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(name varchar(20))  BEGIN  declare a char;  declare x int;  set x = 0;  lbl:LOOP  set x = x+1;  set @a = substr(name, x, 1);  select @a;    if x>length(name)-1  then  leave lbl;  end if;  end loop lbl;  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(name varchar(20))  BEGIN  declare a varchar(30);  declare x int;  set x = 0;    set @p=" ";  lbl:LOOP  set x = x+1;  set a = substr(name, x, 1);    set @p =concat(@p,a,',');  /\*set a=concat(group\_concat(c.name),",") ;  select a;  select ","; \*/    if x>length(name)-1  then  leave lbl;  end if;  end loop lbl;    SELECT @p;  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(name varchar(20))  BEGIN  declare a varchar(30);  declare x int;  set x = 0;    set @p=" ";  set @q=" ";  lbl:LOOP  set x = x+1;  set a = substr(name, x, 1);    if a between 'a' and 'z' then  set @p =concat(@p,a);  else  set @q = concat(@q,a);  end if;  if x>length(name)-1 then  leave lbl;  end if;  end loop lbl;    SELECT @p;  SELECT @q;  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 string3;  delimiter $  create procedure string3()  begin  select concat(left(ename, 1),"(",right(ename,length(ename)-1),") ","is ", job) as employee 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 pro1;  delimiter $  create procedure pro1(name varchar(20))  BEGIN  declare a varchar(30);  declare x int;  set x = 0;    set @p=" ";  set @q=" ";  lbl:LOOP  set x = x+1;  set a = substr(name, x, 1);    if ascii(a) between 65 and 90 then  set @p =concat(@p,a);  else  set @q = concat(@q,a);  end if;  if x>length(name)-1 then  leave lbl;  end if;  end loop lbl;    SELECT @p;  SELECT @q;  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(name varchar(200))  BEGIN  declare a varchar(300);  declare x int;  declare v int;  declare d int;  declare s int;  set x=0;  set v=0;  set d=0;  set s=0;    set @p=0;  set @q=0;  set @r=0;  lbl:LOOP  set x = x+1;  set a = substr(name, x, 1);    if a in ('a','e','i','o','u','A','E','I','O','U') then  set v=v+1;  set @p=v;  end if;    if a between '0' and '9' then  set d=d+1;  set @q=d;  end if;  IF a = ' ' then  set s=s+1;  set @r=s;  end if;  if x>length(name)-1 then  leave lbl;  end if;  end loop lbl;    SELECT @p;  SELECT @q;  SELECT @r;  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(name varchar(200))  BEGIN  declare a varchar(300);  declare x int;  set x = 0;    set @p=" ";  set @q=" ";  lbl:LOOP  set x = x+1;  set a = substr(name, x, 1);    if a between 'a' and 'z' or a between 'A' and 'Z' then  set @p =concat(@p,a);    end if;  if x>length(name)-1 then  leave lbl;  end if;  end loop lbl;    SELECT @p;    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 ques8;  delimiter $  create procedure ques8()  BEGIN  declare x int;  set x := 0;  create table q(id int, message varchar(20));  lbl:loop  set x := x+1;  if x<=10 then  if x%2=1 then  insert into q VALUES(x,"i is odd");  else  insert into q VALUES(x,"i is even");  end if;  else  leave lbl;  end if;  end loop lbl;  end $  delimiter ; |
|  |
| 1. Write a procedure to print five highest paid employees from the emp table using cursor. |
| drop procedure if exists ques9;  delimiter $  create procedure ques9()  begin  declare x int;  declare \_ename varchar(50);  declare \_sal int;  declare c1 cursor for select \* from( select dense\_rank() over(order by sal desc) R1 , ename, sal from emp)e where R1<=5;  open c1;  l1:loop  fetch c1 into x, \_ename, \_sal;  select x, \_ename, \_sal;  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 ques10;  delimiter $  create procedure ques10()  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;  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 ques11;  delimiter $  create procedure ques11()  BEGIN  declare z int;  declare \_deptno int;  set \_deptno := 0;  l1: loop  set \_deptno := \_deptno + 10;  set z := (select count(deptno) from emp where deptno = \_deptno);  if z>0 and \_deptno <=50 THEN  select concat(\_deptno,'->','(',group\_concat(ename),')') as emply from emp where deptno =\_deptno;  else if z=0 and \_deptno <=50 THEN  select concat(\_deptno,'->','(','No employee work in department ', \_deptno, ')') as emply;  ELSE  leave l1;  end if;  end if;  end loop l1;  end $  delimiter ; |
|  |
| 1. Write a procedure to accept customer number and display all his order. (Use customers and orders table) |
| drop procedure if exists ques12;  delimiter $  create procedure ques12(x int)  begin  declare flag bool;  select true into flag from customers where cnum = x;  if flag then  select o.\* from orders o , customers c where c.cnum = x and c.cnum = o.cnum;  ELSE  select "Record not 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 ques13;  delimiter $  create procedure ques13(num int)  begin  declare x int;  declare word varchar(200);  set x :=0;  set word :="";    lbl: loop  if num != 0 then  set x := num%10;  set num := num div 10;  if x = 0 then  set word := concat("zero"," ",word);  else if x = 1 then  set word := concat("one"," ",word);  else if x = 2 then  set word := concat("two"," ",word);  else if x = 3 then  set word := concat("three"," ",word);  else if x = 4 then  set word := concat("four"," ",word);  else if x = 5 then  set word := concat("five"," ",word);  else if x = 6 then  set word := concat("six"," ",word);  else if x = 7 then  set word := concat("seven"," ",word);  else if x = 8 then  set word := concat("eight"," ",word);  else if x = 9 then  set word := concat("nine"," ",word);  end if;  end if;  end if;  end if;  end if;  end if;  end if;  end if;  end if;  end if;  ELSE  leave lbl;  end if;  end loop lbl;  SELECT word as numbers\_in\_words;  end $  delimiter ; |
|  |
| 1. Write a procedure to find the sum of digits.   Input: - 5675  Output: - Twenty Three |
| drop procedure if exists ques14;  delimiter $  create procedure ques14(num int)  begin  declare x int;  declare word varchar(2000);  declare \_sum int;  declare ones int;  declare place int;  set \_sum := 0;  set x :=0;  set word := "";    lbl: loop  if num != 0 then  set x := num%10;  set \_sum := \_sum + x;  set num := num div 10;  else  leave lbl;  end if;  end loop lbl;  select \_sum;    set place := \_sum div 10;    if place = 2 then  set word := concat(word,"twenty");  elseif place = 3 then  set word := concat(word,"thirty");  elseif place = 4 then  set word := concat(word,"forty");  elseif place = 5 then  set word := concat(word,"fifty");  elseif place = 6 then  set word := concat(word,"sixty");  elseif place = 7 then  set word := concat(word,"seventy");  elseif place = 8 then  set word := concat(word,"eighty");  elseif place = 9 then  set word := concat(word,"ninety");  end if;    if place = 1 then  if \_sum = 10 then  set word := concat(word,"ten");  elseif \_sum = 11 then  set word := concat(word,"eleven");  elseif \_sum = 12 then  set word := concat(word,"twelve");  elseif \_sum = 13 then  set word := concat(word,"thirteen");  elseif \_sum = 14 then  set word := concat(word,"fourteen");  elseif \_sum = 15 then  set word := concat(word,"fifteen");  elseif \_sum = 16 then  set word := concat(word,"sixteen");  elseif \_sum = 17 then  set word := concat(word,"seventeen");  elseif \_sum = 18 then  set word := concat(word,"eighteen");  elseif \_sum = 19 then  set word := concat(word,"nineteen");  end if;  end if;        set ones := \_sum % 10;  if place != 1 then  if ones = 1 then  set word := concat(word," ","one");  elseif ones = 2 then  set word := concat(word," ","two");  elseif ones = 3 then  set word := concat(word," ","three");  elseif ones = 4 then  set word := concat(word," ","four");  elseif ones = 5 then  set word := concat(word," ","five");  elseif ones = 6 then  set word := concat(word," ","six");  elseif ones = 7 then  set word := concat(word," ","seven");  elseif ones = 8 then  set word := concat(word," ","eight");  elseif ones = 9 then  set word := concat(word," ","nine");  end if;  end if;    select word;  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 ques15;  delimiter $  create procedure ques15(date1 date, date2 date)  begin  declare \_count int;  set \_count := 0;    lbl : loop  if date1 <= date2 then  if dayname(date1) = 'Sunday' then  set \_count := \_count + 1;  set date1 := date1 + interval 1 day;  else  set date1 := date1 + interval 1 day;  end if;  else  leave lbl;  end if;  end loop lbl;  select concat(\_count," - Sundays") as Ndays;  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 ques16;  delimiter $  create procedure ques16(date1 date, \_weekday varchar(100))  begin  lbl:loop  if dayname(date1) = \_weekday then  select date1 as Weekends;  leave lbl;  else  set date1 := date1 + interval 1 day;  end if;  end loop lbl;  end $  delimiter ; |
|  |