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\_total\_cost(p\_product\_id INT)

RETURNS DECIMAL(10,2)

LANGUAGE plpgsql

AS $$

DECLARE

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

BEGIN

SELECT

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

INTO v\_stock\_value

FROM products p

WHERE p.Product\_id = p\_product\_id ;

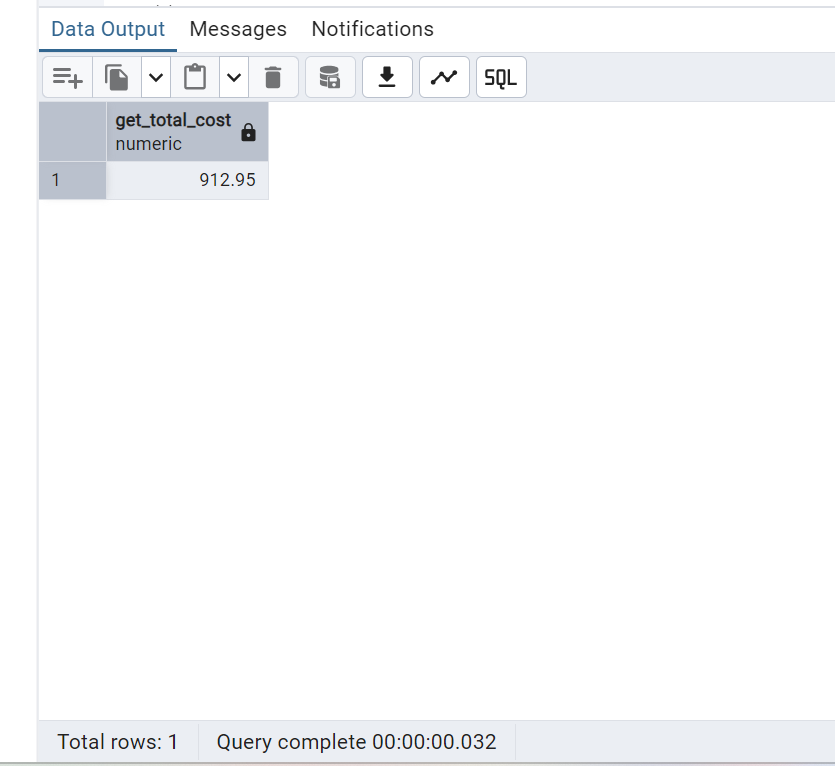
RETURN v\_stock\_value;

END;

$$;

---call functions

SELECT get\_total\_cost (10);



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

Create OR REPLACE procedure update\_price\_with\_curson()

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;

v\_new\_price Decimal(10,2);

BEGIN

OPEN product\_cursor;

LOOP

--fetch the next row

FETCH product\_cursor INTO product\_record;

---exit when no more rows to fetch

EXIT WHEN NOT FOUND;

---calculate new unit price

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

v\_new\_price := product\_record.unit\_price \* 1.1; --10% increase

ELSE

v\_new\_price := product\_record.unit\_price \* 0.95 ;---5% Decrese

END IF;

---update product

UPDATE products

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

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

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

product\_record.product\_name,

product\_record.unit\_price,

v\_new\_price;

END LOOP;

CLOSE product\_cursor;

END;

$$;

--To execute

CALL update\_price\_with\_curson();

