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 sepChar;  DELIMITER $  create Procedure sepChar(s varchar(70))  BEGIN  DECLARE ch varchar(1);  DECLARE count int;  set count:=1;  L1 : LOOP  if count <= length(s) THEN  set ch:= substring(s,count,1);  select ch;  set count:=count+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 sepChar;  DELIMITER $  create Procedure sepChar(s varchar(70))  BEGIN  DECLARE \_str varchar(70);  DECLARE ch varchar(1);  DECLARE count int;  set count:=1;  set \_str = "";  set \_str = concat(\_str,substring(s,count,1));  L1 : LOOP  if count < length(s) THEN  set count:=count+1;  set ch:= substring(s,count,1);  set \_str = concat(\_str,",",ch);  ELSE  leave L1;  end if;  end LOOP L1;  select \_str;  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 alphanum;  delimiter $  CREATE PROCEDURE alphanum(s varchar(50))  BEGIN  DECLARE ch varchar(1);  DECLARE count int;  DECLARE \_alpha varchar(50);  DECLARE \_num varchar(50);  set count :=1;  set \_alpha := "";  set \_num := "";    l1:LOOP  IF count <= length(s) THEN  set ch :=substring(s,count,1);  set count := count+1;  IF ASCII(ch) >= (48) AND ASCII(ch) <= (57) THEN  set \_num :=concat(\_num,ch);  else  set \_alpha := concat(\_alpha,ch);  end if;  ELSE  leave l1;  end if;  end LOOP l1;  select \_alpha,\_num;  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 |
|  |
|  |
| 1. Write a procedure to print all upper and lower characters separately.   Input: - AbCdEfG  Output: - ACEG  bdf |
| drop PROCEDURE IF EXISTS uplow;  delimiter $  create PROCEDURE uplow(s varchar(50))  BEGIN  DECLARE ch varchar(1);  DECLARE \_count int;  DECLARE \_upper varchar(30);  DECLARE \_lower varchar(30);  set \_count := 1;  set \_upper := "";  set \_lower := "";  l1:LOOP  IF \_count <= length(s) THEN  set ch :=substring(s,\_count,1);  IF ASCII(ch) >= (65) AND ASCII(ch) <=(90) THEN  set \_upper := concat(\_upper,ch);  else IF ASCII(ch) >= (97) AND ASCII(ch) <=(122) THEN  set \_lower := concat(\_lower,ch);  end if;  end if;  ELSE  leave l1;  end if;  set \_count := \_count+1;  end LOOP l1;  select \_upper,\_lower;  end $  delimiter ; |
|  |
| 1. Write a procedure to find the number of vowels, digits and white spaces |
| DROP PROCEDURE IF EXISTS vowelchar;  delimiter $  create PROCEDURE vowelchar(s varchar(40))  BEGIN  DECLARE count int;  DECLARE v\_count int;  DECLARE d\_count int;  DECLARE W\_count int;  DECLARE ch varchar(1);  set v\_count = 0;  set d\_count = 0;  set w\_count = 0;  set count := 1;  L1: LOOP  if count <= length(s) THEN  set ch = substring(s,count,1);  if ch = 'a' or ch = 'e' or ch = 'i' or ch = 'o' or ch = 'u' THEN  set v\_count := v\_count + 1;  else if ASCII(ch) >= (48) and ASCII(ch) <= (57) THEN  set d\_count := d\_count + 1;  else if ch = ' ' THEN  set w\_count := w\_count + 1;  end if;  end if;  end if;  ELSE  leave L1;  end if;  set count := count +1;  end LOOP L1;  select v\_count, d\_count, w\_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 alpha;  delimiter $  create PROCEDURE alpha(s varchar(50))  BEGIN  DECLARE ch varchar(1);  DECLARE \_count int;  DECLARE \_str varchar(30);  set \_count := 1;  set \_str := "";    l1:LOOP  IF \_count <= length(s) THEN  set ch :=substring(s,\_count,1);  IF ASCII(ch) >= (97) AND ASCII(ch) <=(122) THEN  set \_str := concat(\_str,ch);  end if;  ELSE  leave l1;  end if;  set \_count := \_count+1;  end LOOP l1;  select \_str;  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 |
| Sql :  create table P(id int,message varchar(50));  pl/sql :  drop procedure if exists insertrow;  delimiter $  create procedure insertrow()  BEGIN  DECLARE \_count int;  set \_count:=0;  L1:LOOP  set \_count := \_count + 1;  if (\_count <= 10) THEN  IF (\_count%2=0) THEN  insert into P values(\_count,"i is even");  else  insert into P values(\_count,"i is odd");  end if;  else  leave L1;  end if;  end LOOP L1;  end $  delimiter ; |
|  |
| 1. Write a procedure to print five highest paid employees from the emp table using cursor. |
| drop procedure if EXISTS findemp;  delimiter $  create PROCEDURE findemp()  BEGIN  DECLARE \_SAL, x int;  DECLARE \_ENAME varchar(30);  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. |
|  |
|  |
| 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) |
|  |
|  |
| 1. Write a procedure to accept customer number and display all his order. (Use customers and orders table) |
| drop procedure IF EXISTS order\_det;  delimiter $  create procedure order\_det(\_cnum int)  BEGIN  declare flag bool;  select TRUE into flag from customers where cnum=\_cnum;  IF flag THEN  select \* from customers c inner join orders o where c.cnum=o.cnum and c.cnum=\_cnum;  ELSE  select "Customer not found";  end if;  end $  delimiter ; |
|  |
| 1. Write a procedure to convert numbers into word   Input: - 45234  Output: - Four Five Two Three Four |
|  |
|  |
| 1. Write a procedure to find the sum of digits.   Input: - 5675  Output: - Twenty Three |
| drop PROCEDURE if EXISTS sumdigit;  delimiter $  create procedure sumdigit(num int)  BEGIN  DECLARE \_sum int;  DECLARE \_rem int;  set \_sum = 0;  set \_rem = 0;  L1 : LOOP  if num != 0 THEN  set \_rem := num % 10;  set \_sum := \_sum + \_rem;  set num := num / 10;  else  leave L1;  end if;  end loop l1;  select \_sum;  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 |
|  |
|  |
| 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 date\_weekday;  delimiter $  create procedure date\_weekday(\_date date, \_day varchar(10))  BEGIN  declare temp int ;  set temp := dayofweek(\_date);  if \_day = "Sunday" THEN  select Date\_add(\_date, INTERVAL (1-temp) DAY) Day\_Output;  else if \_day = "Monday" THEN  select Date\_add(\_date, INTERVAL (2-temp) DAY) Day\_Output;  else if \_day = "Tuesday" THEN  select Date\_add(\_date, INTERVAL (3-temp) DAY) Day\_Output;  else if \_day = "Wednesday" THEN  select Date\_add(\_date, INTERVAL (4-temp) DAY) Day\_Output;  else if \_day = "Thursday" THEN  select Date\_add(\_date, INTERVAL (5-temp) DAY) Day\_Output;  else if \_day = "Friday" THEN  select Date\_add(\_date, INTERVAL (6-temp) DAY) Day\_Output;  else if \_day = "Saturday" THEN  select Date\_add(\_date, INTERVAL (7-temp) DAY) Day\_Output;  end if;  end if;  end if;  end if;  end if;  end if;  end if;  end $  delimiter ; |
|  |