**Day 10 Assignment**

1.Write a function to Calculate the total stock value for a given category:

(Stock value=ROUND(SUM(unit\_price \* units\_in\_stock)::DECIMAL, 2)

Return data type is DECIMAL(10,2)

CREATE OR REPLACE FUNCTION get\_category\_stock\_value(p\_category\_id INT)

RETURNS DECIMAL(10,2)

LANGUAGE plpgsql

AS $$

DECLARE

v\_total\_value DECIMAL(10,2);

BEGIN

SELECT ROUND(SUM(unit\_price \* units\_in\_stock)::DECIMAL, 2)

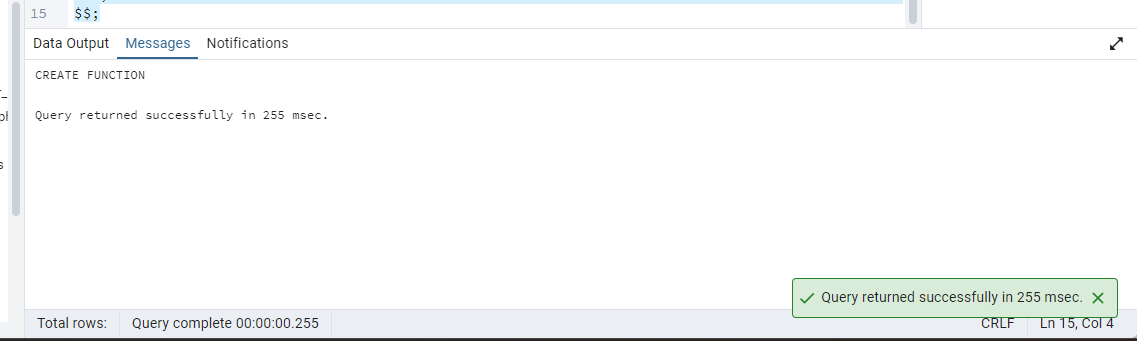
INTO v\_total\_value

FROM products

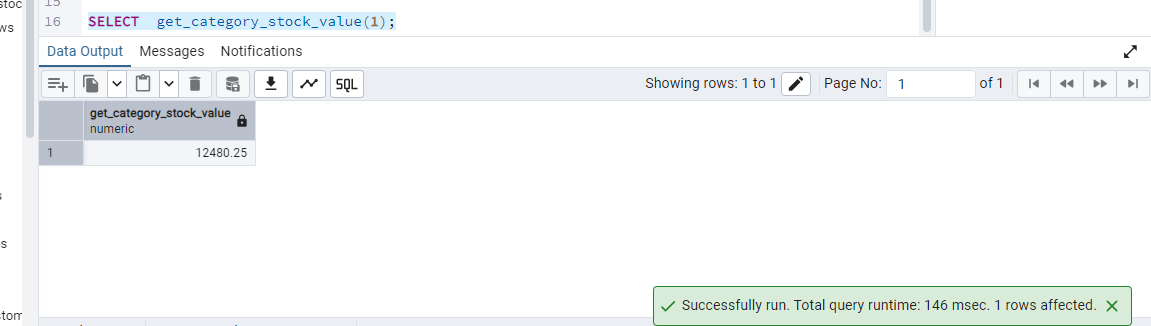
WHERE category\_id = p\_category\_id;

RETURN COALESCE(v\_total\_value, 0.00);

END;



SELECT get\_category\_stock\_value(1);



2.

--2. Try writing a cursor query which I executed in the training.

CREATE OR REPLACE PROCEDURE update\_price\_with\_cursor()

LANGUAGE PLPGSQL

AS $$

DECLARE

product\_cursor CURSOR FOR

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

FROM products

WHERE discontinued = 0;

product\_record RECORD;

n\_new\_price DECIMAL(10,2);

BEGIN

OPEN product\_cursor;

LOOP

FETCH product\_cursor INTO product\_record;

EXIT WHEN NOT FOUND;

IF product\_record.units\_in\_stock < 10 THEN

n\_new\_price := product\_record.unit\_price\*1.1;

ELSE n\_new\_price := product\_record.unit\_price\*0.95;

END IF;

UPDATE products

SET unit\_price = ROUND(n\_new\_price,2)

WHERE product\_id = product\_record.product\_id;

RAISE NOTICE 'Updated % price from % to %',

product\_record.product\_name,

product\_record.unit\_price,

n\_new\_price;

END LOOP;

CLOSE product\_cursor;

END;

$$;

--Call the cursor

CALL update\_price\_with\_cursor();

