"Real-Time Anomaly Detection for Bank Transactions"

Project Description

This project focuses on building a data-driven fraud detection system using a synthetic yet realistic banking transactions dataset. It involves importing, cleaning, and analyzing transactional data stored in a MySQL database to uncover anomalous and suspicious activities. The project utilizes SQL (including window functions and CTEs), exploratory data analysis (EDA), and logic-based rules to flag potential frauds.

The core objective is to simulate real-world banking operations and apply analytical techniques to proactively detect fraudulent behavior such as unusually high transfers, frequent small-value withdrawals, or inconsistent activity across customer accounts.

Problem Statement

Modern digital banking systems handle millions of transactions daily, making it difficult to manually detect fraudulent behavior. The challenge lies in identifying unusual or suspicious transaction patterns that may indicate fraud — without raising too many false alarms.

In this project, the goal is to:

- Analyze customer transaction patterns.
- Identify indicators of potential fraud (e.g., sudden spikes in withdrawal amount, transactions from distant locations, failed attempts).
- Flag anomalous transactions using rule-based logic and analytical SQL queries.
- Provide actionable insights for fraud prevention and auditing.

Banking Database consists of following table:

- **Fraudtransactions**
    - **Transaction_ID**: Unique reference number for each transaction
    - **Customer_ID**: Identifier of the customer
    - **Account_ID**: Customer's bank account number
    - **Transaction_Date**: Date of the transaction
    - **Transaction_Time**: Time of the transaction
    - **Amount**: Amount transacted

- **Transaction_Type**: Credit or Debit
- **Description**: Purpose or context of the transaction
- **Transaction_Mode**: Channel used for transaction
- **Status**:Status of the transaction
- **Location**: Location from where transaction was made

Exploratory Data Analysis (EDA)

- Finding Data Types of all columns in a database table

  ```
  SELECT COLUMN_NAME, DATA_TYPE
  from Information_schema.columns
  where table_name="fraudtransactions" and table_schema="bankingdb";
  ```

COLUMN_NAME	DATA_TYPE
Transaction_ID	bigint
Customer_ID	int
Account_ID	bigint
Transaction_Date	text
Transaction_Time	text
Amount	double
Transaction_Type	text
Description	text
Transaction_Mode	text
Status	text
Location	text

- Total number of transactions

  ```
  select count(*) as Total_Transactions
  from fraudtransactions;
  ```

Total_Transactions
231

- Status of transactions

  ```
  select Status,count(*) as Total_Transactions
  from fraudtransactions
  group by Status
  order by Total_Transactions desc;
  ```

Status	Total_Transactions
Completed	228
Failed	3

- Top 5 Customers based on total transactions

```
select customer_id,count(*) as Total_Transactions
from fraudtransactions
group by customer_id
order by Total_Transactions desc
limit 5;
```

customer_id	Total_Transactions
101	72
104	55
102	53
103	51

- Count of transactions based on location

```
select location,count(*) as Total_Transactions
from fraudtransactions
group by location
order by Total_Transactions desc;
```

location	Total_Transactions
Mumbai	67
Hyderabad	59
Bangalore	56
Delhi	49

- Count of transactions based on mode of transaction

```
select transaction_mode,count(*) as Total_Transactions
from fraudtransactions
group by transaction_mode
order by Total_Transactions desc;
```

transaction_mode	Total_Transactions
POS	67
Internet Banking	63
ATM	52
Mobile Banking	49

- Available Balance per customer

```
select customer_id,
round(sum(case
        when transaction_type="Credit" then amount
   when transaction_type="Debit" then -amount
   else 0
   end),2) as Available_Balance
   from fraudtransactions
   group by customer_id
   order by Available_Balance desc;
```

customer_id	Available_Balance
101	8737.95
102	3382.45
104	-8439.63
103	-12819.06

Fraud Transaction Detection

1. Abnormal High-Value Transactions

Flag transactions that are more than 1.5x the customer's average.

```
with customer_avg as
(select Customer_ID,round(avg(Amount),2) as Avg_Customer_Amount
        from fraudtransactions
   group by customer_ID)

   select *
   from fraudtransactions t
   join customer_avg c
   on t.customer_id=c.customer_id
   where t.amount > 2*c.Avg_Customer_Amount and transaction_type="Debit";
```

Transaction_ID	Customer_ID	Account_ID	Transaction_Date	Transaction_Time	Amount	Transaction_Type	Description	Transaction_Mode	Status	Location	Customer_ID	Avg_Customer_Amount
273665994394	101	987654321001	2025-12-16	18:19:00	3779.3	Debit	ATM Cash Out	ATM	Completed	Bangalore	101	1761.03
670799496139	101	987654321001	2025-12-20	06:51:00	3593.45	Debit	Online Shopping	Internet Banking	Completed	Hyderabad	101	1761.03
713249868174	101	987654321001	2025-07-19	09:03:00	4639.94	Debit	Subscription Fee	Internet Banking	Completed	Hyderabad	101	1761.03
829851334750	101	987654321001	2025-03-22	14:57:00	4185.35	Debit	ATM Withdrawal	ATM	Completed	Bangalore	101	1761.03
963972176017	101	987654321001	2025-04-02	11:20:00	3933.69	Debit	Mobile Recharge	Mobile Banking	Completed	Bangalore	101	1761.03
591979363867	102	987654321002	2025-12-08	18:07:00	4940.67	Debit	Grocery Shopping	POS	Completed	Hyderabad	102	2456.22
885903361820	104	987654321004	2025-09-14	05:08:00	4792.84	Debit	ATM Withdrawal	ATM	Completed	Bangalore	104	2349.34
715509967407	104	987654321004	2025-06-29	10:35:00	4841.71	Debit	Mobile Payment	Mobile Banking	Completed	Mumbai	104	2349.34
364286565492	104	987654321004	2025-12-13	08:06:00	4850.98	Debit	Restaurant Bill	POS	Completed	Delhi	104	2349.34

2. Transactions at Odd Hours

```
select *
from fraudtransactions
```

where hour(transaction_time) between 0 and 4 and transaction_type="Debit";

Transaction_ID	Customer_ID	Account_ID	Transaction_Date	Transaction_Time	Amount	Transaction_Type	Description	Transaction_Mode	Status	Location
893195777447	101	987654321001	2025-11-02	02:10:00	1985.36	Debit	Bill Payment	Internet Banking	Completed	Mumbai
612866779068	101	987654321001	2025-04-08	03:13:00	574.96	Debit	Subscription Fee	Internet Banking	Completed	Hyderabad
835316694200	101	987654321001	2025-11-01	01:21:00	3134.97	Debit	Subscription Fee	Internet Banking	Completed	Bangalore
176726811657	101	987654321001	2025-04-25	00:48:00	3334.16	Debit	Online Shopping	Internet Banking	Completed	Bangalore
226750596909	101	987654321001	2025-01-20	03:42:00	3023.2	Debit	Online Shopping	Internet Banking	Completed	Hyderabad
490596101232	102	987654321002	2025-12-23	00:40:00	4297.99	Debit	Restaurant Bill	POS	Completed	Mumbai
622657187359	102	987654321002	2025-04-08	02:55:00	1282.64	Debit	Retail Purchase	POS	Completed	Delhi
356355366221	102	987654321002	2025-05-30	01:06:00	1234.23	Debit	Restaurant Bill	POS	Completed	Mumbai
694854998490	102	987654321002	2025-08-13	04:14:00	2396.62	Debit	Restaurant Bill	POS	Completed	Bangalore

3. Multiple Failed Transactions by a Customer

```
select DISTINCT Customer_ID, Failed_Transactions
from (
   select Customer_ID,status,
      COUNT(*) OVER (PARTITION BY Customer_ID) AS Failed_Transactions
   FROM fraudtransactions
   WHERE status = 'Failed'
) s
where Failed_Transactions > 2;
```

Customer_ID	Failed_Transactions
102	3

4. Multiple Transactions Within the hour

```
with Txns as (
   select *,
        COUNT(*) OVER (PARTITION BY Customer_ID, Transaction_Date,
HOUR(Transaction_Time)) AS txn_count
   from fraudtransactions
)
select *
FROM Txns
WHERE txn_count > 3;
```

Transaction_ID	Customer_ID	Account_ID	Transaction_Date	Transaction_Time	Amount	Transaction_Type	Description	Transaction_Mode	Status	Location	txn_count
955712084340	101	987654321001	2025-04-25	12:30:00	300	Debit	Retail Purchase	POS	Completed	Mumbai	20
237588413911	101	987654321001	2025-04-25	12:30:00	300	Debit	Grocery Shopping	POS	Completed	Mumbai	20
845638847947	101	987654321001	2025-04-25	12:30:00	300	Debit	Restaurant Bill	POS	Completed	Mumbai	20
274716738176	101	987654321001	2025-04-25	12:30:00	300	Debit	Grocery Shopping	POS	Completed	Mumbai	20
184881801482	101	987654321001	2025-04-25	12:30:00	300	Debit	Retail Purchase	POS	Completed	Mumbai	20
680411980930	101	987654321001	2025-04-25	12:30:00	300	Debit	Restaurant Bill	POS	Completed	Mumbai	20
982671983350	101	987654321001	2025-04-25	12:30:00	300	Debit	Retail Purchase	POS	Completed	Mumbai	20
545119726986	101	987654321001	2025-04-25	12:30:00	300	Debit	Restaurant Bill	POS	Completed	Mumbai	20
581028394033	101	987654321001	2025-04-25	12:30:00	300	Debit	Restaurant Bill	POS	Completed	Mumbai	20

5. Description vs Mode Mismatch (e.g., ATM Withdrawal via Internet Banking)

```
select *
from fraudtransactions
where (description LIKE '%ATM%' AND Transaction_Mode NOT LIKE '%ATM%')
  OR (description LIKE '%Mobile%' AND Transaction_Mode NOT LIKE '%Mobile
Banking%')
  OR (description LIKE '%Bill%' AND Transaction_Mode NOT IN ('Internet Banking',
'Mobile Banking','POS'));
```

Transaction_ID	Customer_ID	Account_ID	Transaction_Date	Transaction_Time	Amount	Transaction_Type	Description	Transaction_Mode	Status	Location
180972272366	103	987654321003	2025-04-27	10:00:00	800	Debit	ATM Withdrawal	Internet Banking	Completed	Bangalore

6. Location Inconsistency on Same Day

```
With location as (
   select
      Customer_ID,
      transaction_date,
      transaction_type,
      COUNT(DISTINCT location) as location_count
   from fraudtransactions
   group by Customer_ID, Transaction_Date, transaction_type
   having COUNT(distinct location) > 1
)
select t.*
from fraudtransactions t
join location l
ON t.Customer_ID = l.Customer_ID AND t.transaction_Date = l.transaction_Date
order by Customer_ID;
```

Transaction_ID	Customer_ID	Account_ID	Transaction_Date	Transaction_Time	Amount	Transaction_Type	Description	Transaction_Mode	Status	Location
176726811657	101	987654321001	2025-04-25	00:48:00	3334.16	Debit	Online Shopping	Internet Banking	Completed	Bangalore
490462379884	101	987654321001	2025-04-25	12:30:00	300	Debit	Retail Purchase	POS	Completed	Mumbai
581028394033	101	987654321001	2025-04-25	12:30:00	300	Debit	Restaurant Bill	POS	Completed	Mumbai
181850729653	101	987654321001	2025-04-25	12:30:00	300	Debit	Retail Purchase	POS	Completed	Mumbai

Recommendations to Reduce Fraud Transactions

1. **Real-Time Monitoring**
    - Use streaming tools to detect anomalies instantly (e.g., high-value, odd-hour, frequent transactions).
2. **Transaction Limits**

- Set dynamic thresholds based on customer history; verify or block high-value transfers.

3. **Multi-Factor Authentication (MFA)**

    - Enforce MFA for high-risk scenarios (late-night, new device, location changes).

4. **Geo-Fencing**

    - Flag or block transactions from unusual or inconsistent locations.

5. **Failed Transaction Controls**

    - Lock accounts or trigger alerts after multiple failed attempts.

6. **Mode & Description Consistency**

    - Validate that transaction purpose matches the mode (e.g., ATM withdrawal not via internet banking).

7. **Customer Behavior Profiling**

    - Track and compare transactions against normal patterns (time, amount, location).

8. **Internal Auditing**

    - Regularly audit high-risk accounts, channels, and locations.

9. **Customer Awareness**

    - Send alerts and tips when suspicious activity is detected.