**📊 Day 63 – SQL Challenge**

**Dataset: RetailEcommerce**

**Table: Customers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CustomerID** | **Name** | **Country** | **SignupDate** | **LoyaltyPoints** |
| 1 | Ananya Rao | India | 2020-01-10 | 1200 |
| 2 | John Smith | USA | 2019-11-25 | 2300 |
| 3 | Priya Patel | India | 2021-02-15 | 800 |
| 4 | David Green | UK | 2020-06-20 | 1500 |
| 5 | Meera Nair | India | 2021-09-10 | 400 |

**Table: Products**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ProductID** | **ProductName** | **Category** | **Price** | **Stock** |
| 101 | Laptop | Electronics | 75000 | 15 |
| 102 | Mobile Phone | Electronics | 35000 | 30 |
| 103 | Headphones | Accessories | 4000 | 50 |
| 104 | Coffee Maker | HomeAppliance | 7000 | 10 |
| 105 | Office Chair | Furniture | 12000 | 20 |

**Table: Orders**

|  |  |  |  |
| --- | --- | --- | --- |
| **OrderID** | **CustomerID** | **OrderDate** | **TotalAmount** |
| 5001 | 1 | 2021-12-01 | 75000 |
| 5002 | 2 | 2021-12-05 | 39000 |
| 5003 | 3 | 2022-01-15 | 35000 |
| 5004 | 4 | 2022-02-10 | 16000 |
| 5005 | 1 | 2022-03-05 | 115000 |
| 5006 | 5 | 2022-03-15 | 7000 |

**Table: OrderDetails**

|  |  |  |  |
| --- | --- | --- | --- |
| **OrderDetailID** | **OrderID** | **ProductID** | **Quantity** |
| 1 | 5001 | 101 | 1 |
| 2 | 5002 | 102 | 1 |
| 3 | 5002 | 103 | 1 |
| 4 | 5003 | 102 | 1 |
| 5 | 5004 | 103 | 2 |
| 6 | 5004 | 104 | 1 |
| 7 | 5005 | 101 | 1 |
| 8 | 5005 | 105 | 2 |
| 9 | 5006 | 104 | 1 |

**Day 63 SQL Questions**

1. **Basic Join** – List all customers with their total order amount.
2. **Aggregation** – Find the top 3 products with the highest sales revenue.
3. **Filtering** – Show all orders placed by customers from India.
4. **Date Functions** – Find the number of orders placed in each month of 2022.
5. **CASE Statement** – Categorize customers into “High Value” (Total ≥ 1,00,000) and “Regular” based on their order spend.
6. **Window Function (ROW\_NUMBER)** – Get the most recent order per customer.
7. **Window Function (RANK)** – Rank customers by their loyalty points.
8. **Subquery** – Find products that have never been ordered.
9. **CTE (Recursive)** – Generate a running total of order amounts by order date.
10. **Advanced Join + Aggregation** – For each category, find the average spend per customer.
11. **Bonus Challenge** 🚀 – Write a query to find the **customer who contributed the highest % of total revenue** and display their contribution percentage.