**Section I** (30 marks)

1. Create table DEPT with the following structure:-

DEPTNO int(2)

DNAME varchar(15)

LOC varchar(10)

**create table Dept(**

**DEPTNO int(2),**

**DNAME varchar(15),**

**LOC varchar(10)**

**);**

Insert the following rows into the DEPT table:-

10 ACCOUNTING NEW YORK

20 RESEARCH DALLAS

30 SALES CHICAGO

40 OPERATIONS BOSTON

**insert into dept(deptno,dname,loc) Values**

**(10,'ACCOUNTING','NEW YORK'),**

**(20,'RESEARCH','DALLAS'),**

**(30,'SALES','CHICAGO'),**

**(40,'OPERATIONS','BOSTON');**

1. Create table EMP with the following structure:-

EMPNO int(4)

ENAME varchar(10)

JOB varchar(9)

HIREDATE date

SAL float(7,2)

COMM float(7,2)

DEPTNO int(2)

**create table EMP(**

**EMPNO int(4),**

**ENMAE varchar(10),**

**JOB varchar(9),**

**HIREDATE date,**

**SAL float(7,2),**

**COMM float(7,2),**

**DEPTNO int(2)**

**);**

Insert the following rows into the EMP table:-

7839 KING MANAGER 1991-11-17 5000 NULL 10

7698 BLAKE CLERK 1981-05-01 2850 NULL 30

7782 CLARK MANAGER 1981-06-09 2450 NULL 10

7566 JONES CLERK 1981-04-02 2975 NULL 20

7654 MARTIN SALESMAN 1981-09-28 1250 1400 30

7499 ALLEN SALESMAN 1981-02-20 1600 300 30

**Insert into EMP(EMPNO ,ENMAE,JOB,HIREDATE,SAL,COMM,DEPTNO ) values**

**(7839,'KING','MANAGER','1991-11-17',5000,null,10),**

**(7698,'BLAKE','CLERK','1981-05-01',2850,null,30),**

**(7782,'CLARK','MANAGER','1981-06-09',2450,NULL,10),**

**(7566,'JONES','CLERK','1981-04-02',2975,NULL,20),**

**(7654,'MARTIN','SALESMAN','1981-09-28',1250,1400,30),**

**(7499,'ALLEN','SALESMAN','1981-02-20',1600,300,30);**

Write SELECT statements to achieve the following:-

1. Display all the employees where SAL between 2500 and 5000 (inclusive of both).

**select \* from emp where sal between 2500 and 5000;**

1. Display all the ENAMEs in descending order of ENAME.

**select ename from emp order by ename desc;**

1. Display all the JOBs in lowercase.

**select lower(job) from emp;**

1. Display the ENAMEs and the lengths of the ENAMEs.

**select ename,length(ename) from emp;**

1. Display the DEPTNO and the count of employees who belong to that DEPTNO .

**select deptno,count(\*) from emp**

**group by deptno;**

1. Display the DNAMEs and the ENAMEs who belong to that DNAME.

**select dname,ename from emp,dept**

**where dept.deptno = emp.deptno;**

1. Display the position at which the string ‘AR’ occurs in the ename.

**select ename,instr(ename,'AR') from emp;**

1. Display the HRA for each employee given that HRA is 20% of SAL.

**select ename,sal,sal\*0.2 HRA from emp;**

**Section II** (10 marks)

1. Write a stored procedure by the name of PROC1 that accepts two varchar strings as parameters. Your procedure should then determine if the first varchar string exists inside the varchar string. For example, if string1 = ‘DAC’ and string2 = ‘CDAC, then string1 exists inside string2. The stored procedure should insert the appropriate message into a suitable TEMPP output table. Calling program for the stored procedure need not be written.

**delimiter //**

**create procedure find\_substr(fir varchar(20),sec varchar(20))**

**begin**

**if (instr(fir,sec)>0) then**

**insert into temp values(sec,'Found');**

**else**

**insert into temp values(sec,'NOT Found');**

**end if;**

**end //**

**delimiter ;**

**call find\_substr('CDAC','dac');**

1. Create a stored function by the name of FUNC1 to take three parameters, the sides of a triangle. The function should return a Boolean value:- TRUE if the triangle is valid, FALSE otherwise. A triangle is valid if the length of each side is less than the sum of the lengths of the other two sides. Check if the dimensions entered can form a valid triangle. Calling program for the stored function need not be written.

**delimiter //**

**create function func1(x int,y int,z int)**

**returns boolean**

**deterministic**

**begin**

**if(x < (y+z)) then**

**return true;**

**elseif(y < (x+z)) then**

**return true;**

**elseif(z < (x+y)) then**

**return true;**

**else**

**return false;**

**end if;**

**return false;**

**end //**

**delimiter ;**

**drop function func1;**