“SQL-Based Banking System for Loan Management, Fraud Detection,and Data Analytics”

Name : prajwal Sandip kamble

Email : prajwalkamble9657@gmail.com

Mobile no : 9370570881

Table of Contents :

1. Introduction :

This project provides a database-driven solution for banking challenges, focusing on loan management. SQL operations track overdue payment, calculate interest, and generate reports. It supports fraud detection, customer segmentation, and regulatory compliance, ensuring efficient data management and analysis.

2. Objectives :

- Automate loan tracking and repayment analysis.
- Provide real-time updates on account balances.
- Support regulatory reporting and fraud prevention.

---

3. Tools and Technologies :

The project uses the following tools:

- SQL
- Database Management System (MySQL)

---

4. Database Design :

Database name : pk

The database includes the following key tables:
- Customers: Stores personal and demographic information of customers

- Account: Tracks the accounts owned by customers and their balances

- Transactions: Records all account-related transactions

- Loans: Manages loan details and repayment status.
- ATM: Tracks ATM locations and cash availability.

- Customer support: Logs customer complaints and support requests.

Tables :

Customers :

- Fields: Customers (CustomerID, Name, Age, Income, ContactNumber, Address).
- Purpose: Stores personal and demographic information of customers.

Account :

- Fields: Accounts (AccountID, CustomerID, AccountType, Balance)
- Purpose: Tracks the accounts owned by customers and their balances

---

Transactions :

- Fields: Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)
- Purpose: Records all account-related transactions

Loan :

- Fields : Loans (LoanID, CustomerID, LoanAmount, InterestRate, LoanStartDate, LoanStatus)
- Purpose: Manages loan details and repayment status.

atm :

- Fields : Loans (LoanID, CustomerID, LoanAmount, InterestRate, LoanStartDate, LoanStatus)
- Purpose: Tracks ATM locations and cash availability.

Customre support :

- Fields : Customer_Support (SupportTicketID, CustomerID, IssueDate, IssueDescription, ResolutionStatus)
- Purpose: Logs customer complaints and support requests.

5. Queries and Functionalities :

1. Identify overdue loans beyond the repayment period.
2. Calculate interest for active loans.
3. Retrieve transaction history for accounts.
4. Segment loans by status (Active, Overdue, Paid).
5. Detect high-value transactions for fraud analysis.
6. Group customers by income and account balance.
7. Generate monthly transaction volume reports.
8. Track ATM withdrawals and cash availability.
9. Monitor unresolved customer complaints.
10. Analyze ross-selling opportunities for high-income customers.

problems faced in banking :

---

1. Fraud Detection

- **Problem**: Identifying fraudulent transactions among millions of records.
- **SQL Tasks**:
    - Write queries to detect unusual transaction patterns.
    - Use conditions to identify transactions exceeding certain thresholds.

---

2. Customer Segmentation

- **Problem**: Categorizing customers based on account balances, transactions, or loan repayments.
- **SQL Tasks**:
    - Create reports grouping customers by income level or spending habits.

---

3. Loan Management

- **Problem**: Tracking overdue loans or calculating interest on active loans.
- **SQL Tasks**:
  - Query overdue payments or generate monthly repayment schedules.

---

4. Account Balances and Transactions

- **Problem**: Managing real-time updates to balances after deposits, withdrawals, or transfers.
- **SQL Tasks**:
  - Automate balance updates using triggers.
  - Example:

---

5. Regulatory Reporting

- **Problem**: Generating reports for compliance with financial regulations.
- **SQL Tasks**:
  - Aggregate data to show transaction volumes or customer account details.

---

6. Customer Support Insights

- **Problem**: Resolving customer complaints and tracking service requests.
- **SQL Tasks**:
  - Store and query support tickets for analysis.

---

7. ATM Management

- **Problem**: Ensuring ATMs have sufficient cash and analyzing withdrawal patterns.
- **SQL Tasks**:
  - Track ATM transactions and calculate cash levels.

---

8. Data Security and Privacy

- **Problem**: Preventing unauthorized access and masking sensitive data.
- **SQL Tasks**:
  - Use views or restricted access queries for secure reporting.

---

9. Revenue Analysis

- **Problem**: Analyzing interest income or fees collected from services.
- **SQL Tasks**:
    - Write queries to calculate monthly or annual revenue.

---

10. Cross-Selling Opportunities

- **Problem**: Identifying customers for targeted offers (e.g., loans, credit cards).
- **SQL Tasks**:
    - Query customers with specific criteria for marketing campaigns.

---

6. Database Design :

Database name : psk

The database includes the following key tables:
- Transactions (TransactionID, CustomerID, TransactionDateTim, TransactionAmount , MerchantID, TransactionType, DeviceIP, ATMID, Location, BeneficiaryName, AccountStatus

---

7. Queries and Functionalities :

The project implements the following SQL queries for fraud detection:

---

common fraud cases :

1. Unusual Transaction Amounts

- **Case**: Transactions with unusually high amounts compared to the customer's regular activity.

2. Multiple Transactions in Short Time

- **Case**: Multiple transactions within a short period, often just below the limit requiring extra verification.

3. Location-Based Fraud

- **Case**: Transactions from geographically distant locations in a short time (impossible travel).

4. Suspicious Merchant Activity

- **Case**: Repeated payments to unknown or flagged merchants.

5. Account Takeover

- **Case**: Transactions from new or unverified devices or IP addresses.

6. Dormant Account Activity

- **Case**: Transactions from accounts that were inactive for a long time.

7. ATM Skimming

- **Case**: Withdrawals from multiple ATMs in different locations using the same card details.

8. High-Frequency Transactions

- **Case**: Large numbers of small transactions to avoid detection.

---

9. Reversal Abuse

- **Case**: Requesting multiple transaction reversals or chargebacks to exploit the system.

---

10. Mismatched Account Holder Details

- **Case**: Payments or transfers to accounts with mismatched beneficiary details.

---

8. Implementation :

1. Set up the database schema as provided in the SQL script.
2. Populate tables with sample data.
3. Execute queries to analyze loan performance and detect issues.

---

9. Conclusion :

This project demonstrates the application of SQL in managing and analyzing banking data. It provides actionable insights for loan management and supports efficient decision-making.

10. References :

1. SQL Documentation.
2. Online Tutorials on Banking Systems and SQL
3. Academic References on Database Management.
