**CONTENTS**

1. **ABSTRACT.**
2. **INTRODUCTION.**
3. **PROJECT DESCRIPTION.**
4. **ER DIAGRAM.**
5. **MODULE DESCRIPTION.**
6. **CONCLUSION**.

**ABSTRACT:**

The Card Management System is a Windows based application that takes care of the complete internal Card Processing activities of a Bank (management and maintenance of cards) like- Registration of Customers for card issue, Opening Card Accounts for the registered customer, Creation of Cards for registered customers, Hot Carding, making the Card ready for Embossing by creation of Card Data file etc. The Card Management System should interact with an external Card Transaction System, thereby making the Card-related information maintained by the bank, up-to-date.

It should make the customer easier to get a cash. Customer should be able to apply for credit card and know his eligibility from his own place just by giving his personal details.

**INTRODUCTION:**

**DATABASE:**

A database is a collection of information that is organized so that it can be easily accessed, managed and updated. Data is organized into rows, columns and tables, and it is indexed to make it easier to find relevant information. Data gets updated, expanded and deleted as new information is added. Databases process workloads to create and update themselves, querying the data they contain and running applications against it.

**DATABASE MANAGEMENT SYSTEM:**

A database management system (DBMS) is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases. A database is not generally portable across different DBMSs, but different DBMS can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Database management systems are often classified according to the database model that they support; the most popular database systems since the 1980s have all supported the relational model as represented by the SQL language. Sometimes a DBMS is loosely referred to as a 'database'.

**MYSQL:**

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. MySQL is the world's most popular open source database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more.

Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications.

* **MySQL is a database management system.**
* **MySQL databases are relational.**
* **MySQL software is Open Source.**
* **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**
* **MySQL Server works in client/server or embedded systems.**
* **A large amount of contributed MySQL software is available.**

Database is an organized collection of data. The data is typically organized to model aspects of reality in a way that supports processes requiring information. A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible. The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified and the database schema, which defines the database’s logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures. The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data.

**PROJECT DESCRIPTION:**

For the given description, we need to identify major data requirements, entities, attributes, relationships between the entities and to develop ER model and schema diagram by applying normalization techniques and to create tables with constraints to the identified schemas.

**LIST OF ENTITIES:**

1). banker

2). Loan.

3).Credit.

4). Account.

5). Customer.

6). Branch.

7). Stock.

**LIST OF ATTRIBUTES:**

Art Gallery Management System entities and their attributes:

* **Banker:** Attributes in Banker are baname,email,id,cust\_id,brnum .
* **Customerlogin Entity:** Attributes of Customerlogin are loannum,amount.
* **Loan Entity:** Attributes of Loanare order\_customer\_id, order\_type, order\_id, customer\_id.
* **Credit Entity:** Attributes of Credit are expiredate,lim,crnum.
* **Account Entity:** Attributes of Account are balance,accnum,category,
* **Customer Entity:** Attributes of Customer are cust\_name,cust\_id,address,city,loannum,crnum,accnum.
* **Branch Entity:** Attributes of Branch are brnum,brcity,brname,loannum

**ENTITIES:**

**Table Creation:**

**LOAN:**

mysql> create table loan(

-> loannum int(30),

-> amount int(30),

-> primary key(loannum));

Query OK, 0 rows affected (0.11 sec)

**CREDIT:**

mysql> create table credit(

-> expireddate int(30),

-> lim int(30),

-> crnum int(30),

-> primary key(crnum));

Query OK, 0 rows affected (0.13 sec)

**ACCOUNT:**

mysql> create table account(

-> balance int(30),

-> accnum int(30),

-> category varchar(30),

-> primary key(accnum));

Query OK, 0 rows affected (0.35 sec)

**CUSTOMER:**

mysql> create table customer(

-> cust\_name varchar(30),

-> cust\_id int(30),

-> address varchar(30),

-> city varchar(30),

-> loannum int(30),

-> accnum int(30),

-> crnum int(30),

-> primary key(cust\_id),

-> constraint cus\_lo foreign key(loannum) references loan(loannum),

-> constraint cus\_cr foreign key(crnum) references credit(crnum),

-> constraint cus\_acc foreign key(accnum) references account(accnum));

Query OK, 0 rows affected (0.41 sec)

**BRANCH:**

mysql> create table branch(

-> brname varchar(30),

-> brcity varchar(30),

-> brnum int(30),

-> loannum int(30),

-> primary key(brnum),

-> constraint br\_lo foreign key(loannum) references loan(loannum));

Query OK, 0 rows affected (0.11 sec)

**BANKER:**

mysql> create table banker(

-> baname varchar(30),

-> email varchar(30),

-> id int(30),

-> cust\_id int(30),

-> brnum int(30),

-> primary key(id),

-> constraint ban\_br foreign key(brnum) references branch(brnum),

-> constraint ban\_cus foreign key(cust\_id) references customer(cust\_id));

Query OK, 0 rows affected (0.13 sec)

**VALUES INSERTION:**

**LOAN:**

mysql> insert into loan values(1,20000);

Query OK, 1 row affected (0.06 sec)

mysql> insert into loan values(4,20000),(2,3000);

Query OK, 2 rows affected (0.06 sec)

Records: 2 Duplicates: 0 Warnings: 0

**CREDIT:**

mysql> insert into credit values(30-11,21,1234);

Query OK, 1 row affected (0.08 sec)

mysql> insert into credit values(30-11,21,4569),(21-08,30,4567);

Query OK, 2 rows affected (0.04 sec)

Records: 2 Duplicates: 0 Warnings: 0

**ACCOUNT:**

mysql> insert into account values(10000,1,"mastercard"),(1,2,"visa");

Query OK, 2 rows affected (0.06 sec)

Records: 2 Duplicates: 0 Warnings: 0

**CUSTOMER:**

mysql> insert into customer values("Yashu.B",83,"Temple road","Adoni",1,1,1234);

Query OK, 1 row affected (0.29 sec)

mysql> insert into customer values("srivalli",26,"Temple2 road","Vij",1,1,1234);

Query OK, 1 row affected (0.06 sec)

**BRANCH:**

mysql> insert into branch values("Klu","Vij",9,1),("Sas","Vij",10,2);

Query OK, 2 rows affected (0.07 sec)

Records: 2 Duplicates: 0 Warnings: 0

**BANKER:**

mysql> insert into banker values("Raju","Raju@gmail.com",12,83,9),

("Ramu","Ramu@

yahoo.com",13,26,10);

Query OK, 2 rows affected (0.07 sec)

Records: 2 Duplicates: 0 Warnings: 0

**SOME QUERIES:**

1)Display customername,creditcard number whose limit is 2000

Select c.cust\_name,cr.crnum

From customer c,credit cr

Where cr.lim='30';

2)Display customer name and balance if balance >10000

Select c.cust\_name,a.balance

From customer c

where a.balance>'10000'

3)To print all the customer names and their id's in descending order

Select c.cust\_name,c.cust\_id

from customer c

order by desc c.cust\_id;

4)To update customer name as aishu where customer id =26

update customer set name="aishu"

where customer.cust\_id=26;

5)To print the customer name starting with "y"

Select c.cust\_

from customer c

where c.cust\_name like "y%";

**CONCLUSION:**

-> Provides the searching facilities based on various factors such as Arts, Cash, Bill, Stock.

-> Art gallery Management System also manage the customer details such as bill details, Stock details, Arts, etc..

-> It tracks all the information of Payment, Customer, Bill, etc..

-> Editing, adding and updating records is improved which results in proper resource management of Arts data.

-> It provides integration of all records of stock.

Practical Training is a very important part of the curriculum as it strengthens the concepts and enhances knowledge about the practical implementation of all the theory concepts. It helps managing the system very efficiently and conveniently. Initially we have identified the different requirements of our project. Then we divided our module and identified the different tasks of each module. Then we have drawn the ER diagram along with the relationships associated. We identified the tables from ER- diagrams and found the different attributes associated with each table. We created the tables with appropriate integrity constraints for each and every table. We inserted records into all the tables created .Then we wrote different queries to retrieve the data from database to satisfy the different tasks .Thus we accomplished every task given in each and every module.