**DAY 10**

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)**

SELECT \* FROM categories

SELECT \* FROM products

CREATE OR REPLACE FUNCTION total\_stockvalue(p\_category\_id INT)

RETURNS DECIMAL(10,2)

LANGUAGE plpgsql

AS $$

DECLARE

v\_stockvalue DECIMAL(10,2);

BEGIN

IF NOT EXISTS (SELECT 1 FROM products WHERE category\_id = p\_category\_id) THEN

RAISE EXCEPTION 'Category id % does not exist',p\_category\_id;

RETURN 0;

END IF;

SELECT

COALESCE(ROUND(SUM(unit\_price \* units\_in\_stock)::DECIMAL,2),0)

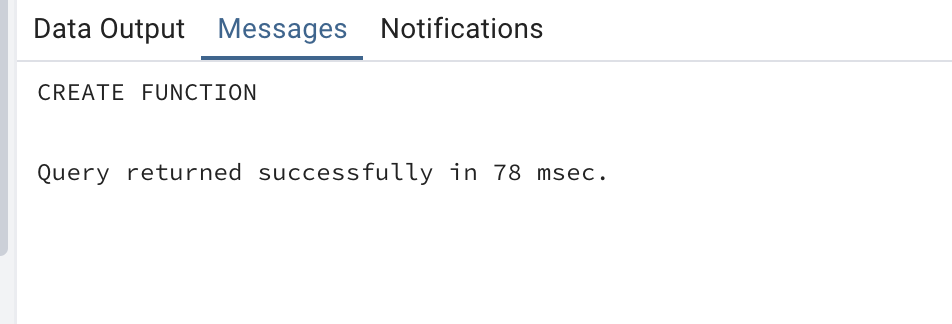
INTO v\_stockvalue

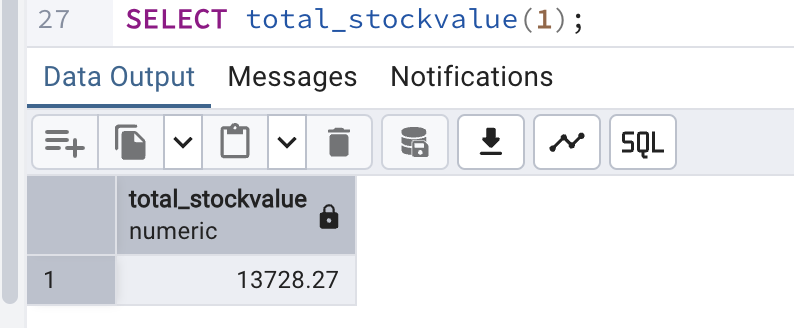
FROM products p

WHERE p.category\_id = p\_category\_id;

RETURN v\_stockvalue;

END;

$$;



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

CREATE OR REPLACE PROCEDURE update\_prices\_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;

v\_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

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

ELSE

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

END IF;

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;

$$;