--Day 8

--1.Create view vw\_updatable\_products (use same query whatever I used in the training)

--Try updating view with below query and see if the product table also gets updated.

--Update query:

--UPDATE updatable\_products SET unit\_price = unit\_price \* 1.1 WHERE units\_in\_stock < 10;

Create View vw\_updatable\_products As

Select product\_id,

product\_name,

unit\_price,

units\_in\_stock,

discontinued

From products

Where discontinued = 0;

A screenshot of a computer

AI-generated content may be incorrect.

Update vw\_updatable\_products

Set unit\_price = unit\_price \* 1.1

Where units\_in\_stock < 10;

A screenshot of a message

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

--2.Transaction:

--Update the product price for products by 10% in category id=1

--Try COMMIT and ROLLBACK and observe what happens.

Begin;

Update public.products

Set unit\_price=unit\_price \*1.10

Where category\_id=2

Commit;

Rollback;

select \* from public.products

Where category\_id=2

A screenshot of a computer

AI-generated content may be incorrect.

--3.Create a regular view which will have below details (Need to do joins):

--Employee\_id,

--Employee\_full\_name,

--Title,

--Territory\_id,-employeeid

--territory\_description,territoryId

--region\_description-region id

Create View employee\_View As

Select

emp.employee\_id,

emp.first\_name||' '||emp.last\_name As employee\_Name,

emp.title,

et.territory\_id,

t.territory\_description,

rgn.region\_description

From employees emp

Join employee\_territories et

On emp.employee\_id = et.employee\_id

Join territories t

On et.territory\_id = t.territory\_id

Join region rgn

On t.region\_id = rgn.region\_id;

A screenshot of a computer message

AI-generated content may be incorrect.

--4.Create a recursive CTE based on Employee Hierarchy

**With recursive cte\_employee\_hierarchy As (**

**-- Anchor member: top-level employees**

SELECT

employee\_id,

first\_name,

last\_name,

title,

reports\_to,

0 As level

From employees emp

Where reports\_to Is null

Union All

-- Recursive member: employees who report to others

SELECT

emp.employee\_id,

emp.first\_name,

emp.last\_name,

emp.title,

emp.reports\_to,

eh.level+1

From employees emp

Join cte\_employee\_hierarchy eh

On eh.employee\_id = emp.reports\_to

)

SELECT

level,

employee\_id,

first\_name||' '||last\_name As employee\_name

From cte\_employee\_hierarchy

Order By level, employee\_id;

A screenshot of a computer

AI-generated content may be incorrect.