Assignment

Sept23/ DBT/126.1

Database Technologies

Diploma in Advance Computing

September 2023

**Procedure and Function**

|  |
| --- |
| * Write a procedure to accept a string and print all characters in separate lines.   Input: - Ram  Output: - R  a  m |
| drop procedure if exists string;  delimiter $  create procedure string(str varchar(20))  BEGIN  declare x int;  set x:=0;  create table a1(name varchar(20));  l: LOOP  set x:= x+1;  if x > length(str) THEN leave l;  ELSE  set @z:=substr(str, x, 1);  insert into a1 values(@z);  end if;  end loop l;  select \* from a1;  end $  delimiter ; |
|  |
| * 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 s;  delimiter $  create procedure s(str varchar(20))  BEGIN  declare x int;  declare t varchar(20);  set x:=1;  set t='';  l: LOOP  if x > length(str) THEN leave l;  ELSE  set @z:=substr(str, x, 1);  set x:= x+1;  set t := concat(t,@z,',');  end if;  end loop l;  select t as 'Seperated Char';  end $  delimiter ; |
|  |
| * Write a procedure to accept an alpha numeric string and separate number and characters of the string.   Input: - SAL1234EEL  Output: - SALEEL  1234 |
| create procedure pro1(str varchar(30))  BEGIN  declare x int;  set x:=0;  set @strings :="";  set @num :="";  l:loop  set x:= x+1;  if substr(str, x, 1) >='0' and substr(str, x, 1) <='9'  then set@num := concat(@num, '', substr(str, x, 1));    else  set @strings := concat(@strings,'', substr(str, x, 1));  end if;  if x>length(str) then leave l;  end if;  end loop l;  select @num;  select @strings;  end $  delimiter ; |
|  |
| * 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(str1 varchar(30), str2 varchar(30))  BEGIN  set @x:=1;  select concat (substr(str1,@x,1),'(',substr(str1,@x+1,length(str1)), ') is ', str2) as 'Output' ;  end $  delimiter ; |
|  |
| * Write a procedure to print all upper and lower characters separately.   Input: - AbCdEfG  Output: - ACEG  bdf |
| drop procedure if exists up;  delimiter $  create procedure up(str varchar(50))  begin  declare x int;  declare str1 varchar(100);  declare str2 varchar(100);  set str1:="";  set str2:="";  set x:=0;  l:loop  set x:=x+1;  if ascii(substr(str,x,1))>=65 and ascii(substr(str,x,1))<=90 then  set str1:= concat(str1,substr(str,x,1));  else if  ascii(substr(str,x,1))>=97 and ascii(substr(str,x,1))<=122 then  set str2:= concat(str2,substr(str,x,1));  else leave l;  end if;  end if;  end loop l;  select str1;  select str2;  end $  delimiter ; |
|  |
| * Write a procedure to find the number of vowels, digits and white spaces |
| drop procedure if exists count1;  delimiter $  create procedure count1(str varchar(50))  BEGIN  declare x int;  set x:=0;  set @digits:="";  set @vow:="";  set @spaces:="";  l:LOOP  set x:=x+1;  if substr(str,x,1)='a' or substr(str,x,1)='o' or substr(str,x,1)='i' or substr(str,x,1)='e' or substr(str,x,1)='u'  then  set @vow:= @vow+1;  end if;  if substr(str, x, 1)>='0' and substr(str, x, 1)<='9' THEN  set @digits:= @digits+1;  end if;  if substr(str, x, 1)=' ' THEN  set @spaces:= @spaces +1;  end if;  if x>length(str) THEN  leave l;  end if;  end loop l;  select @vow;  select @digits;  select @spaces;  end $  delimiter ; |
|  |
| * 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 ; |
|  |
| * 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 ; |
|  |
| * Write a procedure to print five highest paid employees from the emp table using cursor. |
| drop procedure if exists empl;  delimiter $  create procedure empl()  BEGIN  declare \_empno, \_sal int;  declare \_ename varchar(20);  declare c1 cursor for select empno, ename, sal from emp order by sal desc limit 5;  declare exit handler for 1329 select "EOF";  open c1;  l:loop  fetch c1 into \_empno, \_ename, \_sal;  select \_empno, \_ename, \_sal;  end loop l;  close c1;  end $  delimiter ; |
|  |
| * 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 empll;  delimiter $  create procedure empll()  BEGIN  declare \_empno,\_mgr,\_sal,\_comm,\_deptno,\_bonusid int;  declare \_ename,\_job,\_username,\_pwd,\_phone varchar(20);  declare \_hiredate date;  declare \_gender char(1);  declare \_isActive tinyint(1);  declare c2 cursor for select \* from emp where deptno in(10,20,30);  open c2;  l:loop  fetch c2 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);  else if \_deptno=20 then  insert into emp20 values(\_empno,\_ename,\_gender,\_job,\_mgr,\_hiredate,\_sal,\_comm,\_deptno,\_bonusid,\_username,\_pwd,\_phone,\_isActive);  else if \_deptno=30 then  insert into emp30 values(\_empno,\_ename,\_gender,\_job,\_mgr,\_hiredate,\_sal,\_comm,\_deptno,\_bonusid,\_username,\_pwd,\_phone,\_isActive);  end if;  end if;  end if;  end loop l;  close c2;  end $  delimiter ; |
|  |
| * 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 dept;  delimiter $  create procedure dept()  begin  declare \_deptno int;  declare \_ename varchar(100);  declare c1 cursor for select deptno, group\_concat(ename) from emp group by deptno;  open c1;  l:loop  fetch c1 into \_deptno, \_ename;  select \_deptno, group\_concat(\_ename);  end loop l;  close c1;  end $  delimiter ; |
|  |
| * Write a procedure to accept customer number and display all his order. (Use customers and orders table) |
| drop procedure if exists cus;  delimiter $  create procedure cus(x int)  BEGIN  declare \_cnum, \_snum, \_onum int;  declare \_amt, \_rating float;  declare \_cname, \_city,\_type varchar(20);  declare \_odate datetime;  declare c1 cursor for select c.cnum,c.cname,c.city,c.rating, c.snum, o.onum,o.amt,o.odate,o.type from customers c join orders o on c.cnum=o.cnum where c.cnum=x;  declare exit handler for 1329 select "EOF";  open c1;  l:loop  fetch c1 into \_cnum,\_cname,\_city ,\_rating,\_amt,\_snum,\_onum,\_odate,\_type;  select \_cnum,\_cname,\_city ,\_rating,\_amt,\_snum,\_onum,\_amt,\_odate,\_type;  end loop l;  close c1;  end $  delimiter ; |
|  |
| * Write a procedure to convert numbers into word   Input: - 45234  Output: - Four Five Two Three Four |
|  |
|  |
| * Write a procedure to find the sum of digits.   Input: - 5675  Output: - Twenty Three |
|  |
|  |
| * Write a procedure to find how many “Sundays” are present between two given dates.   Input: - Date1 and Date2  Output: - 3 Sunday’s |
|  |
|  |
| * 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’ |
|  |
|  |