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 $  drop table if exists String1;  create table String1 (character1 varchar(1));  create procedure pro1(name1 varchar(50))  begin  declare ch varchar(1);  declare ct int;  set ct := 1;  loop1:loop  if ct <= length(name1) then  set ch := substring(name1,ct,1);  set ct := ct + 1;  insert into String1 values(ch);  else  leave loop1;  end if;  end loop loop1;  select \* from String1;  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(name1 varchar(50))  begin  declare ch varchar(50);  declare ct int;  set ct := 1;  set ch := substring(name1,ct,1);  set ct := ct + 1;  loop1:loop  if ct <= length(name1) then  set ch := concat(ch," , ",substring(name1,ct,1));  set ct := ct + 1;  else  leave loop1;  end if;  end loop loop1;  select ch as name;  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(name1 varchar(50))  begin  declare String1 varchar(50);  declare number1 int;  declare ch varchar(1);  declare ct int ;  set number1 := 1;  set String1 := ' ';  set ct := 1;  loop1:loop  if (ct <= length(name1)) then  set ch := substring(name1,ct,1);  if (ch between 'A' and 'z') then  set String1 := concat(String1,substring(name1,ct,1));  elseif (ch between 0 and 9) then  set number1 := (number1 \* 10 )+ch;  end if;  set ct := ct + 1;  else  leave loop1;  end if;  end loop loop1;  select substring(String1,2);  select substring(number1,2);  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 $  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 ;  or  -- Drop the procedure and table if they exist  DROP PROCEDURE IF EXISTS pro4;  DROP TABLE IF EXISTS String2;  -- Change the delimiter to create the table and procedure  DELIMITER $  -- Create the String2 table  CREATE TABLE String2 (c1 VARCHAR(100));  -- Drop the procedure and table if they exist  DROP PROCEDURE IF EXISTS pro4;  DROP TABLE IF EXISTS String2;  -- Change the delimiter to create the table and procedure  DELIMITER $  -- Create the String2 table  CREATE TABLE String2 (c1 VARCHAR(100));  -- Create the pro4 procedure  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;    -- Get the maximum row number  SELECT MAX(r1) INTO maxRow FROM (SELECT ROW\_NUMBER() OVER() r1 FROM emp) t1;    loop2: LOOP  IF ct <= maxRow THEN  -- Retrieve data from emp table  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;    -- Concatenate and insert into String2  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 and display the content of String2  SELECT \* FROM String2;  END $  -- Reset the delimiter to the default  DELIMITER ;  -- Call the procedure to execute it  CALL pro4(); |
|  |
| 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(name varchar(100))  begin  declare ch varchar(1);  declare ct int ;  declare lower1 varchar(50);  declare upper1 varchar(50);  set lower1 := " ";  set upper1 := " ";  set ct := 0;  loop3:loop  set ct= ct+1;  if ct <= length(name) then  set ch := substring(name,ct,1);  if ASCII(ch) >= 65 and ASCII(ch) <= 90 then  set upper1 := concat(upper1,ch);  elseif ASCII(ch) >=97 and ASCII(ch) <= 122 then  set lower1 := concat(lower1,ch);  end if;  else  leave loop3;  end if;  end loop loop3;  select substring(upper1,2);  select substring(lower1,2);  end $  delimiter ; |
|  |
| 1. Write a procedure to find the number of vowels, digits and white spaces |
| drop procedure if exists separation;  delimiter $  create PROCEDURE separation(string1 varchar(20))  begin  declare count1 int;  declare char1 varchar(1);  declare vowels VARCHAR(20);  declare digit VARCHAR(20);  declare count\_vowels int ;  declare count\_digit int ;  DECLARE count\_whiteSpace int;    set count\_vowels=0;  set count\_digit=0;  set count\_whiteSpace=0;  set vowels = " ";  set digit = " ";  set count1=0;    loop1:loop  set count1 = count1 + 1;  if count1 <= length(string1) then  set char1 = substring(string1,count1,1);  if char1 in ('a','e','i','o','u','A','E','I','O','U') THEN  set vowels = CONCAT(vowels, char1);  set count\_vowels = count\_vowels + 1;  ELSEIF char1 > 0 and char1 < 9 THEN  set digit = CONCAT(digit, char1);  set count\_digit = count\_digit + 1;  elseif char1 = " " THEN  set count\_whiteSpace = count\_whiteSpace +1;    end if ;  ELSE  leave loop1;  end if ;    end loop loop1;  select vowels,count\_vowels as 'Number of Vowels';  select digit,count\_digit as 'Number Of Digits';  select count\_whiteSpace as 'Number of Whitespaces';    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 rem\_char;  delimiter $  create procedure rem\_char(string1 VARCHAR(200))  begin  declare count1 int;  declare sep\_String varchar(20);  DECLARE char1 VARCHAR(1);    set count1 = 0;  set sep\_String= " ";  loop8: LOOP  set count1 = count1 + 1;  if count1 <= length(string1) THEN  set char1 = SUBSTR(string1,count1,1);  if ASCII(char1) >=65 and ASCII(char1) <=122 THEN  set sep\_String = concat(sep\_String,char1);  end if;  else  leave loop8;  end if;  end loop loop8;  select sep\_String;  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 insertion;  delimiter $  CREATE PROCEDURE insertion()  BEGIN  declare count1 int;  drop table if exists R;  create table R(id int, message varchar(20));  set count1=0;  loop9:loop  set count1 = count1 +1;  if count1 <= 10 then  if(count1 % 2 =0 ) THEN  insert into R values (count1, "i is even");  else  insert into R values (count1, "i is odd");  end if;  ELSE  leave loop9;  end if;  end loop loop9;  select \* from R;  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 pro12;  delimiter $  create procedure pro12(custId int)  Begin  declare flag bool;  select distinct true into flag from customers where cnum = custid;  if flag then  select orders.\* from customers,orders where customers.cnum=custid and customers.cnum = orders.cnum;  else  select "No Data Found";  end if;  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 int)  begin  declare inWords varchar(50);  declare x int;  set inWords := " ";  loop1:loop  if num > 0 then  set x := num % 10;  set num := num div 10;  if x = 1 then  set inWords := concat("one ",inWords);  elseif x = 2 then  set inWords := concat("two ",inWords);  elseif x = 3 then  set inWords := concat("three ",inWords);  elseif x = 4 then  set inWords := concat("four ",inWords);  elseif x = 5 then  set inWords := concat("five ",inWords);  elseif x = 6 then  set inWords := concat("six ",inWords);  elseif x = 7 then  set inWords := concat("seven ",inWords);  elseif x = 8 then  set inWords := concat("eight ",inWords);  elseif x = 9 then  set inWords := concat("nine ",inWords);  elseif x = 0 then  set inWords := concat(" ",inWords);  end if;  else  leave loop1;  end if;  end loop loop1;  select inWords;  end $  delimiter ; |
|  |
| 1. Write a procedure to find the sum of digits.   Input: - 5675  Output: - Twenty Three |
| drop function if exists firstNumber;  delimiter $  create function firstNumber(x bigint) returns varchar(20)  deterministic  begin  if x = 1 then  return "one";  elseif x = 2 then  return "two";  elseif x = 3 then  return "three";  elseif x = 4 then  return "four";  elseif x = 5 then  return "five";  elseif x = 6 then  return "six";  elseif x = 7 then  return "seven";  elseif x = 8 then  return "eight";  elseif x = 9 then  return "nine";  elseif x = 0 then  return " ";  end if;  end $  delimiter ;  drop procedure if exists pro14;  delimiter $  create procedure pro14(number1 int)  begin  declare inWords varchar(50);  declare num int;  declare temp int ;  declare tensPlace int;  set inWords := " ";  set num := 0;  loop1:loop  if number1 > 0 then  set temp := number1 % 10;  set number1 := number1 div 10;  set num := num + temp;  else  leave loop1;  end if;  end loop;  set tensPlace := num div 10;  if tensPlace = 2 then  set inWords := concat(inWords,"twenty");  elseif tensPlace = 3 then  set inWords := concat(inWords,"thirty");  elseif tensPlace = 4 then  set inWords := concat(inWords,"forty");  elseif tensPlace = 5 then  set inWords := concat(inWords,"fifty");  elseif tensPlace = 6 then  set inWords := concat(inWords,"sixty");  elseif tensPlace = 7 then  set inWords := concat(inWords,"seventy");  elseif tensPlace = 8 then  set inWords := concat(inWords,"eighty");  elseif tensPlace = 9 then  set inWords := concat(inWords,"ninty");  else  set inWords := concat(inWords," ");  end if ;  if tensPlace = 1 then  if num = 10 then  set inWords := concat(inWords,"ten");  elseif num = 11 then  set inWords := concat(inWords,"eleven");  elseif num = 12 then  set inWords := concat(inWords,"twelve");  elseif num = 13 then  set inWords := concat(inWords,"thirteen");  elseif num = 14 then  set inWords := concat(inWords,"forteen");  elseif num = 15 then  set inWords := concat(inWords,"fifteen");  elseif num = 16 then  set inWords := concat(inWords,"sixteen");  elseif num = 17 then  set inWords := concat(inWords,"seventeen");  elseif num = 18 then  set inWords := concat(inWords,"eighteen");  elseif num = 19 then  set inWords := concat(inWords,"nineteen");  end if;  end if;  if tensplace != 1 then  set inWords := concat(inWords," ",firstNumber(num % 10));  end if;  select inWords;  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 ; |
|  |