**Assignment Day 8**

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

CREATE VIEW vw\_updatable\_products AS

SELECT product\_id, product\_name, unit\_price, units\_in\_stock

FROM products;

A screenshot of a computer

AI-generated content may be incorrect.

Try updating view with below query and

Update query:

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

A screenshot of a computer

AI-generated content may be incorrect.

see if the product table also gets updated

SELECT product\_id, product\_name, unit\_price, units\_in\_stock

FROM products

where units\_in\_stock < 10

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.

COMMIT

BEGIN transaction;

update products

set unit\_price = unit\_price\*1.10

where

category\_id = 1;

COMMIT;A screenshot of a computer

AI-generated content may be incorrect.

ROLLBACK

Rollback was successful but no changes in the price since it was performed after commit

A white background with blue and green text

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,

territory\_description,

region\_description

A white background with blue text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Select \* from employee\_territory\_details

A screenshot of a computer

AI-generated content may be incorrect.

4. Create a recursive CTE based on Employee Hierarchy

WITH RECURSIVE cte\_employee\_hierarchy AS(

----Manager at the Highest Level

select

employee\_id,

first\_name,

last\_name,

reports\_to,

0 As level

from employees e

where reports\_to is NULL

Union ALL

----Recursive query for employees reporting to higher managers

select

e.employee\_id,

e.first\_name,

e.last\_name,

e.reports\_to,

c.level + 1 as level

from employees e

Join cte\_employee\_hierarchy c on e.reports\_to = c.employee\_id

)

select

employee\_id,

first\_name||' '||last\_name as employee\_name,

reports\_to,

level

from cte\_employee\_hierarchy

order by level,employee\_id

A screenshot of a computer

AI-generated content may be incorrect.