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 newlineChar;  delimiter $  create PROCEDURE newlineChar(in string1 varchar(45))  BEGIN  declare x int;  declare z int;    set x := length(string1);  set z := 1;    lb1:LOOP  INSERT into tcv1 values(SUBSTR(string1,z,1));  if (z=x) THEN  leave lb1;  END IF;  set z := z+1;  END loop lb1;    SELECT \* from tcv1;  DELETE from tcv1;  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 |
| create table tcv1(letter varchar(45));  drop PROCEDURE if EXISTS sepChar;  delimiter $  create PROCEDURE sepChar(in string1 varchar(45))  BEGIN  declare x int;  declare z int;    set x := length(string1);  set z := 1;    lb1:LOOP  INSERT into tcv1 values(SUBSTR(string1,z,1));    if (z=x) THEN  leave lb1;  END IF;  set z := z+1;  END loop lb1;    SELECT group\_concat(letter) 'N,A,M,E' from tcv1;  DELETE from tcv1;  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 alphAnum;  delimiter $  CREATE PROCEDURE alphAnum(in string1 varchar(45))  BEGIN  DECLARE x int;  DECLARE z int;  DECLARE y varchar(45);  DECLARE w VARCHAR(45);    set x := LENGTH(string1);  set z := 1;  set y := '';  set w := '';    lb1:LOOP    IF (ASCII(SUBSTR(string1,z,1)) BETWEEN 65 AND 90) or (ASCII(SUBSTR(string1,z,1)) BETWEEN 97 AND 122) THEN  set y := CONCAT(y,substr(string1,z,1));  else  set w := CONCAT(w,SUBSTR(string1,z,1));  END IF;    if x=z THEN  leave lb1;  END IF;    set z := z+1;  END loop lb1;  insert into tcv1 values(y);  INSERT into tcv1 values(w);    SELECT \* from tcv1;  DELETE from tcv1;    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 s\_format;  delimiter $  CREATE PROCEDURE s\_format()  BEGIN  select CONCAT(substr(ename,1,1),'(',substr(ename,2,length(ename)-1),')',' is ',job) from emp;  end $  delimiter ; |
|  |
| 1. Write a procedure to print all upper and lower characters separately.   Input: - AbCdEfG  Output: - ACEG  bdf |
| drop PROCEDURE if EXISTS UpperLower;  delimiter $  CREATE PROCEDURE UpperLower(in string1 VARCHAR(45))  BEGIN  DECLARE x int;  DECLARE y int;  DECLARE w varchar(45);  DECLARE z varchar(45);      set x := length(string1);  set y := 1;  set w := '';  set z := '';    lb1:LOOP  IF (ASCII(substr(string1,y,1)) BETWEEN 65 and 90) THEN  set w = CONCAT(w, substr(string1, y, 1));  ELSE  set z = CONCAT(z, substr(string1, y, 1));  END IF;    set y := y+1;  END LOOP lb1;    INSERT into tcv1 VALUES(w);  INSERT into tcv1 VALUES(z);    SELECT \* from tcv1;  DELETE from tcv1;    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(string1 varchar(30))  begin  declare x varchar(100);  declare w varchar(100);  declare q varchar(100);  declare y int;  declare z int;    set y = length(string1);  set z = 1;  set x = '';  set w = '';  set q = '';  lb1: loop  if (ascii(substr(string1, z, 1)) between 48 and 57) then  set x = concat(x, substr(string1, z, 1));  end if;    if (ascii(substr(string1, z, 1)) in (97, 101, 105, 111, 117)) then  set w = concat(w, substr(string1, z, 1));  end if;    if (substr(string1, z, 1)= ' ') then  set q = concat(q, substr(string1, z, 1));  end if;    if z = y then  leave lb1;  end if;    set z = z + 1;  end loop lb1;  set @R = length(x);  set @O = length(w);  set @E = length(q);  insert into up values(string1,@O, @E, @R);  select \* from up;  delete from up;  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 rmchar;  delimiter $  create procedure rmchar(in string1 varchar(45))  BEGIN  declare x int;  declare z int;  declare w varchar(145);    set x := length(string1);  set z := 1;  set w := '';    lb1:LOOP  IF (ascii(substr(string1,z,1)) BETWEEN 65 and 90) or (ascii(substr(string1,z,1)) BETWEEN 97 and 122) THEN  set w := concat(w, substr(string1,z,1));  END IF;    if(x=z) THEN  leave lb1;  END if;  set z := z + 1;  end loop lb1;    select w 'WITHOUT digit and symbols';  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 in10;  delimiter $  create PROCEDURE in10()  BEGIN  declare id int;  declare message varchar(20);  set id := 1;      l0b1:LOOP  if (id%2=0) THEN  insert into y1 values (id, 'i is even');  ELSE  insert into y1 values (id, 'i is odd');  END if;    if (id=10) THEN  leave l0b1;  END IF;  set id := id + 1;  end loop l0b1;  select \* from y1;  delete from y1;  END $  delimiter ; |
|  |
| 1. Write a procedure to print five highest paid employees from the emp table using cursor. |
| drop PROCEDURE if exists h\_sal;  delimiter $  create procedure h\_sal()  BEGIN  declare \_empno, \_sal int;  declare \_ename, \_job varchar(45);  declare c1 cursor for select empno, ename, job, sal from emp order by sal desc limit 5;  open c1;  foor:LOOP  fetch c1 into \_empno, \_ename, \_job, \_sal;  select \_empno, \_ename, \_job, \_sal;  END LOOP foor;  close c1;  end $  delimiter ;  create table tcv2(c1 int, c2 varchar(45), c3 int);  ALTER : alter table tcv2 modify c1 int;  drop PROCEDURE if exists h\_sal1;  delimiter $  create procedure h\_sal1()  BEGIN  declare \_ename varchar(45);  DECLARE \_r1, \_sal INT;  declare x, record\_cnt int;  declare c1 cursor for select r1, ename, sal from (select dense\_rank() over(order by sal desc) R1, ename,sal from emp) e where R1<=5;    set x:=0;  select found\_rows() into record\_cnt;    open c1;      l1:LOOP  if (x<record\_cnt) THEN  fetch c1 into \_r1, \_ename, \_sal;  insert into tcv2 values(\_r1, \_ename, \_sal);  end if;  set x:= x+1;    if (x=record\_cnt) THEN  leave l1;  end if;    end loop l1;  close c1;    SELECT \* from tcv2;  DELETE from tcv2;  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. |
| SQL : create table emp10 like emp;  SQL: create table emp20 like emp;  SQL: create table emp30 like emp;  ((( IF you also want values along with structure : CREATE TABLE EMP10 AS SELECT \* FROM EMP)))  PL/SQL:  drop PROCEDURE if EXISTS splitrecords;  delimiter $  CREATE PROCEDURE splitrecords()  BEGIN  declare \_empno, \_deptno int;  declare \_ename, \_job varchar(45);    declare c cursor for select empno, deptno, ename, job from emp;  declare exit handler for 1329 select 'EOF';    open c;  foor:LOOP  fetch c into \_empno, \_deptno, \_ename, \_job;    if (\_deptno = 10 ) THEN    insert into emp10 values (\_empno, \_deptno, \_ename, \_job);  end if;    if (\_deptno = 20 ) THEN    insert into emp20 values (\_empno, \_deptno, \_ename, \_job);  end if;    if (\_deptno = 30 ) THEN    insert into emp30 values (\_empno, \_deptno, \_ename, \_job);  end if;    END loop foor;  close c;  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 dnen;  delimiter $  CREATE PROCEDURE dnen()  BEGIN  declare \_empno, \_deptno int;  declare \_ename, \_job varchar(45);    declare c cursor for select empno, deptno, ename, job from emp;  declare exit handler for 1329 select CONCAT(deptno, ' -> (', group\_concat(ename), ')') 'EMPLOYEE NAME'from emp group by deptno;    open c;  foor:LOOP  fetch c into \_empno, \_deptno, \_ename, \_job;  insert into emp10 values (\_empno, \_deptno, \_ename, \_job);    END loop foor;    close c;    END $  delimiter ; |
|  |
| 1. Write a procedure to accept customer number and display all his order. (Use customers and orders table) |
|  |
|  |
| 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(numb varchar(100))  begin  declare lstr int;  declare z int;  declare a varchar(45);  declare l varchar(45);  set lstr = length(numb);  set z = 1;  set a = '';  set l = '';  lb1:loop  set @s = substr(numb,z,1);    if (@s = 1) then  set a = 'One';  end if;    if (@s = 2) then  set a = 'Two';  end if;    if (@s = 3) then  set a = 'Three';  end if;    if (@s = 4) then  set a = 'Four';  end if;    if (@s = 5) then  set a = 'Five';  end if;    if (@s = 6) then  set a = 'Six';  end if;    if (@s = 7) then  set a = 'Seven';  end if;    if (@s = 8) then  set a = 'Eight';  end if;    if (@s = 9) then  set a = 'Nine';  end if;    if (@s = 0) then  set a = 'zero';  end if;    set l = concat(l, a,' ');      if (z=lstr) then  leave lb1;  end if;    set z = z + 1;    end loop lb1;  select l as 'Number into word';    end $  delimiter ; |
|  |
| 1. Write a procedure to find the sum of digits.   Input: - 5675  Output: - Twenty Three |
|  |
|  |
| 1. Write a procedure to find how many “Sundays” are present between two given dates.   Input: - Date1 and Date2  Output: - 3 Sunday’s |
|  |
|  |
| 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’ |
|  |
|  |