

#### ANDHRA UNIVERSITY COLLEGE OF ENGNEERING

## PROJECT REPORT ON

**“PERSONAL BANK ACCOUNT”**

**DEPARTMENT OF INFORMATION TECHNOLOGY AND**

**COMPUTER APPLICATIONS**

**ANDHRA UNIVERSITY COLLEGE OF ENGINEERING**

**ANDHRA PRADESH - 530003**

###### SUBMITTED BY:

**RUTTALA DIVYA VANI 321107311062**

**DR. PRIYANKA KUMARI BHANSALI**

**SUBMITTED TO:**

**CONTENTS**

**CHAPTER-1: INTRODUCTION**…………………………………………………...1

**CHAPTER-2: TERMINOLOGY USED IN THIS PROJECT**………....................2

**CHAPTER-3: DATA REQUIREMENTS**………………………………………….3

**CHAPTER-4: ENTITY RELATIONSHIP DIAGRAM**…………………………...4

**CHAPTER-5: SCHEMA DIAGRAM**……………………………………………....5

**CHAPTER-6: CREATING DATABASE USING MYSQL**……………………….6

**CHAPTER-7: RESULTS**…………………………………………………………..10

**CHAPTER-8: CONCLUSION**……………………………………………………..20

## 1.INTRODUCTION

**Personal Bank Account:** An individual account means a bank account that is only used for non-business activities. The distinction is made between personal accounts and other accounts in banking and accounting because different account types have different implications and treatments. In a business account there are probably many users that are extracting from a large pool of money for the purpose of running a business.

For a personal account, there should only be one person depositing and withdrawing money and therefore security measures will be in place to make sure the right person is accessing the funds. Personal accounts are often kept completely separate from business and joint accounts because they are completely in the interest of one user rather than many. Functioning of a Bank is among the more complicated of corporate operations. Since Banking involves dealing directly with money it becomes difficult. So, this application was made to make all that process easy.

This application provides you convenience and basic banking transactions such as transferring funds between accounts can easily be done any time of the day or night, seven days a week.is fast and efficient. Funds can be transferred between accounts almost instantly. You can open and close a number of different accounts online.

RULES THAT GOVERN THIS PROJECT:

1. The account numbers cannot be null i.e., they cannot take null values.
2. All the customers of the bank have a unique account number.
3. The customers must have a minimum account balance of Rs.1000.

**OBJECTIVE**

The Personal Bank Account is a Database Projectwhich is more efficient, fast, reliable, user friendly. Convenience is a major advantage. Over and above the proposed system does not have any possibility of data loss during processing. This Personal Bank Account project will serve as a useful approach to data base dialogue box to deposit and withdraw the money as well as the balance enquiry for the person. It serves as a helpful approach for the users. It provides easy way of the deposit and withdrawal of the money. It reduces the time taken by the user to save the money. Thus, the project is the user-friendly approach.

### 

### 2.TERMINOLOGY USED IN THIS PROJECT

1. **INTEGER:** One optional sign character (+ or -) followed by at least one digit from 0-9. Leading and trailing blanks are ignored. No other character is allowed.
2. **VARCHAR:** A VARIABLE length string (can contain letters, numbers, and special characters). The size parameter specifies the maximum string length in characters – can be from 0 to 65535. It is used to store alpha numeric values. By default, SQL server sets the size to 50 characters range.
3. **DATE:** Store a date only. No parameters are required when declaring a DATE data type. It stores dates from January 1, 0001 to December 31, 9999.
4. **Mysql Connector:** This database application is made of Python and Mysql. Python can be used in database applications. MySQL Connector enables Python programs to access MySQL databases.

**3. DATA REQUIREMENTS**

**ENTITIES:**

* CUSTOMERS
* BANK
* AMOUNTS

**ATTRIBUTES:**

* CUSTOMERS **--🡪** Customer Name

Account Number

Date of Birth

Address

* BANK --🡪 Bank Name

IFSC Code

Bank ID

Bank Address

* AMOUNTS --🡪 Account number

Customer Name

Total Amount

**RELATIONSHIPS – CARDINALITY**

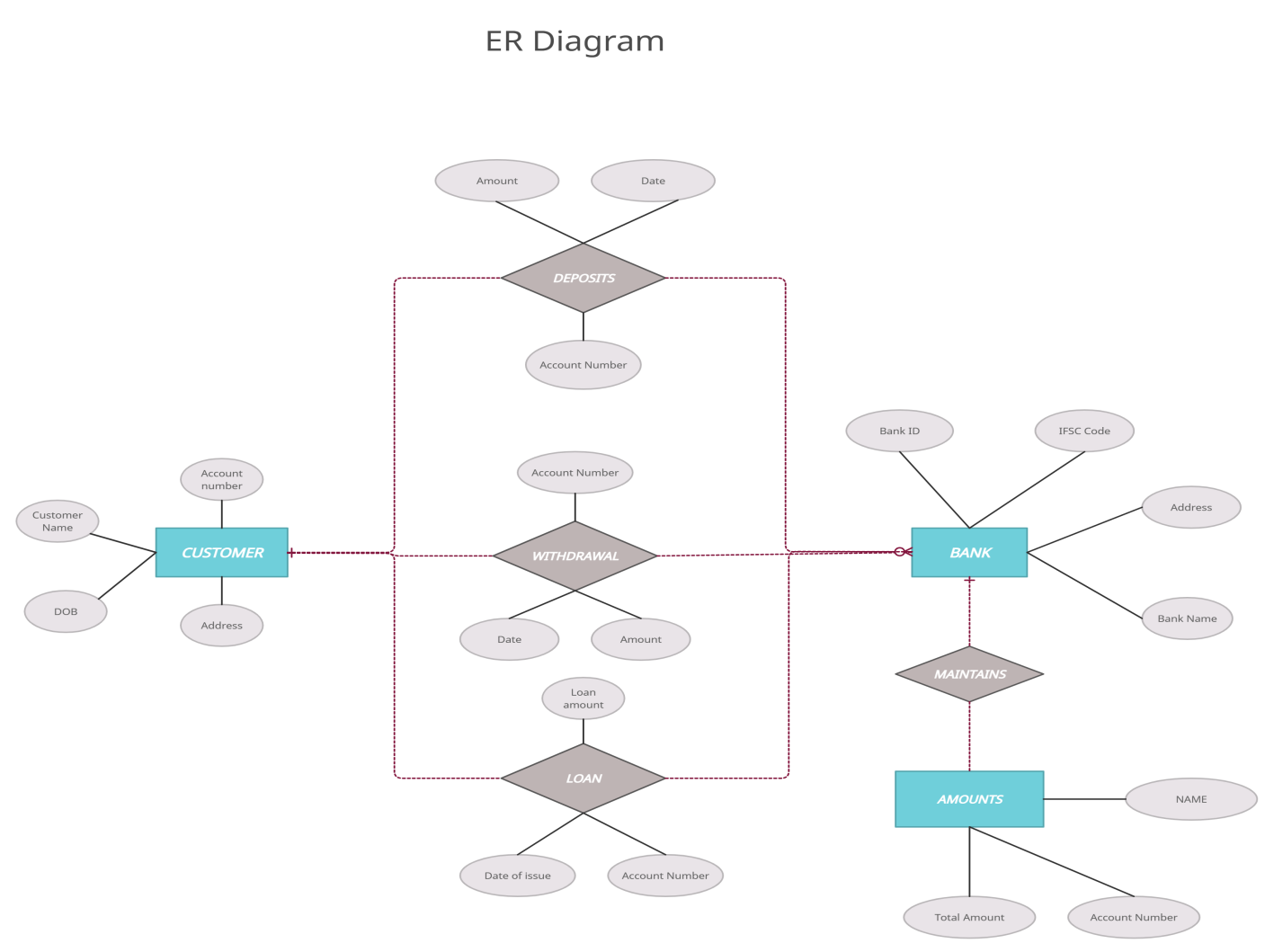
* BANK has CUSTOMERS
* BANK maintains AMOUNT
* BANK offers LOANS
* CUSTOMERS can DEPOSIT
* CUSTOMERS can WITHDRAWAL

**4.ENTITY RELATIONSHIP -DIAGRAM**

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships.ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

Prime reasons for using the ER Diagram:

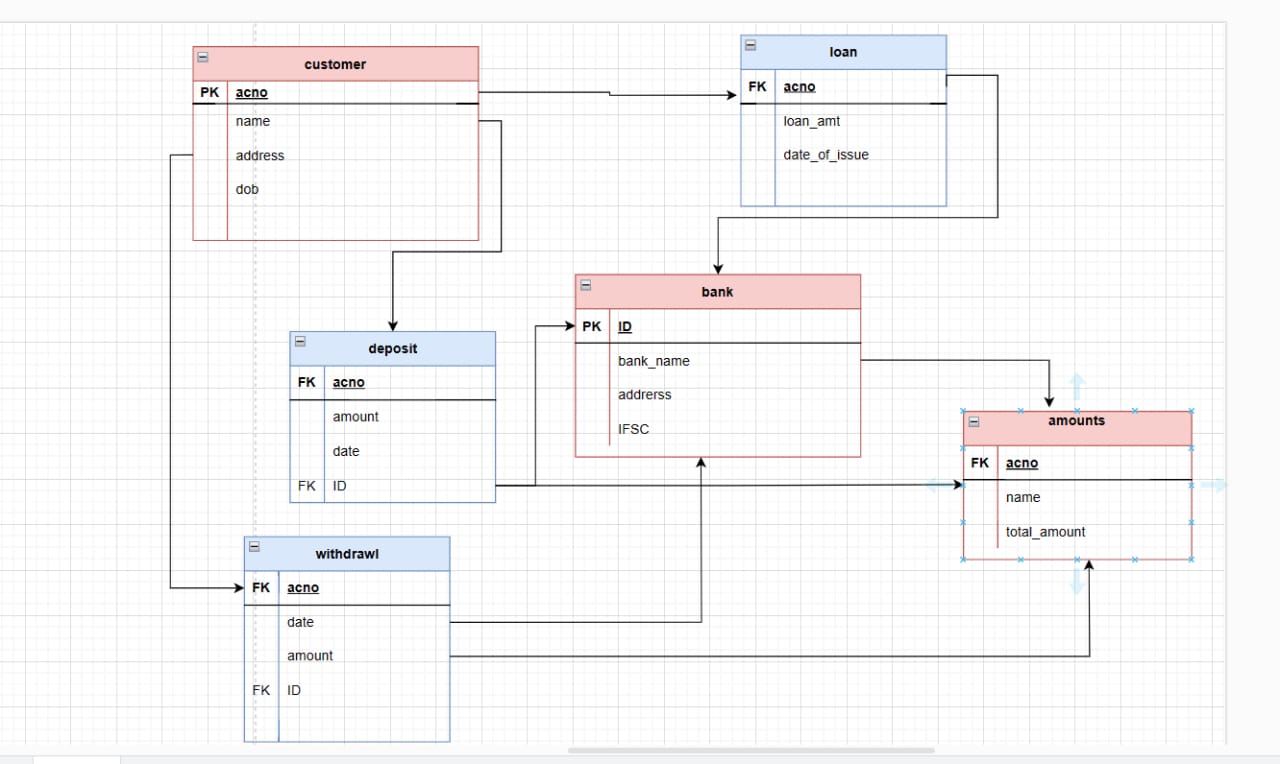
* Helps you to define terms related to entity relationship modeling
* Provide a preview of how all your tables should connect, what fields are going to be on each table
* Helps to describe entities, attributes, relationships
* ER diagrams are translatable into relational tables which allows you to build databases quickly
* ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications
* The database designer gains a better understanding of the information to be contained in the database with the help of ERP diagram
* ERD Diagram allows you to communicate with the logical structure of the database to users.



**5.SCHEMA DIAGRAM**

A database schema is considered the “blueprint” of a database which describes how the data may relate to other tables or other data models. However, the schema does not actually contain data. A database schema defines how data is organized within a relational database; this is inclusive of logical constraints such as, table names, fields, data types, and the relationships between these entities. Schema commonly use visual representations to communicate the architecture of the database, becoming the foundation for an organization’s data management discipline. This process of database schema design is also known as [data modeling](https://www.ibm.com/topics/data-modeling).

These data models serve a variety of roles, such as database users, database administrators, and programmers. For example, it can help database administrators manage normalization processes to avoid data duplication. Alternatively, it can enable analysts to navigate these data structures to conduct reporting or other valuable business analyses. These diagrams act as valuable documentation within the database management system (DBMS), ensuring alignment across various stakeholders.



**6.CREATING DATABASE USING MYSQL**

mysql>create database our\_bank;

mysql>use our\_bank

database changed

mysql>CREATE TABLE Bank(

Bank\_id varchar(20),

Bank\_name varchar(50),

IFSC\_Code varchar(15),

Address varchar(100)

);

Query OK,0 rows affected(1.44sec)

mysql>INSERT INTO Bank values("572940","OURS BANK OF ANDHRA UNIVERSITY","OUR25830015","AU COLLEGE OF ENGINEERING,MADDILAPALEM,VISAKHAPATNAM,530003");

Query OK, 1 row affected (0.18 sec)

mysql>CREATE TABLE customers (

name varchar(20),

acno varchar(20),

Dob varchar(25),

ph int,

Address varchar(55),

date DATE

);

Query OK,0 rows affected(1.44sec)

mysql>CREATE TABLE amounts (

name varchar(20),

acno varchar(20),

balance int);

Query OK,0 rows affected(1.44sec)

mysql>CREATE TABLE credited(

acno varchar(20),

balance int,

date DATE

);

Query OK,0 rows affected(1.44sec)

mysql>CREATE TABLE debited(

acno varchar(20),

balance int,

date DATE );

Query OK,0 rows affected(1.44sec)

mysql> CREATE TABLE loan (

acno varchar(20),

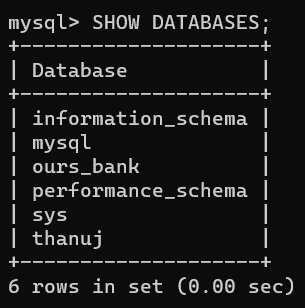
loan\_amt int,

issued\_date DATE

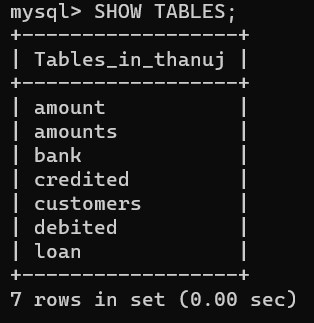
);

Query OK,0 rows affected(1.44sec)

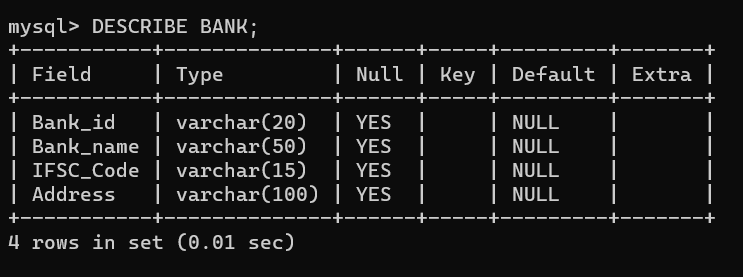
**Databases:**



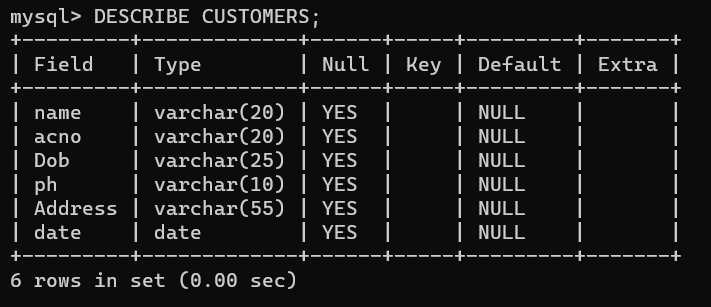
Tables in the database are:



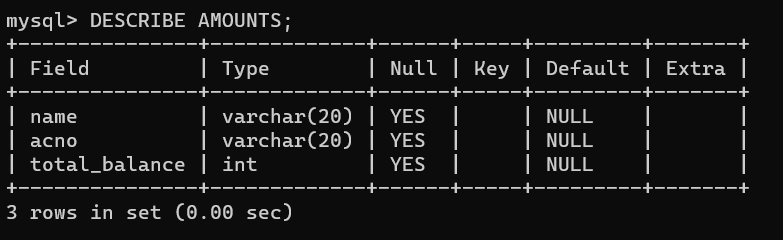
**DESCRIBE BANK**;



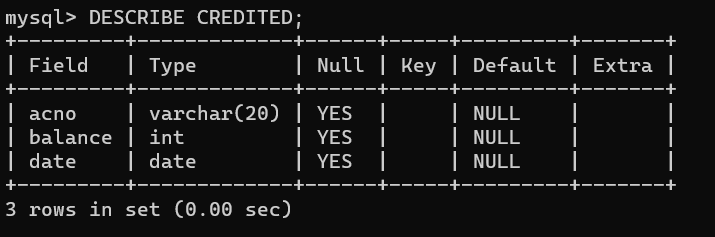
**DESCRIBE CUSTOMERS;**



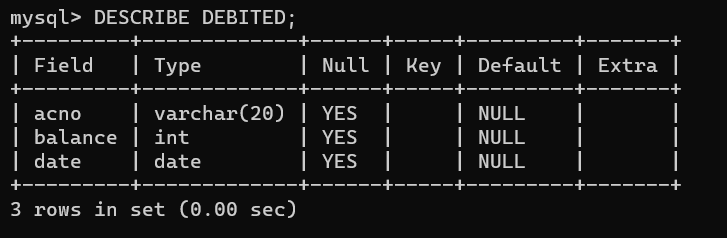
**DESCRIBE AMOUNTS;**



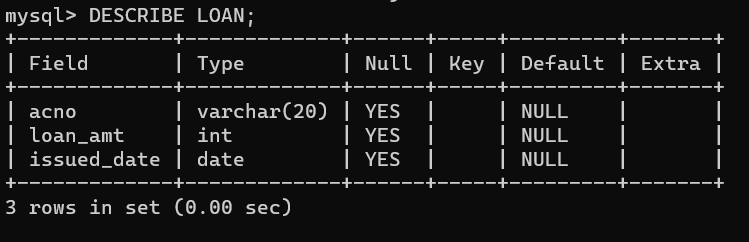
**DESCRIBE CREDITED;**



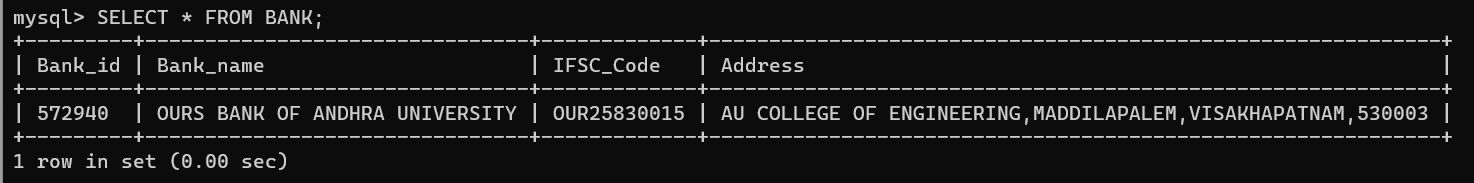
**DESCRIBE DEBITED;**



**DESCRIBE LOAN;**



INSERT INTO Bank values("572940","OURS BANK OF ANDHRA UNIVERSITY","OUR25830015","AU COLLEGE OF ENGINEERING,MADDILAPALEM,VISAKHAPATNAM,530003");



**7.RESULTS**

**TO ESTABLISH A CONNECTION TO THE DATABASE:**

import mysql.connector as a

import datetime as date

dates1=date.datetime.now()

con=a.connect(host='localhost',user='root',passwd='123456789',database='divya')

if con.is\_connected():

print("connection establishes to mysql")

def openAcc():

n=input("Enter your name : ")

ac=input("Enter account no : ")

db=input("Enter your D.O.B : ")

p=input("enter your ph number : ")

ad=input("Enter your address : ")

print("\n")

print(" ==========minimum opening balance 1000 =========”)

print("\n")

ob=int(input("Enter opening Balance : "))

data1=(n,ac,db,p,ad,dates1)

data2=(n,ac,ob)

sql1='insert into customers values(%s,%s,%s,%s,%s,%s)'

sql2='insert into amounts values(%s,%s,%s)'

c=con.cursor()

c.execute(sql1,data1)

c.execute(sql2,data2)

con.commit()

print("\n")

print("=======ACCOUNT CREATED SUCCESSFULLY ======”)

print("\n")

main()

def depoAmo():

am=int(input("Enter Amount : "))

ac=input("Enter Account No : ")

data1=(ac,am,dates1)

sql1='insert into credited values(%s,%s,%s)'

a="Select total\_balance from amounts where acno=%s"

data=(ac,)

c=con.cursor()

c.execute(sql1,data1)

c.execute(a,data)

myresults=c.fetchone()

tam=myresults[0]+am

sql="Update amounts set total\_balance =%s where acno =%s"

d=(tam,ac)

c.execute(sql,d)

con.commit()

print("\n")

print("====YOUR AMOUNT DEPOSITED SUCCESSFULLY ===”)

print("\n")

main()

def witham():

am=int(input("Enter Amount : "))

ac=input("Enter Account No : ")

data1=(ac,am,dates1)

sql1='insert into debited values(%s,%s,%s)'

a="Select total\_balance from amounts where acno=%s"

data=(ac,)

c=con.cursor()

c.execute(sql1,data1)

c.execute(a,data)

myresults=c.fetchone()

tam=myresults[0]-am

sql="Update amounts set total\_balance =%s where acno =%s"

d=(tam,ac)

c.execute(sql,d)

con.commit()

print("\n")

print("==WITHDRAWL AMOUNT SUCCESSFULLY ====”)

print("\n")

main()

def loan():

am=int(input("Enter loan Amount : "))

ac=input("Enter Account No : ")

data1=(ac,am,dates1)

sql1='insert into loan values(%s,%s,%s)'

c=con.cursor()

c.execute(sql1,data1)

con.commit()

print("\n")

print("====LOAN SANCTIONED SUCCESSFULLY====”)

print("\n")

main()

def balance():

ac=input("enter Account No : ")

a="Select total\_balance from amounts where acno=%s"

data=(ac,)

c=con.cursor()

c.execute(a,data)

myresults=c.fetchone()

print("balance for Account : ",ac,"is",myresults[0])

print("\n")

print("==============THANK YOU ==========”)

print("\n")

main()

def closeac():

ac=input("enter Account No : ")

sql1="delete from customers where acno=%s"

sql2="delete from amounts where acno=%s"

sql3="delete from loan where acno=%s"

sql4="delete from credited where acno=%s"

sql5="delete from debited where acno=%s"

data=(ac,)

c=con.cursor()

c.execute(sql1,data)

c.execute(sql2,data)

c.execute(sql3,data)

c.execute(sql4,data)

c.execute(sql5,data)

con.commit()

print("\n")

print("==YOUR ACCOUNT HAS BEEN REMOVED ==”)

print("\n")

main()

def main():

print("\n")

print("=========WELCOME TO OURS BANKING ========”)

print("\n")

print("=======HOW MAY I HELP YOU=====”)

print("\n")

print("""""

1.OPEN NEW ACCOUNT

2.DEPOSIT AMOUNT

3.WITHDRAW AMOUNT

4.APPLY FOR LOAN

5.BALANCE ENQUIRY

6.CLOSE AN ACCOUNT""""")

print("\n")

choice=input("enter task No : ")

if(choice=='1'):

openAcc()

elif(choice=='2'):

depoAmo()

elif(choice=='3'):

witham()

elif(choice=='4'):

loan()

elif(choice=='5'):

balance()

elif(choice=='6'):

closeac()

else:

print("\n")

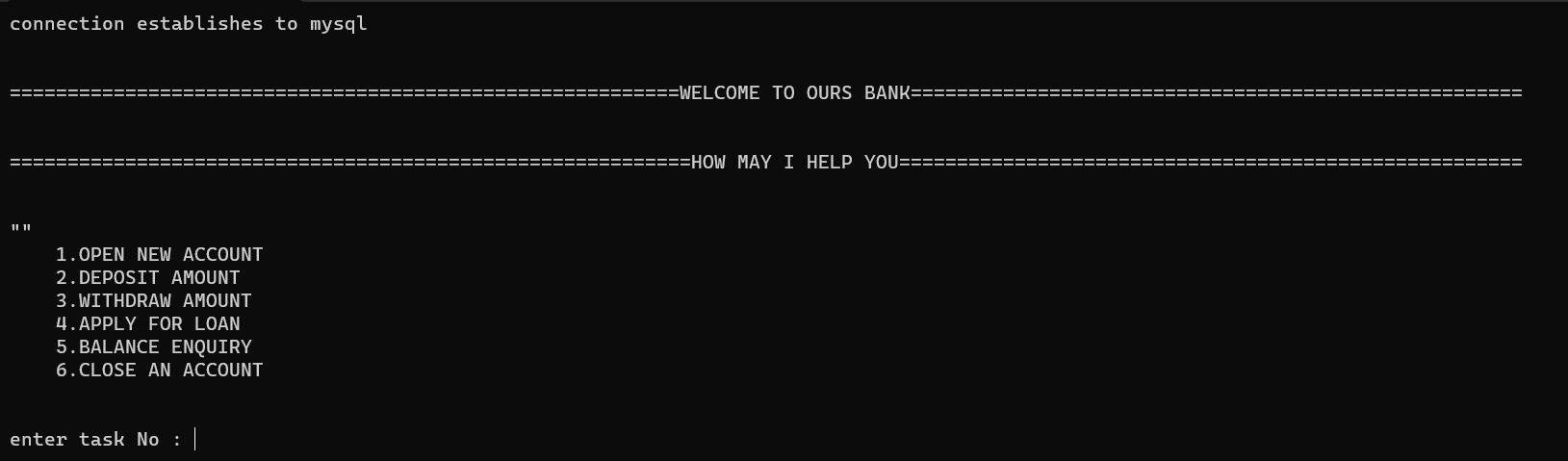
print("====ENTER CORRECT CHOICE====”)

print("\n")

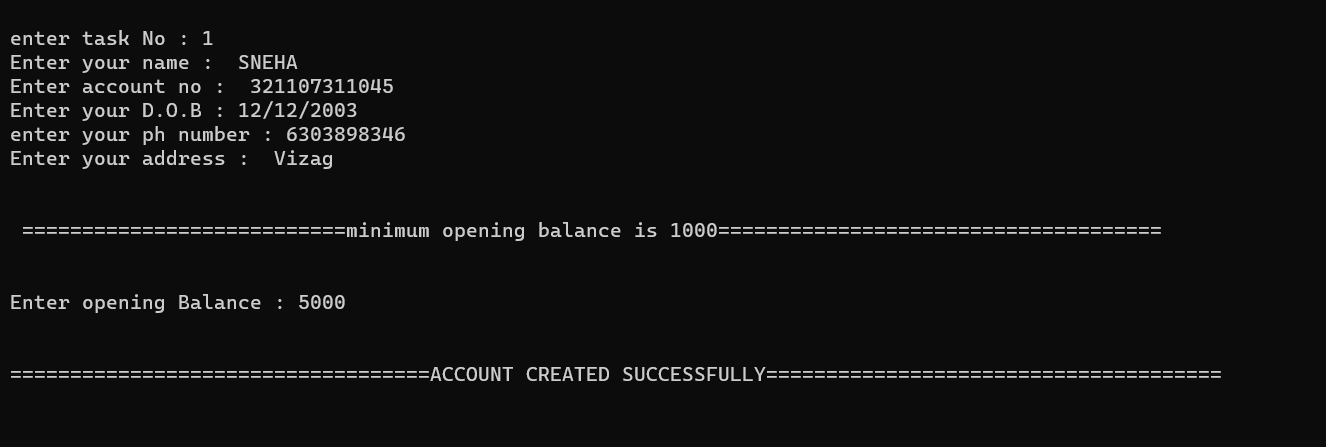
main()

main()

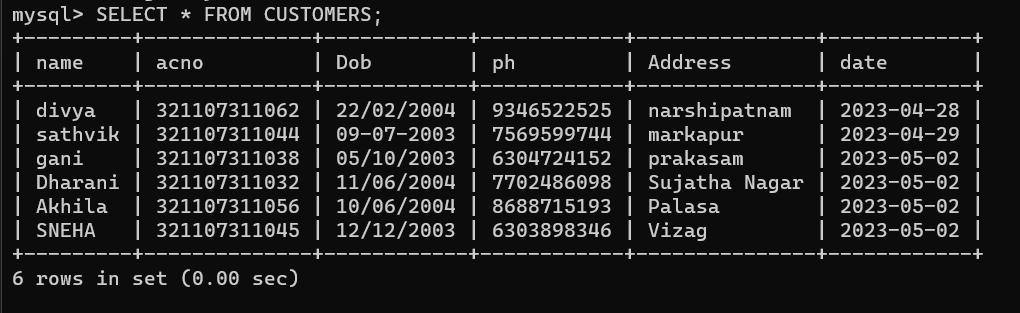
**MENU:**



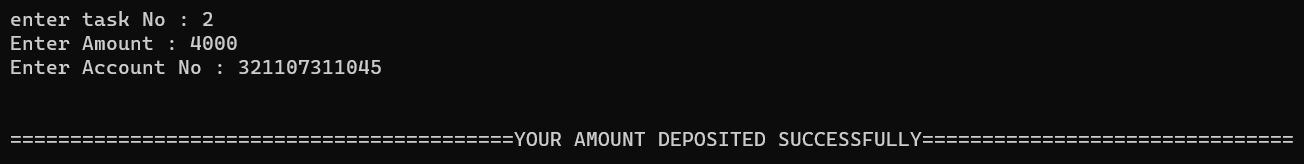
**TASK -1: OPENING AN ACCOUNT**



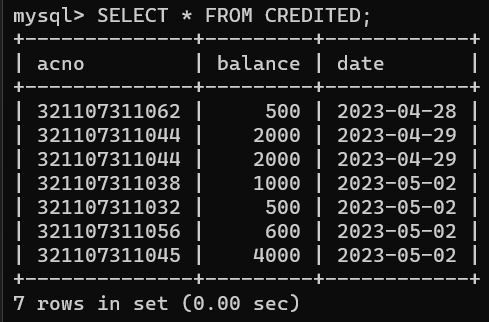
The data given was inserted into the “Customers” table as shown below:

****

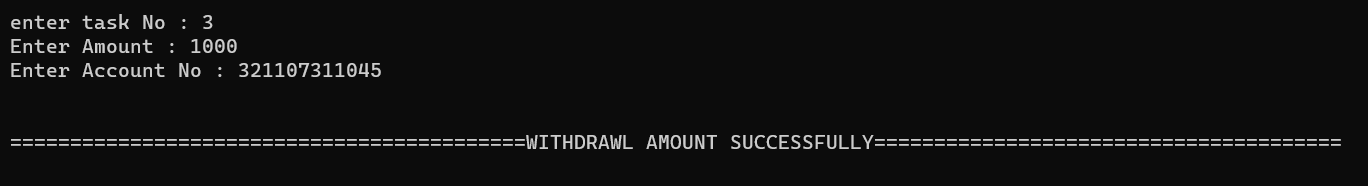
**TASK -2: DEPOSIT AMOUNT:**

****

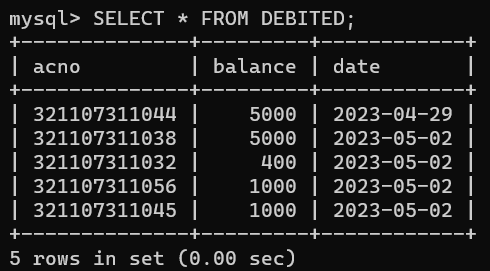
The deposited amount has been updated in the credited table, as shown in the below table:



**TASK -3: WITHDRAWAL AMOUNT:**

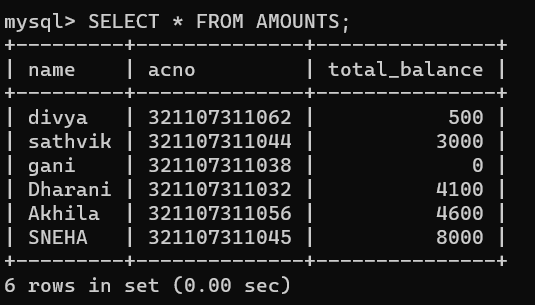


The withdrawal amount has been updated in the debited table, as shown in the below table:



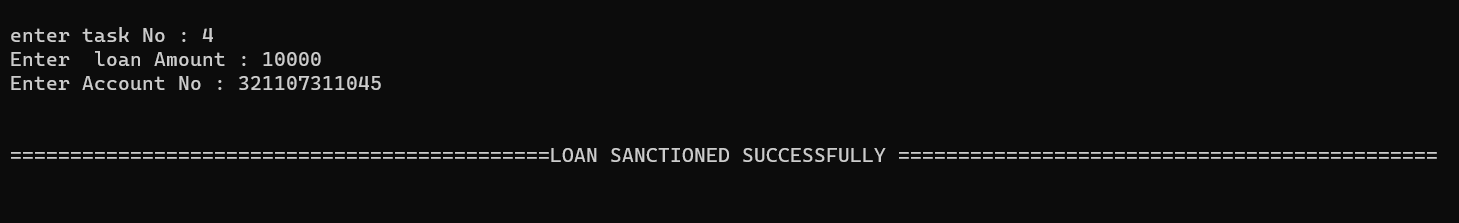
**AMOUNTS TABLE:**

After every deposit and withdrawal, the amount will be updated in the total\_balance field of the amounts table.

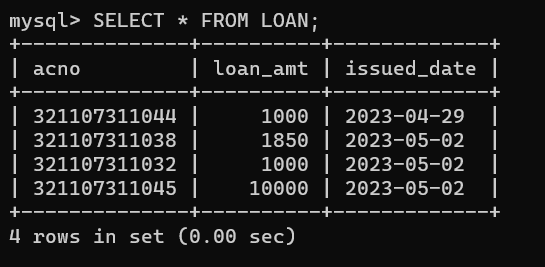


The opening account balance of the ac.no 321107311045 is 5000. After that we have deposited amount 4000 and withdraw amount 1000. Remaining amount is 8000 has been updated in the total\_balance field of amounts table.

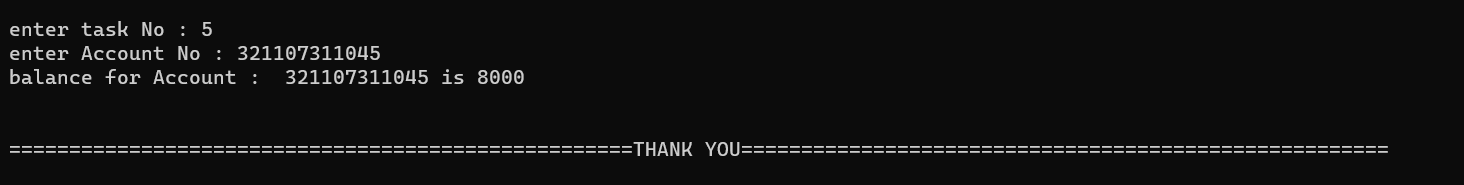
**TASK -4: APPLY FOR LOAN:**

****

The loan amount has been updated in the Loan table, as shown in the below table:



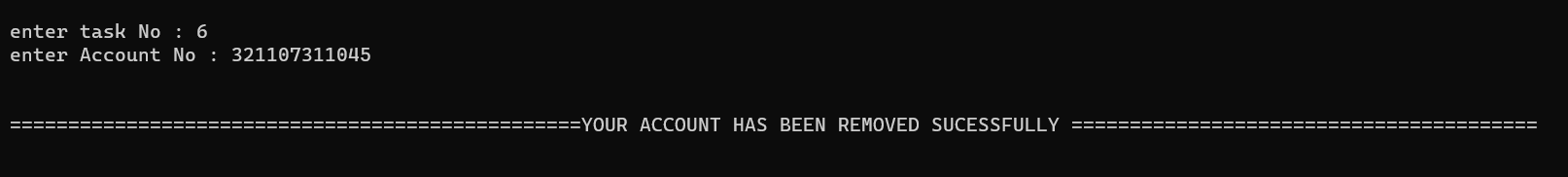
**TASK-5: BALANCE ENQUIRY**

****

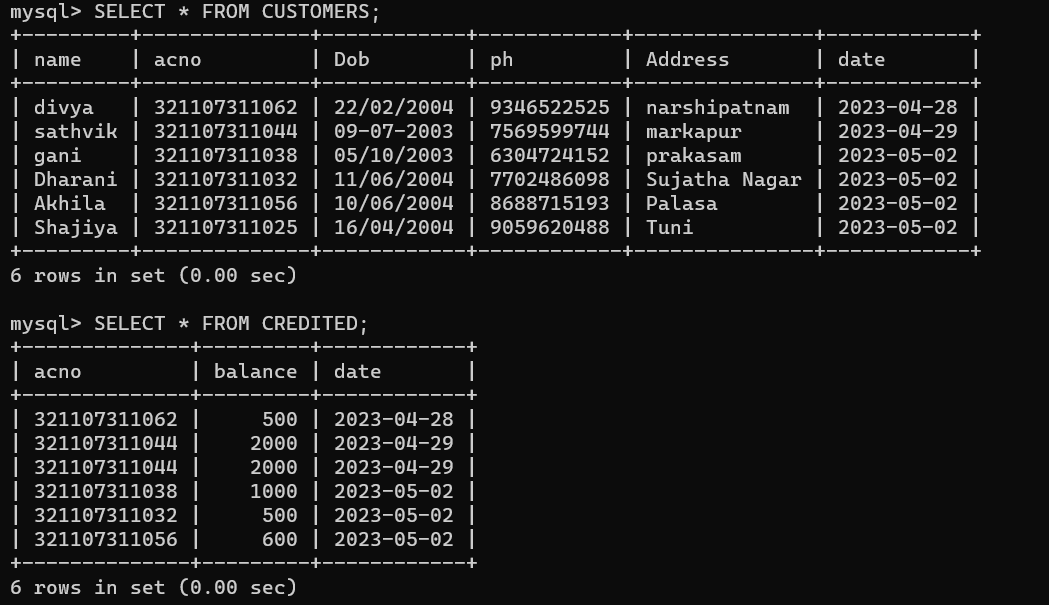
To check your balance, select task 5

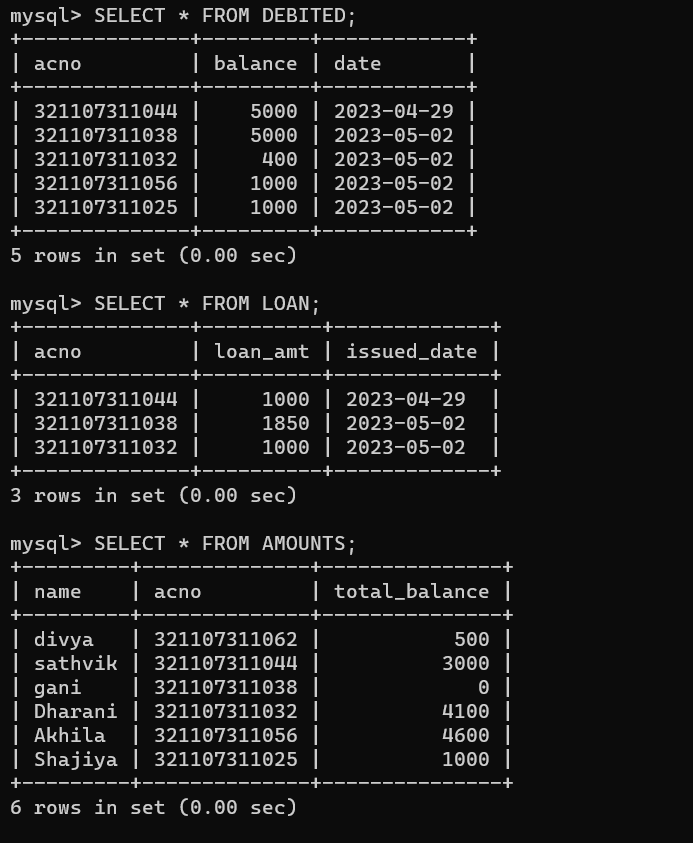
The balance of account no: 321107311045 is 8000.

**TASK-6: REMOVE AN ACCOUNT FROM BANK:**

****

The account 321107311045 has been successfully removed from entire bank records.





The records of the account 321107311045 has been removed from all the tables successfully.

**8.CONCLUSION**

* SQL database management application which is very well used in the modem world in organizing and manipulating a database.
* Though SQL doesn't have the GUI interface like Microsoft access is having and they all manage the database comfortable.
* Depending on the user or users, if an organization has multiple users, then they should go for SQL server-based application.
* This project shows how to create tables in SQL and how to create simple data manipulation language and data definition language and to execute them and also to connect them to the Python language.