**📘 Day 58 – SQL Challenge**

**Dataset Theme:** Banking Transactions & Fraud Detection

**Tables & Sample Data**

**Customers**

|  |  |  |  |
| --- | --- | --- | --- |
| CustomerID | Name | Country | JoinDate |
| 1 | Alice | USA | 2018-01-15 |
| 2 | Bob | India | 2019-05-10 |
| 3 | Charlie | UK | 2020-03-25 |
| 4 | David | Canada | 2021-07-18 |
| 5 | Eva | Germany | 2019-12-01 |

**Accounts**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| AccountID | CustomerID | AccountType | Balance | Status |
| 101 | 1 | Savings | 5000 | Active |
| 102 | 2 | Checking | 12000 | Active |
| 103 | 3 | Savings | 8000 | Active |
| 104 | 4 | Checking | 15000 | Active |
| 105 | 5 | Savings | 2000 | Inactive |

**Transactions**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TransactionID | AccountID | TransDate | Amount | TransactionType | Location |
| 201 | 101 | 2023-01-10 | 500 | Withdrawal | New York |
| 202 | 101 | 2023-01-12 | 2000 | Withdrawal | New York |
| 203 | 101 | 2023-01-15 | 3000 | Withdrawal | Los Angeles |
| 204 | 102 | 2023-01-11 | 4000 | Deposit | Delhi |
| 205 | 102 | 2023-01-15 | 1000 | Withdrawal | Delhi |
| 206 | 103 | 2023-01-20 | 500 | Withdrawal | London |
| 207 | 104 | 2023-01-21 | 6000 | Withdrawal | Toronto |
| 208 | 105 | 2023-01-22 | 1500 | Withdrawal | Berlin |
| 209 | 104 | 2023-01-25 | 7000 | Withdrawal | Toronto |
| 210 | 104 | 2023-01-27 | 7500 | Withdrawal | New York |

**FraudAlerts**

|  |  |  |  |
| --- | --- | --- | --- |
| AlertID | TransactionID | Reason | AlertDate |
| 301 | 203 | Unusual location | 2023-01-16 |
| 302 | 209 | High withdrawal amount | 2023-01-26 |
| 303 | 210 | Cross-border suspicious activity | 2023-01-28 |

**Questions**

1. Find the **total transaction amount per customer**.
2. Identify accounts with **average withdrawal > 2000**.
3. List customers who performed **transactions in more than 1 country**.
4. Find the **largest transaction** for each account.
5. Show customers who had **more than 2 withdrawals in the same week**.
6. Using a **window function**, calculate the **running total balance after each transaction**.
7. Identify **suspicious transactions**: withdrawal > 70% of current account balance.
8. Find customers who had transactions flagged in **FraudAlerts** and their reasons.
9. Detect customers with **multiple transactions in different cities on the same day**.
10. Rank transactions for each customer by **amount (descending)** using RANK().

**Bonus (Advanced):**  
11. Build a query to detect **potential fraud risk accounts**:

* Accounts with ≥ 2 fraud alerts in last 30 days, OR
* Transactions from ≥ 3 different locations in a single month.