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 pro1;  delimiter $  CREATE PROCEDURE pro1(IN string1 VARCHAR(15))  BEGIN  DECLARE len\_str int;  set len\_str:= LENGTH(string1);  set @counter:=1;    lbl1:LOOP  IF @counter < len\_str+1 THEN  SELECT SUBSTR(string1,@counter,1);  set @counter:=@counter+1;  ELSE  leave lbl1;  end IF;  end LOOP lbl1;  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 pro2;  delimiter $  CREATE PROCEDURE pro2(IN string1 VARCHAR(15))  BEGIN  DECLARE len\_str int;  set len\_str:= LENGTH(string1);    set @counter:=1;  set @res:="";  lbl1:LOOP  IF @counter < len\_str+1 THEN  set @res:= CONCAT(@res, SUBSTR(string1,@counter,1),',' );  set @counter:=@counter+1;  ELSE  leave lbl1;  end IF;  end LOOP lbl1;    SELECT SUBSTR(@res,1,len\_str+len\_str-1);  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 pro3;  delimiter $  CREATE PROCEDURE pro3(IN string1 VARCHAR(15))  BEGIN  DECLARE len\_str int;  DECLARE str1 varchar(20);  DECLARE num1 varchar(20);  set @counter:=1;    set str1:= "";  set num1:= "";    set len\_str:= LENGTH(string1);    lbl1:LOOP  IF @counter < len\_str+1 THEN  set @res:= SUBSTR(string1,@counter,1);  set @counter:=@counter+1;    IF ASCII(@res)>64 and ASCII(@res)<123 THEN  set str1:= CONCAT(str1,@res);  ELSE  set num1:= CONCAT(num1,@res);  end IF;    ELSE  leave lbl1;  end IF;  end LOOP lbl1;    SELECT str1,num1;  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 pro4;  delimiter $  CREATE PROCEDURE pro4(string1 VARCHAR(15))  BEGIN  DECLARE len\_str int;  DECLARE counter int;  DECLARE position varchar(20);  DECLARE ch varchar(2);  set @res:="";    set counter:=1;  set len\_str:= LENGTH(string1);    lbl1:LOOP  IF counter < len\_str+1 THEN  set ch:= SUBSTR(string1,counter,1);    IF counter =1 THEN  set @res:=CONCAT(ch,'(');  ELSEIF ch=" " THEN  set @res:=CONCAT(@res,") is ");  set position:=SUBSTR(string1,counter+1,len\_str);  leave lbl1;  ELSE  set @res:=CONCAT(@res,ch);  end IF;    set counter:=counter+1;  ELSE  leave lbl1;  end IF;  end LOOP lbl1;  select CONCAT(@res,position) as 'output';  end $  delimiter ; |
|  |
| 1. Write a procedure to print all upper and lower characters separately.   Input: - AbCdEfG  Output: - ACEG  bdf |
| drop PROCEDURE IF EXISTS pro5;  delimiter $  CREATE PROCEDURE pro5(IN string1 VARCHAR(15))  BEGIN  DECLARE len\_str int;  DECLARE ch varchar(2);  DECLARE upperC\_string varchar(40);  DECLARE lowerC\_string varchar(40);  set @counter:=1;    set len\_str:= LENGTH(string1);  set upperC\_string:="";  set lowerC\_string:="";    lbl1:LOOP  IF @counter < len\_str+1 THEN  SET ch= SUBSTR(string1,@counter,1);    IF ASCII(ch)>64 and ASCII(ch)<91 THEN  set upperC\_string := CONCAT(upperC\_string,ch);  ELSE  set lowerC\_string := CONCAT(lowerC\_string,ch);  end IF;    set @counter:=@counter+1;    ELSE  leave lbl1;  end IF;    end LOOP lbl1;  SELECT upperC\_string,lowerC\_string;  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:=1;  set countv:=0;  set countd:=0;  set countws:=0;    lbl1:LOOP  IF counter < len\_str+1 THEN  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 ASCII(ch)>47 and ASCII(ch)<58 THEN  set countd:= countd+1;  ELSE  if ch=" " THEN  set countws:= countws+1;  end IF;  end IF;    set counter:=counter+1;    ELSE  leave lbl1;  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 pro7;  delimiter $  CREATE PROCEDURE pro7(IN string1 VARCHAR(40))  BEGIN  DECLARE len\_str int;  DECLARE str1 varchar(20);  set @counter:=1;    set str1:= "";    set len\_str:= LENGTH(string1);    lbl1:LOOP  IF @counter < len\_str+1 THEN  set @res:= SUBSTR(string1,@counter,1);  set @counter:=@counter+1;    IF ASCII(@res)>64 and ASCII(@res)<123 THEN  set str1:= CONCAT(str1,@res);  end if;    ELSE  leave lbl1;  end IF;  end LOOP lbl1;    SELECT str1;  end $  delimiter ; |
|  |
| 1. Write a procedure to insert 10 rows in a table having following columns (using loop).   R (id int, message varchar(20)). (Table name: R4)  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 pro8;  delimiter $  CREATE PROCEDURE pro8()  BEGIN  DECLARE tsize int;  set tsize:=10;  set @counter:=1;    lbl1:LOOP  IF @counter < tsize+1 THEN    IF (@counter%2 =1) THEN  INSERT INTO R5 VALUES(@counter,'i is odd');  ELSE  INSERT INTO R5 VALUES(@counter,'i is even');  end IF;    set @counter:=@counter+1;  ELSE  leave lbl1;  end IF;    end LOOP lbl1;  select \* from r5;  end $  delimiter ; |
|  |
| 1. Write a procedure to print five highest paid employees from the emp table using cursor. |
| drop PROCEDURE IF EXISTS pro9;  delimiter $  CREATE PROCEDURE pro9()  BEGIN  DECLARE \_empno,\_sal int;  DECLARE \_ename,\_job VARCHAR(30);    DECLARE c1 CURSOR FOR select empno,ename,job,sal from emp order by sal desc limit 5;  DECLARE EXIT handler FOR 1329 SELECT "End of File";    OPEN c1;  lbl:LOOP  FETCH c1 INTO \_empno,\_ename,\_job,\_sal;    SELECT \_empno,\_ename,\_job,\_sal;  end LOOP lbl;  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 pro10;  delimiter $  CREATE PROCEDURE pro10()  BEGIN  DECLARE \_empno,\_sal,\_deptno int;  DECLARE \_ename,\_job VARCHAR(30);    DECLARE c1 CURSOR FOR select empno,ename,job,sal,deptno from emp;  DECLARE EXIT handler FOR 1329 SELECT "End of File";    OPEN c1;  lbl:LOOP  FETCH c1 INTO \_empno,\_ename,\_job,\_sal,\_deptno;  IF \_deptno=10 THEN  insert into emp10(empno,ename,job,sal,deptno) VALUES(\_empno,\_ename,\_job,\_sal,\_deptno);  ELSEIF \_deptno=20 THEN  insert into emp20(empno,ename,job,sal,deptno) VALUES(\_empno,\_ename,\_job,\_sal,\_deptno);  ELSEIF \_deptno=30 THEN  insert into emp30(empno,ename,job,sal,deptno) VALUES(\_empno,\_ename,\_job,\_sal,\_deptno);  ELSE  select \_deptno;  end IF;    end LOOP lbl;  CLOSE c1;  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 \_deptno int;  DECLARE \_ename VARCHAR(255);    DECLARE c1 CURSOR FOR select deptno,group\_concat(ename) from emp group by deptno;  DECLARE EXIT handler FOR 1329 SELECT "End of File";    OPEN c1;  lbl:LOOP  FETCH c1 INTO \_deptno,\_ename;    set @x := CONCAT(\_deptno, " -> (" ,\_ename , ")");  select @x as 'Result';    end LOOP lbl;  CLOSE c1;  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  DECLARE \_ONUM,\_CNUM,\_SNUM int;  DECLARE \_AMT FLOAT;  DECLARE \_ODATE date;  DECLARE \_type,\_name varchar(10);    DECLARE c1 CURSOR FOR select cname,o.ONUM,o.AMT,o.ODATE,o.CNUM,o.SNUM,o.type from customers c join orders o using(CNUM) where c.cnum=\_cid ;    DECLARE EXIT handler FOR 1329 SELECT "End of File";    OPEN c1;  lbl:LOOP  FETCH c1 INTO \_name,\_ONUM,\_AMT,\_ODATE,\_CNUM,\_SNUM,\_type;    select \_CNUM,\_name,\_ONUM,\_AMT,\_ODATE,\_SNUM,\_type;    end LOOP lbl;  CLOSE c1;  end $  delimiter ; |
|  |
| 1. Write a procedure to convert numbers into word   Input: - 45234  Output: - Four Five Two Three Four |
| drop PROCEDURE IF EXISTS pro13;  delimiter $  CREATE PROCEDURE pro13(num bigint)  BEGIN  DECLARE len\_str int;  DECLARE n int;  DECLARE res varchar(100);  set len\_str:= LENGTH(num);    set @counter:=1;  set res:="";    lbl1:LOOP  IF @counter < len\_str+1 THEN  set n:= SUBSTR(num,@counter,1);  IF n=1 THEN  set res:= concat(res,"One ");  elseif n=2 THEN  set res:= concat(res,"Two ");  elseif n=3 THEN  set res:= concat(res,"Three ");  elseif n=4 THEN  set res:= concat(res,"Four ");  elseif n=5 THEN  set res:= concat(res,"Five ");  elseif n=6 THEN  set res:= concat(res,"Six ");  elseif n=7 THEN  set res:= concat(res,"Seven ");  elseif n=8 THEN  set res:= concat(res,"Eight ");  elseif n=9 THEN  set res:= concat(res,"Nine ");  ELSE  set res:= concat(res,"Zero ");  end IF;    set @counter:=@counter+1;  ELSE  leave lbl1;  end IF;    end LOOP lbl1;  SELECT res;  end $  delimiter ; |
|  |
| 1. Write a procedure to find the sum of digits.   Input: - 5675  Output: - Twenty Three |
| drop PROCEDURE IF EXISTS pro14;  delimiter $  CREATE PROCEDURE pro14(num bigint)  BEGIN  DECLARE len\_str int;  DECLARE n int;  DECLARE res int;  set len\_str:= LENGTH(num);    set @counter:=1;  set res:=0;    lbl1:LOOP  IF @counter < len\_str+1 THEN  set n:= SUBSTR(num,@counter,1);  set res:= res + n;    set @counter:=@counter+1;  ELSE  leave lbl1;  end IF;    end LOOP lbl1;  call pro13(res);  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 ; |
|  |