

SECTION-D

32.

Consider the table ORDERS as given below

O_Id	C_Name	Product	Quantity	Price
1001	Jitendra	Laptop	1	12000
1002	Mustafa	Smartphone	2	10000
1003	Dhwani	Headphone	1	1500

Note: The table contains many more records than shown here.

A) Write the following queries:

- (I) To display the total Quantity for each Product, excluding Products with total Quantity less than 5.

```
SELECT SUM(Quantity) "Total"  
FROM ORDERS  
GROUP BY Products  
WHERE Total<5;
```

- (II) To display the orders table sorted by total price in descending order.

```
SELECT SUM(Price) "TPrice"  
FROM ORDERS  
ORDER BY TPrice DESC;
```

- (III) To display the distinct customer names from the Orders table.

```
SELECT DISTINCT(C_Name)  
FROM ORDERS;
```

- (IV) Display the sum of Price of all the orders for which the quantity is null.

```
SELECT SUM(Price)  
FROM ORDERS  
WHERE Quantity IS NULL;
```

OR

B) Write the output

```
(I) Select c_name, sum(quantity) as  
total_quantity from orders group by c_name;
```

c_name	total_quantity
Jitendra	1
Mustafa	2
Dhwani	1

(II) `Select * from orders where product like '%phone%';`

O_Id	C_Name	Product	Quantity	Price
1002	Mustafa	Smartphone	2	10000
1003	Dhwani	Headphone	1	1500

(III) `Select o_id, c_name, product, quantity, price from orders where price between 1500 and 12000;`

O_Id	C_Name	Product	Quantity	Price
1001	Jitendra	Laptop	1	12000
1002	Mustafa	Smartphone	2	10000
1003	Dhwani	Headphone	1	1500

(IV) `Select max(price) from orders;`
`max(price)`
`12000`

33.

A csv file "Happiness.csv" contains the data of a survey. Each record of the file contains the following data:

- Name of a country
- Population of the country
- Sample Size (*Number of persons who participated in the survey in that country*)
- Happy (*Number of persons who accepted that they were Happy*)

For example, a sample record of the file may be:

`['Signiland', 5673000, 5000, 3426]`

Write the following Python functions to perform the specified operations on this file:

- (I) Read all the data from the file in the form of a list and display all those records for which the population is more than 5000000.

```
def show():
import csv
f=open("happiness.csv",'r')
rec=csv.reader(f)
for i in rec:
    if int(i[1])>5000000:
        print(i)
f.close()
```

- (II) Count the number of records in the file.

```
def Count_records():
import csv
```

```

f=open("happiness.csv",'r')
rec=csv.reader(f)
count=0
for i in rec:
    count+=1
    print(count)
f.close()

```

34.

Saman has been entrusted with the management of Law University Database. He needs to access some information from FACULTY and COURSES tables for a survey analysis. Help him extract the following information by writing the desired SQL queries as mentioned below.

Table: **FACULTY**

F_ID	FName	LName	Hire_Date	Salary
102	Amit	Mishra	12-10-1998	12000
103	Nitin	Vyas	24-12-1994	8000
104	Rakshit	Soni	18-5-2001	14000
105	Rashmi	Malhotra	11-9-2004	11000
106	Sulekha	Srivastava	5-6-2006	10000

Table: **COURSES**

C_ID	F_ID	CName	Fees
C21	102	Grid Computing	40000
C22	106	System Design	16000

(I) To display complete details (from both the tables) of those Faculties whose salary is less than 12000.

```

SELECT *
FROM FACULTY, COURSES
WHERE Salary<12000
AND Faculty.F_ID=Courses.F_ID;

```

(II) To display the details of courses whose fees is in the range of 20000 to 50000 (both values included).

```

SELECT *
FROM COURSES
WHERE Fees BETWEEN 20000 AND 50000;

```

(III) To increase the fees of all courses by 500 which have "Computer" in their Course names.

```

UPDATE Courses

```

SET Fees=Fess+500
WHERE CName LIKE'%Computer%';

(IV) (A) To display names (FName and LName) of faculty taking System Design.

SELECT Fname, LName
FROM FACULTY, COURSES
WHERE CName='System Design'
AND Faculty.F_ID=Courses.F_ID;

OR

(B) To display the Cartesian Product of these two tables.

F_ID	FName	LName	Hire_Date	Salary	C_ID	F_ID	CName	Fees
102	Amit	Mishra	12-10-1998	12000	C21	102	Grid Computing	40000
102	Amit	Mishra	12-10-1998	12000	C22	106	System Design	16000
103	Nitin	Vyas	24-12-1994	8000	C21	102	Grid Computing	40000
103	Nitin	Vyas	24-12-1994	8000	C22	106	System Design	16000
105	Rashmi	Malhotra	11-9-2004	11000	C21	102	Grid Computing	40000
105	Rashmi	Malhotra	11-9-2004	11000	C22	106	System Design	16000
106	Sulekha	Srivastava	5-6-2006	10000	C21	102	Grid Computing	40000
106	Sulekha	Srivastava	5-6-2006	10000	C22	106	System Design	16000