FRAUD DETECTION

PROJECT TEAM: GROUP F
By: Komal Sukheja
Syeda Mahrukh
Muhammad Saad

Data Analytics

Contents

Data Dictionary

https://docs.google.com/spreadsheets/d/1vL-u6_PlYqtGehsB3CdZloZTuOWlXik3/edit?usp=sharing&ouid=101103814249375977643&rtpof=true&sd=true

Database Link

https://drive.google.com/drive/folders/1NKbtAarfo9HMEkP6n3Z21fi5h5k8-yRC?usp=sharing

Executive Summary

The project focuses on developing an effective fraud detection system using transaction data to prevent financial fraud. It also aims to segment customers based on age, and transaction history for tailored services. The ultimate goal is to enhance financial security, customer profiles, and data-driven decision-making for businesses.

Objectives

Identify Fraud Patterns:

Analyze the provided data to identify recurring patterns associated with fraudulent transactions.

Customer Analysis:

Analyze customers based on their transaction behavior, characteristics, and historical data.

Suspicious Activity Detection:

Focuses on improving the ability to spot unusual behaviors in financial transactions, helping to quickly identify possible fraudulent actions and reduce potential harm.

Identify correlations:

Cross-reference Fraud indicators with other available data, such as customer profiles and transaction amounts, to identify correlations and build a more comprehensive understanding of potential fraud markers.

Age-based Analysis:

Investigating if certain age groups are more susceptible to fraudulent transactions and understanding the factors contributing to this trend.

Temporal Trends:

Analyzing transaction patterns over time to uncover seasonal trends that could suggest coordinated fraudulent activities.

Key Problems

Identifying Suspicious Activities for Non-Fraudulent Customers:

This helps in identifying potential cases where customers have exhibited suspicious behavior but haven't been marked as fraudulent.

Identifying Suspicious Activities for Flagged Customers:

This can help in detecting repeated suspicious behavior even after flagging.

Percentage of Fraudulent Transaction:

Gives an insight of the prevalence of fraudulent transactions in the dataset.

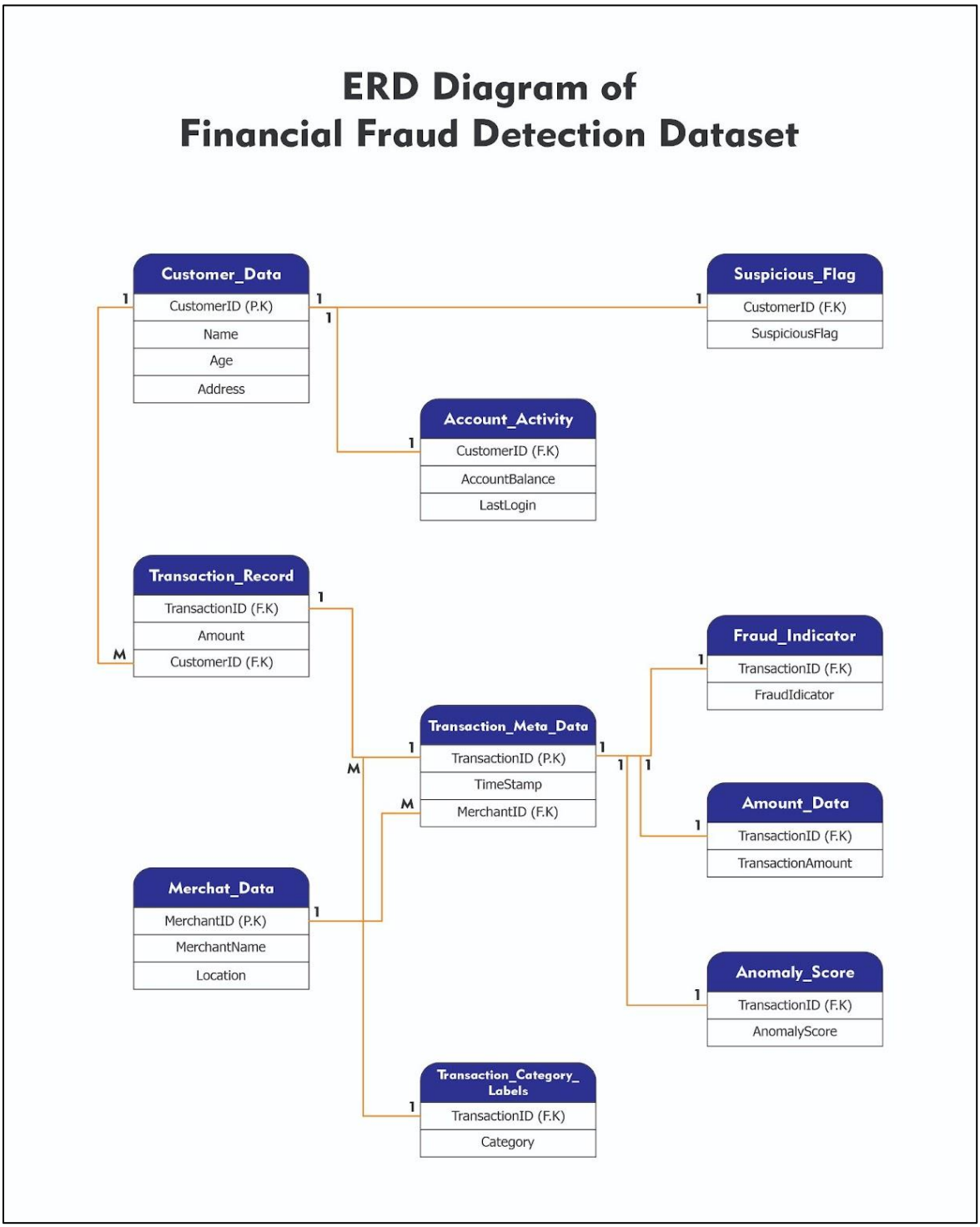
Highest Transaction Amount for specific category:

Useful for understanding the extent of transactions within certain categories.

Prevalence of Fraud on specific days:

Analyze transaction patterns to identify if there are specific days when fraudulent activities are more common.

ER Diagram


ERD Diagram of
Financial Fraud Detection Dataset
Customer_Data
CustomerID (P.K)
Name
Age
Address
Suspicious_Flag
CustomerID (F.K)
SuspiciousFlag
Account_Activity
CustomerID (F.K)
AccountBalance
LastLogin
Transaction_Record
TransactionID (F.K)
Amount
CustomerID (F.K)
Fraud_Indicator
TransactionID (F.K)
FraudIdicator
Transaction_Meta_Data
TransactionID (P.K)
TimeStamp
MerchantID (F.K)
Amount_Data
TransactionID (F.K)
TransactionAmount
Merchat_Data
MerchantID (P.K)
MerchantName
Location
Anomaly_Score
TransactionID (F.K)
AnomalyScore
Transaction_Category_
Labels
TransactionID (F.K)
Category

Data Analysis Process / Approach

- **Data Collection**: Gather relevant transaction records, customer data, and fraud indicators.
- **Data Preparation:** Data was relational and already cleaned, format, and combine datasets for further analysis.
- **Insights and Reporting:** Summarize findings and provide actionable insights

Data Exploration:

• Analyzed fraudulent pattern.

• Explored transaction categories with highest transaction amount.

• Identified top customers and their highest account balance.

• Identified top fraudulent customers.

• Identified customers with suspicious activity but not involved in fraudulent activity.

• Identified customers with their anomaly score.

• Identified fraudulent activities that are associated with merchants.

SQL Queries

Query 1: Identify how many customer's activities are suspicious?

```
select count(*)
from customer_data
join suspicious_activity on customer_data.CustomerID = suspicious_activity.CustomerID
where suspicious_activity.SuspiciousFlag=1;
```

Query 2: Identify how many customers are fraudulent?

```
select count(*)
from customer_data
join suspicious_activity on customer_data.CustomerID = suspicious_activity.CustomerID
join transaction_records on customer_data.CustomerID = transaction_records.CustomerID
join fraud_indicators on transaction_records.TransactionID = fraud_indicators.TransactionID
and fraud_indicators.FraudIndicator=1;
```

Query 2.1: Identify any customers who have a history of suspicious activities but have not been previously flagged as fraudulent?

```
select COUNT(*)
from customer_data
join suspicious_activity on customer_data.CustomerID = suspicious_activity.CustomerID
join transaction_records on customer_data.CustomerID = transaction_records.CustomerID
join fraud_indicators on transaction_records.TransactionID = fraud_indicators.TransactionID
where suspicious_activity.SuspiciousFlag = 1
and fraud_indicators.FraudIndicator=0;
```

Query2.2: Identify any customers who have a history of suspicious activities but been previously flagged as fraudulent?

Solution:

```
select COUNT(*)
from customer_data
join suspicious_activity on customer_data.CustomerID = suspicious_activity.CustomerID
join transaction_records on customer_data.CustomerID = transaction_records.CustomerID
join fraud_indicators on transaction_records.TransactionID = fraud_indicators.TransactionID
where suspicious_activity.SuspiciousFlag = 1
and fraud_indicators.FraudIndicator=1;
```

Query 3-What is the percentage of fraudulent transactions based on the FraudIndicators table?

```
SELECT
 COUNT(*) AS TotalTransactions,
SUM(FraudIndicator) AS FraudulentTransactions,
  (SUM(FraudIndicator) * 100 / COUNT(*)) AS FraudPercentage
FROM
  Fraud_indicators;
```

Query 4: What is the Customer's anomaly Score ?

```
Select
transaction_records.CustomerID,customer_data.Name,round(sum(anomaly_scores.AnomalyScor
e),2) as Customers_Anomaly_Score
from anomaly_scores
join transaction_records on anomaly_scores.TransactionID = transaction_records.TransactionID
join customer_data on transaction_records.CustomerID = customer_data.CustomerID
group by transaction_records.CustomerID,customer_data.Name
order by 1;
```

Query 5: Can we identify transactions with the highest anomaly score and the associated customer and merchant details?

```
select
customer_data.CustomerID as CustomerID,customer_data.Name as
Customer_Name,merchant_data.MerchantName as
Merchant_Name,anomaly_scores.AnomalyScore
from customer_data
join transaction_records on transaction_records.CustomerID = customer_data.CustomerID
join transaction_metadata on transaction_metadata.TransactionID =
transaction_records.TransactionID
join anomaly_scores on transaction_metadata.TransactionID = anomaly_scores.TransactionID
join merchant_data on transaction_metadata.MerchantID = merchant_data.MerchantID
GROUP BY
customer_data.CustomerID,customer_data.Name,merchant_data.MerchantName,anomaly_score
s.AnomalyScore
order by 4 desc;
```

Query 6 - Identify the Age groups of customers who are involved in fraudulent activities.

```
SELECT
                CASE
      WHEN cd.age BETWEEN 0 AND 18 THEN '0-18'
      WHEN cd.age BETWEEN 18 AND 24 THEN '18-24'
      WHEN cd.age BETWEEN 25 AND 34 THEN '25-34'
      WHEN cd.age BETWEEN 35 AND 44 THEN '35-44'
      WHEN cd.age BETWEEN 45 AND 54 THEN '45-54'
      WHEN cd.age BETWEEN 55 AND 64 THEN '55-64'
      ELSE '65+'
   END AS Age_Group,
   COUNT(DISTINCT cd.customerID) AS Number_of_Fraudulent_Customers
   from customer_data cd
   join transaction_records tr ON cd.customerID = tr.customerID
   join transaction_metadata tm ON tr.transactionID = tm.transactionID
   join fraud_indicators fi ON tm.transactionID = fi.transactionID
   where fi.fraudindicator = 1
   GROUP BY Age_groupL
   ORDER BY Age_group;
```

Query 6.1- Is there any correlation between a customer's age and the transaction amount they make? Are older customers more likely to engage in higher value transactions?

```
SELECT
        CASE
      WHEN cd.age BETWEEN 0 AND 18 THEN '0-18'
      WHEN cd.age BETWEEN 18 AND 24 THEN '18-24'
      WHEN cd.age BETWEEN 25 AND 34 THEN '25-34'
      WHEN cd.age BETWEEN 35 AND 44 THEN '35-44'
      WHEN cd.age BETWEEN 45 AND 54 THEN '45-54'
```

```
        WHEN cd.age BETWEEN 55 AND 64 THEN '55-64'
        ELSE '65+'
    END AS Age_Group,
    AVG(ad.transactionAmount) AS Avg_Transaction_Amount,
    sum(ad.transactionAmount) AS Total_Transactions_Made
FROM customer_data cd
 JOIN  transaction_records tr ON cd.customerID = tr.customerID
JOIN  transaction_metadata tm ON tr.transactionID = tm.transactionID
JOIN  amount_data ad ON tm.transactionID = ad.transactionID
 GROUP BY Age_group
 ORDER BY Age_group;
```

Query 6.2- identify the age group that belong to specific category

```
SELECT tcl.Category,
      CASE
         WHEN cd.Age BETWEEN 18 AND 35 THEN 'Young'
         WHEN cd.Age BETWEEN 36 AND 55 THEN 'Adult'
         ELSE 'Old'
      END AS AgeGroup,
      COUNT(*) AS Count
FROM transaction_records tr
JOIN customer_data cd
ON tr.CustomerID = cd.CustomerID
JOIN transaction_category_labels tcl
ON tr.TransactionID = tcl.TransactionID
GROUP BY tcl.Category, AgeGroup
ORDER BY tcl.Category, AgeGroup;
```

OR

```
SELECT
          CASE
        WHEN cd.age BETWEEN 0 AND 18 THEN '0-18'
        WHEN cd.age BETWEEN 18 AND 24 THEN '18-24'
        WHEN cd.age BETWEEN 25 AND 34 THEN '25-34'
        WHEN cd.age BETWEEN 35 AND 44 THEN '35-44'
        WHEN cd.age BETWEEN 45 AND 54 THEN '45-54'
        WHEN cd.age BETWEEN 55 AND 64 THEN '55-64'
        ELSE '65+'
    END AS Age_Group,
    tcl.category AS Transaction_category,
          COUNT(DISTINCT tr.transactionID) AS Number_of_Transactions
    from customer_data cd
    join transaction_records tr ON cd.customerID = tr.customerID
    join transaction_metadata tm ON tr.transactionID = tm.transactionID
    join transaction_category_labels tcl ON tm.transactionID = tcl.transactionID
```

```
GROUP BY Age_group, transaction_category
ORDER BY Age_group, number_of_transactions;
```

Query 7- which customer id has a higher account balance.

```
SELECT CustomerID, AccountBalance
FROM account_activity
ORDER BY AccountBalance DESC
LIMIT 1;
```

Query 8- Which category (food or travel or retail) has done more fraud.

```
SELECT
        tcl.category AS categories,
        sum(ad.transactionamount) AS Fraud_transactionsAmount
 FROM transaction_metadata tm
 JOIN fraud_indicators fi ON tm.TransactionID = fi.transactionID
 JOIN amount_data ad ON tm.TransactionID = ad.transactionID
 JOIN transaction_category_labels tcl ON tm.TransactionID = tcl.transactionID
 WHERE fi.fraudindicator = 1
 GROUP BY categories
 ORDER BY sum(ad.transactionamount) desc;
```

Query 9: What is the highest transaction amount for transactions in the transaction categories?

Solution:

```
select 'Other' as Category,amount_data.TransactionAmount from amount_data
join transaction_category_labels on transaction_category_labels.TransactionID =
amount_data.TransactionID
where amount_data.TransactionAmount = (select MAX(amount_data.TransactionAmount) from
amount_data
                                                                    join
transaction_category_labels on transaction_category_labels.TransactionID =
amount_data.TransactionID
                                                                    where
transaction_category_labels.Category='Other')
union all
select 'Online' as Category,amount_data.TransactionAmount from amount_data
join transaction_category_labels on transaction_category_labels.TransactionID =
amount_data.TransactionID
where amount_data.TransactionAmount = (select MAX(amount_data.TransactionAmount) from
amount_data
                                                                    join
transaction_category_labels on transaction_category_labels.TransactionID =
amount_data.TransactionID
```

```
                                                                where
transaction_category_labels.Category='Online')
union all
select 'Travel' as Category,amount_data.TransactionAmount from amount_data
join transaction_category_labels on transaction_category_labels.TransactionID =
amount_data.TransactionID
where amount_data.TransactionAmount = (select MAX(amount_data.TransactionAmount) from
amount_data
                                                                join
transaction_category_labels on transaction_category_labels.TransactionID =
amount_data.TransactionID
                                                                where
transaction_category_labels.Category='Travel')
union all
select 'Food' as Category,amount_data.TransactionAmount from amount_data
join transaction_category_labels on transaction_category_labels.TransactionID =
amount_data.TransactionID
where amount_data.TransactionAmount = (select MAX(amount_data.TransactionAmount) from
amount_data
                                                                join
transaction_category_labels on transaction_category_labels.TransactionID =
amount_data.TransactionID
                                                                where
transaction_category_labels.Category='Food')
union all
select 'Retail' as Category,amount_data.TransactionAmount from amount_data
join transaction_category_labels on transaction_category_labels.TransactionID =
amount_data.TransactionID
where amount_data.TransactionAmount = (select MAX(amount_data.TransactionAmount) from
amount_data
                                                                join
transaction_category_labels on transaction_category_labels.TransactionID =
amount_data.TransactionID
                                                                where
transaction_category_labels.Category='Retail');
```

Query 10: Are there specific days when fraudulent activities are more prevalent?

```
select a.fraud_day,count(a.fraud_day) as no_of_frauds
from
(select dayname(transaction_metadata.Timestamp) as Fraud_day from transaction_metadata
join fraud_indicators on transaction_metadata.TransactionID = fraud_indicators.TransactionID
where fraud_indicators.FraudIndicator=1) a
group by a.fraud_day;
```

Query 10.1: Are there specific Month when fraudulent activities are more prevalent?

```
select a.fraud_day,count(a.fraud_day) as no_of_frauds
from
(select monthname(transaction_metadata.Timestamp) as Fraud_day from transaction_metadata
join fraud_indicators on transaction_metadata.TransactionID = fraud_indicators.TransactionID
where fraud_indicators.FraudIndicator=1) a
group by a.fraud_day;
```

Query 11-Are there particular transaction amounts that are more frequently associated with fraud?

```
SELECT transaction_records.TransactionID, ROUND(Amount, 1) AS RoundedAmount,
FraudIndicator
FROM transaction_records
JOIN fraud_indicators
ON transaction_records.TransactionID = fraud_indicators.TransactionID
WHERE FraudIndicator = 1
ORDER BY RoundedAmount DESC;
```

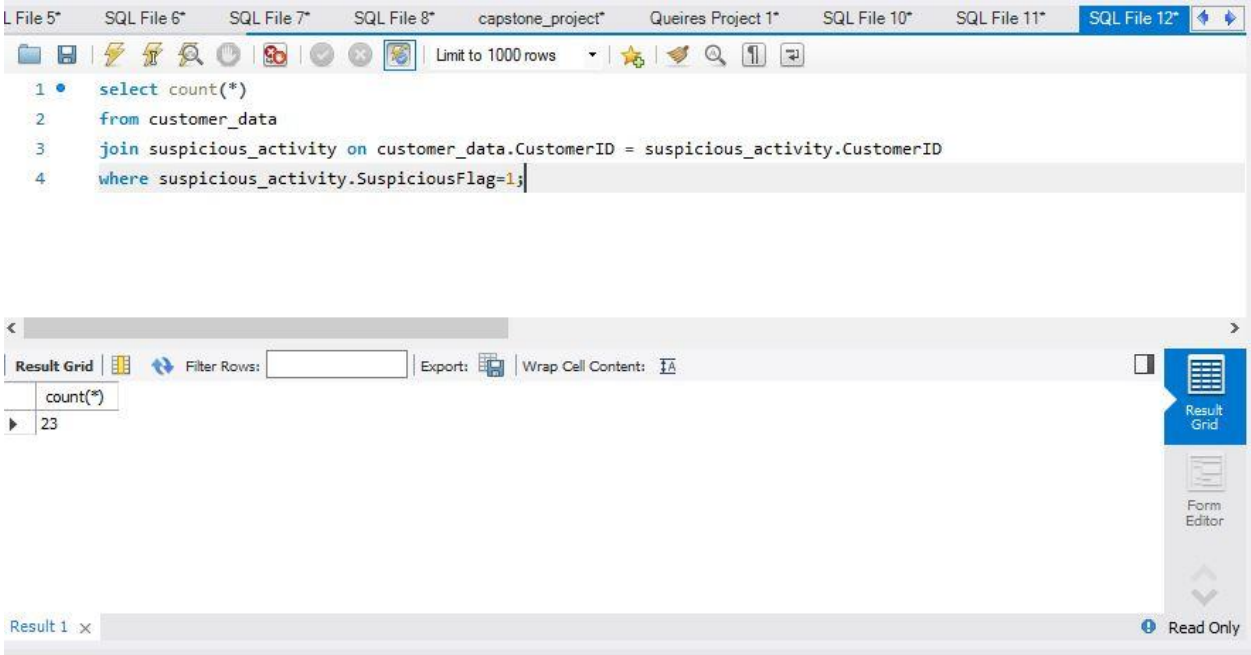
Query 12.-How does the anomaly score distribution vary between different transaction categories?

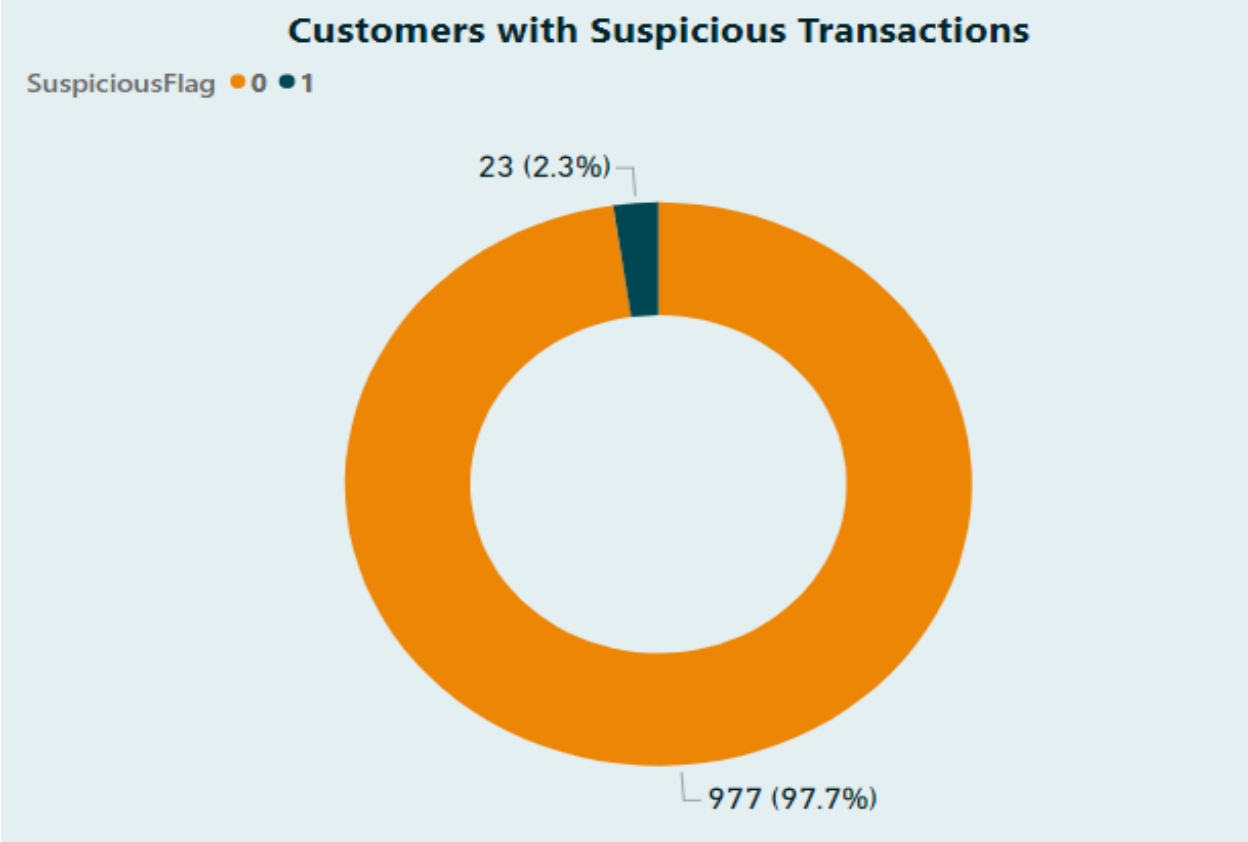
```
SELECT
    tcl.category AS Category,
    ROUND(AVG(sc.anomalyscore),2) AS Avg_anomalyscore,
    ROUND(MIN(sc.anomalyscore),2) AS Min_anomalyscore,
    ROUND(MAX(sc.anomalyscore),2) AS Max_anomalyscore,
    ROUND(STDDEV(sc.anomalyscore),2) AS Stddev_anomalyscore
FROM transaction_category_labels tcl
 JOIN transaction_metadata tm ON tcl.transactionID = tm.transactionID
 JOIN anomaly_scores sc ON tm.transactionID = sc.transactionID
GROUP BY tcl.category
ORDER BY tcl.category;
```

Visualization, Results, and Insights

<u>SQL Query Results and Visualizations</u>

1-Number of customers having suspicious flag are 23



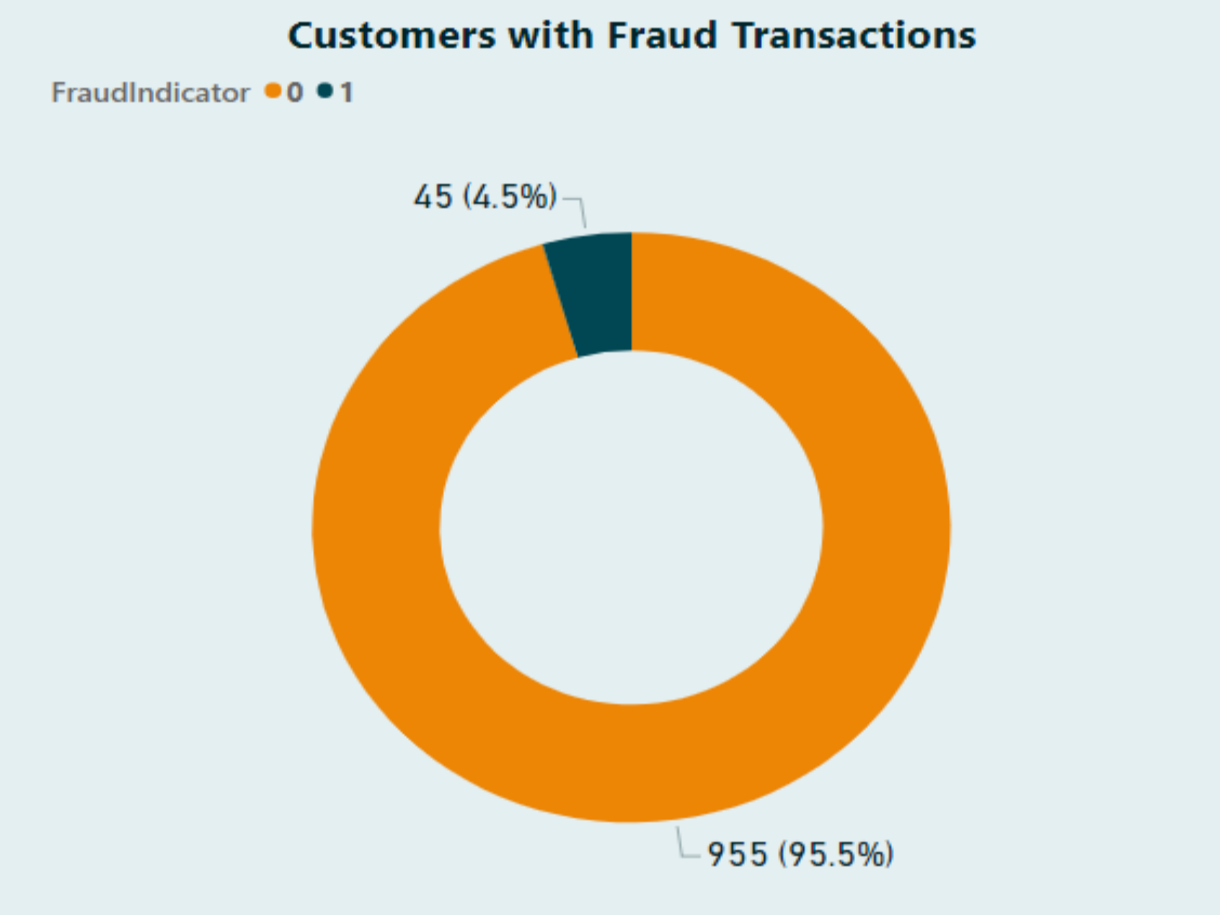


2-Number of customers having fraudulent Indicator are 45

```sql
select count(*)
from customer_data
join suspicious_activity on customer_data.CustomerID = suspicious_activity.CustomerID
join transaction_records on customer_data.CustomerID = transaction_records.CustomerID
join fraud_indicators on transaction_records.TransactionID = fraud_indicators.TransactionID
and fraud_indicators.FraudIndicator=1;
```

Result Grid

count(*)
45



3- Number of customers who have a history of suspicious activities but have not been previously flagged as fraudulent are 22

```
select COUNT(*)
from customer_data
join suspicious_activity on customer_data.CustomerID = suspicious_activity.CustomerID
join transaction_records on customer_data.CustomerID = transaction_records.CustomerID
join fraud_indicators on transaction_records.TransactionID = fraud_indicators.TransactionID
where suspicious_activity.SuspiciousFlag = 1
and fraud_indicators.FraudIndicator=0;
```

COUNT(*)
22

2.2- Number of customers who have a history of suspicious activities but have been previously flagged as fraudulent are 3

```
select COUNT(*)
from customer_data
join suspicious_activity on customer_data.CustomerID = suspicious_activity.CustomerID
join transaction_records on customer_data.CustomerID = transaction_records.CustomerID
join fraud_indicators on transaction_records.TransactionID = fraud_indicators.TransactionID
where suspicious_activity.SuspiciousFlag = 1
and fraud_indicators.FraudIndicator=1;
```

COUNT(*)
3

3- Total Fraud transactions are 45 of 1000, which is 4.5% of the total transactions.

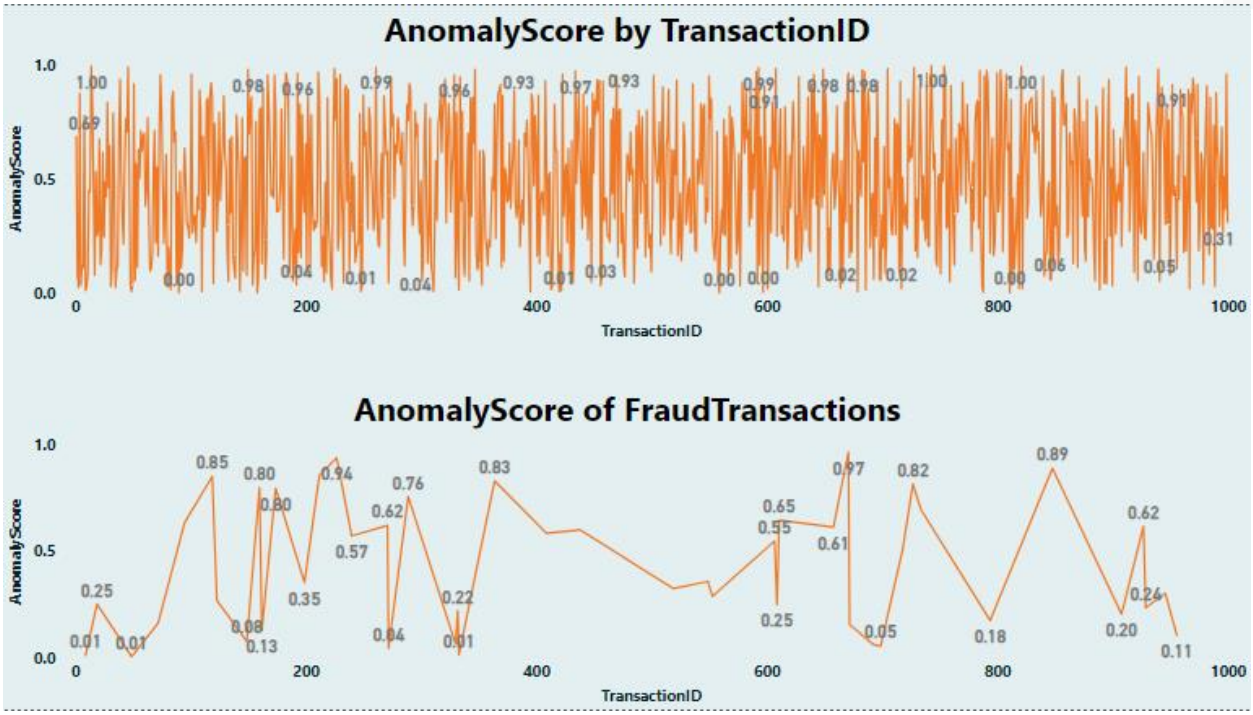
```sql
SELECT
 COUNT(*) AS TotalTransactions,
SUM(FraudIndicator) AS FraudulentTransactions,
  (SUM(FraudIndicator) * 100 / COUNT(*)) AS FraudPercentage
FROM
  Fraud_indicators;
```

TotalTransactions	FraudulentTransactions	FraudPercentage
1000	45	4.5000

4- Anomaly scores of the customers is shown as follows.

```sql
select transaction_records.CustomerID,customer_data.Name,round(sum(anomaly_scores.AnomalyScore),2) as Customers_Anom
from anomaly_scores
join transaction_records on anomaly_scores.TransactionID = transaction_records.TransactionID
join customer_data on transaction_records.CustomerID = customer_data.CustomerID
group by transaction_records.CustomerID,customer_data.Name
order by 1;
```

CustomerID	Name	Customers_Anomaly_Score
1001	Customer 1001	0.03
1003	Customer 1003	0.96
1004	Customer 1004	1.18
1005	Customer 1005	0.66
1007	Customer 1007	0.22
1008	Customer 1008	0.33
1009	Customer 1009	1.26
1012	Customer 1012	1.19



5- Transactions with the highest anomaly score and the associated customerID and merchant ID.

L File 5* | SQL File 6* | SQL File 7* | SQL File 8* | capstone_project* | Queires Project 1* | SQL File 10* | SQL File 11* | SQL File 12*

Limit to 1000 rows

```
select
customer_data.CustomerID as CustomerID,customer_data.Name as Customer_Name,merchant_data.MerchantName as Merchant
from customer_data
join transaction_records on transaction_records.CustomerID = customer_data.CustomerID
join transaction_metadata on transaction_metadata.TransactionID = transaction_records.TransactionID
join anomaly_scores on transaction_metadata.TransactionID = anomaly_scores.TransactionID
join merchant_data on transaction_metadata.MerchantID = merchant_data.MerchantID
GROUP BY customer_data.CustomerID,customer_data.Name,merchant_data.MerchantName,anomaly_scores.AnomalyScore
order by 4 desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

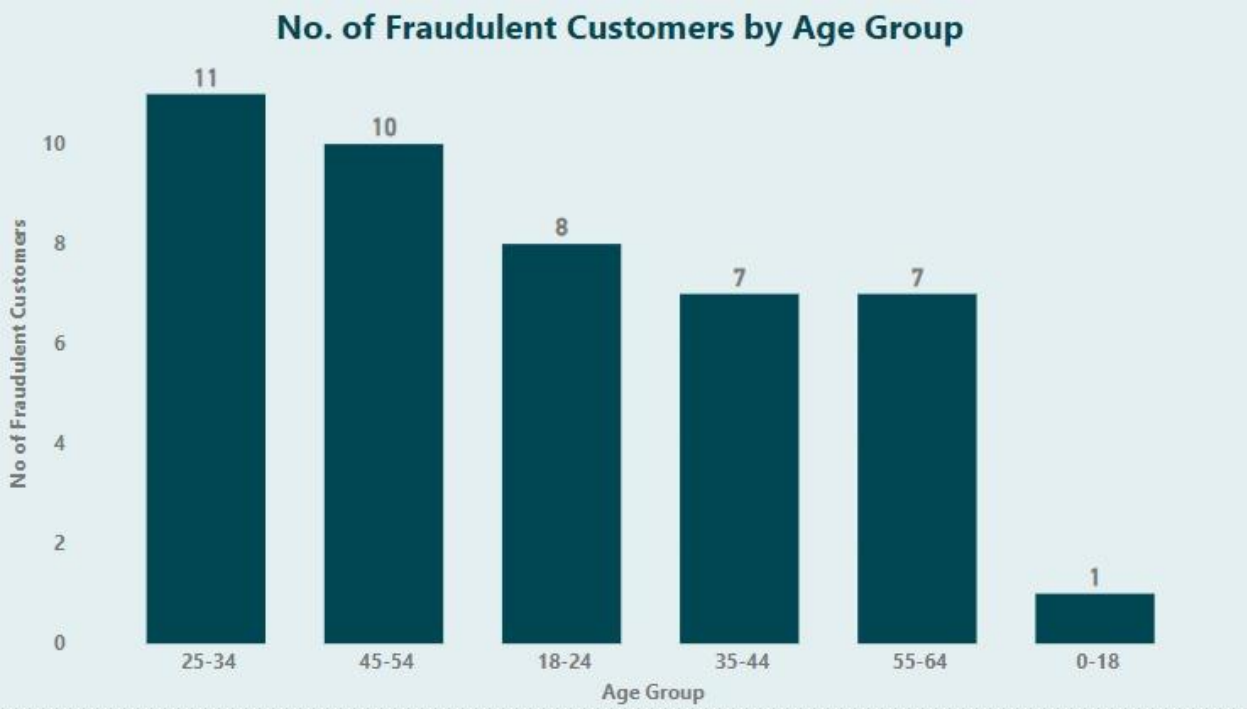
CustomerID	Customer_Name	Merchant_Name	AnomalyScore
1521	Customer 1521	Merchant 2803	0.9990473412377049
1700	Customer 1700	Merchant 2892	0.9973396329012194
1577	Customer 1577	Merchant 2889	0.997038134220019
1625	Customer 1625	Merchant 2827	0.9955617683625665
1238	Customer 1238	Merchant 2274	0.9923378496402676
1747	Customer 1747	Merchant 2044	0.991923320101168
1558	Customer 1558	Merchant 2911	0.9916263388452975

Result 9 | Result Grid | Form Editor | Read Only

6- Age groups of customers who are involved in fraud. 25-34 age group has the highest fraud customers.

```
SELECT
        CASE
        WHEN cd.age BETWEEN 0 AND 18 THEN '0-18'
        WHEN cd.age BETWEEN 18 AND 24 THEN '18-24'
        WHEN cd.age BETWEEN 25 AND 34 THEN '25-34'
        WHEN cd.age BETWEEN 35 AND 44 THEN '35-44'
        WHEN cd.age BETWEEN 45 AND 54 THEN '45-54'
        WHEN cd.age BETWEEN 55 AND 64 THEN '55-64'
        ELSE '65+'
```

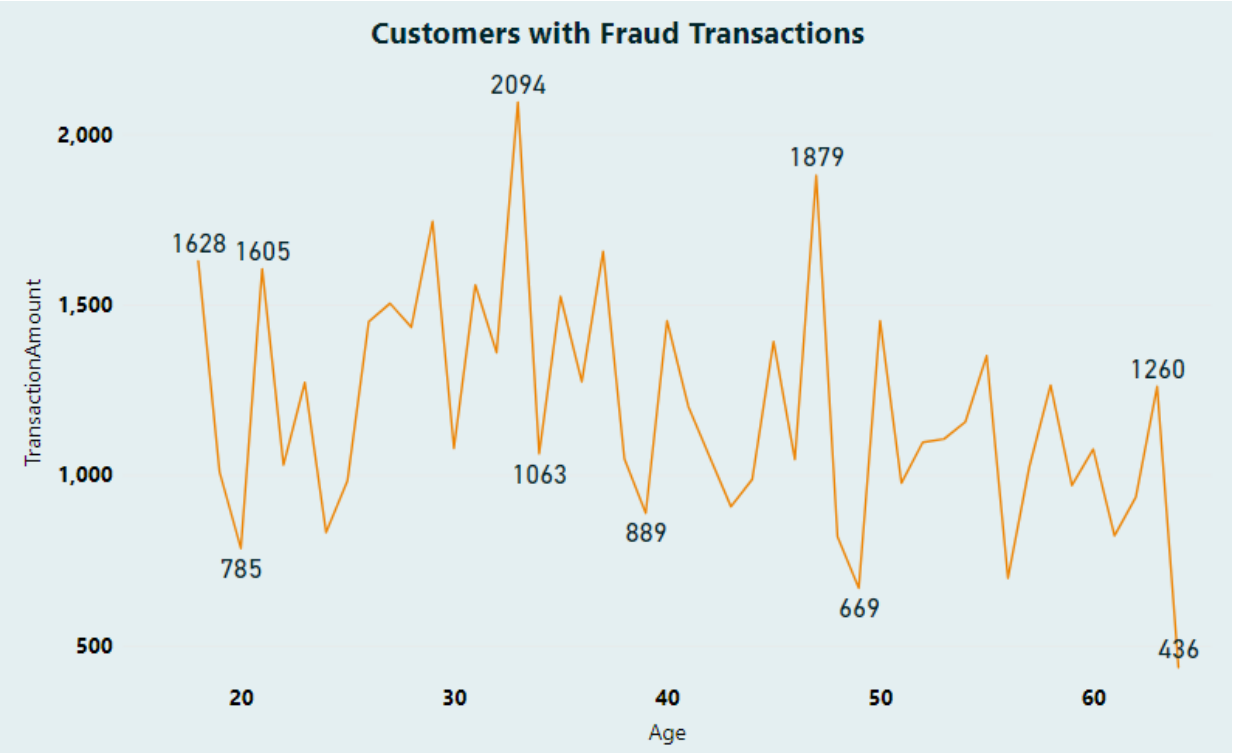
Age_Group	Number_of_Fraudulent_Customers
0-18	1
18-24	8
25-34	11
35-44	7
45-54	10
55-64	7



6.1- There is not significant correlation between a customer's age and the transaction amount they make. However, customers aged 25 and above have higher value transactions.

```sql
        WHEN cd.age BETWEEN 18 AND 24 THEN '18-24'
        WHEN cd.age BETWEEN 25 AND 34 THEN '25-34'
        WHEN cd.age BETWEEN 35 AND 44 THEN '35-44'
        WHEN cd.age BETWEEN 45 AND 54 THEN '45-54'
        WHEN cd.age BETWEEN 55 AND 64 THEN '55-64'
        ELSE '65+'
    END AS Age_Group,
    AVG(ad.transactionAmount) AS Avg_Transaction_Amount,
    sum(ad.transactionAmount) AS Total_Transactions_Made
```

Age_Group	Avg_Transaction_Amount	Total_Transactions_Made
0-18	56.14531224455756	1628.2140550921692
18-24	55.8451554113317	6533.883183125809
25-34	58.48049015184528	14269.23959705025
35-44	54.76674744437088	11993.917690317223
45-54	54.43206362367478	11594.029551842728
55-64	55.250234903393014	9834.541812803956



6.2- Number of transactions for Age groups in each transaction category.

```sql
SELECT
CASE
        WHEN cd.age BETWEEN 0 AND 18 THEN '0-18'
        WHEN cd.age BETWEEN 18 AND 24 THEN '18-24'
        WHEN cd.age BETWEEN 25 AND 34 THEN '25-34'
        WHEN cd.age BETWEEN 35 AND 44 THEN '35-44'
        WHEN cd.age BETWEEN 45 AND 54 THEN '45-54'
        WHEN cd.age BETWEEN 55 AND 64 THEN '55-64'
        ELSE '65+'
```

Age_Group	Transaction_category	Number_of_Transactions
0-18	Online	4
0-18	Retail	4
0-18	Other	5
0-18	Travel	7
0-18	Food	9
18-24	Retail	19
18-24	Travel	20

7- Customer ID 1357 has the highest account balance among all.

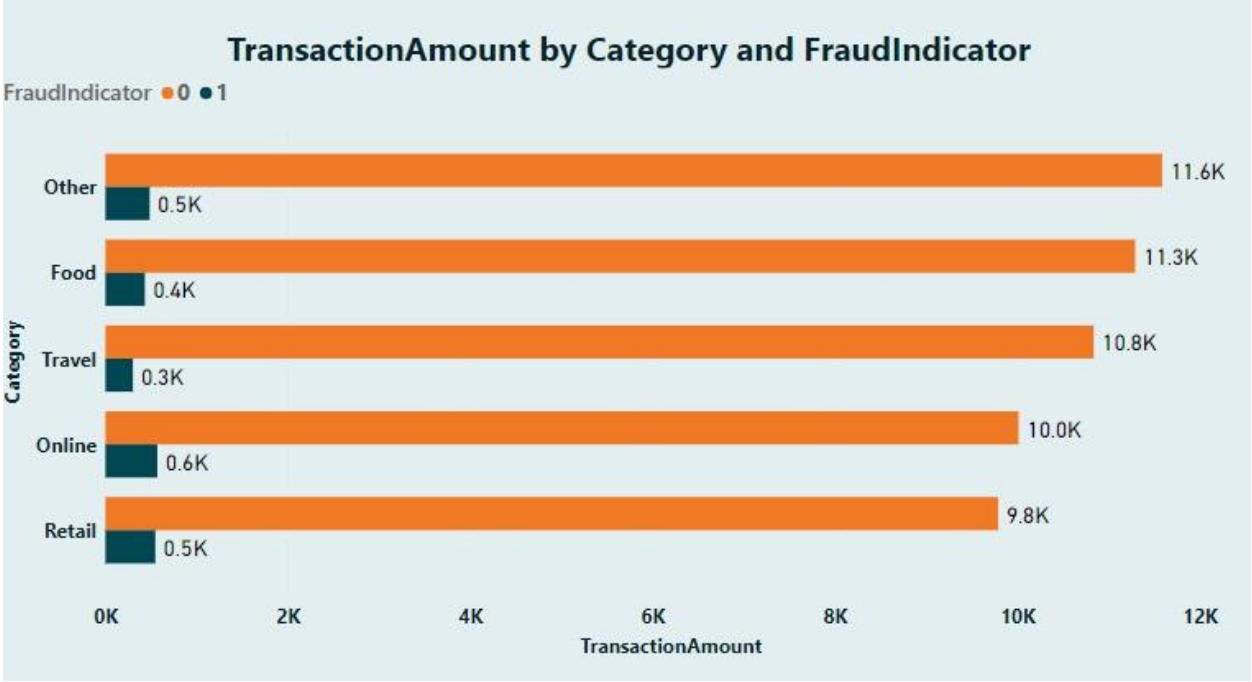
```sql
SELECT CustomerID, AccountBalance
FROM account_activity
ORDER BY AccountBalance DESC
LIMIT 1;
```

CustomerID	AccountBalance
1357	9999.776238940984

8- There are 5 transaction categories, where online and retail has the highest amount of fraud transactions

```sql
tcl.category AS categories,
    sum(ad.transactionamount) AS Fraud_transactionsAmount
FROM transaction_metadata tm
JOIN fraud_indicators fi ON tm.TransactionID = fi.transactionID
JOIN amount_data ad ON tm.TransactionID = ad.transactionID
JOIN transaction_category_labels tcl ON tm.TransactionID = tcl.transactionID
WHERE fi.fraudindicator = 1
GROUP BY categories
ORDER BY sum(ad.transactionamount) desc;
```

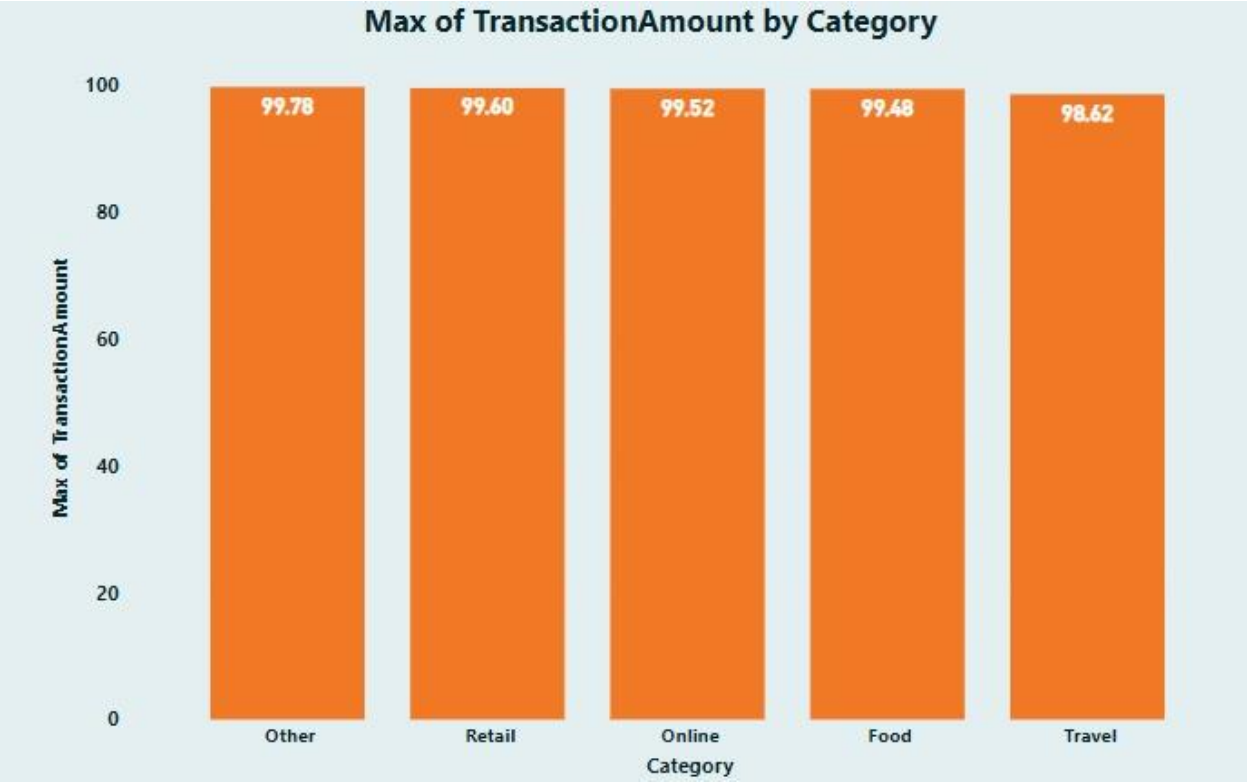
categories	Fraud_transactionsAmount
Online	568.4689777954313
Retail	546.7322044866474
Other	484.34769444357033
Food	431.84680071297856
Travel	301.14039952830274



9- There is no significant difference between highest transaction amount of each category. Retail and Online categories have Highest transaction amount of 99.60 and 99.52 whereas, travel has least 98.52 transaction amount.

```sql
where amount_data.TransactionAmount = (select MAX(amount_data.TransactionAmount) from amount_data
                                       join transaction_category_labels on transaction_category_labels.Transaction
                                       where transaction_category_labels.Category='Food')
union all
select 'Retail' as Category,amount_data.TransactionAmount from amount_data
join transaction_category_labels on transaction_category_labels.TransactionID = amount_data.TransactionID
where amount_data.TransactionAmount = (select MAX(amount_data.TransactionAmount) from amount_data
                                       join transaction_category_labels on transaction_category_labels.Transaction
                                       where transaction_category_labels.Category='Retail');
```

Category	TransactionAmount
Other	99.78432334104625
Online	99.52235672534624
Travel	98.62431306543853
Food	99.47675323072761
Retail	99.60103489989679



10- 38 of 45 frauds are reported in January. And there are very few in February.

L File 5* | SQL File 6* | SQL File 7* | SQL File 8* | capstone_project* | Queires Project 1* | SQL File 10* | SQL File 11* | SQL File 12*

Limit to 1000 rows

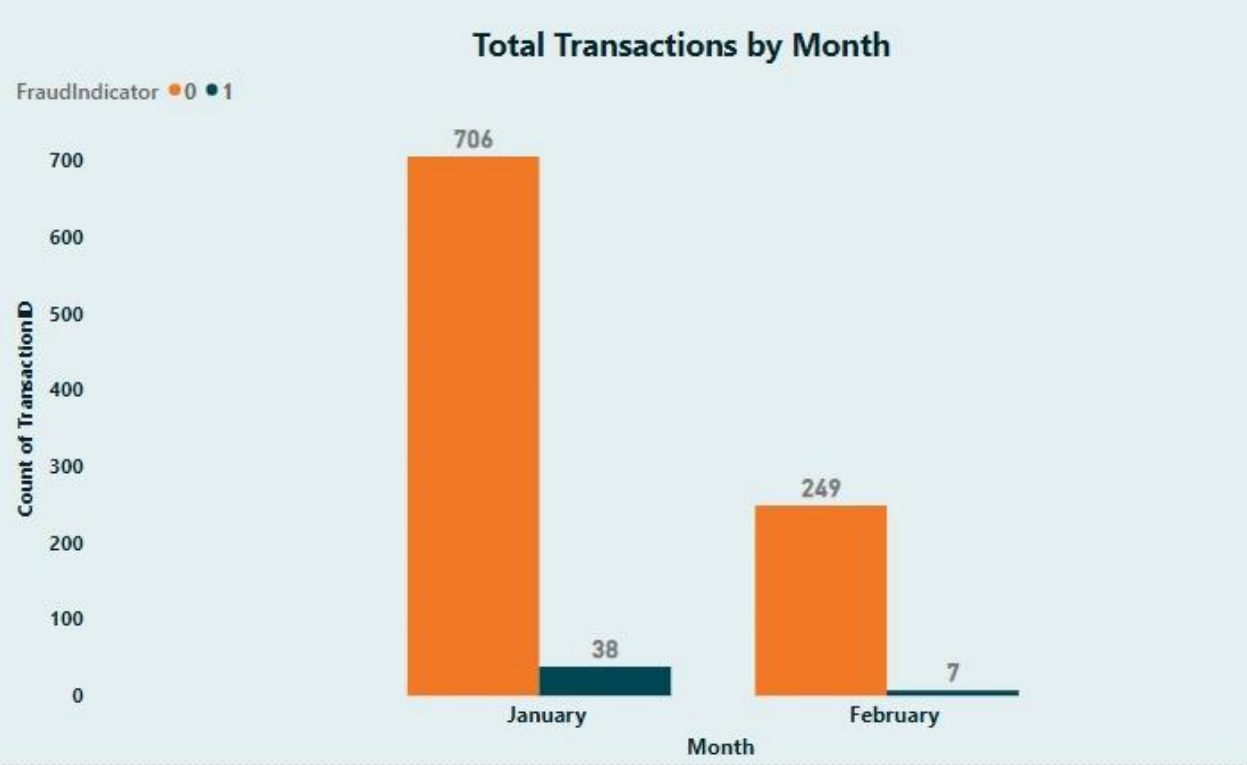
```
select a.fraud_day,count(a.fraud_day) as no_of_frauds
from
(select monthname(transaction_metadata.Timestamp) as Fraud_day from transaction_metadata
join fraud_indicators on transaction_metadata.TransactionID = fraud_indicators.TransactionID
where fraud_indicators.FraudIndicator=1) a
group by a.fraud_day;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Fraud_day	no_of_frauds
January	38
February	7

Result Grid | Form Editor

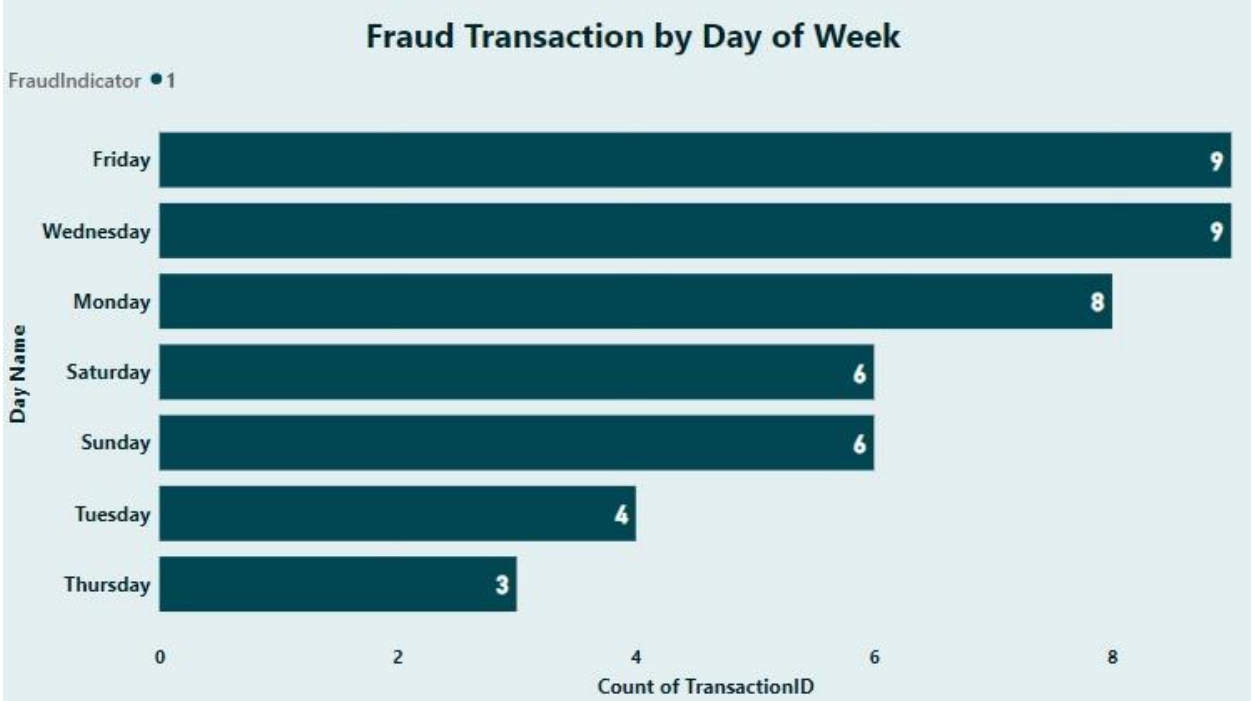
Result 18 | Read Only



10.1- From the days of the week, Monday, Wednesday and Friday have a greater number of frauds.

```
select a.fraud_day,count(a.fraud_day) as no_of_frauds
from
(select dayname(transaction_metadata.Timestamp) as Fraud_day from transaction_metadata
join fraud_indicators on transaction_metadata.TransactionID = fraud_indicators.TransactionID
where fraud_indicators.FraudIndicator=1) a
group by a.fraud_day;
```

Fraud_day	no_of_frauds
Saturday	6
Monday	8
Tuesday	4
Wednesday	9
Thursday	3
Friday	9
Sunday	6



11- TransactionIDs that are associated with fraud are shown as follows:

```
SELECT transaction_records.TransactionID, ROUND(Amount, 1) AS RoundedAmount, FraudIndicator
FROM transaction_records
JOIN fraud_indicators
ON transaction_records.TransactionID = fraud_indicators.TransactionID
WHERE FraudIndicator = 1
ORDER BY RoundedAmount DESC;
```

TransactionID	RoundedAmount	FraudIndicator
160	99.2	1
174	97.1	1
289	96.2	1
330	95.8	1
672	95.2	1
927	93.2	1
848	91.5	1

12- Distribution of anomaly scores between different categories is shown

- Online and travel categories have the highest average anomaly scores
- Online category has the highest standard deviation anomaly scores.

```
        ROUND(AVG(sc.anomalyscore),2) AS Avg_anomalyscore,
        ROUND(MIN(sc.anomalyscore),2) AS Min_anomalyscore,
        ROUND(MAX(sc.anomalyscore),2) AS Max_anomalyscore,
        ROUND(STDDEV(sc.anomalyscore),2) AS Stddev_anomalyscore
FROM transaction_category_labels tcl
 JOIN transaction_metadata tm ON tcl.transactionID = tm.transactionID
 JOIN anomaly_scores sc ON tm.transactionID = sc.transactionID
GROUP BY tcl.category
ORDER BY tcl.category;
```

Category	Avg_anomalyscore	Min_anomalyscore	Max_anomalyscore	Stddev_anomalyscore
Food	0.49	0	1	0.29
Online	0.5	0.01	0.99	0.3
Other	0.48	0.01	0.99	0.27
Retail	0.49	0	1	0.29
Travel	0.5	0	1	0.28

Insights

- **There is not any significant relation between the customers flagged as suspicious and being involved in fraud.**
- **The age group 25-34 are more likely to be involved in fraudulent activities. There is also one minor fraud customer.**
- **No significant correlation between a customer's age and the transaction amount they make.**
- **Online has the highest amount of fraud transactions, and the retail category has the second highest and then the other categories. Travel has the least fraud transactions.**
- **Notable spikes in fraud occurrences are observed on Monday, Wednesday, and Friday, suggesting these days are more susceptible to fraudulent activities.**
- **Fraudulent transactions make up about 4.18% of total Transaction Amount while 95.82% is legitimate.**

Recommendations

1- **Targeted Monitoring of Age Group 25-34:**
Given that the age group 25-34 appears to be more likely to be involved in fraudulent activities, allocate extra resources for monitoring and investigating transactions within this age range. However, avoid overgeneralization and ensure that legitimate customers within this age group are not unduly inconvenienced.

2- **Strict Transaction Monitoring for Under-18 Compliance**
Implement rigorous age verification procedures and closely monitor transactions to uphold age-related regulations effectively and prevent irregular activities for customers under 18.

3- **Elevate Security for Online Transactions:**
Since online transactions are associated with the highest number of fraud incidents, strengthen security measures for online transactions. Implement multi-factor authentication, real-time transaction monitoring to prevent fraudulent online activities effectively.

4- **Focus on Retail Category Monitoring:**
Recognizing that the retail category has the second highest number of fraud transactions, intensify monitoring and verification procedures for retail related transactions. This could involve additional checks on high-value retail transactions.

5- **Increased Vigilance on Mondays, Wednesdays, and Fridays in January:**
Given that most frauds occur on Mondays, Wednesdays, and Fridays in the month of January, allocate additional resources and monitoring efforts during these specific time periods. Implement real-time alerts for potentially suspicious transactions on these days.

6- **Regular Data Analysis and Feedback Loop:**
Establish a regular data analysis and feedback loop to continually refine your fraud prevention strategies. Regularly review and update your fraud detection patterns and emerging threats.