🛍️ Retail Customer Transaction and Fraud Pattern Analysis

🧰Tools : SQL • Power BI • Python

Author : Priya Dutta

🔍 Business Goal:

Analyze suspicious transactions + **generate** business insights

📂 Dataset:

7,500 transactions

3 Tables: Customers, Merchants, Transactions

🧰 Tool-wise Contribution

- **SQL** - Joins, fraud %, revenue by region
- **Power BI** - KPI dashboard, filters, top cities by fraud
- **Python** - Heatmap, city-wise revenue, fraud trends

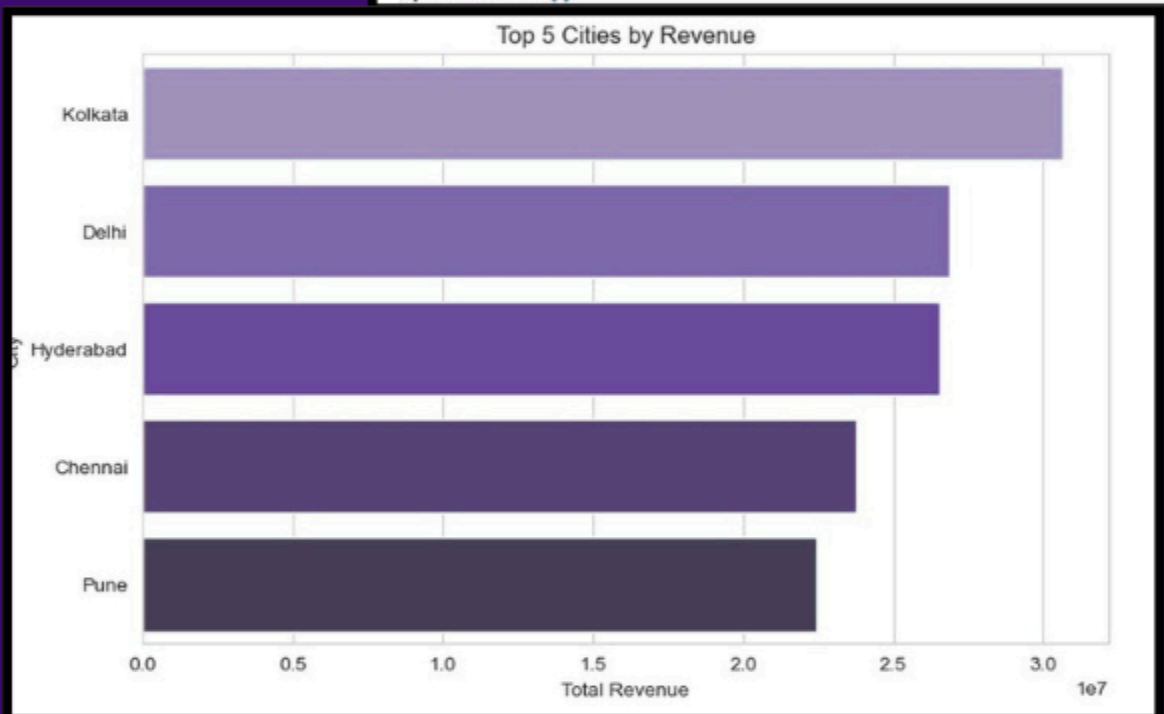
💡 Key Business & Fraud Insights

- **East** = Top revenue region (**₹5.15 Cr**)
- **West** = Highest fraud % (**5.05%**)
- Top Customer **CUST0387** = **₹6.24 Lakhs**
- High Fraud in **Beauty** & **Home Decor** Category

Retail Transaction and Fraud Insights
SQL + Power BI + Python

```python
top_cities_revenue = df.groupby('City')['Amount'].sum().sort_values(ascending=False).head(5)

plt.figure(figsize=(8,5))
sns.barplot(x=top_cities_revenue.values, y=top_cities_revenue.index, palette='Blues_d')
plt.title("Top 5 Cities by Revenue")
plt.xlabel("Total Revenue")
plt.ylabel("City")
plt.tight_layout()
plt.show()
```

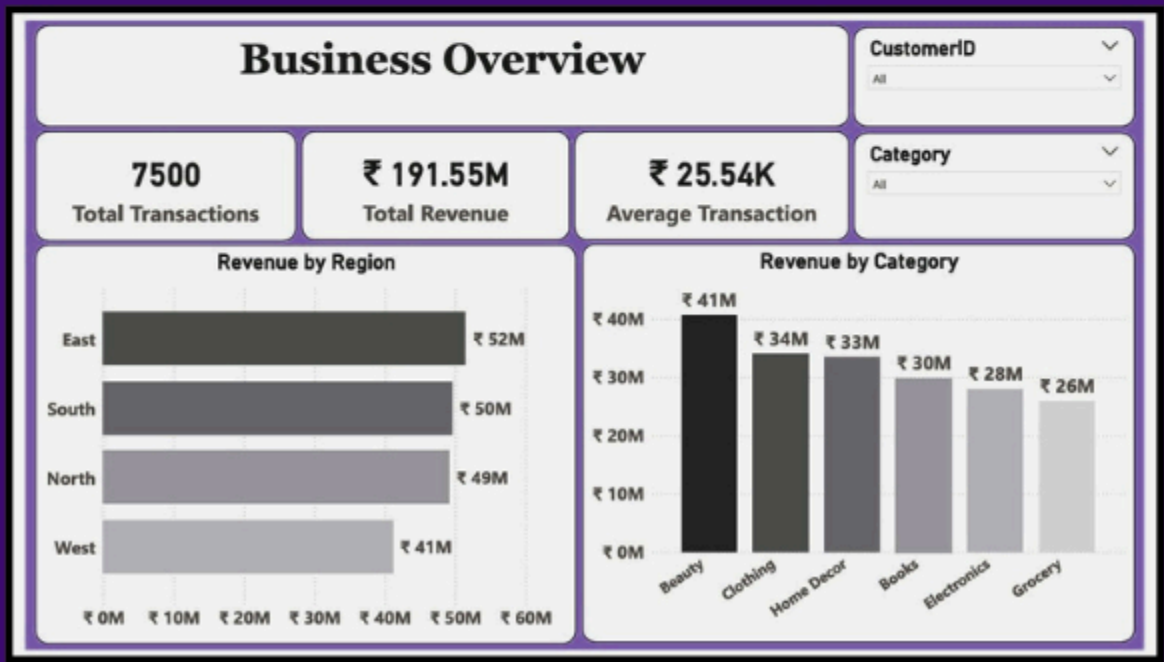



```sql
--Step 3.3: Fraud Count by Region
SELECT
  C.Region,
  COUNT(*) AS Total_Transactions,
  SUM(CAST(T.IsFraud AS INT)) AS Fraud_Transaction
FROM Transactions T
JOIN Customers C ON T.CustomerID = C.CustomerID
GROUP BY C.Region
ORDER BY Fraud_Transactions DESC;
```

100 %

Results | Messages

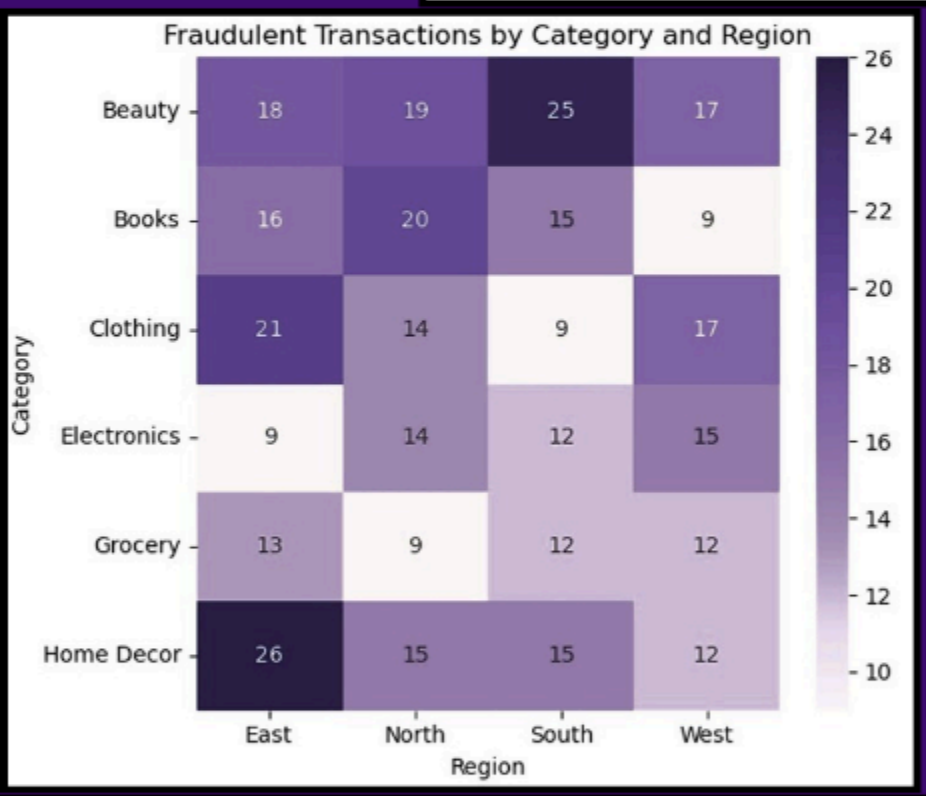
	Region	Total_Transactions	Fraud_Transactions
1	East	2046	103
2	North	1923	91
3	South	1907	88
4	West	1624	82

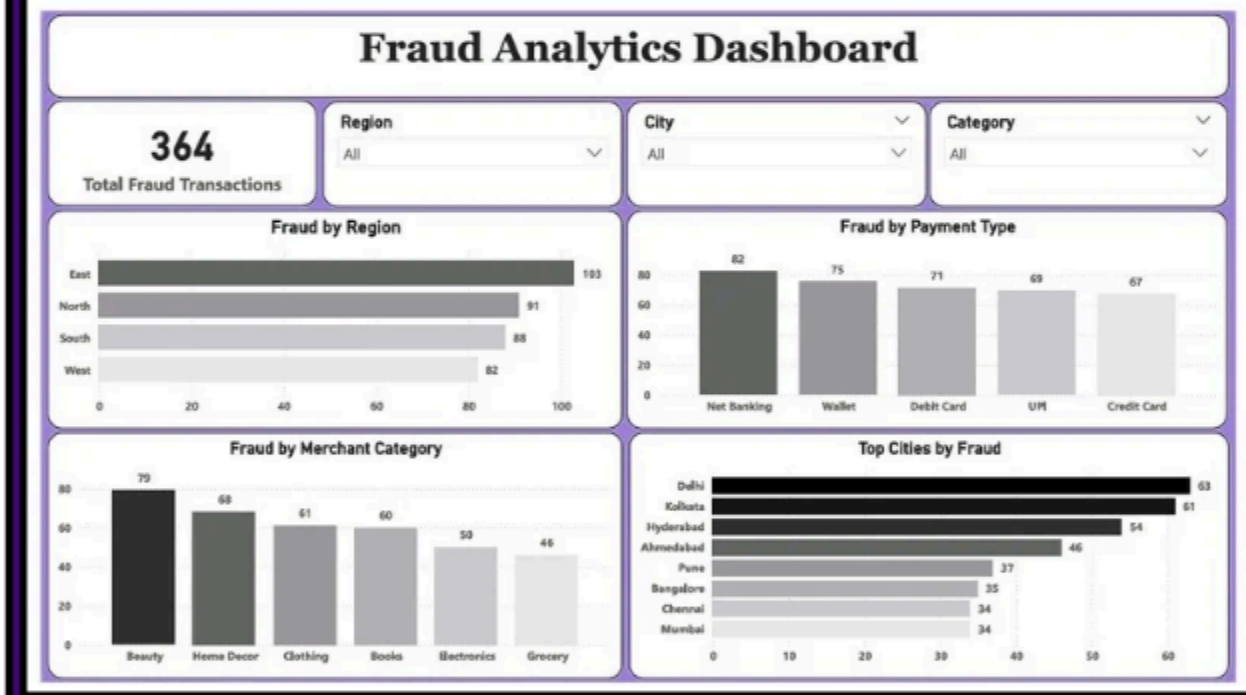


Retail Transaction Insights
SQL + Power BI + Python

```python
heatmap_data = df[df['IsFraud'] == 1].pivot_table(index='Category', columns='Region',
                                                  values='TransactionID', aggfunc='count', fill_value=0)

plt.figure(figsize=(6,5))
sns.heatmap(heatmap_data, annot=True, cmap='Reds')
plt.title('Fraudulent Transactions by Category and Region')
plt.xlabel('Region')
plt.ylabel('Category')
plt.tight_layout()
plt.show()
```






```sql
--Step 3.6.4: Fraud % by Region
SELECT
  C.Region,
  COUNT(*) AS Total_Transactions,
  SUM(CAST(T.IsFraud AS INT)) AS Fraud_Transactions,
  CAST(SUM(CAST(T.IsFraud AS FLOAT)) * 100.0 / COUNT(*) AS DECIMAL(5,2)) AS Fraud_Percentage
FROM Transactions T
JOIN Customers C ON T.CustomerID = C.CustomerID
GROUP BY C.Region
ORDER BY Fraud_Percentage DESC;
```

00 %

Results | Messages

	Region	Total_Transactions	Fraud_Transactions	Fraud_Percentage
1	West	1624	82	5.05
2	East	2046	103	5.03
3	North	1923	91	4.73
4	South	1907	88	4.61

Project Workflow: From Raw Data to Insightful Dashboards

🔹 **SQL - Data Cleaning & Analysis**

➤ Loaded 3 tables: customers, merchants, transactions

➤ Handled nulls, checked uniques, joined data

➤ Wrote 10+ queries: revenue, fraud %, top customers/cities, category-wise fraud

🔹 **Python - Visual Fraud Pattern Analysis**

➤ 📊 Fraud Transactions by Region (bar chart)

➤ 📊 Top 5 Cities by Revenue (bar chart)

➤ 📊 Fraud % by Region (bar chart)

➤ 🔥 Heatmap: Fraud by Region × Category

🔹 **Power BI - 3-Page Report**

➤ 📌 Pages: Business Overview, Top Performers, Fraud Analytics

➤ 📈 KPIs: Total Transactions, Revenue, Avg Value, Fraud Transactions

➤ 🔍 Filters: CustomerID, Region, City, Merchant, Category

Tools & Technologies Used

🔹 SQL:

- Joins, Group By, Case statements
- 10+ exploratory + fraud queries + Data merge

🔹 Python:

- Pandas, Matplotlib, Seaborn
- Bar plots, Heatmap
- EDA

🔹 Power BI:

- KPI cards, bar & column charts
- Page navigation (3 dashboards)
- Slicers: Customer, Region, Category, City, Merchant

Key SQL Queries & Insights

🔹💰 Revenue by Region

```
SELECT
   C.Region,
   COUNT(T.TransactionID) AS Total_Transactions,
   SUM(T.Amount) AS Total_Revenue
FROM Transactions T
JOIN Customers C ON T.CustomerID = C.CustomerID
GROUP BY C.Region
ORDER BY Total_Revenue DESC;
```

💡 Helped identify top-performing zones by revenue

🏆 East emerged as highest with ₹5.15 Cr

Key SQL Queries & Insights

🔹💰Revenue by Merchant Category

```
SELECT
   M.Category,
   COUNT(T.TransactionID) AS
Total_Transactions,
   SUM(T.Amount) AS Total_Revenue
FROM Transactions T
JOIN Merchants M ON T.MerchantID =
M.MerchantID
GROUP BY M.Category
ORDER BY Total_Revenue DESC;
```

💡Top 3 categories by revenue:
Beauty (₹4.06 Cr), Clothing (₹3.40 Cr), Home Decor (₹3.34 Cr)

Key SQL Queries & Insights

🔹👥 Top 5 Customers by Spend

```
SELECT TOP 5
   T.CustomerID,
   SUM(T.Amount) AS Total_Spent
FROM Transactions T
GROUP BY T.CustomerID
ORDER BY Total_Spent DESC;
```

💡Top 5 spenders alone contributed over ₹29 lakh

1.CUST0387 - ₹6.24 lakh

2.CUST0211 - ₹6.20 lakh

3.CUST0059 - ₹5.77 lakh

4.CUST0013 - ₹5.74 lakh

5.CUST0073 - ₹5.72 lakh

Key SQL Queries & Insights

🔹 💵Fraud by Payment Type

```
SELECT
    PaymentType,
    COUNT(*) AS Total_Transactions,
    SUM(CAST(IsFraud AS INT)) AS
Fraud_Count
FROM Transactions
GROUP BY PaymentType
ORDER BY Fraud_Count DESC;
```

💡 Netbanking was the riskiest payment method with 82 frauds, followed by Wallets (75) and Debit Cards (71)

Power BI Dashboards : Business + Fraud in One View

🗂️📊 Page 1 : Business Overview

✅ KPIs:

Total Revenue : ₹19.15 Cr | Total Transactions : 7,500 | AOV : ₹25,540

📍 Revenue by region bar chart revealed : East region brought in ₹5.15 Cr, highest among all

🔎Filters used : Category & CustomerID

Power BI Dashboards : Business + Fraud in One View

🗂️🌟 Page 2 : Top Performers Overview

💰 Identified Top 5 Customers by Spend
CUST0387 led with highest overall purchases

📈 Region-wise bar charts confirmed East's dominance
and Beauty as most purchased category

🔍 Filters used : CustomerID & Merchant Name

Power BI Dashboards : Business + Fraud in One View

🗂️⚠️ Page 3 : Fraud Analytics Dashboard

🚨 Fraud Transactions : 364

⚠️ West region had highest fraud %

📊 Visuals revealed:

- ⚠️High fraud in Beauty & Home Decor
- ⚠️Delhi had the highest fraud cases.
-Fraud-prone payment types-Netbanking & Wallets
-🔍 Filter used - Region, City & Category

SQL Join: Merging Tables for Python-Based EDA

🔄 Why this Step ?

We had 3 separate tables: customers, merchants, transactions

To do complete EDA in Python, we needed customer, merchant, and transaction info together

So we joined the tables in SQL to create one clean dataset for analysis

💡 Merging in SQL saved time and gave full visibility for analysis in Python.

SQL Join: Merging Tables for Python-Based EDA

💻 SQL Join Query :

```
SELECT
    T.TransactionID,
    T.CustomerID,
    C.Name, C.Gender, C.AgeGroup, C.Region,
C.SignupDate,
    T.MerchantID,
    M.MerchantName, M.Category, M.City, M.Country,
    T.TransactionDate, T.Amount, T.PaymentType,
T.IsFraud
FROM Transactions T
JOIN Customers C ON T.CustomerID = C.CustomerID
JOIN Merchants M ON T.MerchantID = M.MerchantID
```

📌 **Result :**

✅ 16 columns

✅ 7,500 rows

✅ Includes fraud flags + business context for Python analysis

Python Visuals: Detecting Fraud Patterns

🔹 1 . Fraud Transactions by Region (Bar Chart)

📊 Compared fraud volume by region

⚠️ East had the highest number of fraud cases

🔹 2 . Top 5 Cities by Revenue (Bar Chart)

📊 Ranked cities by customer spend

🏙️ Kolkata, Delhi and Hyderabad are the top contributors to revenue

🔹 3 . Fraud % by Region (Bar Chart)

📊 Calculated % = Fraud Txns / Total Txns

📍 West showed highest fraud rate

🔹 4 . Heatmap - Region × Category

📊 Showed fraud count across region/category pairs

🔥 Found fraud-prone combos like Home Decor in East, Beauty in South

Business Value: Why This Project Matters

📈What This Project Helped Achieve?

- Identified High-Performing Regions

→ East region brought ₹5.15 Cr in revenue – useful for sales targeting

- Flagged Fraud-Prone Zones

→ West had the highest fraud %, helpful for setting fraud controls

- Uncovered Category Trends

→ Beauty & Clothing earned the most revenue

→ Beauty & Home Decor had the most fraud cases

→ These categories may need closer checks

- Highlighted Valuable Customers

→ Top spender CUST0387 spent ₹6.24 lakh – shows strong loyalty

- Connected Sales & Risk Teams with One View

→ Dashboard gives both sales and fraud teams the insights they need

This project turned raw data into useful strategies – helping improve both revenue and fraud prevention

Key Takeaways from This Project

- Strengthened end-to-end data pipeline handling (SQL → Python → Power BI)
- Gained deeper understanding of fraud patterns across regions & categories
- Practiced visual storytelling with real dashboards & charts
- Learned to clean, analyze & present data across multiple tools
- Built confidence in turning raw data into actionable business insights

📄 Resume - ready • 📊 Interview - friendly

🔍 View full project :

github.com/duttapriya993/Retail_Transaction_Insights

💬 Would love your feedback!

Thank you!