

QUESTION

Above Average Product Prices

Medium

10 Points

Given a table of transactions and products, write a query to return the product ID, product price, and average transaction price of all products with a price greater than the average transaction price.

Output Schema:

Column	Type
product_id	INT
product_price	FLOAT
avg_transaction_price	FLOAT

TABLE SCHEMA

```
1 CREATE TABLE products (  
2   product_id INT PRIMARY KEY,  
3   price DECIMAL(10,2)  
4 );  
5  
6 INSERT INTO products (product_id, price) VALUES  
7 (1, 100.00),  
8 (2, 150.00),  
9 (3, 75.00),  
10 (4, 200.00),  
11 (5, 120.00);  
12  
13 CREATE TABLE transactions (  
14   transaction_id INT PRIMARY KEY,  
15   product_id INT,  
16   amount DECIMAL(10,2),  
17   FOREIGN KEY (product_id) REFERENCES products(product_id)  
18 );  
19  
20 INSERT INTO transactions (transaction_id, product_id, amount) VALUES  
21 (1, 1, 95.00),  
22 (2, 1, 98.00),  
23 (3, 2, 145.00),  
24 (4, 2, 150.00),  
25 (5, 3, 70.00),  
26 (6, 4, 190.00),  
27 (7, 4, 195.00),  
28 (8, 5, 115.00);
```

SOLUTION

```
SELECT
  p.product_id,
  p.price AS product_price,
  t.avg_transaction_price
FROM products p
JOIN (
  SELECT AVG(amount) AS avg_transaction_price
  FROM transactions
) t
ON p.price > t.avg_transaction_price;
```

OUTPUT

▼ Tables

product_id	product_price	avg_transaction_price
2	150	132.25
4	200	132.25

My Thought Process:

I started by calculating the average transaction amount using a subquery (AVG(amount)).

Then I compared each product's price against this average using a simple JOIN, and filtered to include only those products where price > avg_transaction_price.

Business Impact:

This query can highlight which products consistently exceed average transaction values, making them strong candidates for upselling or bundling.

It supports identifying high-margin items that contribute more to revenue.