**CSE 5330 – Database Systems Sec 002**

**Project 2**

***Submitted By:***

1) Rajat Singh Bahadur (1001878700)

2) Riddhi Baburao Joshi (1001955863)

***1.a) Extended Entity Relationship (EER) Schema Diagram:***

***Diagram

Description automatically generated***

***Design of Choices:***

For Project 2 we have created a Bank Enterprise database as per below design considerations:

* Branch is the initial entity of the Bank Enterprise database.
* Branch entity has attributes like Branch\_Name as primary key, Assets, and City.
* Branch offers Account which is another entity having Acc\_No as the primary key and other attributes like Acc\_Bal, Trans\_Date, Account\_type, Branch\_name being referenced from Branch.
* These Accounts are divided into two types which are Savings and Checkings. It has Acc\_No as the referential key. Savings has the following attributes - SV\_Depo, Sv\_With, and Int\_Rate. Checkings has Sv\_Depo, Sv\_With, and Over\_draft as its attributes.
* Employee and Branch entities are related with many to one relation respectively. Employee has attributes E\_Ssn (primary key), E\_Telno, E\_Name, E\_Stdate, E\_Length, Mgr\_Ssn, Branch\_name,E\_Type. Mgr\_Ssn is taken as multivalued attributes since several employees can have one manager.
* Dependent is a weak entity of Employee entity since Dependent table does not have a primary key, D\_Name is considered as partial key. E\_Ssn is the foreign key for Dependent entity referring from Employee.
* Employees can act as loan officer or personal banker for a particular customer.
* Customers can have accounts or can borrow loans.
* Customer has many too many relationships with Account and Loan entity.
* Customer entity provides the data related to the customer's Name, City, Street, and SSN and E\_SSN where SSN is the primary key and E\_SSN is the foreign key referenced from Employee.
* Loan originates from the branch. The loan entity has Loan\_No as its primary key, Branch\_name and Loan\_amt. Loan\_No is referenced by Loan\_Payments, a weak entity.
* Loan\_Payment keeps track of the payments made by every customer by storing the payment\_date, payment\_no, amount and loan\_pay\_no.
* For the relation – “Borrows” we have an entity called cust\_loan providing the details about the loan taken by customers.
* For the relation – “May Have” we have an entity called cust\_account providing the details about customers accounts.
* Cust\_loan and cust\_account are designed to satisfy the requirements of customer having one or more accounts or loan and vice versa.

***Assumptions:***

* Instead of using Bank as an entity, we have used Branch as an initial entity to create the database since there were no significant attributes mentioned for creating Bank.
* Considering the fact that each branch is located in a particular city, the City attribute cannot be multivalued.
* Every manager is an employee of the bank.
* A separate entity for Dependent is created considering that an employee can have more than one dependent and cannot be entered into the Employee entity.
* Savings and checkings accounts are overlapping since a customer can have both savings and checking accounts.

***Limitations:***

* Design shows that a city can have only one branch of the same bank.

***1.b) EER to Relational Schema Mapping:***

***Graphical user interface, application, table, Excel

Description automatically generated***

***Create Table Statements:***

**BRANCH:**

CREATE TABLE BRANCH (BRANCH\_NAME VARCHAR(30),

ASSETS VARCHAR(30),

CITY VARCHAR(20),

PRIMARY KEY(BRANCH\_NAME));

**EMPLOYEE:**

CREATE TABLE EMPLOYEE (

E\_SSN VARCHAR(12),

E\_TELNO INT,

E\_NAME VARCHAR(20),

E\_STDATE VARCHAR(20),

E\_LENGTH INT,

MGR\_SSN VARCHAR(12),

BRANCH\_NAME VARCHAR(30),

E\_TYPE VARCHAR(30),

PRIMARY KEY(E\_SSN),

FOREIGN KEY(BRANCH\_NAME) REFERENCES BRANCH(BRANCH\_NAME));

**DEPENDENT:**

CREATE TABLE DEPENDENT (

D\_NAME VARCHAR(20),

E\_SSN VARCHAR(12),

PRIMARY KEY(D\_NAME),

FOREIGN KEY (E\_SSN) REFERENCES EMPLOYEE(E\_SSN));

**ACCOUNT:**

CREATE TABLE ACCOUNT(

ACC\_NO INT,

ACC\_BAL FLOAT,

TRANS\_DATE VARCHAR(10),

ACCOUNT\_TYPE VARCHAR(20),

BRANCH\_NAME VARCHAR(30),

PRIMARY KEY(ACC\_NO),

FOREIGN KEY (BRANCH\_NAME) REFERENCES BRANCH(BRANCH\_NAME));

**LOAN:**

CREATE TABLE LOAN(

LOAN\_NO INT,

LOAN\_AMT FLOAT,

BRANCH\_NAME VARCHAR(20),

PRIMARY KEY (LOAN\_NO),

FOREIGN KEY (BRANCH\_NAME) REFERENCES BRANCH(BRANCH\_NAME));

**CUSTOMER:**

CREATE TABLE CUSTOMER (

C\_SSN VARCHAR(12),

C\_NAME VARCHAR(20),

C\_STREET VARCHAR(20),

C\_CITY VARCHAR(20),

E\_SSN VARCHAR(12),

PRIMARY KEY (C\_SSN),

FOREIGN KEY (E\_SSN) REFERENCES EMPLOYEE(E\_SSN));

**LOAN\_PAYMENT:**

CREATE TABLE LOAN\_PAYMENT(

PAYMENT\_NO INT,

AMOUNT FLOAT,

PAYMENT\_DATE VARCHAR(10),

LOAN\_NO INT,

LOAN\_PAY\_NO VARCHAR(6),

PRIMARY KEY(PAYMENT\_NO),

FOREIGN KEY (LOAN\_NO) REFERENCES LOAN(LOAN\_NO));

**SAVINGS:**

CREATE TABLE SAVINGS(

SV\_DEPO FLOAT,

SV\_WITH FLOAT,

INT\_RATE FLOAT,

ACC\_NO INT,

FOREIGN KEY (ACC\_NO) REFERENCES ACCOUNT(ACC\_NO));

**CHECKINGS:**

CREATE TABLE CHECKINGS(

CK\_DEPO FLOAT,

CK\_WITH FLOAT,

OVERDRAFTS FLOAT,

ACC\_NO INT,

FOREIGN KEY (ACC\_NO) REFERENCES ACCOUNT(ACC\_NO));

**CUST\_LOAN:**

CREATE TABLE CUST\_LOAN(

C\_SSN VARCHAR(12),

LOAN\_NO INT,

FOREIGN KEY (C\_SSN) REFERENCES CUSTOMER(C\_SSN),

FOREIGN KEY (LOAN\_NO) REFERENCES LOAN(LOAN\_NO));

**CUST\_ACCOUNT:**

CREATE TABLE CUST\_ACCOUNT(

C\_SSN VARCHAR(12),

ACC\_NO INT,

FOREIGN KEY (C\_SSN) REFERENCES CUSTOMER(C\_SSN),

FOREIGN KEY (ACC\_NO) REFERENCES ACCOUNT(ACC\_NO));