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)

Query:

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

RETURNS DECIMAL(10,2) AS $$

DECLARE

stock\_value DECIMAL(10,2);

BEGIN

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

INTO stock\_value

FROM products

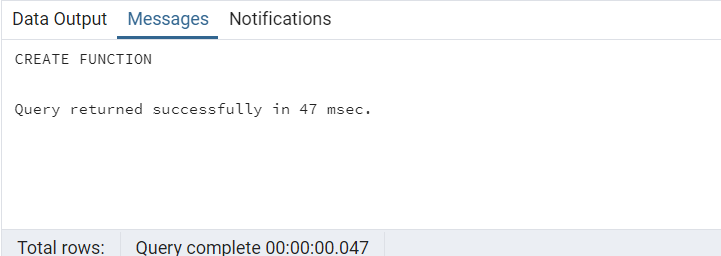
WHERE category\_id = p\_category\_id;

RETURN COALESCE(stock\_value, 0); -- Return 0 if no rows found

END;

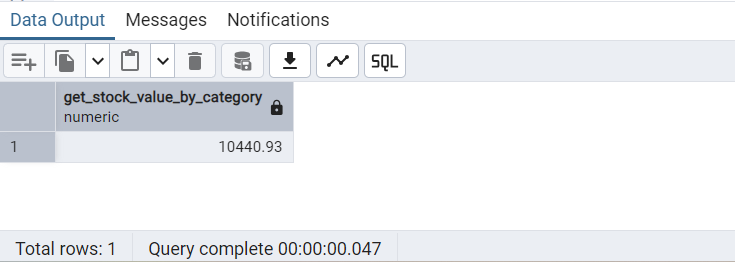
$$ LANGUAGE plpgsql;

Output:



SELECT get\_stock\_value\_by\_category(3);

Output:



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

Query:

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;

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

UPDATE products

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

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

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

product\_record.product\_name,product\_record.unit\_price,v\_new\_price;

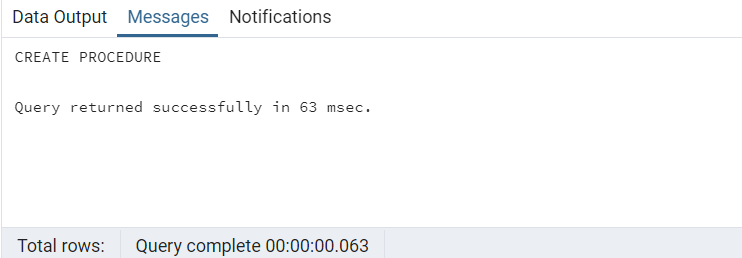
END LOOP;

CLOSE product\_cursor;

END;

$$;

Output:



--Calling the Procedure:

CALL update\_prices\_with\_cursor();

OUTPUT:

