SET 1

I. LAB TEST

1. Write a Python program to store book records in a binary file named **BOOK.DAT.** Each book record should contain the following details: **[BookID, BookName, Author, Price]** After storing the data, define a function **countRecords()** that performs the following operations.(Use the **pickle module** to perform binary file operations)

   1. Displays the details of all books whose price is greater than **Rs. 5000** and counts the number of such books.

   2. Displays the total number of all books stored in the file.

8

```
import pickle
def addBooks():
    file = open("BOOK.DAT", "wb")
    n = int(input("Enter number of books: "))
    print("Enter the book details: ")
    for i in range(n):
        print("\nEnter book details")
        bid = int(input("Book ID: "))
        bname = input("Book Name: ")
        author = input("Author: ")
        price = float(input("Price: "))
        book = [bid, bname, author, price]
        pickle.dump(book, file)
    file.close()
def countRecords():
    file = open("BOOK.DAT", "rb")
    count = 0
    total = 0
    print("\nBooks with price more than Rs. 5000:")
    try:
        while True:
            book = pickle.load(file)
            total += 1
            if book[3] > 5000:
                print(book)
                count += 1
    except EOFError:
        pass
    file.close()
    print("\nNumber of books costing more than Rs. 5000:", count)
    print("Total number of books:", total)
addBooks()
countRecords()
```

2. Consider the following tables **BOOK** and **MEMBER** and answer the questions that follow:

TABLE : BOOK

BookID	BookName	Author	Price
B101	Python Basics	A. Sharma	650
B102	SQL Made Easy	R. Verma	1200
B103	Data Science	S. Mehta	1800
B104	Web Design	K. Rao	900

TABLE: MEMBER

MemID	MName	BookID
201	Rohit Kumar	B101
202	Neha Singh	B103
203	Aman Gupta	B102

4

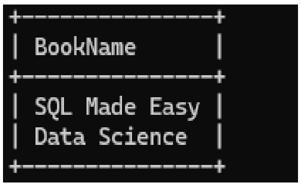
Write SQL commands for the following:

a) To display the names of all books whose **price is more than 1000**.
b) To display **BookName and Author** of books in **ascending order of Price**.
c) To display the **highest price** of a book from the BOOK table.
d) To display the **Member name** and the **Book name** issued to them

Queries

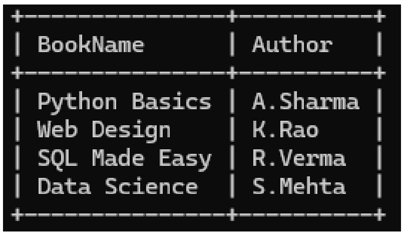
```
mysql> create database;
mysql> use book;
mysql> create table book(BookId varchar(5) primary key,BookName varchar(20),Author
varchar(20),price integer);

mysql> insert into book values('B101','Python Basics','A.Sharma',650),
          ('B102','SQL Made       Easy','R.Verma',1200),
          ('B103','Data Science','S.Mehta',1800),('B104','Web Design','K.Rao',900);

mysql> create table member(MemberID integer primary key,MName varchar(20),BookId
    varchar(5));

mysql> insert into member values(201,'Rohit Kumar','B101'),
          (202,'Neha   Singh','B103'),(203,'Aman Gupta','B102');
```

a) **SELECT BookName FROM BOOK WHERE Price > 1000;**



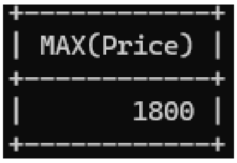
```
+---------------+
| BookName      |
+---------------+
| SQL Made Easy |
| Data Science  |
+---------------+
```

b) **SELECT BookName, Author FROM BOOK ORDER BY Price ASC;**



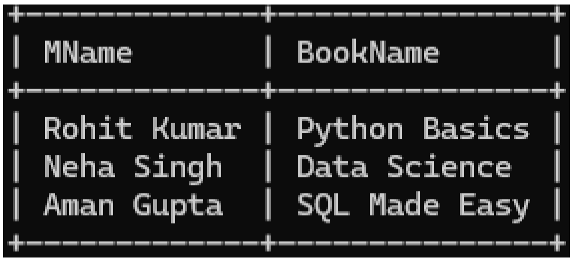
```
+---------------+----------+
| BookName      | Author   |
+---------------+----------+
| Python Basics | A.Sharma |
| Web Design    | K.Rao    |
| SQL Made Easy | R.Verma  |
| Data Science  | S.Mehta  |
+---------------+----------+
```

c) **SELECT MAX(Price) FROM BOOK;**



```
+------------+
| MAX(Price) |
+------------+
|       1800 |
+------------+
```

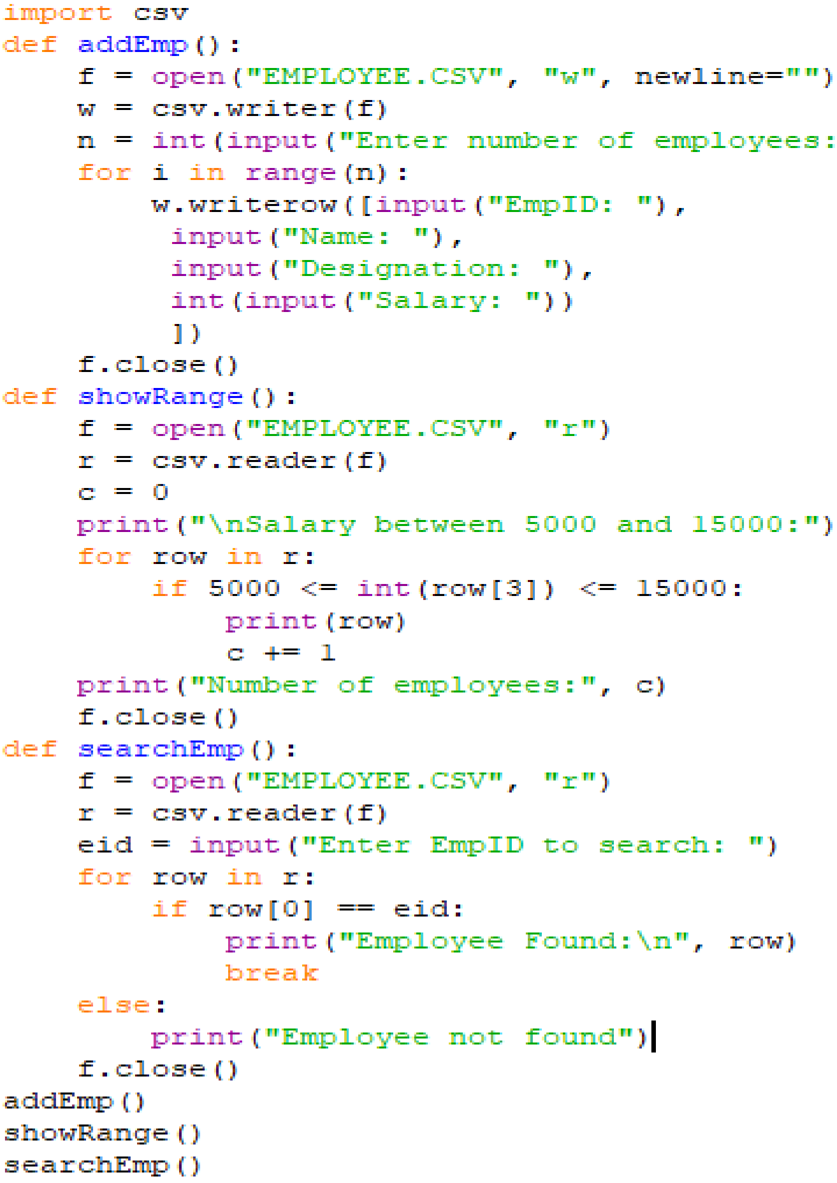
d) **SELECT MEMBER.MName, BOOK.BookName FROM MEMBER, BOOK WHERE MEMBER.BookID = BOOK.BookID;**



```
+-------------+---------------+
| MName       | BookName      |
+-------------+---------------+
| Rohit Kumar | Python Basics |
| Neha Singh  | Data Science  |
| Aman Gupta  | SQL Made Easy |
+-------------+---------------+
```

SET 2

I.	**LAB TEST**	
1.	Write a Python program to store employee records in a **CSV** file named **EMPLOYEE.CSV**. Each employee record should contain the following details:**[EmpID, EmpName, Desig, Salary]** After storing the data, define appropriate functions to perform the following operations: **1.** Display the details of employees whose salary lies in the range of **Rs. 5000 to Rs. 15000** and count the number of such employees. **2.** Search and display the details of an employee using **EmpID entered by the user**.	8



```
import csv
def addEmp():
    f = open("EMPLOYEE.CSV", "w", newline="")
    w = csv.writer(f)
    n = int(input("Enter number of employees: "))
    for i in range(n):
        w.writerow([input("EmpID: "),
         input("Name: "),
         input("Designation: "),
         int(input("Salary: "))
         ])
    f.close()
def showRange():
    f = open("EMPLOYEE.CSV", "r")
    r = csv.reader(f)
    c = 0
    print("\nSalary between 5000 and 15000:")
    for row in r:
        if 5000 <= int(row[3]) <= 15000:
            print(row)
            c += 1
    print("Number of employees:", c)
    f.close()
def searchEmp():
    f = open("EMPLOYEE.CSV", "r")
    r = csv.reader(f)
    eid = input("Enter EmpID to search: ")
    for row in r:
        if row[0] == eid:
            print("Employee Found:\n", row)
            break
    else:
        print("Employee not found")
    f.close()
addEmp()
showRange()
searchEmp()
```

2. Observe the following tables Employee Details and Employee Salary and write queries given below::

TABLE: EMPLOYEE DETAILS

EmpID	FullName	ManagerID	Date of Joining	City
121	John Snow	321	31-01-2014	Toronto
321	Walter White	986	30-01-2015	California
421	Kuldeep Rana	876	27-11-2016	New Delhi

TABLE: EMPLOYEE SALARY

EmpID	Project	Salary	Variable
121	P1	8000	500
321	P2`	10000	1000
421	P1	12000	0

4

Write SQL queries for the following:

1. Write an SQL query to fetch the **EmpID** and **FullName** of all the employees working under Manager with id – **'986'**.
2. Write an SQL query to fetch the different projects available from the Employee_Salary table.
3. Write an SQL query to find the maximum, minimum and average salary of the employees.
4. Write an SQL query to find the employee id whose salary lies in the range **of 9000 & 15000**.

Queries

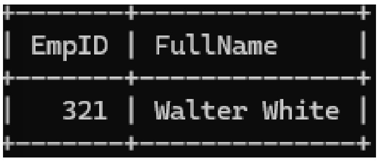
mysql> create table Employee_Details(EmpId varchar(5) primary key,FullName varchar(20),ManagerID varchar(20), Date_of_Joining date,City Varchar(2));

mysql> insert into Employee_Details values(121,'John Snow','321','2014-01-31','Toronto'),(321,'Walter White','986','2015-01-30','California'),(421,'Kuldeep Rana','876','2016-11-27','New Delhi');

mysql> create table Employee_Salary (EmpId varchar(5) primary key,Project varchar(20),Salary integer,Variable integer);

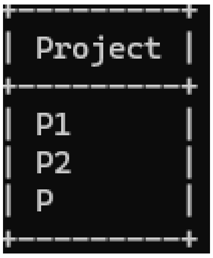
mysql> insert into Employee_Salary values(121,'P1',8000,500),(321,'P2',10000,1000),(421,'P',12000,0);

1. **SELECT EmpID, FullName FROM Employee_Details WHERE ManagerID = 986;**



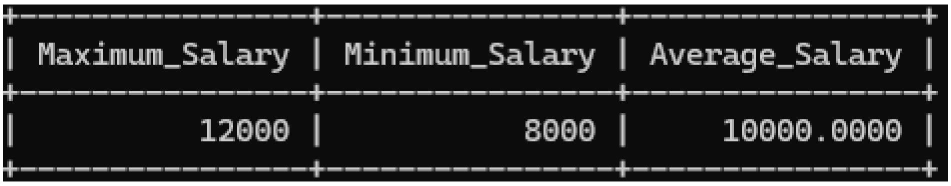
```
+-------+--------------+
| EmpID | FullName     |
+-------+--------------+
|   321 | Walter White |
+-------+--------------+
```

2. **SELECT DISTINCT Project FROM Employee_Salary;**



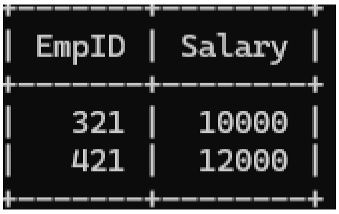
```
+---------+
| Project |
+---------+
| P1      |
| P2      |
| P       |
+---------+
```

3. **SELECT MAX(Salary) AS Maximum_Salary,MIN(Salary) AS Minimum_Salary, AVG(Salary) AS Average_Salary FROM Employee_Salary;**



```
+----------------+----------------+----------------+
| Maximum_Salary | Minimum_Salary | Average_Salary |
+----------------+----------------+----------------+
|          12000 |           8000 |     10000.0000 |
+----------------+----------------+----------------+
```

4. **SELECT EmpID, Salary FROM Employee_Salary WHERE Salary BETWEEN 9000 AND 15000;**



```
+-------+--------+
| EmpID | Salary |
+-------+--------+
|   321 |  10000 |
|   421 |  12000 |
+-------+--------+
```