ANUDIP FOUNDATION

A Project Report on

LOAN MANAGEMENT SYSTEM

By

Batch: ANP-D0453

Student ID:AF0477150

Name: Usha V.Dornal

Under the Guidence of

Mrs. Rajshri Chandrabhan Thete

LOAN MANAGEMENT SYSTEM

The **Loan Management System (LMS)** is designed to streamline and automate the entire loan lifecycle, from loan application to repayment. It aims to enhance efficiency, accuracy, and customer service while minimizing manual errors and improving data management.

Entities:

- Admin
- Customer
- Branch
- Employee
- Loan System
- Department
- Loan
- EMI

VARIOUS ENTITIES:

1.Admin

- User_name
- Password

2.Customer

- Customer_name
- Customer_id
- Customer_ac_no
- Customer_address

3. Branch

- Branch_name
- Branch_id
- Branch_address

4. Employee

- Emp_id
- Emp_name
- Emp_address
- Emp_contact_no
- Branch_id

5. Loan System

- Name_of_branch
- No_of_branch

6. Department

- Dept_id
- Dept_name

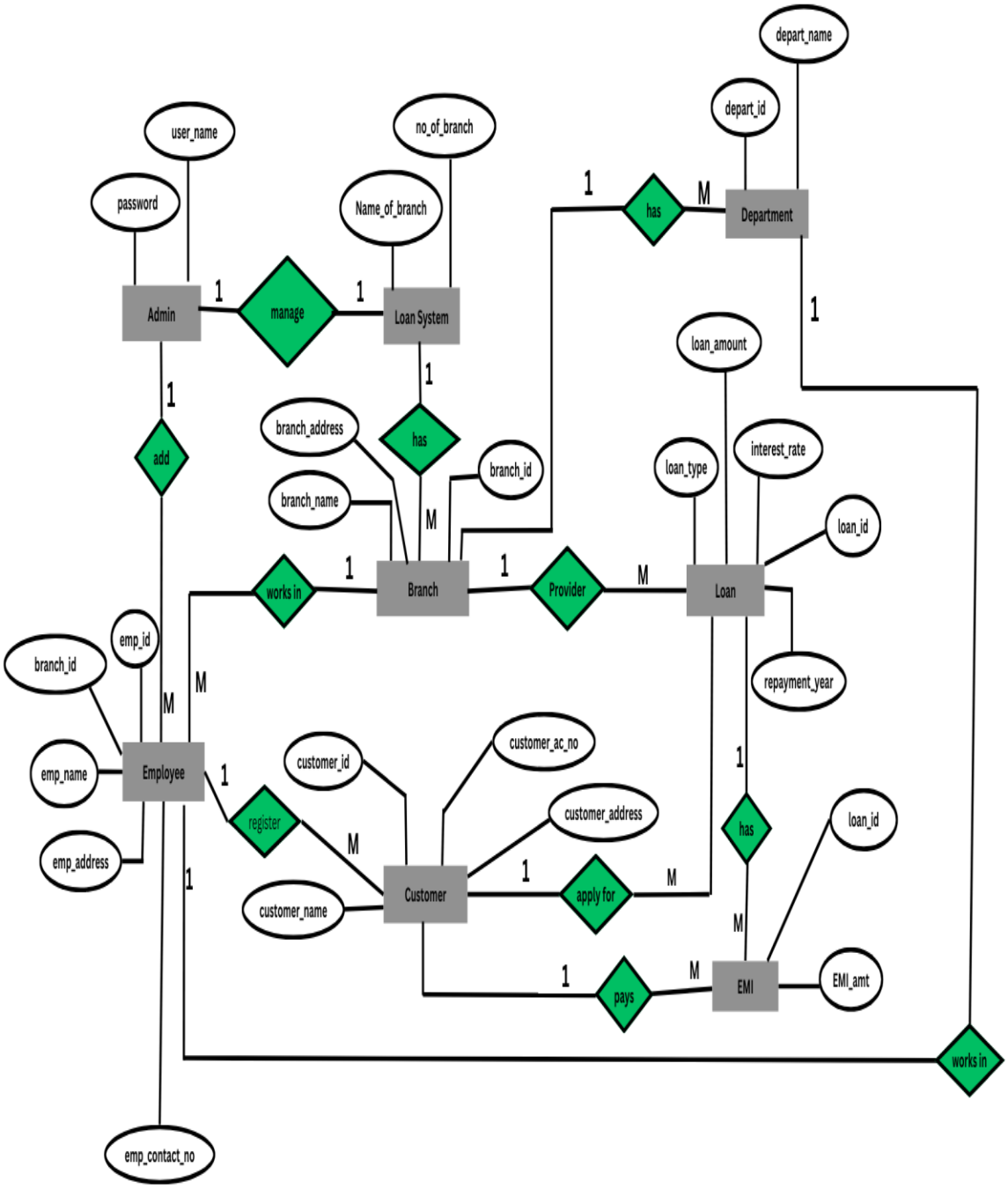
7. Loan

- Loan_type
- Loan_id
- Loan_amount
- Interest_rate
- Repayment_year

8. EMI

- Loan_id
- EMI_amt

ER-DIAGRAM:


depart_name
depart_id
user_name
no_of_branch
password
Name_of_branch
has
Department
Admin
manage
Loan System
loan_amount
add
has
branch_address
branch_id
loan_type
interest_rate
branch_name
loan_id
works in
Branch
Provider
Loan
emp_id
branch_id
repayment_year
customer_ac_no
customer_id
emp_name
Employee
register
customer_address
has
loan_id
emp_address
customer_name
Customer
apply for
pays
EMI
EMI_amt
works in
emp_contact_no

Database Connectivity:

Enter password: **

mysql> create database Loan_Management_System;

Query OK, 1 row affected (0.01 sec)

mysql> use Loan_Management_System;

Database changed

mysql> CREATE TABLE Admin (

-> admin_id INT PRIMARY KEY AUTO_INCREMENT,

-> admin_name VARCHAR(100) NOT NULL,

-> Admin_email VARCHAR(100) NOT NULL;

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 4

mysql> CREATE TABLE Admin (

-> admin_id INT PRIMARY KEY AUTO_INCREMENT,

-> admin_name VARCHAR(100) NOT NULL,

-> Admin_email VARCHAR(100) NOT NULL UNIQUE

->);

Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Loan_System (

```
   ->  system_id INT PRIMARY KEY AUTO_INCREMENT,
   -> Name_Of_branch VARCHAR(100) NOT NULL,
   ->  no_of_branch INT NOT NULL
   -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Department (
   ->  depart_id INT PRIMARY KEY AUTO_INCREMENT,
   -> depart_name VARCHAR(100) NOT NULL,
   ->   system_id INT,
   ->  FOREIGN KEY (system_id) REFERENCES Loan_System(system_id)
   -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Branch (
   ->  branch_id INT PRIMARY KEY AUTO_INCREMENT,
   ->  branch_name VARCHAR(100) NOT NULL,
   ->   branch_address VARCHAR(255) NOT NULL,
   -> system_id INT,
   ->  FOREIGN KEY (system_id) REFERENCES Loan_System(system_id)
   -> );
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> CREATE TABLE Employee (
    ->  emp_id INT PRIMARY KEY AUTO_INCREMENT,
    -> emp_name VARCHAR(100) NOT NULL,
    ->  emp_address VARCHAR(255) NOT NULL,
    ->  emp_contact_no VARCHAR(15) NOT NULL UNIQUE,
    ->   branch_id INT,
    ->  FOREIGN KEY (branch_id) REFERENCES Branch(branch_id)


    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE Customer (
    ->  customer_id INT PRIMARY KEY AUTO_INCREMENT,
    ->   customer_name VARCHAR(100) NOT NULL,
    ->   customer_ac_no VARCHAR(50) NOT NULL UNIQUE,
    ->  customer_address VARCHAR(255) NOT NULL,
    ->   branch_id INT,
    ->   FOREIGN KEY (branch_id) REFERENCES Branch(branch_id)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE Loan (
```

```
    ->   loan_id INT PRIMARY KEY AUTO_INCREMENT,
    ->  loan_amount DECIMAL(15,2) NOT NULL,
    ->  loan_type VARCHAR(50) NOT NULL,
    ->    interest_rate DECIMAL(5,2) NOT NULL,
    ->  repayment_year INT NOT NULL,
    ->  customer_id INT,
    ->    FOREIGN KEY (customer_id) REFERENCES Customer(customer_id)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE EMI (
    -> EMI_id INT PRIMARY KEY AUTO_INCREMENT,
    ->   EMI_amt DECIMAL(15,2) NOT NULL,
    ->   loan_id INT,
    ->  FOREIGN KEY (loan_id) REFERENCES Loan(loan_id)
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> show tables;
+--------------------------------+
| Tables_in_loan_management_system |
+--------------------------------+
| admin                          |
| branch                         |
| customer                       |
| department                     |
| emi                            |
| employee                       |
| loan                           |
| loan_system                    |
+--------------------------------+
```

CONCLUSION:

The **Loan Management System (LMS)** is designed to automate and streamline the loan lifecycle, enhancing efficiency and accuracy in managing loan applications, approvals, and repayments. It provides real-time tracking of customer information, loan details, and payment records while ensuring accurate EMI calculations and reducing manual errors. The system improves customer satisfaction by offering faster loan processing and better data management. With robust security measures and comprehensive reporting, it supports regulatory compliance and ensures data integrity. Overall, the LMS is a reliable, user-friendly, and scalable solution for efficient loan administration and financial operations.