Bank Management System Database

Project Description:

In this project, we will design and implement a database management system for a hypothetical bank. The system should be able to store information about customers, accounts, transactions, loans, and other banking operations. The system should also provide a user-friendly interface (if possible and if we have time) to allow bank employees and customers to access and manage this information.

Entities:

We will maintain information about the following set of entities:

1. Customer: A customer is an individual or organization that has an account with the bank. The customer entity can store information such as name, address, contact information, and account information.
2. Account: An account is a record of financial transactions associated with a customer. The account entity can store information such as account type (e.g. savings, checking), balance, and transaction history.
3. Transaction: A transaction is a record of financial exchange between the bank and a customer. The transaction entity can store information such as the date, amount, and type of transaction (e.g. deposit, withdrawal, transfer).
4. Loan: A loan is a sum of money borrowed by a customer from the bank. The loan entity can store information such as loan amount, interest rate, and repayment schedule.
5. Branch: A branch is a physical location of the bank where customers can conduct transactions. The branch entity can store information such as an address, phone number, and hours of operation.
6. Bank entity would typically represent the main or parent bank that oversees all of its branch locations. The Bank entity might include information such as the bank's name, logo, contact information, and other details about the bank's operations and management.
7. Employee entity would typically represent the employees who work for the bank, such as tellers, customer service representatives, loan officers, and other staff.
8. Subclasses of Employee
9. Dependent of Employee

Relationships:

The following relationships will hold between our entities:

1. Customer-Account: A customer can have multiple accounts, and each account is associated with a single customer. This is a one-to-many relationship.
2. Account-Transaction: An account can have multiple transactions, and each transaction is associated with a single account. This is a one-to-many relationship.
3. Customer-Loan: A customer can have multiple loans, and each loan is associated with a single customer. This is a one-to-many relationship.
4. Loan-Transaction: A loan can have multiple transactions, and each transaction is associated with a single loan. This is a one-to-many relationship.
5. Bank-Branch: In a bank management system database project, the relationship between the Bank entity and the Branch entity is typically a one-to-many relationship, where a Bank can have multiple Branches, but each Branch is associated with only one Bank.
6. Dependent-Employee: The relationship between the Employee entity and the Dependent entity is a one-to-many relationship, where an Employee can have multiple Dependents, but each Dependent is associated with only one Employee.
7. Specialized employees (job type): Bank Teller, Technician, data analyst, branch manager
8. Specialized employees (branch type): either working in main branch or secondary branches

Constraints:

1. Primary Key Constraint: A primary key constraint is used to ensure that each record in a table has a unique identifier. For example, each customer can be identified by a unique customer ID, each account by a unique account number, and each transaction by a unique transaction ID.
2. Foreign Key Constraint: A foreign key constraint is used to enforce referential integrity between two tables. For example, the account table can have a foreign key that references the customer ID in the customer table, ensuring that each account is associated with a valid customer.
3. Not Null Constraint: A not-null constraint is used to ensure that a column in a table cannot have a null value. For example, the customer name column in the customer table cannot be null.
4. Check Constraint: A check constraint is used to enforce data integrity by restricting the values that can be entered into a column. For example, a check constraint can be used to ensure that the balance in an account cannot be negative.
5. Unique Constraint: A unique constraint is used to ensure that no duplicate values are entered into a column. For example, a unique constraint can be applied to the customer ID column in the customer table to ensure that each customer has a unique ID.
6. Default Constraint: A default constraint is used to provide a default value for a column in the event that a value is not specified when a new record is inserted into the table. For example, a default constraint can be used to set the status of a new account to "active" by default.

Sample Queries:

1. Retrieving customer information
2. Retrieving account information
3. Retrieving transaction history for a customer
4. Retrieving loan information
5. Retrieving transaction history for a loan
6. Retrieving all customers who have a loan
7. Updating the balance of an account
8. Adding a new transaction

Group Members:

1. Mohammad Al Zaatari (ID: 202201899)
2. Ahmad Bero (ID: 202216938)
3. Joe Al Hajj (ID: 202205929)
4. Adam Harb (ID: 202205355)