



**AMERICAN  
INTERNATIONAL  
UNIVERSITY-  
BANGLADESH**

# [PROJECT]

Topic: Banking Management System

Course: Introduction to Database

Section: D

## GROUP MEMBERS:

S.L.	NAME	ID	CONTRIBUTION
1	Saikat, Shahriar Ahmed	20-42816-1	Table Data Insertion, Conclusion
2	Rahman, MD. Taufiqur	22-46116-1	Normalization
3	Rakib, Sadikuzzaman	22-46971-1	ER Diagram, Finalization
4	Habib, S. M Ahsan	22-48386-3	Introduction, Case Study, Table Description, Query

SUBMITTED TO:

Supta Richard Philip

FACULTY OF COMPUTER SCIENCE

## Contents

1. Introduction .....	2
2. Case Study .....	3
3. ER Diagram .....	4
4. Normalization .....	5
5. Finalization .....	11
6. Table Description.....	15
7. Table Data Insertion.....	21
8. Query .....	27
9. Conclusion .....	29

# 1. Introduction

In today's world, we heavily depend on information and technology. Data is super important because it helps us understand things and make decisions. Imagine a database as a digital storage place where we keep all this valuable data. For a bank, company or any kind of organization, having a good database is like having a treasure of essential information. It holds important stuff like employee records, transaction records, and salary details. This makes things run smoothly and allows easy access to the information we need.

We designed a good and efficient database named as "Banking Management System" where we stored a lot of information about a bank. We stored there bank information, customer information, transaction information, salary details etc. Let's see the scenario and entity relation diagram of our projects.

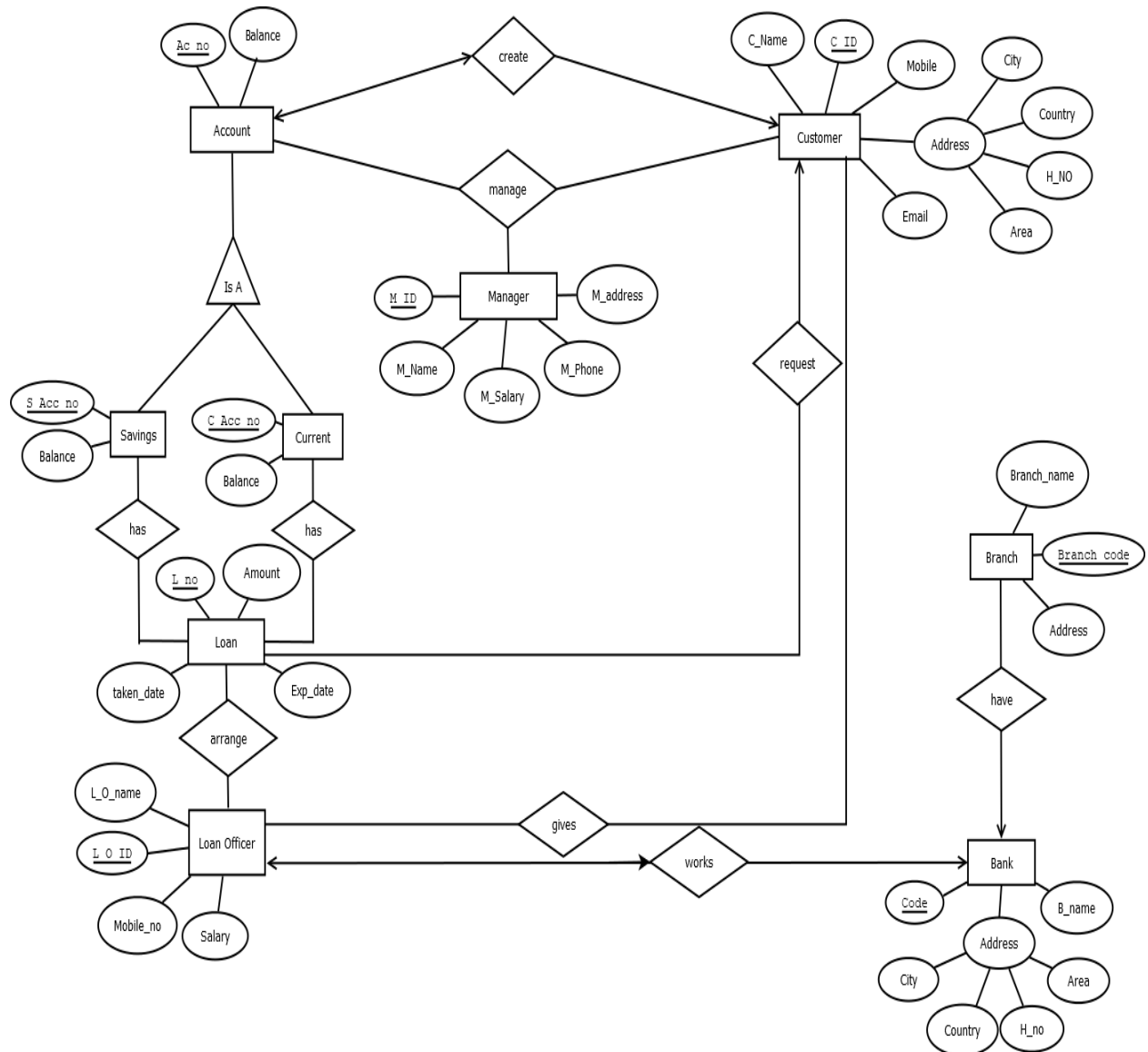
## 2. Case Study

In a Banking Management system, a bank has its own name, address (city, country) and code. A customer can create account in bank. A customer can create one to many account but an account can be created by a customer. A customer has an unique customer ID, name, mobile number, email, address (city, country). An account contains account number and balance. The bank has a manager who manages the process between accounts and customers.

The system stores manager ID, name, salary, phone number and address. Account can be divided into savings account and current account. Savings account has savings account number and balance and current account has current account number and balance. Loans can be taken from both savings account and current account. Loan has loan number, loan amount, taken date, expire date. Loans are arranged by loan Officer. The system also stores loan officers name, id, salary, mobile number. One customer can request for many loans. A loan officer can gives loan to many customer.

A bank has many loan officers but a loan officer can works in one bank. A bank may have many branches. A branch has its own name, code and address. Now construct an ER Diagram based on given information.

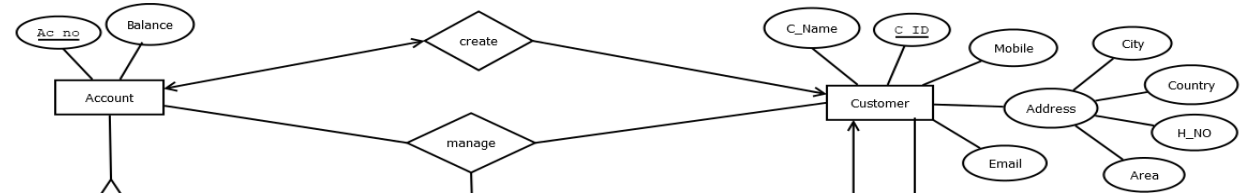
# 3. ER Diagram



Banking Management System

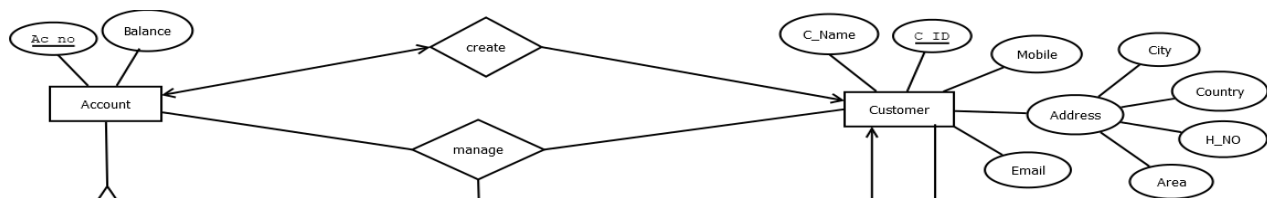
# 4. Normalization

Create:



<b>UNF</b>	C_Name, C ID, Mobile, Address, City, Country, H_NO, Area, Email, Ac no, Balance
<b>1NF</b>	C_Name, C ID, Mobile, City, Country, H_NO, Area, Email, Ac no, Balance
<b>2NF</b>	<ol style="list-style-type: none"> <li>1. C_Name, C ID, Mobile, City, Country, H_NO, Area, Email</li> <li>2. Ac no, Balance</li> <li>3. C ID (PK), Ac no (FK)</li> </ol>
<b>3NF</b>	<ol style="list-style-type: none"> <li>1. C_Name, C ID, Mobile, H_NO, Email</li> <li>2. Ac no, Balance</li> <li>3. C ID (PK), Ac no (FK)</li> <li>4. City, Country, H_NO, Area,</li> </ol>

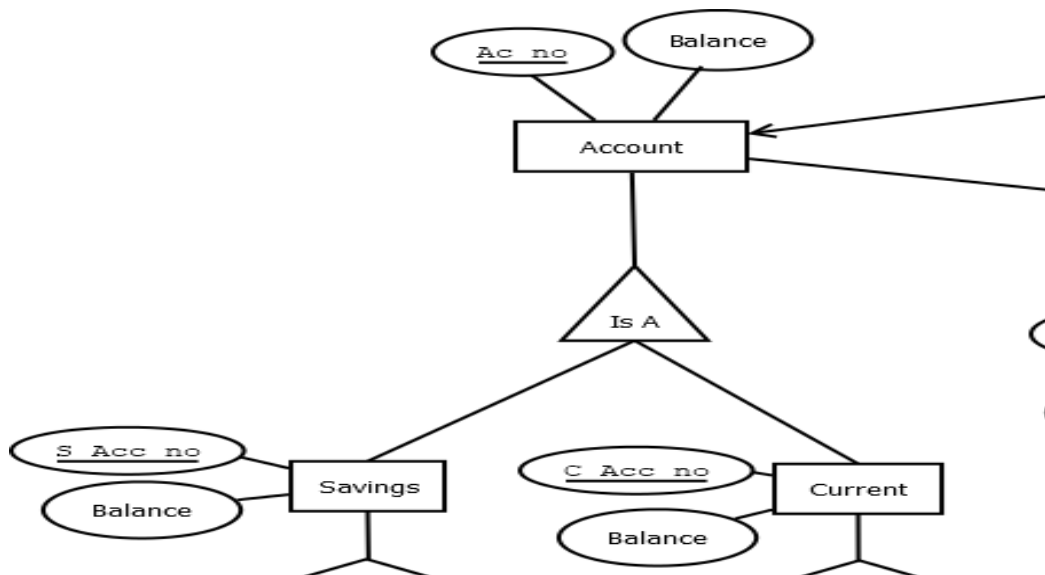
Manage:



<b>UNF</b>	C_Name, C ID, Mobile, Address, City, Country, H_NO, Area, Email, Ac no, Balance, M ID, M_Name, M_Salary, M_Phone, M_Address
<b>1NF</b>	C_Name, C ID, Mobile, City, Country, H_NO, Area, Email, Ac no, Balance, M ID, M_Name, M_Salary, M_Phone, M_Address
<b>2NF</b>	<ol style="list-style-type: none"> <li>1. C_Name, C ID, Mobile, City, Country, H_NO, Area, Email</li> <li>2. Ac no, Balance</li> <li>3. M ID, M_Name, M_Salary, M_Phone, M_Address</li> <li>4. C ID (PK), Ac no (FK), M ID(FK)</li> </ol>

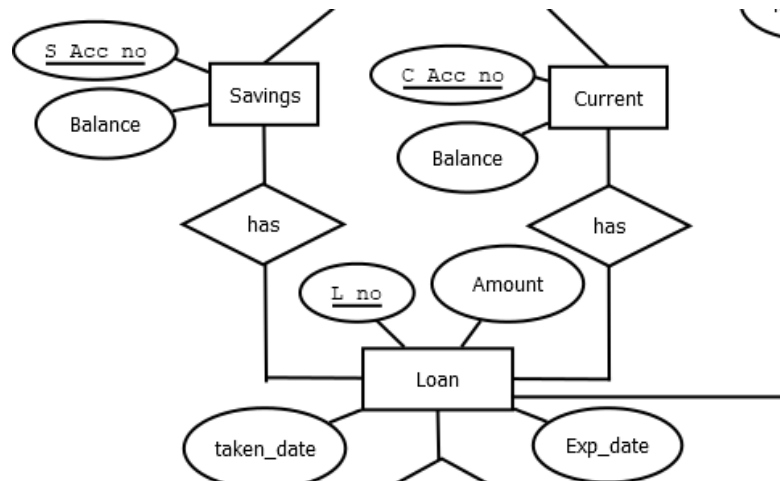
<b>3NF</b>	<ol style="list-style-type: none"> <li>1. C_Name, C ID, Mobile, H_NO, Area, Email</li> <li>2. Ac no, Balance</li> <li>3. M ID, M_Name, M_Salary, M_Phone, M_Address</li> <li>4. C ID (PK), Ac no (FK), M ID(FK)</li> <li>5. C ID (PK), Ac no (FK), M ID(FK)</li> </ol>
------------	--

Saving account and current account is a account:



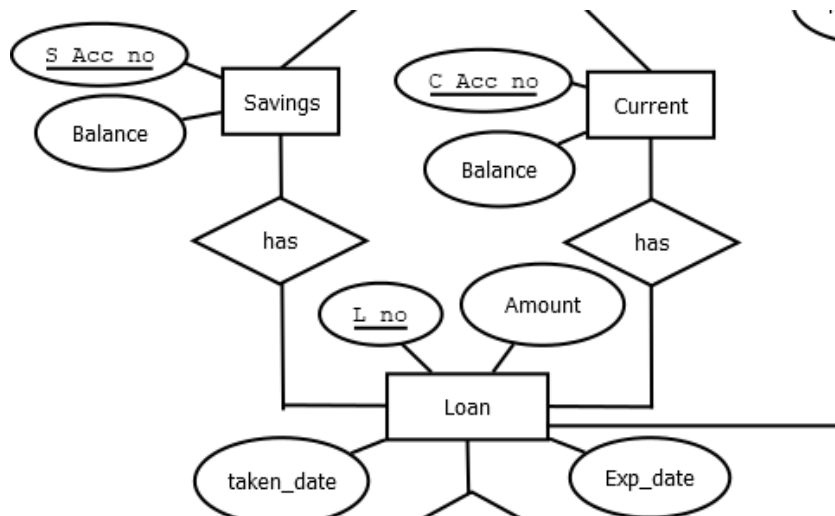
<b>UNF</b>	Ac no, Balance, S Ac no, Balance, C Ac no, Balance
<b>1NF</b>	Ac no, Balance, S Ac no, Balance, C Ac no, Balance
<b>2NF</b>	<ol style="list-style-type: none"> <li>1. Ac no, Balance,</li> <li>2. S Ac no, Balance</li> <li>3. C Ac no, Balance</li> <li>4. Ac no(PK), S Ac no(FK), C Ac no(FK)</li> </ol>
<b>3NF</b>	<ol style="list-style-type: none"> <li>1. Ac no, Balance,</li> <li>2. S Ac no, Balance</li> <li>3. C Ac no, Balance</li> <li>4. Ac no(PK), S Ac no(FK), C Ac no(FK)</li> </ol>

## SAVINGS HAS LOAN:



<b>UNF</b>	S_Acc_no, Balance, L_no, Amount, taken_date, Exp_date
<b>1NF</b>	S_Acc_no, Balance, L_no, Amount, taken_date, Exp_date
<b>2NF</b>	1. S_Acc_no, Balance 2. L_no, Amount, taken_date, Exp_date 3. S_Ac_no(PK), L_no(FK)
<b>3NF</b>	Same as 2NF

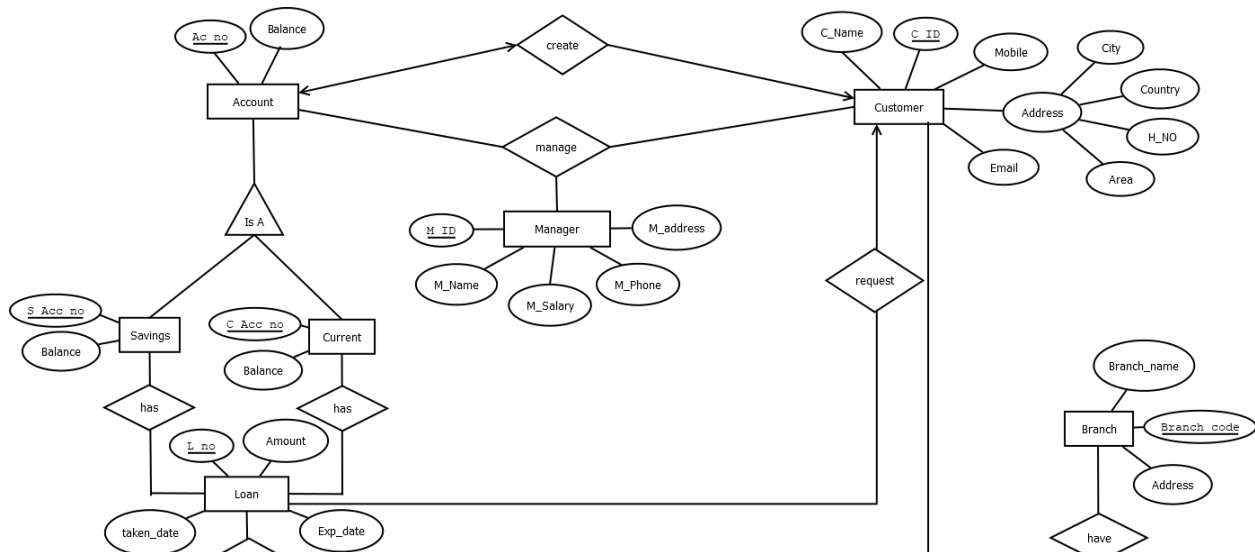
## CURRENT HAS LOAN:





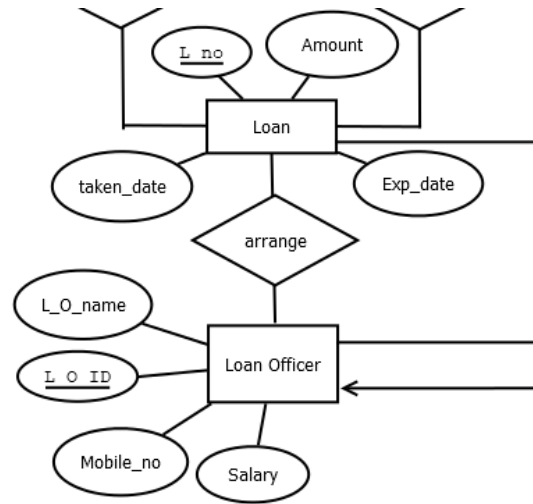
<b>UNF</b>	C Ac no, Balance, L no, Amount, taken_date, Exp_date
<b>1NF</b>	C Ac no, Balance, L no, Amount, taken_date, Exp_date
<b>2NF</b>	<ol style="list-style-type: none"> <li>1. C Ac no, Balance</li> <li>2. L no, Amount, taken_date, Exp_date</li> <li>3. S Ac no(PK), L no(FK)</li> </ol>
<b>3NF</b>	Same as 2NF

### CUSTOMER REQUEST LOAN:



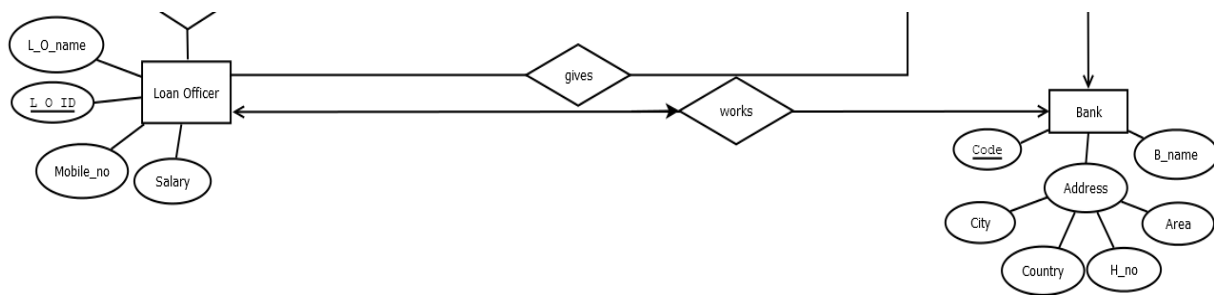
<b>UNF</b>	C_Name, C ID, Mobile, Address, City, Country, H_NO, Area, Email, L no, Amount, taken_date, Exp_date
<b>1NF</b>	C_Name, C ID, Mobile, City, Country, H_NO, Area, Email, L no, Amount, taken_date, Exp_date
<b>2NF</b>	<ol style="list-style-type: none"> <li>1. C_Name, C ID, Mobile, City, Country, H_NO, Area, Email</li> <li>2. L no, Amount, taken_date, Exp_date</li> <li>3. C ID (PK), L no(FK)</li> </ol>
<b>3NF</b>	<ol style="list-style-type: none"> <li>1. C_Name, C ID, Mobile, H_NO, Email</li> <li>2. L no, Amount, taken_date, Exp_date</li> <li>3. C ID (PK), L no(FK)</li> <li>4. City, Country, H_NO, Area,</li> </ol>

## LOAN OFFICER ARRANGE LOAN:



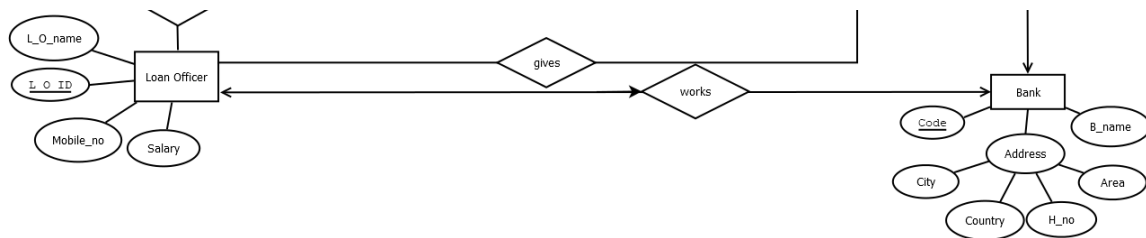
<b>UNF</b>	L_O_name, L O ID, Mobile_no, Salary, L no, Amount, taken_date, Exp_date
<b>1NF</b>	L_O_name, L O ID, Mobile_no, Salary, L no, Amount, taken_date, Exp_date
<b>2NF</b>	1. L_O_name, L O ID, Mobile_no, Salary 2. L no ,Amount, taken_date, Exp_date 3. L O ID(PK), L no(FK)
<b>3NF</b>	Same as 2NF

## LOAN OFFICER GIVES CUSTOMER LOAN:



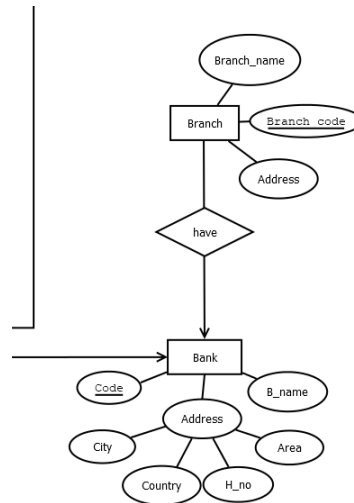
<b>UNF</b>	C_Name, C ID, Mobile, Address ,City, Country, H_NO, Area, Email, L_O_name, L O ID, Mobile_no, Salary
<b>1NF</b>	C_Name, C ID, Mobile, City, Country, H_NO, Area, Email, L_O_name, L O ID, Mobile_no, Salary
<b>2NF</b>	<ol style="list-style-type: none"> <li>1. C_Name, C ID, Mobile, City, Country, H_NO, Area, Email</li> <li>2. L_O_name, L O ID, Mobile_no, Salary</li> <li>3. C ID (PK), L O ID(FK)</li> </ol>
<b>3NF</b>	<ol style="list-style-type: none"> <li>1. C_Name, C ID, Mobile, H_NO, Email</li> <li>2. L_O_name, L O ID, Mobile_no, Salary</li> <li>3. C ID (PK), L O ID(FK)</li> <li>4. City, Country, H_NO, Area</li> </ol>

#### LOAN OFFICER WORKS ON BANK:



<b>UNF</b>	L_O_name, L O ID, Mobile_no, Salary, Code, Address, City, Country, H_no, Area, B_name
<b>1NF</b>	L_O_name, L O ID, Mobile_no, Salary, Code, City, Country, H_no, Area, B_name
<b>2NF</b>	<ol style="list-style-type: none"> <li>1. L_O_name, L O ID, Mobile_no, Salary</li> <li>2. Code, City, Country, H_no, Area, B_name</li> <li>3. L O ID(PK), Code(FK)</li> </ol>
<b>3NF</b>	<ol style="list-style-type: none"> <li>1. L_O_name, L O ID, Mobile_no, Salary</li> <li>2. Code, H_no, B_name</li> <li>3. L O ID(PK), Code(FK)</li> <li>4. City, Country, H_no, Area</li> </ol>

BANK HAVE BRANCH:



<b>UNF</b>	Code, Address, City, Country, H_no, Area, B_name, Branch_name, Branch code, Address
<b>1NF</b>	Code, City, Country, H_no, Area, B_name, Branch_name, Branch code, Address
<b>2NF</b>	1. Code, City, Country, H_no, Area, B_name 2. Branch_name, Branch code, Address 3. Code(PK), Branch code(FK)
<b>3NF</b>	1. Code, H_no, B_name 2. Branch_name, Branch code, Address 3. Code(PK), Branch code(FK) 4. City, Country, H_no, Area

## 5. Finalization

### 1. CustomerInfo:

- C\_ID (Primary Key)
- C\_Name
- Mobile\_No
- C\_Email
- Address

## **2. AccountInfo:**

- Acc\_no (Primary Key)
- Balance

## **3. CustomerAddress:**

- H\_No (Primary Key)
- City
- Area
- Country

## **4. CustomerAccount:**

- C\_ID (Primary Key, Foreign Key referencing CustomerInfo)
- Acc\_no (Foreign Key referencing AccountInfo)

## **5. ManagerInfo:**

- M\_ID (Primary Key)
- M\_Name
- M\_Salary
- M\_Phone
- M\_Address

## **6. AccountManagement:**

- C\_ID (Primary Key, Foreign Key referencing CustomerInfo)
- Acc\_no (Foreign Key referencing AccountInfo)
- M\_ID (Foreign Key referencing ManagerInfo)

## **7. SavingsAccount:**

- S\_Acc\_no (Primary Key)
- Balance

### **8. LoanInfo:**

- L\_no (Primary Key)
- L\_amount
- taken\_date
- Exp\_date

### **9. SavingsAccountLoanRegistration:**

- L\_no (Primary Key, Foreign Key referencing LoanInfo)
- S\_Acc\_no (Foreign Key referencing SavingsAccount)

### **10. CurrentAccount:**

- C\_Acc\_no (Primary Key)
- Balance

### **11. CurrentAccountLoanRegistration:**

- L\_no (Primary Key, Foreign Key referencing LoanInfo)
- C\_Acc\_no (Foreign Key referencing CurrentAccount)

### **12. CustomerLoanRegistration:**

- C\_ID (Primary Key, Foreign Key referencing CustomerInfo)
- L\_no (Foreign Key referencing LoanInfo)

### **13. LoanOfficer:**

- L\_O\_ID (Primary Key)
- L\_O\_name
- Mobile\_number
- Salary

### **14. LoanRegistertoLoanOfficer:**

- L\_ID (Primary Key)
- L\_O\_ID (Foreign Key referencing LoanOfficer)

**15. BankInfo:**

Code (Primary Key)

- B\_name

- Address

**16. BankAddress:**

- H\_no (Primary Key)

- Area

- City

- Country

**17. LoanOfficerofBank:**

- L\_O\_ID (Primary Key, Foreign Key referencing LoanOfficer)

- Code (Foreign Key referencing BankInfo)

**18. BranchInfo:**

- Branch\_code (Primary Key)

- Branch\_name

- Address

**19. BranchRegistration:**

- Code (Primary Key, Foreign Key referencing BankInfo)

- Branch\_code (Foreign Key referencing BranchInfo)

# 6. Table Description

```
1) CREATE TABLE CustomerInfo(
    C_ID VARCHAR2(10) PRIMARY KEY,
    C_Name VARCHAR2(50),
    Mobile_No NUMBER(15),
    C_Email VARCHAR2(30),
    Address VARCHAR2(50));

DESCRIBE CustomerInfo;
```

Object Type **TABLE** Object **CUSTOMERINFO**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERINFO	C_ID	Varchar2	10	-	-	1	-	-	-
	C_NAME	Varchar2	50	-	-	-	✓	-	-
	MOBILE_NO	Number	-	15	0	-	✓	-	-
	C_EMAIL	Varchar2	30	-	-	-	✓	-	-
	ADDRESS	Varchar2	50	-	-	-	✓	-	-
									1 - 5

```
2) CREATE TABLE AccountInfo(
    Acc_no NUMBER(20) PRIMARY KEY,
    Balance NUMBER(20));

DESCRIBE AccountInfo;
```

Object Type **TABLE** Object **ACCOUNTINFO**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ACCOUNTINFO	ACC_NO	Number	-	20	0	1	-	-	-
	BALANCE	Number	-	20	0	-	✓	-	-
									1 - 2

```
3) CREATE TABLE CustomerAddress(
    H_No VARCHAR2(10) PRIMARY KEY,
    City VARCHAR2(50),
    Area VARCHAR2(20),
    Country VARCHAR2(20));

DESCRIBE CustomerAddress;
```



Object Type **TABLE** Object **CUSTOMERADDRESS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERADDRESS	H_NO	Varchar2	10	-	-	1	-	-	-
	CITY	Varchar2	50	-	-	-	✓	-	-
	AREA	Varchar2	20	-	-	-	✓	-	-
	COUNTRY	Varchar2	20	-	-	-	✓	-	-
									1 - 4

- 4) CREATE TABLE CustomerAccount(  
     C\_ID VARCHAR2(10) PRIMARY KEY,  
     Acc\_no NUMBER(20),  
     CONSTRAINT AN FOREIGN KEY(Acc\_no) REFERENCES  
     AccountInfo(Acc\_no));  
 DESCRIBE CustomerAccount;

Object Type **TABLE** Object **CUSTOMERACCOUNT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERACCOUNT	C_ID	Varchar2	10	-	-	1	-	-	-
	ACC_NO	Number	-	20	0	-	✓	-	-
									1 - 2

- 5) CREATE TABLE ManagerInfo(  
     M\_ID VARCHAR2(10) PRIMARY KEY,  
     M\_Name VARCHAR2(20),  
     M\_Salary NUMBER(10),  
     M\_Phone NUMBER(15),  
     M\_Address VARCHAR2(30));  
 DESCRIBE ManagerInfo;

Object Type **TABLE** Object **MANAGERINFO**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MANAGERINFO	M_ID	Varchar2	10	-	-	1	-	-	-
	M_NAME	Varchar2	20	-	-	-	✓	-	-
	M_SALARY	Number	-	10	0	-	✓	-	-
	M_PHONE	Number	-	15	0	-	✓	-	-
	M_ADDRESS	Varchar2	30	-	-	-	✓	-	-
									1 - 5

- 6) CREATE TABLE AccountManagement(  
     C\_ID VARCHAR2(10) PRIMARY KEY,  
     Acc\_no NUMBER(20),  
     M\_ID VARCHAR2(10),  
     CONSTRAINT FK\_AccountInfo FOREIGN KEY(Acc\_no) REFERENCES AccountInfo(Acc\_no),  
     CONSTRAINT FK\_ManagerInfo FOREIGN KEY(M\_ID) REFERENCES ManagerInfo(M\_ID));  
 DESCRIBE AccountManagement;

Object Type **TABLE** Object **ACCOUNTMANAGEMENT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ACCOUNTMANAGEMENT	C_ID	Varchar2	10	-	-	1	-	-	-
	ACC_NO	Number	-	20	0	-	✓	-	-
	M_ID	Varchar2	10	-	-	-	✓	-	-
									1 - 3

```
7) CREATE TABLE SavingsAccount(
      S_Acc_no NUMBER(20) PRIMARY KEY,
      Balance NUMBER(10));
DESCRIBE SavingsAccount;
```

Object Type **TABLE** Object **SAVINGSACCOUNT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SAVINGSACCOUNT	S_ACC_NO	Number	-	20	0	1	-	-	-
	BALANCE	Number	-	10	0	-	✓	-	-
									1 - 2

```
8) CREATE TABLE LoanInfo(
      L_no VARCHAR2(10) PRIMARY KEY,
      L_amount NUMBER(10),
      taken_date DATE,
      Exp_date DATE);
DESCRIBE LoanInfo;
```

Object Type **TABLE** Object **LOANINFO**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LOANINFO	L_NO	Varchar2	10	-	-	1	-	-	-
	L_AMOUNT	Number	-	10	0	-	✓	-	-
	TAKEN_DATE	Date	7	-	-	-	✓	-	-
	EXP_DATE	Date	7	-	-	-	✓	-	-
									1 - 4

```
9) CREATE TABLE Savings_AccLoanRegistration(
      L_no VARCHAR2(10) PRIMARY KEY,
      S_Acc_no NUMBER(20));
DESCRIBE Savings_AccLoanRegistration;
```

Object Type **TABLE** Object **SAVINGS\_ACCLOANREGISTRATION**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SAVINGS_ACCLOANREGISTRATION	L_NO	Varchar2	10	-	-	1	-	-	-
	S_ACC_NO	Number	-	20	0	-	✓	-	-
1 - 2									

```
10) CREATE TABLE CurrentAccount(
      C_Acc_no NUMBER(20) PRIMARY KEY,
      Balance NUMBER(20));
DESCRIBE CurrentAccount;
```

Object Type **TABLE** Object **CURRENTACCOUNT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CURRENTACCOUNT	C_ACC_NO	Number	-	20	0	1	-	-	-
	BALANCE	Number	-	20	0	-	✓	-	-
1 - 2									

```
11) CREATE TABLE Current_AccLoanRegistration(
      L_no VARCHAR2(10) PRIMARY KEY,
      C_Acc_no NUMBER(20));
DESCRIBE Current_AccLoanRegistration;
```

Object Type **TABLE** Object **CURRENT\_ACCLOANREGISTRATION**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CURRENT_ACCLOANREGISTRATION	L_NO	Varchar2	10	-	-	1	-	-	-
	C_ACC_NO	Number	-	20	0	-	✓	-	-
1 - 2									

```
12) CREATE TABLE CustomerLoanRegistration(
      C_ID VARCHAR2(10) PRIMARY KEY,
      L_no VARCHAR2(10),
      CONSTRAINT LNbyCustomer FOREIGN KEY(L_no) REFERENCES LoanInfo(L_no));
DESCRIBE CustomerLoanRegistration;
```

Object Type **TABLE** Object **CUSTOMERLOANREGISTRATION**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERLOANREGISTRATION	C_ID	Varchar2	10	-	-	1	-	-	-
	L_NO	Varchar2	10	-	-	-	✓	-	-
1 - 2									

```
13) CREATE TABLE LoanOfficer(
      L_O_ID VARCHAR2(10) PRIMARY KEY,
      L_O_name VARCHAR2(20),
      Mobile_number NUMBER(15),
      Salary NUMBER(10));
DESCRIBE LoanOfficer;
```

Object Type **TABLE** Object **LOANOFFICER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LOANOFFICER	L_O_ID	Varchar2	10	-	-	1	-	-	-
	L_O_NAME	Varchar2	20	-	-	-	✓	-	-
	MOBILE_NUMBER	Number	-	15	0	-	✓	-	-
	SALARY	Number	-	10	0	-	✓	-	-
									1 - 4

```
14) CREATE TABLE LoanRegistertoLoanOfficer(
      L_ID VARCHAR2(20) PRIMARY KEY,
      L_O_ID VARCHAR2(10),
      CONSTRAINT LoanCount FOREIGN KEY(L_O_ID) REFERENCES LoanOfficer(L_O_ID));
DESCRIBE LoanRegistertoLoanOfficer;
```

Object Type **TABLE** Object **LOANREGISTERTOLOANOFFICER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LOANREGISTERTOLOANOFFICER	L_ID	Varchar2	20	-	-	1	-	-	-
	L_O_ID	Varchar2	10	-	-	-	✓	-	-
									1 - 2

```
15) CREATE TABLE BankInfo(
      Code VARCHAR2(10) PRIMARY KEY,
      B_name VARCHAR2(20),
      Address VARCHAR2(20));
DESCRIBE BankInfo;
```

Object Type **TABLE** Object **BANKINFO**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BANKINFO	CODE	Varchar2	10	-	-	1	-	-	-
	B_NAME	Varchar2	20	-	-	-	✓	-	-
	ADDRESS	Varchar2	20	-	-	-	✓	-	-
									1 - 3

```
16) CREATE TABLE BankAddress_Central(
      H_no VARCHAR2(10) PRIMARY KEY,
      Area VARCHAR2(10),
      City VARCHAR2(20),
      Country VARCHAR2(20));
DESCRIBE BankAddress_Central;
```

Object Type **TABLE** Object **BANKADDRESS\_CENTRAL**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BANKADDRESS_CENTRAL	H_NO	Varchar2	10	-	-	1	-	-	-
	AREA	Varchar2	10	-	-	-	✓	-	-
	CITY	Varchar2	20	-	-	-	✓	-	-
	COUNTRY	Varchar2	20	-	-	-	✓	-	-
									1 - 4

```

17) CREATE TABLE LoanOfficerofBank(
      L_O_ID VARCHAR2(10) PRIMARY KEY,
      Code VARCHAR2(10),
      CONSTRAINT Emp_ID FOREIGN KEY(Code) REFERENCES BankInfo(Code));
DESCRIBE LoanOfficerofBank;

```

Object Type **TABLE** Object **LOANOFFICEROFBANK**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LOANOFFICEROFBANK	L_O_ID	Varchar2	10	-	-	1	-	-	-
	CODE	Varchar2	10	-	-	-	✓	-	-
1 - 2									

```

18) CREATE TABLE BranchInfo(
      Branch_code VARCHAR2(10) PRIMARY KEY,
      Branch_name VARCHAR2(20),
      Address VARCHAR2(20));
DESCRIBE BranchInfo;

```

Object Type **TABLE** Object **BRANCHINFO**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BRANCHINFO	BRANCH_CODE	Varchar2	10	-	-	1	-	-	-
	BRANCH_NAME	Varchar2	20	-	-	-	✓	-	-
	ADDRESS	Varchar2	20	-	-	-	✓	-	-
1 - 3									

```

19) CREATE TABLE BranchRegistration(
      Branch_Code VARCHAR2(10) PRIMARY KEY,
      code VARCHAR2(10),
      CONSTRAINT Branch_ID FOREIGN KEY(Branch_code) REFERENCES
      BranchInfo(Branch_code));

```

DESCRIBE BranchRegistration;

Object Type **TABLE** Object **BRANCHREGISTRATION**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BRANCHREGISTRATION	CODE	Varchar2	10	-	-	1	-	-	-
	BRANCH_CODE	Varchar2	10	-	-	-	✓	-	-
1 - 2									

# 7. Table Data Insertion

**Table 1: Customer Info**

INSERT INTO CustomerInfo VALUES ('cus0001', 'Habib', 01854056655, 'habib@gmail.com', 'Badda, Dhaka');

INSERT INTO CustomerInfo VALUES ('cus0002', 'Rakib', 01754053455, 'rakib@gmail.com', 'Uttara, Dhaka');

INSERT INTO CustomerInfo VALUES ('cus0003', 'Rahman', 01854564658, 'rahman@gmail.com', 'Mirpur, Dhaka');

INSERT INTO CustomerInfo VALUES ('cus0004', 'Saikat', 01995586787, 'saikat@gmail.com', 'Tongi, Gazipur');

INSERT INTO CustomerInfo VALUES ('cus0005', 'Toufiq', 01998456355, 'toufiq@gmail.com', 'Rayerbag, Dhaka');

SELECT \* from CustomerInfo;

C_ID	C_NAME	MOBILE_NO	C_EMAIL	ADDRESS
cus0001	Habib	1854056655	habib@gmail.com	Badda, Dhaka
cus0002	Rakib	1754053455	rakib@gmail.com	Uttara, Dhaka
cus0003	Rahman	1854564658	rahman@gmail.com	Mirpur, Dhaka
cus0004	Saikat	1995586787	saikat@gmail.com	Tongi, Gazipur
cus0005	Toufiq	1998456355	toufiq@gmail.com	Rayerbag, Dhaka

**Table 2: Account Info**

INSERT INTO AccountInfo VALUES (20428161, 50000);

INSERT INTO AccountInfo VALUES (22461161, 70000);

INSERT INTO AccountInfo VALUES (22483863, 30000);

INSERT INTO AccountInfo VALUES (22469711, 40000);

INSERT INTO AccountInfo VALUES (20428181, 10000);

SELECT \* from AccountInfo;

ACC_NO	BALANCE
20428161	50000
22461161	70000
22483863	30000
22469711	40000
20428181	10000

**Table 3: Customer Address**

INSERT INTO CustomerAddress VALUES ('550/2','Dhaka','Badda','Bangladesh');

INSERT INTO CustomerAddress VALUES (105,'Dhaka','Uttara','Bangladesh');

INSERT INTO CustomerAddress VALUES ('337','Dhaka','Mirpur-2','Bangladesh');

INSERT INTO CustomerAddress VALUES ('59/2','Gazipur','Tongi','Bangladesh');

INSERT INTO CustomerAddress VALUES ('73/1','Dhaka','Rayerbag','Bangladesh');

SELECT \* from CustomerAddress;

H_NO	CITY	AREA	COUNTRY
550/2	Dhaka	Badda	Bangladesh
105	Dhaka	Uttara	Bangladesh
337	Dhaka	Mirpur-2	Bangladesh
59/2	Gazipur	Tongi	Bangladesh
73/1	Dhaka	Rayerbag	Bangladesh

**Table 4: Customer Account**

INSERT INTO CustomerAccount VALUES ('cus0001', 22483863);

INSERT INTO CustomerAccount VALUES ('cus0002', 22469711);

INSERT INTO CustomerAccount VALUES ('cus0003', 22461161);

INSERT INTO CustomerAccount VALUES ('cus0004', 20428161);

INSERT INTO CustomerAccount VALUES ('cus0005', 20428181);

SELECT \* from CustomerAccount;

C_ID	ACC_NO
cus0001	22483863
cus0002	22469711
cus0003	22461161
cus0004	20428161
cus0005	20428181

**Table 5: Manager Info**

INSERT INTO ManagerInfo VALUES ('mgr0001', 'Fahim', 50000, 01766995999, 'Airport, Dhaka');

SELECT \* from ManagerInfo;

M_ID	M_NAME	M_SALARY	M_PHONE	M_ADDRESS
mgr0001	Fahim	50000	1766995999	Airport, Dhaka

**Table 6: Account Management**

INSERT INTO AccountManagement VALUES ('cus0001',22483863,'mgr0001');

INSERT INTO AccountManagement VALUES ('cus0002',22469711,'mgr0001');

INSERT INTO AccountManagement VALUES ('cus0003',22461161,'mgr0001');

```

INSERT INTO AccountManagement VALUES ('cus0004',20428161,'mgr0001');
INSERT INTO AccountManagement VALUES ('cus0005',20428181,'mgr0001');
SELECT * from AccountManagement;

```

C_ID	ACC_NO	M_ID
cus0001	22483863	mgr0001
cus0002	22469711	mgr0001
cus0003	22461161	mgr0001
cus0004	20428161	mgr0001
cus0005	20428181	mgr0001

**Table 7: Savings Account**

```

INSERT INTO SavingsAccount VALUES (2022483863,10000);
INSERT INTO SavingsAccount VALUES (2022469711,15000);
INSERT INTO SavingsAccount VALUES (2022461161,30000);
INSERT INTO SavingsAccount VALUES (2020428161,18000);
INSERT INTO SavingsAccount VALUES (2020428181,2000);
SELECT * from SavingsAccount;

```

S_ACC_NO	BALANCE
2022483863	10000
2022469711	15000
2022461161	30000
2020428161	18000
2020428181	2000

**Table 8: Loan Info**

```

INSERT INTO LoanInfo VALUES ('Loan001',15000,'10-March-2022','30-Sep-2022');
INSERT INTO LoanInfo VALUES ('Loan002',10000,'02-April-2022','30-Oct-2022');
INSERT INTO LoanInfo VALUES ('Loan003',50000,'15-February-2022','30-Mar-2023');
INSERT INTO LoanInfo VALUES ('Loan004',30000,'20-January-2023','29-Feb-2024');
INSERT INTO LoanInfo VALUES ('Loan005',20000,'07-June-2022','30-Jan-2023');
SELECT * from LoanInfo;

```

L_NO	L_AMOUNT	TAKEN_DATE	EXP_DATE
Loan001	15000	10-MAR-22	30-SEP-22
Loan002	10000	02-APR-22	30-OCT-22
Loan003	50000	15-FEB-22	30-MAR-23
Loan004	30000	20-JAN-23	29-FEB-24
Loan005	20000	07-JUN-22	30-JAN-23



**Table 9: Savings Acc Loan Registration**

```
INSERT INTO Savings_AccLoanRegistration VALUES ('Loan001',2022483863);
INSERT INTO Savings_AccLoanRegistration VALUES ('Loan002',2022469711);
INSERT INTO Savings_AccLoanRegistration VALUES ('Loan003',2022461161);
INSERT INTO Savings_AccLoanRegistration VALUES ('Loan004',2020428161);
INSERT INTO Savings_AccLoanRegistration VALUES ('Loan005',2020428181);
SELECT * from Savings_AccLoanRegistration;
```

L_NO	S_ACC_NO
Loan001	2022483863
Loan002	2022469711
Loan003	2022461161
Loan004	2020428161
Loan005	2020428181

**Table 10: Current Account**

```
INSERT INTO CurrentAccount VALUES (0122483863,56000);
INSERT INTO CurrentAccount VALUES (0222469711,20000);
INSERT INTO CurrentAccount VALUES (0322461161,25000);
INSERT INTO CurrentAccount VALUES (0420428161,39000);
INSERT INTO CurrentAccount VALUES (0520428181,75000);
SELECT * from CurrentAccount;
```

C_ACC_NO	BALANCE
122483863	56000
222469711	20000
322461161	25000
420428161	39000
520428181	75000

**Table 11: Current Acc Loan Registration**

```
INSERT INTO Current_AccLoanRegistration VALUES ('Loan001',0122483863);
INSERT INTO Current_AccLoanRegistration VALUES ('Loan002',0222469711);
INSERT INTO Current_AccLoanRegistration VALUES ('Loan003',0322461161);
INSERT INTO Current_AccLoanRegistration VALUES ('Loan004',0420428161);
INSERT INTO Current_AccLoanRegistration VALUES ('Loan005',0520428181);
SELECT * from Current_AccLoanRegistration;
```

L_NO	C_ACC_NO
Loan001	122483863
Loan002	222469711
Loan003	322461161
Loan004	420428161
Loan005	520428181

**Table 12: Customer Loan Registration**

```

INSERT INTO CustomerLoanRegistration VALUES ('cus0001','Loan001');
INSERT INTO CustomerLoanRegistration VALUES ('cus0002','Loan002');
INSERT INTO CustomerLoanRegistration VALUES ('cus0003','Loan003');
INSERT INTO CustomerLoanRegistration VALUES ('cus0004','Loan004');
INSERT INTO CustomerLoanRegistration VALUES ('cus0005','Loan005');
SELECT * from CustomerLoanRegistration;

```

C_ID	L_NO
cus0001	Loan001
cus0002	Loan002
cus0003	Loan003
cus0004	Loan004
cus0005	Loan005

**Table 13: Loan Officer**

```

INSERT INTO LoanOfficer VALUES ('LO_0001','Ahsan',01627257483,20000);
INSERT INTO LoanOfficer VALUES ('LO_0002','Sadikuzzaman',01760496799,20000);
INSERT INTO LoanOfficer VALUES ('LO_0003','Shahriar',01827257483,20000);
INSERT INTO LoanOfficer VALUES ('LO_0004','Taufiqur',01927257483,20000);
SELECT * from LoanOfficer;

```

L_O_ID	L_O_NAME	MOBILE_NUMBER	SALARY
LO_0001	Ahsan	1627257483	20000
LO_0002	Sadikuzzaman	1760496799	20000
LO_0003	Shahriar	1827257483	20000
LO_0004	Taufiqur	1927257483	20000

**Table 14: Loan Register to Loan Officer**

```

INSERT INTO LoanRegistertoLoanOfficer VALUES ('L_01','LO_0001');
INSERT INTO LoanRegistertoLoanOfficer VALUES ('L_02','LO_0002');
INSERT INTO LoanRegistertoLoanOfficer VALUES ('L_03','LO_0003');
INSERT INTO LoanRegistertoLoanOfficer VALUES ('L_04','LO_0004');

```

```
INSERT INTO LoanRegistertoLoanOfficer VALUES ('L_05','LO_0002');
SELECT * from LoanRegistertoLoanOfficer;
```

L_ID	L_O_ID
L_01	LO_0001
L_02	LO_0002
L_03	LO_0003
L_04	LO_0004
L_05	LO_0002

**Table 15: Bank Info**

```
INSERT INTO BankInfo VALUES ('DB13082023','TAKE CARE BANK','Dhaka');
SELECT * from BankInfo;
```

CODE	B_NAME	ADDRESS
DB13082023	TAKE CARE BANK	Dhaka

**Table 16: Bank Address**

```
INSERT INTO BankAddress_Central VALUES (59/1,'Kuratoli','Dhaka','Bangladesh');
SELECT * from BankAddress_Central;
```

H_NO	AREA	CITY	COUNTRY
59	Kuratoli	Dhaka	Bangladesh

**Table 17: Loan Officer of Bank**

```
INSERT INTO LoanOfficerofBank VALUES ('LO_0001','DB13082023');
INSERT INTO LoanOfficerofBank VALUES ('LO_0002','DB13082023');
INSERT INTO LoanOfficerofBank VALUES ('LO_0003','DB13082023');
INSERT INTO LoanOfficerofBank VALUES ('LO_0004','DB13082023');
SELECT * from LoanOfficerofBank;
```

L_O_ID	CODE
LO_0001	DB13082023
LO_0002	DB13082023
LO_0003	DB13082023
LO_0004	DB13082023

**Table 18: Branch Info**

```
INSERT INTO BranchInfo VALUES ('Main','Kuratoli','Dhaka');
INSERT INTO BranchInfo VALUES ('B001','Gazipur','Tongi,Gazipur');
INSERT INTO BranchInfo VALUES ('B002','Uttara','Dhaka');
INSERT INTO BranchInfo VALUES ('B003','Rampura','Dhaka');
SELECT * from BranchInfo;
```

BRANCH_CODE	BRANCH_NAME	ADDRESS
Main	Kuratoli	Dhaka
B001	Gazipur	Tongi,Gazipur
B002	Uttara	Dhaka
B003	Rampura	Dhaka

**Table 19: Branch Registration**

```
INSERT INTO BranchRegistration VALUES ('Main','DB13082023');
INSERT INTO BranchRegistration VALUES ('B001','DB13082023');
INSERT INTO BranchRegistration VALUES ('B002','DB13082023');
INSERT INTO BranchRegistration VALUES ('B003','DB13082023');
SELECT * from BranchRegistration;
```

BRANCH_CODE	CODE
Main	DB13082023
B001	DB13082023
B002	DB13082023
B003	DB13082023

## 8. Query

### Aggregate Function:

- SELECT L\_O\_Name, L\_O\_ID, MOBILE\_NUMBER, SUM(SALARY)  
FROM LOANOFFICER WHERE L\_O\_Name LIKE '%a%'  
GROUP BY L\_O\_Name, L\_O\_ID, MOBILE\_NUMBER;

L_O_NAME	L_O_ID	MOBILE_NUMBER	SUM(SALARY)
Taufiqur	LO_0004	1927257483	20000
Ahsan	LO_0001	1627257483	20000
Shahriar	LO_0003	1827257483	20000
Sadikuzzaman	LO_0002	1760496799	20000

### Single Row Function:

- SELECT L\_no,  
TO\_DATE(Exp\_date, 'YYYY-MM-DD') - TO\_DATE(taken\_date, 'YYYY-MM-DD')  
AS Days\_Difference  
FROM LoanInfo  
WHERE L\_no = 'Loan001';

L_NO	DAYS_DIFFERENCE
Loan001	7489

- SELECT M\_ID, M\_Name, CONCAT (M\_Name, M\_ID),  
LENGTH(M\_Name), INSTR(M\_Name, 'A') FROM ManagerInfo;

M_ID	M_NAME	CONCAT(M_NAME,M_ID)	LENGTH(M_NAME)	INSTR(M_NAME,'A')
mgr0001	Fahim	Fahimmgr0001	5	0

### Multiple Row Function:

- SELECT BRANCH\_CODE  
FROM BRANCHREGISTRATION  
WHERE CODE = 'DB13082023'  
GROUP BY BRANCH\_CODE;

BRANCH_CODE
B003
Main
B002
B001

### Single Row Sub Query:

- SELECT C\_Name, Mobile\_No,  
(SELECT Balance FROM AccountInfo WHERE Acc\_no = '22483863') AS  
Account\_Balance  
FROM CustomerInfo  
WHERE C\_ID = 'cus0001';

C_NAME	MOBILE_NO	ACCOUNT_BALANCE
Habib	1854056655	30000

### Multiple row Sub Query:

- SELECT L\_no, L\_amount  
FROM LoanInfo  
WHERE L\_amount >  
(SELECT AVG(L\_amount) FROM LoanInfo);

L_NO	L_AMOUNT
Loan003	50000
Loan004	30000

### Joining:

#### Equijoin:

- SELECT ci.C\_Name, ci.Mobile\_No, ci.C\_Email, ca.Acc\_no FROM CustomerInfo ci, CustomerAccount ca WHERE ci.C\_ID = ca.C\_ID;

C_NAME	MOBILE_NO	C_EMAIL	ACC_NO
Habib	1854056655	habib@gmail.com	22483863
Rakib	1754053455	rakib@gmail.com	22469711
Rahman	1854564658	rahman@gmail.com	22461161
Saikat	1995586787	saikat@gmail.com	20428161
Toufiq	1998456355	toufiq@gmail.com	20428181

#### Outer join:

- SELECT ci.C\_ID, ci.C\_Name,  
(SELECT MAX(Balance) FROM AccountInfo ai  
WHERE ai.Acc\_no = ca.Acc\_no)  
AS Balance FROM CustomerInfo ci,  
CustomerAccount ca  
WHERE ci.C\_ID = ca.C\_ID (+);

C_ID	C_NAME	BALANCE
cus0001	Habib	30000
cus0002	Rakib	40000
cus0003	Rahman	70000
cus0004	Saikat	50000
cus0005	Toufiq	10000

## 9. Conclusion

In summary, the Banking Management System project has created a well-organized database for handling different parts of banking. It's like arranging puzzle pieces – customer, account, loans, banks and branches- so everything fits together smoothly. This makes sure that all the information is accurate and helps in managing banking tasks effectively.