## SET -1

1. Write a menu driven python program using separate user defined functions to Implement following operations in stack using list

   > To insert the numbers in the stack

   > To delete the numbers from stack.

   > To display the status of stack

2. Consider the following table.

**TABLE: FLIGHTS**

| FL_NO | STARTPLACE | ENDPLACE | NO_FLIGHTS | NO_STOPS |
|---|---|---|---|---|
| IC301 | MUMBAI | DELHI | 8 | 0 |
| IC799 | BANGALORE | DELHI | 2 | 1 |
| MC101 | INDORE | MUMBAI | 3 | 0 |
| IC302 | DELHI | MUMBAI | 8 | 0 |
| AM812 | KANPUR | BANGALORE | 3 | 1 |

A. Write the SQL commands for the following:

i. To display the contents of the table FLIGHTS in the descending order of FL_NO.
ii. To increase the number of flights by one whose ENDPLACE is Delhi
iii. To remove the details of the flights having number of stops as 0.

B. Write the output of the following:

(i) Select ENDPLACE , COUNT(*) from FLIGHTS group by ENDPLACE;
(ii) Select STARTPLACE from FLIGHTS where NO_STOPS=1;

1.

```
def isempty(stack):
    if stack==[]:
        return True
    else :
        return False
def push(stack,item):
    stack.append(item)
    top=len(stack)-1
def pop(stack):
    if isempty(stack):
        return "underflow"
    else:
        item=stack.pop()
        if len(stack)==0:
            top=None
        else:
            top=len(stack)-1
        return item
def peek(stack):
    if isempty(stack):
        return "underflow"
    else:
        top=len(stack)-1
        return(stack[top])
def display(stack):
    if isempty(stack):
        print("Stack empty")
    else:
        top=len(stack)-1
        print(stack[top])
        for a in range (top-1,-1,-1):
            print(stack[a],end=" ")
#---main
st=[]
top=None
while True:
    print("STACK OPERATIONS")
    print("1---push")
    print("2---pop")
    print("3---peek")
    print("4--display stack")
    print("5--exit")
    ch=int(input("enter your choice"))
    if ch==1:
        item=int(input("enter item"))
        push(st,item)
    elif ch==2:
        item=pop(st)
        if item=="underflow":
            print("stack is empty")
        else:
            print("deleted item is",item)
    elif ch==3:
        item=peek(st)
        if item=="underflow":
            print("stack is empty")
        else:
            print("topmost item is",item)
    elif ch==4:
```

```
        display(st)
    elif ch==5:
        break
    else:
        print("invalid choice")'''
```

2.

A.(i) SELECT * FROM FLIGHTS
ORDER BY FL_NO DESC;
(ii) UPDATE FLIGHTS
SET NO_FLIGHTS=N0_FLIGHTS + 1
WHERE ENDPLACE= “DELHI” ;
(iii) DELETE FROM FLIGHTS
WHERE NO_STOPS=1;

B. (i) ENDPLACE COUNT(*)
DELHI 2
MUMBAI 2
BANGALORE 1
(ii) STARTPLACE
BANGALORE
KANPUR

## SET – 2

1. Write a Python program using separate user defined functions to Implement following operations in stack using dictionary

> To create a Dictionary for n number of students with name as a key and marks as value

> To insert the keys (name of the student) of the dictionary into a Stack, where the corresponding value (marks) is greater than 75.

> To display the content of the stack.

2. Consider the following table:

**TABLE: DOCTOR**

| ID | NAME | DEPT | GENDER | EXPERIENCE |
|---|---|---|---|---|
| 101 | John | ENT | M | 12 |
| 104 | Smith | ORTHOPEDIC | M | 5 |
| 107 | George | SKIN | M | 10 |
| 114 | Lara | SKIN | F | 3 |
| 109 | K George | ORTHOPEDIC | F | 9 |

A. Write the SQL commands for the following:

i. To display NAME of all doctors who are in "ORTHOPEDIC" having more than 10 years' experience from the table DOCTOR.
ii. To display all the information of female Doctor.
iii. To remove the details of ENT doctors.

B Write the output of the following:

(i) Select SUM(EXPERIENCE), GENDER from DOCTOR group by GENDER;
(ii) Select Distinct DEPT from DOCTOR;

**#STACK IMPLEMENTATION USING DICTIONARY**

```
def push(S,N):
  S.append(N)
def pop(S):
  if S!=[]:
    return S.pop()
  else:
    return None
def create():
  n=int(input("Enter total number of element"))
  for a in range(n):
    name=input("enter name")
    marks=int(input("Enter the marks "))
    dic[name]=marks
  print("The dictionary created is")
  print(dic)
dic={}
st=[]
create()
for k in dic:
   if dic[k]>=75:
     push(st,k)
print("The output of the above code is:")
```

```
while True:
    if st!=[]:
        print(pop(st), end=" ")
    else:
        break
```

A.(i) SELECT NAME FROM DOCTOR
WHERE DEPT= “ORTHOPEDIC”
AND EXPERIENCE>10;
(ii) SELECT * FROM DOCTOR
WHERE GENDER= “F”;
(iii) DELETE FROM DOCTOR
WHERE DEPT= “ENT”;

B. (i)

| SUM | GENDER |
|---|---|
| 27 | M |
| 12 | F |

(ii)

| DEPT |
|---|
| ENT |
| ORTHOPEDIC |
| SKIN |

## SET – 3

1. Write a Python program using separate user defined functions to Implement following operations in stack using list.

> To input a list of numbers

> To traverse the content of the list and insert even numbers into the stack

> To display the content of the stack.

2. Consider the following table:

**Table: Employees**

| Empid | Firstname | Lastname | Address | City |
|---|---|---|---|---|
| 010 | Ravi | Kumar | Raj nagar | GZB |
| 105 | Harry | Waltor | Gandhi nagar | GZB |
| 152 | Sam | Tones | 33 Elm St. | Paris |
| 215 | Sarah | Ackerman | 440 U.S. 110 | Paris |
| 300 | Robert | Samuel | 9 Fifth Cross | Paris |

A. Write the SQL commands for the following:

(i) To show firstname, lastname, address of all employees, who all are in GZB city.
(ii) To display the content of Employees table in descending order of Firstname.
(iii) To remove the record having Empid as 215 .

B. Write the output of the following:

(i) Select count(*), City from Employees group by City;
(ii) Select Firstname from Employees where Firstname like “S%”;

```
def push(S,N):
   S.append(N)
def pop(S):
   if S!=[]:
      return S.pop()
   else:
      return None
def create():
   print("enter the number of elements in the list :")
   N=int(input())
   i=0
   while i<N:
      print("Enter individual element of the list")
      x=int(input())
      lst.append(x)
      i=i+1
   print("The created list is:")
   print(lst)
lst=[]
st=[]
create()
for k in lst:
  if k%2==0:
     push(st,k)
```

```
print("The output of the above code is:")

while True:

    if st!=[]:

        print(pop(st), end=" ")

    else:

        break
```

**A**. (i) SELECT Firstname, Lastname, Address FROM Employees
WHERE City= “GZB”;
(ii) SELECT * FROM Employees
ORDER BY Firstname DESC;
(iii) DELETE FROM Employees
WHERE Empid=215;

B. (i)

| count(*) | City |
|---|---|
| 2 | GZB |
| 3 | Paris |

(ii) Firstname
Sam
Sarah

## SET – 4

**1.** Create a MySQL-Python connectivity code to create a table MYBOOK to store 5 rows of data. Display the information of those books whose price is greater than 500.
(Create the table in SQL through Python only)

| BNO | NAME | PRICE |
|---|---|---|
| 011 | COMPUTER | 300 |
| 012 | MATHS | 500 |
| 011 | PHYSICS | 600 |
| 012 | ENGLISH | 800 |
| 011 | BIOLOGY | 750 |

2. . Consider the tables PRODUCT and SUPPLIER with following information**.**

**TABLE : PRODUCT**

| P_ID | Manufacture | Price | SN |
|---|---|---|---|
| TP01 | LAK | 40 | S1 |
| FW05 | ABC | 45 | S2 |
| BS01 | ABC | 55 | S3 |
| SH06 | XYZ | 120 | S4 |
| FW12 | XYZ | 95 | S5 |

**TABLE: SUPPLIER**

| SN | PNAME | SNAME | QTY | CITY |
|---|---|---|---|---|
| S1 | BREAD | BRITANIA | 150 | DELHI |
| S2 | CAKE | BRITANIA | 250 | MUMBAI |
| S3 | COFFEE | NESACAFE | 170 | MUMBAI |
| S4 | CHOCOLATE | AMUL | 380 | DELHI |
| S5 | SAUCE | KISSAN | 470 | JAIPUR |

A. Write the SQL commands for the following:

(i) To display all the information of PRODUCT table in descending order of price.

(ii)To increase value of quantity by 100 where city is Mumbai.

(iii)To display Supplier name, Manufacture, Price and product name of the supplier where City is Delhi.

B. Write the output of the following:

(i) Select COUNT(*) ,CITY from SUPPLIER group by CITY;

(ii) Select MAX(PRICE),MIN(PRICE) from PRODUCT;

```
import mysql.connector
#import mysql.connector as mysq
mydb=mysql.connector.connect(host="localhost",user="root",passwd="mysql",database="spsrohini")
mycursor=mydb.cursor()
mycursor.execute("create table if not exists  mybook(bno int ,name char(12),price float )")
query="insert into mybook values( {},'{}',{})".format(011,"COMPUTER",300)
mycursor.execute(query)
query="insert into mybook values( {},'{}',{})".format(012,"MATHS",500)
mycursor.execute(query)
query="insert into mybook values( {},'{}',{})".format(013,"PHYSICS",900)
mycursor.execute(query)
query="insert into mybook values( {},'{}',{})".format(014,"ENGLISH",800)
mycursor.execute(query)
query="insert into mybook values( {},'{}',{})".format(015,"BIOLOGY",750)
mycursor.execute(query)
mydb.commit()
mycursor.execute("select * from mybook where price> 500")
data=mycursor.fetchall()
for x in data:
    print(x)
```

A.(i) SELECT * FROM PRODUCT
ORDER BY  Price DESC;

(ii) UPDATE SUPPLIER
SET QTY=QTY+100
WHERE CITY= “DELHI”

(iii) SELECT SNAME, PNAME, Manufacture, Price
FROM PRODUCT, SUPPLIER
WHERE PRODUCT.SN=SUPPLIER.SN
AND SUPPLIER.CITY= “DELHI”;

B. (i)

| COUNT(*) | CITY |
|---|---|
| 2 | DELHI |
| 2 | MUMBAI |
| 1 | JAIPUR |

(ii)

| MAX(PRICE) | MIN(PRICE) |
|---|---|
| 120 | 40 |