**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;

--selecting the records from view for the below condition

select \* from vw\_updatable\_products WHERE units\_in\_stock < 10;

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;

--checking if the products table is also updated

select \* from products WHERE units\_in\_stock < 10 and discontinued = 0;

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.

select \* from products where category\_id=1;

A screenshot of a computer

AI-generated content may be incorrect.

BEGIN;

update products

set unit\_price = unit\_price \* 1.10

where category\_id=1;

select \* from products where category\_id=1;

A screenshot of a computer

AI-generated content may be incorrect.

ROLLBACK;

select \* from products where category\_id=1;

A screenshot of a computer

AI-generated content may be incorrect.

commit;

select \* from products where category\_id=1;

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,

territory\_description,

region\_description

create view vw\_employee\_territory\_region as

select e.Employee\_id,

e.first\_name || ' ' || e.last\_name as Employee\_full\_name,

e.title,

t.territory\_id,

t.territory\_description,

r.region\_description

from employees e

join employee\_territories et on e.employee\_id = et.employee\_id

join territories t on t.territory\_id = et.territory\_id

join region r on r.region\_id = t.region\_id;

select \* from vw\_employee\_territory\_region;

A screenshot of a computer

AI-generated content may be incorrect.