



पुर्णमा International School

Shree Swaminarayan Gurukul, Zundal

GRADE : XII

COMPUTER SCIENCE

PRACTICAL FILE

Index

No.	Practical Name	Date	Signature
1	Program to read and display file content line by line with each word separated by “ #”		
2	Program to read the content of file and display the total number of consonants, uppercase, vowels and lower case characters.		
3	Program to read the content of file line by line and write it to another file except for the lines contains „a” letter in it.		
4	Program to create binary file to store Rollno and Name, Search any Rollno and display name if Rollno found otherwise “Rollno not found”		
5	Program to create binary file to store Rollno, Name and Marks and update marks of entered Rollno.		
6	Program to generate random number 1-6, simulating a dice.		
7	Program to implement Stack in Python using List.		
8	Create a CSV file by entering user-id and password, read and search the password for given user- id.		
9	Consider the following MOVIE table and write the SQL queries based on it.		
10	Write following queries.		
11	Consider the following matches, teams table and write SQL queries based on it.		
12	Consider the following table and write the queries:		
13	Perform create database and Write a MySQL connectivity program in Python.		
14	Perform all the operations with reference to table ‘students through MySQL-Python connectivity.		

Program 1: Program to read and display file content line by line with each word separated by “ #”

```
#Program to read content of file line by  
line#and display each word separated by  
'#'
```

```
f = open("file1.txt")  
for line in f:  
    words = line.split()  
    for w in words:  
        print(w+'#',end=" ")  
    print()  
f.close()
```

NOTE : if the original content of file is:

India is my
countryI love
python
Python learning is fun

OUTPUT

```
India#is#my#country#  
I#love#python#  
Python#learning#is#fun  
#
```

Program 2: Program to read the content of file and display the total number of consonants, uppercase, vowels and lower case characters.

#Program to read content of file
#and display total number of vowels, consonants, lowercase and uppercase characters

```
f = open("file1.txt")
v=0
c=0
u=0
l=0
o=0
data = f.read()
vowels=['a','e','i','o','u']
for ch in data:
    if ch.isalpha():
        if ch.lower() in vowels:
            v+=1
        else:
            c+=1
    if ch.isupper():
        u+=1
    elif ch.islower():
        l+=1
    elif ch!=' ' and ch!='\n':
        o+=1
print("Total Vowels in file      :",v)
print("Total Consonants in file   :",c)
print("Total Capital letters in file :",u)
print("Total Small letters in file   :",l)
print("Total Other than letters      :",o)
f.close()
```

NOTE : if the original content of file is:

India is my countryI
love python
Python learning is fun123@

OUTPUT

```
Total Vowels in file      : 16
Total Consonants in file   : 30
Total Capital letters in   : 2
file
Total Small letters in file : 44
Total Other than letters   : 4
```

Program 3: Program to read the content of file line by line and write it to another file except for the lines contains "a" letter in it.

#Program to read line from file and write it to another line#Except
for those line which contains letter 'a'

```
f1 = open("file2.txt")
f2 = open("file2copy.txt","w")

for line in f1:
    if 'a' not in line:
        f2.write(line)
print("## File Copied Successfully! ##")
f1.close()
f2.close()
```

NOTE: Content of file2.txt

a quick brown fox
one two three four
five six seven
India is my country
eight nine ten
bye!

OUTPUT

File Copied Successfully!

NOTE: After copy content of file2copy.txt

one two three four
five six seven eight
nine ten bye!

यु.टी.

Program 4: Program to create binary file to store Rollno and Name, Searchany Rollno and display name if Rollno found otherwise “Rollno not found”

```
import pickle
def write():
D={}
f=open("Studentdetails.dat","wb")
while True:
    r = int(input ("Enter Roll no : "))
    n = input("Enter Name : ")
    D['Roll No'] = r
    D['Name'] = n
    pickle.dump(D,f)
    ch = input("More ? (Y/N)")
    if ch in 'Nn':
        break
    f.close()

def Search() :
    found = 0
    rollno= int(input("Enter Roll no Whose name you want to display :"))
    f = open("Studentdetails.dat", "rb")
    try:
        while True:
            rec = pickle.load(f)
            if rec['Roll No']==rollno:
                print(rec['Name'])
                found = 1
                break
    except EOFError:
        if found == 0:
            print("Sorry not Found....")
            f.close()

write()
Search()
```

OUTPUT

```
Enter Roll no : 1
Enter Name : jayna
More ? (Y/N)y
Enter Roll no : 2
Enter Name : Daksh
More ? (Y/N)y
Enter Roll no : 3
Enter Name : vyom
More ? (Y/N)N
Enter Roll no Whose name you want to
display :3
vyom
```

Program 5: Program to create binary file to store Rollno, Name and Marks and update marks of entered Rollno.

#Program to create a binary file to store Rollno and name
#Search for Rollno and display record if found #otherwise
"Roll no. not found"

```
import pickle
def Write():
    f = open("Studentdetails.dat", 'wb')
    while True:
        r = int(input ("Enter Roll no : "))
        n = input("Enter Name : ")
        m = int(input ("Enter Marks : "))
        record = [r,n,m]
        pickle.dump(record, f)
        ch = input("Do you want to enter more?(Y/N)")
        if ch in 'Nn':
            break

    f.close()

def Read():
    f = open("Studentdetails.dat", 'rb')
    try:
        while True:
            rec=pickle.load(f)
            print(rec)
    except EOFError:
        f.close()

def Update():
    f = open("Studentdetails.dat", 'rb+')
    rollno = int(input("Enter roll no whose marks you want to update"))
    try:
        while True:
            pos=f.tell()
            rec = pickle.load(f)
            if rec[0]==rollno:
                um = int(input("Enter Update Marks:"))
                rec[2]=um
                f.seek(pos)
                pickle.dump(rec,f)
                #print(rec)
    except EOFError:
        f.close()
```

Write()
Read()
Update()
Read()

OUTPUT

Enter Roll no : 1

Enter Name : jiya

Enter Marks : 56

Do you want to enter more ?(Y/N)y

Enter Roll no : 2

Enter Name : henil

Enter Marks : 67

Do you want to enter more ?(Y/N)n

[1, 'jiya', 56]

[2, 'henil', 67]

Enter roll no whose marks you want to update : 2

Enter Update Marks:76

[1, 'jiya', 56]

[2, 'henil', 76]

यु.जो

Program 6: Program to generate random number 1-6, simulating a dice.

```
# Program to generate random number between 1 - 6# To
simulate the dice
import random
while True:
    print("="*55)
    print("*****Roling Dice*****")
    print("="*55)
    num = random.randint(1,6)
    if num ==6:
        print("Hey.....You got",num,".....Congratulations!!!!")
    elif num ==1:
        print("Well tried.... But you got",num)
    else:
        print("You got:",num)
        ch=input("Roll again? (Y/N)")
        if ch in "Nn":
            break
        print("Thank for playing!!!!!!!!")
```

OUTPUT

```
=====
*****Roling Dice*****
=====
You got: 2
Roll again? (Y/N)y
Thank for playing!!!!!!!!
=====
*****Roling Dice*****
=====
Hey.....You got 6 .....Congratulations!!!!
=====
*****Roling Dice*****
=====
Well tried.... But you got 1
=====
*****Roling Dice*****
=====
Well tried.... But you got 1
=====
*****Roling Dice*****
=====
Well tried.... But you got 1
=====
*****Roling Dice*****
=====
You got: 2
Roll again? (Y/N)n
```


Program 7: Write a program to implement a stack for the employee details (empno,name).Code:

#stack implementation using functions

#program to create a stack of employee(empno,name,sal).

```
employee=[]
def push():
    empno=input("Enter empno ")
    name=input("Enter name ")
    sal=input("Enter sal ")
    emp=(empno,name,sal)
    employee.append(emp)
def pop():
    if(employee==[]):
        print("Underflow / Employee Stack in empty")
    else:
        empno,name,sal=employee.pop()
        print("poped element is ")
        print("empno ",empno," name ",name," salary ",sal)
def traverse():
    if not (employee==[]):
        n=len(employee)
        for i in range(n-1,-1,-1):
            print(employee[i])
    else:
        print("Empty , No employee to display")
while True:
    print("1. Push")
    print("2. Pop")
    print("3. Traversal")
    print("4. Exit")
    ch=int(input("Enter your choice "))
    if(ch==1):
        push()
    elif(ch==2):
        pop()
    elif(ch==3):
        traverse()
    elif(ch==4):
        print("End")
        break
    else:
        print("Invalid choice")
```

Output:

=====RESTART: D:/python/Grade 12/employee pra2.py=====

1. Push

2. Pop

3. Traversal

4. Exit

Enter your choice 1

Enter empno 12

Enter name jayna

Enter sal 30000

1. Push

2. Pop

3. Traversal

4. Exit

Enter your choice 3

('12', 'jayna', '30000')

1. Push

2. Pop

3. Traversal

4. Exit

यु.जे.टी

संस्कृत विश्वविद्यालय

Program 8: Create a CSV file by entering user-id and password, read and search the password for given user- id.

```
import csv
with open("7.csv", "w") as obj:
    fileobj = csv.writer(obj)
    fileobj.writerow(["User Id", "password"])
    while(True):
        user_id = input("enter id: ")
        password = input("enter password: ")
        record = [user_id, password]
        fileobj.writerow(record)
        x = input("press Y/y to continue and N/n to terminate the program\n")
        if x in "Nn":
            break
        elif x in "Yy":
            continue
with open("7.csv", "r") as obj2:
    fileobj2 = csv.reader(obj2)
    given = input("enter the user id to be searched\n")

    for i in fileobj2:
        next(fileobj2)

        # print(i,given)
        if i[0] == given:
            print(i[1])

break
```


SQL
QUERIES
Practical

यु.टी.

Program 9 : Consider the following MOVIE table and write the SQL queries based on it.

Movie_ID	MovieName	Type	ReleaseDate	Production Cost	BusinessCost
M001	The Kashmir Files	Action	2022/01/26	1245000	1300000
M002	Attack	Action	2022/01/28	1120000	1250000
M003	Loop Lapeta	Thriller	2022/02/01	250000	300000
M004	Badhai Do	Drama	2022/02/04	720000	68000
M005	Shabaash Mithu	Biography	2022/02/04	1000000	800000
M006	Gehraiyaan	Romance	2022/02/11	150000	120000

- Display all information from movie.
- Display the type of movies.
- Display movieid, moviename, total_earning by showing the business done by the movies. Calculate the business done by movie using the sum of productioncost and businesscost.
- Display movieid, moviename and productioncost for all movies with productioncost greater than 150000 and less than 1000000.
- Display the movie of type action and romance.
- Display the list of movies which are going to release in February, 2022.

Answers:

- select * from movie;

```
mysql> select * from movie;
```

Movie_id	moviename	type	releasedate	productioncost	businesscost
M001	The Kashmir Files	Action	2022-01-26	1245000	1300000
M002	Attack	Action	2022-01-28	1120000	1250000
M003	Loop Lapeta	Thriller	2022-02-01	250000	300000
M004	Bdhai Do	Drama	2022-02-04	720000	4800000
M005	Shabaash Mothu	Biography	2022-02-04	1000000	800000
M006	Gehriyaan	Romance	2022-02-11	150000	120000

6 rows in set (0.01 sec)

- select distinct from a movie;

```
mysql> select distinct type from movie;
```

type
Action
Thriller
Drama
Biography
Romance

```
5 rows in set (0.00 sec)
```

- c) select movieid, moviename, productioncost + businesscost "total earning" from movie;

```
mysql> select movie_id, moviename, productioncost + businesscost "total earning" from movie;
```

movie_id	moviename	total earning
M001	The Kashmir Files	2545000
M002	Attack	2370000
M003	Loop Lapeta	550000
M004	Bdhai Do	5520000
M005	Shabaash Mothu	1800000
M006	Gehriyaan	270000

```
6 rows in set (0.03 sec)
```

- d) select movie_id, moviename, productioncost from movie where productcost is >150000 and <1000000;

```
mysql> select movie_id, moviename, productioncost from movie where productioncost >150000 and productioncost <1000000;
```

movie_id	moviename	productioncost
M003	Loop Lapeta	250000
M004	Bdhai Do	720000

```
2 rows in set (0.03 sec)
```

- e) select moviename from movie where type = 'action' or type = 'romance';

```
mysql> select moviename from movie where type = 'action' or type = 'romance';
```

moviename
The Kashmir Files
Attack
Gehriyaan

```
3 rows in set (0.00 sec)
```

- f) select moviename from movie where month(releasedate)=2;

```
mysql> select moviename from movie where month(releasedate)=2;
```

moviename
Loop Lapeta
Bdhai Do
Shabaash Mothu
Gehriyaan

```
4 rows in set (0.00 sec)
```

Program 10: Write following queries.

- a. Write a query to display cube of 5.
- b. Write a query to display the number 563.854741 rounding off to the next hundred.
- c. Write a query to display "put" from the word "Computer".
- d. Write a query to display today's date into DD.MM.YYYY format.
- e. Write a query to display 'DIA' from the word "MEDIA".
- f. Write a query to display moviename - type from the table movie.
- g. Write a query to display first four digits of productioncost.
- h. Write a query to display last four digits of businesscost.
- i. Write a query to display weekday of release dates.
- j. Write a query to display dayname on which movies are going to be released.

Answers:

a) `select pow(5,3);`

```
mysql> select pow(5,3);
+-----+
| pow(5,3) |
+-----+
|      125 |
+-----+
1 row in set (0.01 sec)
```

b) `select round(563.854741,-2);`

```
mysql> select round(563.854741,-2);
+-----+
| round(563.854741,-2) |
+-----+
|              600 |
+-----+
1 row in set (0.00 sec)
```

c) `select mid("Computer",4,3);`

```
mysql> select mid("Computer",4,3);
+-----+
| mid("Computer",4,3) |
+-----+
| put |
+-----+
1 row in set (0.00 sec)
```

d) select concat(day(now()), concat('.',month(now()),concat('.',year(now()))))
"Date";

```
mysql> select concat(day(now()),concat('.',month(now()),concat('.',year(now())))) "Date";
+-----+
| Date |
+-----+
| 9.1.2022 |
+-----+
1 row in set (0.00 sec)
```

e) select right("Media",3);

```
mysql> select right("Media",3);
+-----+
| right("Media",3) |
+-----+
| dia |
+-----+
1 row in set (0.00 sec)
```

f) select concat(moviename,concat(' - ',type)) from movie;

```
mysql> select concat(moviename,concat(' - ',type)) from movie;
+-----+
| concat(moviename,concat(' - ',type)) |
+-----+
| The Kashmir Files - Action |
| Attack - Action |
| Loop Lapeta - Thriller |
| Bdhai Do - Drama |
| Shabaash Mothu - Biography |
| Gehriyaan - Romance |
+-----+
6 rows in set (0.00 sec)
```

g) select left(productioncost,4) from movie;

```
mysql> select left(productioncost,4) from movie;
+-----+
| left(productioncost,4) |
+-----+
| 1245 |
| 1120 |
| 2500 |
| 7200 |
| 1000 |
| 1500 |
+-----+
6 rows in set (0.00 sec)
```


h)select right (businesscost,4)from movie;

```
mysql> select right(businesscost,4) from movie;
+-----+
| right(businesscost,4) |
+-----+
| 0000                  |
| 0000                  |
| 0000                  |
| 0000                  |
| 0000                  |
| 0000                  |
+-----+
6 rows in set (0.00 sec)
```

i)select weekday(releasedate) from movie;

```
mysql> select weekday(releasedate) from movie;
+-----+
| weekday(releasedate) |
+-----+
| 2                    |
| 4                    |
| 1                    |
| 4                    |
| 4                    |
| 4                    |
+-----+
6 rows in set (0.00 sec)
```

j)select dayname(releasedate) from movie;

```
mysql> select dayname(releasedate) from movie;
+-----+
| dayname(releasedate) |
+-----+
| Wednesday            |
| Friday               |
| Tuesday              |
| Friday               |
| Friday               |
| Friday               |
+-----+
6 rows in set (0.01 sec)
```

Program 11 : Suppose your school management has decided to conduct cricket matches between students of Class XI and Class XII. Students of each class are asked to join any one of the four teams – Team Titan, Team Rockers, Team Magnet and Team Hurricane. During summer vacations, various matches will be conducted between these teams. Help your sports teacher to do the following:

- A. Create a database “Sports”.
- B. Create a table “TEAM” with following considerations:
 - a) It should have a column TeamID for storing an integer value between 1 to 9, which refers to unique identification of a team.
 - b) Each TeamID should have its associated name (TeamName), which should be a string of length not less than 10 characters.
 - c) Using table level constraint, make TeamID as the primary key.
 - a) Show the structure of the table TEAM using a SQL statement.
 - b) As per the preferences of the students four teams were formed as given below. Insert these four rows in TEAM table:
 - d) Row 1: (1, Tehlka)
 - e) Row 2: (2, Toofan)
 - f) Row 3: (3, Aandhi)
 - g) Row 3: (4, Shailab)
 - a) Show the contents of the table TEAM using a DML statement.
 - b) Now create another table MATCH_DETAILS and insert data as shown below. Choose appropriate data types and constraints for each attribute.

MatchID	MatchDate	FirstTeamID	SecondTeamID	FirstTeamScore	SecondTeamScore
M1	2021/12/20	1	2	107	93
M2	2021/12/21	3	4	156	158
M3	2021/12/22	1	3	86	81
M4	2021/12/23	2	4	65	67
M5	2021/12/24	1	4	52	88
M6	2021/12/25	2	3	97	68

Answers:

- a) create database sports;

```
mysql> create database sports
-> ;
Query OK, 1 row affected (0.01 sec)

mysql> use sports;
Database changed
mysql>
```

b) Creating table with the given specification

create table team -> (teamid int(1), -> teamnamevarchar(10), primary key(teamid));

c) desc team;

```
mysql> desc team;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| teamid | int | NO | PRI | NULL | |
| teamname | varchar(10) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

□ **Inserting data:**

mysql> insert into team -> values(1,'Tehlka');

```
mysql> insert into team
-> values(1,'Tehlka');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into team
-> values(2,'Toofan');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into team
-> values(3,'Aandhi');
Query OK, 1 row affected (0.03 sec)
```

```
mysql> insert into team
-> values(4,'Shailab');
Query OK, 1 row affected (0.03 sec)
```

□ **Show the content of table - team:**

select * from team;

```
mysql> select * from team;
+-----+-----+
| teamid | teamname |
+-----+-----+
|      1 | Tehlka   |
|      2 | Toofan   |
|      3 | Aandhi   |
|      4 | Shailab  |
+-----+-----+
4 rows in set (0.00 sec)
```

□ **Creating another table:**

create table match_details

-> (matchid varchar(2) primary key,

-> matchdate date,

-> firstteamid int(1) references team(teamid),

-> secondteamid int(1) references team(teamid),

-> firstteamscore int(3),

-> secondteamscore int(3));

```
mysql> create table match_details
-> (matchid varchar(2) primary key,
-> matchdate date,
-> firstteamid int(1) references team(teamid),
-> secondteamid int(1) references team(teamid),
-> firstteamscore int(3),
-> secondteamscore int(3));
Query OK, 0 rows affected, 4 warnings (0.03 sec)
```

```
mysql> select * from match_details;
+-----+-----+-----+-----+-----+-----+
| matchid | matchdate | firstteamid | secondteamid | firstteamscore | secondteamscore |
+-----+-----+-----+-----+-----+-----+
| M1      | 2021-12-20 | 1           | 2           | 107            | 93              |
| M2      | 2021-12-21 | 3           | 4           | 156            | 158             |
| M3      | 2021-12-22 | 1           | 3           | 86             | 81              |
| M4      | 2021-12-23 | 2           | 4           | 65             | 67              |
| M5      | 2021-12-24 | 1           | 4           | 52             | 88              |
| M6      | 2021-12-25 | 2           | 3           | 97             | 68              |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

➤ Write following queries according to program no 11.

- Display the matchid, teamid, teamscore whoscored more than 70 in first iningalong with team name.
- Display matchid, teamname and secondteamscore between 100 to 160.
- Display matchid, teamnames along with matchdates.
- Display unique team names
- Display matchid and matchdate played by Anadhi and Shailab.

Answers:

- select match_details.matchid, match_details.firstteamid, team.teamname, match_details.firstteamscore from match_details, team where match_details.firstteamid = team.teamid and match_details.first

```
mysql> select * from match_details;
```

matchid	matchdate	firstteamid	secondteamid	firstteamscore	secondteamscore
M1	2021-12-20	1	2	107	93
M2	2021-12-21	3	4	156	158
M3	2021-12-22	1	3	86	81
M4	2021-12-23	2	4	65	67
M5	2021-12-24	1	4	52	88
M6	2021-12-25	2	3	97	68

6 rows in set (0.01 sec)

- select match_details.matchid, match_details.firstteamid, team.teamname, match_details.firstteamscore from match_details, team where match_details.firstteamid = team.teamid and match_details.firstteamscore > 70;

```
mysql> select matchid, teamname, secondteamscore from match_details, team where match_details.secondteamid=team.teamid and match_details.secondteamscore between 100 and 160;
```

matchid	teamname	secondteamscore
M2	Shailab	158

1 row in set (0.00 sec)

- select matchid, teamname, firstteamid, secondteamid, matchdate from match_details, team where match_details.firstteamid = team.teamid;

```
mysql> select matchid, teamname, firstteamid, secondteamid, matchdate from match_details, team where match_details.firstteamid=team.teamid;
```

matchid	teamname	firstteamid	secondteamid	matchdate
M1	Tehlka	1	2	2021-12-20
M2	Aandhi	3	4	2021-12-21
M3	Tehlka	1	3	2021-12-22
M4	Toofan	2	4	2021-12-23
M5	Tehlka	1	4	2021-12-24
M6	Toofan	2	3	2021-12-25

6 rows in set (0.00 sec)

- select distinct(teamname) from match_details, team where match_details.firstteamid = team.teamid;

```
mysql> select distinct(teamname) from match_details, team where match_details.firstteamid=team.teamid;
+-----+
| teamname |
+-----+
| Tehlka   |
| Aandhi   |
| Shailab  |
| Toofan   |
+-----+
4 rows in set (0.03 sec)
```

e) select matchid,matchdate from match_details, team wherematch_details.firstteamid = team.teamid and team.teamname in ('Aandhi','Shailab');

```
mysql> select matchid,matchdate from match_details, team where match_details.firstteamid=team.teamid and team.teamname
in ('Aandhi','Shailab');
+-----+-----+
| matchid | matchdate |
+-----+-----+
| M2      | 2021-12-21 |
| M4      | 2021-12-23 |
+-----+-----+
2 rows in set (0.00 sec)
```

Program 12: Consider the following table and write the queries:

itemno	item	dcode	qty	unitprice	stockdate
S005	Ballpen	102	100	10	2018/04/22
S003	Gel Pen	101	150	15	2018/03/18
S002	Pencil	102	125	5	2018/02/25
S006	Eraser	101	200	3	2018/01/12
S001	Sharpner	103	210	5	2018/06/11
S004	Compass	102	60	35	2018/05/10
S009	A4 Papers	102	160	5	2018/07/17

- Display all the items in the ascending order of stockdate.
- Display maximum price of items for each dealer individually as per dcode from stock.
- Display all the items in descending orders of itemnames.

- d) Display average price of items for each dealer individually as per doce from stock which avergae price is more than 5.
- e) Diisplay the sum of quantity for each dcode.

Answers:

- a) select * from stock order by stockdate;

```
mysql> select * from stock order by stockdate;
```

itemno	item	dcode	qty	unitprice	stockdate
S006	Eraser	101	200	3	2018-01-12
S002	Pencil	102	125	5	2018-02-25
S003	Gel Pen	101	150	15	2018-03-18
S005	BallPen	102	100	10	2018-04-22
S004	Compass	102	60	35	2018-05-10
S009	A4 Papers	102	160	5	2018-07-17
S001	Sharpner	103	210	5	2018-11-06

7 rows in set (0.00 sec)

- b) Select dcode, max(unitprice) from stock group by code;

```
mysql> select dcode,max(unitprice) from stock group by dcode;
```

dcode	max(unitprice)
103	5
102	35
101	15

3 rows in set (0.01 sec)

- c) select * from stock order by item desc;

```
mysql> select * from stock order by item desc;
```

itemno	item	dcode	qty	unitprice	stockdate
S001	Sharpner	103	210	5	2018-11-06
S002	Pencil	102	125	5	2018-02-25
S003	Gel Pen	101	150	15	2018-03-18
S006	Eraser	101	200	3	2018-01-12
S004	Compass	102	60	35	2018-05-10
S005	BallPen	102	100	10	2018-04-22
S009	A4 Papers	102	160	5	2018-07-17

7 rows in set (0.01 sec)

- d) select dcode,avg(unitprice) from stock group by dcodehaving avg(unitprice)>5;

```
mysql> select dcode,avg(unitprice) from stock group by dcode having avg(unitprice)>5;
```

dcode	avg(unitprice)
102	13.7500
101	9.0000

2 rows in set (0.00 sec)

e) select dcode,sum(qty) from stock group by dcode;

```
mysql> select dcode,sum(qty) from stock group by dcode;
+-----+-----+
| dcode | sum(qty) |
+-----+-----+
| 103   | 210      |
| 102   | 445      |
| 101   | 350      |
+-----+-----+
3 rows in set (0.03 sec)
```

PYTHON

DATABASE

CONNECTIVITY

Program 13. Write a MySQL connectivity program in Python to

- **Create a database school**
- **Create a table students with the specifications - ROLLNO integer, STNAME character(10) in MySQL and perform the following operations:**
 - o **Insert two records in it**
 - o **Display the contents of the table**

Answer:

```
import mysql.connector
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="root",
    port="3306",
    #database="my_first_db"
)
def c_database():
    try:
        dn=input("Enter Database Name=")
        c.execute("create database {}".format(dn))
        c.execute("use {}".format(dn))
        print("Database created successfully")
    except Exception as a:
        print("Database Error",a)
#Function to Drop Database as per users choice
def d_database():
    try:
        dn=input("Enter Database Name to be dropped=")
        c.execute("drop database {}".format(dn))
        print("Database deleted sucessfully")
    except Exception as a:
        print("Database Drop Error",a)

#Function to create Table
def c_table():
```

```

try:
    c.execute("create table students(rollno int(3),sname varchar(20));")
    #mycursor.execute("CREATE TABLE customers (name VARCHAR(255), address
VARCHAR(255))")
    print("Table created successfully")
except Exception as a:
    print("Create Table Error",a)

#Function to Insert Data
def e_data():
    try:
        while True:
            rno=int(input("Enter student rollno="))
            name=input("Enter student name=")
            c.execute("use {}".format('STD12'))
            c.execute("insert into students values({}, '{}');".format(rno,name))
            mydb.commit()
            choice=input("Do you want to add more record<y/n>=")
            if choice in "Nn":
                break
    except Exception as a:
        print("Insert Record Error",a)

#Function to Display Data
def d_data():
    try:
        c.execute("select * from students")
        data=c.fetchall()
        for i in data:
            print(i)
    except Exception as a:
        print("Display Record Error",a)

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="root",
    port="3306",
    #database="my_first_db"
)
c=mydb.cursor()
while True:
    print("MENU\n1. Create Database\n2. Drop Database \n3. Create Table\n4. Insert Record \n5.
Display Entire Data\n6. Exit")
    choice=int(input("Enter your choice<1-6>="))
    if choice==1:
        c_database()
    elif choice==2:

```

```
d_database()
elif choice==3:
    c_table()
elif choice==4:
    e_data()
elif choice==5:
    d_data()
elif choice==6:
    break
else:
    print("Wrong option selected")
```

Output:

```
MENU
1. Create Database
2. Drop Database
3. Create Table
4. Insert Record
5. Display Entire Data
6. Exit
Enter your choice<1-6>=1
Enter Database Name=school
Database created successfully
MENU
1. Create Database
2. Drop Database
3. Create Table
4. Insert Record
5. Display Entire Data
6. Exit
Enter your choice<1-6>=3
Table created successfully
-----
MENU
1. Create Database
2. Drop Database
3. Create Table
4. Insert Record
5. Display Entire Data
6. Exit
Enter your choice<1-6>=4
Enter student rollno=11
Enter student name=Raj
Do you want to add more record<y/n>=n
MENU
```

Program 14: Perform all the operations with reference to table 'students' through MySQL-Python connectivity.

```
import mysql.connector as ms
db=ms.connect(
    host="localhost",
    user="root",
    passwd="root",
    port = 3306,
    database="daksh"
)
#cn=db.cursor()

def insert_rec():
    try:
        while True:
            rn=int(input("Enter roll number:"))
            sname=input("Enter name:")
            marks=float(input("Enter marks:"))
            gr=input("Enter grade:")
            cn.execute("insert into student values({}, {}, {}, {})".format(rn,sname,marks,gr))
            db.commit()
            ch=input("Want more records? Press (N/n) to stop entry:")
            if ch in 'Nn':
                print("Record Inserted.....")
                break

    except Exception as e:
        print("Error", e)
def update_rec():
    try:
        rn=int(input("Enter rollno to update:"))
        marks=float(input("Enter new marks:"))
        gr=input("Enter Grade:")
        cn.execute("update student set marks={},gr='{}' where rn={}".format(marks,gr,rn))
        db.commit()
```

```

        print("Record Updated.....")
    except Exception as e:
        print("Error",e)
def delete_rec():
    try:

        rn=int(input("Enter rollno to delete:"))
        cn.execute("delete from student where rn={}".format(rn))
        db.commit()
        print("Record Deleted.....")
    except Exception as e:
        print("Error",e)
def view_rec():
    try:
        cn.execute("select * from student")
        records = cn.fetchall()
        for record in records:
            print(record)
        #db.commit()
        #print("Record...")
    except Exception as e:
        print("Error",e)
db = ms.connect(
    host="localhost",
    user="root",
    passwd="root",
    port=3306,
    database="daksh"
)
cn = db.cursor()
while True:
    print("MENU\n1. Insert Record\n2. Update Record \n3. Delete Record\n4. Display Record \n5. Exit")
    ch=int(input("Enter your choice<1-4>="))
    if ch==1:
        insert_rec()
    elif ch==2:
        update_rec()
    elif ch==3:
        delete_rec()
    elif ch==4:
        view_rec()
    elif ch==5:
        break
    else:
        print("Wrong option selected")

```

Output:

MENU

1. Insert Record
2. Update Record
3. Delete Record
4. Display Record
5. Exit

Enter your choice<1-4>=1

Enter roll number:111

Enter name:Sagar

Enter marks:98

Enter grade:A1

Want more records? Press (N/n) to stop entry:y

Enter roll number:114

Enter name:Vipul

Enter marks:83

Enter grade:A2

Want more records? Press (N/n) to stop entry:n

MENU

1. Insert Record
2. Update Record
3. Delete Record
4. Display Record
5. Exit

Enter your choice<1-4>=4

युजिता

```
(1, 'Amrita', Decimal('88.00'), 'A2')
(102, 'Jay', Decimal('98.00'), 'A1')
(111, 'Sagar', Decimal('98.00'), 'A1')
(114, 'Vipul', Decimal('83.00'), 'A2')
```

MENU

1. Insert Record
2. Update Record
3. Delete Record
4. Display Record
5. Exit

Enter your choice<1-4>=2

Enter rollno to update:1

Enter new marks:98

Enter Grade:A1

MENU

1. Insert Record
2. Update Record
3. Delete Record
4. Display Record
5. Exit

Enter your choice<1-4>=4

```
(1, 'Amrita', Decimal('98.00'), 'A1')
(111, 'Sagar', Decimal('98.00'), 'A1')
(114, 'Vipul', Decimal('83.00'), 'A2')
```

पुणे

Enter your choice<1-4>=3

Enter rollno to delete:1

MENU

1. Insert Record
2. Update Record
3. Delete Record
4. Display Record
5. Exit

Enter your choice<1-4>=4

(102, 'Jay', Decimal('98.00'), 'A1')
(111, 'Sagar', Decimal('98.00'), 'A1')
(114, 'Vipul', Decimal('83.00'), 'A2')

MENU

1. Insert Record
2. Update Record
3. Delete Record
4. Display Record
5. Exit

Enter your choice<1-4>=4

(102, 'Jay', Decimal('98.00'), 'A1')
(111, 'Sagar', Decimal('98.00'), 'A1')
(114, 'Vipul', Decimal('83.00'), 'A2')

MENU

1. Insert Record
2. Update Record
3. Delete Record
4. Display Record
5. Exit

Enter your choice<1-4>=5

>>>

पुष्पा

