Banking System DataBase

- **Project Title:** Banking System DataBase
- **Student Name**: Meron Goitom
- **Student Id**: 9225226654
- **Github username**: merongoitom

Checkpoint #1	Date Submitted
Checkpoint#1	02/20/2024

Section II: Table Contents

Section III: Project Description

- Banking Database Management System

The motivation behind creating this database system is to streamline banking operations, enhance customer service, and improve data management efficiency. Traditional banking processes often involve manual paperwork, redundant data entry, and limited accessibility to customer information. By implementing a centralized database system, we aim to address these challenges by digitizing processes, enabling real-time access to customer data, and automating routine tasks. This database system will also help comply with regulatory requirements, analyze customer behavior, and identify new revenue generation opportunities.

Our database system is designed to serve as a comprehensive platform for managing all aspects of banking operations. It consists of multiple interconnected modules that handle customer management, account operations, transaction processing, loan management, and reporting. The system utilizes a relational database model to organize and store data efficiently, ensuring data integrity and consistency. With user-friendly interfaces and intuitive workflows, bank staff can easily navigate the system to perform tasks such as account opening, transaction processing, and customer support. The system also incorporates robust security measures to protect sensitive customer information and prevent unauthorized access.

- Unique Features

1. Advanced Analytics: It also manages the system and integrates powerful analytics tools that leverage customer data to generate insights and predictive models. By analyzing transaction patterns, customer behavior, and market trends, banks can identify opportunities for personalized offerings, risk mitigation, and revenue optimization.

2. Omni-Channel Integration: Unlike traditional banking systems, our database system offers seamless integration across multiple channels, including online banking, mobile banking, ATMs, and branch offices. Customers can initiate transactions through any channel and experience consistent service quality, enhancing their overall banking experience.

3. Automated Compliance: Our database system incorporates automated compliance checks and reporting features to ensure adherence to regulatory requirements such as KYC (Know Your Customer), AML (Anti-Money Laundering), and GDPR (General Data Protection Regulation). By automating compliance processes, banks can reduce manual errors, minimize regulatory risks, and streamline audit procedures.

- Software Tools Benefiting from Our Database System:

1. Salesforce Financial Services Cloud: Salesforce Financial Services Cloud is a popular CRM platform banks use to manage customer relationships and sales activities. By integrating with our database system, Financial Services Cloud can access real-time customer data, transaction history, and account information, enabling personalized interactions, targeted marketing campaigns, and improved customer engagement.

2. Oracle Financial Services Analytical Applications (OFSAA): OFSAA is a comprehensive suite of analytical applications designed for financial institutions to perform risk management, profitability analysis, and regulatory reporting. By leveraging data from our database system, OFSAA can enhance its analytical capabilities, enabling banks to gain deeper insights into their operations, optimize capital allocation, and meet regulatory compliance requirements more effectively.

Section IV

● Functional Database Requirements

1. Customer

1.1. A customer shall have an address.

1.2. A customer shall have a multiple loan

1.3. A customer shall be able to transfer funds between their accounts.

1.4. A customer can have one or more accounts

1.5. A customer shall be able to view their savings balance

1.6. A customer shall be able to view their checking balance

1.7. A customer shall be able to make loan payments.

1.8. Many customers belong to one branch but one branch belongs to many customers.

2. Account

2.1. An account shall have a unique account ID

2.2. Account shall be able to be created by multiple customers

2.3. An account shall have many transactions.

2.4. An account shall be able to receive deposits from customers.

2.5. An account shall have multiple loans.

2.6. An account shall have a unique customer ID

3. Branch

3.1. A branch shall have many employees.

3.2. A branch shall serve multiple customers.

3.3.A branch shall provide multiple services like loan processing, transactions, and credit cards.

3.4 A branch shall have multiple ATMs.

4. Loan

4.1. A loan shall have a unique loan ID

4.2. A loan shall have an account ID

4.3. A loan shall have an amount.

4.4 . A loan shall have multiple transactions.

4.5.A loan shall be created by multiple customers.

5. Employee

5.1. Employee shall have a unique employee ID

5.2. Employee shall have many credit cards

5.3. Employees shall manage multiple customers.

5.4. Employee shall work in multiple branches.

5.5. An employee shall have a manager who is also an employee.

5.6.An employee shall be assigned as a bank teller and accountant but not advising

6.Credit Card

6.1. A credit card shall have a unique account ID

6.2. A credit card shall have a limited amount.

6.3. A credit card shall have multiple transactions

6.4 . A credit card shall have a customer ID

6.5. A credit card shall be created by multiple accounts.

6.6.A credit card shall be created by multiple customers.

7. Transaction

7.1. A transaction shall have a type indicating whether it is a deposit, withdrawal transfer, or payment.

7.2. A transaction shall be able to be associated with a specific account whether checking or saving

7.3. A transaction shall update the account balance accordingly.

7.4 . A transaction shall have one unique transaction ID.

8. ATM

8.1. An ATM shall be linked to one branch.

8.2. An ATM shall allow multiple transactions.

8.3. An ATM shall allow customers to withdraw cash, check balances, and deposit

Funds.

8.4. An ATM shall be able to process the account balance.

9. Bank

9.1. A bank shall be able to have many customers

9.2. A bank should be able to have many branches.

9.3. A bank shall be able to have many employees.

9.4 A bank shall be able to have multiple employees also managers.

10.Saving Account

10.1. Savings accounts shall be able to have a specific account ID.

10.2. A savings account shall be able to have multiple transactions.

10.3 A savings account shall be able to have a customer ID.

10.3. saving account shall be able created by multiple customers.

11. Checking Account

11.1. A checking account shall have a specific account ID.

11.2.A checking account shall have multiple transactions.

11.3.A checking account shall have a customer ID.

11.4. A checking account shall be able created by many customers.

12. Interest Rate

12.1. The interest rate shall be able to have an amount.

12.2 interest rate shall be able to have a unique id

13.Address

13.1. The address shall be able to have a unique customer ID

13.2. The address shall be able to have an account ID.

14. Account Type

14.1 Account type shall be able to have a unique account ID.

15 . Credit Score

15.1. Credit score should be able to have account ID

15.2. Credit score shall be able to have customer name.

16. Account Status

16.1. customer shall be able to look at their account status.

17. Check

17.1. Check shall be able to have a saving account ID

17.2 Check should be able to have a checking account ID.

Section V

- ## Non-Functional Database Requirements

1. Performance:

1.1. Response Time: The system shall respond to user queries and transactions within 2 seconds on average.

1.2. Transaction Throughput: The database system shall support a minimum of 1000 transactions per second during peak hours.

1.3. Batch Processing: Batch processing jobs shall be completed within 1 hour for daily reconciliation and reporting tasks.

1.4. Indexing: Indexes shall be optimized to minimize query execution time and improve overall system performance.

1.5. Data Retrieval: Queries involving complex joins and aggregations shall return results within 5 seconds to ensure timely access to information.

2. Security

2.1. Access Control: Role-based access control (RBAC) shall be implemented to restrict access to sensitive data based on user roles and privileges.

2.2. Encryption: Data encryption shall be enforced for all sensitive information stored in the database, including customer credentials and financial transactions.

2.3. Audit Trail: The system shall maintain a comprehensive audit trail of database activities, including user logins, modifications, and access attempts.

2.4. Intrusion Detection: Intrusion detection and prevention systems (IDPS) shall be deployed to monitor database traffic and detect suspicious activities or unauthorized access attempts.

2.5. Data Masking: Sensitive data such as account numbers and social security numbers shall be masked in database queries and reports to protect confidentiality.

3. Scalability:

3.1. Horizontal Scaling: The database architecture shall support horizontal scaling by adding more database nodes to distribute the workload and accommodate growth.

3.2. Load Balancing: Load balancing mechanisms shall be implemented to evenly distribute database requests across multiple servers for optimal performance.

3.3. Sharding: Data sharding techniques shall be employed to partition large datasets across multiple servers to improve scalability and performance.

3.4. Auto-scaling: The database system shall support auto-scaling capabilities to dynamically adjust resource allocation based on demand to handle fluctuating workloads.

3.5. Partitioning: Data partitioning strategies shall be implemented to divide data logically and distribute it across storage devices for efficient management and scalability.

4. Capability:

4.1. Data Integrity: The database system shall enforce data integrity constraints to ensure the accuracy and consistency of stored information.

4.2. Backup and Recovery: Regular backups shall be performed to safeguard data integrity, and robust recovery procedures shall be in place to restore data in case of failures.

4.3. Replication: Data replication shall be implemented to create redundant copies of critical data for high availability and disaster recovery purposes.

4.4. Concurrency Control: Concurrency control mechanisms shall be employed to manage concurrent access to data and prevent data corruption or inconsistencies.

4.5. Compliance: The database system shall comply with industry regulations such as GDPR, PCI DSS, and SOX to protect customer data and ensure regulatory compliance.

5. Environmental:

5.1. Energy Efficiency: The data center hosting the database servers shall be designed for energy efficiency, utilizing energy-efficient hardware and cooling systems.

5.2. Cooling Systems: The data center shall be equipped with efficient cooling systems to maintain optimal operating temperatures and minimize environmental impact.

5.3. Green Computing: The database system shall prioritize energy-efficient computing practices and use renewable energy sources where possible to reduce carbon footprint.

5.4. E-waste Management: Proper e-waste management practices shall be followed for disposing of obsolete hardware and electronic components responsibly.

5.5. Compliance with Environmental Regulations: The database system shall comply with environmental regulations and standards regarding electronic waste disposal and energy conservation.

6. Coding Standards:

6.1. Naming Conventions: Database objects such as tables, columns, and stored procedures shall follow consistent naming conventions for clarity and maintainability.

6.2. Documentation: The database system shall be well-documented, including data dictionaries, schema diagrams, and code comments, to facilitate understanding and future development.

6.3. Error Handling: Robust error handling mechanisms shall be implemented to handle exceptions gracefully and prevent data loss or corruption.

6.4. Code Review: Database code shall undergo regular peer reviews to ensure adherence to coding standards, identify potential issues, and maintain code quality.

6.5. Version Control: Changes to database schema and code shall be managed using version control systems to track revisions and facilitate collaboration among developers.