**TRANSACTION FRAUD ANALYSIS**

# Overview

This project focuses on analyzing transactional data for fraud detection and customer value assessment. Significant tables include Financial Data, Merchant Details, Customer Data, and Card Details. Tools used include Python for data processing, SQL for querying and relational database management, and Power BI for interactive data visualization and insights. The goal is to enhance fraud detection and identify high-value customers.

# Dataset Overview

**1. Financial Data**

* **Columns**: transaction\_id, customer\_id, merchant\_id, amount, is\_fraudulent, card\_ID, CLV, location\_ID
* **Description**: Contains transaction records including customer IDs, merchant IDs, transaction amounts, fraud indicators, card details, customer lifetime value (CLV), and corresponding location IDs. This data is crucial for analyzing patterns in fraudulent activities.

**2. Customer Details**

* **Columns**: customer\_id, Customer\_Age, Account\_Tenure\_Years, Transaction\_Frequency
* **Description**: Contains information about customers, including their age, how long they have had their accounts, and their transaction frequency. These attributes help in understanding customer behavior and assessing risk factors related to fraud.

**3. Merchant Details**

* **Columns**: merchant\_id, purchase\_category
* **Description**: Lists merchants and their associated purchase categories. This data aids in analyzing transaction types and potential fraud hotspots based on merchant categories.

**4. Location Details**

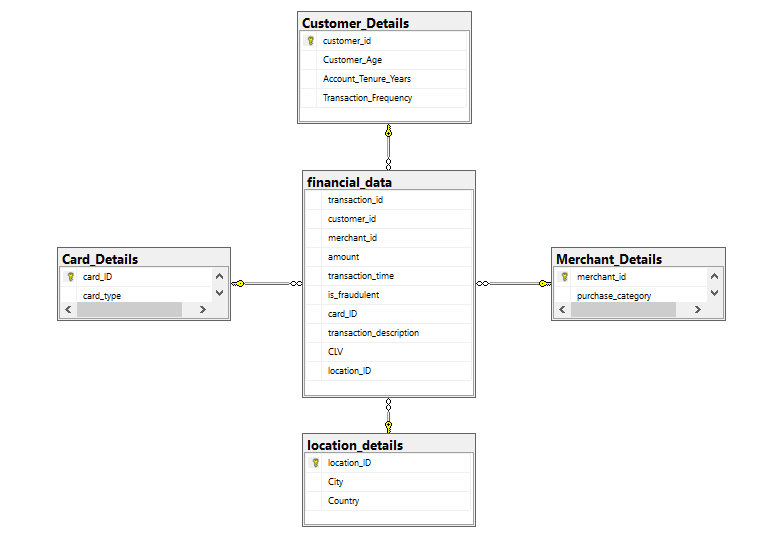
* **Columns**: location\_ID, City, Country
* **Description**: Contains unique city names and their respective countries, providing geographical context to the transactions. This helps in regional analysis of fraud patterns and customer demographics.

**5. Card Details**

* **Columns**: card\_ID, card\_type
* **Description**: Includes card IDs and their corresponding types. This data can be used to evaluate fraud trends based on different card types, offering insights into vulnerabilities associated with specific cards.

# Analysis in SQL

***Database Diagram***



***-- Developing an index on the customer\_id and merchant\_id within the financial\_data table***

CREATE INDEX idx\_customer\_merchant

ON financial\_data (customer\_id, merchant\_id);

***-- 1. Calculating Average Transaction Amount per Merchant***

With AvgAmountPerMerchant as (

Select merchant\_id,AVG(amount) as avg\_amount

from financial\_data

group by merchant\_id

)

select merchant\_id, avg\_amount

from AvgAmountPerMerchant

order by avg\_amount desc;

***-- 2. Retrieving Cities with the Highest CLV (Customer Lifetime Value) for Fraudulent Transactions***

select top 10 ld.City, AVG(fd.CLV) as Average\_CLV

from financial\_data fd

JOIN location\_details ld on fd.location\_ID = ld.location\_ID

where fd.is\_fraudulent = 1

group by ld.City

order by Average\_CLV desc;

***-- 3. Finding top Merchants with the Most Fraudulent Transactions***

select top 5 merchant\_id, COUNT(\*) as fraudulent\_transaction\_count

from financial\_data

where is\_fraudulent = 1

group by merchant\_id

order by fraudulent\_transaction\_count desc;

***-- 4. Showing top 10 Customers by total Transaction Amount***

select distinct top 10

customer\_id,

round(sum(amount) over (partition by customer\_id), 2) as total\_transaction\_amount

from financial\_data

order by total\_transaction\_amount desc;

***-- 5. Categorizing the transactions by amount***

select customer\_id,

count(\*) as transaction\_count,

sum(amount) as total\_amount,

CASE

When sum(amount) < 100000 then 'Low Value'

When sum(amount) between 100000 and 500000 then 'Medium Value'

When sum(amount) > 500000 then 'High Value'

ELSE 'Unknown'

end as transaction\_value\_category

from financial\_data

group by customer\_id

order by total\_amount desc;

***-- 6. Customers with the transactions at Merchants related to "shopping" or "retail"***

select distinct customer\_id

from financial\_data

where merchant\_id IN (

select merchant\_id

from merchant\_details

where purchase\_category LIKE '%Shopping%' or purchase\_category LIKE '%Retail%'

);

***-- 7. Recognizing top Countries with the most fraudulent transactions***

select ld.Country,

count(fd.transaction\_id) as fraud\_count,

round(SUM(fd.amount),2) as total\_fraud\_amount

from financial\_data fd

JOIN location\_details ld

ON fd.location\_ID = ld.location\_ID

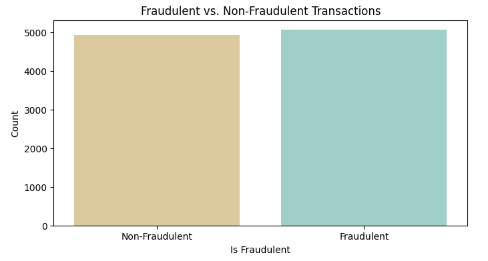
where fd.is\_fraudulent = 1

group by ld.Country

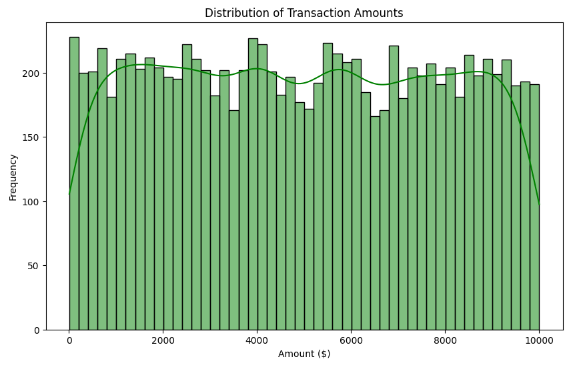
order by fraud\_count desc;

# Analysis in Python

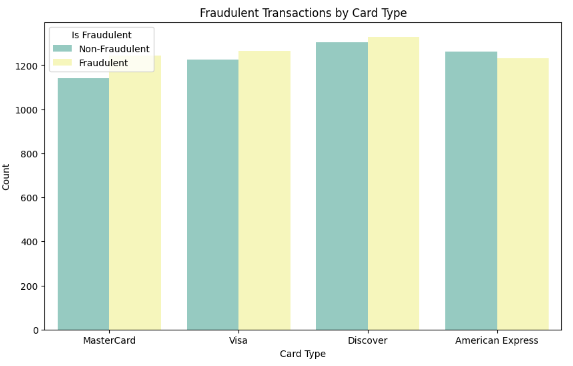
***1. Counts of different types transactions***



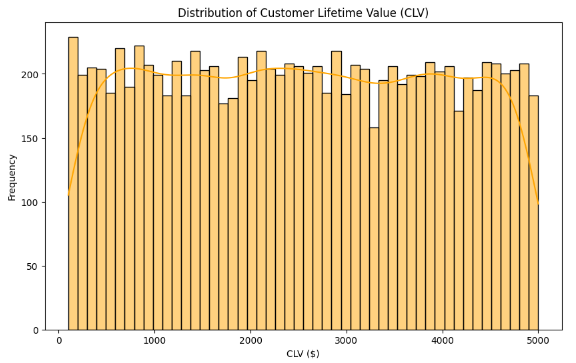
***2. Distribution of the transactions***



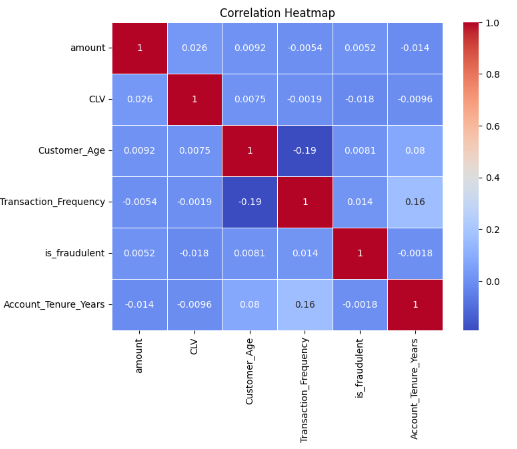
***3. Transactions by card type***



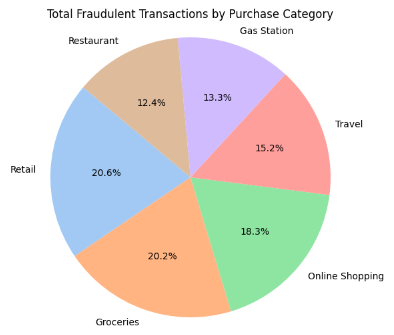
***4. Distribution of CLV***



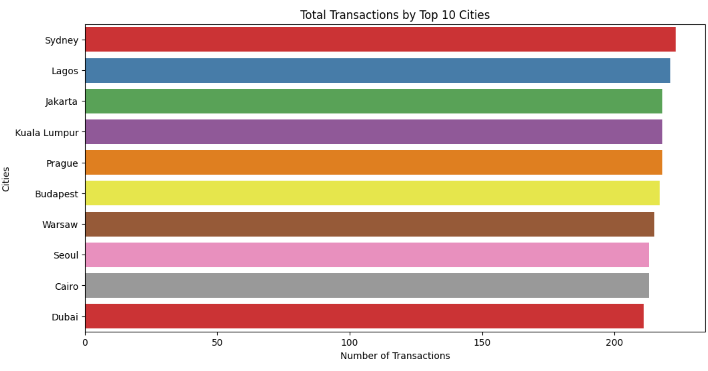
***5. Correlation Heatmap***



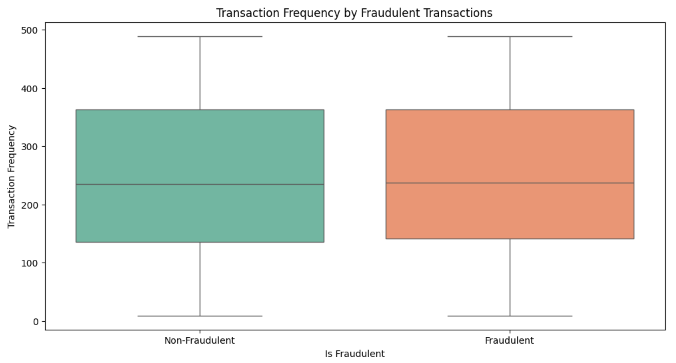
***6. Fraudulent Transactions by Curchase Category***



***7. Transactions by top 10 cities***

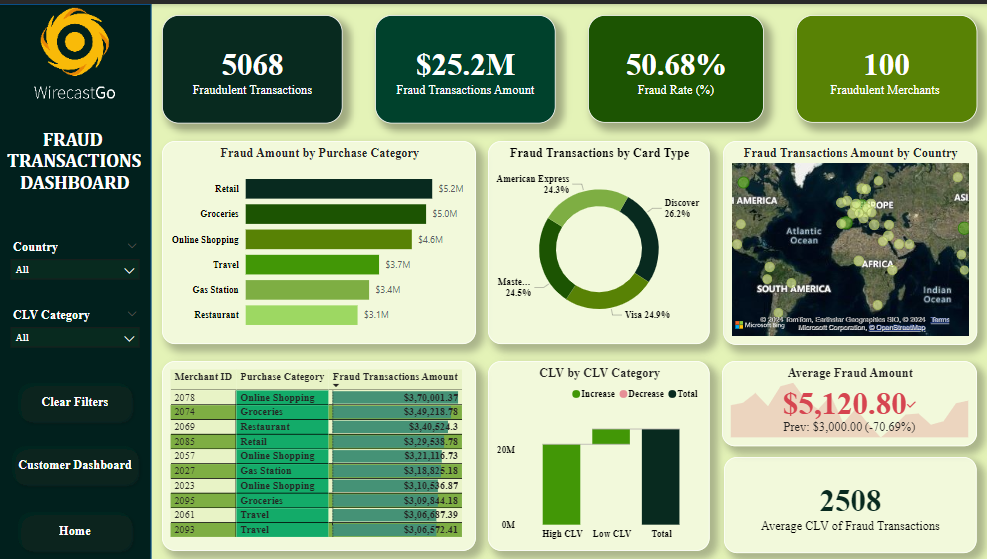


***8. Boxplot for Transaction\_Frequency***

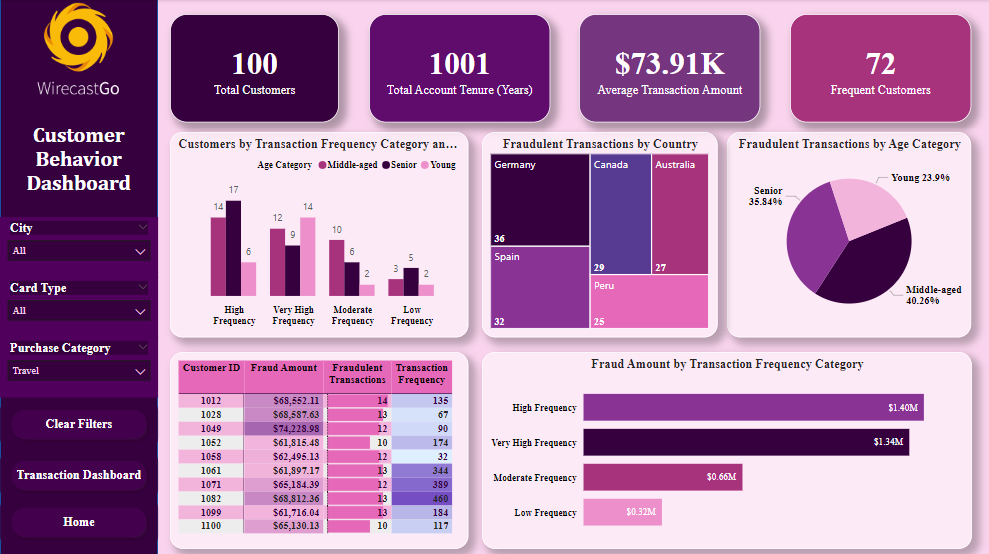


# Analysis in Power BI

  
Home Page



Fraud Transaction Dashboard



Customer Insights Dashboard