

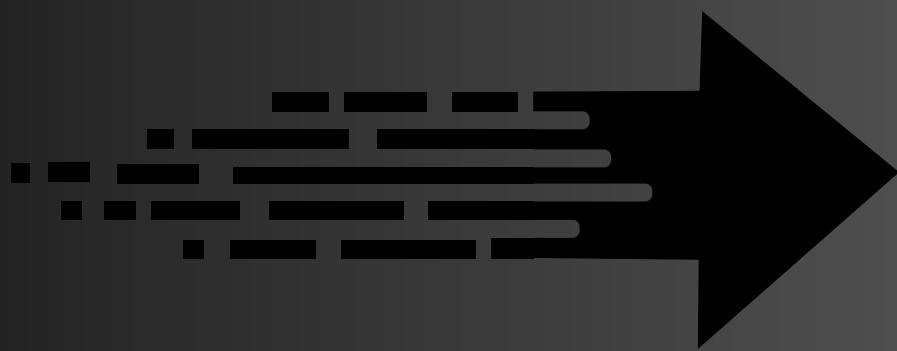


# LeetCode

*50 Days 50 SQL Questions*  
*Work In Progress*

**Day 16/50**

***16. Average Selling  
Price***





# *50 Days 50 SQL Questions*

## *Work In Progress*

# *Question*

Write a solution to find the average selling price for each product. `average_price` should be **rounded to 2 decimal places**. If a product does not have any sold units, its average selling price is assumed to be 0.

Return the result table in **any order**.





# *50 Days 50 SQL Questions*

## *Work In Progress*

# *Table*

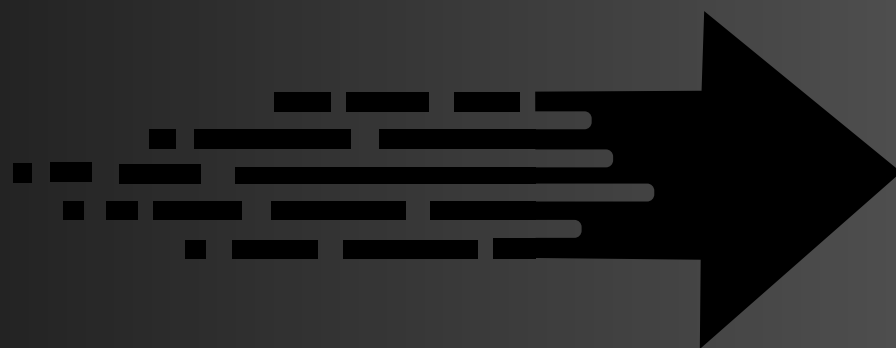
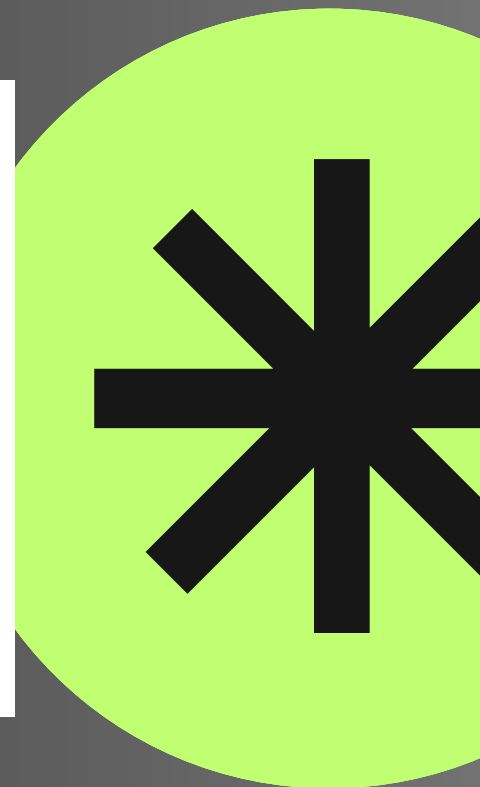
**Input:**

Prices table:

product_id	start_date	end_date	price
1	2019-02-17	2019-02-28	5
1	2019-03-01	2019-03-22	20
2	2019-02-01	2019-02-20	15
2	2019-02-21	2019-03-31	30

UnitsSold table:

product_id	purchase_date	units
1	2019-02-25	100
1	2019-03-01	15
2	2019-02-10	200
2	2019-03-22	30





# *50 Days 50 SQL Questions*

## *Work In Progress*

# *Approach*

Used a Left Join along with 2 conditions with AND op to get purchase\_date between start\_date and end\_date and group the result by product\_id. For avg price, I divided the sum of price\*units by the sum of units and rounded it by 2 decimal pts. and I also used Coalesce to give 0 to product\_ids whose price is 0.





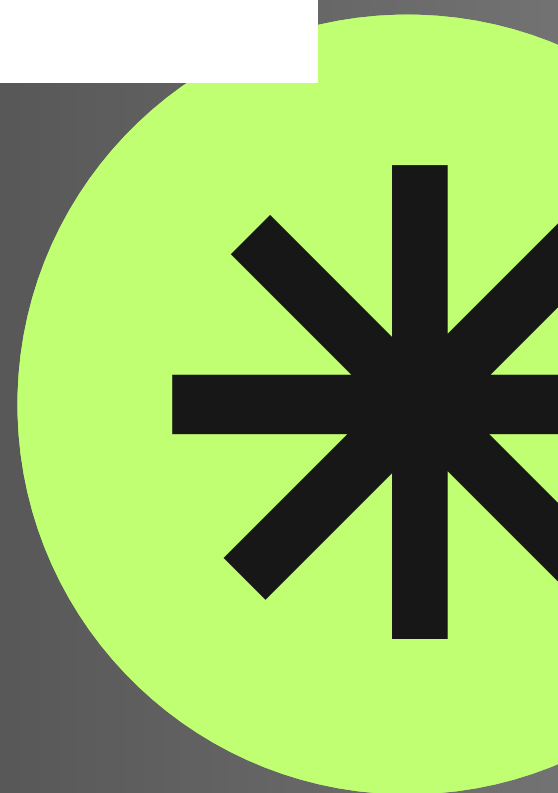
# *50 Days 50 SQL Questions*

## *Work In Progress*

# *Query*

MySQL   Auto

```
1 # Write your MySQL query statement below
2 SELECT
3   a.product_id ,
4   COALESCE(ROUND(SUM(a.price*b.units)/COALESCE(SUM(units),1),2),0) AS average_price
5 FROM Prices AS a LEFT JOIN UnitsSold AS b
6 ON a.product_id = b.product_id
7 AND b.purchase_date >= a.start_date AND b.purchase_date <= a.end_date
8 GROUP BY a.product_id;
```





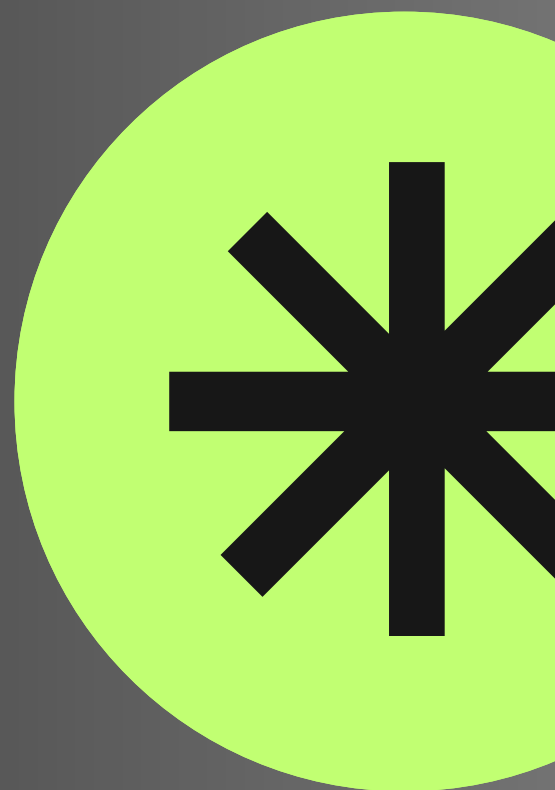
# *50 Days 50 SQL Questions*

## *Work In Progress*

# *Output*

Output:

product_id	average_price
1	6.96
2	16.96





# *50 Days 50 SQL Questions*

## *Work In Progress*

*Share your  
thoughts in the  
comment section*

*Follow For  
More*



*Thank You :)*

