

productname,

unitprice

FROM products

WHERE discontinued = 0

AND unit\_price > 30;

-----------------------------------------------------------------------------------------------------------------

5) LIMIT/FETCH

Retrieve the first 10 orders from the orders table.

Retrieve orders starting from the 11th order,

-- 1. Retrieve the first 10 orders from the orders table

SELECT \*

FROM orders

ORDER BY "orderID"

LIMIT 10;

-- 2. Retrieve orders 11–20 (i.e., starting from the 11th row, fetch 10 rows)

SELECT \*

FROM orders

ORDER BY "orderID"

OFFSET 10

LIMIT 10;

-----------------------------------------------------------------------------------------------------------------

6) Filtering (IN, BETWEEN)

List all customers who are either Sales Representative, Owner

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

--1. List all customers whose contact title is either 'Sales Representative' or 'Owner'

SELECT \*

FROM customers

WHERE contacttitle IN ('Sales Representative', 'Owner');

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

SELECT \*

FROM orders

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

-----------------------------------------------------------------------------------------------------------------

7) Filtering

List all products whose category\_id is not 1, 2, or 3.

Find customers whose company name starts with "A".

-- 1. List all products whose category\_id is not 1, 2, or 3

SELECT

productID,

productname,

categoryID

FROM products

WHERE categoryID NOT IN (1, 2, 3);

-- 2. Find customers whose company name starts with "A"

SELECT

customerid,

companyname,

contactname,

country

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

SELECT \* from orders;

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

);

-----------------------------------------------------------------------------------------------------------------

9) Increase(Update) the unit price of all products in category\_id =2 by 10%.

(HINT: unit\_price =unit\_price \* 1.10)

UPDATE products

SET unitprice = unitprice \* 1.10

WHERE categoryid = 2;

SELECT \* FROM products;

-----------------------------------------------------------------------------------------------------------------