**CSE 5330-002  
Project 2  
Part 1 & 2**

**Team 23**

LATHEESH MANGERI-1002116009

HARISH TALLAPANENI-1002121160

**PART 1**

**Entity Relationship Diagram (ERD)**

**A diagram of a software company

Description automatically generated with medium confidence**

**Documentation:**

1. **Missing or Incomplete Requirements:**
   * **Dependent Information:** The requirements mention tracking dependents, but the specific information about dependents is notdetailed. Additional attributes like dependent name, age, etc., need to be defined.
   * **Manager Information:** Although the manager of each employee is mentioned, the details about the manager entity are missing. Attributes of the manager entity should be specified.
   * **Business Customer Details:** Additional details about business customers, such as business type, industry, etc., are not provided.
   * **Loan Payment Details:** The nature of the loan payment is not fully described. Additional information about the payment, such as payment method, should be included.
2. **Assumptions:**

* **Personal Banker Entity:** An assumption is made that Personal Banker is a role within the Employee entity. However, the specific attributes for this role are not explicitly mentioned in the requirements. In the relational schema, only the relationship is represented, and the attributes of the Personal Banker role are not specified. If there are specific attributes for a Personal Banker that need to be tracked, they should be added to the Employee table or a separate table should be created for Personal Banker.
* **Loan Payment Number:** In the Loan Payment entity, the attribute Payment Number is included as part of the primary key. However, it is not clear from the requirements whether this number should be unique across all loans or only unique within a specific loan. The assumption here is that Payment Number is unique within a specific loan, and the primary key includes both Payment Number and Loan Number.
* **Manager-Employee Relationship:** The requirements mention that the bank keeps track of the manager of each employee (ManagerSSN in the Employee table). However, it is not clear if an employee must have a manager, and if so, whether it is a mandatory relationship. The assumption is that the ManagerSSN in the Employee table is optional, allowing for employees who do not have a manager.
* **Dependents Relationship:**The requirements state that the bank keeps track of employees' dependents, but the nature of this relationship is not explicitly defined. The assumption is made that an employee can have multiple dependents, and the relationship is represented by the DependentsOf relationship from Employee to Dependent.

**PART 2**

**Relational Database Schema**

**A diagram of a company

Description automatically generated**

**Code to CREATE TABLES:**

-- Branch Table

CREATE TABLE Branch (

BranchID INT PRIMARY KEY,

Name VARCHAR(255),

City VARCHAR(255),

Assets DECIMAL(15,2)

);

-- Customer Table

CREATE TABLE Customer (

CustomerID INT PRIMARY KEY,

SSN VARCHAR(20),

Name VARCHAR(255),

Address VARCHAR(255)

);

-- Individual Table

CREATE TABLE Individual (

CustomerID INT PRIMARY KEY,

-- Other attributes specific to individuals

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

-- Business Table

CREATE TABLE Business (

CustomerID INT PRIMARY KEY,

-- Other attributes specific to businesses

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

-- Employee Table

CREATE TABLE Employee (

SSN VARCHAR(20) PRIMARY KEY,

Name VARCHAR(255),

Address VARCHAR(255),

Telephone VARCHAR(20),

StartDate DATE,

LengthOfEmployment INT,

ManagerSSN VARCHAR(20),

FOREIGN KEY (ManagerSSN) REFERENCES Employee(SSN)

);

-- Dependent Table

CREATE TABLE Dependent (

DependentID INT PRIMARY KEY,

Name VARCHAR(255),

Relationship VARCHAR(50),

EmployeeSSN VARCHAR(20),

FOREIGN KEY (EmployeeSSN) REFERENCES Employee(SSN)

);

-- Account Table

CREATE TABLE Account (

AccountNumber INT PRIMARY KEY,

Balance DECIMAL(15,2),

LastAccessDate DATE,

CustomerID INT,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

-- SavingsAccount Table

CREATE TABLE SavingsAccount (

AccountNumber INT PRIMARY KEY,

InterestRate DECIMAL(5,2),

FOREIGN KEY (AccountNumber) REFERENCES Account(AccountNumber)

);

-- CheckingAccount Table

CREATE TABLE CheckingAccount (

AccountNumber INT PRIMARY KEY,

Overdrafts DECIMAL(15,2),

FOREIGN KEY (AccountNumber) REFERENCES Account(AccountNumber)

);

-- Loan Table

CREATE TABLE Loan (

LoanNumber INT PRIMARY KEY,

Amount DECIMAL(15,2),

BranchID INT,

FOREIGN KEY (BranchID) REFERENCES Branch(BranchID)

);

-- LoanPayment Table

CREATE TABLE LoanPayment (

PaymentNumber INT,

Date DATE,

Amount DECIMAL(15,2),

LoanNumber INT,

PRIMARY KEY (PaymentNumber, LoanNumber),

FOREIGN KEY (LoanNumber) REFERENCES Loan(LoanNumber)

);

-- PersonalBanker Table

CREATE TABLE PersonalBanker (

EmployeeSSN VARCHAR(20),

CustomerID INT,

PRIMARY KEY (EmployeeSSN, CustomerID),

FOREIGN KEY (EmployeeSSN) REFERENCES Employee(SSN),

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);

**Choices made during the EER-to-relational mapping for the provided banking system requirements:**

1. **Entity to Table Mapping:**
   * **Choice:** Each entity in the EER diagram is mapped to a corresponding table in the relational schema.
   * **Reason:** This is a straightforward mapping to represent entities and their attributes in a relational database.
2. **Primary Keys:**
   * **Choice:** Each table has a primary key defined.
   * **Reason:** Primary keys uniquely identify each record in a table, ensuring data integrity and supporting efficient querying.
3. **Subtype Entities:**
   * **Choice:** Subtype entities (Individual and Business) are represented as separate tables.
   * **Reason:** This choice follows the Table-per-Type (TPT) inheritance mapping strategy. It allows for a clear representation of distinct attributes for individual and business customers.
4. **Foreign Keys:**
   * **Choice:** Foreign keys are used to represent relationships between tables.
   * **Reason:** Foreign keys establish relationships between tables, supporting referential integrity and ensuring that relationships between entities are maintained.
5. **UNIQUE Constraints:**
   * **Choice:** UNIQUE constraints are applied to certain attributes like SSN in the Customer table.
   * **Reason:** Ensures data integrity by preventing duplicate values in critical fields.
6. **NOT NULL Constraints:**
   * **Choice:** NOT NULL constraints are applied to attributes like Name and Address.
   * **Reason:** Ensures that essential information is always provided, preventing null values in key attributes.
7. **Composite Primary Key:**
   * **Choice:** A composite primary key is used in the PersonalBanker table.
   * **Reason:** This choice reflects a many-to-many relationship between Employee and Customer, where a PersonalBanker can be associated with multiple customers, and an employee can act as a PersonalBanker for multiple customers.
8. **Referential Integrity:**
   * **Choice:** FOREIGN KEY constraints are used to enforce referential integrity.
   * **Reason:** These constraints ensure that relationships between tables are maintained, preventing orphaned records and supporting data consistency.
9. **Data Types:**
   * **Choice:** Appropriate data types are chosen for each attribute.
   * **Reason:** Data types ensure that the stored data is accurate and consistent, optimizing storage and retrieval.
10. **Composite Primary Key in LoanPayment:**
    * **Choice:** The LoanPayment table has a composite primary key consisting of PaymentNumber and LoanNumber.
    * **Reason:** Reflects the fact that a payment number alone does not uniquely identify a payment across all loans, but it does uniquely identify a payment for a specific loan.