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(str varchar(20))
BEGIN
declare x int;
set x:=1;
lb1:LOOP
 select substr(str,x,1) result1;
 set x=x+1;
if x>length(str) THEN
leave lb1 ;
end if;
end  loop lb1;
end $
delimiter ;
```

2. 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(str varchar(20))
BEGIN

declare x int;
set @y:="";
set x:=1;
lb1:LOOP
        if x=1 then
         SET @y:= CONCAT(@y,'',substr(str,x,1)) ;
        else
        SET @y:= CONCAT(@y,',',substr(str,x,1)) ;
         end if;
         set x=x+1;

if x>length(str) THEN
leave lb1 ;

end if;
end  loop lb1;
```

```
end $
delimiter ;
```

3. 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 @ch:="";
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 @ch:=concat(@ch,'',substr(str1,x,1));

end if;
set x:=x+1;



if x > length(str1) then leave l;

end if;
end loop l;
end .
delimiter ;
```

4. 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()

  BEGIN
        -- select * from emp;
 SELECT concat(substr(ename,1,1),'(',substr(ename,2),')', '' is '' , job) from emp;



        end .
        delimiter ;
```

5. 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 @num1:="";
set @num2:="";

l:LOOP
        if ascii( substr(str1,x,1))>=ascii('A')  AND ascii(substr(str1,x,1))<= ascii('Z') THEN
                set @num1:=concat(@num1,substr(str1,x,1));
        ELSE
                set @num2:=concat(@num2,substr(str1,x,1));

        end if;
        set x:=x+1;



        if x > length(str1) then leave l;

        end if;
end loop l;
        end .
        delimiter ;
```

6. Write a procedure to find the number of vowels, digits and white spaces

```
drop procedure if exists pro1;
delimiter .

create procedure pro1(str1 varchar(200))

begin
 declare x int;
 set @digit:=0;
 set @vowles:=0;
 set @spaces:=0;
 set x:=1;

 l:LOOP

 if substr(str1,x,1)='a' or substr(str1,x,1)='e' or substr(str1,x,1)='i' or substr(str1,x,1)='o' or
substr(str1,x,1)='u' THEN set @vowles:=@vowles+1;
 end if;
 if substr(str1,x,1)=' ' then set @spaces:=@spaces+1;
 end if;
 if substr(str1,x,1)>='0' and substr(str1,x,1)<='9' THEN set @digit:=@digit+1;

 end if;
 set x:=x+1;
 if x> length(str1) then
```

```
 leave l;
 end if;
 end loop l;



end .
delimiter ;
```

7. 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(str1 varchar(500))
begin
declare x int;
set x:=1;
set @ex:="";
lb:LOOP
if not(  substr(str1,x,1)>='a' and  substr(str1,x,1)<='z' ) then
set @ex:=concat(@ex,substr(str1,x,1));
end if;
set x:=x+1;
if x> length(str1) then
leave lb ;
end if;
end  loop lb ;
end $
delimiter ;
```

8. 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 | is i 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 pro1;
delimiter $
create procedure pro1(z int)
begin
declare x int;
set x:=1;

lb:LOOP
if MOD(X,2)=0 then
INSERT into r15 values(x, concat(x," is even"));
```

```
ELSE
INSERT into r15 values(x, concat(x," is odd"));
end if;
set x:=x+1;
if x> z then
leave lb ;
end if;
end  loop lb ;
end $
delimiter ;
```

9. Write a procedure to print five highest paid employees from the emp table using cursor.

```
-- first 5 highest paid employee
drop procedure if exists pro2;
delimiter $
create procedure pro2(x int)
BEGIN
declare _ename varchar(200);
declare _sal int;
declare  c1 cursor for select ename,sal from  emp order by sal desc ;
open c1;
l:LOOP
fetch c1 into _ename,_sal;
select _ename,_sal;
set x=x-1;
if x=0 THEN
leave l;
end if;
end loop l;
end $
delimiter ;
```

10. 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.

```
delimiter $
create procedure pro3()
BEGIN
  declare _empno,_ename,_deptno varchar(100);
  declare _sal int;
  declare c1 cursor for select  empno,ename,sal,deptno from emp where deptno  =10;
  declare c2 cursor for select  empno,ename,sal,deptno from emp where deptno =20;
  declare c3 cursor for select  empno,ename,sal,deptno from emp where deptno =30;
declare exit handler for 1329 select "EOF";
open c1;
open c2;
open c3;
  l:loop
   fetch c1 into _empno,_ename,_sal,_deptno;
   INSERT INTO EMP10(EMPNO,ENAME,SAL,DEPTNO)
   VALUES( _empno,_ename,_sal,_deptno);
   fetch c2 into _empno,_ename,_sal,_deptno;
   INSERT INTO EMP20(EMPNO,ENAME,SAL,DEPTNO)
   VALUES( _empno,_ename,_sal,_deptno);
   fetch c3 into _empno,_ename,_sal,_deptno;
```

```
  INSERT INTO EMP30(EMPNO,ENAME,SAL,DEPTNO)
   VALUES( _empno,_ename,_sal,_deptno);
end loop l;
close c1;
close c2;
close c3;

end $
delimiter ;
```

11. 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)

12. 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(_cno int)
begin



select * from customers natural join orders where cnum=_cno;
end $
delimiter ;
```

13. Write a procedure to convert numbers into word
    Input: - 45234
    Output: - Four Five Two Three Four

```
-- 126.1 13
drop procedure if exists pro5;
delimiter $
create procedure pro5(num int)
 begin
 declare z varchar(100);
 declare x int;
 declare len1 int;
 declare curdigit int;
 set z="";
 set x:=0;
 set num:=reverse(num);
 set len1:=length(num);
l:LOOP
  set x:=x+1;
   if x>len1
then
leave l;
end if;
 set curdigit := mod(num,10);
```

```
  set num=num DIV 10;
if curdigit=1 then
 set z:=concat(z,"One ");
   end if;
 if curdigit=2 then
  set z:=concat(z,"Two ");
  end if;
  if curdigit=3 then
  set z:=concat(z,"Three ");
  end if;
if curdigit=4 then
  set z:=concat(z,"Four ");
 end if;
if curdigit=5 then
  set z:=concat(z,"Five ");
   end if;
if curdigit=6 then
  set z:=concat(z,"Six ");
end if;
if curdigit=7 then
  set z:=concat(z,"Seven ");
end if;
if curdigit=8 then
  set z:=concat(z,"Eight ");
end if;
if curdigit=9 then
  set z:=concat(z,"Nine ");
  end if;
  end loop l;
select z;
end $
delimiter ;
```

14. Write a procedure to find the sum of digits.
    Input: - 5675
    Output: - Twenty Three

```
drop procedure if exists pro5;
delimiter $
create procedure pro5(num int)
  begin
  declare tsum, x , p, len1, curdigit, tlen int;
  declare tchar varchar(100);
  set tchar = "";
  set p=0;
  set tsum=0;
  set x:=0;
  set len1:=length(num);
l:LOOP
   set x:=x+1;
    if x>len1 then
leave l;
end if;
  set curdigit := mod(num,10);
  set tsum:=tsum+curdigit;
  set num=num DIV 10;
  end loop l;
```

```
 select tsum;
 set tlen:=length(tsum);
 ll:loop
 set p:=p+1;
 IF p>tlen -1
 then  leave ll;
 end if;
  if tlen=2 THEN
  if tsum=10 then
set tchar:=concat(tchar,"Ten ");
  end if;
if tsum=11 then
 set tchar:=concat(tchar,"Eleven ");
 end if;
 if tsum=12 then
 set tchar:=concat(tchar,"twelve ");
 end if;
if tsum=13 then
 set tchar:=concat(tchar,"thirteen ");
end if;
if tsum=14 then
 set tchar:=concat(tchar,"fourteen ");
  end if;
if tsum=15 then
 set tchar:=concat(tchar,"fifteen ");
end if;
if tsum=16 then
 set tchar:=concat(tchar,"sixteen ");
end if;
if tsum=17 then
 set tchar:=concat(tchar,"seventeen ");
end if;
if tsum=18 then
 set tchar:=concat(tchar,"Eighteen ");
 end if;
 if tsum=19 then
 set tchar:=concat(tchar,"Nineteen ");
 end if;
  end if;
 end loop ll;
 select tchar;

end $
delimiter ;
```

15. 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 pro7;
delimiter $
create procedure pro7(d1 date ,d2 date)
BEGIN
 select concat( count(hiredate)," Sundays") from emp where hiredate between(d1) and (d2) and
dayname(hiredate)='Sunday';

 end $
```

```
delimiter ;
```

16. 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 pro8;
delimiter $
create procedure pro8(d1 date ,dayn varchar(30))
BEGIN
declare x int;
set x:=1;
l:LOOP

if dayname(d1)=dayn THEN
SELECT d1 da,dayname(d1) dn;
leave l;
end if;

set d1:= date_add(d1,interval x day);
end loop l;

 end $
 delimiter ;
```