Practical file

For

AISSCE 2022-23 Examination

As a part of the Informatics Practices (065)

SUBMITTED BY

SACHIN

Roll No. 17691728

Under the Guidance of

Dr Sunil Kumar Vats PGT (Computer Science)

Python Programs

1. Program to add two numbers.

```
x = int(input('Enter integer value:'))
print('Value entered:', x)
print('Type:', type(x))
```

2. Program to find simple interest.

```
p = float(input('Enter principle'))
r = float(input('Enter rate'))
t = float(input('Enter time'))
i=p*r*t/100
print("Interest=",i)
```

3. Program to check whether a number is even or odd.

```
a=float(input('Enter a number'))
if a%2==0:
    print(a,"is even")
else:
    print(a,"is odd")
```

4. Program to check whether a number is divisible by 2 or 3 using nested if.

```
num=float(input('Enter a number'))
if num%2==0:
    if num%3==0:
        print ("Divisible by 3 and 2")
    else:
        print ("divisible by 2 not divisible by 3")
else:
    if num%3==0:
        print ("divisible by 3 not divisible by 2")
    else:
        print  ("not Divisible by 2 not divisible by 3")
```

5. Menu based program to find sum, subtraction, multiplication and division of values.

```
print("1. Sum of two numbers")
print("2. Subtaction of two numbers")
print("3. Multiplication of two numbers")
print("4. Division of two numbers")
choice=int(input('Enter your choice'))
if choice==1 :
    a=int(input('Enter first number'))
    b=int(input('Enter second number'))
    c=a+b
    print("Sum=",c)
elif choice==2 :
    a=int(input('Enter first number'))
    b=int(input('Enter second number'))
    c=a-b
    print("Subtraction=",c)
elif choice==3 :
    a=int(input('Enter first number'))
    b=int(input('Enter second number'))
    c=a*b
    print("Multiplication=",c)
elif choice==4 :
    a=int(input('Enter first number'))
    b=int(input('Enter second number'))
    c=a/b
    print("Division=",c)
else :
    print("Wrong choice")
```

6. Program to print result depending upon the percentage.

```
a=int(input('Enter your percentage'))
eligible= a>=33
compartment = a>=20 and a<33
fail=a<20
if eligible :
    print("Pass");
elif compartment :
    print("compartment");
elif fail:
    print("Fail");
```

7. Program to find sum of even numbers from 1 to 7.

```
sum=0
for num in range(8):
    if num%2==0:
        sum=sum+num
print("Sum of even values=",sum)
```

8. Program to find factorial of a number.

```
n=eval(input("Enter a number="))
i=1
f=1
while i<=n:
    f=f*i #1*2*3*4*5
    i=i+1
print("Factorial of",n,"=",f)
```

9. Program to print table of any number.

```
n=eval(input("Enter a number whose table you want="))
i=1
while i<=10:
    print(n,"X",i,"=",n*i)
    i=i+1
```

10. Program to print following pattern.

```
*
**
***
****
*****

for i in range(1,6):
    print()
    for j in range(1,i+1):
        print ('*',end="")
```

11. Program to sort values in a list.

```
aList=[10,5,1,3]
print("Original List",aList)
n=len(aList)
for i  in range(n-1):
    for j in range(0,n-i-1):
        if aList[j]>aList[j+1]:
            aList[j],aList[j+1]=aList[j+1],aList[j]
print("Sorted List",aList)
```

12. Program to find minimum value in a list.

```
L=eval(input('Enter list values'))
length=len(L)  #length=6
min=L[0]   #min=10
```

```
loc=-1
for i in range(length): #0,1,2,3,4,5
    if L[i]<min:
        min=L[i]  #min=1
        loc=i
print("Minimum value=",min)
print("Location=",loc)
```

13. Program to check whether a value exists in dictionary

```
aDict={'Bhavna':1,"Richard":2,"Firoza":3,"Arshnoor":4}
val=eval(input('Enter value'))
flag=0
for k in aDict:
    if val==aDict[k]:
        print("value found at key",k)
        flag=1
if flag==0:
    print("value not found")
```

14. Program to find largest among two numbers using a user defined function .

```
def largest():
    a=int(input("Enter first number="))
    b=int(input("Enter second number="))
    if a>b :
        print ("Largest value=%d"%a)
    else:
        print ("Largest value=%d"%b)
    return
largest()
```

15. Program to find sum of two numbers using a user defined function with parameters.

```
def sum(a,b):  #a and b are formal parameters
    c=a+b
    print("sum=",c)

sum(4,5)
n1,n2=eval(input('Enter two values'))
sum(n1,n2)
```

16. Program to find simple interest using a user defined function with parameters and with return value.

```
def interest(p1,r1,t1 ):
    i=p*r*t/100
    return(i)

p=int(input("Enter principle="))
r=int(input("Enter rate="))
t=int(input("Enter rate="))
in1=interest(p,r,t)
print("Interest=",in1)
```

17. Program to pass a list as function argument and modify it.

```
def changeme( mylist ):
    print ("inside the function before change ", mylist)
    mylist[0]=1000
    print ("inside the function after change ", mylist)
    return

list1 = [10,20,30]
print ("outside function before calling function", list1)
```

```
changeme( list1 )
print ("outside function after calling function", list1)
```

18. Program to use default arguments in a function.

```
def printinfo( name, age = 35 ): #default argument
    print ("Name: ", name)
    print ("Age ", age)
    return

printinfo("aman",45)
printinfo("Parth")
```

19. Program to write rollno, name and marks of a student in a data file Marks.dat.

```
count=int(input('How many students are there in the class'))
fileout=open("Marks.dat","a")
for i in range(count):
    print("Enter details of student",(i+1),"below")
    rollno=int(input("Enter rollno:"))
    name=input("name")
    marks=float(input('marks'))
    rec=str(rollno)+","+name+","+str(marks)+"\n"
    fileout.write(rec)
fileout.close()
```

20. Program to read and display contents of file Marks.dat.

```
fileinp=open("Marks.dat","r")
while str:
    str=fileinp.readline()
    print(str)
fileinp.close()
```

21. Program to read and display those lines from file that start with alphabet ‘T’.

```
file1=open("data.txt","r")
count=0
str1=file1.readlines()
print(str1)
for i in str1:
    if i[0]=='T':
        print (i)
file1.close()
```

22. Program to read and display those lines from file that end with alphabet ‘n’.

```
file1=open("data.txt","r")
count=0
str1=file1.readlines()
for i in str1:
    if i[-2]=='n':
        count+=1
print("Number of lines which end with 'n'=",count)
file1.close()
```

23. Program to count number of words in data file data.txt.

```
file1=open("data.txt","r")
line=" "
count=0
while line:
    line=file1.readline()
    s=line.split()
    for word in s:
        count+=1
```

```
print("Number of words=",count)
file1.close()
```

24. Program to count number of characters in data file data.txt.

```
file1=open("data.txt","r")
ch=" "
count=0
while ch:
    ch=file1.read(1)
    count+=1
print("Number of characters=",count)
file1.close()
```

25. Program to write data in a csv file student.csv.

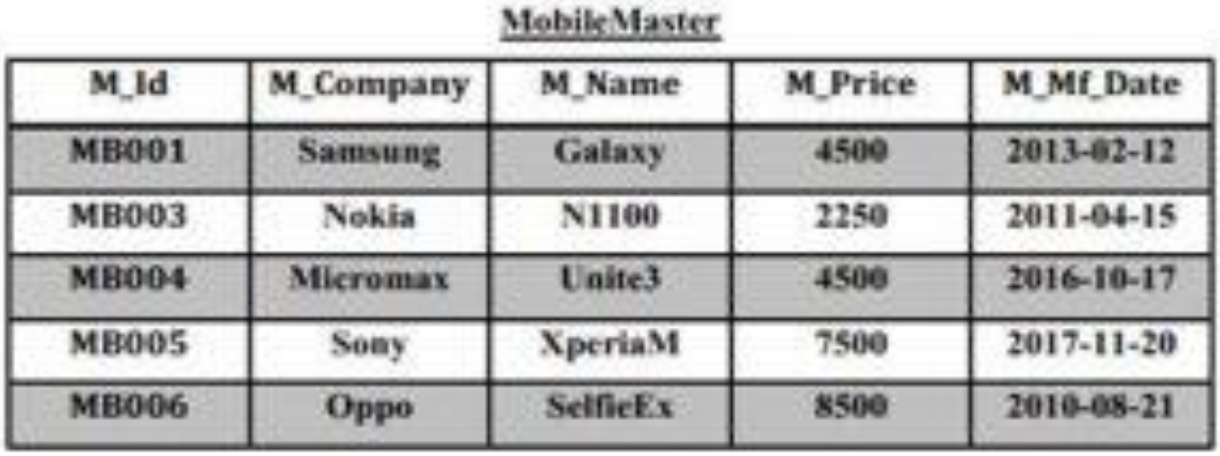
```
import csv
fh=open("d:\student.csv","w")
stuwriter=csv.writer(fh)
stuwriter.writerow([1,'aman',50])
stuwriter.writerow([2,'Raman',60])
fh.close()
```

26. Program to readand display data from a csv file student.csv.

```
import csv
fh=open("d:\student.csv","r")
stureader=csv.reader(fh)
for rec in stureader:
    print(rec)
fh.close()
```

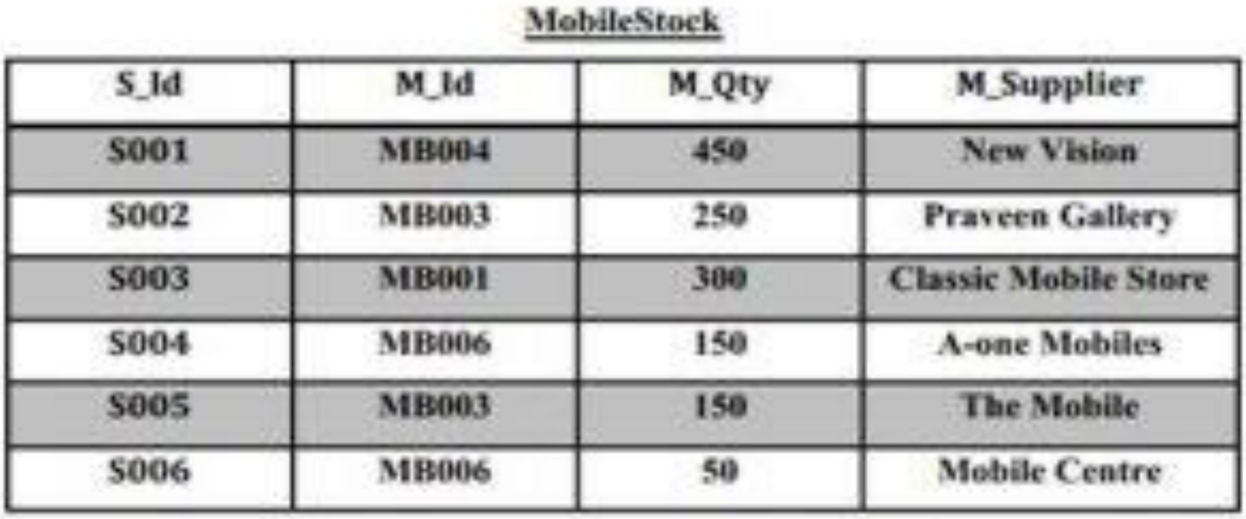
SQL Queries

SQL 1



MobileMaster

M_Id	M_Company	M_Name	M_Price	M_Mf_Date
MB001	Samsung	Galaxy	4500	2013-02-12
MB003	Nokia	N1100	2250	2011-04-15
MB004	Micromax	Unite3	4500	2016-10-17
MB005	Sony	XperiaM	7500	2017-11-20
MB006	Oppo	SelfieEx	8500	2010-08-21



MobileStock

S_Id	M_Id	M_Qty	M_Supplier
S001	MB004	450	New Vision
S002	MB003	250	Praveen Gallery
S003	MB001	300	Classic Mobile Store
S004	MB006	150	A-one Mobiles
S005	MB003	150	The Mobile
S006	MB006	50	Mobile Centre

(i) Display the Mobile company, Mobile name & price in descending order of their manufacturing date.

Ans. SELECT M_Compnay, M_Name, M_Price FROM MobileMaster
ORDER BY M_Mf_Date DESC;

(ii) List the details of mobile whose name starts with “S”.

Ans. SELECT * FROM MobileMaster
WHERE M_Name LIKE “S%‟;

(iii) Display the Mobile supplier & quantity of all mobiles except “MB003‟.

```
SELECT M_Supplier, M_Qty FROM MobileStock
WHERE M_Id <>”MB003”;
```

(iv) To display the name of mobile company having price between 3000 & 5000.

```
Ans. SELECT M_Company FROM MobileMaster
      WHERE M_Price BETWEEN 3000 AND 5000;
```

****Find Output of following queries**

(v) SELECT M_Id, SUM(M_Qty) FROM MobileStock GROUP BY M_Id;

MB004	450
MB003	400
MB003	300
MB003	200

(vi) SELECT MAX(M_Mf_Date), MIN(M_Mf_Date) FROM MobileMaster;

```
      2017-11-20        2010-08-21
```

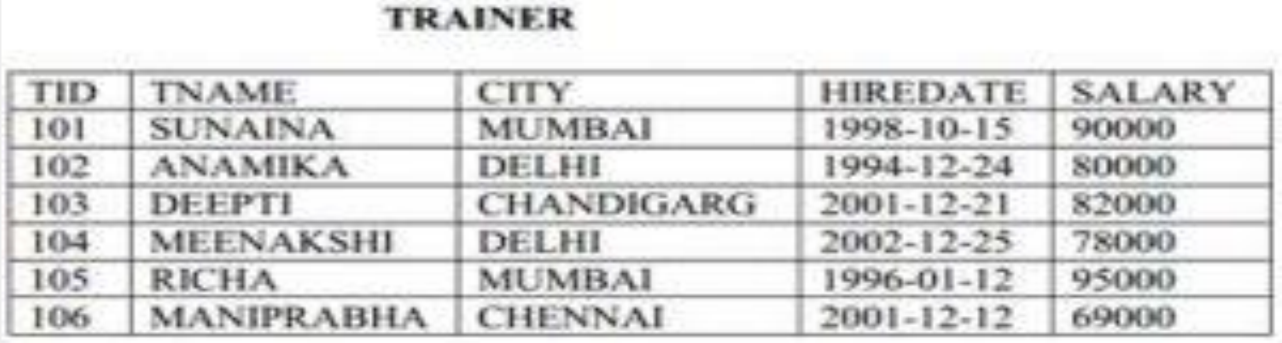
(vii) SELECT M1.M_Id, M1.M_Name, M2.M_Qty, M2.M_Supplier
FROM MobileMaster M1, MobileStock M2
WHERE M1.M_Id=M2.M_Id AND M2.M_Qty>=300;

MB004	Unite3	450	New_Vision
MB001	Galaxy	300	Classic Mobile Store

(viii) SELECT AVG(M_Price) FROM MobileMaster;

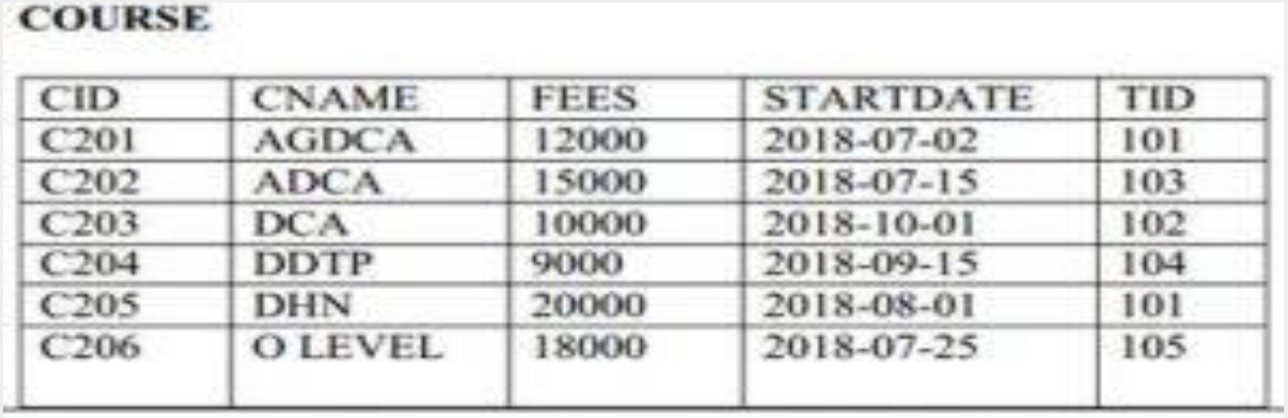
5450

SQL 2



TRAINER

TID	TNAME	CITY	HIREDATE	SALARY
101	SUNAINA	MUMBAI	1998-10-15	90000
102	ANAMIKA	DELHI	1994-12-24	80000
103	DEEPTI	CHANDIGARG	2001-12-21	82000
104	MEENAKSHI	DELHI	2002-12-25	78000
105	RICHA	MUMBAI	1996-01-12	95000
106	MANIPRABHA	CHENNAI	2001-12-12	69000



COURSE

CID	CNAME	FEES	STARTDATE	TID
C201	AGDCA	12000	2018-07-02	101
C202	ADCA	15000	2018-07-15	103
C203	DCA	10000	2018-10-01	102
C204	DDTP	9000	2018-09-15	104
C205	DHN	20000	2018-08-01	101
C206	O LEVEL	18000	2018-07-25	105

i. Display the Trainer Name, City & Salary in descending order of theirHiredate.

Ans. SELECT TNAME, CITY, SALARY FROM TRAINER
ORDER BY HIREDATE;

ii. To display the TNAME and CITY of Trainer who joined the Institute in the month of December 2001.

Ans. SELECT TNAME, CITY FROM TRAINER
WHERE HIREDATE BETWEEN ‘2001-12-01’ AND ‘2001-12-31’;

iii. To display TNAME, HIREDATE, CNAME, STARTDATE from tables TRAINER and COURSE of all those courses whose FEES is less than or equal to 10000.

Ans. SELECT TNAME,HIREDATE,CNAME,STARTDATE FROM
TRAINER, COURSE

```
WHERE TRAINER.TID=COURSE.TID AND FEES<=10000;
```

iv. To display number of Trainers from each city.

```
Ans.  SELECT CITY, COUNT(*) FROM TRAINER GROUP BY CITY;
```

****Find Output of following queries**

v. SELECT TID, TNAME, FROM TRAINER WHERE CITY NOT IN(‘DELHI’, ‘MUMBAI’);

```
Ans.
103   DEEPTI
106   MANIPRABHA
```

vi. SELECT DISTINCT TID FROM COURSE;

```
Ans.
101
103
102
104
105
```

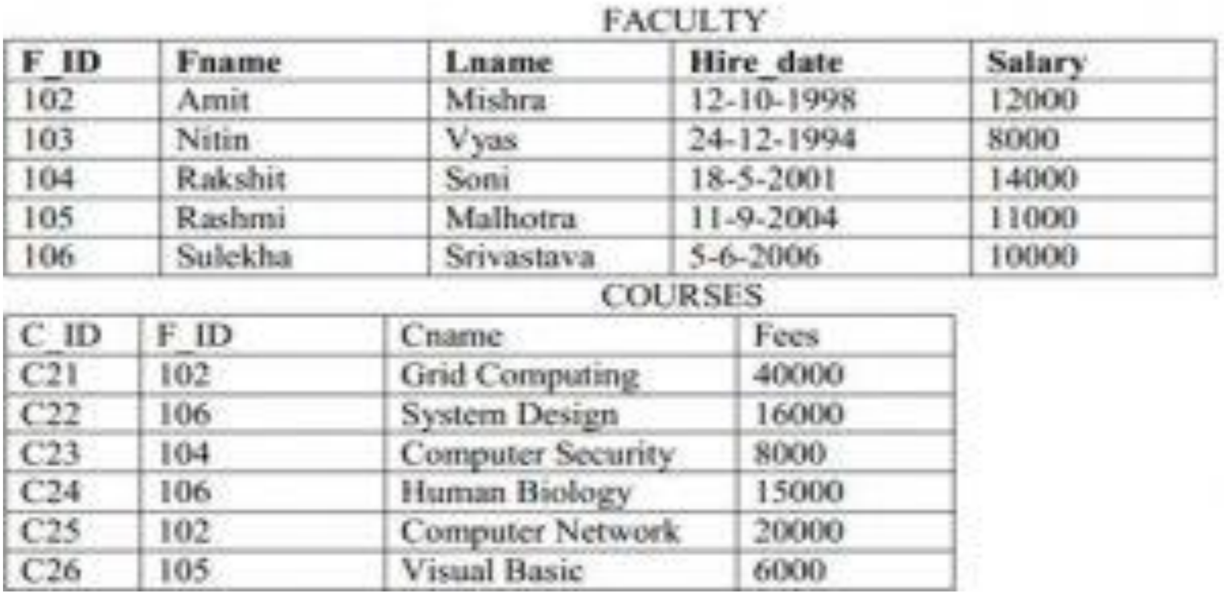
vii. SELECT TID, COUNT(*), MIN(FEES) FROM COURSE GROUP BY TID HAVING COUNT(*)>1;

```
Ans.
101    2    12000
```

viii. SELECT COUNT(*), SUM(FEES) FROM COURSE WHERE STARTDATE< ‘2018-09-15’;

```
Ans.
4    65000
```

SQL 3



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

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

i) To display details of those Faculties whose salary is greater than 12000.

```
Ans: Select * from faculty
where salary > 12000;
```

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

```
Ans: Select * from Courses
where fees between 15000 and 50000;
```

iii) To increase the fees of all courses by 500 of “System Design” Course.

```
Ans: Update courses set fees = fees + 500
where Cname = “System Design”;
```

(iv) To display details of those courses which are taught by ‘Sulekha’ in descending order of courses.

```
Ans: Select * from faculty,courses
where faculty.f_id = course.f_id  and fac.fname = 'Sulekha'
order by cname desc;
```

****Find output of following**

v) Select COUNT(DISTINCT F_ID) from COURSES;

```
Ans: 4
```

vi) Select MIN(Salary) from FACULTY,COURSES where COURSES.F_ID = FACULTY.F_ID;

Ans: 6000

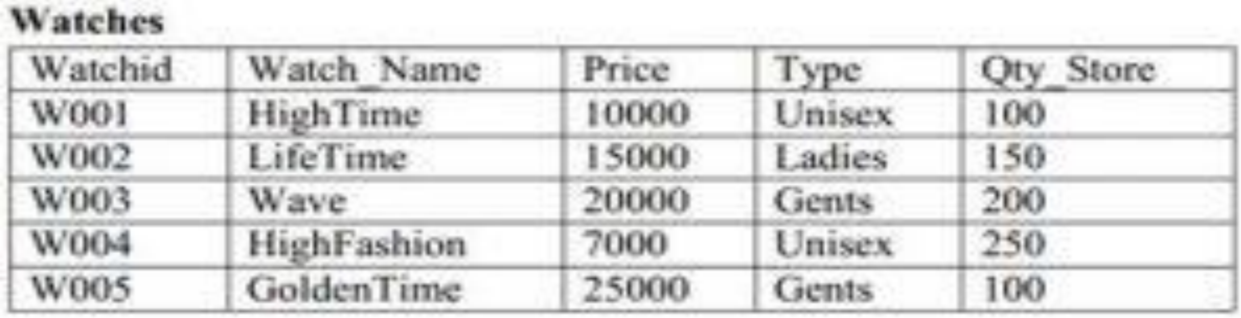
vii) Select sum(fees) from COURSES where F_ID = 102;

Ans: 60000

vii) Select avg(fees) from COURSES;

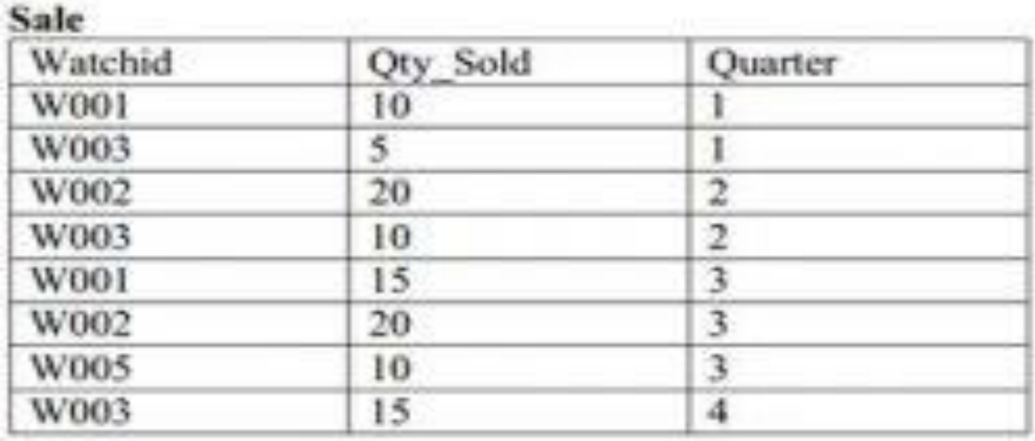
Ans: 17500

SQL 4



Watches

Watchid	Watch_Name	Price	Type	Qty_Store
W001	HighTime	10000	Unisex	100
W002	LifeTime	15000	Ladies	150
W003	Wave	20000	Gents	200
W004	HighFashion	7000	Unisex	250
W005	GoldenTime	25000	Gents	100



Sale

Watchid	Qty_Sold	Quarter
W001	10	1
W003	5	1
W002	20	2
W003	10	2
W001	15	3
W002	20	3
W005	10	3
W003	15	4

i. To display all the details of those watches whose name ends with ‘Time’

Ans select * from watches

where watch_name like ‘%Time’;

ii. To display watch’s name and price of those watches which have price range in between 5000-15000.

Ans. select watch_name, price from watches

where price between 5000 and 15000;

iii. To display total quantity in store of Unisex type watches.

```
Ans. select sum(qty_store) from watches where type like ’Unisex’;
```

iv. To display watch name and their quantity sold in first quarter.

```
Ans. select watch_name,qty_sold from watches w,sale s
where w.watchid=s.watchid and quarter=1;
```

v. select max(price), min(qty_store) from watches;

```
Ans. 25000   100
```

vi. select quarter, sum(qty_sold) from sale group by quarter;

```
1      15
2      30
3      45
4      15
```

vii. select watch_name,price,type from watches w, sales where w.watchid!=s.watchid;

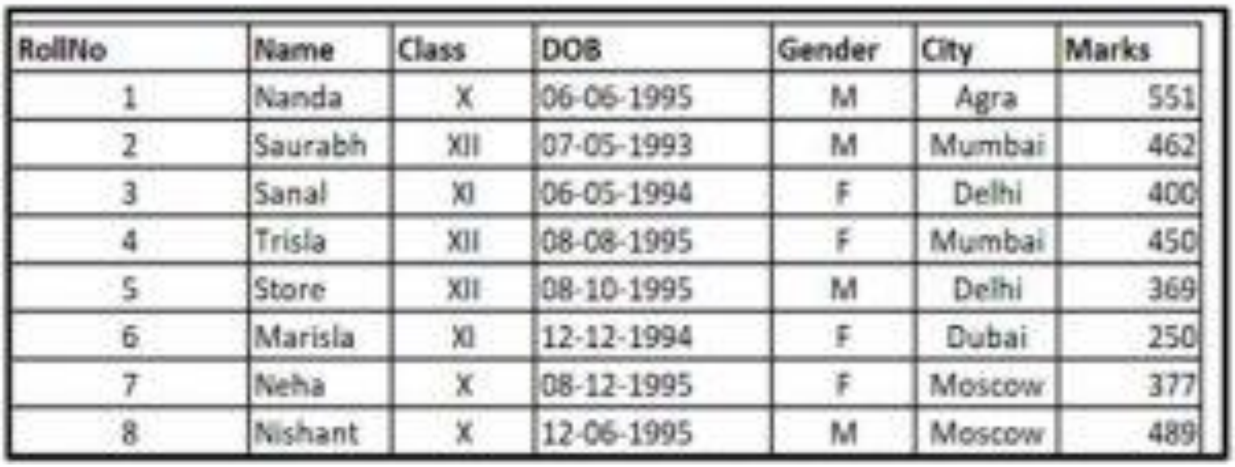
```
HighFashion       7000       Unisex
```

viii. select watch_name, qty_store, sum(qty_sold), qty_store-sum(qty_sold) “Stock” from watches w, sale s where w.watchid=s.watchid group by s.watchid;

```
HighTime      100     25       75
LifeTime      150     40       110
Wave          200     30       170
Golden Time   100     10       90
```

SQL 5

Table : **STUDENT**



RollNo	Name	Class	DOB	Gender	City	Marks
1	Nanda	X	06-06-1995	M	Agra	551
2	Saurabh	XII	07-05-1993	M	Mumbai	462
3	Sanal	XI	06-05-1994	F	Delhi	400
4	Trisla	XII	08-08-1995	F	Mumbai	450
5	Store	XII	08-10-1995	M	Delhi	369
6	Marisla	XI	12-12-1994	F	Dubai	250
7	Neha	X	08-12-1995	F	Moscow	377
8	Nishant	X	12-06-1995	M	Moscow	489

(i) To display the records from table student in alphabetical order as per the name of the student.

Ans. Select * from student

order by name;

(ii) To display Class, Dob and City whose marks is between 450 and 551.

Ans. Select class, dob, city from student

where marks between 450 and 551;

(iii) To display Name, Class and total number of students who have secured more than 450 marks, class wise

Ans. Select name,class, count(*) from student

group by class

having marks> 450;

(iv) To increase marks of all students by 20 whose class is “XII

Ans. Update student

set marks=marks+20

where class=’XII’;

****Find output of the following queries.**

(v) SELECT COUNT(*), City FROM STUDENT GROUP BY CITY HAVING COUNT(*)>1;

```
2          Mumbai
2          Delhi
2          Moscow
```

(vi) SELECT MAX(DOB),MIN(DOB) FROM STUDENT;

```
08-12-1995    07-05-1993
```

(iii) SELECT NAME,GENDER FROM STUDENT WHERE CITY=’Delhi’;

```
Sanal    F
Store    M
```