Query

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

Objective

Calculate the **average price of each product**, weighted by the number of units sold during the time period when a specific price was effective.

Step-by-Step Breakdown

1. `FROM Prices p`

- We start with the `Prices` table, which defines the **price of each product for specific time periods** (using `start_date` and `end_date`).

2. `LEFT JOIN UnitsSold u ON ...`

- We join the `UnitsSold` table to get **sales data**.
- Join condition:
  - `p.product_id = u.product_id`: Match products.
  - `purchase_date BETWEEN p.start_date AND p.end_date`: Only include units sold **during the valid price period**.

3. `SUM(p.price * u.units)`

- Multiplies **price by number of units sold** to get total revenue during each price window.

4. `SUM(u.units)`

- Gets the **total number of units sold** for that product.

5. `ROUND(..., 2)`

- The division gives the **average price weighted by units sold**.
- Rounded to 2 decimal places for clarity.

6. `IFNULL(..., 0)`

- If no units were sold (i.e., `SUM(u.units)` is NULL), the result is NULL.
- `IFNULL(..., 0)` converts NULL to 0 to indicate that the average price is effectively zero when no sales happened.

7. `GROUP BY p.product_id`

- Grouping ensures the average is calculated **per product**.

Key SQL Concepts Explained

Concept	Description
LEFT JOIN	Ensures all products are included, even if no sales occurred.
BETWEEN ... AND ...	Filters records to include only dates within a price's validity window.
SUM(price × units)	Computes total revenue generated.
Weighted Average	`(sum of price × units) / (sum of units)` calculates price average weighted by volume.
IFNULL()	Replaces `NULL` values with a default—here `0` for no sales.
ROUND(x, n)	Rounds the result to `n` decimal places for neat output.

Summary for Interviews & Revision

Topic	Details
Weighted Average Price	Important in business analytics where different prices apply across time.
Date Filtering	`BETWEEN` is used to match valid time ranges in temporal data.
Handling NULLs	`IFNULL()` is essential when aggregating over left joins.
LEFT JOIN Usage	Allows inclusion of records (products) even if there's no matching data (sales).