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 Print1;  delimiter $  CREATE PROCEDURE Print1(in p\_input\_string VARCHAR(20))  BEGIN    declare v\_length int;  declare v\_char varchar(1);  declare i int;  set v\_length := LENGTH(p\_input\_string);  set i := 1;  l1: loop  if i <= v\_length then  set v\_char := SUBSTR(p\_input\_string, i, 1);  select v\_char as output;  set i := i + 1;  else  leave l1;  end if;  end loop l1;      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 Print1;  delimiter $  CREATE PROCEDURE Print1(in p\_input\_string VARCHAR(20))  BEGIN    declare v\_length int;  declare v\_char varchar(1);  declare i int;  set v\_length := LENGTH(p\_input\_string);  set i := 1;  SET @output := '';  l1: loop  if i <= v\_length then  set v\_char := SUBSTR(p\_input\_string, i, 1);  set i := i + 1;  SET @output := CONCAT(@output, v\_char,",");  else  leave l1;  end if;  end loop l1;    select @output as output;  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 Print1;  delimiter $  CREATE PROCEDURE Print1(in p\_input\_string VARCHAR(20))  BEGIN    declare v\_length int;  declare v\_char varchar(1);  declare i int;  set v\_length := LENGTH(p\_input\_string);  set i := 1;  SET @outputNum := '';  SET @outputStr := '';  l1: loop  if i <= v\_length then  set v\_char := SUBSTR(p\_input\_string, i, 1);  if ascii(v\_char) >= 49 and ascii(v\_char) <=57 then    SET @outputNum := CONCAT(@outputNum, v\_char);  else    SET @outputStr := CONCAT(@outputStr, v\_char);  end if;  set i := i + 1;  else  leave l1;  end if;  end loop l1;    select @outputNum as output1;  select @outputStr as str1;  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 Print1;  delimiter $  CREATE PROCEDURE Print1()  BEGIN    select concat(left(ename,1),"(",substr(ename,2,length(ename)),")"," is " ,job)as output 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 Print1;  delimiter $  CREATE PROCEDURE Print1(in str1 varchar(30))  BEGIN    declare i int;  declare str\_len int;  declare ch varchar(1);    set i := 1;  set str\_len := length(str1);  set @uppcase := '';  set @lowcase := '';      l1:LOOP    if i <= str\_len THEN  set ch := substr(str1,i,1);    if ascii(ch) >=97 and ascii(ch) <= 122 THEN  set @lowcase:=concat(@lowcase,ch);  set i := i+1;      else  set @uppcase:=concat(@uppcase,ch);  set i := i+1;  end if;  else  leave l1;  end if;    end loop l1;  select @uppcase as "upper case";  select @lowcase as "lower case";    END $  delimiter ; |
|  |
| 1. Write a procedure to find the number of vowels, digits and white spaces |
| drop procedure if exists Print1;  delimiter $  CREATE PROCEDURE Print1(in str1 varchar(30))  BEGIN    declare i int;  declare str\_len int;  declare ch varchar(1);    set i := 1;  set str\_len := length(str1);  set @digits := '';  set @vowels := '';  set @blanksp := '';    l1:LOOP    if i <= str\_len THEN    set ch := substr(str1,i,1);      if ch in ('a','e','i','o','u') then  set @vowels:=concat(@vowels,ch);    end if;    if ascii(ch)>=48 and ascii(ch)<=57 then  set @digits:=concat(@digits,ch);    end if;    if ch = ' ' then  set @blanksp:=concat(@blanksp,ch);      end if;  set i := i+1;  else  leave l1;  end if;    end loop l1;  select @digits as "digits", length(@digits) as "count";  select @vowels as "Vowels",length(@vowels) as "count";  select @blanksp as "blanksp",length(@blanksp) as "count";    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 Print1;  delimiter $  CREATE PROCEDURE Print1(in str1 varchar(30))  BEGIN    declare i int;  declare str\_len int;  declare ch varchar(1);    set i := 1;  set str\_len := length(str1);  set @result := '';        l1:LOOP    if i <= str\_len THEN  set ch := substr(str1,i,1);    if (ascii(ch) >= 97 and ascii(ch) <= 122) or (ascii(ch)>=65 and ascii(ch)<=90) THEN  set @result:=concat(@result,ch);    else    set i := i+1;  end if;  else  leave l1;  end if;    end loop l1;  select @result as "result";      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 string2;  delimiter $  create procedure string2()  BEGIN  declare i int;  set i:=1;  drop table if exists t;  create table t(id int auto\_increment primary key,message varchar(30));  l1:loop  if i <=10 then    if (i%2)=0 then    insert into t(message) values("id is even");  ELSE    insert into t(message) values("id is odd");  end if;  set i:=i+1;    else  leave l1;  end if;    end loop l1;  select \* from t;  end $  delimiter ; |
|  |
| 1. Write a procedure to print five highest paid employees from the emp table using cursor. |
| drop procedure if exists Print1;  delimiter $  CREATE PROCEDURE Print1()  BEGIN  declare \_ename,\_job varchar(30);  declare \_sal,\_deptno int;  declare count1 int;  declare c1 cursor for select ename,job,sal,deptno from emp;    set count1 := 0;  open c1;    l1 : loop    if count1<=5 then  fetch c1 into \_ename,\_job,\_sal,\_deptno;    select \_ename "Name",\_job "Job",\_sal "Sal",\_deptno "deptno";    set count1 := count1 + 1;  ELSE  leave l1;  end if;  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. |
|  |
|  |
| 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 pro1;  delimiter $  create procedure pro1()  begin  select concat(deptno,'->','(',group\_concat(ename),')') as output from emp group by deptno;  end $  delimiter ; |
|  |
| 1. Write a procedure to accept customer number and display all his order. (Use customers and orders table) |
| drop procedure if exists Print1;  delimiter $  CREATE PROCEDURE Print1(customerID int)  BEGIN    select \* from customers join orders where orders.cnum=customers.cnum and orders.cnum=customerID;    END $  delimiter ; |
|  |
| 1. Write a procedure to convert numbers into word   Input: - 45234  Output: - Four Five Two Three Four |
| drop procedure if exists Print1;  delimiter $  CREATE PROCEDURE Print1(num int)  BEGIN      declare lastDigit int;  declare alldigit varchar(200) default '';      drop table if exists word;  create table word(num int auto\_increment primary key,numWord varchar(25));  insert into word(numWord) values("One"),("Two"),("Three"),("Four"),("Five"),("Six"),("Seven"),("Eight"),("Nine");    l1:loop  if num > 0 THEN  set lastDigit:= num%10;  set num:=(num div 10);  if lastDigit=0 then  set alldigit := concat('zero', ' ' ,alldigit);  else  set alldigit := concat((select numWord from word where lastDigit=word.num), ' ' ,alldigit);  end if;  ELSE  leave l1;  end if;  end loop l1;  select alldigit;  END $  delimiter ; |
|  |
| 1. Write a procedure to find the sum of digits.   Input: - 5675  Output: - Twenty Three |
| drop procedure if exists Print1;  delimiter $  CREATE PROCEDURE Print1(num int)  BEGIN      declare lastDigit int;  declare lastDigit2 int;  declare countDigit int;  declare alldigit varchar(200) default '';  declare sumDigit int;    drop table if exists decimalPlace;  create table decimalPlace(num int auto\_increment primary key,wrd varchar(25),decWord varchar(25),forOne varchar(25));  insert into decimalplace(wrd,decWord,forOne) values("One","Ten","Eleven"),("Two","Twenty","Twelve"),("Three","Thirty","Thirteen"),("Four","Fourty","Fourteen"),("Five","Fifty","Fifteen"),("Six","Sixty","Sixteen"),("Seven","Seventy","Seventeen"),("Eight","Eighty","Eighteen"),("Nine","Ninty","Nineteen"),("Ten","Ten","Ten");  l1:loop  if num > 0 THEN  set lastDigit:= num%10;  set sumDigit := sumDigit + lastDigit;  set num:=(num/10);  ELSE  leave l1;  end if;  end loop l1;        l2:loop  if sumDigit > 0 THEN  set lastDigit2:= sumDigit%10;  set sumDigit:=(sumDigit/10);  set countDigit := countDigit +1;    set alldigit := concat((select numWord from decimalPlace where lastDigit2=decimalPlace.num ), ' ' ,alldigit);  ELSE  leave l2;  end if;  end loop l2;  select alldigit;  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 Print1;  delimiter $  CREATE PROCEDURE Print1(date1 date,date2 date)  BEGIN    declare count int;  set count := 0;    l1:loop  if date1 <= date2 THEN    if date\_format(date1,'%a') = 'Sun' THEN    set count := count +1;  set date1:=date1 + interval 1 day;  else  set date1:=date1 + interval 1 day;  end if;  else  leave l1;  end if;  end loop l1;  select count as "Count of Sundays";  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 Print1;  delimiter $  CREATE PROCEDURE Print1(date1 date,day\_name varchar(20))  BEGIN  declare count int;  set count := 0;    l1:loop  if date\_format(date1,'%a')!= 'Sat' THEN  set date1:=date1 + interval 1 day;  ELSE  select date1;  leave l1;  end if;  end loop l1;  END $  delimiter ; |
|  |