```
import mysql.connector as c
con=c.connect(host='localhost',user='root',passwd='admin
',database='bank')
cursor=con.cursor()
```

#TO CREATE ACCOUNT

```
def insert():
   while True:
      accno=int(input("enter account no"))
      name=input("enter name")
      phno=int(input("enter phone no"))
      city=input("enter city")
      bal=int(input("enter balance"))
      if bal<1000:
         print("CANNOT ADD AMOUNT LESS THAN 1000")
         continue
      panno=input("enter pan no")
      adharno=input("enter adhar no")
      typ=input("enter type of account: SAVING or
CURRENT")
      dob=input("enter DOB")
      adress=input("enter address")
      s="insert into account
values('{}','{}',{},'{}',{},'{}','{}','{}','{}','{}')".format(accno,na
me,phno,city,bal,panno,adharno,typ,dob,adress)
      cursor.execute(s)
      con.commit()
      print("your account has been created")
      x=int(input("enter 1 for more\nenter 2 to exit"))
      if x==2:
         break
```

#DISPLAY ALL ACCOUNTS

```
def showacc():
   s="select * from account"
   cursor.execute(s)
   record=cursor.fetchall()
   for i in record:
      print(i)
```

#DISPLAY SELECTED ACCOUNTS

```
def selectacc():
   accno=int(input("enter  acc no to be searched"))
   sel="select * from account where accno= %s" %(accno,)
   cursor.execute(sel)
   record=cursor.fetchone()
   if cursor.rowcount>0:
      print(record)
   else:
      print("          INVALID ACCOUNT NUMBER          ")
```

#DISPLY BALANCE

```
def showbal():
   n=int(input("enter  acc no "))
   sel="select * from account where accno= %s" %(n,)
   cursor.execute(sel)
   record=cursor.fetchone()
   if cursor.rowcount>0:
      print("\t\tBALANCE IS :::::",record[4])
   else:
      print("          INVALID ACCOUNT NUMBER          ")
```

#DEPOSIT AMOUNT

```
def deposit():
   amount=int(input("enter amount to be deposited"))
   accno=int(input("enter account no"))
   depo="select * from account where accno= %s" %(n,)
   cursor.execute(depo)
   record=cursor.fetchone()
   updation="update account set bal=bal+{} where
accno='{}' ".format(amount,accno)
   cursor.execute(updation)
   con.commit()
   if cursor.rowcount>0:
      print("        amount deposited         ")
   else:
      print("           INVALID ACCOUNT NUMBER          ")
```

 # TO WITHDRAW

```
def withdraw():
   amount=int(input("enter amount to be withdrawn "))
   accno=int(input("enter account no"))
   wtdw="select * from account where accno= %s" %(n,)
   cursor.execute(wtdw)
   record=cursor.fetchone()
  # if cursor.rowcount>0:
   if amount>record[4]:
      print('NOT HAVING SUFFICIENT AMOUNT IN
ACCOUNT')
   elif amount<record[4]:
      updation="update account set bal=bal-{} where
accno='{}' ".format(amount,accno)
```

```
        cursor.execute(updation)
        con.commit()
        if cursor.rowcount>0:
            print('AMOUNT WITHDRAWN')
        else:
            print("          INVALID ACCOUNT NUMBER          ")
```

#TO CLOSE ACCOUNT

```
def closeacc():
                accno=int(input("enter acc no to be closed"))
                delt="delete from account where accno='{}'
".format(accno)
                cursor.execute(delt)
                con.commit()
                if cursor.rowcount>0:
                   print("        ACCOUNT CLOSED        ")
                else:
                    print("          INVALID ACCOUNT NUMBER
")
```

#To Update the Address/ Phone number

```
def update():
   while True:
        choose=int(input ("enter 1 to change address/nenter
2 to change phoneno/nenter 3 to exit"))
        if choose==1:
           accno=int(input("enter accno"))
           adress=input("enter new address")
           newadd="update account set address='{}' where
accno={}".format(address,accno)
           cursor.execute(newadd)
```

```
            con.commit()
            if cursor.rowcount>0:
               print('ADDRESS CHANGED')
            else:
               print("          INVALID ACCOUNT NUMBER          ")


         if choose==2:
            accno=int(input("enter accno"))
            phno=input("enter new phno")
            newph="update account set phno='{}' where
accno={}".format(phno,accno)
            cursor.execute(newph)
            con.commit()
            if cursor.rowcount>0:
               print('PHNO CHANGED')
            else:
               print("          INVALID ACCOUNT NUMBER          ")
         if choose==3:
            break
```

#Main

```
def main():
   while True:

      print("""
1. OPEN NEW ACCOUNT
2. DISPLAY ALL ACCOUNT HOLDER'S DETAILS
3. DISPLAY PARTICULAR ACCOUNT HOLDER'S DETAILS
4. DISPLAY CURRENT BALANCE
5. DEPOSIT AMOUNT
6. WITHDRAW AMOUNT
```

```
7. CLOSE AN ACCOUNT
8.CHANGE PHNO/ADDRESS
9. TO EXIT
            """)
      choice=int(input("ENTER TASK NUMBER"))
      if choice==1:
         insert()
      elif choice==2:
         showacc()
      elif choice==3:
         selectacc()
      elif choice==4:
         showbal()
      elif choice==5:
         deposit()
      elif choice==6:
         withdraw()
      elif choice==7:
         closeacc()
      elif choice==8:
         update()
      elif choice==9:
         break
      else:
         print(" \t\t\tSELECT ANY OPTION FROM THE
MENU GIVEN ABOVE")
```