Assignment

Sept23/ DBT/126.1

Database Technologies

Diploma in Advance Computing

September 2023

**Procedure and Function**

|  |
| --- |
| 1. Write a procedure to accept a string and print all characters in separate lines.   Input: - Ram  Output: - R  a  m |
| drop procedure if exists str1;  delimiter $  create procedure str1(str varchar(20))  BEGIN  declare x int;  declare y int;  set x :=0;  set y := 0;  set y := length(str);    l:LOOP  set x:=x+1;  if x<=y then  select substring(str,x,1) as string;  ELSE  leave l;  End IF;    end loop l;  END $  delimiter ; |
|  |
| 1. Write a procedure to accept a string and print every character separated by a comm sign.   Input: - SALEEL  Output: - S, A, L, E, E, L |
| drop procedure if exists str1;  delimiter $  create procedure str1(str varchar(20))  BEGIN  declare x int;  set x :=1;  set @a="";    l:LOOP    if x=1 then  set @a= concat(@a, '' , substring(str,x,1));  ELSE  set @a= concat(@a, ',' , substring(str,x,1));    End IF;    if x>length(str) then leave l;  end if;  set x=x+1;  end loop l;  END $  delimiter ; |
|  |
| 1. Write a procedure to accept an alpha numeric string and separate number and characters of the string.   Input: - **SAL1234EEL**  Output: - SALEEL  1234 |
| drop procedure if exixts sal1;  delimiter $  create procedure sal1(str1 varchar(20))  BEGIN  declare a int;  set a:=0;    set @x:=" ";  set @y:=" ";  set @z:=" ";      S:LOOP    set a:=a+1;    set @x:=substring(str1,a,1);  IF(@x BETWEEN 'a' and 'z') THEN  set @y:=concat(@y,@x);  END IF;      IF(@x BETWEEN '0' and '9') THEN  set @z:=concat(@z,@x);  END IF;      IF a>length(str1)-1 THEN  LEAVE S;  END IF;  END LOOP S;  SELECT @y AS NAME;  SELECT @z AS NUMBER1;  END $  delimiter ; |
|  |
| 1. Write a procedure to print all employee name and his job in following format.   Input: - KING PRESIDENT  SCOTT ANALYST  Output: - K(ING) is PRESIDENT  S(COTT) is ANALYST |
| drop procedure if exists emp;  delimiter $  create procedure emp(str1 varchar(20),str2 varchar(20))  BEGIN  declare a int;  set a:=1;  set @x := "";  set @y := "";  set @z := "";  /\*set @m := ""; \*/      S:LOOP      IF a=1 then  set @x:= substring(str1,a,1);    ELSE  set @y := concat(@y,substring(str1,a,1));    END IF;    if a>length(str1) THEN LEAVE S;  END IF;  set a:=a+1;    END LOOP S;    set @z := str2;  Select concat(@x,'(',@y,')',' is ',@z) as string1;      END $  delimiter ;  ORRRRRRRRRRRRRRRR  drop procedure if exists emp;  delimiter $  create procedure emp(str1 varchar(20))  BEGIN  declare a int;  declare b int;  set b:=1;  set a:=1;  set @x := "";  set @y := "";  set @z := "";  set @m := "";  set @h := "";  set @j := "";      SPAC : LOOP    set @m:=substring(str1,b,1);  set @h:=concat(@h,@m);    if (@m=" ") THEN LEAVE SPAC;  END IF;    set b:=b+1;  END LOOP SPAC;    S:LOOP    IF a=1 then  set @x:= substring(str1,a,1);  END IF;    set a:=a+1;  IF a<(length(@h)) THEN  set @y := concat(@y,substring(str1,a,1));    ELSE  set @j :=concat(@j,substring(str1,a,1));    END IF;    if a>length(str1) THEN LEAVE S;  END IF;      END LOOP S;    Select concat(@x,'(',@y,')',' is ',@j) as string1;      END $  delimiter ; |
|  |
| 1. Write a procedure to print all upper and lower characters separately.   Input: - AbCdEfG  Output: - ACEG  bdf |
| drop procedure if exists LPUP;  delimiter $  create procedure LPUP(str1 varchar(20))  BEGIN  declare a int;  set a:=0;    set @x:=" ";  set @y:=" ";  set @z:=" ";  set @m:=" ";      S:LOOP    set a:=a+1;    set @x:=ASCII(substring(str1,a,1));  set @m:=substring(str1,a,1);  IF(@x BETWEEN '65' and '90') THEN  set @y:=concat(@y,@m);  END IF;      IF(@x BETWEEN '97' and '122') THEN  set @z:=concat(@z,@m);  END IF;      IF a>length(str1)-1 THEN  LEAVE S;  END IF;  END LOOP S;  SELECT @y AS UPPER\_CASE;  SELECT @z AS LOWER\_CASE;  END $  delimiter ; |
|  |
| 1. Write a procedure to find the number of vowels, digits and white spaces |
| drop procedure if exists vow;  delimiter $  create procedure vow(str1 varchar(20))  BEGIN  declare a int;  declare b int;  declare c int;  declare d int;  set a:=0;    set b:=0;    set c:=0;    set d:=0;    set @x:=" ";  set @y:=" ";  set @z:=" ";  set @l:="";  set @m:="";  set @n:="";    S:LOOP  set a:=a+1;  set @x := substring(str1,a,1);    IF @x IN ( 'A', 'E', 'I', 'O', 'U' ) THEN  set @y:=concat(@y,@x);  set b:=b+1;  set @l:=b;  END IF;    IF( @x BETWEEN '0' and '9')THEN  set @z:=concat(@z,@x);  set c:=c+1;  set @m:=c;  END IF;    IF( @x=" ") THEN  set d:=d+1;  set @n:=d;  END IF;    if a>length(str1) THEN LEAVE S;  END IF;    END LOOP S;  Select @l AS VOWELS;  Select @m AS NUM;  Select @n AS SPACE1;      END $  delimiter ; |
|  |
| 1. Write a procedure to remove all characters in a string except alphabets   Input: - saleel.bagde123@gmail.com  Output: - saleelbagdegmailcom |
| drop procedure if exists email;  delimiter $  create procedure email(str1 varchar(100))  BEGIN  declare a int;  set a := 0;  set @x :="";  set @y :="";    l: LOOP  set a := a+1;  set @x := substring(str1,a,1);    if @x between 'a' and 'z' THEN  set @y := concat(@y,@x);  END IF;    if a>length(str1) Then leave l;  END IF;    end LOOP l;    select @y AS alpha;  END $  delimiter ; |
|  |
| 1. Write a procedure to insert 10 rows in a table having following columns (using loop).   R (id int, message varchar(20)).  Output: -  id message  ---- -----------  1 i is odd  2 i is even  3 i is odd  4 i is even  5 i is odd  6 i is even  7 i is odd  8 i is even  9 i is odd  10 i is even |
| drop procedure if exists oddeven1;  delimiter $  create procedure oddeven1()  BEGIN  create table R (id int,message varchar(20));  END $  delimiter ;  drop procedure if exists oddeven2;  delimiter $  create procedure oddeven2()  BEGIN  declare i int;  set i:=1;  S:LOOP      if (MOD(i,2)=0) THEN  set @y:=concat(i,' is even');  insert into R (id,message) values (i,@y);    ELSE  set @y:=concat(i,' is odd');  insert into R (id,message) values (i,@y);  END IF;    set i:=i+1;    if i>10 then leave S;  END IF;  END LOOP S;  END $  delimiter ; |
|  |
| 1. Write a procedure to print five highest paid employees from the emp table using cursor. |
| drop procedure if exists top5Salary;  delimiter $  create procedure top5Salary()  BEGIN  declare \_salary int;  declare count1 int;  declare c1 cursor for select distinct sal from emp order by sal desc;  set count1 = 0;  open c1;  drop table if exists top5Salary;  create table top5Salary (sal int);  loop1:loop  set count1 = count1 + 1;  if count1 <= 5 then  fetch c1 into \_salary;  insert into top5Salary values (\_salary);  ELSE  leave loop1;  end if;  end loop loop1;  close c1;  select \* from top5Salary;  end $  delimiter ; |
|  |
| 1. Create the following table named (emp10, emp20, and emp30) which have the same structure of emp table.   Write a procedure to split employee records from emp table according to their department numbers and insert those records in the appropriate table using cursor.` |
| drop procedure if exists split\_EmpData;  delimiter $  create procedure split\_EmpData()  BEGIN  declare \_EMPNO,\_MGR,\_SAL,\_COMM,\_DEPTNO,\_BONUSID int;  declare \_ENAME,\_GENDER,\_JOB,\_PHONE,\_USERNAME,\_PWD varchar(50);  declare \_HIREDATE date;  declare \_isActive bool;  declare c1 cursor for select \* from emp;  declare exit handler for 1329 select "Exception is handled";  drop table if exists emp10;  drop table if exists emp20;  drop table if exists emp30;  create table emp10 like emp;  create table emp20 like emp;  create table emp30 like emp;    open c1;  loop10:loop    fetch c1 into \_EMPNO,\_ENAME,\_GENDER,\_JOB,\_MGR,\_HIREDATE,\_SAL,\_COMM,\_DEPTNO,\_BONUSID,  \_USERNAME,\_PWD,\_PHONE,\_isActive;    if(\_deptno =10) THEN  insert into emp10 values(\_EMPNO,\_ENAME,\_GENDER,\_JOB,\_MGR,\_HIREDATE,\_SAL,\_COMM,\_DEPTNO,\_BONUSID,  \_USERNAME,\_PWD,\_PHONE,\_isActive);    elseif (\_deptno =20) THEN  insert into emp20 values(\_EMPNO,\_ENAME,\_GENDER,\_JOB,\_MGR,\_HIREDATE,\_SAL,\_COMM,\_DEPTNO,\_BONUSID,  \_USERNAME,\_PWD,\_PHONE,\_isActive);    elseif (\_deptno =30) THEN  insert into emp30 values(\_EMPNO,\_ENAME,\_GENDER,\_JOB,\_MGR,\_HIREDATE,\_SAL,\_COMM,\_DEPTNO,\_BONUSID,  \_USERNAME,\_PWD,\_PHONE,\_isActive);    elseif(\_deptno !=10 and \_deptno !=20 and \_deptno !=30) THEN  select "Employee department is different than 10 20 and 30";    end if;  end loop loop10;  close c1;  select \_EMPNO,\_ENAME,\_GENDER,\_JOB,\_MGR,\_HIREDATE,\_SAL,\_COMM,\_DEPTNO,\_BONUSID,  \_USERNAME,\_PWD,\_PHONE,\_isActive from emp10;  select \* from emp20;  select \* from emp30;  end $  delimiter ; |
|  |
| 1. Write a procedure to display the department number and employee name in the following format.   Output: -  10 -> (AARAV, THOMAS, CLARK, KING, MILLER)  20 -> (SHARMIN, BANDISH, SMITH, JONES, SCOTT, FRED, ADAMS, FORD)  30 -> (GITA, ALLEN, WARD, MARTIN, BLAKE, TURNER, JAMES, HOFFMAN, GRASS)  40 –> (No employee work in department 40…)  50 -> (VRUSHALI, SANGITA, SUPRIYA) |
| drop procedure if exists pro11;  delimiter $  create procedure pro11()  begin  declare flag1 bool;  declare flag2 bool;  declare count1 int;  declare \_deptno int;  declare \_deptno1 int;  drop table if exists emp\_dept1;  create table emp\_dept1 (c1 varchar(1000));  set \_deptno := 1;  set \_deptno1 := 1;  lbl1:loop  if \_deptno in (select deptno from dept) then  if \_deptno1 in (select deptno from emp) then  insert into emp\_dept1 values(concat(\_deptno," -> ",(select group\_concat(ename) from emp join dept where emp.deptno=\_deptno and dept.deptno=emp.deptno group by emp.deptno)));  set \_deptno := \_deptno + 1;  set \_deptno1 := \_deptno1 + 1;  else  insert into emp\_dept1 values(concat(\_deptno," -> ","(No employee work in department 40...)"));  set \_deptno := \_deptno + 1;  set \_deptno1 := \_deptno1 + 1;  end if;  elseif \_deptno > (select max(deptno) from dept) then  leave lbl1;  else  set \_deptno := \_deptno + 1;  set \_deptno1 := \_deptno1 + 1;  end if;  end loop lbl1;  select \* from emp\_dept1;  end $  delimiter ; |
|  |
| 1. Write a procedure to accept customer number and display all his order. (Use customers and orders table) |
| drop procedure if exists order1;  delimiter $  create procedure order1(x int)  BEGIN  select c.cname,o. \* from customers c inner join orders o on c.cnum=o.cnum where c.cnum=x;    END $  delimiter ; |
|  |
| 1. Write a procedure to convert numbers into word   Input: - 45234  Output: - Four Five Two Three Four |
| drop procedure if exists ntw;  delimiter $  CREATE procedure ntw(num int)  BEGIN  declare x varchar(50);  declare y int;  set x := "";  l:loop  if num>0 THEN  set y:= num%10;  set num := num div 10;  if y = 1 then  set x := concat("one ",x);  elseif y = 2 THEN  set x := concat("two ",x);  elseif y = 3 THEN  set x := concat("three ",x);  elseif y = 4 THEN  set x := concat("four ",x);  elseif y = 5 THEN  set x := concat("five ",x);  elseif y = 6 THEN  set x := concat("six ",x);  elseif y = 7 THEN  set x := concat("seven ",x);  elseif y = 8 THEN  set x:= concat("eight ",x);  elseif y = 9 THEN  set x := concat("nine ",x);  elseif y = 0 THEN  set x := concat("zero ",x);  end if;  ELSE  leave l;  end if;  end loop l;  select x AS inwords;    END $  delimiter ; |
|  |
| 1. Write a procedure to find the sum of digits.   Input: - 5675  Output: - Twenty Three |
| drop procedure if exists sum1;  delimiter $  create procedure sum1(num1 int)  BEGIN  declare sumnum int(50);  declare rem int(50);    set @ans:="";  set @dig1:="";  set @dig2:="";  set @sumnum :=0;        SUM1 : LOOP    set rem:=num1%10;  set @sumnum:=@sumnum+rem;  set num1=num1 div 10;    if num1 = 0 THEN leave SUM1;  END if;    END LOOP SUM1;    set @dig2:= floor(@sumnum/10);  set @dig1:= @sumnum-(@dig2\*10);  /\*if dig3 > 0 then  case  when dig3=1 then set ans=concat(ans, 'one hundred');  when dig3=2 then set ans=concat(ans, 'two hundred');  when dig3=3 then set ans=concat(ans, 'three hundred');  when dig3=4 then set ans=concat(ans, 'four hundred');  when dig3=5 then set ans=concat(ans, 'five hundred');  when dig3=6 then set ans=concat(ans, 'six hundred');  when dig3=7 then set ans=concat(ans, 'seven hundred');  when dig3=8 then set ans=concat(ans, 'eight hundred');  when dig3=9 then set ans=concat(ans, 'nine hundred');  else set ans = ans;  end case;  end if;\*/  if @dig2 =1 then  case  when (@dig2\*10 + @dig1) = 10 then set @ans=concat(@ans,' ten');  when (@dig2\*10 + @dig1) = 11 then set @ans=concat(@ans,' eleven');  when (@dig2\*10 + @dig1) = 12 then set @ans=concat(@ans,' twelve');  when (@dig2\*10 + @dig1) = 13 then set @ans=concat(@ans,' thirteen');  when (@dig2\*10 + @dig1) = 14 then set @ans=concat(@ans,' fourteen');  when (@dig2\*10 + @dig1) = 15 then set @ans=concat(@ans,' fifteen');  when (@dig2\*10 + @dig1) = 16 then set @ans=concat(@ans,' sixteen');  when (@dig2\*10 + @dig1) = 17 then set @ans=concat(@ans,' seventeen');  when (@dig2\*10 + @dig1) = 18 then set @ans=concat(@ans,' eighteen');  when (@dig2\*10 + @dig1) = 19 then set @ans=concat(@ans,' nineteen');  else set @ans:=@ans;  end case;  else  if @dig2 > 0 then  case  when @dig2=2 then set @ans=concat(@ans, ' twenty');  when @dig2=3 then set @ans=concat(@ans, ' thirty');  when @dig2=4 then set @ans=concat(@ans, ' fourty');  when @dig2=5 then set @ans=concat(@ans, ' fifty');  when @dig2=6 then set @ans=concat(@ans, ' sixty');  when @dig2=7 then set @ans=concat(@ans, ' seventy');  when @dig2=8 then set @ans=concat(@ans, ' eighty');  when @dig2=9 then set @ans=concat(@ans, ' ninety');  else set @ans:=@ans;  end case;  end if;  if @dig1 > 0 then  case  when @dig1=1 then set @ans=concat(@ans, ' one');  when @dig1=2 then set @ans=concat(@ans, ' two');  when @dig1=3 then set @ans=concat(@ans, ' three');  when @dig1=4 then set @ans=concat(@ans, ' four');  when @dig1=5 then set @ans=concat(@ans, ' five');  when @dig1=6 then set @ans=concat(@ans, ' six');  when @dig1=7 then set @ans=concat(@ans, ' seven');  when @dig1=8 then set @ans=concat(@ans, ' eight');  when @dig1=9 then set @ans=concat(@ans, ' nine');8  else set @ans:=@ans;  end case;  end if;  end if;    select @ans as sumofnum;  END $  delimiter ; |
|  |
| 1. Write a procedure to find how many “Sundays” are present between two given dates.   Input: - Date1 and Date2  Output: - 3 Sunday’s |
| drop procedure if exists sun;  delimiter $  create procedure sun(Date1 date, Date2 date)  BEGIN  declare counter INT;  declare currDay DATE;  set currDay=Date1;  set counter = 0;  While currDay<=Date2 DO  IF DAYOFWEEK(currDay)=1 THEN  set counter:=counter+1;  END IF;  SET currDay = DATE\_ADD(currDay, INTERVAL 1 DAY);  END While;  select concat (counter,"Sunday's") AS output1;  end $  delimiter ; |
|  |
| 1. Writer a procedure which will accept date and weekday name from the user and print upcoming date on than weekday   Input: - (‘2023-04-26’, ‘Saturday’)  Output: - ‘2023-04-29’ |
| drop procedure if exists pro16;  delimiter $  create procedure pro16(date1 date, weekday1 varchar(10))  begin  lbl1:loop  if date\_format(date1,'%W')=weekday1 then  select date1;  leave lbl1;  else  set date1 := date1+interval 1 day;  end if;  end loop lbl1;  end $  delimiter ; |
|  |