PROJECT 1

BANKING MANAGEMENT SYSTEM (HDFC)

New Employee Personal and Banking Details Form

BIG DATA FOR MANAGERS & ANALYTICS- 1

FORE School of Management
New Delhi

Submitted to: Prof. Ashok Harnal

Submitted by: Group 6

Hritika Gupta, 045020

Khushi, 045027

Nandini Arya, 045034

BDA Batch- 04

Bank Management System (BMS) Database Design

The Bank Management System (BMS) database is meticulously designed to manage and organise various aspects of a bank's operations, including customer data management, financial transactions, loans, and employee information. This database is critical for ensuring that the bank's processes are efficient, secure, and accessible. It provides a structured way to store, retrieve, and analyse information related to customers, their accounts, transactions, and other banking activities.

This database supports the bank's operations by providing a secure and organised system to manage the following:

- **Customer Information**: Centralized storage of customer data for easy retrieval and management.
- **Account and Transaction Management**: Ensures accurate and timely processing of financial transactions.
- **Loan Management**: Tracks loan details and repayment schedules to ensure effective loan servicing.
- **Card Management**: Manages the issuance and maintenance of debit and credit cards.
- **Branch and Employee Information**: Organizes branch and employee data for operational efficiency.
- **Security**: Safeguards customer data through secure login systems and encrypted storage of sensitive information.

Database Components:

- **Customers Table:** Stores detailed information about the bank's customers. Each record is unique to a customer, capturing essential personal details like name, contact information, and account type.

- **Accounts Table:** This table contains information about the various accounts held by customers, including savings, checking, and other account types. It tracks the account number, type, and current balance.

- **Transactions Table:** Records all financial transactions involving bank accounts, including deposits, withdrawals, and transfers. Each transaction is linked to a specific account and includes the transaction amount, date, and type.

- **Loans Table:** Manages information about loans provided to customers. This includes the type of loan, the amount borrowed, interest rate, and the repayment schedule.

- **Cards Table:** Contains information about the debit and credit cards issued to customers. Details include the card number, type (debit/credit), expiry date, and card limit.

- **Branch Table:** Stores information about the bank's branches, including branch name, address, contact details, and the branch manager's information. Each branch is uniquely identified by an ID.

- **Employees Table:** Records data about bank employees, including their name, role, salary, and the branch where they are employed.

- **Account_Holders Table:** Manages the relationship between accounts and their holders, particularly in cases where an account has multiple holders (e.g., joint accounts).

- **Loan_Payments Table:** Tracks payments made towards loans. This table records the payment amount, date, and the remaining balance for each loan.

- **Fixed_Deposits Table:** Manages fixed deposit accounts held by customers. It includes information like deposit amount, interest rate, start date, and maturity date.

- **Loan_Applications Table:** Records the loan applications submitted by customers. The table tracks the application status, loan type, amount requested, and the date of application.

- **Beneficiaries Table:** Contains information about beneficiaries added by customers for fund transfers. It includes the beneficiary’s name, account number, and bank details.

- **User_Login Table:** Manages login credentials for customers accessing the bank's online services. It stores the username, hashed password, and a link to the corresponding customer record.

Relationship Table

Relationship	Type	Description
Customers to Accounts	One-to-Many	A customer can have multiple accounts, but each account belongs to one customer.
Accounts to Transactions	One-to-Many	An account can have multiple transactions, but each transaction is linked to one account.
Customers to Loans	One-to-Many	A customer can take out multiple loans, but each loan is associated with one customer.
Customers to Cards	One-to-Many	A customer can have multiple cards, but each card is linked to one customer.
Branch to Employees	One-to-Many	A branch can employ multiple employees, but each employee works at one branch.
Accounts to Account_Holders	One-to-Many	An account can have multiple holders (e.g., joint accounts), but each holder is linked to one account.

Loans to Loan_Payments	One-to-Many	A loan can have multiple payments, but each payment is associated with one loan.
Customers to Beneficiaries	One-to-Many	A customer can have multiple beneficiaries for fund transfers, but each beneficiary is linked to one customer.
Customers to User_Login	One-to-One	Each customer has one unique login credential, but a login credential corresponds to one customer.
Branch to Accounts	One-to-Many	A branch can manage multiple accounts, but each account is associated with one branch.
Employees to Branch	Many-to-One	Multiple employees can work at one branch, but each employee is linked to one branch.
Loans to Loan_Applications	One-to-Many	A loan application can lead to one loan, but a loan can have multiple applications (if applicable).

SQL Statements:

1. **Customers:**

```
Create table Customers (
Customer_id char(6) Primary key,
First_name varchar(20) not null,
Last_name varchar(20) not null,
Gender enum('Male', 'Female', 'Other'),
dob date not null,
Age int not null check(Age>=18),
Address varchar(50),
Pan_no char(10),
Phone_number char(10) not null unique,
Email varchar(40) unique
);
```

```
mysql> desc Customers;
+--------------+-------------------------------+------+-----+---------+-------+
| Field        | Type                          | Null | Key | Default | Extra |
+--------------+-------------------------------+------+-----+---------+-------+
| Customer_id  | char(6)                       | NO   | PRI | NULL    |       |
| First_name   | varchar(20)                   | NO   |     | NULL    |       |
| Last_name    | varchar(20)                   | NO   |     | NULL    |       |
| Gender       | enum('Male','Female','Other') | YES  |     | NULL    |       |
| dob          | date                          | NO   |     | NULL    |       |
| Age          | int                           | NO   |     | NULL    |       |
| Address      | varchar(50)                   | YES  |     | NULL    |       |
| Pan_no       | char(10)                      | YES  |     | NULL    |       |
| Phone_number | char(10)                      | NO   | UNI | NULL    |       |
| Email        | varchar(40)                   | YES  | UNI | NULL    |       |
+--------------+-------------------------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

```
INSERT INTO Customers
VALUES (110003, 'Laxmi', 'Gupta’', 'Female', '1995-03-14',
FLOOR(DATEDIFF(CURDATE(), '1995-03-14') / 365), 'Narayan Nagar New Delhi',
'XYZAB1235P', '129012901', 'laxmi.gupta@gmail.com');

INSERT INTO Customers
VALUES (110002, 'Laxman', 'Singh', 'Male', '1995-03-15',
FLOOR(DATEDIFF(CURDATE(), '1995-03-15') / 365), 'Malviya Nagar New Delhi',
'XYZAB1234G', '129012902', 'laxman.singh@gmail.com');
```

2. **Accounts**

```
CREATE TABLE Accounts (
account_number CHAR(14) PRIMARY KEY,
Customer_id CHAR(6) NOT NULL,
 account_type ENUM('Savings', 'Checking') NOT NULL,
 balance DECIMAL(15, 2) NOT NULL CHECK (balance>=0),
 created_at DATE NOT NULL,
 FOREIGN KEY (Customer_id) REFERENCES
 Customers(Customer_id) ON DELETE CASCADE
 );
```

```
mysql> desc Accounts;
+----------------+----------------------------+------+-----+---------+-------+
| Field          | Type                       | Null | Key | Default | Extra |
+----------------+----------------------------+------+-----+---------+-------+
| account_number | char(14)                   | NO   | PRI | NULL    |       |
| Customer_id    | char(6)                    | NO   | MUL | NULL    |       |
| account_type   | enum('Savings','Checking') | NO   |     | NULL    |       |
| balance        | decimal(15,2)              | NO   |     | NULL    |       |
| created_at     | date                       | NO   |     | NULL    |       |
+----------------+----------------------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
INSERT INTO Accounts VALUES ('23456789012345', '110002', 'Checking', 3000.00,
'2023-02-15');

INSERT INTO Accounts VALUES ('23456789012347', '110003', 'Checking', 5000.00,
'2024-02-15');
```

3. **Transactions**

CREATE TABLE Transactions (
transaction_id CHAR(11) PRIMARY KEY,
account_number CHAR(14),
transaction_type ENUM('Credit', 'Debit') NOT NULL,
amount DECIMAL(15, 2) NOT NULL,
transaction_date DATE NOT NULL,
description VARCHAR(255),
FOREIGN KEY (account_number) REFERENCES Accounts(account_number) ON DELETE CASCADE
);

```
mysql> desc Transactions;
+------------------+-------------------------+------+-----+---------+-------+
| Field            | Type                    | Null | Key | Default | Extra |
+------------------+-------------------------+------+-----+---------+-------+
| transaction_id   | char(11)                | NO   | PRI | NULL    |       |
| account_number   | char(14)                | YES  | MUL | NULL    |       |
| transaction_type | enum('Credit','Debit') | NO   |     | NULL    |       |
| amount           | decimal(15,2)           | NO   |     | NULL    |       |
| transaction_date | date                    | NO   |     | NULL    |       |
| description      | varchar(255)            | YES  |     | NULL    |       |
+------------------+-------------------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

INSERT INTO Transactions
VALUES ('T0000000001', '23456789012347”', 'Credit', 1000.00, '2023-03-01', 'Salary deposit');

INSERT INTO Transactions
VALUES ('T0000000002', '23456789012345', 'Debit', 500.00, '2023-03-15', 'ATM withdrawal');

4. **Loans Table**

CREATE TABLE Loans (
loan_id char(11) Primary key,
Customer_id CHAR(6),
loan_type ENUM("Personal", "Home", "Auto", "Education", "Business", "Mortgage", "Agricultural", "Gold") not null,
loan_amount DECIMAL(15, 2) not null,
interest_rate DECIMAL(5, 2) not null,
loan_start_date DATE not null,
loan_end_date DATE not null,
outstanding_amount DECIMAL(15, 2) not null check(outstanding_amount >0),
foreign key (Customer_id) references Customers(Customer_id)
);

```
mysql> desc Loans;
+-------------------+-----------------------------------------------------------------------------------------------------------------+------+-----+---------+-------+
| Field             | Type                                                                                                            | Null | Key | Default | Extra |
+-------------------+-----------------------------------------------------------------------------------------------------------------+------+-----+---------+-------+
| loan_id           | char(11)                                                                                                        | NO   | PRI | NULL    |       |
| Customer_id       | char(6)                                                                                                         | YES  | MUL | NULL    |       |
| loan_type         | enum('Personal','Home','Auto','Education','Business','Mortgage','Agricultural','Gold') | NO   |     | NULL    |       |
| loan_amount       | decimal(15,2)                                                                                                   | NO   |     | NULL    |       |
| interest_rate     | decimal(5,2)                                                                                                    | NO   |     | NULL    |       |
| loan_start_date   | date                                                                                                            | NO   |     | NULL    |       |
| loan_end_date     | date                                                                                                            | NO   |     | NULL    |       |
| outstanding_amount | decimal(15,2)                                                                                                  | NO   |     | NULL    |       |
+-------------------+-----------------------------------------------------------------------------------------------------------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

INSERT INTO Loans VALUES ('L0000000001', '110003', 'Personal', 10000.00, 5.5,'2023-01-15', '2024-01-15', 8000.00);

INSERT INTO Loans VALUES ('L0000000002', '110002', 'Home', 200000.00, 3.5, '2023-02-01', '2033-02-01', 195000.00);

5. **Cards Table**

CREATE TABLE Cards (
card_number char(16),
Customer_id CHAR(6),
card_type ENUM('Credit', 'Debit') not null,
expiry_date DATE not null,
cvv CHAR(3) not null unique,
card_limit DECIMAL(15, 2) not null,
Primary key (card_number),
foreign key (Customer_id) references Customers(Customer_id)
);

```
mysql> desc Cards;
+-------------+-----------------------+------+-----+---------+-------+
| Field       | Type                  | Null | Key | Default | Extra |
+-------------+-----------------------+------+-----+---------+-------+
| card_number | char(16)              | NO   | PRI | NULL    |       |
| Customer_id | char(6)               | YES  | MUL | NULL    |       |
| card_type   | enum('Credit','Debit') | NO   |     | NULL    |       |
| expiry_date | date                  | NO   |     | NULL    |       |
| cvv         | char(3)               | NO   | UNI | NULL    |       |
| card_limit  | decimal(15,2)         | NO   |     | NULL    |       |
+-------------+-----------------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

INSERT INTO Cards VALUES ('9876543210123456', '110003', 'Credit', '2025-07-31', '123', 100000.00);

INSERT INTO Cards VALUES ('9876543210123457', '110002', 'Debit', '2024-12-31', '456', 50000.00);

6. **Branch Table**

CREATE TABLE Branch (
branch_id char(11),
branch_name VARCHAR(100) not null unique,
branch_address VARCHAR(255) not null,
branch_phone CHAR(10) not null unique,
branch_manager VARCHAR(100) not null,
Primary key (branch_id)
);

```
mysql> desc Branch;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| branch_id      | char(11)     | NO   | PRI | NULL    |       |
| branch_name    | varchar(100) | NO   | UNI | NULL    |       |
| branch_address | varchar(255) | NO   |     | NULL    |       |
| branch_phone   | char(10)     | NO   | UNI | NULL    |       |
| branch_manager | varchar(100) | NO   |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

INSERT INTO Branch
VALUES ('BR001', 'Narayan Branch', 'Narayan, New Delhi', '0114000001', 'Mr. Sharma');

INSERT INTO Branch
VALUES ('BR002', 'Malviya Nagar Branch', 'Malviya Nagar, New Delhi', '0114000002', 'Ms. Kapoor');

7. **Employees Table**

CREATE TABLE Employees (
employee_id CHAR(5) PRIMARY KEY,
first_name VARCHAR(50) NOT NULL,
last_name VARCHAR(50) NOT NULL,
branch_id CHAR(11) NOT NULL,
role ENUM('Manager', 'Teller', 'Loan Officer', 'Customer Service', 'IT Support') NOT NULL,
salary DECIMAL(10, 2) NOT NULL CHECK (salary > 0),
hire_date DATE NOT NULL,
FOREIGN KEY (branch_id) REFERENCES Branch(branch_id) ON DELETE CASCADE
);

```
mysql> desc  Employees;
+-------------+-----------------------------------------------------------------------------+------+-----+-----
----+-------+
| Field       | Type                                                                        | Null | Key | Defa
ult | Extra |
+-------------+-----------------------------------------------------------------------------+------+-----+-----
----+-------+
| employee_id | char(5)                                                                     | NO   | PRI | NULL
    |       |
| first_name  | varchar(50)                                                                 | NO   |     | NULL
    |       |
| last_name   | varchar(50)                                                                 | NO   |     | NULL
    |       |
| branch_id   | char(11)                                                                    | NO   | MUL | NULL
    |       |
| role        | enum('Manager','Teller','Loan Officer','Customer Service','IT Support') | NO   |     | NULL
    |       |
| salary      | decimal(10,2)                                                               | NO   |     | NULL
    |       |
| hire_date   | date                                                                        | NO   |     | NULL
    |       |
+-------------+-----------------------------------------------------------------------------+------+-----+-----
----+-------+
7 rows in set (0.00 sec)
```

INSERT INTO Employees
VALUES ('EMP01', 'Radha', 'Kumai', 'BR001', 'Manager', 75000.00, '2019-05-15');

INSERT INTO Employees
VALUES ('EMP02', 'Shyam', 'Singh', 'BR002', 'Teller', 35000.00, '2021-08-01');

8. **Account_Holders Table**

```
CREATE TABLE Account_Holders (
account_holder_id CHAR(5) PRIMARY KEY,
account_number CHAR(14) NOT NULL,
Customer_id CHAR(6) NOT NULL,
relationship_type ENUM('Primary', 'Joint') NOT NULL,
FOREIGN KEY (account_number) REFERENCES Accounts(account_number) ON
DELETE CASCADE,
FOREIGN KEY (Customer_id) REFERENCES Customers(Customer_id) ON DELETE
CASCADE
);
```

```
mysql> desc Account_Holders;
+------------------+-------------------------+------+-----+---------+-------+
| Field            | Type                    | Null | Key | Default | Extra |
+------------------+-------------------------+------+-----+---------+-------+
| account_holder_id | char(5)                 | NO   | PRI | NULL    |       |
| account_number    | char(14)                | NO   | MUL | NULL    |       |
| Customer_id       | char(6)                 | NO   | MUL | NULL    |       |
| relationship_type | enum('Primary','Joint') | NO   |     | NULL    |       |
+------------------+-------------------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
INSERT INTO Account_Holders VALUES ('AH001', '23456789012345', '110002',
'Primary');
INSERT INTO Account_Holders VALUES ('AH002', '23456789012347', '110003',
'Joint');
```

9. Loan_Payments Table

```
Create table Loan_Payments(
payment_id char(5) Primary key,
loan_id char(11) not null unique,
Payment_date DATE not null,
Payment_amount Decimal(15,2) check(Payment_amount > 0),
Remaining_balance Decimal(15,2) check(Remaining_balance >= 0),
foreign key(loan_id ) references Loans(loan_id)
);
```

```
mysql> desc Loan_Payments;
+-------------------+---------------+------+-----+---------+-------+
| Field             | Type          | Null | Key | Default | Extra |
+-------------------+---------------+------+-----+---------+-------+
| payment_id        | char(5)       | NO   | PRI | NULL    |       |
| loan_id           | char(11)      | NO   | UNI | NULL    |       |
| Payment_date      | date          | NO   |     | NULL    |       |
| Payment_amount    | decimal(15,2) | YES  |     | NULL    |       |
| Remaining_balance | decimal(15,2) | YES  |     | NULL    |       |
+-------------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

```
INSERT INTO Loan_Payments
VALUES ('PAY01', 'L0000000001', '2023-08-01', 50000.00, 1450000.00);

INSERT INTO Loan_Payments
VALUES ('PAY02', 'L0000000002', '2023-08-02', 25000.00, 4250000.00);
```

10. Fixed_Deposits Table

```
Create table Fixed_Deposits(
Fd_id char(5) Primary key,
account_number char(6) not null,
Customer_id char(6) not null,
Deposit_amount DECIMAL(15, 2) not null check (deposit_amount > 0),
Interest_rate DECIMAL(15, 2) not null check(interest_rate > 0 AND interest_rate <=
100), start_date DATE not null,
maturity_date DATE not null,
maturity_amount DECIMAL(15, 2) GENERATED ALWAYS AS (Deposit_amount *
POWER(1 + (Interest_rate/100), DATEDIFF(maturity_date, start_date) / 365))
STORED,
UNIQUE (account_number, start_date),
check (maturity_date > start_date),
foreign key (account_number) references Accounts(account_number),
foreign key (Customer_id) references Customers(Customer_id)
);
```

```
mysql> desc Fixed_Deposits;
+-----------------+---------------+------+-----+---------+-------------------+
| Field           | Type          | Null | Key | Default | Extra             |
+-----------------+---------------+------+-----+---------+-------------------+
| Fd_id           | char(5)       | NO   | PRI | NULL    |                   |
| account_number  | char(6)       | NO   | MUL | NULL    |                   |
| Customer_id     | char(6)       | NO   | MUL | NULL    |                   |
| Deposit_amount  | decimal(15,2) | NO   |     | NULL    |                   |
| Interest_rate   | decimal(15,2) | NO   |     | NULL    |                   |
| start_date      | date          | NO   |     | NULL    |                   |
| maturity_date   | date          | NO   |     | NULL    |                   |
| maturity_amount | decimal(15,2) | YES  |     | NULL    | STORED GENERATED  |
+-----------------+---------------+------+-----+---------+-------------------+
8 rows in set (0.00 sec)
```

```
INSERT INTO Fixed_Deposits
(Fd_id, account_number, Customer_id, Deposit_amount, Interest_rate, start_date,
maturity_date)
VALUES ('FD002', '234567', '110002', 50000.00, 7.0, '2023-03-15', '2025-03-15');

INSERT INTO Fixed_Deposits
(Fd_id, account_number, Customer_id, Deposit_amount, Interest_rate, start_date,
maturity_date)
VALUES ('FD001', '234568', '110003', 25000.00, 7.0, '2023-03-14', '2025-03-15');
```

11. Loan_Applications Table

```
Create table Loan_Applications(
 application_id char(6) Primary key,
 Customer_id char(6) not null unique,
 loan_type enum("Personal", "Home", "Auto", "Education", "Business", "Mortgage",
 "Agricultural", "Gold") not null unique, application_date DATE not null,
 loan_amount DECIMAL(15, 2) not null check (loan_amount > 0),
 interest_rate DECIMAL(5, 2) not null,
 loan_term int not null check (loan_term > 0),
 application_status enum('Pending', 'Approved', 'Rejected') not null DEFAULT 'Pending',
 foreign key (Customer_id) references Customers(Customer_id)
 );
```

```
mysql> desc Loan_Applications;
+--------------------+-------------------------------------------------------------------------------------------------
---+------+-----+---------+-------+
| Field              | Type
   | Null | Key | Default | Extra |
+--------------------+-------------------------------------------------------------------------------------------------
---+------+-----+---------+-------+
| application_id     | char(6)
   | NO   | PRI | NULL    |       |
| Customer_id        | char(6)
   | NO   | UNI | NULL    |       |
| loan_type          | enum('Personal','Home','Auto','Education','Business','Mortgage','Agricultural','Gold
') | NO   | UNI | NULL    |       |
| application_date   | date
   | NO   |     | NULL    |       |
| loan_amount        | decimal(15,2)
   | NO   |     | NULL    |       |
| interest_rate      | decimal(5,2)
   | NO   |     | NULL    |       |
| loan_term          | int
   | NO   |     | NULL    |       |
| application_status | enum('Pending','Approved','Rejected')
   | NO   |     | Pending |       |
+--------------------+-------------------------------------------------------------------------------------------------
---+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

```
INSERT INTO Loan_Applications
VALUES ('APP001', '110003', 'Home', '2023-01-05', 500000.00, 7.5, 120, 'Approved');

INSERT INTO Loan_Applications
VALUES ('APP002', '110002', 'Personal', '2023-01-10', 200000.00, 12.0, 60, 'Pending');
```

12. Beneficiaries Table

```
Create table Beneficiaries(
    Beneficiary_account_number char(12) Primary key,
    account_holder_id char(5) not null Unique,
    Beneficiary_name varchar(40) not null,
    bank_name varchar(6) not null,
    ifsc_code char(11) not null ,
    relationship enum('Family', 'Friend', 'Business', 'Other') not null,
    foreign key (account_holder_id) references Account_Holders(account_holder_id)
);
```

```
mysql> desc Beneficiaries;
+--------------------------+-----------------------------------------+------+-----+---------+-------+
| Field                    | Type                                    | Null | Key | Default | Extra |
+--------------------------+-----------------------------------------+------+-----+---------+-------+
| Beneficiary_account_number | char(12)                              | NO   | PRI | NULL    |       |
| account_holder_id        | char(5)                                 | NO   | UNI | NULL    |       |
| Beneficiary_name         | varchar(40)                             | NO   |     | NULL    |       |
| bank_name                | varchar(6)                              | NO   |     | NULL    |       |
| ifsc_code                | char(11)                                | NO   |     | NULL    |       |
| relationship             | enum('Family','Friend','Business','Other') | NO   |     | NULL    |       |
+--------------------------+-----------------------------------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

```
INSERT INTO Beneficiaries
VALUES ('BEN001', 'AH001', 'Sita Agarwaal', 'SBI', 'SBIN0001234', 'Family');

INSERT INTO Beneficiaries
VALUES ('BEN002', 'AH002', 'Shyam Singh', 'HDFC'', 'HDFC0005678', 'Friend');
```

13. User_Login Table

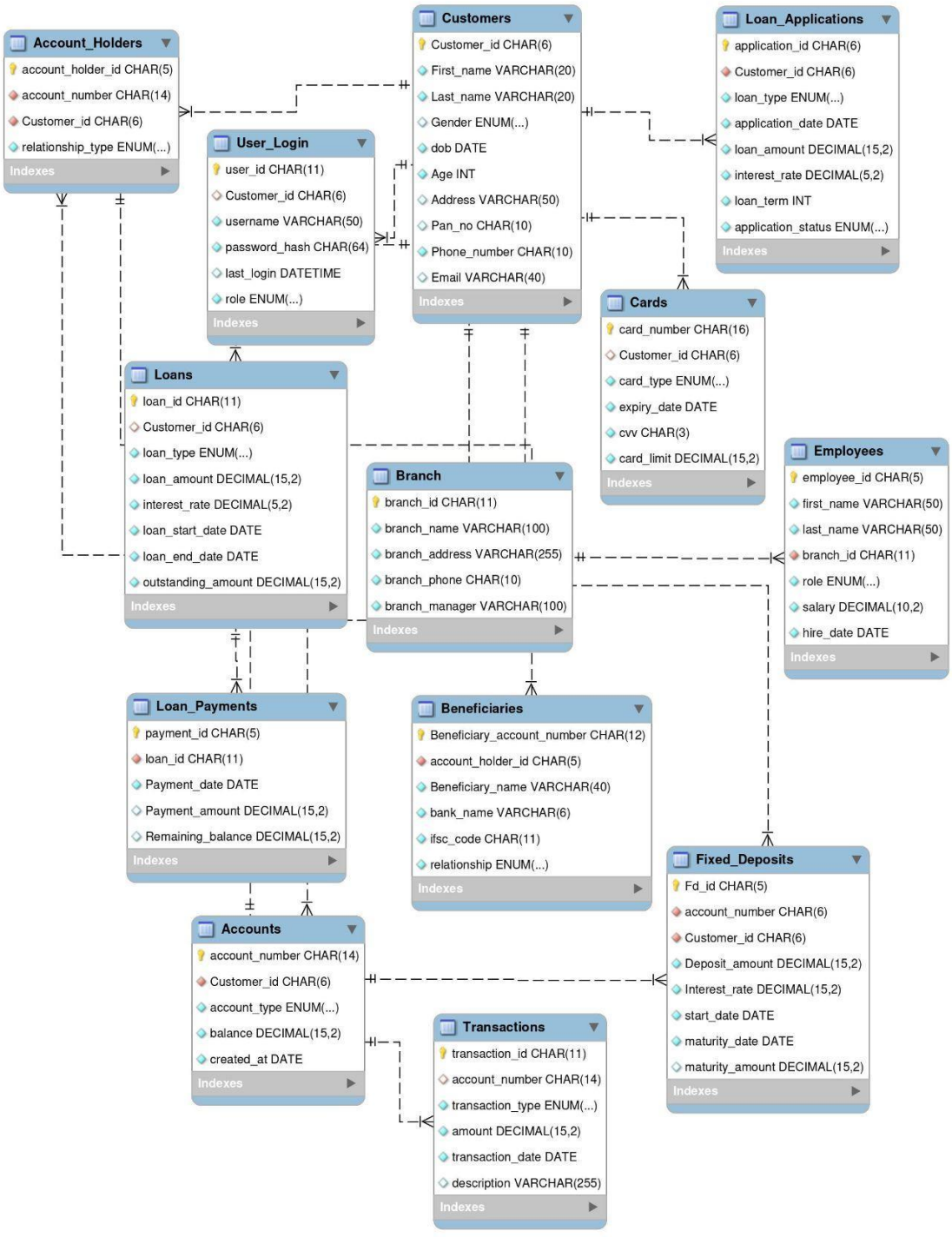
CREATE TABLE User_Login (
user_id CHAR(11) PRIMARY KEY,
Customer_id CHAR(6),
username VARCHAR(50) NOT NULL UNIQUE,
password_hash CHAR(64) NOT NULL, -- Assuming SHA-256 hash
last_login DATETIME,
role ENUM('Customer', 'Admin') NOT NULL,
FOREIGN KEY (Customer_id) REFERENCES Customers(Customer_id)
);

```
mysql> desc User_Login;
+---------------+--------------------------+------+-----+---------+-------+
| Field         | Type                     | Null | Key | Default | Extra |
+---------------+--------------------------+------+-----+---------+-------+
| user_id       | char(11)                 | NO   | PRI | NULL    |       |
| Customer_id   | char(6)                  | YES  | MUL | NULL    |       |
| username      | varchar(50)              | NO   | UNI | NULL    |       |
| password_hash | char(64)                 | NO   |     | NULL    |       |
| last_login    | datetime                 | YES  |     | NULL    |       |
| role          | enum('Customer','Admin') | NO   |     | NULL    |       |
+---------------+--------------------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

INSERT INTO User_Login
VALUES ('USER001', '110003', 'Laxmi Gupta', SHA2('ram_password', 256), '2024-08-01 10:00:00', 'Customer');

INSERT INTO User_Login
VALUES ('USER002', '110002', 'laxman.singh', SHA2('laxman_password', 256), '2024-08-01 11:00:00', 'Customer');

ERD Diagram:



GRANT Access:

Bank Administrator: The Bank Administrator grant all access as he needs full access to manage all aspects of the database.

Loan Officer: The Loan Officer needs to access and manage customer and loan-related information.

Teller/Clerk: The Teller/Clerk needs access to handle customer information, account transactions, and view transaction records.

GRANT Statements:

To bank_admin :

```
GRANT ALL PRIVILEGES ON Bank.* TO 'bank_admin'@'localhost';
```

To loan_officer:

```
-- Grant SELECT and INSERT access to the Customers, Loans, and Loan_Applications tables
GRANT SELECT, INSERT ON Bank.Customers TO 'loan_officer'@'localhost';
GRANT SELECT, INSERT ON Bank.Loans TO 'loan_officer'@'localhost';
GRANT SELECT, INSERT ON Bank.Loan_Applications TO 'loan_officer'@'localhost';

-- Grant UPDATE access to specific columns in the Loans and Loan_Applications tables
GRANT UPDATE (interest_rate, outstanding_amount) ON Bank.Loans TO 'loan_officer'@'localhost';
GRANT UPDATE (application_status) ON Bank.Loan_Applications TO 'loan_officer'@'localhost';
```

To Teller/Clerk:

```
-- Grant SELECT and INSERT access to the Customers and Accounts tables
GRANT SELECT, INSERT ON Bank.Customers TO 'teller'@'localhost';
GRANT SELECT, INSERT ON Bank.Accounts TO 'teller'@'localhost';

-- Grant UPDATE access to specific columns in the Customers and Accounts tables
GRANT UPDATE (address, email) ON Bank.Customers TO 'teller'@'localhost';
GRANT UPDATE (balance) ON Bank.Accounts TO 'teller'@'localhost';

-- Grant SELECT access to Transactions table
GRANT SELECT ON Bank.Transactions TO 'teller'@'localhost';
```