**Project 2**

**Part -1 : EER Model**

**A diagram of a network

Description automatically generated**

**Entities:**

1. Branch:
   1. BranchName
   2. Branch\_id (PK)
   3. City
   4. Assets
2. Customer:
   1. SSN (PK)
   2. Customer\_ID
   3. Name
   4. Address
3. Individual:
   1. SSN (PK)(FK) (inherits from Customer)
4. Business:
   1. BusinessName (PK) (inherits from Customer)
   2. SSN (FK)
5. Employee:
   1. SSN (PK),
   2. Employee\_ID
   3. Name,
   4. Address,
   5. Telephone,
   6. StartDate,
   7. LengthOfEmployment,
   8. ManagerSSN (FK)
6. Dependent:
   1. DependentName (PK),
   2. Relationship, (Assumption made)
   3. DOB,
   4. EmployeeSSN (FK)
7. Account:
   1. AccountNumber (PK),
   2. Balance,
   3. LastAccessDate
8. CheckingAccount:
   1. AccountNumber (PK) (inherits from Account),
   2. Overdraft, ...
9. SavingsAccount:
   1. AccountNumber (PK) (inherits from Account),
   2. InterestRate, ...
10. Loan:
    1. LoanNumber (PK),
    2. LoanAmount,
    3. BranchName (FK)
11. LoanPayment:
    1. PaymentNumber (PK),
    2. PaymentDate,
    3. Amount,
    4. LoanNumber (FK)

**Relationships:**

Owned\_by: (SSN (FK), AccountNumber (FK))

Holds: (SSN (FK), LoanNumber (FK))

Manages: (SSN (FK), BranchName (FK))

HasPer\_Banker: (SSN (FK), CustomerSSN (FK))

ManagesDependent: (EmployeeSSN (FK), DependentName (FK))

EmployeeReportsToEmployee: (ManagerSSN (FK), SubordinateSSN (FK))

MakesPayment: (LoanNumber (FK), PaymentNumber (FK))

**Part 2: map the EER schema design to a relational database schema**

**Relational Schema :**

**A diagram of a computer flowchart

Description automatically generated**

**Create Table Listing:**

CREATE TABLE Branch (

BranchID VARCHAR(50) PRIMARY KEY,

BranchName VARCHAR(50),

City VARCHAR(50),

Assets DECIMAL(15, 2)

);

-- Customer table

CREATE TABLE Customer (

SSN VARCHAR(45) PRIMARY KEY,

Name VARCHAR(100),

Address VARCHAR(255),

LoanNumber INT,

Branch\_id VARCHAR(50),

Customer\_ID INT ,

FOREIGN KEY (Branch\_id) REFERENCES Branch(Branch\_id),

FOREIGN KEY (LoanNumber) REFERENCES Loan(LoanNumber)

);

-- IndividualCustomer table

CREATE TABLE Individual (

SSN VARCHAR(45) PRIMARY KEY,

FOREIGN KEY (SSN) REFERENCES Customer(SSN),

-- Other attributes specific to IndividualCustomer

);

-- BusinessCustomer table

CREATE TABLE Business (

BusinessName VARCHAR(100) PRIMARY KEY,

SSN VARCHAR(45),

FOREIGN KEY (SSN) REFERENCES Customer(SSN),

);

-- Employee table

CREATE TABLE Employee (

SSN VARCHAR(45) PRIMARY KEY,

Name VARCHAR(100),

Address VARCHAR(255),

Telephone VARCHAR(15),

StartDate DATE,

LengthOfEmployment INT,

ManagerSSN VARCHAR(9),

FOREIGN KEY (ManagerSSN) REFERENCES Employee(SSN),

FOREIGN KEY (Branch\_id) REFERENCES Branch(Branch\_id)

);

-- Dependent table

CREATE TABLE Dependent (

DependentName VARCHAR(100) PRIMARY KEY,

Relationship VARCHAR(50),

DOB DATE,

EmployeeSSN VARCHAR(9),

FOREIGN KEY (EmployeeSSN) REFERENCES Employee(SSN)

);

-- Account table

CREATE TABLE Account (

AccountNumber INT PRIMARY KEY,

Balance DECIMAL(15, 2),

LastAccessDate DATE

);

-- CheckingAccount table

CREATE TABLE CheckingAccount (

AccountNumber INT PRIMARY KEY,

Overdraft DECIMAL(15, 2),

FOREIGN KEY (AccountNumber) REFERENCES Account(AccountNumber)

);

-- SavingsAccount table

CREATE TABLE SavingsAccount (

AccountNumber INT PRIMARY KEY,

InterestRate DECIMAL(5, 4),

FOREIGN KEY (AccountNumber) REFERENCES Account(AccountNumber)

);

-- Loan table

CREATE TABLE Loan (

LoanNumber INT PRIMARY KEY,

LoanAmount DECIMAL(15, 2),

BranchName VARCHAR(50),

FOREIGN KEY (Branch\_id) REFERENCES Branch(Branch\_id)

);

-- LoanPayment table

CREATE TABLE LoanPayment (

PaymentNumber INT PRIMARY KEY,

PaymentDate DATE,

Amount DECIMAL(15, 2),

LoanNumber INT

);

-- Relationships tables

-- Operates relationship

CREATE TABLE Operates (

BranchName VARCHAR(50),

AccountNumber INT,

PRIMARY KEY (BranchName, AccountNumber),

FOREIGN KEY (BranchName) REFERENCES Branch(BranchName),

FOREIGN KEY (AccountNumber) REFERENCES Account(AccountNumber)

);

-- Has relationship

CREATE TABLE Owned\_by (

SSN VARCHAR(9),

AccountNumber INT,

PRIMARY KEY (SSN, AccountNumber),

FOREIGN KEY (SSN) REFERENCES Customer(SSN),

FOREIGN KEY (AccountNumber) REFERENCES Account(AccountNumber)

);

-- Holds relationship

CREATE TABLE Holds (

SSN VARCHAR(9),

LoanNumber INT,

PRIMARY KEY (SSN, LoanNumber),

FOREIGN KEY (SSN) REFERENCES Customer(SSN),

FOREIGN KEY (LoanNumber) REFERENCES Loan(LoanNumber)

);

-- Manages relationship

CREATE TABLE Manages (

SSN VARCHAR(9),

BranchName VARCHAR(50),

PRIMARY KEY (SSN, BranchName),

FOREIGN KEY (SSN) REFERENCES Employee(SSN),

FOREIGN KEY (BranchName) REFERENCES Branch(BranchName)

);

-- HasPer\_Banker relationship

CREATE TABLE HasPer\_Banker (

SSN VARCHAR(9),

CustomerSSN VARCHAR(9),

PRIMARY KEY (SSN, CustomerSSN),

FOREIGN KEY (SSN) REFERENCES Employee(SSN),

FOREIGN KEY (CustomerSSN) REFERENCES Customer(SSN)

);

-- ManagesDependent relationship

CREATE TABLE ManagesDependent (

EmployeeSSN VARCHAR(9),

DependentName VARCHAR(100),

PRIMARY KEY (EmployeeSSN, DependentName),

FOREIGN KEY (EmployeeSSN) REFERENCES Employee(SSN),

FOREIGN KEY (DependentName) REFERENCES Dependent(DependentName)

);

-- EmployeeReportsToEmployee relationship

CREATE TABLE EmployeeReportsToEmployee (

ManagerSSN VARCHAR(9),

SubordinateSSN VARCHAR(9),

PRIMARY KEY (ManagerSSN, SubordinateSSN),

FOREIGN KEY (ManagerSSN) REFERENCES Employee(SSN),

FOREIGN KEY (SubordinateSSN) REFERENCES Employee(SSN)

);

-- MakesPayment relationship

CREATE TABLE MakesPayment (

LoanNumber INT,

PaymentNumber INT,

PRIMARY KEY (LoanNumber, PaymentNumber),

FOREIGN KEY (LoanNumber) REFERENCES Loan(LoanNumber),

FOREIGN KEY (PaymentNumber) REFERENCES LoanPayment(PaymentNumber)

);