Project Final Report

Abhishek Ningala

Database Management System

CS 5200

BANK DATABASE MANAGEMENT SYSTEM

About the Project:

Keeping money safe has always been a concern of people and hence they tend to deposit it at a bank. One can choose to open an account at a bank and deposit the money. A person can choose to have multiple accounts at different banks.

In this project, we propose a database model which consists of different banking entities. For customers to open bank accounts, all the customers must be registered in the Customer table, and all the bank accounts must be registered in the Account table. The deposits information on each transaction associated with each bank account is stored in the Transaction table. Loan and claims information can also be stored in their respective tables.

Description of the Application:

To begin with, a bank entity must be registered in the application. Any number of banking entities can be registered, but there must be at least one registered to use the main functions.

Secondly, the customer must also be registered with name and contact information. Only then they can proceed to open a bank account in the chosen bank. Once the bank has been created, the customer can obtain a loan with specifying the time in years, loan amount and interest rate.

The application also allows you to display the list of banks, accounts, clients, loans, and transactions.

The main functions of the database are as follows:

1. Register a new Branch
2. Register a new Customer
3. Register a new Account
4. Print the list of branches
5. Print the list of customers
6. Print the list of accounts
7. Delete a particular customer
8. Perform a new transaction
9. List all the transactions
10. Create a Loan
11. Create a claim
12. List all the claims
13. Update a particular Customer's Details
14. Transfer money from one account to another account

Tools and Technologies Used:

In this project, Oracle's MySQL is the backend database system. MySQL workbench version 8.022 is used. A simple terminal application coded in python is used as frontend. PyCharm IDE version 2020.2 is used.

Oracle's MySQL Downloads page: MySQL: MySQL Community Downloads

PyCharm IDE Downloads page: Download PyCharm: Python IDE for Professional Developers by JetBrains

Justification for SQL Database (MySQL):

- We have chosen a SQL Database (MySQL) for our bank database system as it is fast and easy – can send and receive responses from a database as quickly as possible.
- Uses a single DDL language for different roles – Developer, User, DBA.
- Uses a standard language for different Data Management Systems.
- Follows ACID properties (atomicity, consistency, isolation, durability) which guarantees stability and security of every transaction.

The following packages are needed:

- pymysql
- random
- tabulate
- datetime

The following functions and methods were used in code:

- input()
- connect()
- execute()
- fetchall()
- commit()
- isnumeric()
- int()
- float()
- str()
- len()
- lower()
- range()
- randint()
- keys()

Usage:

There are 2 python files provided: *1_Bank_DB_init.py* and *2_Bank_DB_main.py*.

The first file ***Bank_DB_init.py*** should be run first. This file is used to create the database and all the bank entities required for the bank database system. This file also inserts some sample data to start.

One important package to install is pymysql. This package provides an interface for connecting to a MySQL database from Python3.

Credentials to the database are accepted from the user. Please provide *your* username and password to connect to the database on your system.

The second file ***Bank_DB_main.py*** provides the Menu to interact with the database and should be run after running the first file.

Few important packages to install before running the second file are:

Random – Used to generate a random integer used for account numbers in the Account table

Tabulate – Used to print tabular data in nicely formatted tables

Datetime – Used for working with date and time in python

Credentials to the database are accepted from the user. Please provide *your* username and password to connect to the database on your system.

CRUD:

The following CRUD operations could be demonstrated:

Create Operation: Registering a new Branch, Customer, Account; Creating a Loan, Claim, Performing a new transaction

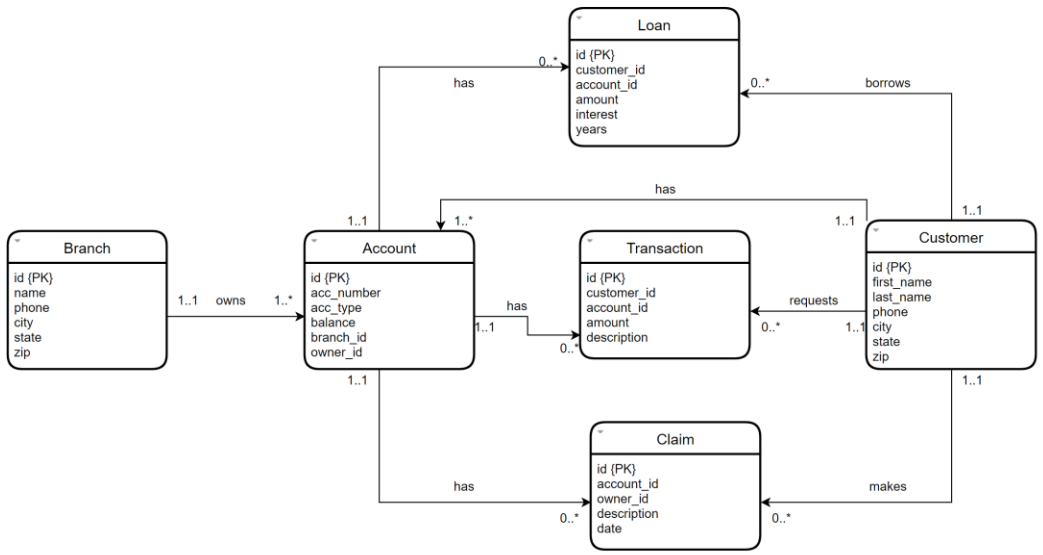
Read Operation: List the Branches, Customers, Accounts, Transactions, Claims

Update Operation: Update a Customers Information, Transfer money from one account to another

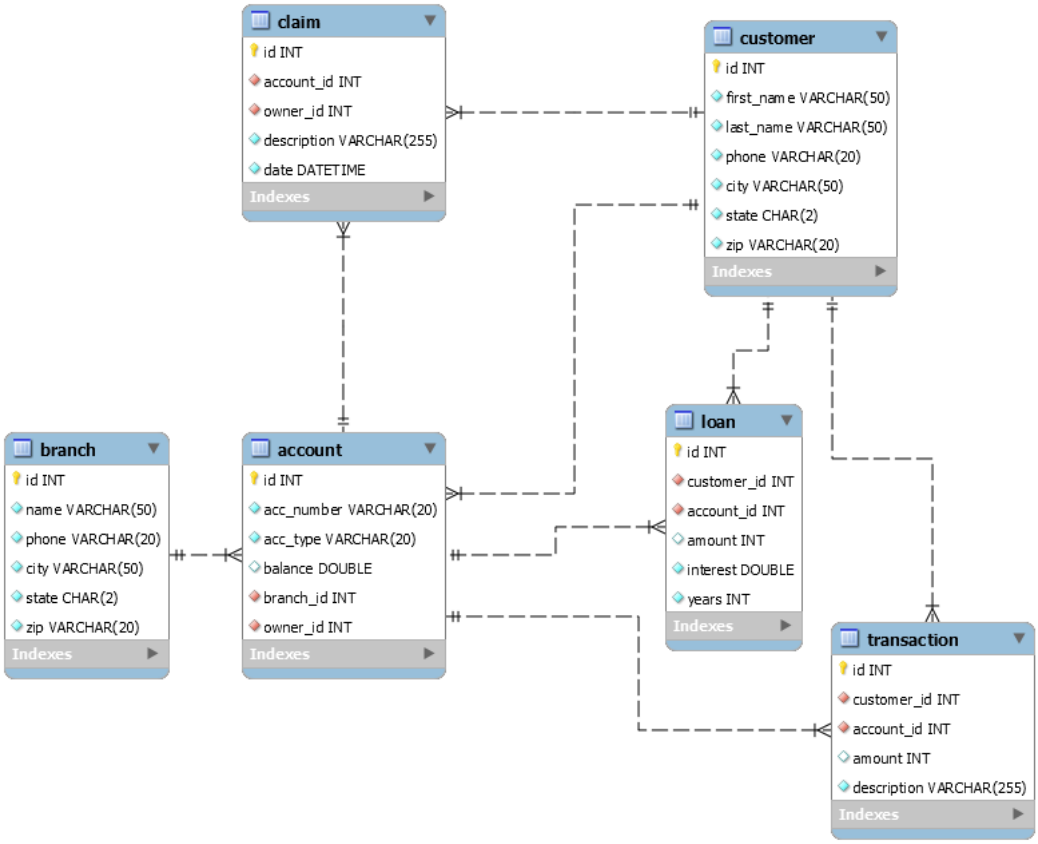
Delete Operation: Delete a user/customer

Each of these operations could be performed by choosing the appropriate options from the Interactive menu. If option *15* is chosen, the database gracefully exits.

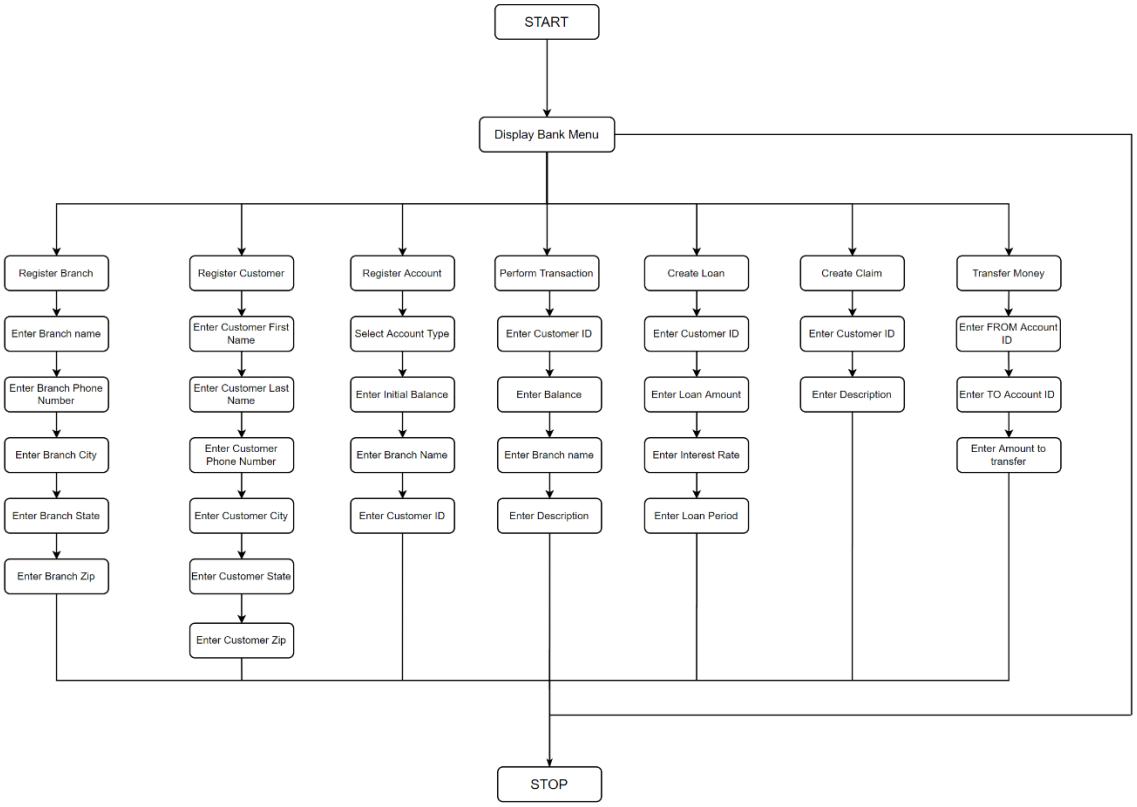
Conceptual Diagram:



Logical Diagram:



Flowchart:



User Interaction:

The user starts the program

A bank menu is displayed with all the options

User registers a new Bank Branch

User registers a new Customer

User registers a new Bank Account

User can now select any option – registration of a new branch, another customer, another bank account, etc.

User can also create a loan, do a transaction, fill a claim, etc.

User can list branches, customers, accounts, loans, transactions, or claims.

User can transfer money from one account to another account

User selects the “Exit” option.

The program ends execution.

Lessons Learned:

- Learned how to provide an interface to MySQL database and interactively control via the bank menu.
- I got a high level understanding how databases work with the frontend terminal application.
- I have successfully managed to complete the project in a span of 17 days.
- All code is working properly

Future work:

This is a high-level project for a bank database management system. More entities could be added to create a complicated bank database which can provide more functionalities.

References:

"Chapter 2 Installation." *MySQL*, dev.mysql.com/doc/workbench/en/wb-installing.html.

"Get Started - Help: PyCharm." *PyCharm Help*, www.jetbrains.com/help/pycharm/quick-start-guide.html.

"Welcome to Python.org." *Python.org*, www.python.org/doc/.

Dickson, Craig. "How to Create and Manipulate SQL Databases with Python." *FreeCodeCamp.org*, FreeCodeCamp.org, 31 Aug. 2020, www.freecodecamp.org/news/connect-python-with-sql/.

"Python Database Connection: Access Python MySQL Database." *Edureka*, 18 May 2020, www.edureka.co/blog/python-database-connection/.