

Exercices – Series 3 (Procedures and functions)

Introduction

Here, we'll take time to discover how to use procedures and functions, how they are used and called, either in PL/SQL programs or in SQL*Plus prompt.

0 – Easy exercises first (as per usual...)

Exercise 0.1

Write a procedure that counts from 10 to zero.

Exercise 0.2

Write a procedure that takes three number parameters : the two first ones are read by the procedure, then the third stores their minimum. Write a program that uses this procedure to display the minimum of 2 numbers (for example, 45 and 23).

Your code should produce the following result :

Minimum of (23, 45) : 23

PL/SQL procedure successfully completed.

Exercise 0.3

Write a procedure that computes the square of a passed value. Write a program that uses this function.

Your code should produce the following result :

Square of (23): 529

PL/SQL procedure successfully completed.

Exercise 0.4

Write a standalone function that returns the total number of customers in the CUSTOMERS table (cf. Series 1, exercise 0.4). Write a program that uses this function.

Your code should produce the following result :

Total no. of Customers: 6

PL/SQL procedure successfully completed.

Exercise 0.5

Write a function that computes and returns the maximum of two values. Write a program that calls this function.

Your code should produce the following result :

Maximum of (23,45): 78

PL/SQL procedure successfully completed.

Exercise 0.6

Write a recursive function that calculates the factorial of a given number. Write a program that calls this function.

Your code should produce the following result :

Factorial 6 is 720

PL/SQL procedure successfully completed.

Exercise 0.7

Write a function that calculates b^n , with n superior or equal to zero.

1 – Let's mix SQL with PL/SQL

Here, we'll use again the table defined in annex A.1 of Series 2 (genealogic tree).

Exercise 1.1

Write a function that takes two people numbers as parameters, and returns true if and only if these two people have a common parent.

Exercise 1.2

Write a function that returns true if and only if two people (identified by their numbers) are first cousins.

Exercise 1.3

Write a function that displays the name of one person (identified by his/her number) as well as the name of all his/her male ancestors (father, grandfather, great-grandfather...).

Exercise 1.4

Write a function that displays the name of all the male members from the family of one person (identified by his/her number).

Exercise 1.5

Write a function that takes two family members A and B (identified by their numbers) and returns true if and only if B is a descendant of A.

Exercise 1.5

Write a function that takes two family members A and B (identified by their numbers) and in case B is a descendant of A or the contrary, returns the number of generations separating them (-1 otherwise)..

Annex A.1 – The PERSON table

```
CREATE TABLE Person(  
    numpers NUMBER PRIMARY KEY,  
    name VARCHAR2(30) NOT NULL,  
    firstname VARCHAR2(30),  
    father REFERENCES Person(numpers),  
    mother REFERENCES Person(numpers)  
);  
  
INSERT INTO Person VALUES (1, 'de Montmirail, dit le Hardi', 'Godefroy', NULL, NULL);  
INSERT INTO Person VALUES (16, 'ET', NULL, NULL, NULL);  
INSERT INTO Person VALUES (2, 'Nelson', 'Prince Rogers', 1, 16);  
INSERT INTO Person VALUES (3, 'Bowie', 'David', NULL, NULL);  
INSERT INTO Person VALUES (4, 'Mercury', 'Freddie', NULL, NULL);  
INSERT INTO Person VALUES (13, 'Gore', 'Tipper', NULL, NULL);  
INSERT INTO Person VALUES (5, 'George', 'Bart', 3, 4);  
INSERT INTO Person VALUES (12, 'Cersei', 'Jamie', NULL, NULL);  
INSERT INTO Person VALUES (6, 'Quatro', 'Suzi', NULL, NULL);  
INSERT INTO Person VALUES (7, 'May', 'Brian', 4, 6);  
INSERT INTO Person VALUES (8, 'Dijkstra', 'Edger', 2, 13);  
INSERT INTO Person VALUES (9, 'Stark', 'Eddard', 4, 13);  
INSERT INTO Person VALUES (10, 'Bach', 'Johann Sebastian', 9, 6);  
INSERT INTO Person VALUES (17, 'Lannister', 'Cersei', NULL, NULL);  
INSERT INTO Person VALUES (11, 'Lannister', 'Joffrey', 12, 17);  
INSERT INTO Person VALUES (15, 'Socrates', NULL, NULL, NULL);  
INSERT INTO Person VALUES (19, 'Stark', 'Catelyn', NULL, NULL);  
INSERT INTO Person VALUES (20, 'Stark', 'Robb', 9, 19);  
INSERT INTO Person VALUES (21, 'Lannister', 'Tyrion', 9, 17);  
INSERT INTO Person VALUES (14, 'Snow', 'John', 9, 6);  
INSERT INTO Person VALUES (18, 'Deacon', 'John', 7, 13);
```