Day 2 SQL Practice – Products & Orders Dataset

On Day 2 of my SQL journey, I practiced intermediate concepts using a custom Products & Orders dataset. Today's focus was on deepening my understanding of joins, aggregation, subqueries, and analytical thinking.

🔍 Reflections & Learning Summary

1. List all product names and their categories
Simple SELECT statement to retrieve basic attributes from a single table.

2. Find the total quantity ordered for each product
Used LEFT JOIN to ensure even products not in any order appear with NULL or 0 total quantity. Grouped by product name and aggregated using SUM().

3. Display all orders along with product names and category
Used INNER JOIN to combine order details with product metadata. Selected multiple columns to reflect a complete order view.

4. Calculate the total order value for each customer (Quantity × Price)
Joined Orders and Products, then calculated the total using multiplication inside SUM(). Grouped by customer for summarization.

5. Which customers have placed orders for Electronics products?
Filtered the joined result on product category. Used DISTINCT to avoid duplicates if a customer ordered more than one Electronics item.

6. Find the most expensive product and who bought it
Used a subquery to find MAX(Price) and filtered main query based on that. Combined with join to fetch related order details.

7. List the product(s) that have never been ordered
Used LEFT JOIN and checked for NULLs in Orders table to identify unmatched products.

8. Show the average order quantity per product category
Grouped joined data by product category and used AVG() to get mean quantity ordered.

9. Find all customers who ordered more than 2 units of any product
Filtered on Quantity > 2 and optionally used DISTINCT to simplify result set.

Bonus Challenge: Customers who ordered products priced above the average price
Used a subquery in WHERE clause to compare price to overall product average. Applied DISTINCT to clean the result.

💡 Key Takeaways from Day 2

- Reinforced the use of JOINs and how each type (INNER, LEFT) changes result sets
- Improved confidence with subqueries and filters
- Learned to analyze results both from a data integrity and business logic perspective
- Realized the importance of thinking beyond syntax — context matters too!