# ***Half Yearly Exam and Board Practical Exam Instructions:***

***The below file contains***

1. ***12 nos. of python programs***
2. ***6 nos. of SQL connectivity question and the respective queries***
3. ***1 no. of SQL - Connectivity template***

***Prepare all the contents of this file alone for the practical examination.***

## 1- SEARCHING

To write a menu – driven program to search for a Key element using the following techniques based on user's choice.

1. Linear Search Technique
2. Binary Search Technique
3. Exit

```
def lsearch(L,key):
    n=len(L)
    for i in range(n):
        if L[i]==key:
            print("Element found at:",i+1)
            break
    else:
        print("Element not found")

def bsearch(L,key):
    low=0
    high=len(L)-1
    while low<=high:
        mid=(low+high)//2
        if L[mid]==key:
            print("Element found at:",mid+1)
            break
        elif L[mid]<key:
            low=mid+1
        else:
            high=mid-1
    else:
        print("Element not found")
print('''Choose searching technique
1.Linear search
2.Binary search
3.Exit''')
while True:
    ch=int(input("Enter your choice"))
    if ch==1:
        L=eval(input("Enter list to be searched"))
        x=int(input("Enter element to be searched:"))
        lsearch(L,x)
    elif ch==2:
        L=eval(input("Enter list to be searched in ascending order"))
        x=int(input("Enter element to be searched:"))
        bsearch(L,x)
    elif ch==3:
        break
    else:
        print("Invalid choice")
```

## 2- SORTING

To write a menu – driven program to sort the list of numbers using following techniques based on user's choice.

1. Bubble Sort.
2. Insertion Sort.
3. Exit

```
def bsort(L):
    n=len(L)
    for i in range(0,n-1):
        for j in range(0,n-i-1):
            if L[j]<L[j+1]:
                L[j],L[j+1]=L[j+1],L[j]
    print("Sorted list is:",L)

def isort(L):
    n=len(L)
    for i in range(1,n):
        locp=L[i]
        j=i-1
        while j>=0 and locp<L[j]:
            L[j+1]=L[j]
            j=j-1
        else:
            L[j+1]=locp
    print("Sorted list is:",L)


print('''Choose sorting tecnique
1.Bubble sort
2.Insertion sort
3.Exit''')
while True:
   ch=int(input("Enter your choice"))
   if ch==1:
      L=eval(input("Enter list to be sorted"))
      bsort(L)
   elif ch==2:
      L=eval(input("Enter list to be sorted"))
      isort(L)
   elif ch==3:
      break
   else:
      print("Invalid choice")
print("Completed...")
```

## 3- TEXT FILE-1

*To write a menu – driven program to perform the following on a text file.*

1. *Replacing spaces by # in each line.*
2. *Counting the numbers of lines.*
3. *Copying the line starting with "T" to another file.*
4. *Exit*

```
S="""Our deepest fear is not that we are inadequate
Our deepest fear is that we are powerful beyond measure
It is our light, not our darkness
That most frightens us
We ask ourselves
Who am I to be brilliant, gorgeous, talented, fabulous?
Actually, who are you not to be ?
You are a child of God
This is new file
"""
f=open('text.txt','w')
f.write(S)
f.close()

def replace_space():
   f=open('text.txt','r')
   s=f.read()
   print(s.replace(' ','#'))

def count_lines():
   f=open('text.txt','r')
   l=f.readlines()
   c=len(l)
   print("Total number of lines:",c)
```

```
def copy():
    f1=open('text.txt','r')
    f2=open('new.txt','w')
    l=f1.readlines()
    for i in l:
        if i[0]=='T':
            f2.write(i)
    f1.close()
    f2.close()
    print("Copied...")
    f2=open('new.txt','r')
    s=f2.read()
    print(s)
    f2.close()

print("""1.Replacing spaces by # in each line.
2.Counting the numbers of lines.
3.Copying the line starting with “T” to another file.
4.Exit
""")
while True:
    c=int(input("Enter your choice:"))
    if c==1:
        replace_space()
    elif c==2:
        count_lines()
    elif c==3:
        copy()
    elif c==4:
        print("Thank you")
        break
```

## 4- TEXT FILE-2

To write a menu – driven program to perform the following on a text file.

1. Display the number of vowels
2. Display the number of consonants
3. Display the number of upper case characters
4. **Display the number of lower case characters**

```
S="""Our deepest fear is not that we are inadequate
Our deepest fear is that we are powerful beyond measure
It is our light, not our darkness
That most frightens us
We ask ourselves
Who am I to be brilliant, gorgeous, talented, fabulous?
Actually, who are you not to be ?
You are a child of God
```

```
"""
f=open('text.txt','w')
f.write(S)
f.close()

def count_vowel():
    f=open('text.txt','r')
    check=f.read()
    v=0
    for i in check:
        if i in "aeiouAEIOU":
            v+=1
    print("Number of vowels:",v)
    f.close()

def count_cons():
    f=open('text.txt','r')
    check=f.read()
    c=0
    for i in check:
        if i in "bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ":
            c+=1
    print("Number of consonants:",c)
    f.close()

def count_lower():
    f=open('text.txt','r')
    check=f.read()
    l=0
    for i in check:
        if i.islower():
            l+=1
    print("Number of lowercase characters:",l)
    f.close()

def count_upper():
    f=open('text.txt','r')
    check=f.read()
    u=0
    for i in check:
        if i.isupper():
            u+=1
    print("Number of uppercase characters:",u)
    f.close()


print("""
1.Count vowels
2.Count consonants
3.Count lowercase character
4.Count uppercase characters
5.Exit
""")
while True:
    c=int(input("Enter your choice:"))
    if c==1:
        count_vowel()
    elif c==2:
       count_cons()
    elif c==3:
        count_lower()
    elif c==4:
        count_upper()
    elif c==5:
        print("Thank you")
        break
    else:
        print("Invalid choice...")
```

## 5. BINARY FILE-1

To write a menu – driven program to perform the following tasks with Students’ Mark Details {rno,sname,total} on a Binary file

1. *Insert record*
2. *Display record*
3. *Update record*
4. *Exit*

```
import pickle
def insert():
    rno=int(input("enter rollno"))
    name=input("enter name")
    marks=int(input("enter marks"))
    rec={"rno":rno,"name":name,"marks":marks}
    f=open("student.dat","ab")
    pickle.dump(rec,f)
    print("record inserted")
    f.close()
def display():
    f=open("student.dat","rb")
    while True:
        try:
            rec=pickle.load(f)
            print(rec)
        except EOFError:
            break
    f.close()
def search():
    r=int(input("enter rollno"))
    f=open("student.dat","rb")
    while True:
        try:
            rec=pickle.load(f)
            if rec["rno"]==r:
                print(rec)
                break
        except EOFError:
            break
    else:
        print("Record not found")
    f.close()
def update():
    r=int(input("Enter rollno to be updated for"))
    m=int(input("Enter marks to be updated"))
    templist=[]
    found=False
    f=open("student.dat","rb")
    while True:
        try:
            rec=pickle.load(f)
            if rec["rno"]==r:
                rec["marks"]=m
                templist.append(rec)
                print("Record updated..")
                found=True
            else:
                templist.append(rec)
        except EOFError:
            break
    f.close()
    if found==True:
        f=open("student.dat","wb")
        for rec in templist:
            pickle.dump(rec,f)
        f.close()
    else:
        print("record not found")

print("""1.INSERT RECORD
2.DISPLAY RECORD
3.UPDATE RECORD
4.EXIT""")
while True:
    choice=int(input("enter your choice"))
    if choice==1:
        insert()
    elif choice==2:
        display()
    elif choice==3:
        update()
    elif choice==4:
        print("Thank you")
        break
    else:
        print("invalid choice")
```

## 6. BINARY FILE-2

To write a menu – driven program to perform the following tasks with Employees salary Details { Eno,Ename,salary} on a Binary file

1. Insert record
2. Display record
3. Delete record
4. Exit

```
import pickle
def insert():
    eno=int(input("enter employee no"))
    ename=input("enter employee name")
    salary=int(input("enter salary"))
    rec={"eno":eno,"ename":ename,"salary":salary}
    f=open("employee.dat","ab+")
    pickle.dump(rec,f)
    print("record inserted")
    f.close()
def display():
    f=open("employee.dat","rb")
    while True:
        try:
            rec=pickle.load(f)
            print(rec)
        except EOFError:
            break
    f.close()
def delete():
    r=int(input("enter employee number to be deleted"))
    templist=[]
    found=False
    f=open("employee.dat","rb")
    while True:
        try:
            rec=pickle.load(f)
            if rec["eno"]==r:
                print("record deleted")
                found=True
            else:
                templist.append(rec)
        except EOFError:
            break
    f.close()
    if found==True:
        f=open("employee.dat","wb")
        for row in templist:
            pickle.dump(row,f)
        f.close()
    else:
        print("record not found")

print("""1.INSERT RECORD
2.DISPLAY RECORD
3.DELETE RECORD
4.EXIT""")
while True:
    choice=int(input("enter your choice"))
    if choice==1:
        insert()
    elif choice==2:
        display()
    elif choice==3:
        delete()
    elif choice==4:
        print("Thank you")
        break
    else:
        print("invalid choice")
```

## 7. CSV-1

To write a menu – driven program to perform the following tasks on a CSV file with book details[Bno,Bname,Bauthor,Bpublisher]

1. Add New Book
2. View Books
3. Update Book Info
4. Exit

```
import csv
data=[[1,'book1','author1','pub1'],
      [2,'book2','author2','pub2'],
      [3,'book3','author3','pub3'],
      [4,'book4','author4','pub4'],
      [5,'book5','author5','pub5']]

with open('book.csv','w',newline='') as f:
    writer=csv.writer(f)
    writer.writerows(data)

def insert():
  with open('book.csv','a',newline='') as f:
    book_data=eval(input('enter the book details to be added as a list'))
    writer=csv.writer(f)
    writer.writerow(book_data)
    print('Data added!!')
def display():
  with open('book.csv','r',newline='') as f:
   reader=csv.reader(f)
   for row in reader:
     print(row)

def update():
  temp_list=[]
  with open('book.csv','r',newline='') as f:
    reader=csv.reader(f)
    no=input('enter the book_no to be updated')
    flag=False
    for row in reader:
      if row[0]==no:
          list1=eval(input('enter the new book details as a list'))
          temp_list.append(list1)
          flag=True
          print('Record updated!!')
        else:
          temp_list.append(row)
  if flag==True:
    with open('book.csv','w',newline='') as f:
      writer=csv.writer(f)
      writer.writerows(temp_list)
  else:
    print('Record not found in our database')

print('''
1.Insert
2.Display
3.Update
4.Exit''')
while True:
  choice=int(input('enter your choice'))
  if choice==1:
    insert()
  elif choice==2:
    display()
  elif choice==3:
    update()
  elif choice==4:
    print('Thank You')
    break
  else:
    print('invalid choice')
```

# 8. CSV-2

To write a menu – driven program to perform the following tasks on a CSV file with book details[Bno,Bname,Bauthor,Bpublisher]

1. Add New Book
2. View Books
3. Delete Book Info
4. Exit

```
import csv
data=[[1,'book1','author1','pub1'],
      [2,'book2','author2','pub2'],
      [3,'book3','author3','pub3'],
      [4,'book4','author4','pub4'],
      [5,'book5','author5','pub5']]

with open('book.csv','w',newline='') as f:
    writer=csv.writer(f)
    writer.writerows(data)

 def insert():
    with open('book.csv','a+',newline='') as f:
        book_data=eval(input('enter the book details as a list'))
        writer=csv.writer(f)
        writer.writerow(book_data)
        print('Data added!!')

def display():
    with open('book.csv','r+',newline='') as f:
      reader=csv.reader(f)
      for row in reader:
          print(row)

def delete():
    temp_list=[]
    with open('book.csv','r+',newline='') as f:
        reader=csv.reader(f)
        no=input('enter the book_no to be deleted')
        flag=False
        for row in reader:
           if row[0]==no:
                flag=True
                print('Record deleted')
                continue

           else:
                    temp_list.append(row)
    if flag==True:
        with open('book.csv','w',newline='') as f:
            writer=csv.writer(f)
            writer.writerows(temp_list)
    else:
        print('Record not found in our database')
print('''
1.Insert
2.Display
3.Delete
4.Exit''')
while True:
    choice=int(input('enter your choice'))
    if choice==1:
        insert()
    elif choice==2:
        display()
    elif choice==3:
        delete()
    elif choice==4:
        print('Thank You....')
        break
    else:
        print('invalid choice')
```

## 9. STACK-1

To write a menu – driven program to perform the following tasks on a STUDENT Stack

STACK OPERATIONS

1.Insert a student detail into stack

2.Remove a student detail from stack

3. Display the contents of stack

4. EXIT

```
def insert(stk):
    roll_no=int(input("enter roll number:"))
    name=input("enter name:")
    total=int(input("Enter total marks"))
    l=[roll_no,name,total]
    stk.append(l)
    top=len(stk)-1
    print("record inserted")
def remove(stk):
    if stk==[]:
        print("underflow---empty stack")
    else:
        item=stk.pop()
        if stk==[]:
            top=None
        else:
            top=len(stk)-1
        print("Removed item",item)
def display(stk):
    if stk==[]:
        print("underflow---empty stack")
    else:
        top=len(stk)-1
        print(stk[top],"<--top")
        for i in range(top-1,-1,-1):
            print(stk[i])

print("""1.Insert a student detail
2.Remove a student detail
3.dispaly the student details
4.Exit""")
stk=[]
top=None
while True:
    c=int(input("enter your choice:"))
    if c==1:
        insert(stk)
    elif c==2:
        remove(stk)
    elif c==3:
        display(stk)
    elif c==4:
        print("Thank You")
        break
    else:
        print("invalid choice")
```

## 10.STACK-2

To write a menu – driven program to perform the following tasks on a CITY Stack

STACK OPERATIONS

1.Insert a City name and Pin code of a city into stack

2.Remove a City name and Pin code of a city from stack

3.Display the contents of stack

4. EXIT

```
def insert(stk):
    city_name=int(input("enter a city name:"))
    pin_code=input("enter a pin code:")
    l=[city_name,pin_code]
    stk.append(l)
    top=len(stk)-1
    print("record inserted")
def remove(stk):
    if stk==[]:
        print("underflow---empty stack")
    else:
        item=stk.pop()
        if stk==[]:
            top=None
        else:
            top=len(stk)-1
        print("Removed item",item)
def display(stk):
    if stk==[]:
        print("underflow---empty stack")
    else:
        top=len(stk)-1
        print(stk[top],"<--top")
        for i in range(top-1,-1,-1):
            print(stk[i])

print("""1.Insert city and pin code detail
2.Remove a city detail
3.dispaly the city details
4.Exit""")
stk=[]
top=None
while True:
    c=int(input("enter your choice:"))
    if c==1:
        insert(stk)
    elif c==2:
        remove(stk)
    elif c==3:
        display(stk)
    elif c==4:
        print("Thank You")
        break
    else:
        print("invalid choice")
```

## 11.Built-in module:

To write a menu – driven program to do the following based on user’s choice.

1. *Use of statistics  module*

2. *Use of random module*
3. *Use of url.request & webbrowser module*
4. *Exit*

```
import statistics
import random
import urllib.request
import webbrowser
def stat():
    L=eval(input("Enter the list"))
    print("Mean is",statistics.mean(L))
    print("Median is",statistics.median(L))
    print("Mode is",statistics.mode(L))
def rand():
    num=random.randint(10,20)
    for i in range(5):
        guess=int(input("Guess the number between 10 - 20:"))
        if guess==num:
            print("You won!!!")
            break
    else:
        print("You lost.The number was",num)
def ur():
    weburl=urllib.request.urlopen("http://google.com")
    url=weburl.geturl()
    print(url)
    webbrowser.open_new(url)
print("""
1.Statistics
2.Random
3.Urllib and webbrowser
4.Exit""")

while True:
    ch=int(input("Enter choice:"))
    if ch==1:
        stat()
    elif ch==2:
        rand()
    elif ch==3:
        ur()
    elif ch==4:
        print("Thank You!!!")
        break
    else:
        print("Wrong choice!!!")
```

## 12.User defined module:

*Write a program to create user defined module “arithmetic.py” .*
*Write a program to enter two numbers and perform the arithmetic operations like +,-,*, /, // and % by importing the above module based on user’s choice.*

```
# arithmetic.py
def add(x,y):
   return x+y
def sub(x,y):
   return x-y
def mul(x,y):
   return x*y
def div(x,y):
   return x/y
def floor(x,y):
   return x//y
def mod(x,y):
   return x%y

# main program
import arithmetic
while True:
   op=input("Enter operator +,-,*,/,//,% ")
   if op=='+':
      num1=int(input("Enter number1:"))
      num2=int(input("Enter number2"))
      result=arithmetic.add(num1,num2)
      print(result)
   elif op=='-':
      num1=int(input("Enter number1:"))
      num2=int(input("Enter number2"))
      result=arithmetic.sub(num1,num2)
      print(result)
   elif op=='*':
      num1=int(input("Enter number1:"))
      num2=int(input("Enter number2"))
      result=arithmetic.mul(num1,num2)
      print(result)
   elif op in ('/','//','%'):
      num1=int(input("Enter number1:"))
      num2=int(input("Enter number2"))
      if num2==0:
         print("Second number cannot be zero ")
      else:
         if op=='/':
            result=arithmetic.div(num1,num2)
            print(result)
         elif op=='//':
            result=arithmetic.floor(num1,num2)
            print(result)
         elif op=='%':
            result=arithmetic.mod(num1,num2)
            print(result)
   else:
      print("Invalid operator!!!")
```

## *PYTHON SQL CONNECTIVITY-1:*

*To write a menu driven program to perform the following task on **"BOARD_EXAM"** database:*

i) *Creating the following table "CARDEN"*

ii) *Inserting the given records*

| Ccode | CarName | Make | Color | Capacity | Charges |
|---|---|---|---|---|---|
| 501 | A-Star | Suzuki | RED | 3 | 14 |
| 503 | Indigo | Tata | SILVER | 3 | 12 |
| 502 | Innova | Toyota | WHITE | 7 | 15 |
| 509 | SX4 | Suzuki | SILVER | 4 | 14 |
| 510 | C Class | Mercedes | RED | 4 | 35 |

iii) Query 1: To display the names of all the silver-coloured Cars

***Select CarName from CARDEN where color="silver";***

iv) Query 2 :To increase charges by 10 where capacity is greater than 3

***Update CARDEN set charges=charges+10 where capacity>3;***

v) Query 3 :To display the CarName ,Capacity in descending order of capacity

***Select CarName, Capacity from CARDEN order by capacity desc;***

## *PYTHON-SQL CONNECTIVITY-2:*

*To write a menu driven program to perform the following task on **"BOARD_EXAM"** database:*

**i)** *Creating the following table "SPORTS"*

**ii)** Inserting the given records

| Stud_no | Class | Name | game | grade |
|---|---|---|---|---|
| 10 | 7 | Sameer | Cricket | B |

| | | | | |
|---|---|---|---|---|
| 11 | 8 | Sujith | Tennis | A |
| 12 | 7 | Kamal | Swimming | B |
| 13 | 7 | Venna | Tennis | C |
| 14 | 9 | Archana | Basketball | A |
| 15 | 10 | Arpit | Cricket | A |

**iii)** Query 1: To display the name of students with game as swimming

***Select name from SPORTS where game=’swimming’ ;***

**iv)** Query 2: To delete the rows where class is 10

***Delete from SPORTS where class=10;***

**v)** Query 3: To display the number of students from each class.

***Select count(*) , class from SPORTS group by class;***

## *PYTHON SQL CONNECTIVITY-3:*

*To write a menu driven program to perform the following task on **“BOARD_EXAM”** database:*

**i)** *Creating the following table “ORDERS”*

**ii)** *Inserting the given records*

| OderID | Pname | Quantity | Rate | Sale_date | Discount |
|---|---|---|---|---|---|
| 1001 | Pen | 10 | 20 | 2019-10-05 | nil |
| 1002 | Pencil | 20 | 10 | 2019-10-21 | nil |
| 1003 | Book | 10 | 100 | 2019-11-12 | 50 |
| 1004 | Eraser | 100 | 5 | 2019-12-05 | 25 |
| 1005 | Copy | 50 | 20 | 2019-12-10 | nil |

**iii)** Query1:To display Pname, Quantity and Rate for all the orders that are either pen or pencil

***Select pname,quantity,rate from ORDERS where Pname in(‘pen’,’pencil’);***

**iv)** Query2:To display orders which are not getting any discount

***Select * from ORDERS where discount IS NULL;***

**v)** Query3: Update Rate by 5%.

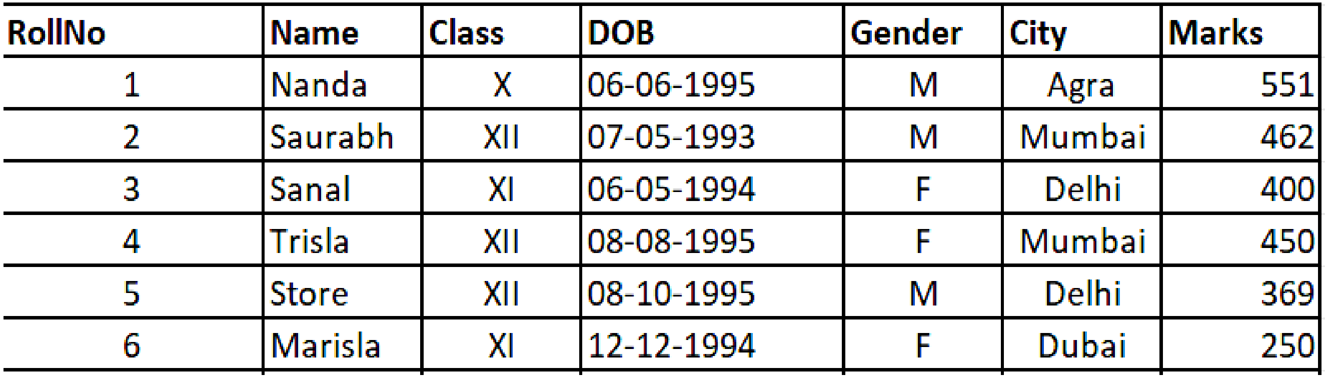
***Update ORDERS set rate= rate + rate * 5/100;***

## PYTHON SQL CONNECTIVITY-4:

*To write a menu driven program to perform the following task on **“BOARD_EXAM”** database:*

**i)** *Creating the following table “STUDENT”*

**ii)** *Inserting the given records*



| RollNo | Name | Class | DOB | Gender | City | Marks |
|---|---|---|---|---|---|---|
| 1 | Nanda | X | 06-06-1995 | M | Agra | 551 |
| 2 | Saurabh | XII | 07-05-1993 | M | Mumbai | 462 |
| 3 | Sanal | XI | 06-05-1994 | F | Delhi | 400 |
| 4 | Trisla | XII | 08-08-1995 | F | Mumbai | 450 |
| 5 | Store | XII | 08-10-1995 | M | Delhi | 369 |
| 6 | Marisla | XI | 12-12-1994 | F | Dubai | 250 |

**iii)** Query 1: To display the records from table student in alphabetical order as per the name of the student.

*Select * from STUDENT order by Name;*

**iv)** Query 2: To display Class, Dob and City whose marks is between 450 and 551.

***Select Name, Class,DOB, City from STUDENT where marks between 450 and451;***

**v)** Query 3:To display the highest marks

***Select Max(marks) from STUDENT;***

## PYTHON SQL CONNECTIVITY-5:

*To write a menu driven program to perform the following task on **"BOARD_EXAM"** database:*

*(i) Creating the following table "EMPLOYEE"*

*(ii) Inserting the given records*

| Fcode | Fname | Sex | Salary | Subject | JoinDate |
|---|---|---|---|---|---|
| F1001 | Hari Charan | M | 101000 | English | 2000-10-11 |
| F1004 | Mallika | F | 202000 | Programming | 2003-11-30 |
| F1002 | Arun | M | 202000 | Web design | 2001-09-06 |
| F1005 | Priya | F | 201000 | Web design | 2004-12-15 |
| F1009 | Tanya | F | 203000 | Programming | 2006-12-31 |
| F1006 | Amar | M | 108000 | Data structure | 2005-05-02 |
| F1008 | Avinash | M | 105000 | Maths | 2002-03-15 |

(iii) Query 1: TodisplayFcode,Fname,JoinDate,Salaryofallfacultymembersin descending order of their salary.

***Select fcode, fname, joindate,salary from EMPLOYEE order by salary desc;***

iv) Query2:Toincreasethesalaryofallfacultiesby2000,whoareeitherteachingthesubject "Programming" or teaching the subject "Maths".

***Update EMPLOYEE set salary=salary + 2000 where subject ='programming' or subject='maths';***

v) Query 3: To display details of all faculty whose name ends with letter "n".

*select * from EMPLOYEE where fname like "%n";*

## PYTHON-SQL CONNECTIVITY-6:

*To write a menu driven program to perform the following task on **"BOARD_EXAM"** database:*

*i) Creating the following table "ITEMS"*

*ii) Inserting the given records*

| ItemNo | Name | Dcode | Qty | UnitPrice | StockDate |
|---|---|---|---|---|---|
| 5005 | CD | 101 | 100 | 25 | 2010-03-31 |
| 5003 | Floppy | 101 | 150 | 10 | 2010-01-01 |
| 5002 | Pendrive | 101 | 125 | 200 | 2010-02-14 |
| 5006 | Mouse | 102 | 200 | 300 | 2009-03-19 |
| 5001 | Keyboard | 102 | 60 | 400 | 2009-12-19 |

(i) Query1: Display the Item name with stockdate in the year 2010.

Select * from ITEMS where year(stockdate)=2010;

(ii) Query2: To display the ItemNo and name of those items from the above table whose Unitprice is more than Rs10.

Select itemno,Name from ITEMS where unitprice>10;

(iii) Query 3: To display the total Qty of each Dcode.

**Select Dcode, sum(qty) from ITEMS group by Dcode;**

## SQL TEMPLATE PROGRAM

```
import mysql.connector as s
mc=s.connect(host='localhost',user='root',passwd='',database='exam')
if mc.is_connected()==False:
   print("Error...")
cr=mc.cursor()

sql='CREATE TABLE if not exists CARDEN (Ccode int primary key,CarName varchar(25),Make varchar(25),Color varchar(10),Capacity int,Charges int,datetime date)'
cr.execute(sql)
mc.commit()
```

```
print('Table created.......')
def insertion():
    ccode=int(input("Enter the ccode;"))
    cname=input("Enter the car name:")
    make=input("Enter make:")
    color=input("Enter color")
    capacity=int(input("Enter capacity"))
    charges=int(input("Enter charges:"))
    dt=input("Enter date")
    sql="INSERT INTO CARDEN VALUES({},'{}','{}','{}','{}',{},'{}' )".format(
ccode,cname,make,color,capacity,charges,dt)
    cr.execute(sql)
    mc.commit()
    print("Record inserted")

def query1():
    sql='SELECT CarName FROM CARDEN  WHERE COLOR="SILVER"'
    cr.execute(sql)
    data=cr.fetchall()
    flag = False
    for i in data:
        print(i)
    if flag == False:
        print("No record found")

def query2():
    sql='select CarName,Make,Capacity from CARDEN where Capacity>5'
    cr.execute(sql)
    data=cr.fetchall()
    flag = False
    for i in data:
        print(i)
        flag = True
    if flag == False:
        print("No record found")

def query3():
    sql='select max(Charges) from CARDEN'
    cr.execute(sql)
    data=cr.fetchall()
    for i in data:
        print(i)
        flag=True
    if flag == False:
        print("No record found")
print('''1.insert record
2.query1
3.query2
4.query3
5.Exit''')
while True:
    ch=int(input('enter your choice '))
    if ch==1:
        insertion()
    elif ch==2:
        query1()
    elif ch==3:
        query2()
    elif ch==4:
        query3()
    elif ch==5:
        break
    else:
        print('invalid choice')
print("Thank You")
```