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 products;

create function calculate\_stockvalue(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

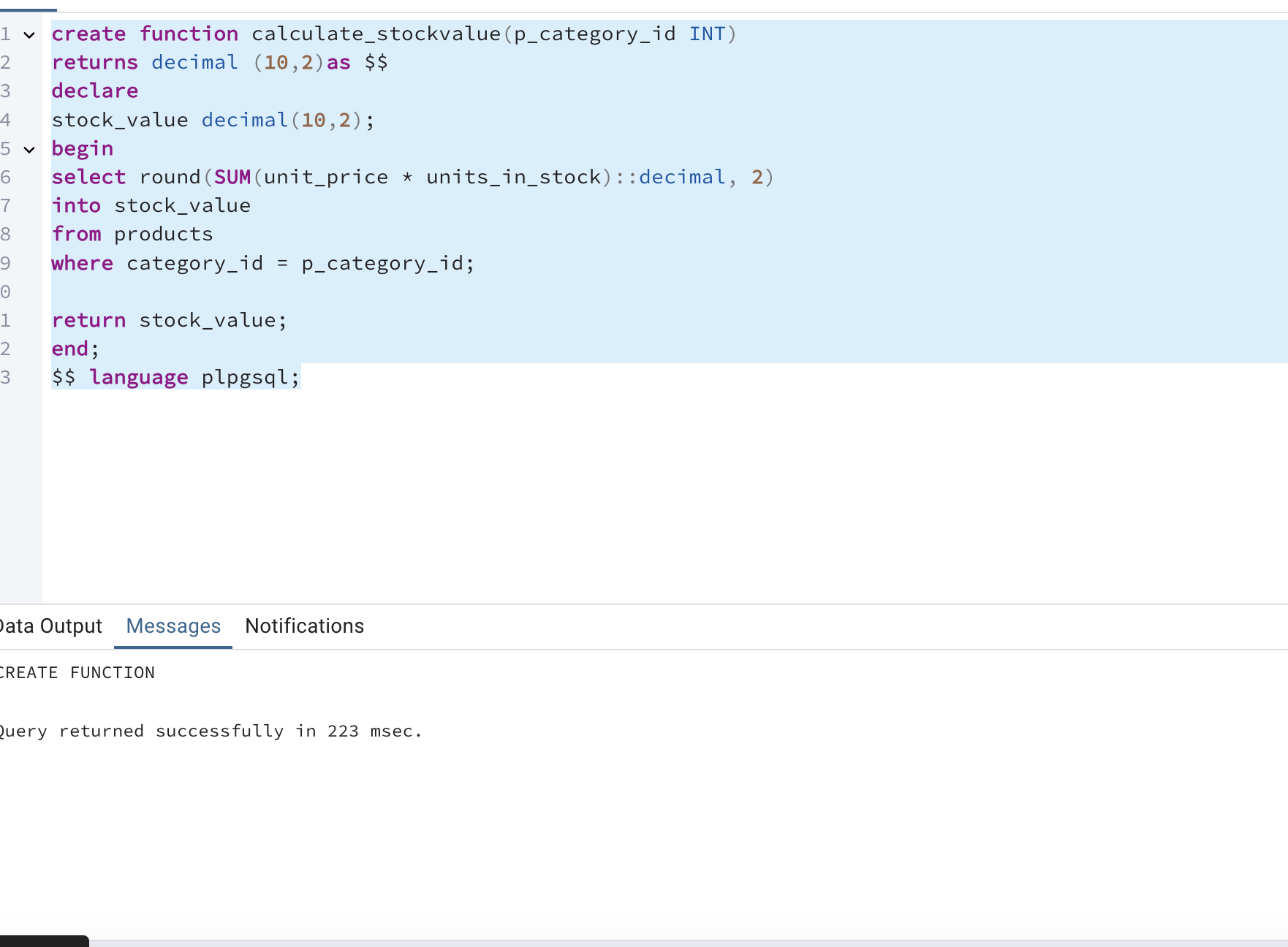
where category\_id = p\_category\_id;

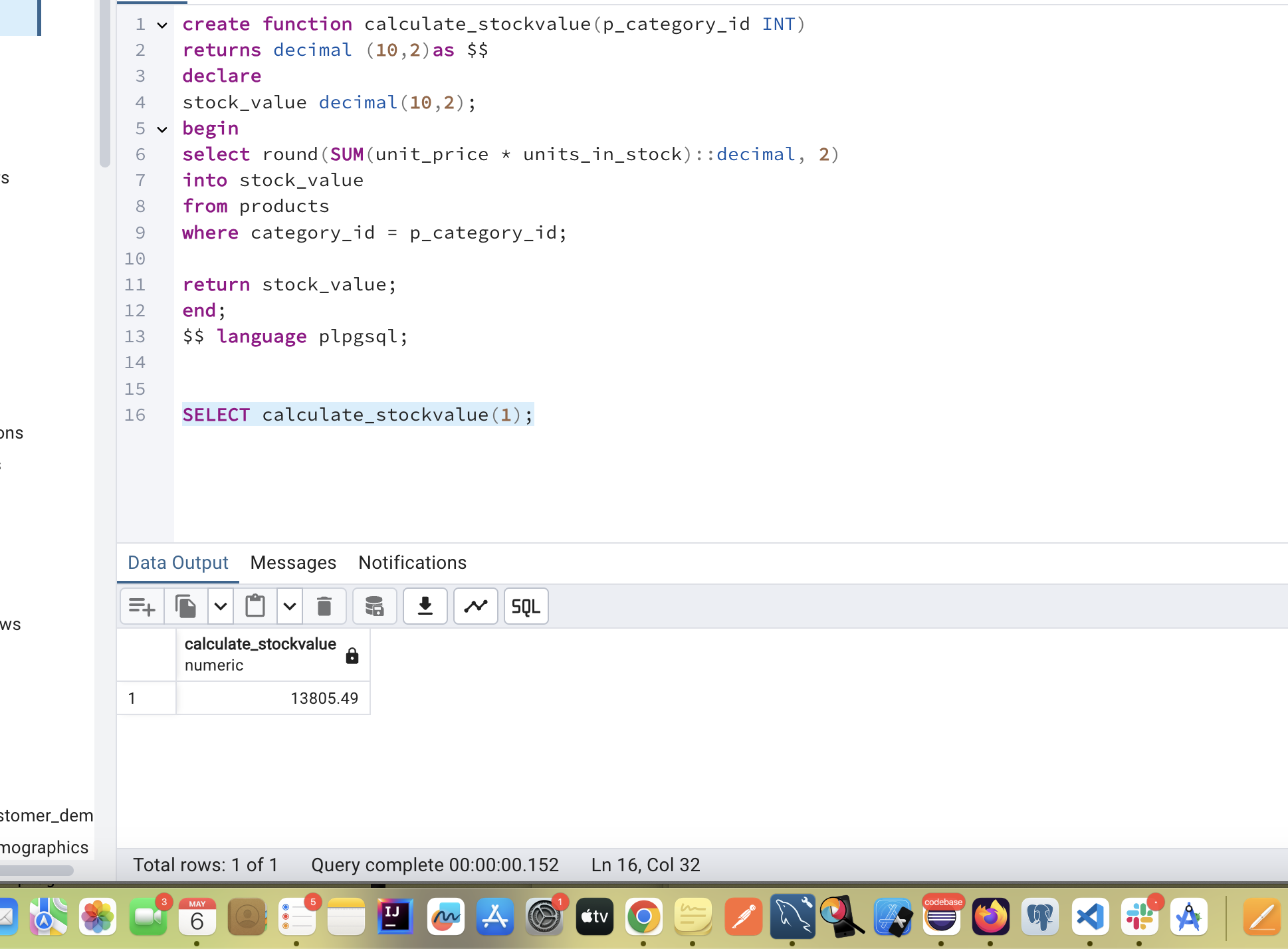
return stock\_value;

end;

$$ language plpgsql;

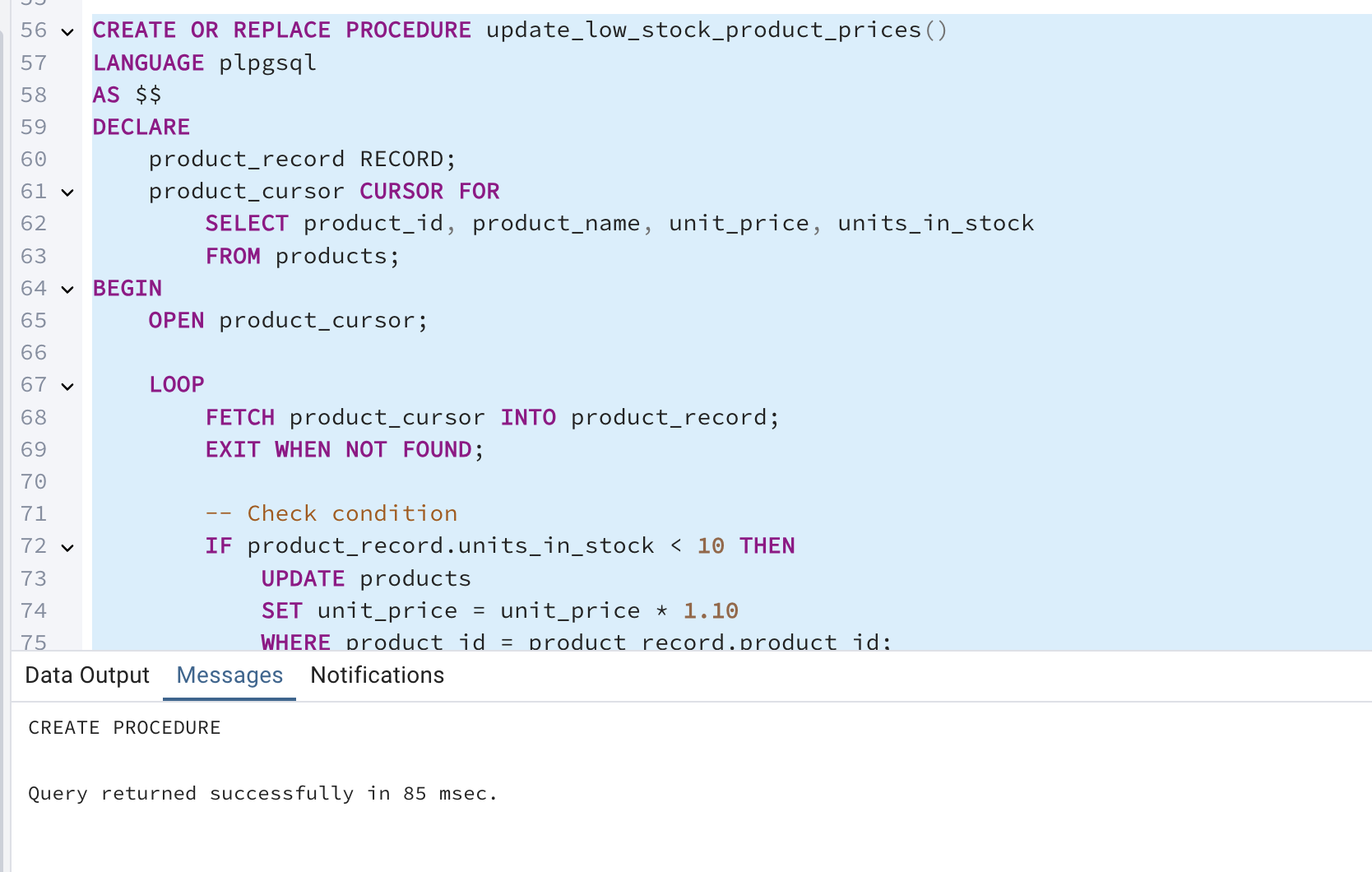
SELECT calculate\_stockvalue(1);





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

So based on the video, the question is The cursor will iterate through the products and adjust prices based on units in stock



CREATE OR REPLACE PROCEDURE update\_low\_stock\_product\_prices()

LANGUAGE plpgsql

AS $$

DECLARE

product\_record RECORD;

product\_cursor CURSOR FOR

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

FROM products;

BEGIN

OPEN product\_cursor;

LOOP

FETCH product\_cursor INTO product\_record;

EXIT WHEN NOT FOUND;

-- Check condition

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

UPDATE products

SET unit\_price = unit\_price \* 1.10

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

RAISE NOTICE 'Updated product % - new price: %',

product\_record.product\_name,

(product\_record.unit\_price \* 1.10);

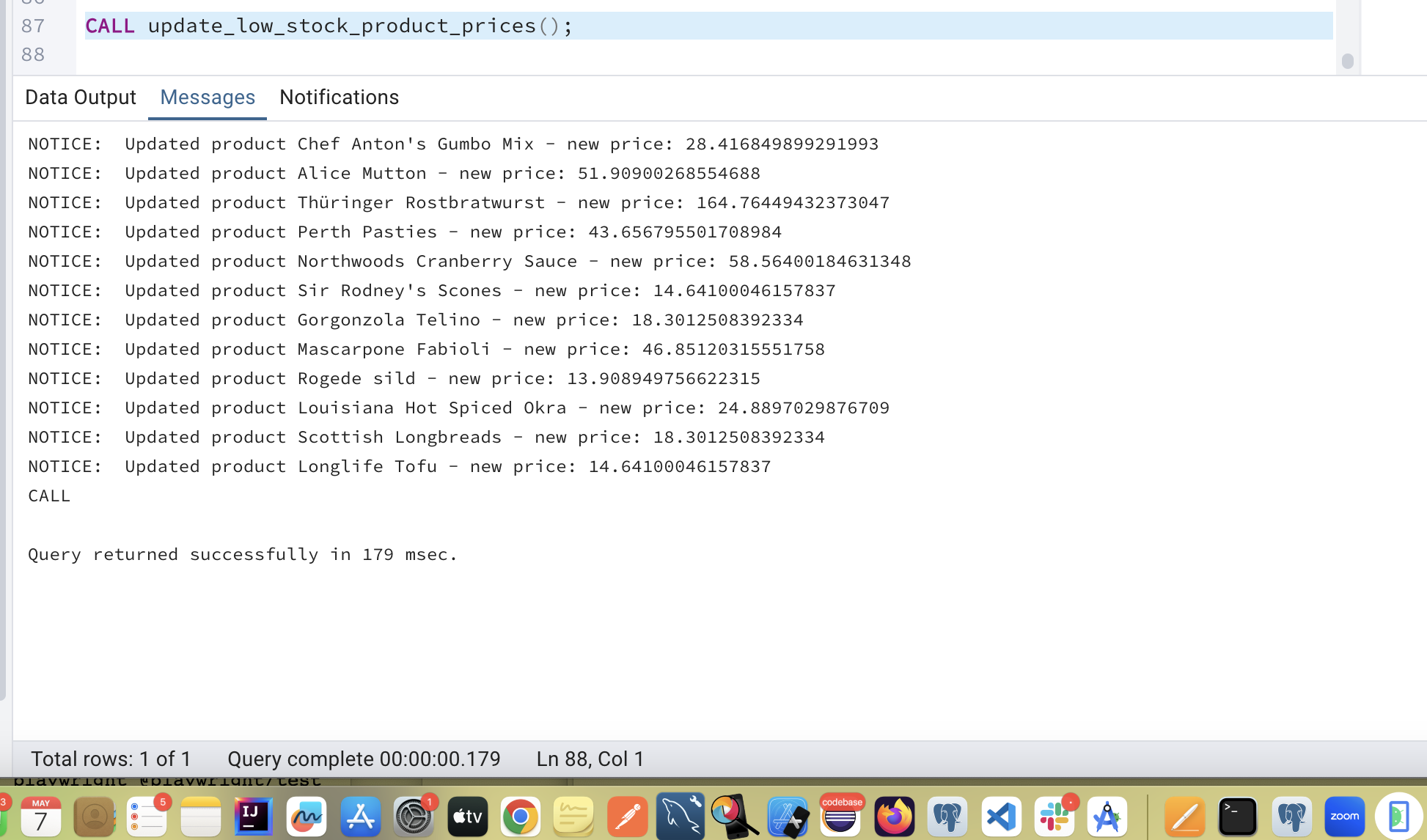
END IF;

END LOOP;

CLOSE product\_cursor;

END;

$$;



CALL update\_low\_stock\_product\_prices();