DANK the BANK

Central Online Banking System

DBMS Project Report
MidEvaluation

Group 41

Divyam - 2020058

Aamleen - 2020002

Nishaant - 2020091

Kushagra - 2020075

Scope:

With the exponential increase in population, the traffic on Banking portals is tremendous which further results in managing a vast amount of Data of customers and that too while taking the topmost care of security. This traffic and data is ever increasing and here we present our Banking portal - DANK the Bank (DTB) that effectively uses the concepts of RDBMS to provide a secure, efficient and central banking system to its customers with a user-friendly, interactive interface as a cherry on the top.

Users will be provided with 2 types of accounts - Savings and Current. These accounts will be equipped with multiple user-friendly services online. All the data will be centrally stored and managed using which application processing time for any service will be reduced significantly. The portal is also integrated with features that caters to Employees, which can make it easy for them to manage and for the branches to keep their progress track. Traditionally one of the major hassles faced is while applying and approving loans, as it needs a lot of scrutiny. Through our portal, users can easily apply for loans and the officials on the other end will be provided with all the required data from our Database so that they can reduce the approval time significantly. Once the loan is approved, users can easily pay-back & track their loans.

One of the most important features will be to make security our top-most priority. We will achieve this by making our DataBase completely secure and "smart" by placing proper checks and using DBMS Concepts. Transactions can be done from User-Bank, User-User in various modes and these will be updated on a real-time basis with proper confirmations.

Thus, using the concepts of DBMS, we aim to deliver a one-stop solution for Online Banking System with Central DataBase.

Stakeholders:

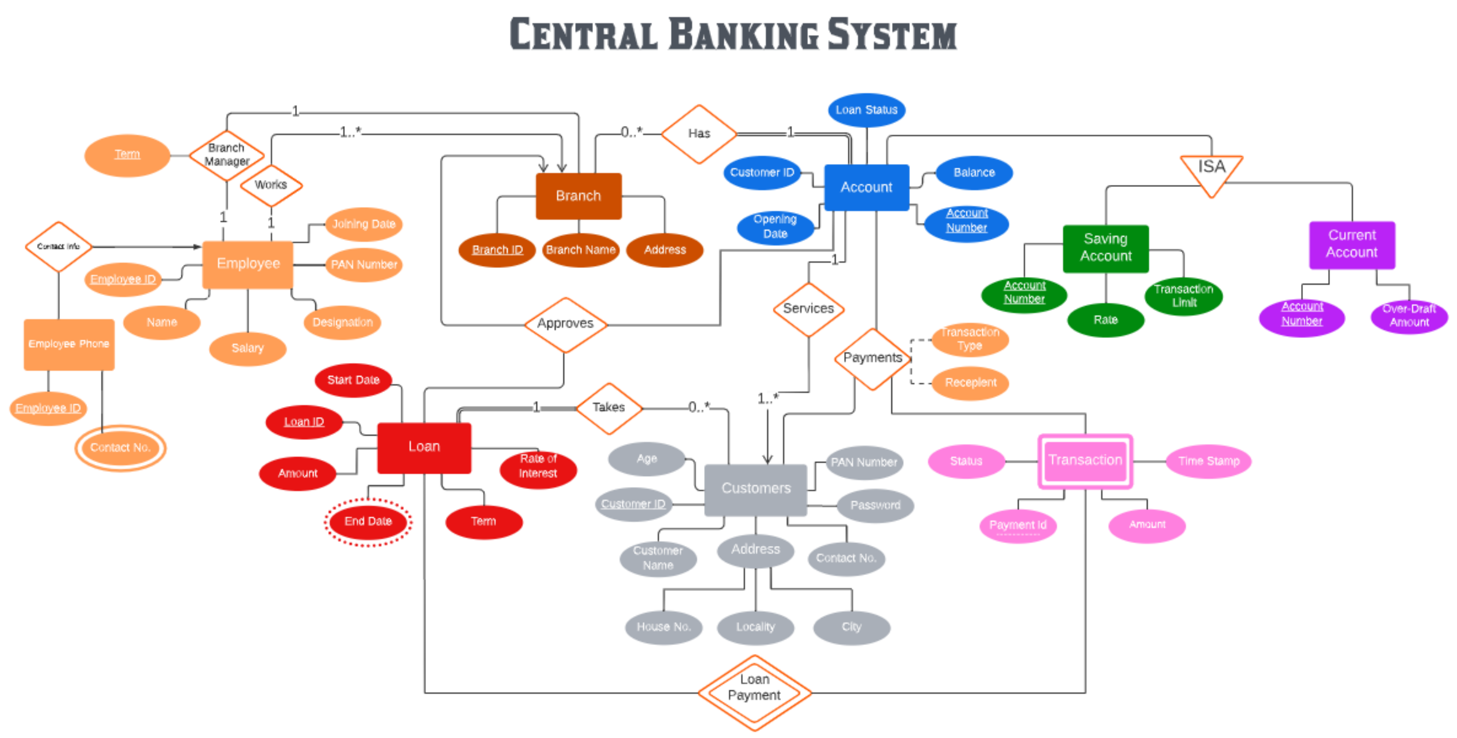
1. Branches Of Banks
    1. Employees Of Banks
    2. Loan Department
2. Users/Customers

Project Description/Aims:

An Online Central Banking System portal that integrates users and employees of Bank to effectively enhance the Banking services.

1) Banks can have multiple Branches. Storing information of each Branch in the Database
2) Each Branch will have its Employees and Manager. Keeping track of Branch Managers and the Employees under them. Uniquely connect each branch with its employees and manager.
3) Storing Data of Customers/Users and linking them to their accounts.
4) Each Branch will have 2 types of Accounts – Savings & Current. Storing various information related to accounts. Linking Accounts to Branch using IFSC Code. Also liking accounts to users to provide various services to them.
5) Keeping track of transactions made by users, like their type, status, amount, confirmation, etc.
6) Making successful transactions between User & User and also between User & Banks.
7) Providing Loan opportunities to customers where they can easily apply
8) Tracking the loan status before and after its approval stage. Highlighting significant defaulters & taking actions accordingly.
9) Helping users to pay back the loan successfully.
10) Providing some additional bonus services to the users (Optional)

ER DIAGRAM


CENTRAL BANKING SYSTEM
Branch Manager
Term
Works
Contact Info
Employee
Joining Date
PAN Number
Employee ID
Name
Salary
Designation
Employee Phone
Employee ID
Contact No.
Branch
Branch ID
Branch Name
Address
Has
Account
Loan Status
Customer ID
Opening Date
Balance
Account Number
ISA
Saving Account
Account Number
Rate
Transaction Limit
Current Account
Account Number
Over-Draft Amount
Approves
Services
Payments
Transaction Type
Recepient
Loan
Start Date
Loan ID
Amount
End Date
Term
Rate of Interest
Takes
Customers
Age
Customer ID
Customer Name
Address
House No.
Locality
City
PAN Number
Password
Contact No.
Transaction
Status
Time Stamp
Payment Id
Amount
Loan Payment
1
1..*
0..*
1
1
1
1
1..*
1
0..*

Identification Of Weak Entity

Transaction: Transaction is a weak entity because it depends on the Account and Customer Entity. If there is no account, then there is no point in having any transaction entity itself. Also, the transactions that are made, are done after confirmation from Customer and Account Balance and limits. The Key of Transaction is also dependent on the customer paying.

Identification of Ternary Relationship

Payments: Payments is a ternary relationship because it involves 3 entities. This relationship comes into effect when a customer makes some transactions, which need to be validated by the Account status. Thus 3 entities come into play - Customer, Account & Transaction.

Approves: Approves is a ternary relationship that comes into effect when a customer applies for a loan, which is approved/denied by the branch. Thus 3 entities are involved in this - Customer who applies, Branch which approves/denies, and the Loan which stores all the necessary information and tracks the status before and after loan.

DATABASE SCHEMA

Accounts:

AccountNo	VarChar(100) NotNULL
OpeningDate	Date NotNULL
LoanStatus	VarChar(100)
Balance	Double
Customer_ID	VarChar(100) NotNULL

Primary Key: AccountNo
Foreign Key: Customer_ID

Employee:

Employee_ID	VARCHAR(100) NOT NULL
Name	VARCHAR NOT NULL
Salary	Int NOT NULL
Designation	VARCHAR(100) NOT NULL
PAN No.	VARCHAR(100)
Joining Date	DATE NOT NULL
Contact No.	VarChar(10)

	Not Null

Primary Key: Employee_ID

Customers:

Customer_ID	VarChar(100) NotNULL
Name	VarChar(100) NotNULL
Age	Int NotNULL
HouseNo	VarChar(20) NotNULL
Locality	VarChar(100) NotNULL
City	VarChar(50) NotNULL
ContactNo	Bigint NotNULL
PAN	VarChar(30) NotNULL
Password	VarChar(100) NotNULL

Primary Key: Customer_ID

Branch:

Branch_ID	Bigint NotNULL

Branch_Name	VarChar(100) NotNULL
Address	VarChar(100) NotNULL

Primary Key: Branch_ID

Transactions:

Payment_ID	VarChar(100) NotNULL
Amount	Double
Date	DateTime NotNULL
Status	VarChar(100) NotNULL

Primary Key: Payment_ID

Loans:

StartDate	Date NotNULL
Loan_ID	VarChar(100) NotNULL
Amount	Double NotNULL
InterestRate	Double
Term	Int NotNULL

EndDate	Date

Primary Key: Loan_ID

Branch Managers:

Employee_ID	VARCHAR(100) NOT NULL
Branch_ID	BigInt NOT NULL
Name	VARCHAR(100) NOT NULL
Designation	VARCHAR(100) NOT NULL
Term	Int NotNULL
Joining_Date	Date NotNULL

Remaining attributes are same as Employee

Primary Key: Employee_ID, Branch ID

Customer_Account_Transactions:

Customer_ID	VarChar(100) NotNULL
Payment_ID	VarChar(100) NotNULL
AccountNo	VarChar(100) NotNULL
Amount	Double
Transaction_Type	VarChar(100) NotNULL
Recipient	VarChar(100)

	NotNULL

Primary Key: Payment_ID, Customer_ID

RELATIONAL SCHEMA

Branches (Branch ID, Branch Name, Address)
Employees (Employee ID, Name, PAN Number, Salary, Designation, Joining Date, {Contact No.})

Works (Employee ID, Branch ID)
Branch_Managers (Employee ID, Name, Branch ID, Designation, Term)
Branch_Account (Branch ID, Account No.)

Accounts (Account No., Customer ID, Opening Date, Balance, Loan Status)
Savings_Accounts (Account No., Rate, Transaction Limit)
Current_Accounts (Account No., OverDraft Amount)

Customers (Customer ID, Name, PAN Number, Age, Password,Customer Name,Address(House No.,Locality,City))

Loans(Loan Id,Term, Rate of Interest, End Date,Start Date)
Branch_Loan_Account(Branch ID, Loan ID, Customer ID)
Loan_Transactions(Loan ID, Payment ID)

Transactions(Payment ID, Status, Amount, Date)

Customer_Account(Account No., Customer ID, Type)
Customer_Account_Transactions(Payment ID, Customer ID, Account No,Type, Recipient)

SQL QUERIES

1. List all loan defaulters name, account number, loan_id
2. List account numbers of all employees
3. List Loan_ids that were successfully paid by a customer
4. List customers who are older than 60 and have taken a loan with their rate of interest
5. List all the employees working under a branch manager
6. List status of every type of payment from a customer.
7. List status of transactions of payment > 10000
8. List the customer's number, customer's Name, branch id and loan amount for people who have taken loans.
9. List account number, and net balance across accounts of a customer
10. List the total number of withdrawals and total number of deposits being done by customer
11. List name, customerID for failed transactions
12. List the managers with salary > 50000 who joined before 2010
13. List the No of customers with savings account and current account
14. List accounts with loan status pending and roi >= 6%
15. List Customers who have both savings and current accounts\