NAIPUNNYA PUBLIC SCHOOL KOCHI

SEAPORT-AIRPORT ROAD,THRIKKAKARA,KOCHI -682021

COMPUTER SCIENCE

PRACTICAL RECORD TERM-2

NAME: ……..

CLASS: ……..

REG NO: …… YEAR: .....

Certified that this is a Bonafide Record of Practical Work

PRINCIPAL **Teacher-in-charge**

Submitted for the Practical Examination held on……..…..….

at……...……

Examiner: Date…….....

INDEX

PRACTICALS TERM 2 – PYTHON QUESTIONS

QUESTION NO.	OBJECTIVE & SOLUTIONS
1.	Write a program to implement all basic operations of stack , such as adding element (PUSH), removing element(POP) and displaying the stack elements(TRAVERSAL) using list.

SOURCE CODE:	stack=[]
	n=int(input('Enter the limit : '))
	
	def PUSH():
	  if isFull():
	  print('Stack overflow.')
	  else:
	  x=eval(input('Enter the element : '))
	  stack.append(x)
	def POP():
	  if isEmpty():
	  print('Stack underflow.')
	  else:
	  stack.pop()
	def TRAVERSAL():
	  if isEmpty():
	  print('Stack underflow')
	  else:

```
        for i in stack[::-1]:
            print(i)
def PEEK():
    if isEmpty():
        print('Stack underflow')
    else:
        print(stack[-1])
def isFull():
    if len(stack)==n:
        return True
    else:
        return False
def isEmpty():
    if len(stack)==0:
        return True
    else:
        return False


while True:
    print('1.Push\n2.Pop\n3.Peek\n4.Traversal\n5.Exit')
    c=int(input('Enter your choice : '))
    if c==1:
        print()
        x=int(input('Enter the limit (max=%s): '%(n-len(stack))))
        for i in range(x):
            PUSH()
```

```
elif c==2:
    POP()
elif c==3:
    print('The last element is :',end=' ')
    PEEK()
elif c==4:
    print('The elements are :')
    TRAVERSAL()
elif c==5:
    break
else:
    print('Invalid entry.')
print()
```

OUTPUT:	Enter the limit : 3 1.Push 2.Pop 3.Peek 4.Traversal 5.Exit Enter your choice : 1 Enter the limit (max=3): 3 Enter the element : 1 Enter the element : 2 Enter the element : 3 1.Push 2.Pop 3.Peek 4.Traversal 5.Exit Enter your choice : 2 1.Push 2.Pop 3.Peek 4.Traversal 5.Exit Enter your choice : 3

	The last element is : 2 1.Push 2.Pop 3.Peek 4.Traversal 5.Exit Enter your choice : 4 The elements are : 2 1 1.Push 2.Pop 3.Peek 4.Traversal 5.Exit Enter your choice : 5 >>>

QUESTION NO.	OBJECTIVE & SOLUTIONS
2.	Write a program to display unique vowels present in the given word using stack.

SOURCE CODE:	def PUSH(x):
	  stack.append(x)
	def TRAVERSAL(x):
	  for i in stack[::-1]:
	  print(i)
	x=input('Enter the word : ')
	v='aeiou'
	stack=[]
	for i in x:
	  if i.lower() in v and i not in stack:
	  PUSH(i)
	TRAVERSAL(stack)

OUTPUT:	1)Enter the word : Computer e u o 2)Enter the word : Source e u

	o 3)Enter the word : Output u

QUESTION NO.	OBJECTIVE & SOLUTIONS
3.	Write a menu based program to perform push and pop operations on a Stack. Each node of the Stack contains the following Member's details as given below: Member id integer Member Name string Age integer

SOURCE CODE:	stack=[]
	n=int(input('Enter the limit : '))
	
	def PUSH():
	  if isFull():
	  print('Stack overflow.')
	  else:
	  l=[]
	  l.append(int(input('Enter the Member Id : ')))
	  l.append(input('Enter the Member name : '))
	  l.append(int(input('Enter the age : ')))
	  stack.append(l)
	def POP():
	  if isEmpty():
	  print('Stack underflow.')
	  else:
	  stack.pop()

```
def TRAVERSAL():
    if isEmpty():
        print('Stack underflow')
    else:
        for i in stack[::-1]:
            print('Member Id : %s\nMember Name : %s\nAge : %s\n'%(i[0],i[1],i[2]))


def PEEK():
    if isEmpty():
        print('Stack underflow')
    else:
        print('Member Id : %s\nMember Name : %s\nAge : %s\n'%(stack[-1][0],stack[-1][1],stack[-1][2]))
def isFull():
    if len(stack)==n:
        return True
    else:
        return False
def isEmpty():
    if len(stack)==0:
        return True
    else:
        return False


while True:
```

```
print('1.Push\n2.Pop\n3.Peek\n4.Traversal\n5.Exit')
c=int(input('Enter your choice : '))
print()
if c==1:
    print()
    if isFull():
        print('Stack overflow.')
        continue
    x=int(input('Enter the limit (max=%s): '%(n-len(stack))))
    for i in range(x):
        if isFull():
            print('Stack overflow.')
            continue
        PUSH()
elif c==2:
    POP()
elif c==3:
    print('The last element is :')
    PEEK()
elif c==4:
    print('The elements are :')
    TRAVERSAL()
elif c==5:
    break
else:
    print('Invalid entry.')
```

	print()
OUTPUT:	Enter the limit : 2 1.Push 2.Pop 3.Peek 4.Traversal 5.Exit Enter your choice : 1 Enter the limit (max=2): 2 Enter the Member Id : 1 Enter the Member name : a Enter the age : 2 Enter the Member Id : 2 Enter the Member name : b Enter the age : 2 1.Push 2.Pop 3.Peek 4.Traversal 5.Exit Enter your choice : 1 Stack overflow. 1.Push 2.Pop

	3.Peek 4.Traversal 5.Exit Enter your choice : 2 1.Push 2.Pop 3.Peek 4.Traversal 5.Exit Enter your choice : 1 Enter the limit (max=1): 1 Enter the Member Id : 2 Enter the Member name : b Enter the age : 3 1.Push 2.Pop 3.Peek 4.Traversal 5.Exit Enter your choice : 3 The last element is : Member Id : 2 Member Name : b Age : 3

	1.Push 2.Pop 3.Peek 4.Traversal 5.Exit Enter your choice : 4 The elements are : Member Id : 2 Member Name : b Age : 3 Member Id : 1 Member Name : a Age : 2 1.Push 2.Pop 3.Peek 4.Traversal 5.Exit Enter your choice : 5 >>>

PRACTICALS TERM 2 – SQL QUESTIONS

<table>
<tr><th>QUESTION NO.</th><th>OBJECTIVE & SOLUTIONS</th></tr>
<tr><td></td><td>
Table : Employee
<table>
<tr><th>EmpNo</th><th>EmpName</th><th>Salary</th><th>Zone</th><th>Age</th><th>Grade</th><th>DeptId</th></tr>
<tr><td>1001</td><td>R Jain</td><td>30000</td><td>West</td><td>28</td><td>A</td><td>10</td></tr>
<tr><td>1002</td><td>H Sinha</td><td>35000</td><td>Centre</td><td>30</td><td>A</td><td>10</td></tr>
<tr><td>1003</td><td>R Bajpayee</td><td>35000</td><td>West</td><td>40</td><td>NULL</td><td>20</td></tr>
<tr><td>1004</td><td>T Prasad</td><td>38000</td><td>North</td><td>38</td><td>C</td><td>30</td></tr>
<tr><td>1005</td><td>S Mahajan</td><td>48000</td><td>East</td><td>26</td><td>NULL</td><td>20</td></tr>
<tr><td>1006</td><td>H Singh</td><td>34000</td><td>South</td><td>28</td><td>B</td><td>10</td></tr>
<tr><td>1007</td><td>Shelly</td><td>36000</td><td>North</td><td>26</td><td>A</td><td>30</td></tr>
<tr><td>1008</td><td>PK Swamy</td><td>18000</td><td>East</td><td>24</td><td>B</td><td>40</td></tr>
<tr><td>1009</td><td>S Tripathi</td><td>26000</td><td>West</td><td>34</td><td>C</td><td>10</td></tr>
<tr><td>1010</td><td>Kamal D</td><td>17000</td><td>North</td><td>32</td><td>NULL</td><td>40</td></tr>
</table>
Table : Department
<table>
<tr><th>Deptid</th><th>DeptName</th><th>MinSal</th><th>MaxSal</th><th>HOD</th></tr>
<tr><td>10</td><td>Sales</td><td>25000</td><td>35000</td><td>1</td></tr>
<tr><td>20</td><td>Finance</td><td>30000</td><td>50000</td><td>5</td></tr>
<tr><td>30</td><td>Admin</td><td>25000</td><td>40000</td><td>7</td></tr>
<tr><td>40</td><td>Marketing</td><td>15000</td><td>20000</td><td>4</td></tr>
</table>
</td></tr>
<tr><td>4.</td><td>Write SQL Commands for questions (1) to (v) based on the tables Employee &
Department
I. Create tables Employee & Department
II. Insert the first record into the tables Employee & Department
III. Display the various department numbersfrom the table Employee. A department</td></tr>
</table>

	number should be displayed only once. IV. Display the employee number, name and salary ofthose employees whose salary is between 35000 and 40000. V. Display the names and salaries of all the employees who are working neither in West zone nor in Centre zone.

SOURCE CODE:	I. create table Employee(EmpNo int, EmpName varchar(20), Salary int, Zone varchar(10), Age int, Grade char(1), DeptId int);
	create table Department(Deptid int, DeptName varchar(15), MinSal int, MaxSal int, HOD int);
	II. insert into Employee values(1001, 'R Jain', 30000, West, 28, 'A', 10);
	insert into Department values(10, 'Sales', 25000, 35000, 1);
	III. select distinct DeptId from Employee;
	IV. select EmpNo, EmpName, Salary from Employee where 35000<Salary<40000;
	V. select EmpName, Salary from Employee where Zone not in ('West','Centre');

OUTPUT:	1)
	mysql> create table Employee(EmpNo int, EmpName varchar(20), Salary int, Zone varchar(10), Age int, Grade char(1), DeptId int);
	Query OK, 0 rows affected (0.02 sec)
	
	mysql> Create table Department(Deptid int, DeptName varchar(15), MinSal int, MaxSal int, HOD int);
	Query OK, 0 rows affected (0.02 sec)

2)

```
mysql> insert into Employee
    -> values
    -> (1001, 'R Jain', 30000, 'West', 28, 'A', 10);
```

```
mysql> insert into Department values(10, 'Sales', 25000, 35000, 1);
Query OK, 1 row affected (0.01 sec)
```

3)

```
+--------+
| DeptId |
+--------+
|     10 |
|     20 |
|     30 |
|     40 |
+--------+
4 rows in set (0.02 sec)
```

4)

```
+-------+------------+--------+
| EmpNo | EmpName    | Salary |
+-------+------------+--------+
|  1001 | R Jain     |  30000 |
|  1002 | H Sinha    |  35000 |
|  1003 | R Bajpayee |  35000 |
|  1004 | T Prasad   |  38000 |
|  1005 | S Mahajan  |  48000 |
|  1006 | H Singh    |  34000 |
|  1007 | Shelly     |  36000 |
|  1008 | PK Swamy   |  18000 |
|  1009 | S Tripathi |  26000 |
|  1010 | Kamal D    |  17000 |
+-------+------------+--------+
10 rows in set (0.02 sec)
```

5)

```
+------------+--------+
| EmpName    | Salary |
+------------+--------+
| R Jain     |  30000 |
| H Sinha    |  35000 |
| R Bajpayee |  35000 |
| T Prasad   |  38000 |
| S Mahajan  |  48000 |
| H Singh    |  34000 |
| Shelly     |  36000 |
| PK Swamy   |  18000 |
| S Tripathi |  26000 |
| Kamal D    |  17000 |
+------------+--------+
10 rows in set (0.00 sec)
```

QUESTION NO.	OBJECTIVE & SOLUTIONS
5.	Write SQL Commands for questions (1) to (viii) based on the tables Employee & Department. I. To get the name of the column Deptid to D_id. II. Display the name of those employees whose names starts with 'H'. III. List the name of employees not having any Grade. IV. Display the list of employees in descending order of employee code. V. Find the average salary at each department. VI. Find maximum salary of each department and display the name of that department which has maximum salary more than 37000. VII. To delete the records whose grade is not entered. VIII. Display the name and salary of those employees whose grade is A and from the sales department after incrementing by10%
SOURCE CODE:	I. alter table Department change column Deptid D_id int; alter table Employee change column Deptid D_id int; II. select EmpName from Employee where EmpName like 'H%'; III. select EmpName from Employee where Grade=Null; IV. select * from Employee order by EmpNo desc; V. select avg(Salary) from Employee group by D_id; VI. select max(Salary), DeptName from Employee, Department where Employee.D_id=Department.D_id group by DeptName; VII. delete from Employee where Grade is Null; VIII. select EmpName, Salary*1.1 as Salary from Employee where D_id=10;

OUTPUT:

I.

```
mysql> alter table Department change column Deptid D_id int;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
mysql> alter table Employee change column Deptid D_id int;
Query OK, 0 rows affected (0.00 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

II.

```
+---------+
| EmpName |
+---------+
| H Sinha |
| H Singh |
+---------+
2 rows in set (0.03 sec)
```

III.

```
+------------+
| EmpName    |
+------------+
| R Bajpayee |
| S Mahajan  |
| Kamal D    |
+------------+
3 rows in set (0.00 sec)
```

IV.

```
+-------+-----------+--------+--------+------+-------+--------+
| EmpNo | EmpName   | Salary | Zone   | Age  | Grade | DeptId |
+-------+-----------+--------+--------+------+-------+--------+
|  1010 | Kamal D   |  17000 | North  |   32 | NULL  |     40 |
|  1009 | S Tripathi|  26000 | West   |   34 | C     |     10 |
|  1008 | PK Swamy  |  18000 | East   |   24 | B     |     40 |
|  1007 | Shelly    |  36000 | North  |   26 | A     |     30 |
|  1006 | H Singh   |  34000 | South  |   28 | B     |     10 |
|  1005 | S Mahajan |  48000 | East   |   26 | NULL  |     20 |
|  1004 | T Prasad  |  38000 | North  |   38 | C     |     30 |
|  1003 | R Bajpayee|  35000 | West   |   40 | NULL  |     20 |
|  1002 | H Sinha   |  35000 | Centre |   30 | A     |     10 |
|  1001 | R Jain    |  30000 | West   |   28 | A     |     10 |
+-------+-----------+--------+--------+------+-------+--------+
10 rows in set (0.02 sec)
```

V.

```
+-------------+
| avg(Salary) |
+-------------+
|  31250.0000 |
|  41500.0000 |
|  37000.0000 |
|  17500.0000 |
+-------------+
4 rows in set (0.02 sec)
```

VI.

```
+-------------+-----------+
| max(Salary) | DeptName  |
+-------------+-----------+
|       38000 | Admin     |
|       48000 | Finance   |
|       18000 | Marketing |
|       35000 | Sales     |
+-------------+-----------+
4 rows in set (0.02 sec)
```

VII.

```
mysql> delete from Employee where Grade is Null;
Query OK, 3 rows affected (0.01 sec)

mysql> select * from Employee;
+-------+-----------+--------+--------+------+-------+------+
| EmpNo | EmpName   | Salary | Zone   | Age  | Grade | D_id |
+-------+-----------+--------+--------+------+-------+------+
|  1001 | R Jain    |  30000 | West   |   28 | A     |   10 |
|  1002 | H Sinha   |  35000 | Centre |   30 | A     |   10 |
|  1004 | T Prasad  |  38000 | North  |   38 | C     |   30 |
|  1006 | H Singh   |  34000 | South  |   28 | B     |   10 |
|  1007 | Shelly    |  36000 | North  |   26 | A     |   30 |
|  1008 | PK Swamy  |  18000 | East   |   24 | B     |   40 |
|  1009 | S Tripathi|  26000 | West   |   34 | C     |   10 |
+-------+-----------+--------+--------+------+-------+------+
7 rows in set (0.00 sec)
```

VIII.

```
+------------+---------+
| EmpName    | Salary  |
+------------+---------+
| R Jain     | 33000.0 |
| H Sinha    | 38500.0 |
| H Singh    | 37400.0 |
| S Tripathi | 28600.0 |
+------------+---------+
4 rows in set (0.01 sec)
```

QUESTION NO.	OBJECTIVE & SOLUTIONS
6.	Write SQL Commands for questions 1 to 5 based on the table TEACHER

NO	NAME	AGE	DEPARTMENT	DOJ	SALARY	SEX
1	Mohitesh K	34	Computer	01/10/97	12000	M
2	Jaya Priya	31	History	24/03/98	18000	F
3	Prachi S	32	Mathematics	12/12/98	30000	M
4	Mishra A	35	History	07/01/99	40000	F
5	Maurya T	24	Mathematics	08/05/97	25000	M

1) To show all information about the teachers whose salary is greater than 20000.

2) To list all female teachers who are from History department.

3) To list all names of all teachers beginning with 'M' sorted by Name in descending order.

4) To count number of teachers with age less than 32.

5) To display the maximum salary .

SOURCE CODE:

1) Select * from TEACHER where SALARY>20000;
2) Select NAME from TEACHER where SEX='F';
3) Select NAME from TEACHER where NAME like 'M%' order by NAME desc;
4) Select Count(*) from TEACHER where AGE>32;
5) Select max(SALARY) from TEACHER;

OUTPUT:

1)

```
+------+----------+------+------------+------------+--------+------+
| NO   | NAME     | AGE  | DEPARTMENT | DOJ        | SALARY | SEX  |
+------+----------+------+------------+------------+--------+------+
|    3 | Prachi S |   32 | Mathematics | 0000-00-00 |  30000 | M    |
|    4 | Mishra A |   35 | History     | 0000-00-00 |  40000 | F    |
|    5 | Maurya T |   24 | Mathematics | 0000-00-00 |  40000 | F    |
+------+----------+------+------------+------------+--------+------+
3 rows in set (0.00 sec)
```

2)

```
+------------+
| NAME       |
+------------+
| Jaya Priya |
| Mishra A   |
| Maurya T   |
+------------+
3 rows in set (0.00 sec)
```

3)

```
+------------+
| NAME       |
+------------+
| Mohitesh K |
| Mishra A   |
| Maurya T   |
+------------+
3 rows in set (0.00 sec)
```

4)

```
+----------+
| Count(*) |
+----------+
|        2 |
+----------+
1 row in set (0.01 sec)
```

5)

```
+-------------+
| max(SALARY) |
+-------------+
|       40000 |
+-------------+
1 row in set (0.00 sec)
```

<table>
<tr><th>QUESTION NO.</th><th>OBJECTIVE & SOLUTIONS</th></tr>
<tr><td>7.</td><td>Write SQL Commands for questions 1 to 3 on the basis of table ADMIN and give the output for queries 4 and 5.

<table>
<tr><td>CODE</td><td>TNAME</td><td>SUBJECT</td></tr>
<tr><td>1001</td><td>Ravi Shankar</td><td>English</td></tr>
<tr><td>1009</td><td>Priya Rai</td><td>Physics</td></tr>
<tr><td>1203</td><td>Lisa Anand</td><td>English</td></tr>
<tr><td>1309</td><td>Anita Rai</td><td>Hindi</td></tr>
<tr><td>1400</td><td>George R</td><td>Hindi</td></tr>
</table>

1) To alter the table to add new column EXPERIENCE.

2) To update table ADMIN by giving all staff 10 yrs experience.

3) To display the records in the descending order of staff name .

4) To display the number of staff names beginning with letter ‘R’.

5) To display the number of teachers in each subject</td></tr>
<tr><td>SOURCE CODE:</td><td>1) Alter table ADMIN add EXPERIENCE int;
2) Update ADMIN set EXPERIENCE=10;
3) Select * from ADMIN order by TNAME;
4) Select count(TNAME) from ADMIN where TNAME like ‘R%’;
5) Select count(*), SUBJECT from ADMIN group by SUBJECT;</td></tr>
</table>

OUTPUT:

1)

```
mysql> Alter table ADMIN add EXPERIENCE int;
Query OK, 0 rows affected (0.13 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from ADMIN;
+------+-------------+---------+------------+
| CODE | TNAME       | SUBJECT | EXPERIENCE |
+------+-------------+---------+------------+
| 1001 | Ravi Shankar | English |       NULL |
| 1009 | Priya Rai    | Physics |       NULL |
| 1203 | Lisa Anand   | English |       NULL |
| 1309 | Anita Rai    | Hindi   |       NULL |
| 1400 | George R     | Hindi   |       NULL |
+------+-------------+---------+------------+
5 rows in set (0.00 sec)
```

2)

```
mysql> update ADMIN set EXPERIENCE=10;
Query OK, 5 rows affected (0.02 sec)
Rows matched: 5  Changed: 5  Warnings: 0

mysql> select * from ADMIN;
+------+-------------+---------+------------+
| CODE | TNAME       | SUBJECT | EXPERIENCE |
+------+-------------+---------+------------+
| 1001 | Ravi Shankar | English |         10 |
| 1009 | Priya Rai    | Physics |         10 |
| 1203 | Lisa Anand   | English |         10 |
| 1309 | Anita Rai    | Hindi   |         10 |
| 1400 | George R     | Hindi   |         10 |
+------+-------------+---------+------------+
5 rows in set (0.00 sec)
```

3)

```
mysql> Select * from ADMIN order by TNAME;
+------+-------------+---------+------------+
| CODE | TNAME       | SUBJECT | EXPERIENCE |
+------+-------------+---------+------------+
| 1309 | Anita Rai    | Hindi   |         10 |
| 1400 | George R     | Hindi   |         10 |
| 1203 | Lisa Anand   | English |         10 |
| 1009 | Priya Rai    | Physics |         10 |
| 1001 | Ravi Shankar | English |         10 |
+------+-------------+---------+------------+
5 rows in set (0.00 sec)
```

4)

```
+--------------+
| count(TNAME) |
+--------------+
|            1 |
+--------------+
1 row in set (0.01 sec)
```

5)

```
+----------+---------+
| count(*) | SUBJECT |
+----------+---------+
|        2 | English |
|        2 | Hindi   |
|        1 | Physics |
+----------+---------+
3 rows in set (0.00 sec)
```

QUESTION NO.	OBJECTIVE & SOLUTIONS
8.	Write queries for (i) to (v) based on the table STUDENT

ST_ID	ST_CODE	ST_NAME	SUBJECT	MARKS
1	101	Andrew	English	68
2	102	Neha	Physics	70
3	103	John	Maths	90
4	104	Arjun	Science	85
5	105	Joseph	English	92
6	106	Prithvi	Maths	83
7	107	Nehla	English	85
8	108	Arun	Science	70

1) To get the SUBJECT and the average marks scored by the students in that subject from the table STUDENT.

2) To change the name of the column ST_CODE to ADMN_NO.

3) To get the student names sorted by marks in the descending order.

4) To get the number of students who secured more than 80% marks from the table student.

5) To get the student names that start with "a" and are at least 5 characters in length

SOURCE CODE:	1) Select SUBJECT, avg(MARKS) from STUDENT group by SUBJECT; 2) Alter table STUDENT change ST_CODE ADMN_NO int; 3) Select ST_NAME from STUDENT order by MARKS desc; 4) Select count(*) from STUDENT where MARKS>80; 5) Select ST_NAME from STUDENT where ST_NAME like 'A%' and length(ST_NAME)>4;
OUTPUT:	1) +---------+------------+ | SUBJECT | avg(MARKS) | +---------+------------+ | English | 81.6667 | | Maths | 86.5000 | | Physics | 70.0000 | | Science | 77.5000 | +---------+------------+ 4 rows in set (0.00 sec) 2) mysql> Alter table STUDENT change ST_CODE ADMN_NO int; Query OK, 0 rows affected (0.01 sec) Records: 0 Duplicates: 0 Warnings: 0 mysql> desc STUDENT; +---------+-------------+------+-----+---------+-------+ | Field | Type | Null | Key | Default | Extra | +---------+-------------+------+-----+---------+-------+ | ST_ID | int(11) | YES | | NULL | | | ADMN_NO | int(11) | YES | | NULL | | | ST_NAME | varchar(20) | YES | | NULL | | | SUBJECT | varchar(10) | YES | | NULL | | | MARKS | int(11) | YES | | NULL | | +---------+-------------+------+-----+---------+-------+ 5 rows in set (0.00 sec)

3)

```
+---------+
| ST_NAME |
+---------+
| Joseph  |
| John    |
| Arjun   |
| Nehla   |
| Prithvi |
| Neha    |
| Arun    |
| Andrew  |
+---------+
8 rows in set (0.00 sec)
```

4)

```
+----------+
| count(*) |
+----------+
|        5 |
+----------+
1 row in set (0.00 sec)
```

5)

```
+---------+
| ST_NAME |
+---------+
| Andrew  |
| Arjun   |
+---------+
2 rows in set (0.03 sec)
```

PRACTICALS TERM 2 – SQL INTERFACING WITH PYTHON

QUESTION NO.	OBJECTIVE & SOLUTIONS
9	Write a program to connect Python with MySQL using database connectivity and perform the following operations on data in database BookShop: Fetch, Update and delete the data. A) CREATE A TABLE B) INSERT THE DATA C) UPDATE THE RECORD D) DELETE THE DATA

SOURCE CODE:	import mysql.connector
	mydb=mysql.connector.connect(
	host='localhost',
	user='root',
	password='1125899839733759',
	database='bookshop')
	cursor=mydb.cursor()
	
	#creating the table
	cursor.execute('create table book(bookno int, bookname varchar(20), price float, author varchar(20),publisher varchar(20))')
	
	while True:
	c=int(input('1. Insert data\n2. Update a record.\n3. Delete a record\n4. Exit\nEnter your choice (1,2,3,4) : '))
	print()
	if c==1:
	l=()
	l+=(int(input('Enter the book number : '))),
	a=input('Enter the name of the book : ')
	a=""+a+""

```
        l+=(a),
        l+=(eval(input('Enter the price : '))),
        a=input('Enter the name of the author : ')
        a=""+a+""
        l+=(a),
        a=input('Enter the publisher : ')
        a=""+a+""
        l+=(a),
        cursor.execute('insert into book
values(%s,%s,%s,%s,%s)'%(l))
        mydb.commit()
        print('\nRecord inserted.')
    elif c==2:
        l=[]
        n=int(input('Enter the book number of the record to be
updated : '))
        cursor.execute('select bookno from book')
        for i in cursor:
            for j in i:
                l.append(j)
        if n in l:
            a=int(input('1. Bookname\n2. Price\n3. Author\n4.
Publisher\nEnter the value to be updated : '))
            if a==1:
                nn=int(input('Enter the new bookname : '))
                cursor.execute('update book set bookname=%s where
bookno=%s'%(nn,n))
                mydb.commit()
            elif a==2:
                np=eval(input('Enter the new price of the book : '))
                cursor.execute('update book set price=%s where
bookno=%s'%(np,n))
                mydb.commit()
            elif a==3:
                na=input('Enter the author : ')
                cursor.execute('update book set author=%s where
bookno=%s'%(na,n))
                mydb.commit()
            elif a==4:
                np=int(input('Enter the publisher : '))
                cursor.execute('update book set publisher=%s where
bookno=%s'%(np,n))
                mydb.commit()
```

```
            else:
                print('Invalid entry.')
                break
        else:
            print('Record not found.')

    elif c==3:
        l=[]
        n=int(input('Enter the book number of the record to be
deleted : '))
        cursor.execute('select bookno from book')
        for i in cursor:
            for j in i:
                l.append(j)
        if n in l:
            cursor.execute('delete from book where
bookno=%s'%(n))
            mydb.commit()
            print('Record deleted.')
        else:
            print('Record not found.')

    elif c==4:
        break

    else:
        print('Invalid entry.')
    print()
```

OUTPUT:	Output in python: 1. Insert data 2. Update a record. 3. Delete a record 4. Exit Enter your choice (1,2,3,4) : 1 Enter the book number : 1 Enter the name of the book : a Enter the price : 2 Enter the name of the author : a Enter the publisher : a Record inserted. 1. Insert data 2. Update a record. 3. Delete a record 4. Exit Enter your choice (1,2,3,4) : 1 Enter the book number : 2 Enter the name of the book : b Enter the price : 4 Enter the name of the author : b Enter the publisher : b

	Record inserted. 1. Insert data 2. Update a record. 3. Delete a record 4. Exit Enter your choice (1,2,3,4) : 2 Enter the book number of the record to be updated : 1 1. Bookname 2. Price 3. Author 4. Publisher Enter the value to be updated : 2 Enter the new price of the book : 1 1. Insert data 2. Update a record. 3. Delete a record 4. Exit Enter your choice (1,2,3,4) : 3 Enter the book number of the record to be deleted : 2 Record deleted.

1. Insert data

2. Update a record.

3. Delete a record

4. Exit

Enter your choice (1,2,3,4) : 4

>>>

Effects on database bookshop:

Table created:

```
mysql> show tables;
+--------------------+
| Tables_in_bookshop |
+--------------------+
| book               |
+--------------------+
1 row in set (0.00 sec)

mysql> desc book;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| bookno    | int(11)     | YES  |     | NULL    |       |
| bookname  | varchar(20) | YES  |     | NULL    |       |
| price     | float       | YES  |     | NULL    |       |
| author    | varchar(20) | YES  |     | NULL    |       |
| publisher | varchar(20) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

2 Records inserted :

```
mysql> select * from book;
+--------+----------+-------+--------+-----------+
| bookno | bookname | price | author | publisher |
+--------+----------+-------+--------+-----------+
|      1 | a        |     2 | a      | a         |
|      2 | b        |     4 | b      | b         |
+--------+----------+-------+--------+-----------+
2 rows in set (0.00 sec)
```

1st record updated :

```
mysql> select * from book;
+--------+----------+-------+--------+-----------+
| bookno | bookname | price | author | publisher |
+--------+----------+-------+--------+-----------+
|      1 | a        |     1 | a      | a         |
|      2 | b        |     4 | b      | b         |
+--------+----------+-------+--------+-----------+
2 rows in set (0.00 sec)
```

2nd record deleted :

```
mysql> select * from book;
+--------+----------+-------+--------+-----------+
| bookno | bookname | price | author | publisher |
+--------+----------+-------+--------+-----------+
|      1 | a        |     1 | a      | a         |
+--------+----------+-------+--------+-----------+
1 row in set (0.00 sec)
```