Easy

Table: Prices

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| product_id    | int     |
| start_date    | date    |
| end_date      | date    |
| price         | int     |
+---------------+---------+
```

(product_id, start_date, end_date) is the primary key (combination of columns with unique values) for this table.
Each row of this table indicates the price of the product_id in the period from start_date to end_date.
For each product_id there will be no two overlapping periods. That means there will be no two intersecting periods for the same product_id.

Table: UnitsSold

```
+---------------+---------+
| Column Name   | Type    |
+---------------+---------+
| product_id    | int     |
| purchase_date | date    |
| units         | int     |
+---------------+---------+
```

This table may contain duplicate rows.
Each row of this table indicates the date, units, and product_id of each product sold.

Write a solution to find the average selling price for each product. average_price should be **rounded to 2 decimal places**.

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

The result format is in the following example.

Example 1:

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 |
+------------+---------------+-------+
Output:
+------------+--------------+
| product_id | average_price |
+------------+--------------+
| 1 | 6.96 |
| 2 | 16.96 |
+------------+--------------+
Explanation:
Average selling price = Total Price of Product / Number of products sold.
Average selling price for product 1 = ((100 * 5) + (15 * 20)) / 115 = 6.96
Average selling price for product 2 = ((200 * 15) + (30 * 30)) / 230 = 16.96

```
# Write your MySQL query statement below
SELECT p.product_id, IFNULL(ROUND(SUM(units*price)/SUM(units),2),0)AS average_price
FROM Prices p
LEFT JOIN UnitsSold u ON p.product_id=u.product_id AND (u.purchase_date BETWEEN start_date AND
end_date)
GROUP BY p.product_id;
```