TEAM 13
Database Design and Initial ERD
Topic: Banking and Financial Service Management

Database Purpose:

We all know that money is a commodity that is accepted by the public as a medium of economic exchange. When it comes to managing money, it is important to have mutual trust between the service provider and the customer. Banking and Financial service enterprise management is a sophisticated system that keeps track of the customer's account and logs all the inward and outward transactions. It replaces all the paper records and makes the banking operations more effective for the banking employees and customers. Additionally, this application provides services like loans, insurances, and cards.

Business Problems Addressed:

- Allow bank employees and customers to easily set up a bank account and provide additional services like loans, insurances, and cards
- Customers have the freedom to choose any type of loan, insurance, and card
- Customers can raise the request for the service and bank employees can approve or decline that request based on customer's financial background
- Allows bank employees and customers to track recent transactions made
- Allow customers to look up the status of the services that they have opted for in real time
- Display statistical data to all the managers (Ex: Accounts created, different services availed, customer with poor credit scores, most popular branch, etc.)

Business Rules:

- Each branch offers services like loans, insurance, and cards
- A Bank Manager may add or update employee details and assign them a type (Insurance agent, Loan approver)
- A Bank employee may add, update, and validate customers' details
- A Bank employee may approve the loan, insurance, card requests made by customers based on their financial history
- Customer can have one or more accounts (Checking or Savings)
- Each account is linked to one and only one Customer
- Customer may opt for multiple services like loan, insurance, and card services
- A Customer may transfer money to other accounts, request for loan and insurance, and perform card transactions
- A service can have multiple transactions
- A service can be of different types. Ex. Different types of loan, insurance, and card

Design Decisions:

Entity Name	Why Entity Included	How Entity is Related to Other Entities
Bank Branch	This Branch entity is included because a Bank cannot exist without its branches. A Bank should have at least one branch to exist. This entity stores the list of all the Bank Branches. It stores branch details like Branch Code, Branch Name and Branch Address	The Branch entity has a one-to-many relationship with Account entity. It is connected to Account entity with BranchCode. The Branch entity has a one-to-many relationship with the Employee entity. It is connected to Employee entity with BranchCode. BranchCode field cannot be null in both Account and Employee entity.
Account	The account entity is a particularly important entity. A Branch cannot exist without any customer account. Account creation is a basic requirement for a customer to access any service within a branch. A customer can have multiple accounts and can access services like withdraw, deposit, card, loan, and insurances with these accounts. The Account entity stores details like Account ID, Branch Code, Account Type, Balance, Customer ID (Customer linked to Account), and Employee ID (Person responsible for customer Account creation)	Each account is linked to one and only one customer. The foreign key CustomerID is present in the Account Table. CustomerID field cannot be null in Account Table. As an account cannot exist without a customer. Each Account is linked to an Employee. Meaning, one employee will be responsible for account creation for that customer. The relationship is maintained by adding the foreign key CreatedBy (EmployeID) in Account entity. Account entity is also linked to Transaction, Card, Insurance and Loan entities with zero-to-many relationships. This means that each customer account may or may not access these services. The relationship is established by maintaining a foreign key AccountID in each of the above entities.
Person	Person table is table in which we store all the personal information of	Each Person data is linked to one and only one customer or employee.

	both Employee as well as Customer. Personal data like Name, DOB, SSN (Social Security Number), Phone Number, Address, City, state, zip code is stored in this table.	The table has PersonID as the primary key. PersonID acts as foreign key for Employee table and Customer Data table.
Employee	A Branch has multiple employees. To hold accountability of what each employee does within a branch Employee entity was created. Each employee performs certain duties within the branch. Every service like card, loan and insurance cannot be initiated for a customer without the approval of a designated employee. The Employee entity contains details like Employee ID, Name, SSN, Email address, Phone, Qualification, Address, Employee Type ID, Branch Code.	Employee entity is connected to Person table. Employee table stores PersonID and refers to Person table for Employee's personal details. Employee entity is connected to BankBranch entity with one and only one relationship. This means an employee can be a part of only one Branch. The Employee entity is linked to Account entity. The account entity stores the EmployeeID of the person responsible for Account Creation. The foreign key EmployeeID cannot be null in Account entity. Employee entity has one and only one relationship with Employee Type. An Employee can be assigned only one Employee Type. The foreign key EmployeeTypeID is used in Employee entity. Employee entity is linked to Card, Insurance, and Loan entities with zero-to-many relationships. An Employee may or may not approve service requests like Card, Insurance, or Loan. The relationship is maintained by using the foreign Key EmployeeID (Approved By) in the above-mentioned entities.
Employee Type	A branch includes multiple employees with each having a designated role or duty. Roles within the branch include Branch Manager, Employee Details	The Employee Type entity has zero-to-many relationships with Employee entity. Each employee within a branch will necessarily have a type assigned.

	Approver, Insurance Agent, Loan Approver, etc. To maintain a list of such employee roles this entity was created. The entity maintains details like Employee Type ID and Employee Type.	The EmployeeTypeID is a primary key in Employee Type entity which is referred to in the Employee entity.
Customer Data	The Customer Data table is used to keep track of the Customer's personal details and financial history.	The Customer Data entity is associated with the Account entity and the Customer Financial History. CustomerID is the primary key in the Customer Data entity and is the foreign key in both the Account and Financial History entities. Each customer can have multiple accounts. The Customer Data refers to the Person table using PersonID as foreign key. This helps to keep track of customer's personal data.
Customer Financial History	Customer Financial History is an essential entity which contains the vital details such as credit score for each customer which is especially important for the banks to approve the loan and insurance requests. This entity will contain the status and the last updated time as well.	The Customer Financial History entity has the FinancialHistoryID as its primary key and CustomerID as its foreign key. This entity has a one-to-one relationship with the Customer Data entity.
Card Type	There are many types of cards offered by the banks like debit cards, credit cards, ATM cards, etc. So, the card type entity will have a list of a variety of cards that are being issued by the bank.	The Card Type entity is connected to Card Entity using the primary key CardTypeID in it. There can be zero or many cards of a card type.
Card Provider	Card Provide entity tell us about the different company that are available to provide variety of card services.	The card provider table consists of CardProviderID as primary key and Name of providers. This table is connected to the Card table using its primary key CardProviderID.

Card	A bank should be able to issue cards to a customer as a service. The Card entity will have the details of the customer requesting a card, card type, card balance, interest rate, status, etc., all the information related to a card. It will also have the EmployeeID who approves the card after verifying customer's financial background.	Card entity is associated with Employee entity with many to one relationship. A card will be approved by one employee. Card entity has EmployeeID as foreign key. There can be zero-to-many relationships with the Transaction entity where a card can have several related transactions. Transaction has CardID as one of its foreign keys. Card entity is connected to Account with many to one relationship. The Card entity has AccountID as a foreign key. A customer may take multiple cards and will be linked to their account using AccountID and CardID.
Loan Type	There are many types of loan offered by the banks like education loan, home loan, car loan etc. So, the loan type entity will have a list of a variety of loans that are being offered by the bank.	The Loan Type entity is connected to Loan Entity using the primary key LoanTypeID in it. There can be zero or many loans of a loan type.
Loan	A bank should be able to loan money to a customer as a service. The loan entity will have the details of the customer taking the loan, loan type, loan amount, interest rate, date of debarment, etc., all the information related to a loan. It will also have the EmployeeID who approves the loan after verifying customer's financial background.	Loan entity is associated with Employee entity with many to one relationship. A loan will be approved by one employee. Loan entity has EmployeeID as foreign key. There can be zero-to-many relationships with the Transaction entity where a loan can have several related transactions. Transaction has LoanID as one of its foreign keys. Loan is connected to Account with many to one relationship. Loan has AccountID as foreign key. A customer may take multiple

		loans and these loans will be linked to their account using AccountID and LoanID.
Insurance Type	There are many types of insurance offered by the banks like health insurance, car insurance, business insurance, loan insurance, etc. So, the insurance type entity will have a list of a variety of insurances that are being offered by the bank.	The Insurance Type entity is connected to Insurance Entity using the primary key InsuranceTypeID in it. There can be zero or many insurances of an insurance type.
Insurance	A bank should be able to give insurance to a customer as a service. The insurance entity will have the details of the customer taking the insurance, insurance type, insured amount, status, etc., all the information related to insurance. It will also have the EmployeeID who approves the insurance after verifying customer's financial background.	Insurance entity is associated with Employee entity with many to one relationship. Insurance will be approved by one employee. Insurance entity has EmployeeID as foreign key. There can be zero-to-many relationships with the Transaction entity where an insurance can have several related transactions or claims. Transaction has InsuranceID as one of its foreign keys. Insurance is connected to Account with many to one relationship. Insurance has AccountID as foreign key. A customer may take multiple insurances and these insurances will be linked to their account using AccountID and InsuranceID.
Transaction	Transaction entity is an essential entity. It maintains a log of all the inward and outward transactions related to an account. This entity keeps track of all transactions associated with Card, Loan, Insurance, and Account Transfers. The entity can be used to generate statements.	Transaction entity is connected to Account, Card, Insurance and Loan entity with one and only one relationship. Each transaction is linked to only one Account. The relationship is maintained by using foreign key AccountID in Transaction entity. This field cannot be null for a transaction. The relationship to Card, Insurance or Loan is maintained by using foreign keys CardID or InsuranceID or LoanID respectively in Transaction entity.

Transaction Type	Transaction type table defines the type of transaction that can exist in our Banking and financial service management.	Transaction type table consists of TransactionTypeID as primary key and their description. The IDs range from 1 to 7, where: ID = 1 -> Account to account transfer ID = 2 -> Insurance Installment (customer to bank) ID = 3 -> Insurance Claim (Bank to customer) ID = 4 -> Loan Disbursement (Bank to Customer) ID = 5 -> Loan Repayment (customer to bank) ID = 6 -> Card Transaction ID = 7 -> Card Refill

Updated ER Diagram:

[Lucid Chart Link]