📘 Project Title

Credit Card Financial Analytics Dashboard – Power BI + SQL + DAX Project

---

📌 Project Overview

This project focuses on analyzing credit card data to uncover customer spending behavior, revenue trends, card performance, and week-over-week business growth. The solution is built using **Power BI** for data visualization, **SQL** for data extraction, and **DAX** for analytical measures.

☑ The dashboard helps answer critical business questions such as:

- Which card category generates the highest revenue?
- What is the revenue contribution by age, income, and profession?
- Which transaction mode is preferred by users?
- How does revenue grow week-over-week?

📊 Key Business Insights

Revenue Insights

- **Total Revenue:** $55M
- **Highest Revenue Card Type:** Platinum (25%)
- **Highest Revenue Quarter:** Q4 ($13M)
- **Preferred Transaction Method:** Swipe (generates $35M)
- **Interest Earned:** $8M

👥 *Customer Insights*

- **Top Age Group by Revenue:** 40–50 years (4.5K customers)
- **Income Segment Contribution:** High-income segment (> $70K) contributes **53% revenue**
- **Top Customer Profession:** Businessmen ($17M revenue)
- **Average Satisfaction Score:** 3.19 / 5

🔄 *Week-over-Week Analysis*

- Calculated using **DAX time intelligence**
- Shows growth or decline in revenue compared to previous week

---

🗂️ Dataset Description

Tables Used:

1. `cust_detail` – Customer demographic details
2. `cc_detail` – Credit card transactions and generated revenue

🔗 Data Model View

The data model follows a **star schema** design with a **one-to-many** relationship:

```
cust_detail (1) ———< (*) cc_detail
```

Relationship:

From Table	Column	To Table	Column	Relationship Type
cust_detail	Client_Num	cc_detail	Client_Num	One-to-Many (1:*)

🔎 **Usage**:

- `cust_detail` acts as a **dimension table** (used for slicing data)
- `cc_detail` acts as a **fact table** (stores revenue and transaction-level data)

🧠 DAX Measures Used

1. *Age Group Classification*

```
AgeGroup = SWITCH(
    TRUE(),
    'ccdb cust_detail'[Customer_Age] < 30, "20-30",
    AND('ccdb cust_detail'[Customer_Age] >= 30,'ccdb
cust_detail'[Customer_Age] < 40), "30-40",
```

```
    AND('ccdb cust_detail'[Customer_Age] >= 40,'ccdb
cust_detail'[Customer_Age] < 50), "40-50",
    AND('ccdb cust_detail'[Customer_Age] >= 50, 'ccdb
cust_detail'[Customer_Age] < 60), "50-60",
    'ccdb cust_detail'[Customer_Age] >= 60, "60+",
    "Unknown"
)
```

2. Income Group Classification

```
IncomeGgroup = SWITCH(
    TRUE(),
    'ccdb cust_detail'[Income] < 35000, "Low",
    AND('ccdb cust_detail'[Income] >= 35000, 'ccdb cust_detail'[Income] <
70000), "Med",
    'ccdb cust_detail'[Income] >= 70000, "High",
    "Unknown"
)
```

3. Week Number Calculation

```
Week_num2 = WEEKNUM('ccdb cc_detail'[Week_Start_Date])
```

4. Revenue Calculation

```
Revenue =
    'ccdb cc_detail'[Annual_Fees] +
    'ccdb cc_detail'[Total_Trans_Amt] +
    'ccdb cc_detail'[Interest_Earned]
```

5. Current Week Revenue

```
Current_Week_Revenue = CALCULATE(
    SUM('ccdb cc_detail'[Revenue]),
    FILTER(
        ALL('ccdb cc_detail'),
        'ccdb cc_detail'[Week_num2] = MAX('ccdb cc_detail'[Week_num2])
    ))
```

6. Previous Week Revenue

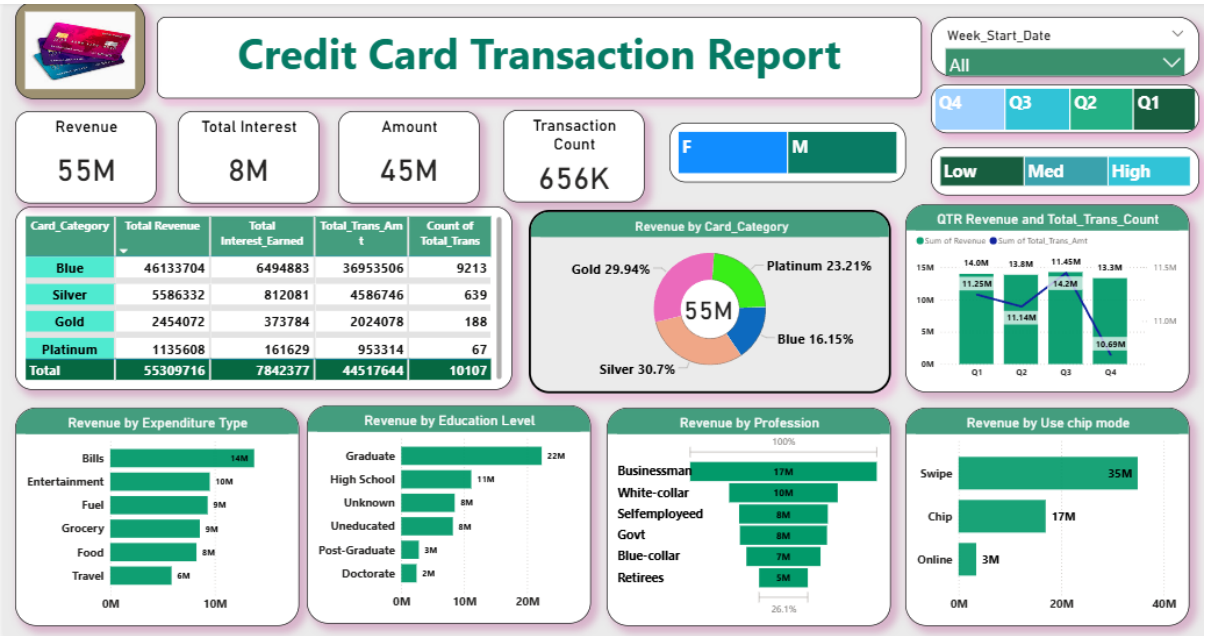
```
Previous_week_Revenue = CALCULATE(
    SUM('ccdb cc_detail'[Revenue]),
    FILTER(
        ALL('ccdb cc_detail'),
        'ccdb cc_detail'[Week_num2] = MAX('ccdb cc_detail'[Week_num2]) - 1
    ))
```

7. Week-over-Week Revenue %

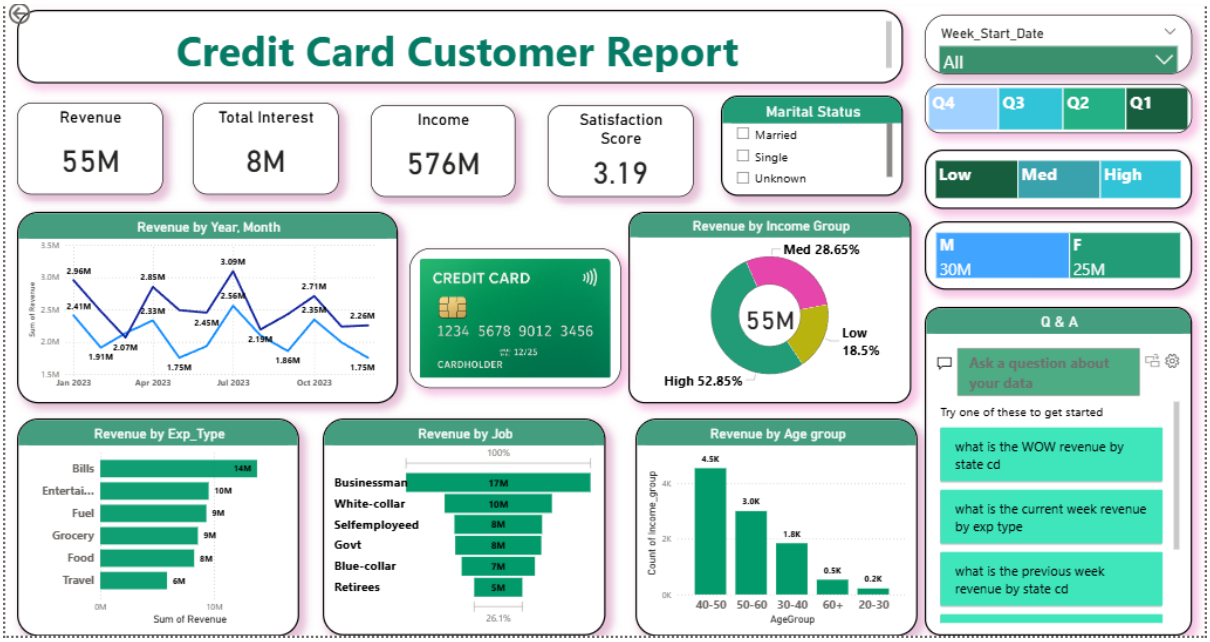
```
WOW_Revenue = DIVIDE(
    [Current_Week_Revenue] - [Previous_week_Revenue],
    [Previous_week_Revenue]
)
```

📈 Dashboards Included

📷 Credit Card Transaction Report (Power BI Dashboard)



📷 Credit Card Customer Report (Power BI Dashboard)



👩‍💻 Author

Nilam Lokhande

🔗 LinkedIn: https://www.linkedin.com/in/nilam-lokhande-b043a2376/

🔗 GitHub: https://github.com/nilamlokhande

📌 Conclusion & Future Scope

This dashboard successfully showcases how **SQL data**, combined with **Power BI visualizations** and **DAX logic**, can provide powerful insights into customer behavior and business performance.

----------END----------
--