

F-41

TZ BANKING SYSTEM PROJECT

By Safarov Islom

INTRODUCTION

This banking system is being created to automate and optimize digital financial services. The main goal of the project is to provide customers with the opportunity to carry out convenient, secure and fast financial transactions.



BACKGROUND

- 1. Effective user management
- 2. Card and transaction management
- 3. Security and fraud detection
- 4. Reporting and monitoring
- 5. Special services (VIP)

BUSINESS MODEL AND REVENUE SOURCES

BUSINESS NEEDS AND CHALLENGES

- 1. Transaction fees
- 2. VIP services
- 3. Credit and lending services
- 4. Automated payment services
- 5. Fraud detection service

BENEFITS FOR BUSSINES

1. Fast transactions – customers can transfer money in seconds.
2. Security – fraud prevention by blocking and monitoring suspicious transactions.
3. Analysis and settlement – providing financial analysis to the bank and users.
4. Convenient services for customers – increasing customer loyalty through automatic payments, cashback, VIP services.
5. Increasing the efficiency of the bank – reducing overhead costs by digitizing all processes.

ABOUT DATA

I used python to insert data into these tables because it saves a lot of time and is an easier way to generate random data. I used various libraries for this, for connecting to sql, creating values, and so on. I only inserted data manually into the users and cards tables, and the rest of the tables were filled with triggers and stored procedures. In addition to generating data in python, I also used it to execute stored procedures, because writing and executing a single query in sql itself takes a lot of time, and inserting random data makes the data random, so all the values in my tables are related to each other just like in real life.

TRIGGERS

- What is Triggers in MS SQL Server?

In Microsoft SQL Server, a trigger is a special type of stored procedure that executes automatically in response to specific database events.

- How they help us in banking system?

Triggers provide automated security and auditing by instantly recording transaction history and blocking illegal account changes. They ensure no balance is updated without a permanent, traceable record.

- The Triggers I created and used:
 1. `trg_UpdateTotalBalance` - to update balance of user after creating card, making transfer and else.
 2. `trg_Fraud_150M` - to control large transactions.
 3. `trg_AutoVIP` - to give VIP status automatically.
 4. `trg_WelcomeBonus` - to give Welcome Bonus for new users.
 5. `trg_VelocityCheck` - to Block suspicious transactions.
 6. `trg_SavingPlan_5Percent` - to save automatically after transaction.
 7. `trg_AuditLogs_StatusChange` - Audit of transaction status.
 8. `trg_AnomalousActivity` - for unusual movement analysis.

STORE PROCEDURES

- What is SP in MS SQL Server?

A SP is a saved block of SQL code that you can reuse over and over. In banking, it's like a "pre-set routine" that handles tasks—like transferring money—safely and efficiently in a single call.

- Why we need them in Banking?

SPs provide security and consistency by grouping complex transactions into a single, pre-approved command that prevents errors and hacking. They ensure that if a transfer starts, it either finishes completely or doesn't happen at all.

- The SPs I created and used:
 1. `sp_MakeTransfer` - to insert data into transactions tbl.
 2. `sp_AtmOperation` - for deposit and withdrawal.
 3. `sp_ProcessScheduledPayments` - for generate scheduled payments.
 4. `sp_EmergencyBlock` - for emergency blocking cards.
 5. `sp_VerifyAndTransfer_2FA` - for verify and transfer a large transactions.
 6. `sp_ApplyLoan` - to applying loans.
 7. `sp_TransferToHeirs` - for transferring wealth to heirs.
 8. `sp_ApplyCashback` - to transfer/give cashbacks.
 9. `sp_GenerateReports` - to generate reports (daily, weekly, monthly).

VIEWS

- What is and why we need Views in Banking?

A View is a virtual table based on a saved query that shows specific data without giving access to the underlying tables. In banking, they are used to hide sensitive info (like SSNs) while letting staff see only what they need, such as account balances. We use them to make various business decisions based on the results.

- Views i created and used:
 1. `vw_UserFinancialStatus` - User financial portrait
 2. `vw_FraudulentActivities` - Suspicious transaction monitoring
 3. `vw_BankDailySummary` - Daily bank turnover
 4. `vw_BlockedUserDetails` - Audit of blocked users
 5. `vw_ActiveLoans` - Active loans and debts
 6. `vw_CardUsageAnalytics` - Card usage analysis
 7. `vw_VipCustomerService` - VIP customer service
 8. `vw_UpcomingPayments` - Payments in a week
 9. `vw_UserMonitoring` - User Monitoring (With Block History)
 10. `vw_CustomerRetentionRisk` - Customer Retention Risk

CONCLUSION

This banking system is aimed at automating innovative financial services. This project will create a convenient and secure financial ecosystem for users, while expanding the possibilities for earning money for the bank. Also, with the work done in the process of this project, young and novice data analysts can improve their skills and learn a lot, for example, they can strengthen their skills by studying automation, visualization, data analysis, etc.

F-4|

THANK YOU

By Safarov Islom