import mysql.connector

from mysql.connector import Error

try:

mydb = mysql.connector.connect(host='localhost',

user='root',

passwd='root123',

#database='BankDB' # \*\*Remove the comment after creating database \*\*

)

mycursor = mydb.cursor() # Created Database BankDB

mycursor.execute('SHOW DATABASES LIKE "BankDB"')

S = list(map(str, mycursor.fetchall()))

x = 0

print(S)

if ("('BankDB',)" in S):

mydb = mysql.connector.connect(host='localhost',

user='root',

passwd='root123',

database='BankDB'

)

mycursor = mydb.cursor()

else:

mycursor.execute('CREATE DATABASE BankDB')

mydb = mysql.connector.connect(host='localhost',

user='root',

passwd='root123',

database='BankDB'

)

mycursor = mydb.cursor()

mycursor.execute(

'create table bank(ACCNO varchar(10),NAME varchar(20),MOBILE varchar(10),EMAIL varchar(14),ADDRESS varchar(14),CITY varchar(10),COUNTRY varchar(10),BALANCE float(16))')

print("Table Created")

except Error as e:

print("Error is ", e)

def Menu(): # Function to display the menu

print("\*" \* 140)

print("MAIN MENU")

print("1. Insert Record/Records")

print("2. Display Records as per Account Number")

print(" a. Sorted as per Account Number")

print(" b. Sorted as per Customer Name")

print(" c. Sorted as per Customer Balance")

print("3. Search Record Details as per the account number")

print("4. Update Record")

print("5. Delete Record")

print("6. TransactionsDebit/Withdraw from the account")

print(" a. Debit/Withdraw from the account")

print(" b. Credit into the account")

print("7. Exit")

print("\*" \* 140)

def MenuSort():

print(" a. Sorted as per Account Number")

print(" b. Sorted as per Customer Name")

print(" c. Sorted as per Customer Balance")

print(" d. Back")

def MenuTransaction():

print(" a. Debit/Withdraw from the account")

print(" b. Credit into the account")

print(" c. Back")

def Create():

try:

mycursor.execute(

'create table bank(ACCNO varchar(10),NAME varchar(20),MOBILE varchar(10),EMAIL varchar(20),ADDRESS varchar(10),CITY varchar(10),COUNTRY varchar(10),BALANCE integer(15))')

print("Table Created")

Insert()

except:

print("Table Already Exist")

Insert()

def Insert():

while True: # Loop for accepting records

Acc = input("Enter account no")

Name = input("Enter Name")

Mob = input("Enter Mobile")

email = input("Enter Email")

Add = input("Enter Address")

City = input("Enter City")

Country = input("Enter Country")

Bal = float(input("Enter Balance"))

Rec = [Acc, Name.upper(), email, Mob, Add.capitalize(), City.upper(), Country.upper(), Bal]

print(Rec)

Cmd = "insert into bank values(%s,%s,%s,%s,%s,%s,%s,%s)"

mycursor.execute(Cmd, Rec)

mydb.commit()

print('VALUES INSERTED INTO TABLE')

ch = input("Do you want to enter more records")

if ch == 'N' or ch == 'n':

break

def DispSortAcc(): # Function to Display records as per ascending order of Account Number

try:

cmd = "select \* from bank order by ACCNO"

mycursor.execute(cmd)

S = mycursor.fetchall()

F = "%15s %15s %15s %15s %15s %15s %15s %15s"

print(

F % ("ACCNO ", " NAME ", "EMAIL ADDRESS ","MOBILE", " COMPLETE ADDRESS", "CITY", "COUNTRY", "BALANCE"))

print("=" \* 140)

for i in S:

for j in i:

print("%14s" % j, end=' ')

print()

print("=" \* 140)

except:

print("Table doesn't exist")

def DispSortName(): # Function to Display records as per ascending order of Name

try:

cmd = "select \* from bank order by NAME"

mycursor.execute(cmd)

S = mycursor.fetchall()

F = "%15s %15s %15s %15s %15s %15s %15s %15s"

print(F % ("ACCNO", "NAME", "EMAIL ADDRESS","MOBILE", "COMPLETE ADDRESS", "CITY", "COUNTRY", "BALANCE"))

print("=" \* 140)

for i in S:

for j in i:

print("%14s" % j, end=' ')

print()

print("=" \* 140)

except:

print("Table doesn't exist")

def DispSortBal(): # Function to Display records as per ascending order of Balance

try:

cmd = "select \* from bank order by BALANCE"

mycursor.execute(cmd)

S = mycursor.fetchall()

F = "%15s %15s %15s %15s %15s %15s %15s %15s"

print(F % ("ACCNO", "NAME", "EMAIL ADDRESS", "MOBILE", "COMPLETE ADDRESS", "CITY", "COUNTRY", "BALANCE"))

print("=" \* 140)

for i in S:

for j in i:

print("%14s" % j, end=' ')

print()

print("=" \* 140)

except:

print("Table doesn't exist")

def DispSearchAcc(): # Function to Search for the Record from the File with respect to the account number

try:

cmd = "select \* from bank"

mycursor.execute(cmd)

S = mycursor.fetchall()

ch = input("Enter the accountno to be searched")

for i in S:

if i[0] == ch:

print("=" \* 140)

F = "%15s %15s %15s %15s %15s %15s %15s %15s"

print(

F % ("ACCNO", "NAME", "EMAIL ADDRESS", "MOBILE", "COMPLETE ADDRESS", "CITY", "COUNTRY", "BALANCE"))

print("=" \* 140)

for j in i:

print('%14s' % j, end=' ')

else:

print("Record Not found")

except:

print("Record Not found / Table doesn't exist")

def Delete(): # Function to delete the details of a customer

try:

cmd = "select \* from BANK"

mycursor.execute(cmd)

S = mycursor.fetchall()

A = input("Enter the account no whose details to be changed")

for i in S:

i = list(i)

if i[0] == A:

cmd = "delete from bank where accno=%s"

val = (i[0],)

mycursor.execute(cmd, val)

mydb.commit()

print("Account Deleted")

break

except:

print("Record not found No such Table")

def Update(): # Function to change the details of a customer

try:

cmd = "select \* from bank"

mycursor.execute(cmd)

S = mycursor.fetchall()

A = input("Enter the accound no whose details to be changed")

for i in S:

i = list(i)

if i[0] == A:

ch = input("Change Name(Y/N)")

if ch == 'y' or ch == 'Y':

i[1] = input("Enter Name")

i[1] = i[1].upper()

ch = input("Change Mobile(Y/N)")

if ch == 'y' or ch == 'Y':

i[2] = input("Enter Mobile")

ch = input("Change Email(Y/N)")

if ch == 'y' or ch == 'Y':

i[3] = input("Enter email")

i[3] = i[3].upper()

ch = input("Change Address(Y/N)")

if ch == 'y' or ch == 'Y':

i[4] = input("Enter Address")

i[4] = i[4].upper()

ch = input("Change city(Y/N)")

if ch == 'y' or ch == 'Y':

i[5] = input("Enter City")

i[5] = i[5].upper()

ch = input("Change Country(Y/N)")

if ch == 'y' or ch == 'Y':

i[6] = input("Enter country")

i[6] = i[6].upper()

ch = input("Change Balance(Y/N)")

if ch == 'y' or ch == 'Y':

i[7] = float(input("Enter Balance"))

cmd = "UPDATE bank SET NAME=%s,MOBILE=%s,EMAIL=%s,ADDRESS=%s,CITY=%s,COUNTRY=%s,BALANCE=%s WHERE ACCNO=%s"

val = (i[1], i[2], i[3], i[4], i[5], i[6], i[7], i[0])

mycursor.execute(cmd, val)

mydb.commit()

print("Account Updated")

break

else:

print("")

except:

print("Record not found / No such table")

def Debit(): # Function to Withdraw the amount by assuring the min balance of Rs 5000

try:

cmd = "select \* from bank"

mycursor.execute(cmd)

S = mycursor.fetchall()

print("Please Note that the money can only be debited if min balance of Rs 5000 exists")

acc = input("Enter the account no from which the money is to be debited")

for i in S:

i = list(i)

if i[0] == acc:

Amt = float(input("Enter the amount to be withdrawn "))

if i[7] - Amt >= 5000:

i[7] -= Amt

cmd = "UPDATE bank SET BALANCE=%s WHERE ACCNO=%s"

val = (i[7], i[0])

mycursor.execute(cmd, val)

mydb.commit()

print("Amount Debited")

break

else:

print("There must be min balance of Rs 5000")

break

else:

print("Record Not found")

except:

print("Table Doesn't exist")

def Credit(): # Function to Withdraw the amount by assuring the min balance of Rs 5000

try:

cmd = "select \* from bank"

mycursor.execute(cmd)

S = mycursor.fetchall()

acc = input("Enter the account no from which the money is to be CREDITED")

for i in S:

i = list(i)

if i[0] == acc:

Amt = float(input("Enter the amount to be credited "))

i[7] += Amt

cmd = "UPDATE bank SET BALANCE=%s WHERE ACCNO=%s"

val = (i[7], i[0])

mycursor.execute(cmd, val)

mydb.commit()

print("Amount Credited")

break

else:

print("Record Not found")

except:

print("Table Doesn't exist")

while True:

Menu()

ch = input("Enter your Choice")

if ch == "1":

Insert()

elif ch == "2":

while True:

MenuSort()

ch1 = input("Enter choice a/b/c/d")

if ch1 in ['a', 'A']:

DispSortAcc()

elif ch1 in ['b', 'B']:

DispSortName()

elif ch1 in ['c', 'C']:

DispSortBal()

elif ch1 in ['d', 'D']:

print("Back to the main menu")

break

else:

print("Invalid choice")

elif ch == "3":

DispSearchAcc()

elif ch == "4":

Update()

elif ch == "5":

Delete()

elif ch == "6":

while True:

MenuTransaction()

ch1 = input("Enter choice a/b/c")

if ch1 in ['a', 'A']:

Debit()

elif ch1 in ['b', 'B']:

Credit()

elif ch1 in ['c', 'C']:

print("Back to the main menu")

break

else:

print("Invalid choice")

elif ch == "7":

print("Exiting...")

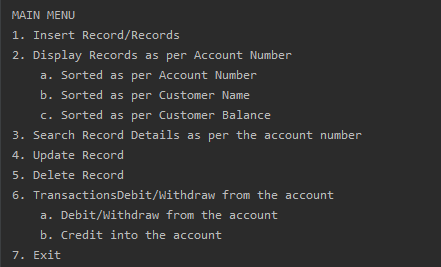
break

else:

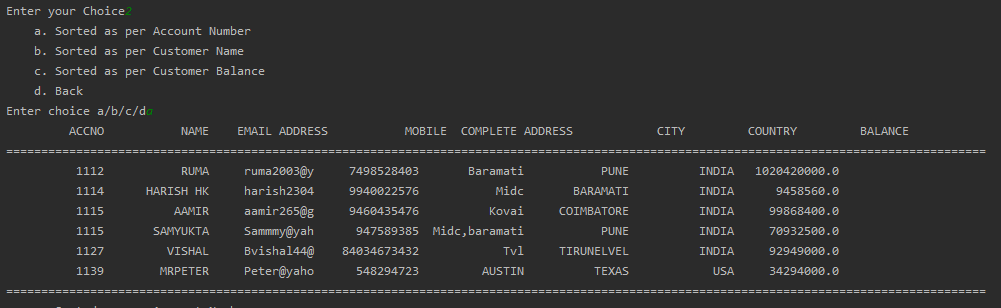
print("Wrong Choice Entered")

**OUTPUT :**

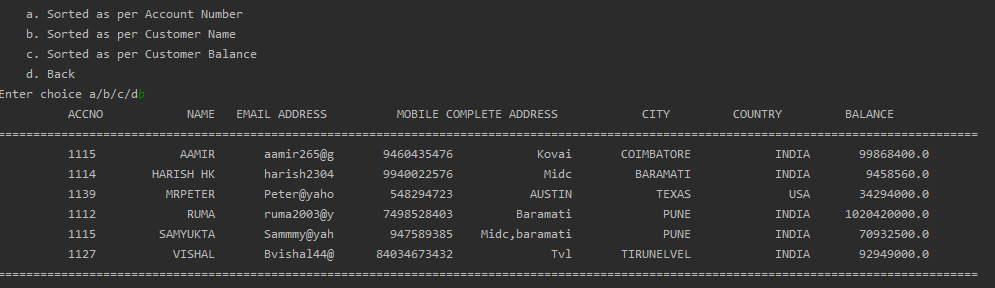
**MAIN MENU**

****

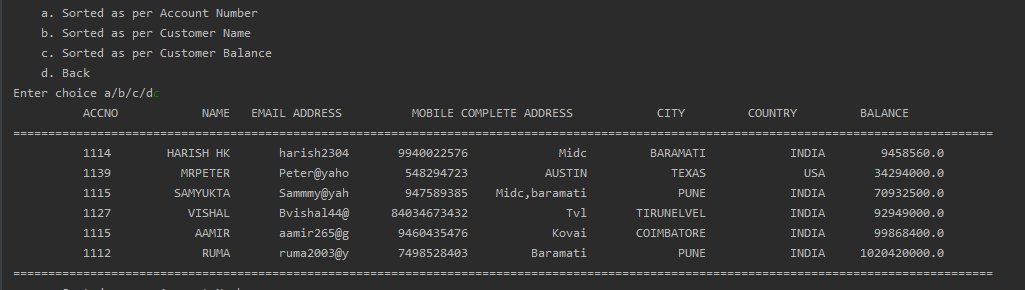
**SORT BY ACCOUNT NUMBER**

****

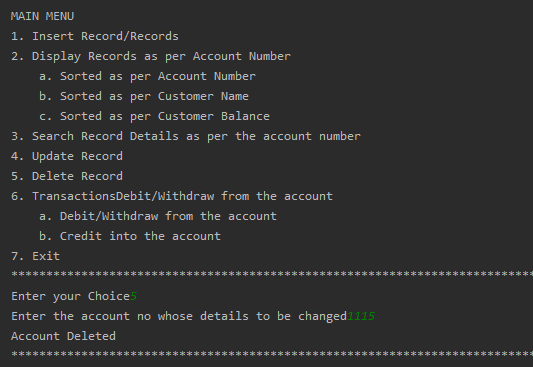
**SORT BY NAME**

****

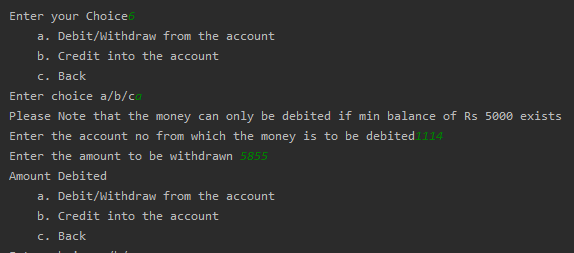
**SORT BY BALANCE**

****

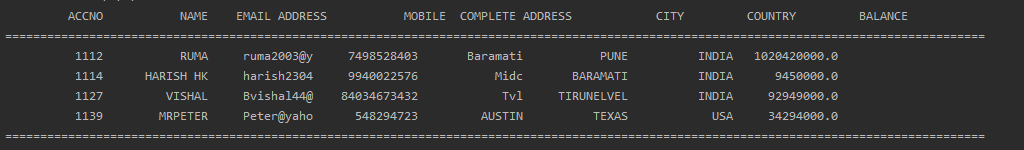
**DELETE RECORD**

****

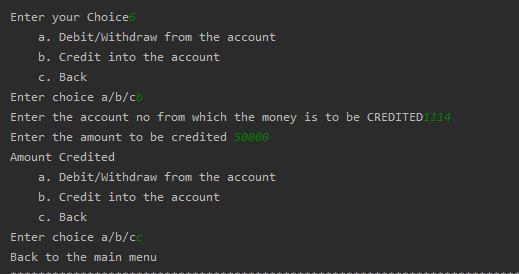
**DEBIT AMOUNT**

****

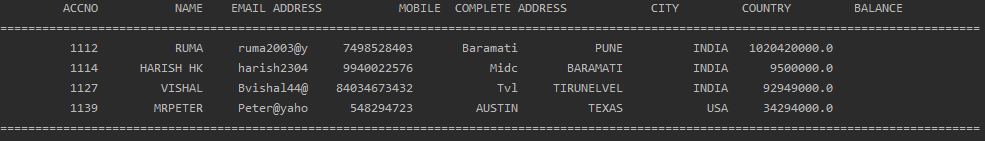
**DISPLAYING TABLE DATA AFTER DEBIT**

****

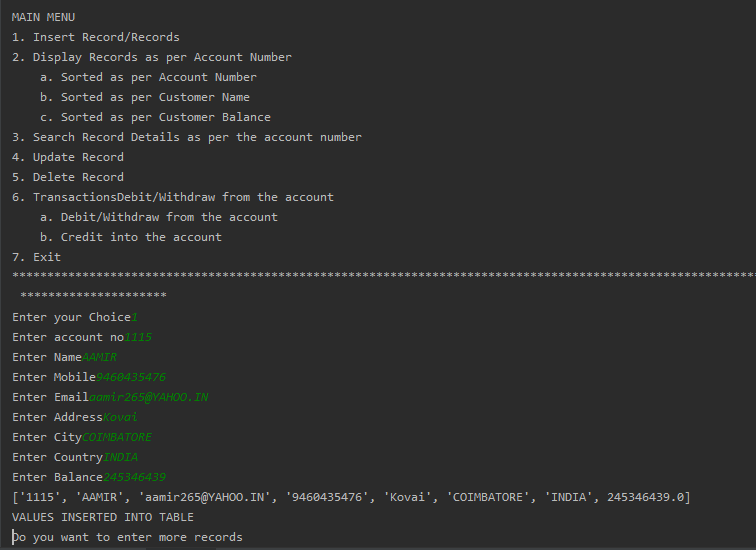
**CREDIT AMOUNT**

****

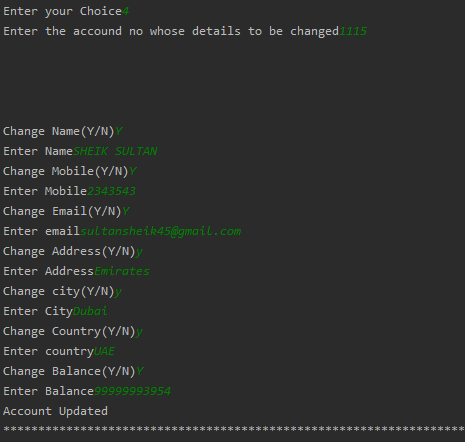
**DISPLAY AFTER CREDIT**

****

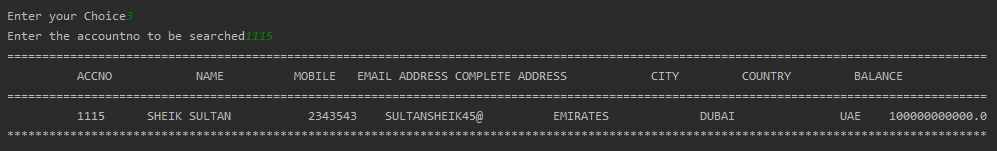
**INSERT DATA INTO TABLE**

****

**UPDATE ACCOUNT DATA**

****

**SEARCH FOR ACCOUNTS BY ACCOUNT NUMBER**

****