**Lab 2**

**Banking System - Project Report**

Course: DBI202 | Instructor: Ms. Nguyễn Thị Thu Thảo

**Group Members:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Student ID** | **Full Name** | **Group** | **Group Mark** | **Contribution (%)** | **Mark** | **Note** |
| SE203334 | Hà Nguyễn Tiến Đạt | 7 |  | 100 |  |  |
| SE190596 | Trần Hữu Việt | 7 |  | 100 |  |  |
| SE193659 | Mai Thành Được | 7 |  | 100 |  |  |
| SS190849 | Lê Nguyên Ngọc | 7 |  | 100 |  |  |

### 1. Scenario

This lab focuses on analyzing the core components of a Banking System database such as entities, attributes, functional dependencies, and keys. Through this exercise, our group aims to understand how to model real-world data into structured entities, define relationships and functional dependencies, and identify keys that ensure data integrity in a database.

**Business Rules**

* Bank have Customer.
* Banks are identified by a *name*, *code*, *address of main office*.
* Banks have branches.
* Branches are identified by a *branch\_no*, *branch\_name*, *address*.
* Customers are identified by *name*, *cust-id*, *phone number*, *address*.
* Customer can have one or more accounts.
* Accounts are identified by *account\_no*, *acc\_type*, *balance*.
* Customer can avail loans.
* Loans are identified by *loan\_id*, *loan\_type* and *amount*.
* Account and loans are related to bank's branch.

**2. Entities and Attributes:**

+ Entities:

* **Bank**: Attributes of Bank Entity are Bank Name, Code and Address.   
  *Code is Primary Key* for Bank Entity.
* **Customer**: Attributes of Customer Entity are Customer\_id, Name, Phone Number and Address.  
  *Customer\_id is Primary Key* for Customer Entity.
* **Branch**: Attributes of Branch Entity are Branch\_id, Name and Address.   
  *Branch\_id is Primary Key* for Branch Entity.
* **Account**: Attributes of Account Entity are Account\_number, Account\_Type and Balance.   
  *Account\_number is Primary Key* for Account Entity.
* **Loan**: Attributes of Loan Entity are Loan\_id, Loan\_Type and Amount.   
  *Loan\_id is Primary Key* for Loan Entity.

+ Relationships & Cardinality:

* Bank has Branch: **1–N**

*One bank has many branches but one branch can not belong to many banks, so the relationship between Bank and Branch is one to many relationship.*

* Branch manages Account: **1–N**

*One branch manages many accounts but one account can not belong to many branches, so the relationship between Branch and Account is one to many relationship.*

* Branch offer Loan: **1–N**

*One branch can have many loans but one loan can not belong to many branches, so the relationship between Branch and Loan is one to many relationship.*

* Account held by Customers: **M–N**

*One customer can have more than one accounts and also one account can be held by one or more customers, so the relationship between Account and Customers is many to many relationship.*

* Loan availed by Customer: **M–N** *(Assume loan can be jointly held by many Customers)*

*One customer can have more than one loans and also one loan can be availed by one or more customers, so the relationship between Loan and customers is many to many relationship.*

+ ER Diagram:

A diagram of a bank

AI-generated content may be incorrect.

+ Relational Diagram:

* **Bank** (Code, Name, Address)
* **Branch** (Branch\_id, Name, Address)
* **Customer** (CusID, Name, Phone, Address)
* **Account** (Account\_No, Acc\_Type, Balance)
* **Loan** (Loan\_id, Loan\_type, Amount)
* **Customer\_Account** (CusID, Account\_No) - *Junction table*
* **Customer\_Loan** (CusID, Loan\_id) - *Junction table*

1. **Functional Dependencies (FDs):**

* **Bank**  
  Code → BankName, Address  
  ***(****The bank code uniquely determines the bank name and address)*
* **Branch**  
  BranchID → Code, BranchName, Address  
  *(The branch ID uniquely determines the branch name, address, and the bank it belongs to)*
* **Customer**  
  CustomerID → Name, Phone, Address  
  *(The customer ID uniquely determines the customer's name, phone number, and address)*
* **Account**  
  AccountNumber → Type, Balance  
  *(The account number uniquely determines the account type, balance, and the branch managing it)*
* **Loan**  
  LoanID → Type, Amount  
  *(The loan ID uniquely determines the loan type, amount, and the branch that issued it)*

## ****Key Derivation:****

|  |  |  |
| --- | --- | --- |
| **Entity** | **Candidate Key(s)** | **Primary Key** |
| **Bank** | {Code} | Code |
| **Branch** | {Branch\_id} | Branch\_id |
| **Customer** | {CusID} | CusID |
| **Account** | {Account\_No} | Account\_No |
| **Loan** | {Loan\_id} | Loan\_id |
| **Customer\_account** | {CusID, Account\_No} | {CusID, Account\_No} |
| **Customer\_loan** | {CusID, Loan\_id} | {CusID, Loan\_id} |

1. **Conclusion and Reflection**

By analyzing the banking system, the team designed a relational data model, identifying core entities, attributes, functional dependencies, and primary keys. The challenge of identifying the right relationships helped the team realize that analytical skills and key inference are the foundation for building a well-structured database from real-world scenarios.