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 pro2;  delimiter $  create procedure pro2(in x varchar(20))  BEGIN  declare y int;  declare pos int;  set pos:=0;  set y:=0;  set y:=length(x);  label2:LOOP    set pos:=pos+1;    if pos<=y then  select substr(x,pos,1) as string;  ELSE  leave label2;  end if;  end LOOP label2;  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 pro1;  delimiter $  create procedure pro1(in x varchar(20))  BEGIN  declare y int;  declare z varchar(20);  declare output varchar(20);  set y:=1;  set output:="";  lbl1:loop  if y>length(x) then  leave lbl1;  END IF;  set z:=substr(x,y,1);  set y:=y+1;  set output:=concat(output,z,",");  end loop lbl1;  select substr(output,1,length(output)-1) AS OUTPUT;  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 exists pro1;  delimiter $  create procedure pro1(in x varchar(20))  BEGIN  declare y varchar(20);  declare num varchar(20);  declare strn varchar(20);  declare z int;  set strn="";  set z:=1;  set num="";  lbl1:loop  set y:=substr(x,z,1);  set z=z+1;  if ascii(y)>48 and ascii(y)<57 THEN  set num:=concat(num,y);  ELSE  set strn=concat(strn,y);  end if;    if z>length(x) THEN  leave lbl1;  end if;  end loop lbl1;  select strn;  select num;  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 pro1;  delimiter $  create procedure pro1(in x varchar(20))  BEGIN  declare y int;  declare z varchar(20);  declare output varchar(20);  declare a varchar(20);  declare b varchar(20);  declare d varchar(20);  set y:=1;  set output:="";  lbl1:loop  set z:=substr(x,y,1);  set y:=y+1;  if z=" " then  leave lbl1;  END IF;  set output:=concat(output,z);  end loop lbl1;  set a:=substr(output,1,1);  set b:=substr(output,2,length(output));  set d:=substr(x,length(output)+1,length(x));  select concat(a,"(",b,")"," is ",d) as output;  end $  delimiter ;  OR  drop procedure if exists pro4;  delimiter $  drop table if exists String2;  create table String2 (c1 varchar(100));  create procedure pro4()  begin  declare name1 varchar(10);  declare job1 varchar(20);  declare maxRow int;  declare ct int;  declare String1 varchar(100);  set ct := 1;  select max(r1) into maxRow from (select row\_number() over() r1 from emp) t1;  loop2:loop  if ct<=maxRow then  select ename into name1 from (select ename,row\_number() over() r1 from emp) t1 where r1=ct;  select job into job1 from (select job,row\_number() over() r1 from emp) t1 where r1=ct;  select concat(substring(name1,1,1),'(',substring(name1,2),') is ',job1) into String1;  insert into String2 values (String1);  set ct := ct + 1;  else  leave loop2;  end if;  end loop loop2;  select \* from string2;  end $  delimiter ; |
|  |
| 1. Write a procedure to print all upper and lower characters separately.   Input: - AbCddEfG  Output: - ACEG  bdf |
| drop procedure if exists pro1;  delimiter $  create procedure pro1(in x varchar(20))  BEGIN  declare y varchar(20);  declare CAP varchar(20);  declare small varchar(20);  declare z int;  set CAP="";  set z:=1;  set small="";  lbl1:loop  set y:=substr(x,z,1);  set z=z+1;  if ascii(y)>=65 and ascii(y)<=90 THEN  set CAP:=concat(CAP,y);  ELSE  set small=concat(small,y);  end if;    if z>length(x) THEN  leave lbl1;  end if;  end loop lbl1;  select CAP;  select small;  end $  delimiter ; |
|  |
| 1. Write a procedure to find the number of vowels, digits and white spaces |
| drop PROCEDURE IF EXISTS pro6;  delimiter $  CREATE PROCEDURE pro6(IN string1 VARCHAR(40))  BEGIN  DECLARE len\_str int;  DECLARE counter int;  DECLARE ch varchar(2);  DECLARE countv int;  DECLARE countd int;  DECLARE countws int;    set ch :='';  set len\_str:=LENGTH(string1);  set counter:=0;  set countv:=0;  set countd:=0;  set countws:=0;    lbl1:LOOP  IF counter > LENGTH(string1) THEN  leave lbl1;  end if;  set counter:=counter+1;  SET ch := SUBSTR(string1,counter,1);    IF (ch='A' or ch='E' or ch='I' or ch='O' or ch='U') THEN  set countv:= countv+1;  elseif ch between 0 and 9 THEN  set countd:= countd+1;  ELSE  if ch=" " THEN  set countws:= countws+1;  end IF;  end IF;    end LOOP lbl1;  SELECT countv 'Vowels count',countd 'digits count',countws 'white spaces count';  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 pro1;  delimiter $  create procedure pro1(in x varchar(20))  BEGIN  declare y varchar(20);  declare email varchar(20);  declare z int;  set z:=1;  set email="";  lbl1:loop  set y:=substr(x,z,1);  set z=z+1;  if ascii(y)>=97 and ascii(y)<=122 THEN – if y between 97 and 122 then  set email:=concat(email,y);  end if;    if z>length(x) THEN  leave lbl1;  end if;  end loop lbl1;  select email;  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 pro1;  delimiter $  create procedure pro1()  BEGIN  declare id int;  declare msg varchar(20);  set msg="";  set id=1;  label1:LOOP  if ((id%2)=0) THEN  set msg:= "i is even";  insert into p8 values (id,msg);  else  set msg:="i is odd";  insert into p8 values (id,msg);  end if;  set id=id+1;  if id>10 THEN  leave label1;  end if;  end loop label1;  end $  delimiter ; |
|  |
| 1. Write a procedure to print five highest paid employees from the emp table using cursor. |
| drop procedure if exists pro1;  delimiter $  create procedure pro1()  BEGIN  declare \_sal int;  declare \_ename varchar(40);  declare c1 cursor for select ename,sal from emp order by sal desc limit 5;  declare exit handler for 1329 select "EOF";  open c1;  lb:loop  fetch c1 into \_ename,\_sal;    select \_ename,\_sal;  end loop lb;  close c1;  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 que11;  delimiter $  create procedure que11()  BEGIN    select concat(deptno,' -> ',group\_concat(ename)) from emp group by deptno ;    end $  delimiter ; |
|  |
| 1. Write a procedure to accept customer number and display all his order. (Use customers and orders table) |
| DROP procedure if exists pro12;  delimiter $  create procedure pro12(cid int)  begin  select c.cname,o.\* from customers c join orders o on c.cnum=o.cnum where c.cnum=cid;  end $  delimiter ; |
|  |
| 1. Write a procedure to convert numbers into word   Input: - 45234  Output: - Four Five Two Three Four |
| DROP PROCEDURE IF EXISTS pro1;  DELIMITER $  CREATE PROCEDURE pro1(IN number INT(10))  BEGIN  DECLARE i INT(10);  DECLARE z INT;  DECLARE output VARCHAR(50);  DECLARE units VARCHAR(10);  SET i = 0;  SET units = "";  SET output = "";  SET z = 0;    lb: LOOP  SET z = z+1;  IF z > LENGTH(number) THEN  LEAVE lb;  END IF;    SET i = SUBSTR(number, z, 1);  IF i = 0 THEN  SET units = "Zero";  ELSEIF i = 9 THEN  SET units = "nine";  ELSEIF i = 8 THEN  SET units = "eight";  ELSEIF i = 7 THEN  SET units = "seven";  ELSEIF i = 6 THEN  SET units = "six";  ELSEIF i = 5 THEN  SET units = "five";  ELSEIF i = 4 THEN  SET units = "four";  ELSEIF i = 3 THEN  SET units = "three";  ELSEIF i = 2 THEN  SET units = "two";  ELSEIF i = 1 THEN  SET units = "one";  END IF;    SET output = CONCAT(output, units, " ");  END LOOP lb;    SELECT output AS "Output";  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 pro15;  delimiter $  create procedure pro15(date1 date,date2 date)  begin  declare count int;  set count := 0;  lbl1:loop  if date1<date2 then  if date\_format(date1,'%W') = "Sunday" then  set count := count + 1;  set date1 := date1 + interval 1 day;  else  set date1 := date1 + interval 1 day;  end if;  else  leave lbl1;  end if;  end loop lbl1;  select count;  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 ; |
|  |