**DATABASE SYSTEMS** **PROJECT SUMMARY**

**PROJECT DESCRIPTION**

**PART 1**

You will first design an ER/EER schema diagram based upon the above Bank database requirements and create the schema diagram and documentation report describing your design choices. As part of this assignment, you should identify any missing or incomplete requirements, and explicitly state them in your documentation. You should also explicitly state any assumptions you made that were not part of the requirements listed above.

**PART 2**

The second part of the assignment will be to map the EER schema design to a relational database schema, and create the tables corresponding to the relational schema using the ORACLE/MySQL DBMS. You will add to your report a listing of the CREATE TABLE statements. Specify as many constraints as possible (key, referential integrity) as you can in the relational schema. You should state the choices you made during the ER/EER-to-relational mapping, and the reasons for your choices.

**TOOLS & REFERENCES**

* MySQL 8.0 Command Line
* <https://app.diagrams.net/>
* <https://www.tutorialspoint.com/dbms/er_model_to_relational_model.htm>

**PART 1: Entity-Relationship (ER) diagram**

As per the Bank Enterprise database requirements, we have designed an ER Diagram as shown below:

Diagram

Description automatically generated

**ENTITIES**

The ER diagram has following Entities:

* BANK\_BRANCH
* EMPLOYEE
* CUSTOMER
* CUST\_BANKER
* ACCOUNT
* CUS\_ACC
* LOAN
* BORROW
* PAYMENT
* Dependent

**Choices and Assumptions**

* The Bank Enterprise Database has entities BANK\_BRANCH, EMPLOYEE, CUSTOMER, CUST\_BANKER, ACCOUNT, CUS\_ACC, LOAN, BORROW, PAYMENT and Dependent derived form the given problem statement.
* The EMPLOYEE entity holds the details of all the employees and has attributes SSN, Name, Telephone\_Number, Start\_Date, Manager\_SSN.
* The CUSTOMER entity holds the details of the customers and has attributes Name, SSN, Street, City.
* The DEPENDENT entity has an attribute Name and it holds the names of the dependents listed for a given customer.
* The BANK\_BRANCH entity has all the details about the branch of the bank and has attributes Branch\_Name, Branch\_City, Assets.
* The ACCOUNT entity has all the details related to the account whether it is savings or checking account and has the attributes Account\_Number, Balance, Account\_Type, Interest\_Rate, Overdraft\_Amount.
* The PAYMENTS entity records a list of all the payments made by the customer to the bank and has the attributes Payment\_Date, Payment\_Amount, Payment\_number.
* The LOAN entity has the details of all the loans disbursed by the bank and has the entities Loan\_Number and Amount.

**Multiplicity and Relation between Entities**

* BANK\_BRANCH and ACCOUNT has a one-to-many relationship with each other because each branch can have multiple accounts, but all the accounts belong to that one branch. Similarly, BANK\_BRANCH and LOAN has a one-to-many relationship.
* LOAN and CUSTOMER have a many-to-many relationship as customers can get multiple loans.
* CUSTOMER and EMPLOYEE has a one-to-one relationship as each customer is associated to a particular banker.
* EMPLOYEE and DEPENDENT have a one-to-many relationship as an employee can have more than one dependent.

**The following are the missing prerequisites, as well as the assumptions we made:**

* Missing Attributes for Privileges, following attributes are assumed and added
* Primary keys for the tables are created as follows,

1. The Branch\_Name attribute of Bank\_Branch acts as a Primary key for that table.
2. The Loan\_number attribute of Loan acts as a Primary key for that table.
3. The SSN attribute of Customer acts as a Primary key for that table.
4. The SSN attribute of Employee acts as a Primary key for that table.
5. The Payment\_Number attribute of Payment acts as a Primary key for that table.
6. The Account\_Number attribute of Account acts as a Primary key for that table.

**PART 2: Mapping ER diagram into a Relational Schema**

Diagram

Description automatically generated

**The above diagram is made into 10 tables after converting it into relational schema:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Table Name** | **Primary Key Name Assigned** |
| 1. | BANK\_BRANCH | Branch\_Name |
| 2. | LOAN | Loan\_Number |
| 3. | PAYMENT | Payment\_Number |
| 4. | CUSTOMER | SSN |
| 5. | ACCOUNT | Account\_Number |
| 6. | EMPLOYEE | SSN |
| 7. | DEPENDENT | Name |
| 8. | CUST\_BANKER | Customer\_SSN |
| 9. | CUST\_ACCOUNT | Account\_Number |
| 10. | BORROW | Loan\_Number |

* Table CUST\_BANKER has two foreign keys Customer\_SSN and Relationship\_Officer with tables CUSTOMER and EMPLOYEE respectively.
* The ACCOUNT table has a foreign key Branch\_Name which is related to the Branch\_Name in BANK\_BRANCH table.
* The CUS\_ACC table has two foreign keys Account\_Number and Account\_Holder with tables ACCOUNT and CUSTOMER respectively.
* The LOAN table has a foreign key Loan\_Branch which is related to Branch\_Name in the BANK\_BRANCH table.
* The BORROW table has two foreign keys Loan\_Borrower and Loan\_Customer referencing tables LOAN Loan\_Number and CUSTOMER SSN respectively.
* The PAYMENT table has a foreign key Loan\_Number which is related to Loan\_Number in LOAN table.

**Create Table Queries**

CREATE TABLE BANK\_BRANCH (

Branch\_Name varchar(100) NOT NULL,

Assets decimal(10,2),

Branch\_City varchar(100) NOT NULL,

PRIMARY KEY (Branch\_Name)

);

CREATE TABLE EMPLOYEE (

SSN varchar(10) NOT NULL,

Name varchar(100) NOT NULL,

Telephone\_Number varchar(10) NOT NULL,

Start\_Date date NOT NULL,

Manager\_SSN varchar(10) NOT NULL,

PRIMARY KEY (SSN)

);

CREATE TABLE CUSTOMER (

SSN varchar(10) NOT NULL,

Name varchar(100) NOT NULL,

Street varchar(100) NOT NULL,

City varchar(100) NOT NULL,

PRIMARY KEY (SSN)

);

CREATE TABLE CUST\_BANKER (

Customer\_SSN varchar(10) NOT NULL,

Realtionship\_Officer varchar(10) NOT NULL,

Role varchar(10),

FOREIGN KEY (Customer\_SSN) REFERENCES CUSTOMER(SSN),

FOREIGN KEY (Realtionship\_Officer) REFERENCES EMPLOYEE(SSN)

);

CREATE TABLE ACCOUNT (

Account\_Number varchar(100) NOT NULL,

Balance decimal(10,2) NOT NULL,

Account\_Type varchar(10) NOT NULL,

Interest\_Rate decimal(10,2),

Overdraft\_Amount decimal(10,2),

Branch\_Name varchar(100) NOT NULL,

PRIMARY KEY (Account\_Number),

FOREIGN KEY (Branch\_Name) REFERENCES BANK\_BRANCH(Branch\_Name)

);

CREATE TABLE CUS\_ACC (

Account\_Number varchar(100) NOT NULL,

Account\_Holder varchar(10) NOT NULL,

Access\_Date date NOT NULL,

FOREIGN KEY (Account\_Number) REFERENCES ACCOUNT(Account\_Number),

FOREIGN KEY (Account\_Holder) REFERENCES CUSTOMER(SSN)

);

CREATE TABLE LOAN (

Loan\_Number int NOT NULL,

Loan\_Branch varchar(100) NOT NULL,

Amount decimal(10,2) NOT NULL,

PRIMARY KEY (Loan\_Number),

FOREIGN KEY (Loan\_Branch) REFERENCES BANK\_BRANCH(Branch\_Name)

);

CREATE TABLE BORROW (

Loan\_Number int NOT NULL,

Loan\_Customer varchar(10) NOT NULL,

FOREIGN KEY (Loan\_Borrower) REFERENCES LOAN(Loan\_Number),

FOREIGN KEY (Loan\_Customer) REFERENCES CUSTOMER(SSN)

);

CREATE TABLE PAYMENT (

Payment\_Number int NOT NULL,

Payment\_Date date NOT NULL,

Payment\_Amount decimal(10,2) NOT NULL,

Loan\_Number int NOT NULL,

PRIMARY KEY (Payment\_Number),

FOREIGN KEY (Loan\_Number) REFERENCES LOAN(Loan\_Number)

);

CREATE TABLE Dependent(

Name varchar(100) NOT NULL,

Employee\_SSN varchar(10) NOT NULL,

PRIMARY KEY (Name),

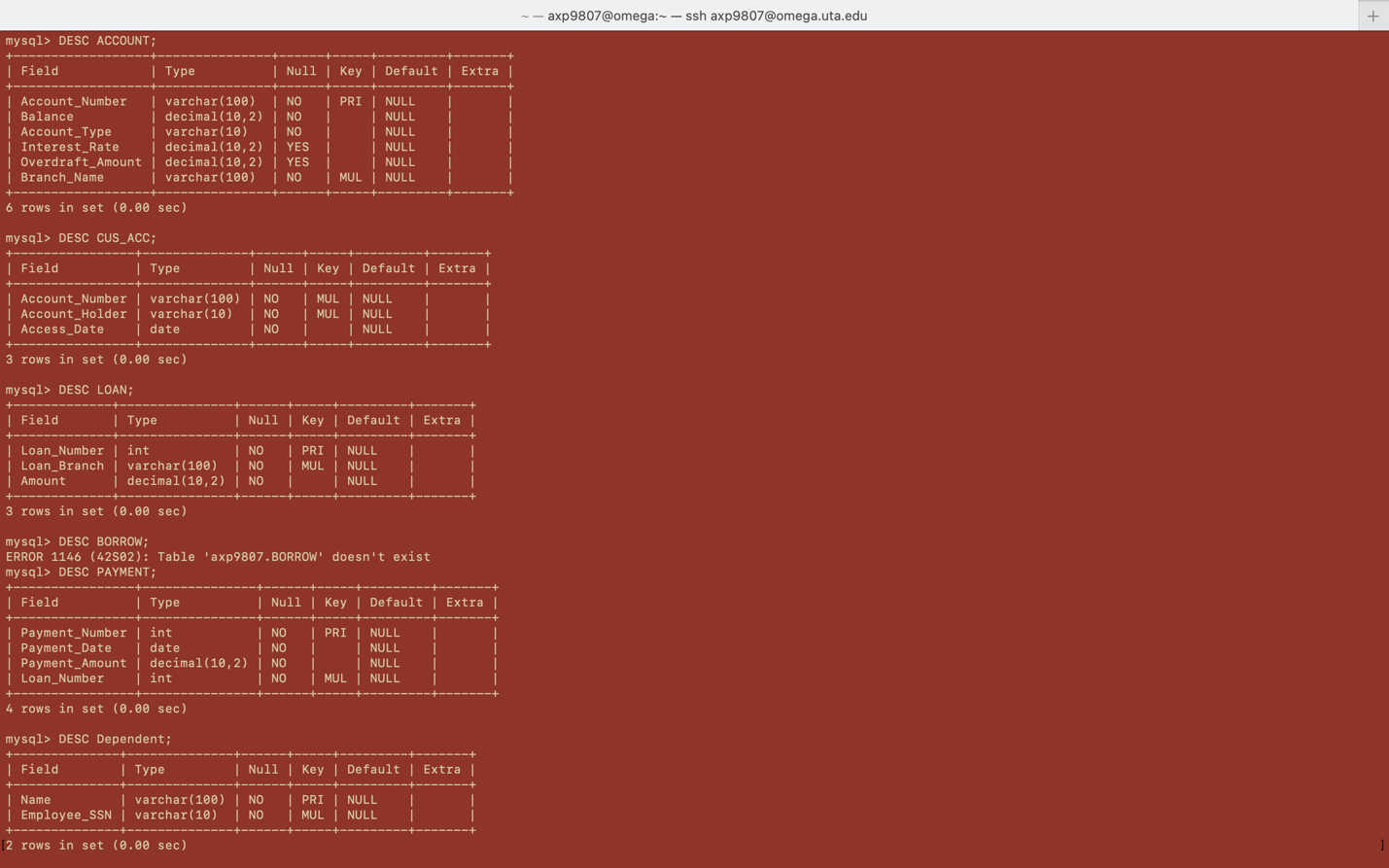
FOREIGN KEY (Employee\_SSN) REFERENCES EMPLOYEE(SSN)

);

**Table Description Screenshot**

**Text

Description automatically generated with medium confidence**

****