|  |
| --- |
| NMIMS University Logo |

**Project Report on**

**“BANKING SYSTEM”**

**Relational Database Management System**

**B.Tech** *2ND YEAR*

**(Branch – CSE)**

|  |
| --- |
|  |

**Submitted To: Submitted By:**

*Prof. Varsha Nemade VED BHATKAR(B205)*

*KUNJ JOSHI (B226)*

**ACKNOWLEDGEMENT**

This Project report was completed because of support from the groupmates, although not all of them can be mentioned.

We are greatly indebted to our good supervisor **Prof. Varsha Nemade** for her useful and necessary observation, suggestions, contribution, and corrections. We would not have been able to achieve anything in this research without your supervision.

**Student’s Name**

**VED BHATKAR**

**KUNJ JOSHI**

**Introduction**

The “Bank Account Management System” project is a model database handling project. This model enables the customers to perform the basic banking transactions by sitting at their office or at homes through PC or laptop. The system provides the access to the customer to create an account, deposit/withdraw the cash from his account, also to view reports of all accounts present. The customers can access the banks website for viewing their Account details and perform the transactions on account as per their requirements.

The primary aim of this “Bank Account Management System” is to provide an improved design methodology, which envisages the future expansion, and modification, which is necessary for a core sector like banking. This necessitates the design to be expandable and modifiable and so a modular approach is used in developing the application software.

Anybody who is an Account holder in this bank can become a member of Bank Account

Management System. He has to fill a form with his personal details and Account Number.

Bank is the place where customers feel the sense of safety for their property. In the bank, customers deposit and withdraw their money. Transaction of money also is a part where customer takes shelter of the bank. Now to keep the belief and trust of customers, there is the positive need for management of the bank, which can handle all this with comfort and ease.

Smooth and efficient management affects the satisfaction of the customers and staff members, indirectly. And of course, it encourages management committee in taking some needed decision for future enhancement of the bank.

Now a day’s, managing a bank is tedious job up to certain limit. So, software that reduces the work is essential. Also, today’s world is a genuine computer world and is getting faster and faster day-by-day. Thus, considering above necessities, the software for bank management has become necessary which would be useful in managing the bank more efficiently.

All transactions are carried out online by transferring from accounts in the same Bank or international bank. The software is meant to overcome the drawbacks of the manual system.

**Abstract**

* The Bank Account Management System is an application for maintaining a person's account in a bank.
* In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System.
* To develop a project for solving financial applications of a customer in banking environment to nurture the needs of an end banking user by providing various ways to perform banking tasks.
* Also, to enable the user’s workspace to have additional functionalities which are not provided under a conventional banking project.
* This project is made possible with python language and MySQL.

**AIM of this project.**

* The main aim of designing and developing this banking System Python primarily based.
* Engineering project is to provide secure and efficient net banking facilities to the banking customers over the internet.
* Python GUI, MYSQL database used to develop this bank application where all banking customers can use,

this GUI to enter details and use it to see their accounts.

* Users will have all options and features in the project,

ranging from creating new account, withdraw amount, amount deposit, Display Customer Details,

Closing a bank account.

**Main Purpose**

* The Traditional way of maintaining details of a user in a bank was to enter the details and record them. Every time the user needs to perform some transactions he has to go to bank and perform the necessary actions, which may not be so feasible all the time.
* It may be a hard-hitting task for the users and the bankers too.
* The project gives real life understanding of Online Banking System and activities performed by various roles in the supply chain.
* Here, we provide automation for banking system through Internet. Online Banking System project captures activities performed by different roles in real life banking which provides enhanced techniques for maintaining the required information up to date, which results in efficiency.
* The project gives real life understanding of Online Banking System and activities performed by various roles in the supply chain.

**Conclusion**

* This project is developed to nurture the needs of a user in a banking sector by embedding all the tasks of transactions taking place in a bank. Future version of this project will still be much enhanced than the current version.
* Banks are providing internet banking services also so that the customers can be attracted. By asking the bank employs we came to know that maximum numbers of internet bank account holders are youth and businessman.
* Online banking is an innovative tool that is fast becoming a necessity. It is a successful strategic weapon for banks to remain profitable in a volatile and competitive marketplace of today. If proper training should be given to customer by the bank employs to open an account will be beneficial secondly the GUI should be made friendlier from where the first-time customers can directly make and access their accounts.
* Thus, the Bank Management System it is developed and executed successfully.

**Reference**

1. Fundamentals of database systems by (Elmasri Navathe, 2000),

Website: https://archive.org/stream/FundamentalsOfDatabaseSystemselmasrinavathe#

page/n51/mode/2up

2. Online Bank Account Management System

Website: https://www.geeksforgeeks.org/er-diagram-of-bank-management-system/

3. Learning MYSQL, Python GUI

Website: http://www.w3schools.com

4. MySQL video tutorials

Website: http://www.freehinditutorial.com, http://www.youtube.com

**GUI PART**

**CODE:**

from tkinter import \*

import tkinter.messagebox as MessageBox

import mysql.connector as a

con = a.connect(host="localhost",user="root",passwd="12345678",database="BankSystem")

def OpenAccount():

win1 = Tk()

win1.title("NEW ACCOUNT")

win1.geometry("500x600")

cid = Label(win1 , text = "Enter your preferred Customer ID : ")

name = Label(win1 , text = "Enter you name : ")

phone\_no = Label(win1 , text = "Enter your phone number : ")

address = Label(win1 , text = "Enter your address : ")

acc\_no= Label(win1 , text = "Enter your preferred account number : ")

acc\_type = Label(win1 , text = "Enter your Account type : ")

bal = Label(win1 , text = "Enter your opening balance : ")

branch\_id = Label(win1 , text = "Enter branch ID : ")

insert\_button = Button(win1 , text="CREATE A NEW ACCOUNT",command=insert)

insert\_button.place(x = 180, y = 400)

cid.place(x = 50, y =120)

name.place(x = 50 , y = 150)

phone\_no.place(x = 50 , y = 180)

address.place(x = 50 , y = 210)

acc\_no.place(x = 50 , y = 240)

acc\_type.place(x = 50 , y = 270)

bal.place(x = 50 , y = 300)

branch\_id.place(x = 50 , y = 330)

global e\_cid

global e\_name

global e\_phoneno

global e\_address

global e\_accno

global e\_acctype

global e\_bal

global e\_branchid

e\_cid = Entry(win1)

e\_cid.place(x = 300 , y = 120)

e\_name = Entry(win1)

e\_name.place(x = 300 , y = 150)

e\_phoneno = Entry(win1)

e\_phoneno.place(x = 300 , y = 180)

e\_address = Entry(win1)

e\_address.place(x = 300 , y = 210)

e\_accno = Entry(win1)

e\_accno.place(x = 300 , y = 240)

e\_acctype = Entry(win1)

e\_acctype.place(x = 300 , y = 270)

e\_bal = Entry(win1)

e\_bal.place(x = 300 , y = 300)

e\_branchid = Entry(win1)

e\_branchid.place(x = 300 , y = 330)

def insert():

Customer\_Id = e\_cid.get()

Customer\_name = e\_name.get();

Phone\_no = e\_phoneno.get();

Address = e\_address.get();

Account\_no = e\_accno.get();

Account\_type = e\_acctype.get();

Balance = e\_bal.get();

Branch\_Id = e\_branchid.get();

if(Customer\_Id=="" or Customer\_name=="" or Phone\_no=="" or Address=="" or Account\_no=="" or Account\_type=="" or Balance=="" or Branch\_Id==""):

MessageBox.showinfo("Insert Status" , "All Fields are required")

else:

c = con.cursor()

data1 = (Customer\_Id , Customer\_name , Phone\_no , Address)

data2 = (Account\_no , Account\_type , Balance , Customer\_Id , Branch\_Id)

sql1 = "insert into Customer values(%s,%s,%s,%s)"

sql2 = "insert into Accounts values(%s,%s,%s,%s,%s)"

c.execute(sql1,data1)

c.execute(sql2,data2)

c.execute("commit");

MessageBox.showinfo("Insert Status" , "New Account has been created!")

con.close();

def DepositAmount():

win2 = Tk()

win2.title("Deposit Amount")

win2.geometry("500x600")

ano1 = Label(win2 , text = "Enter your account number = ")

amount = Label(win2 , text = "Enter amount to be deposited = ")

b1 = Button(win2 , text = "DEPOSIT",command=DepoAmo)

b1.place(x = 180 , y = 400)

ano1.place(x = 50 , y = 180)

amount.place(x = 50 , y = 240)

global e1

global e2

e1 = Entry(win2)

e1.place(x = 300 , y = 180)

e2 = Entry(win2)

e2.place(x = 300 , y = 240)

def DepoAmo():

account1 = e1.get()

am1 = int(e2.get());

if(account1=="" or am1==""):

MessageBox.showinfo("Deposit Status" , "All fields are required")

else:

c = con.cursor()

a = "select Balance from Accounts where Account\_no = %s"

data = (account1,)

c.execute(a,data)

myresult = c.fetchone()

tam = myresult[0] + am1

sql = "update Accounts set Balance = %s where Account\_no = %s"

d = (tam,account1)

c.execute(sql,d)

con.commit()

MessageBox.showinfo("Deposit Status" , "Your amount has to be been deposited to your account!")

con.close();

def WithdrawAmount():

win3 = Tk()

win3.title("Withdraw Amount")

win3.geometry("500x600")

ano1 = Label(win3 , text = "Enter your account number : ")

amount = Label(win3 , text = "Enter amount to be withdrawn : ")

b1 = Button(win3 , text = "Withdraw Amount",command=WithAmo)

b1.place(x=180 ,y = 400)

ano1.place(x = 50 , y = 180)

amount.place(x = 50 , y = 240)

global n1

global n2

n1 = Entry(win3)

n1.place(x = 300 , y = 180)

n2 = Entry(win3)

n2.place(x = 300 , y = 240)

def WithAmo():

account1 = n1.get()

am1 = int(n2.get());

if(account1=="" or am1==""):

MessageBox.showinfo("Withdrawal Status" , "All fields are required")

else:

c = con.cursor()

a = "select Balance from Accounts where Account\_no = %s"

data = (account1,)

c.execute(a,data)

myresult = c.fetchone()

tam = myresult[0] - am1

sql = "update Accounts set Balance = %s where Account\_no = %s"

d = (tam,account1)

c.execute(sql,d)

con.commit()

MessageBox.showinfo("Withdrawal Status" , "Your amount is successfully withdrawed!")

con.close();

def DisplayDetails():

win4 = Tk()

win4.title("Display Details")

win4.geometry("400x300")

cid = Label(win4 , text = "Enter your customer id : ")

b1 = Button(win4 , text = "DISPLAY",command=dispcust)

cid.place(x = 50 , y = 60)

b1.place(x = 120 , y = 200)

global a1

a1 = Entry(win4)

a1.place(x = 200 , y = 60)

def dispcust():

win5 = Tk()

win5.title("Your Customer Deatils")

win5.geometry("500x600")

cid1 = a1.get();

data=cid1

c = con.cursor()

sql = "select \* from Customer limit 0,10"

c.execute(sql,data)

i=0

for Customer in c:

for j in range(len(Customer)):

e = Entry(win5 , width=10)

e.grid(row=i,column=j)

e.insert(END,Customer[j])

i=i+1

win5.mainloop()

def DeleteAccount():

win6 = Tk()

win6.title("Delete Account")

win6.geometry("500x600")

d1 = Label(win6 , text = "Enter your Customer ID: ")

d2 = Label(win6 , text = "Enter your Account number : ")

b1 = Button(win6 , text = "Delete",command=Del)

d1.place(x = 50 , y = 180)

d2.place(x = 50 , y = 270)

b1.place(x = 180 , y = 400)

global f1

global f2

f1 = Entry(win6)

f1.place(x = 300 , y = 180)

f2 = Entry(win6)

f2.place(x = 300 , y = 270)

def Del():

ano1 = f2.get();

sql2 = "delete from Accounts where Account\_no = '%s'" %ano1

data2=ano1

c = con.cursor()

c.execute(sql2,data2)

con.commit()

MessageBox.showinfo("Delete Account" , "Your account has been succesffully deleted!")

window = Tk()

window.title("Bank Management System")

window.geometry("500x700")

window.configure(bg="Turquoise")

button1 = Button(window , text = "Open A New Account",font=("bold",12),command=OpenAccount)

button2 = Button(window , text = "Display Customer Details",font=("bold",12),command=DisplayDetails)

button3 = Button(window , text = "Withdraw Amount",font=("bold",12),command=WithdrawAmount)

button4 = Button(window , text = "Deposit Amount",font=("bold",12),command=DepositAmount)

button5 = Button(window , text = "Close An Account",font=("bold",12),command=DeleteAccount)

button6 = Button(window , text = "Balance Enquiry",font=("bold",10))

button1.place(x = 100 , y = 180)

button2.place(x = 100 , y = 360)

button3.place(x = 100 , y = 540)

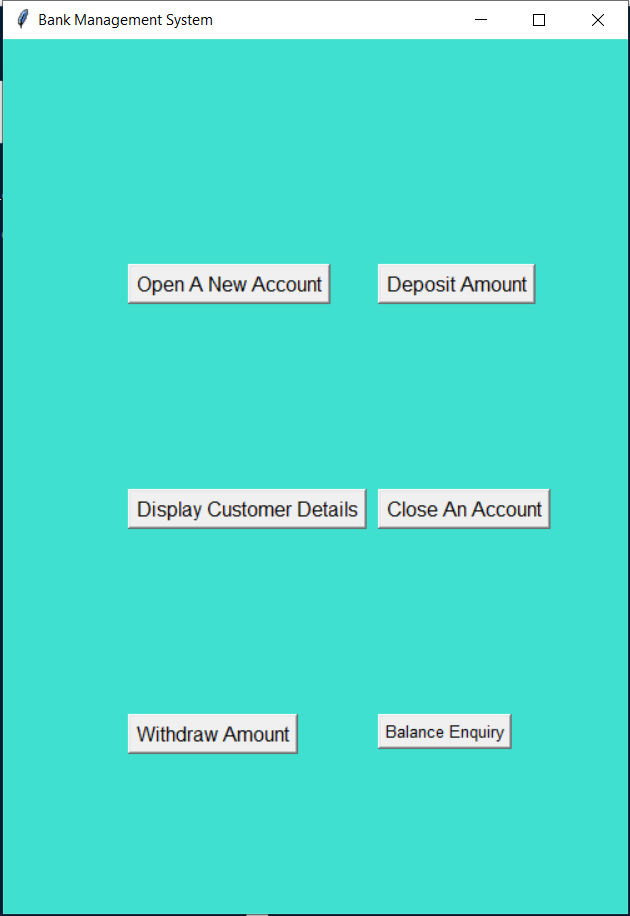
button4.place(x = 300 , y = 180)

button5.place(x = 300 , y = 360)

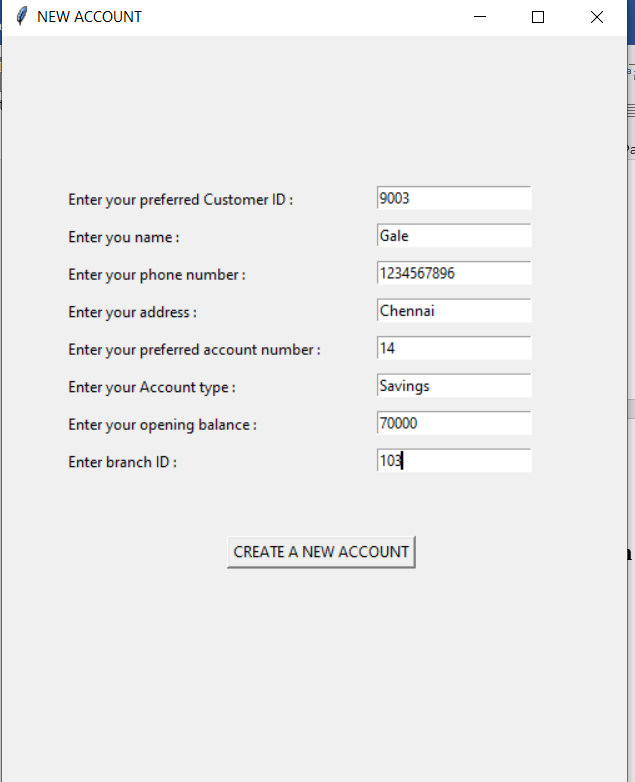
button6.place(x = 300 , y = 540)

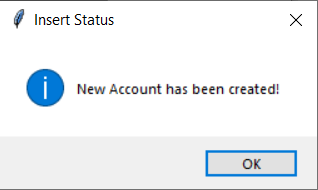
window.mainloop()

**OUTPUT:**

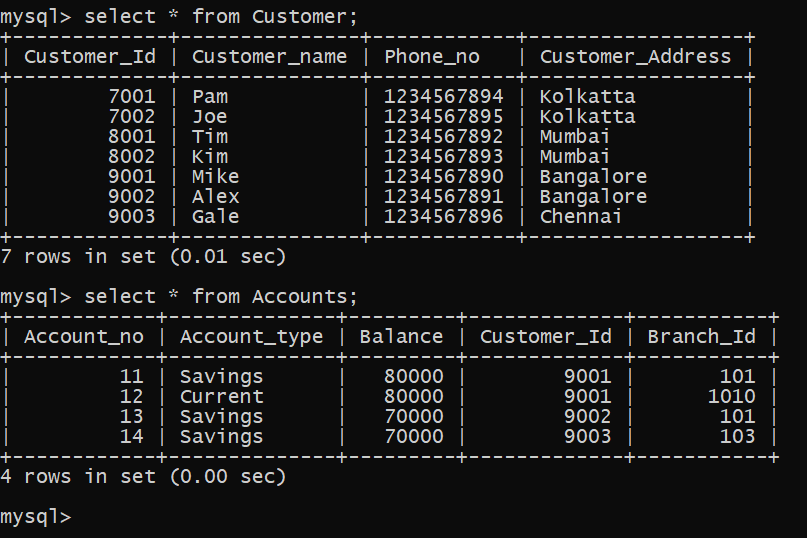


**Clicking on Open a new account button and entering values we get,**

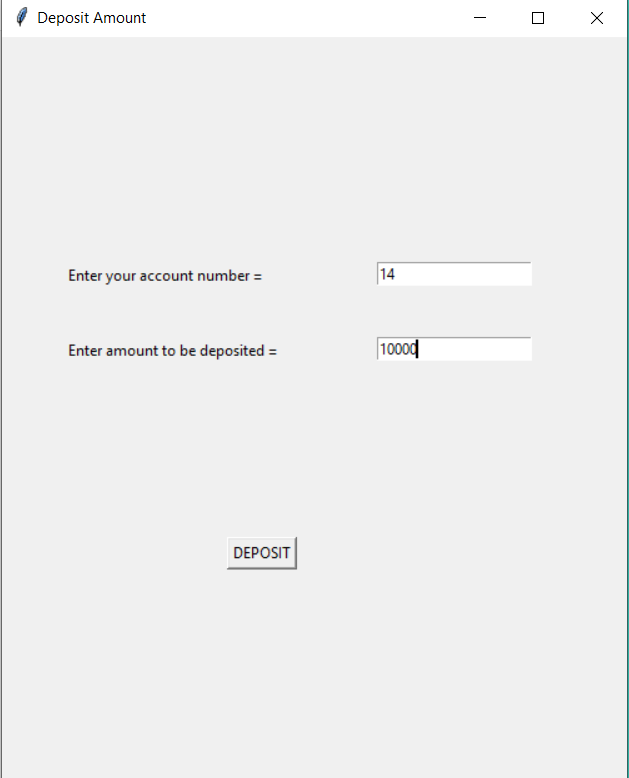


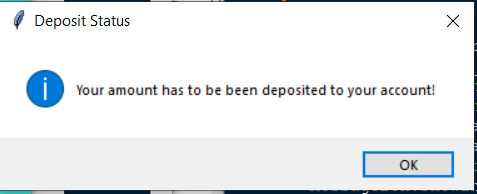


**Checking in database, we see that entry is done,**

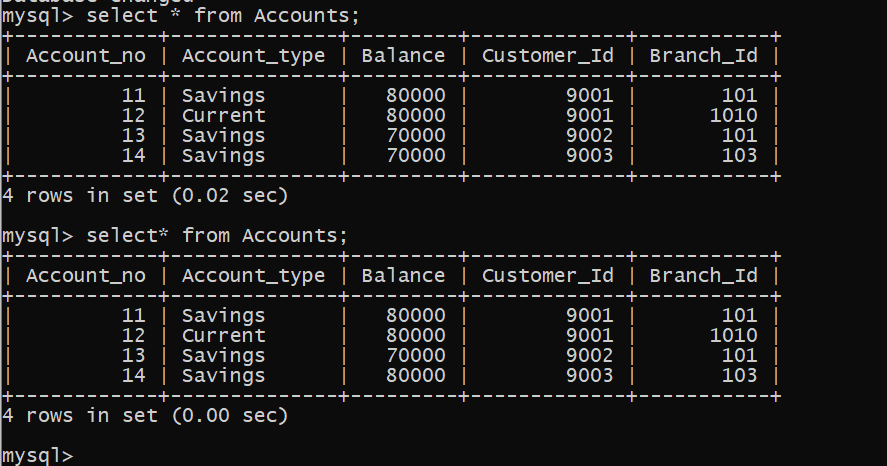


**Now clicking on Deposit button we can see,**





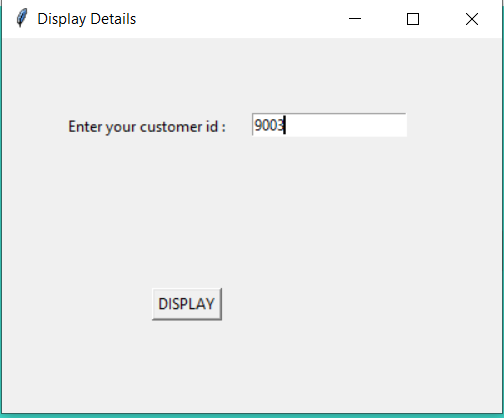
**Checking in mysql database ,**

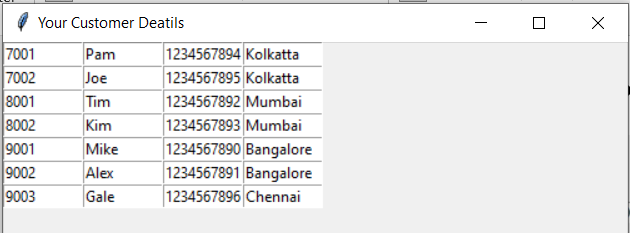


**Balance has been increased by Deposit action**

**Similarly Withdraw button acts the same.**

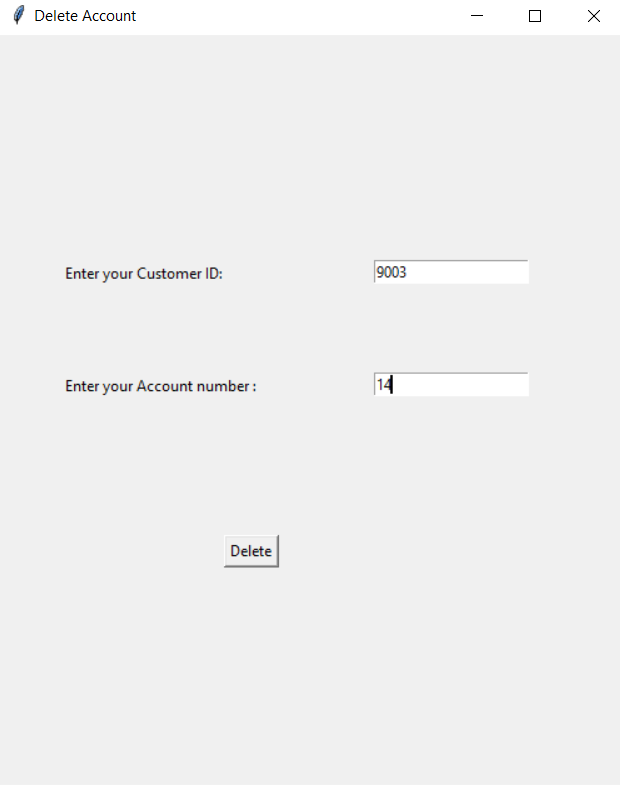
**Moving ahead with Display Details ,**

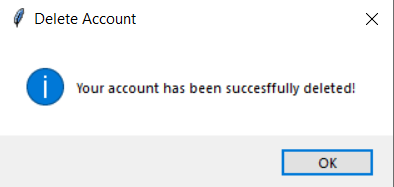




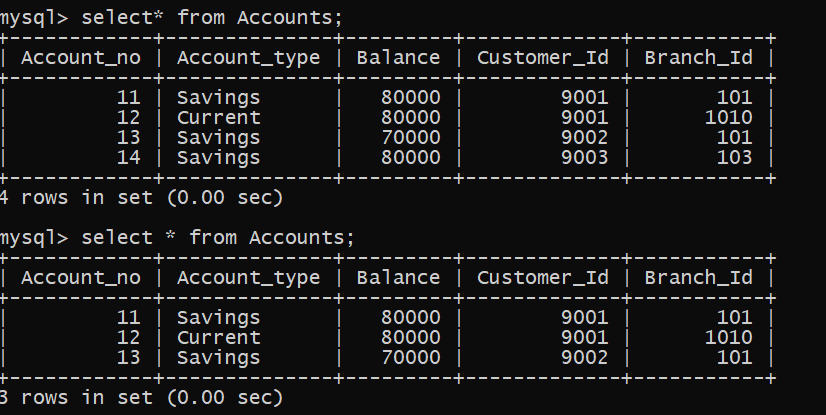
**The customer details have been shown .**

**Now deleting an account using delete button ,**





**Checking in mysql database ,**



**The account has been deleted.**

**THE END**