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(x VARCHAR(10))  BEGIN  set @i=1;  declare curr;  set @leng=LENGTH(x);  while @i<=@leng do  set curr=SUBSTR(x,@i,1);  select curr;  set @i=@i+1;  end WHILE;    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(A VARCHAR(10))  BEGIN  set @x=1;  set @r="";  lbl:LOOP  if @x=1  then  set @r:=concat(@r,'',substr(A,@x,1));  else  set @r:=concat(@r,',',substr(A,@x,1));  end if;  set @x:=@x+1;  if @x>length(A) then leave lbl;  end if;  end loop lbl;  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(str1 varchar(50))  BEGIN  declare x int;  set x:=1;  set @strings:="";  set @num:="";  l:LOOP  if substr(str1,x,1)>='0' and substr(str1,x,1)<='9' THEN  set @num:=concat(@num,'',substr(str1,x,1));  ELSE  set @strings:=concat(@strings,'',substr(str1,x,1));  end IF;  set x=x+1;  if x>length(str1) then leave l;  end if;  end loop l;  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(name varchar(30),role varchar(30))  BEGIN  declare x int;  set @ans="";  set x=1;  l:LOOP  if x=1 then  set @ans=concat(@ans,substr(name,x,1),'(');  end IF;  if x=length(name) THEN  set @ans=concat(@ans,substr(name,x,1),')',' is ',role);  end if;  if x!=1 and x!=length(name) THEN  set @ans:=concat(@ans,substr(name,x,1));  end if;  set x=x+1;  if x> length(name) then leave l;  end if;  end loop l;  end .  delimiter ; |
|  |
| 1. Write a procedure to print all upper and lower characters separately.   Input: - AbCdEfG  Output: - ACEG  bdf |
| drop PROCEDURE if exists pro1;  delimiter .  create PROCEDURE pro1(str1 varchar(50))  BEGIN  DECLARE x int;  set x=1;  set @capital="";  set @lowwer="";  l:LOOP  if ASCII(SUBSTR(str1,x,1))>= ASCII('a') and ASCII(substr(str1,x,1))<=ASCII('z') THEN  set @lowwer:=concat(@lowwer,SUBSTR(str1,x,1));  ELSE  set @capital:=concat(@capital,SUBSTR(str1,x,1));  end IF;  set x=x+1;  if x>length(str1) then leave l;  end if;  end loop l;  end .  delimiter ; |
|  |
| 1. Write a procedure to find the number of vowels, digits and white spaces |
| DROP PROCEDURE if EXISTS PRO1;  DELIMITER .  CREATE PROCEDURE pro1 (stringg VARCHAR(50))  BEGIN  DECLARE x int;  SET x=1;  SET @num=0;  SET @vow=0;  SET @spa=0;  l:LOOP  if substr(stringg,x,1)>='0'and substr(stringg,x,1)<='9' THEN  set @num=@num+1;  end if;  if substr(stringg,x,1)='a' or substr(stringg,x,1)='e' or substr(stringg,x,1)='i' or substr(stringg,x,1)='o' or substr(stringg,x,1)='u' THEN  set @vow=@vow+1;  end if;  if substr(stringg,x,1)=' ' then  set @spa=@spa+1;  end if;  set x=x+1;  if x>length(stringg) then leave l;  end if;  end loop l;  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 remove;  delimiter $  create procedure remove(str varchar(100))  BEGIN  declare x int;  declare str1 varchar(50);  set x:=1;  set str1:='';  l:loop  if x>length(str) then leave l;  end if;  if (ascii(substr(str,x,1))>=65 and ascii(substr(str,x,1))<=90) or  (ascii(substr(str,x,1))>=97 and ascii(substr(str,x,1))<=122) then    set str1:= concat(str1,substr(str,x,1));  set x:= x+1;  else  set x:=x+1;    end if;  end loop l;  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)).  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 even;  delimiter $  create procedure even(x int)  begin  declare a,len1 int;  set a:=0;  set len1:=x;  lb1:loop  set a:= a+1;  if a%2=0 then  insert into even values(a,' is even');  else  insert into even values(a,' is odd');  end if;    if a>=len1 then  leave lb1;  end if;    end loop lb1;  select \* from even;  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 s int;  declare n varchar(20);  declare c1 cursor for select ename,sal from emp order by sal DESC limit 5;  open c1;  l1:LOOP  fetch c1 into n,s ;  select n,s;  end loop l1;  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 pro2;  delimiter $  create PROCEDURE pro2()  BEGIN    declare \_EMPNO,\_MGR,\_SAL,\_COMM,\_DEPTNO,\_BONUSID int;  declare \_ENAME,\_JOB,\_user,\_PWD,\_PHONE varchar(40);  declare \_isActive boolean;  declare \_GENDER char;  declare \_HIREDATE date;  declare c1 cursor for select \* from emp where deptno=10;  declare c2 cursor for select \* from emp where deptno=20;  declare c3 cursor for select \* from emp where deptno=30;    open c1;  open c2;  open c3;  l1:LOOP  fetch c1 into \_EMPNO,\_ENAME,\_GENDER,\_JOB,\_MGR,\_HIREDATE,\_SAL,\_COMM,\_DEPTNO,\_BONUSID,\_user,\_PWD,\_PHONE,\_isActive;    insert into emp10 values(\_EMPNO,\_ENAME,\_GENDER,\_JOB,\_MGR,\_HIREDATE,\_SAL,\_COMM,\_DEPTNO,\_BONUSID,\_user,\_PWD,\_PHONE,\_isActive);    fetch c2 into \_EMPNO,\_ENAME,\_GENDER,\_JOB,\_MGR,\_HIREDATE,\_SAL,\_COMM,\_DEPTNO,\_BONUSID,\_user,\_PWD,\_PHONE,\_isActive;    insert into emp20 values(\_EMPNO,\_ENAME,\_GENDER,\_JOB,\_MGR,\_HIREDATE,\_SAL,\_COMM,\_DEPTNO,\_BONUSID,\_user,\_PWD,\_PHONE,\_isActive);      fetch c3 into \_EMPNO,\_ENAME,\_GENDER,\_JOB,\_MGR,\_HIREDATE,\_SAL,\_COMM,\_DEPTNO,\_BONUSID,\_user,\_PWD,\_PHONE,\_isActive;    insert into emp30 values(\_EMPNO,\_ENAME,\_GENDER,\_JOB,\_MGR,\_HIREDATE,\_SAL,\_COMM,\_DEPTNO,\_BONUSID,\_user,\_PWD,\_PHONE,\_isActive);    end loop l1;  close c1;  close c2;  close c3;      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 pro3;  delimiter $  create procedure pro3()  BEGIN  select concat(deptno,' -> ','(',group\_concat(ename),')') as "Result" 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 pro4;  delimiter $  create procedure pro4(a int)  BEGIN  select \* from orders where cnum=a;  end $  delimiter ; |
|  |
| 1. Write a procedure to convert numbers into word   Input: - 45234  Output: - Four Five Two Three Four |
| drop procedure if exists pro5;  delimiter $  create procedure pro5(b, VARCHAR(20))  BEGIN  set @leng=LENGTH(b);  set @curr='';  set @i=1;  set @ans='';  set @a='';  while @i<= @leng do  set @curr=SUBSTR(b,@i,1);  if @curr='1' THEN  set @a="one";  set @ans=concat(@ans,' ',@a);    ELSEIF @curr='2' THEN  set @a="two";  set @ans=concat(@ans,' ',@a);    ELSEIF @curr='3' THEN  set @a="three";  set @ans=concat(@ans,' ',@a);    ELSEIF @curr='4' THEN  set @a="four";  set @ans=concat(@ans,' ',@a);    ELSEIF @curr='5' THEN  set @a="five";  set @ans=concat(@ans,' ',@a);    ELSEIF @curr='6' THEN  set @a="six";  set @ans=concat(@ans,' ',@a);    ELSEIF @curr='7' THEN  set @a="seven";  set @ans=concat(@ans,' ',@a);    ELSEIF @curr='8' THEN  set @a="eight";  set @ans=concat(@ans,' ',@a);    ELSEIF @curr='9' THEN  set @a="nine";  set @ans=concat(@ans,' ',@a);    ELSEIF @curr='0' THEN  set @a="zero";  set @ans=concat(@ans,' ',@a);        end if;    set @i:=@i+1;  end while;  select @ans;  end $  delimiter ; |
|  |
| 1. Write a procedure to find the sum of digits.   Input: - 5675  Output: - Twenty Three |
| drop procedure if exists pro99;  delimiter $  create procedure pro99(num int)  BEGIN  set @temp=num;  set @r=0;  while @temp != 0 do  set @r=@r+mod(@temp,10);  set @temp=floor(@temp/10);  end while;  select @r;  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 pro19;  delimiter $  create procedure pro19(d1 date,d2 DATE)  BEGIN  set @coun=0;  lb:loop  set @r=dayofweek(d1);  if @r=1 THEN  set @coun=@coun+1;  end if;  set d1=date\_add(d1,interval 1 DAY);  if d1=d2 then leave lb;  end if;  end loop lb;  select @coun;  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 ; |
|  |