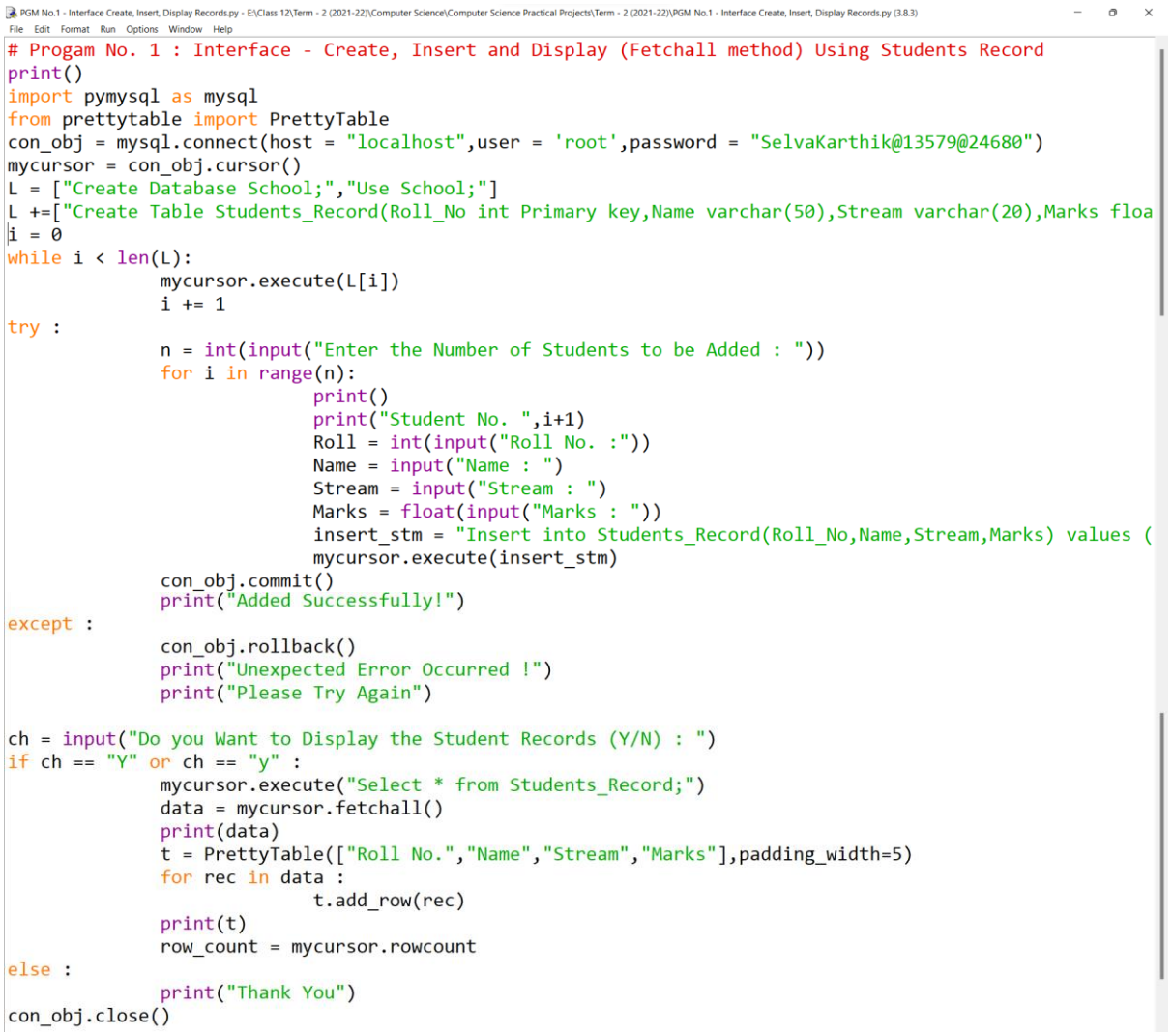
Python Experiments with Codes and Outputs (7 Experiments):

1. Write a program to Create, Insert and Display (Fetchall method) Students Record using Interfacing Concepts.

Source Code:



Input/Output:

```
Enter the Number of Students to be Added : 3

Student No.  1
Roll No. :1001
Name : Ashik
Stream : Computer Science
Marks : 98.3

Student No.  2
Roll No. :1002
Name : Bala
Stream : Commerce
Marks : 87.1

Student No.  3
Roll No. :1003
Name : Ciril
Stream : Biology
Marks : 99.4
Added Successfully!
Do you Want to Display the Student Records (Y/N) : Y
```

```
Do you Want to Display the Student Records (Y/N) : Y
+------------------+---------------+--------------------------+---------------+
|     Roll No.     |      Name     |          Stream          |     Marks     |
+------------------+---------------+--------------------------+---------------+
|       1001       |     Ashik     |     Computer Science     |      98.3     |
|       1002       |      Bala     |         Commerce         |      87.1     |
|       1003       |     Ciril     |         Biology          |      99.4     |
+------------------+---------------+--------------------------+---------------+
>>>
```

2. Write a program to Create, Insert,Update and Display (Fetchall method\Fetchmany method) Students Record Using Interfacing Concepts.

Source Code:

PGM No.2 - Interface Create, Insert, Update and Display.py - E:\Class 12\Term - 2 (2021-22)\Computer Science\Computer Science Practical Projects\Term - 2 (2021-22)\PGM No.2 - Interface Create, Insert, Update and Display.py (3.8.3)

File Edit Format Run Options Window Help

```
# Progam No. 2 : Interface - Create, Insert,Update and Display (Fetchall method\Fetchmany method) Using S
import mysql.connector as mysql
from prettytable import PrettyTable
con_obj = mysql.connect(host = "localhost",user = 'root',password = "SelvaKarthik@13579@24680")
mycursor = con_obj.cursor()
L = ["Create Database School;","Use School;"]
L +=["Create Table Students_Record(Roll_No int Primary key,Name varchar(50),Stream varchar(20),Marks floa
i = 0
while i < len(L):
        mycursor.execute(L[i])
        i += 1
try :
        n = int(input("Enter the Number of Students to be Added : "))
        for i in range(n):
                print()
                print("Student No. ",i+1)
                Roll = int(input("Roll No. :"))
                Name = input("Name : ")
                Stream = input("Stream : ")
                Marks = float(input("Marks : "))
                insert_stm = "Insert into Students_Record(Roll_No,Name,Stream,Marks) values (
                mycursor.execute(insert_stm)
        con_obj.commit()
        print("Added Successfully!")
```

```
except :
                con_obj.rollback()
                print("Unexpected Error Occurred !")
                print("Please Try Again")
while True :
                ch = input("Do you Want to Update any Record (Y/N) : ")
                if ch.lower() != "y" :
                                break
                else :
                                try :
                                                update_roll = int(input("Enter the Roll No. to be Updated : "))
                                                update_marks = float(input("Enter the Marks to be Updated : "))
                                                insert_stm = "Update Students_Record SET Marks = {0} where Roll
                                                mycursor.execute(insert_stm)
                                                print("Rows Affected : ",mycursor.rowcount)
                                                con_obj.commit()
                                                print("Successfully Updated !!")
                                except :
                                                con_obj.rollback()
                                                print("Unexpected Error Occurred !")
                                                print("Please Check All given Data is Correct and Try Again")

ch = input("Do you Want to Display the Student Records (Y/N) : ")
if ch == "Y" or ch == "y" :
                mycursor.execute("Select * from Students_Record;")
                data = mycursor.fetchall()
                t = PrettyTable(["Roll No.","Name","Stream","Marks"],padding_width=5)
                for rec in data :
                                t.add_row(rec)
                print(t)
                row_count = mycursor.rowcount
                print("Total No. of Records : ",row_count)
else :
                print("Thank You")
con_obj.close()
```

Input/Output

```
Python 3.8.3 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Class 12\Term - 2 (2021-22)\Computer Science\Computer Science Practical Projects\Term - 2
(2021-22)\PGM No.2 - Interface Create, Insert, Update and Display.py
Enter the Number of Students to be Added : 3

Student No.  1
Roll No. :1001
Name : Ashik
Stream : Computer Science
Marks : 98.4

Student No.  2
Roll No. :1002
Name : Bala
Stream : Commerce
Marks : 87.1

Student No.  3
Roll No. :1003
Name : Ciril
Stream : Biology
Marks : 99.4
Added Successfully!
Do you Want to Update any Record (Y/N) : Y
Enter the Roll No. to be Updated : 1002
Enter the Marks to be Updated : 90.5
Rows Affected :  1
Successfully Updated !!
Do you Want to Update any Record (Y/N) : N
Do you Want to Display the Student Records (Y/N) : Y
+------------------+---------------+--------------------------+---------------+
|     Roll No.     |      Name     |          Stream          |     Marks     |
+------------------+---------------+--------------------------+---------------+
|       1001       |     Ashik     |     Computer Science     |      98.4     |
|       1002       |      Bala     |         Commerce         |      90.5     |
|       1003       |     Ciril     |         Biology          |      99.4     |
+------------------+---------------+--------------------------+---------------+
Total No. of Records :  3
>>>
```

3. Write a python program Create, Insert, Delete and Display (Fetchall method) Students Record using Interfacing Concepts.

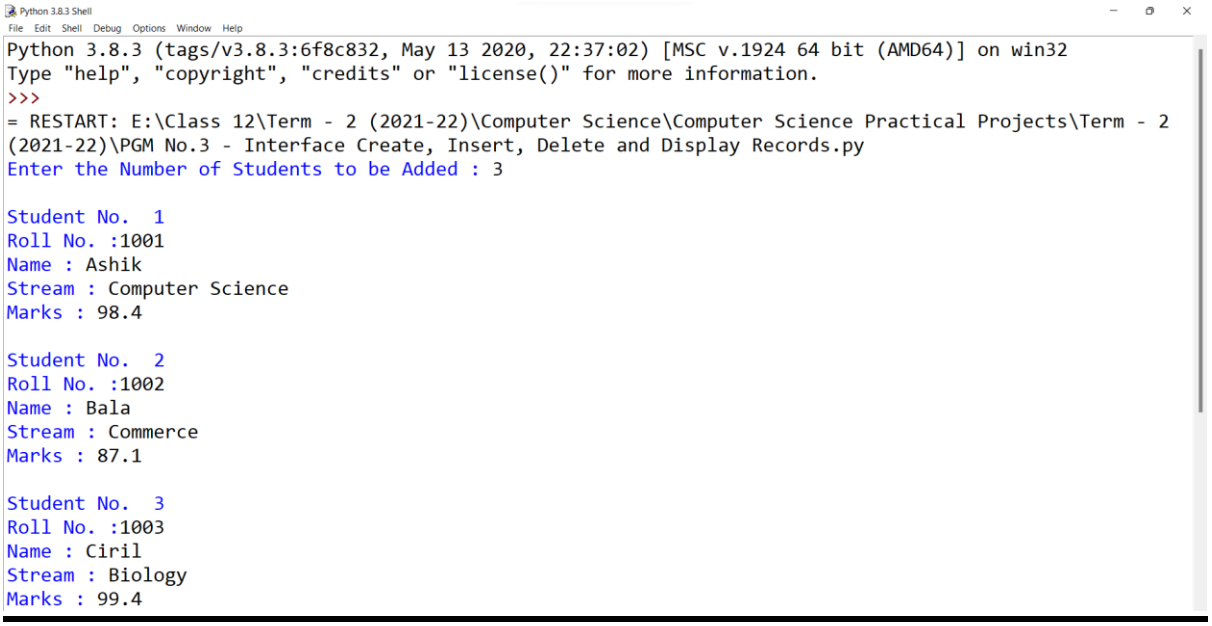
Source Code:

PGM No.3 - Interface Create, Insert, Delete and Display Records.py - E:\Class 12\Term - 2 (2021-22)\Computer Science\Computer Science Practical Projects\Term - 2 (2021-22)\PGM No.3 - Interface Create, Insert, Delete and Display Records.py (3.8.3)

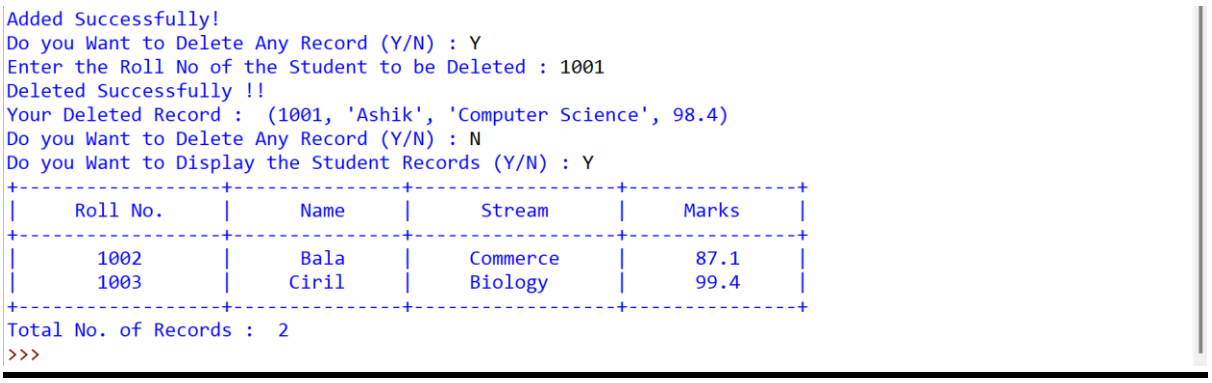
File Edit Format Run Options Window Help

```
# Progam No. 3 : Interface - Create, Insert, Delete and Display(Fetchall method) using Students Record
import mysql.connector as mysql
from prettytable import PrettyTable
con_obj = mysql.connect(host = "localhost",user = 'root',password = "SelvaKarthik@13579@24680")
mycursor = con_obj.cursor()
L = ["Create Database School;","Use School;"]
L += ["Create Table Students_Record(Roll_No int Primary key,Name varchar(50),Stream varchar(20),Marks flo
i = 0
while i < len(L):
        mycursor.execute(L[i])
        i += 1
try :
        n = int(input("Enter the Number of Students to be Added : "))
        for i in range(n):
                print()
                print("Student No. ",i+1)
                Roll = int(input("Roll No. :"))
                Name = input("Name : ")
                Stream = input("Stream : ")
                Marks = float(input("Marks : "))
                insert_stm = "Insert into Students_Record(Roll_No,Name,Stream,Marks) values (
                mycursor.execute(insert_stm)
        con_obj.commit()
        print("Added Successfully!")
except :
        con_obj.rollback()
        print("Unexpected Error Occurred !")
        print("Please Try Again")

while True :
        ch = input("Do you Want to Delete Any Record (Y/N) : ")
        if ch.lower() != "y"  :
                break
        else :
                if mycursor.rowcount == 0 :
                        print("No Records To Delete !!")
                        break
                else :
                        try :
                                delete_roll = int(input("Enter the Roll No of the
                                delete_record_stm = "Select * from Students_Recor
                                insert_stm = "Delete from Students_Record where R
                                mycursor.execute(delete_record_stm)
                                deleted_record = mycursor.fetchone()
                                mycursor.execute(insert_stm)
                                con_obj.commit()
                                print("Deleted Successfully !!")
                                print("Your Deleted Record : ",deleted_record)
                        except :
                                con_obj.rollback()
                                print("Unexpected Error Occurred !")
                                print("Please Try Again")
                                break

ch = input("Do you Want to Display the Student Records (Y/N) : ")
if ch == "Y" or ch == "y" :
        mycursor.execute("Select * from Students_Record;")
        data = mycursor.fetchall()
        t = PrettyTable(["Roll No.","Name","Stream","Marks"],padding_width=5)
        for rec in data :
                t.add_row(rec)

        print(t)
        row_count = mycursor.rowcount
        print("Total No. of Records : ",row_count)
else :
        print("Thank You")
con_obj.close()
```

Input/Output:



```
Python 3.8.3 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Class 12\Term - 2 (2021-22)\Computer Science\Computer Science Practical Projects\Term - 2
(2021-22)\PGM No.3 - Interface Create, Insert, Delete and Display Records.py
Enter the Number of Students to be Added : 3

Student No.  1
Roll No. :1001
Name : Ashik
Stream : Computer Science
Marks : 98.4

Student No.  2
Roll No. :1002
Name : Bala
Stream : Commerce
Marks : 87.1

Student No.  3
Roll No. :1003
Name : Ciril
Stream : Biology
Marks : 99.4
```



```
Added Successfully!
Do you Want to Delete Any Record (Y/N) : Y
Enter the Roll No of the Student to be Deleted : 1001
Deleted Successfully !!
Your Deleted Record :  (1001, 'Ashik', 'Computer Science', 98.4)
Do you Want to Delete Any Record (Y/N) : N
Do you Want to Display the Student Records (Y/N) : Y
+------------------+---------------+------------------+---------------+
|     Roll No.     |      Name     |      Stream      |     Marks     |
+------------------+---------------+------------------+---------------+
|       1002       |      Bala     |     Commerce     |      87.1     |
|       1003       |     Ciril     |     Biology      |      99.4     |
+------------------+---------------+------------------+---------------+
Total No. of Records :  2
>>>
```

4. Write a program To Create, Insert, Display(Fetchall, Fetchmany, Fetchone) Students Record using Interfacing Concepts.

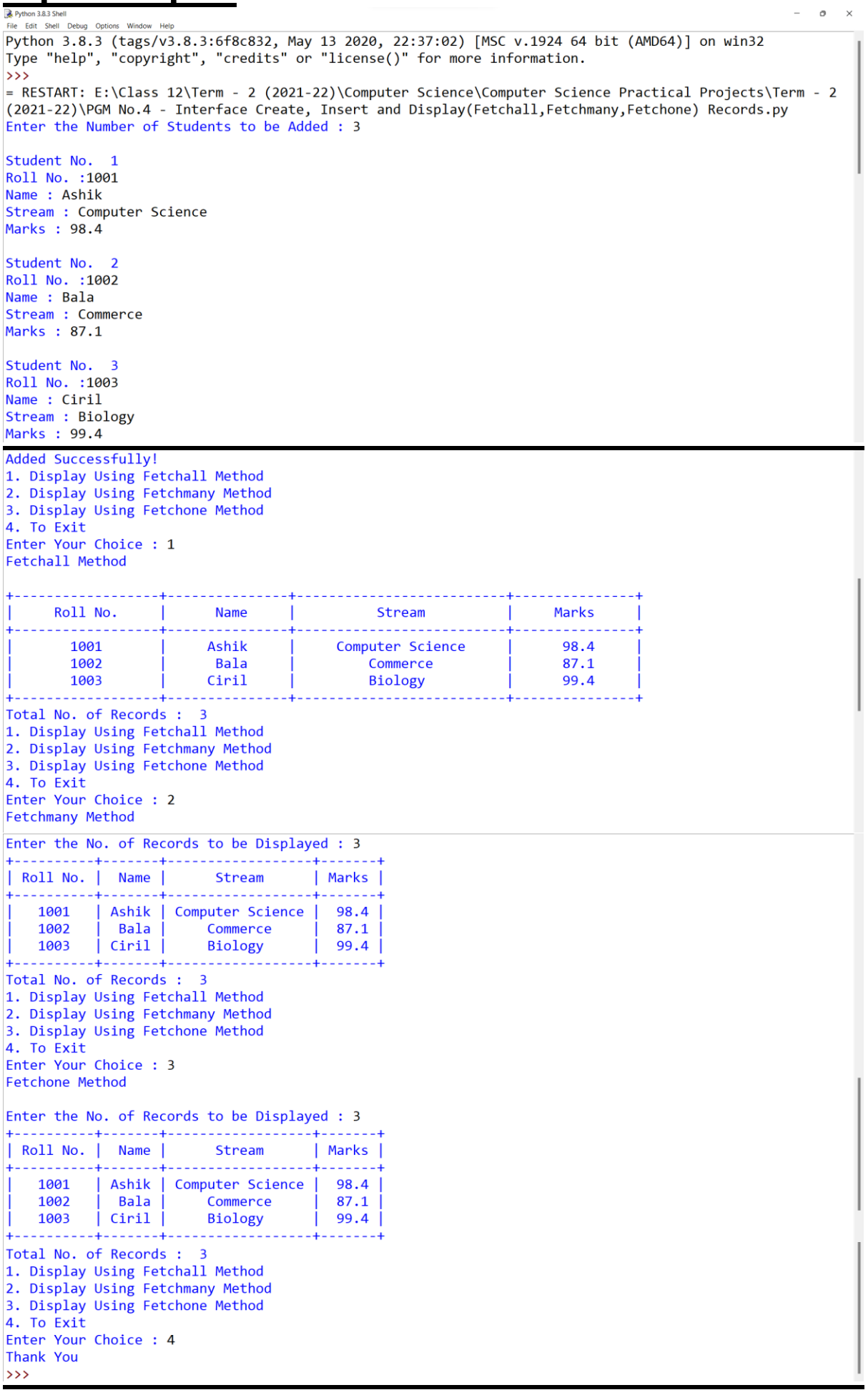
Source Code:

```
PGM No.4 - Interface Create, Insert and Display(Fetchall,Fetchmany,Fetchone) Records.py - E:\Class 12\Term - 2 (2021-22)\Computer Science\Computer Science Practical Projects\Term - 2 (2021-22)\PGM No.4 - Interface Create, Insert and Display(Fetchall,Fetch
File  Edit  Format  Run  Options  Window  Help
# Progam No. 4 : Interface - Create, Insert and Display(FetchAll,Fetchone,Fetchmany) using Students Recor
import mysql.connector as mysql
from prettytable import PrettyTable
con_obj = mysql.connect(host = "localhost",user = 'root',password = "SelvaKarthik@13579@24680")
mycursor = con_obj.cursor()
L = ["Create Database School;","Use School;"]
L += ["Create Table Students_Record(Roll_No int Primary key,Name varchar(50),Stream varchar(20),Marks flo
i = 0
def fetchall():
        con_obj = mysql.connect(host = "localhost",user = 'root',password = "SelvaKarthik@13579@246
        mycursor = con_obj.cursor()
        print("Fetchall Method")
        print()
        mycursor.execute("Select * from Students_Record;")
        data = mycursor.fetchall()
        t = PrettyTable(["Roll No.","Name","Stream","Marks"],padding_width=5)
        for rec in data :
                t.add_row(rec)
        print(t)
        row_count = mycursor.rowcount
        print("Total No. of Records : ",row_count)
        con_obj.close()
def fetchmany():
        con_obj = mysql.connect(host = "localhost",user = 'root',password = "SelvaKarthik@13579@246
```

```
            mycursor = con_obj.cursor()
            print("Fetchmany Method")
            print()
            mycursor.execute("Select * from Students_Record;")
            n = int(input("Enter the No. of Records to be Displayed : "))
            data = mycursor.fetchmany(n)
            t = PrettyTable(["Roll No.","Name","Stream","Marks"])
            for rec in data :
                        t.add_row(rec)
            print(t)
            row_count = mycursor.rowcount
            print("Total No. of Records : ",row_count)
            con_obj.close()
def fetchone():
            con_obj = mysql.connect(host = "localhost",user = 'root',password = "SelvaKarthik@13579@246
            mycursor = con_obj.cursor()
            print("Fetchone Method")
            print()
            n = int(input("Enter the No. of Records to be Displayed : "))
            mycursor.execute("Select * from Students_Record;")
            t = PrettyTable(["Roll No.","Name","Stream","Marks"])
            for i in range(n):
                        data = mycursor.fetchone()
                        t.add_row(data)
```

```
            print(t)
            row_count = mycursor.rowcount
            print("Total No. of Records : ",row_count)
            con_obj.close()

while i < len(L):
            mycursor.execute(L[i])
            i += 1
try :
            n = int(input("Enter the Number of Students to be Added : "))
            for i in range(n):
                        print()
                        print("Student No. ",i+1)
                        Roll = int(input("Roll No. :"))
                        Name = input("Name : ")
                        Stream = input("Stream : ")
                        Marks = float(input("Marks : "))
                        insert_stm = "Insert into Students_Record(Roll_No,Name,Stream,Marks) values (
                        mycursor.execute(insert_stm)
            count = mycursor.rowcount
            con_obj.commit()
            print("Added Successfully!")
except :
            con_obj.rollback()
            print("Unexpected Error Occurred !")
            print("Please Try Again")

ch = 0
while ch != 4 :
            print("1. Display Using Fetchall Method")
            print("2. Display Using Fetchmany Method")
            print("3. Display Using Fetchone Method")
            print("4. To Exit")
            ch = int(input("Enter Your Choice : "))
            if ch == 1 :
                        fetchall()

            elif ch == 2 :
                        fetchmany()

            elif ch == 3 :
                        fetchone()

else :
            print("Thank You")
con_obj.close()
```

Input/Output:



```
Python 3.8.3 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Class 12\Term - 2 (2021-22)\Computer Science\Computer Science Practical Projects\Term - 2
(2021-22)\PGM No.4 - Interface Create, Insert and Display(Fetchall,Fetchmany,Fetchone) Records.py
Enter the Number of Students to be Added : 3

Student No.  1
Roll No. :1001
Name : Ashik
Stream : Computer Science
Marks : 98.4

Student No.  2
Roll No. :1002
Name : Bala
Stream : Commerce
Marks : 87.1

Student No.  3
Roll No. :1003
Name : Ciril
Stream : Biology
Marks : 99.4
Added Successfully!
1. Display Using Fetchall Method
2. Display Using Fetchmany Method
3. Display Using Fetchone Method
4. To Exit
Enter Your Choice : 1
Fetchall Method

+------------------+---------------+--------------------------+---------------+
|     Roll No.     |      Name     |          Stream          |     Marks     |
+------------------+---------------+--------------------------+---------------+
|       1001       |     Ashik     |     Computer Science     |      98.4     |
|       1002       |      Bala     |         Commerce         |      87.1     |
|       1003       |     Ciril     |         Biology          |      99.4     |
+------------------+---------------+--------------------------+---------------+
Total No. of Records :  3
1. Display Using Fetchall Method
2. Display Using Fetchmany Method
3. Display Using Fetchone Method
4. To Exit
Enter Your Choice : 2
Fetchmany Method
Enter the No. of Records to be Displayed : 3
+----------+-------+------------------+-------+
| Roll No. |  Name |      Stream      | Marks |
+----------+-------+------------------+-------+
|   1001   | Ashik | Computer Science |  98.4 |
|   1002   |  Bala |     Commerce     |  87.1 |
|   1003   | Ciril |     Biology      |  99.4 |
+----------+-------+------------------+-------+
Total No. of Records :  3
1. Display Using Fetchall Method
2. Display Using Fetchmany Method
3. Display Using Fetchone Method
4. To Exit
Enter Your Choice : 3
Fetchone Method

Enter the No. of Records to be Displayed : 3
+----------+-------+------------------+-------+
| Roll No. |  Name |      Stream      | Marks |
+----------+-------+------------------+-------+
|   1001   | Ashik | Computer Science |  98.4 |
|   1002   |  Bala |     Commerce     |  87.1 |
|   1003   | Ciril |     Biology      |  99.4 |
+----------+-------+------------------+-------+
Total No. of Records :  3
1. Display Using Fetchall Method
2. Display Using Fetchmany Method
3. Display Using Fetchone Method
4. To Exit
Enter Your Choice : 4
Thank You
>>>
```

5. Write a program to Push, Pop and Display using Stack Operations.

Source Code:

PGM No.5 - Stack Operations Push, Pop, Display.py - E:\Class 12\Term - 2 (2021-22)\Computer Science\Computer Science Practical Projects\Term - 2 (2021-22)\PGM No.5 - Stack Operations Push, Pop, Display.py (3.8.3)

File Edit Format Run Options Window Help

```
# Program 5 : Stack Operations - Push, Pop and Display
stack = []
def push(n):
                stack.append(n)
def pop():
                 pop_value = stack.pop()
                 return pop_value
def display():
                print()
                print(stack[-1],"<---- top")
                for i in range(len(stack)-2,-1,-1):
                                print(stack[i])
                print()
ch = 0
while ch != 4 :
                print("1. Push")
                print("2. Pop")
                print("3. Display")
                print("4.To Exit ")
                ch = int(input("Enter Your Choice : "))
                if ch == 1 :
                                n = int(input("Enter the Value to be Pushed into the Stack : "))
                                push(n)
                elif ch == 2 :
                                if len(stack) == 0 :
                                                print("Stack is Empty !")
                                else :
                                                result = pop()
                                                print("Popped Value : ",result)
                elif ch == 3 :
                                display()
else :
                print("Thank You")
```

Input/Output:

Python 3.8.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Class 12\Term - 2 (2021-22)\Computer Science\Computer Science Practical Projects\Term - 2
(2021-22)\PGM No.5 - Stack Operations Push, Pop, Display.py
1. Push
2. Pop
3. Display
4.To Exit
Enter Your Choice : 1
Enter the Value to be Pushed into the Stack : 5
1. Push
2. Pop
3. Display
4.To Exit
Enter Your Choice : 1
Enter the Value to be Pushed into the Stack : 3
1. Push
2. Pop
3. Display
4.To Exit
Enter Your Choice : 1
Enter the Value to be Pushed into the Stack : 2
```

```
1. Push
2. Pop
3. Display
4.To Exit
Enter Your Choice : 1
Enter the Value to be Pushed into the Stack : 9
1. Push
2. Pop
3. Display
4.To Exit
Enter Your Choice : 2
Popped Value :  9
1. Push
2. Pop
3. Display
4.To Exit
Enter Your Choice : 3

2 <---- top
3
5
```

```
1. Push
2. Pop
3. Display
4.To Exit
Enter Your Choice : 4
```

6. Write a program to Push, Pop and Display Book Details using Stack Operations.

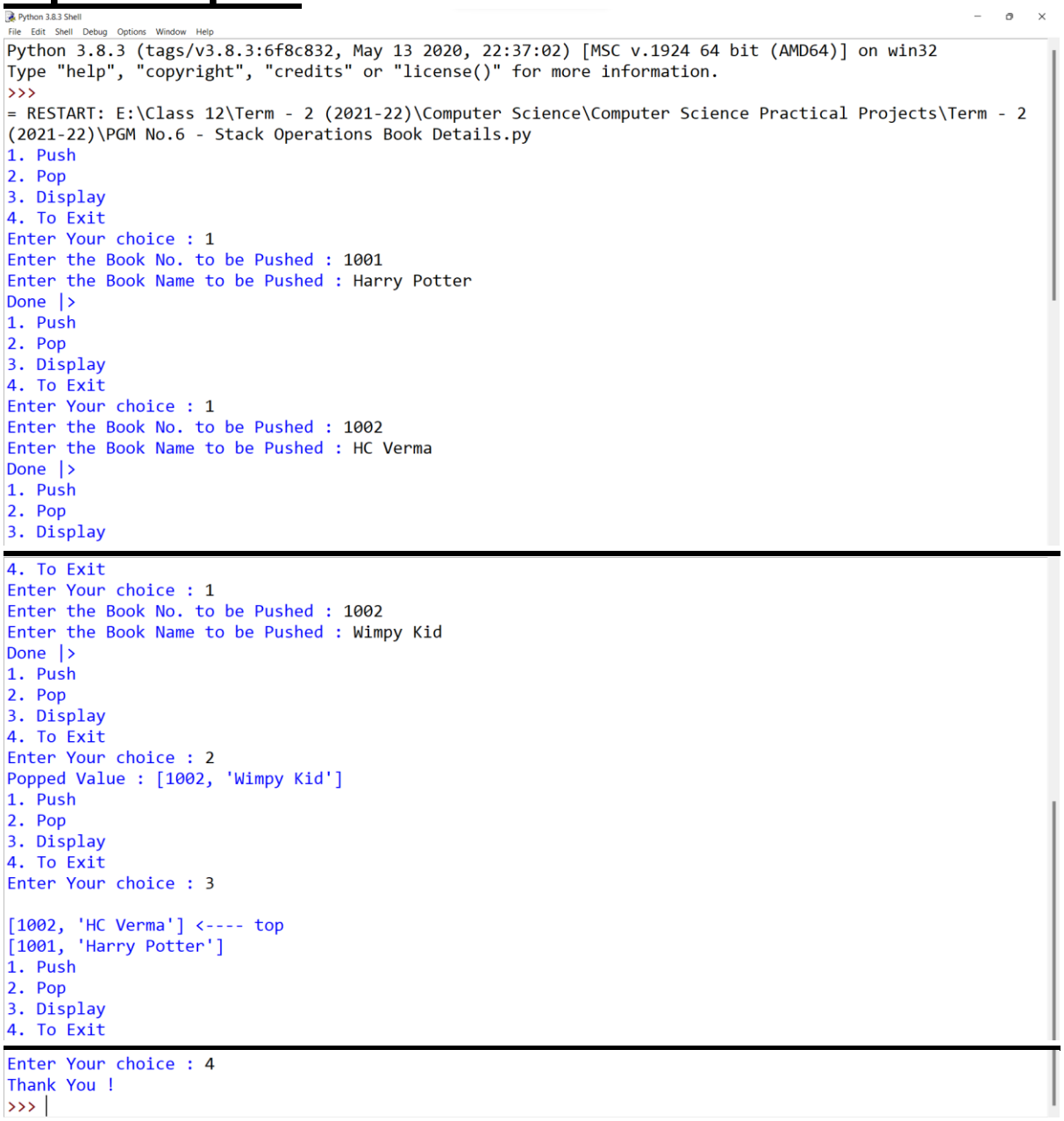
Source Code:

PGM No.6 - Stack Operations Book Details.py - E:\Class 12\Term - 2 (2021-22)\Computer Science\Computer Science Practical Projects\Term - 2 (2021-22)\PGM No.6 - Stack Operations Book Details.py (3.8.3)

File Edit Format Run Options Window Help

```
# Program 6 : Stack Opertaions : Push, Pop and Display Book Details
stack_books = []
def Push(list_pushed):
        stack_books.append(list_pushed)
        print("Done |>")
def Pop():
        pop_value = stack_books.pop()
        return pop_value
def Display():
        print()
        print(stack_books[-1],"<---- top")
        for i in range(len(stack_books)-2,-1,-1):
                print(stack_books[i])
        print
ch = 0
while ch != 4 :
        print("1. Push")
        print("2. Pop")
        print("3. Display")
        print("4. To Exit ")
        ch = int(input("Enter Your choice : "))
        if ch == 1 :
                book_no = int(input("Enter the Book No. to be Pushed : "))
                book_name = input("Enter the Book Name to be Pushed : ")
                list_pushed = [book_no,book_name]
                Push(list_pushed)
        if ch == 2 :
                if len(stack_books) == 0 :
                        print("Empty Stack")
                else :
                        print("Popped Value :",Pop())
        if ch == 3 :
                Display()
print("Thank You !")
```

Input/Output :



7. Write a program to Push only Numbers which is Divisible by 5, Pop and Display using Stack Operations.

Source Code:

PGM No.7 - Stack Operations Divisible by 5.py - E:\Class 12\Term - 2 (2021-22)\Computer Science\Computer Science Practical Projects\Term - 2 (2021-22)\PGM No.7 - Stack Operations Divisible by 5.py (3.8.3)

File Edit Format Run Options Window Help

```
# Program No. 7 - Stack Operations Push (Divisible by 5), Pop and Display
stack = []
def Push(List):
        for i in List :
                if i % 5 == 0 :
                        stack.append(i)
        print("Done |>")
def Pop():
        pop_value = stack.pop()
        print("Popped Value : ",pop_value)
def Display() :
        print()
        print(stack[-1],"<--- top")
        for i in range(len(stack)-2,-1,-1):
                print(stack[i])
        print()
ch = 0
while ch != 4 :
        print("1. Push")
        print("2. Pop")
        print("3. Display")
        print("4. To Exit")
        ch = int(input("Enter Your Choice : "))
        if ch == 1 :
                L = eval(input("Enter the List of Numbers : "))
                Push(L)
        elif ch == 2 :
                Pop()
        elif ch == 3 :
                Display()
else :
        print("Thank You")
```

Input/Output:

Python 3.8.3 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\Class 12\Term - 2 (2021-22)\Computer Science\Computer Science Practical Projects\Term - 2
(2021-22)\PGM No.7 - Stack Operations Divisible by 5.py
1. Push
2. Pop
3. Display
4. To Exit
Enter Your Choice : 1
Enter the List of Numbers : [5,3,15,10,2,4]
Done |>
1. Push
2. Pop
3. Display
4. To Exit
Enter Your Choice : 3

10 <--- top
15
5

1. Push
2. Pop
3. Display
4. To Exit
Enter Your Choice : 2
Popped Value :  10
1. Push
2. Pop
3. Display
4. To Exit
Enter Your Choice : 3

15 <--- top
5

1. Push
2. Pop
3. Display
4. To Exit
Enter Your Choice : 4
Thank You
>>>
```

SQL Experiments with Codes and Outputs (5 Experiments):

SQL - EXPERIMENT-1

```
mysql> select * from Flights;
+----------+----------+-----------+----------+-----------------+-------+-------------+
| FLightNo | Start    | End       | NO_STOPS | Airlines        | FARE  | TAX_PERCENT |
+----------+----------+-----------+----------+-----------------+-------+-------------+
| IC3      | MUMBAI   | DELHI     |        0 | INDIAN AIRLINES |  3400 |           6 |
| IC7      | BANGLORE | DELHI     |        1 | INDIAN AIRLINES |  1050 |          10 |
| MC1      | INDORE   | MUMBAI    |        0 | DECCAN AIRLINES |  3500 |           4 |
| IC3      | DELHI    | MUMBAI    |        0 | INDIAN AIRLINES |  4300 |          10 |
| AM8      | KANPUR   | BANGLORE  |        1 | JET AIRWAYS     | 10450 |           8 |
| IC8      | MUMBAI   | KOCHI     |        4 | INDIAN AIRLINES |  8300 |           4 |
| AM5      | DELHI    | TRIVANDRUM |       5 | JET AIRWAYS     | 13450 |           8 |
| MU4      | MUMBAI   | MADRAS    |        3 | SAHARA          |  9400 |           5 |
| IC7      | DELHI    | AHMEDABAD |        0 | INDIAN AIRLINES |  6500 |          10 |
+----------+----------+-----------+----------+-----------------+-------+-------------+
9 rows in set (0.00 sec)
```

(i) To display the AIRLINES wise average FARE and TAXP_PERCENT.

```
mysql> SELECT Airlines, avg(Fare), avg(TAX_PERCENT) from Flights GROUP BY Airlines;
+-----------------+-----------+------------------+
| Airlines        | avg(Fare) | avg(TAX_PERCENT) |
+-----------------+-----------+------------------+
| INDIAN AIRLINES | 4710.0000 |           8.0000 |
| DECCAN AIRLINES | 3500.0000 |           4.0000 |
| JET AIRWAYS     | 11950.0000 |          8.0000 |
| SAHARA          | 9400.0000 |           5.0000 |
+-----------------+-----------+------------------+
4 rows in set (0.01 sec)
```

(ii) To display the FlightNo and Fare is the ascending order of TAX_PERCENT.

```
mysql> SELECT FlightNo,Fare from Flights ORDER BY TAX_PERCENT;
+----------+-------+
| FlightNo | Fare  |
+----------+-------+
| MC1      |  3500 |
| IC8      |  8300 |
| MU4      |  9400 |
| IC3      |  3400 |
| AM8      | 10450 |
| AM5      | 13450 |
| IC7      |  1050 |
| IC3      |  4300 |
| IC7      |  6500 |
+----------+-------+
9 rows in set (0.00 sec)
```

(iii) To display the Total Price of the flights from DELHI to MUMBAI, where the Total Price is calculated as: FARE+FARE*TAX_PERCENT/100.

```
mysql> SELECT FARE+FARE*(TAX_PERCENT/100) AS "TOTAL PRICE FROM MUMBAI TO DELHI" FROM Flights WHERE Start='DELHI' AND End='MUMBAI';
+--------------------------------+
| TOTAL PRICE FROM MUMBAI TO DELHI |
+--------------------------------+
|                        4730.0000 |
+--------------------------------+
1 row in set (0.00 sec)
```

(iv) To display the minimum fare “Indian Airlines” is offering.

```
mysql>  select MIN(FARE) FROM Flights WHERE Airlines='INDIAN AIRLINES';
+-----------+
| MIN(FARE) |
+-----------+
|      1050 |
+-----------+
1 row in set (0.01 sec)
```

(v) To display the AIRLINES in the ascending order of NO_STOPS and descending order of AIRLINES.

```
mysql> SELECT Airlines FROM Flights ORDER BY NO_STOPS ASC, Airlines DESC;
+-----------------+
| Airlines        |
+-----------------+
| INDIAN AIRLINES |
| INDIAN AIRLINES |
| INDIAN AIRLINES |
| DECCAN AIRLINES |
| JET AIRWAYS     |
| INDIAN AIRLINES |
| SAHARA          |
| INDIAN AIRLINES |
| JET AIRWAYS     |
+-----------------+
9 rows in set (0.00 sec)
```

(vi) To display the total no. of flights from FLIGHTS where STARTING starts with a letter ‘M’ and the no. of stops is more than 2.

```
mysql> SELECT COUNT(*) AS "TOTAL FLIGHTS" FROM Flights
    ->
    ->           WHERE START LIKE 'M%' AND NO_STOPS>2;
+-----------------+
| TOTAL FLIGHTS |
+-----------------+
|             2 |
+-----------------+
1 row in set (0.01 sec)
```

(vii) SELECT FL_NO, AIRLINES FROM FLIGHTS
WHERE STARTING = ‘DELHI’;

```
mysql> SELECT FlightNo,Airlines FROM Flights WHERE Start='DELHI';
+----------+-----------------+
| FlightNo | Airlines        |
+----------+-----------------+
| IC3      | INDIAN AIRLINES |
| AM5      | JET AIRWAYS     |
| IC7      | INDIAN AIRLINES |
+----------+-----------------+
3 rows in set (0.00 sec)
```

(viii) SELECT COUNT (DISTINCT ENDING) FROM FLIGHTS;

```
mysql> SELECT COUNT(DISTINCT END) FROM Flights;
+----------------------+
| COUNT(DISTINCT END) |
+----------------------+
|                    7 |
+----------------------+
1 row in set (0.01 sec)
```

(ix) SELECT AIRLINES, SUM (FARE) FROM FLIGHTS
GROUP BY AIRLINES HAVING SUM (FARE)> 10000

```
mysql> SELECT Airlines,SUM(FARE) FROM Flights GROUP BY Airlines HAVING SUM(FARE)>10000;
+-----------------+-----------+
| Airlines        | SUM(FARE) |
+-----------------+-----------+
| INDIAN AIRLINES |     23550 |
| JET AIRWAYS     |     23900 |
+-----------------+-----------+
2 rows in set (0.00 sec)
```

(x) SELECT FL_NO, NO_STOPS, AIRLINES, FARE * TAXPERC/100 AS TAX
FROM FLIGHTS WHERE NO_ STOPS BETWEEN 1 AND 3;

```
mysql> SELECT FlightNo,NO_STOPS,Airlines,FARE*TAX_PERCENT/100 AS TAX FROM Flights WHERE NO_STOPS BETWEEN 1 AND 3;
+----------+----------+-----------------+----------+
| FlightNo | NO_STOPS | Airlines        | TAX      |
+----------+----------+-----------------+----------+
| IC7      |        1 | INDIAN AIRLINES | 105.0000 |
| AM8      |        1 | JET AIRWAYS     | 836.0000 |
| MU4      |        3 | SAHARA          | 470.0000 |
+----------+----------+-----------------+----------+
3 rows in set (0.00 sec)
```

SQL - EXPERIMENT-2

```
mysql> SELECT * FROM TEACHERS;
+------+-----------+-----------+------------+-------+------+------+
| SID  | TNAME     | DEPT      | DOB        | BASIC | HRA  | DA   |
+------+-----------+-----------+------------+-------+------+------+
|  101 | SIDDHARTH | ACCOUNTS  | 1975-12-25 | 12000 | 1000 |  300 |
|  104 | JOHN      | PHYSICS   | 1993-01-13 | 23000 | 2300 |  500 |
|  107 | NAMAN     | CHEMISTRY | 1994-08-12 | 32000 | 4000 |  500 |
|  114 | NUPUR     | ENGLISH   | 1978-07-09 | 12000 | 2200 | 1000 |
|  109 | JANVI     | PHYSICS   | 1975-05-01 | 42000 | 2700 | 1200 |
|  105 | SELVA     | ACCOUNTS  | 1964-06-21 | 18900 | 1690 |  300 |
|  117 | JANAKI    | COMPUTER  | 1967-03-17 | 15300 | 2310 |  220 |
|  111 | BINOY     | ECONOMICS | 1987-09-15 | 19800 | 2400 |  260 |
|  130 | AMRESH    | COMPUTER  | 1983-10-05 | 21700 | 2600 |  300 |
+------+-----------+-----------+------------+-------+------+------+
9 rows in set (0.00 sec)
```

(a) To display the names of the staff who are in Accounts department.

```
mysql> SELECT TNAME FROM TEACHERS WHERE DEPT='ACCOUNTS';
+-----------+
| TNAME     |
+-----------+
| SIDDHARTH |
| SELVA     |
+-----------+
2 rows in set (0.01 sec)
```

(b) To display the Total Pay and Average Pay of all Employees. The salary for one person is calculated as: BASIC+HRA+DA

```
mysql> SELECT SUM(BASIC+HRA+DA) AS "TOTAL PAY",AVG(BASIC+HRA+DA) AS "AVERAGE PAY" FROM TEACHERS;
+-----------+-------------+
| TOTAL PAY | AVERAGE PAY |
+-----------+-------------+
|    222480 |  24720.0000 |
+-----------+-------------+
1 row in set (0.01 sec)
```

(c) To display the names and dept of all staff who doesn't have 'a' in their name.

```
mysql> SELECT TNAME, DEPT FROM TEACHERS WHERE TNAME NOT LIKE '%A%' ;
+-------+-----------+
| TNAME | DEPT      |
+-------+-----------+
| JOHN  | PHYSICS   |
| NUPUR | ENGLISH   |
| BINOY | ECONOMICS |
+-------+-----------+
3 rows in set (0.01 sec)
```

(d) To display the SID and Department of the teachers in accordance to the seniority based on DOB.

```
mysql> SELECT SID, DEPT FROM TEACHERS ORDER BY DOB;
+------+-----------+
| SID  | DEPT      |
+------+-----------+
|  105 | ACCOUNTS  |
|  117 | COMPUTER  |
|  109 | PHYSICS   |
|  101 | ACCOUNTS  |
|  114 | ENGLISH   |
|  130 | COMPUTER  |
|  111 | ECONOMICS |
|  104 | PHYSICS   |
|  107 | CHEMISTRY |
+------+-----------+
9 rows in set (0.00 sec)
```

(e) To display the report containing SID, Salary (BASIC + HRA + DA) and PF (BASIC*0.12).

```
mysql> SELECT SID, BASIC+HRA+DA  AS "SALARY", BASIC*0.12 AS PF FROM TEACHERS;
+------+--------+---------+
| SID  | SALARY | PF      |
+------+--------+---------+
|  101 |  13300 | 1440.00 |
|  104 |  25800 | 2760.00 |
|  107 |  36500 | 3840.00 |
|  114 |  15200 | 1440.00 |
|  109 |  45900 | 5040.00 |
|  105 |  20890 | 2268.00 |
|  117 |  17830 | 1836.00 |
|  111 |  22460 | 2376.00 |
|  130 |  24600 | 2604.00 |
+------+--------+---------+
9 rows in set (0.00 sec)
```

(f) To display the details of the teachers who have their DOB in 1975.

```
mysql> SELECT * FROM TEACHERS WHERE DOB LIKE '1975%' ;
+------+-----------+----------+------------+-------+------+------+
| SID  | TNAME     | DEPT     | DOB        | BASIC | HRA  | DA   |
+------+-----------+----------+------------+-------+------+------+
|  101 | SIDDHARTH | ACCOUNTS | 1975-12-25 | 12000 | 1000 |  300 |
|  109 | JANVI     | PHYSICS  | 1975-05-01 | 42000 | 2700 | 1200 |
+------+-----------+----------+------------+-------+------+------+
2 rows in set (0.00 sec)
```

(g) To display the no. of teachers in each department.

```
mysql> SELECT DEPT, COUNT(*) FROM TEACHERS GROUP BY DEPT;
+-----------+----------+
| DEPT      | COUNT(*) |
+-----------+----------+
| ACCOUNTS  |        2 |
| PHYSICS   |        2 |
| CHEMISTRY |        1 |
| ENGLISH   |        1 |
| COMPUTER  |        2 |
| ECONOMICS |        1 |
+-----------+----------+
6 rows in set (0.01 sec)
```

(h) SELECT MAX(DOB), MIN(DOB) from Teacher;

```
mysql> SELECT MAX(DOB), MIN(DOB) from TEACHERS;
+------------+------------+
| MAX(DOB)   | MIN(DOB)   |
+------------+------------+
| 1994-08-12 | 1964-06-21 |
+------------+------------+
1 row in set (0.00 sec)
```

(i) SELECT TNAME FROM TEACHER WHERE DEPT IN ('ENGLISH', 'ECONOMICS, ACCOUNTS') AND YEAR (DOB) BETWEEN 1965 AND 1990;

```
mysql> SELECT TNAME FROM TEACHERS WHERE DEPT IN('ENGLISH','ECONOMICS','ACCOUNTS')AND YEAR(DOB) BETWEEN 1965 AND 1990;
+-----------+
| TNAME     |
+-----------+
| SIDDHARTH |
| NUPUR     |
| BINOY     |
+-----------+
3 rows in set (0.00 sec)
```

(j) SELECT DISTINCT (DEPT) FROM TEACHER;

```
mysql> SELECT DISTINCT (DEPT) FROM TEACHERS;
+-----------+
| DEPT      |
+-----------+
| ACCOUNTS  |
| PHYSICS   |
| CHEMISTRY |
| ENGLISH   |
| COMPUTER  |
| ECONOMICS |
+-----------+
6 rows in set (0.00 sec)
```

SQL-3

Date:

AIM: To create tables FACULTY and COURSES and execute the SQL commands.

QUERY:

Create tables FACULTY and COURSES, write and execute the SQL queries for the given questions and observe the outputs.

Table: FACULTY

F_ID	Fname	Lname	Hire_date	Salary
102	Amit	Mishra	12/10/98	12000
103	Nitin	Vyas	24/12/94	8000
104	Rakshit	Soni	18/05/96	14000
105	Rashmi	Malhotra	11/09/97	11000
106	Sulekha	Srivastava	05/06/99	10000

Table: COURSES

C_ID	F_ID	Cname	Fees
C21	102	Grid Computing	40000
C22	106	System Design	16000
C23	104	Computer Security	8000
C24	106	Human Biology	15000
C25	102	Computer Network	20000
C26	105	Visual Basic	6000

PROCEDURE:

1. Open mysql software.
2. Enter the password.
3. Enter the following commands in the mysql prompt, mysql >
4. Create Database
5. Create table FACULTY(F_ID integer, Fname char(25), Lname char(25), Hire_Date date, Salary integer.
6. Insert into FACULTY values(102,'Amit','Mishra',12/10/98,12000);

(Repeat this command with different values to insert all the records in the table FACULTY)

7. Create table COURSES(C_ID char(5), F_ID integer, Cname char(25), Fees integer);
8. Insert into COURSES values('C21',102,'Grid Computing', 40000);

(Repeat this command with different values to insert all the records in the table COURSES

SQL Commands:

1. **To display details of those faculties whose salary is greater than 12000.**

```
mysql> Select * from FACULTY where Salary > 12000;
+------+---------+-------+------------+--------+
| F_ID | Fname   | Lname | Hire_date  | Salary |
+------+---------+-------+------------+--------+
|  104 | Rakshit | Soni  | 1996-05-18 |  14000 |
+------+---------+-------+------------+--------+
1 row in set (0.01 sec)
```

2. **To display the details of courses whose fees is in the range of 15000 to 50000(both values are included)**

```
mysql> Select * from COURSES where Fees between 15000 and 50000;
+------+------+------------------+-------+
| C_ID | F_ID | Cname            | Fees  |
+------+------+------------------+-------+
| C21  |  102 | Grid Computing   | 40000 |
| C22  |  106 | System Design    | 16000 |
| C24  |  106 | Human Biology    | 15000 |
| C25  |  102 | Computer Network | 20000 |
+------+------+------------------+-------+
4 rows in set (0.00 sec)
```

3. **To increase the fees of all courses by 500 of the course whose Cnmae starts with “Computer”.**

```
mysql> UPDATE COURSES set Fees = Fees + 500 where Cname like "Computer%";
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> Select * from Courses;
+------+------+--------------------+-------+
| C_ID | F_ID | Cname              | Fees  |
+------+------+--------------------+-------+
| C21  |  102 | Grid Computing     | 40000 |
| C22  |  106 | System Design      | 16000 |
| C23  |  104 | Computer Secuirty  |  8500 |
| C24  |  106 | Human Biology      | 15000 |
| C25  |  102 | Computer Network   | 20500 |
| C26  |  105 | Visual Basic       |  6000 |
+------+------+--------------------+-------+
6 rows in set (0.00 sec)
```

4. **To display details of those courses which are taught by 'Sulekha' in desending order of courses.**

```
mysql> SELECT C_ID,F.F_ID,Cname,Fname,Lname,Fees from COURSES C, FACULTY F where Fname = "Sulekha" and C.F_ID = F.F_ID order by Cname desc;
+------+------+--------------+---------+----------+-------+
| C_ID | F_ID | Cname        | Fname   | Lname    | Fees  |
+------+------+--------------+---------+----------+-------+
| C22  |  106 | System Design | Sulekha | Srivatsa | 16000 |
| C24  |  106 | Human Biology | Sulekha | Srivatsa | 15000 |
+------+------+--------------+---------+----------+-------+
2 rows in set (0.00 sec)
```

5. **To display the number of make F_ID in the COURSES table.**

```
mysql> Select Count(Distinct(F_ID)) as "No. of F_ID" from COURSES;
+--------------+
| No. of F_ID  |
+--------------+
|            4 |
+--------------+
1 row in set (0.00 sec)
```

6. **To display the name of the faculty who gets maximum salary.**

```
mysql> Select Fname, Lname from FACULTY where Salary = (Select Max(Salary) from FACULTY);
+---------+-------+
| Fname   | Lname |
+---------+-------+
| Rakshit | Soni  |
+---------+-------+
1 row in set (0.00 sec)
```

7. **To display the faculty name with the name of the courses taught by them.**

```
mysql> Select Fname,Lname,Cname from Faculty,Courses where Faculty.F_ID = COURSES.F_ID;
+---------+-----------+--------------------+
| Fname   | Lname     | Cname              |
+---------+-----------+--------------------+
| Amit    | Mishra    | Grid Computing     |
| Sulekha | Srivastava | System Design     |
| Rakshit | Soni      | Computer Secuirty  |
| Sulekha | Srivastava | Human Biology     |
| Amit    | Mishra    | Computer Network   |
| Rashmi  | Malhotra  | Visual Basic       |
+---------+-----------+--------------------+
6 rows in set (0.00 sec)
```

RESULT:

The SQL commands for the given queries are executed successfully and the outputs are observed.

SQL-4

Date:

AIM: To create DATABASE,TABLES and execute the SQL commands according to the following Query.

QUERY:Create tables CABHUB and CUSTOMER , write and execute the SQL queries for the given questions and observe the outputs.

Table: CABHUB

Vcode	Vehicle Name	Make	Color	Capacity	Charges
100	Innova	Toyota	WHITE	7	15
102	SX4	Suzuki	BLUE	4	14
104	C Class	Mercedes	RED	4	35
105	A-Star	Suzuki	WHITE	3	14
108	Indigo	Tata	SILVER	3	12

Table: CUSTOMER

Ccode	CName	Vcode
1	Hemant sahu	101
2	Raj lal	108
3	Feroza Shah	105
4	Ketan Dhal	104

PROCEDURE:

1. Open mysql software.

2. Enter the password.

3. Enter the following commands in the mysql prompt, mysql >

4. Create Database

5. Create table CABHUB(Vcode integer, VehicleName char(20),Make char(10),Color char(10),Capacity integer,Charges integer);

6. Insert into CABHUB values(101,'Innova','Toyota','WHITE',7,15);

(Repeat this command with different values to insert all the records in the table CABHUB

7. Create table CUSTOMER(Ccode integer,Cname varchar(25),Vcode integer);

8. Insert into CUSTOMER values(1,'Hemant sahu',101);

(Repeat this command with different values to insert all the records in the table CUSTOMER)

SQL Commands:

1. **To display all white coloured vehicles.**

```
mysql> SELECT * from CABHUB where Color = "WHITE";
+-------+--------------+--------+-------+----------+---------+
| Vcode | Vehicle_Name | Make   | Color | Capacity | Charges |
+-------+--------------+--------+-------+----------+---------+
|   100 | Innova       | Toyota | WHITE |        7 |      15 |
|   105 | A-Star       | Suzuki | WHITE |        3 |      14 |
+-------+--------------+--------+-------+----------+---------+
2 rows in set (0.00 sec)
```

2. **To display the VehicleName , Make and Capacity in ascending order of their seating capacity.**

```
mysql> SELECT Vehicle_Name,Make,Capacity from CABHUB ORDER BY Capacity;
+--------------+----------+----------+
| Vehicle_Name | Make     | Capacity |
+--------------+----------+----------+
| A-Star       | Suzuki   |        3 |
| Indigo       | Tata     |        3 |
| SX4          | Suzuki   |        4 |
| C Class      | Mercedes |        4 |
| Innova       | Toyota   |        7 |
+--------------+----------+----------+
5 rows in set (0.00 sec)
```

3. **To display the highest charges at which a vehicle can be hired from CABHUB.**

```
mysql> Select Max(Charges) as "Highest Charges" from CABHUB;
+-----------------+
| Highest Charges |
+-----------------+
|              35 |
+-----------------+
1 row in set (0.00 sec)
```

4. **To display the customer name and corresponding name of the vehicle hired by them.**

```
mysql> SELECT Cname,Vehicle_name from CUSTOMER,CABHUB where CUSTOMER.Vcode = CABHUB.Vcode;
+-------------+--------------+
| Cname       | Vehicle_name |
+-------------+--------------+
| Raj lal     | Indigo       |
| Feroza Shah | A-Star       |
| Ketan Dhal  | C Class      |
+-------------+--------------+
3 rows in set (0.00 sec)
```

5. **To display the number of make vehicle in the table CABHUB.**

```
mysql> SELECT Count(DISTINCT(Make)) from CABHUB;
+------------------------+
| Count(DISTINCT(Make)) |
+------------------------+
|                      4 |
+------------------------+
1 row in set (0.00 sec)
```

6. **To display the vehicle name, Name of the customer who has hired at the lowest charge.**

```
mysql> SELECT Vehicle_Name, Cname from CABHUB,CUSTOMER where CABHUB.Vcode = CUSTOMER.Vcode and Charges = (SELECT Min(Charges) from CABHUB);
+--------------+---------+
| Vehicle_Name | Cname   |
+--------------+---------+
| Indigo       | Raj lal |
+--------------+---------+
1 row in set (0.00 sec)
```

7. **To display the number of vehicles based on color from each other.**

```
mysql> Select Color, count(Vcode) as "No. of Vehicles" from CABHUB GROUP BY Color;
+--------+-----------------+
| Color  | No. of Vehicles |
+--------+-----------------+
| WHITE  |               2 |
| BLUE   |               1 |
| RED    |               1 |
| SILVER |               1 |
+--------+-----------------+
4 rows in set (0.01 sec)
```

8. **To display the VehicleName whose capacity is 4.**

```
mysql> SELECT Vehicle_Name from CABHUB where Capacity = 4;
+--------------+
| Vehicle_Name |
+--------------+
| SX4          |
| C Class      |
+--------------+
2 rows in set (0.00 sec)
```

RESULT:

Thus, the SQL commands are executed successfully using MYSQL.

SQL 5

Write queries (a) to (d) based on the tables **EMPLOYEE** and **DEPARTMENT** given below:

Table: EMPLOYEE

EMPID	NAME	DOB	DEPTID	DESIG	SALARY
120	Alisha	23-Jan-1978	D001	Manager	75000
123	Nitin	10-Oct-1977	D002	AO	59000
129	Navjot	12-Jul-1971	D003	Supervisor	40000
130	Jimmy	30-Dec-1980	D004	Sales Rep	
131	Faiz	06-Apr-1984	D001	Dep Manager	65000

Table: DEPARTMENT

DEPTID	DEPTNAME	FLOORNO
D001	Personal	4
D002	Admin	10
D003	Production	1
D004	Sales	3

(a) Create a Employee table with the attributes where the Empid is the primary key, and Department table with attributes where the DEPTID is the primary key.

(b) Insert the details with the above tables.

(c) To display the average salary of all employees, department wise.

```
mysql> SELECT DEPARTMENT.DEPTID,DEPARTMENT.DEPTNAME,avg(SALARY) from DEPARTMENT,EMPLOYEE where
 EMPLOYEE.DEPTID = DEPARTMENT.DEPTID GROUP BY DEPARTMENT.DEPTID;
+--------+------------+-------------+
| DEPTID | DEPTNAME   | avg(SALARY) |
+--------+------------+-------------+
| D001   | Personel   |       70000 |
| D002   | Admin      |       59000 |
| D003   | Production |       40000 |
| D004   | Sales      |        NULL |
+--------+------------+-------------+
4 rows in set (0.05 sec)
```

(d) To display name and respective department name of each employee whose salary is more than 50000.

```
mysql> SELECT NAME,DEPTNAME from EMPLOYEE,DEPARTMENT where SALARY > 50000 and DEPARTMENT.DEPTID = EMPLOYEE.DEPTID;
+--------+----------+
| NAME   | DEPTNAME |
+--------+----------+
| Alisha | Personel |
| Nitin  | Admin    |
| Faiz   | Personel |
+--------+----------+
3 rows in set (0.00 sec)
```

(e) To display the names of employees whose salary is not known, in alphabetical order.

```
mysql> SELECT NAME from EMPLOYEE where SALARY is null ORDER BY NAME ;
+-------+
| NAME  |
+-------+
| Jimmy |
+-------+
1 row in set (0.01 sec)
```

(f) To display DEPTID from the table **EMPLOYEE** without repetition.

```
mysql> SELECT DISTINCT(DEPTID) from EMPLOYEE;
+--------+
| DEPTID |
+--------+
| D001   |
| D002   |
| D003   |
| D004   |
+--------+
4 rows in set (0.00 sec)
```