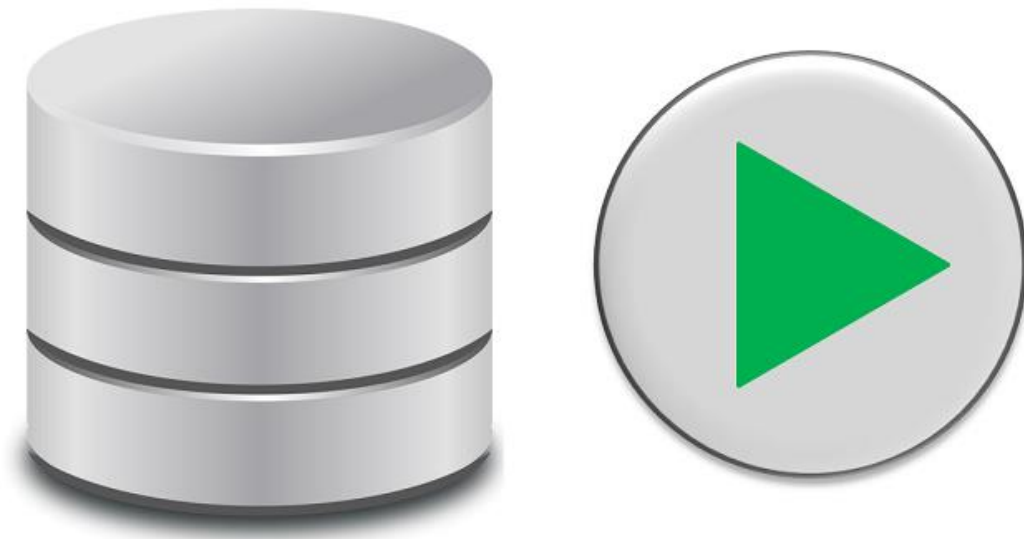




ORACLE PL/SQL



MÓDULO 09

FUNCIONES Y PROCEDIMIENTOS

GUSTAVO CORONEL
desarrollasoftware.com



ORACLE PL/SQL



Contenido

FUNCIONES	3
PROCEDIMIENTOS	5
PARÁMETROS	6
Uso de NOCOPY	8
CURSOS VIRTUALES	10
JAVA ORIENTADO A OBJETOS	10
PROGRAMACIÓN DE BASE DE DATOS CON JAVA JDBC	10



ORACLE PL/SQL



FUNCIONES

Sintaxis

```
CREATE OR REPLACE FUNCTION nombre_función (  
    Arg1 [ IN | OUT | IN OUT ] tipo,  
    Arg2 [ IN | OUT | IN OUT ] tipo,  
    . . . ) RETURN tipo  
AS  
  
    Declaraciones_Locales  
  
BEGIN  
  
    Cuerpo_Función  
  
EXCEPTION  
  
    Tratamiento_Excepciones  
  
END nombre_procedimiento;
```

Script 1

```
create or replace function cantemp( p_deptno dept.deptno%type) return number  
as  
    v_cont number;  
begin  
    select count(*) into v_cont from emp where deptno = p_deptno;  
    return v_cont;  
end cantemp;
```

Ejecución 1

```
SQL> declare  
2     v_cant number;  
3     begin  
4         v_cant := cantemp( 10 );  
5         dbms_output.put_line( v_cant || ' empleados' );  
6     end;  
7     /  
3 empleados  
  
PL/SQL procedure successfully completed.
```



ORACLE PL/SQL



Ejecución 2

```
declare
  cursor c_dept is select deptno from dept;
  v_cant number;
begin
  for v_dept in c_dept loop
    v_cant := cantemp(v_dept.deptno);
    dbms_output.put_line( 'dept=' || v_dept.deptno || ' empleados: ' || v_cant );
  end loop;
end;
```



ORACLE PL/SQL



PROCEDIMIENTOS

Sintaxis

```
CREATE OR REPLACE PROCEDURE nombre_procedimiento (  
    Arg1 [ IN | OUT | IN OUT ] tipo,  
    Arg2 [ IN | OUT | IN OUT ] tipo,  
    . . . )  
AS  
  
    Declaraciones_Locales  
  
BEGIN  
  
    Cuerpo_Procedimiento  
  
EXCEPTION  
  
    Tratamiento_Excepciones  
  
END nombre_procedimiento;
```

Script 2

```
create or replace procedure adddept(  
    p_deptno dept.deptno%type,  
    p_dname dept.dname%type,  
    p_loc dept.loc%type )  
as  
begin  
    insert into dept(deptno, dname, loc) values(p_deptno, p_dname, p_loc);  
end adddept;
```

Ejecución

```
SQL> exec adddept( 15, 'demo', 'lima');  
  
PL/SQL procedure successfully completed.
```



ORACLE PL/SQL



PARÁMETROS

MODO	DESCRIPCIÓN
IN	<p>El valor del parámetro real se pasa al procedimiento cuando se produce la llamada al mismo.</p> <p>Dentro del procedimiento, el parámetro formal se comporta como una constante PL/SQL, se considera de solo lectura y no puede ser modificado.</p> <p>Cuando el procedimiento finaliza y devuelve el control al entorno desde donde se produjo la llamada, el parámetro real no se modifica.</p>
OUT	<p>Se ignora cual valor que el parámetro real pueda tener cuando se produce la llamada al procedimiento.</p> <p>Dentro del procedimiento, el parámetro formal se comporta como una variable sin inicializar, por lo que su valor inicial es NULL.</p> <p>Puede leerse y escribir en dicha variable. Cuando el procedimiento finaliza y devuelve el control al entorno desde donde se produjo la llamada, se asigna el valor del parámetro formal al parámetro real.</p>
IN OUT	Este modo es una combinación de los modos IN y OUT.

Script 3

```
Create or Replace Procedure prTestOUT1
( p_Raise IN Boolean, p_Dato Out Varchar2 )
Is
    Excep1 Exception;
Begin
    p_Dato := 'Alianza Campeon';
    If p_Raise Then
        Raise Excep1;
    Else
        Return;
    End If;
End;
```



ORACLE PL/SQL



Ejecución 1

```
SQL> Declare
  2  Rpta Varchar2(20) := 'Shakira';
  3  Begin
  4  DBMS_Output.Put_Line( 'Valor Inicial: ' || Rpta );
  5  prTestOUT1( False, Rpta );
  6  DBMS_Output.Put_Line( 'Valor Final: ' || Rpta );
  7  End;
  8  /
Valor Inicial: Shakira
Valor Final: Alianza Campeon

PL/SQL procedure successfully completed.
```

Ejecución 2

```
SQL> Declare
  2  Rpta Varchar2(20) := 'Shakira';
  3  Begin
  4  DBMS_Output.Put_Line( 'Valor Inicial: ' || Rpta );
  5  prTestOUT1( True, Rpta );
  6  DBMS_Output.Put_Line( 'Valor Final: ' || Rpta );
  7  Exception
  8  When Others Then
  9  DBMS_Output.Put_Line( 'Valor Después del error: ' || Rpta );
 10 End;
 11 /
Valor Inicial: Shakira
Valor Después del error: Shakira

PL/SQL procedure successfully completed.
```



ORACLE PL/SQL



Uso de NOCOPY

Script 4

```
Create or Replace Procedure prTestOUT2
( p_Raise IN Boolean, p_Dato Out NOCOPY Varchar2 )
Is
    Excep1 Exception;
Begin
    p_Dato := 'Alianza Campeon';
    If p_Raise Then
        Raise Excep1;
    Else
        Return;
    End If;
End;
```

Ejecución 1

```
SQL> Declare
2   Rpta Varchar2(20) := 'Shakira';
3   Begin
4   DBMS_Output.Put_Line( 'Valor Inicial: ' || Rpta );
5   prTestOUT2( False, Rpta );
6   DBMS_Output.Put_Line( 'Valor Final: ' || Rpta );
7   End;
8   /
Valor Inicial: Shakira
Valor Final: Alianza Campeon

PL/SQL procedure successfully completed.
```




ORACLE PL/SQL



Ejecución 2

```
SQL> Declare
  2  Rpta Varchar2(20) := 'Shakira';
  3  Begin
  4  DBMS_Output.Put_Line( 'Valor Inicial: ' || Rpta );
  5  prTestOUT2( True, Rpta );
  6  DBMS_Output.Put_Line( 'Valor Final: ' || Rpta );
  7  Exception
  8  When Others Then
  9  DBMS_Output.Put_Line( 'Valor Después del error: ' || Rpta );
 10 End;
 11 /
Valor Inicial: Shakira
Valor Después del error: Alianza Campeon

PL/SQL procedure successfully completed.
```



ORACLE PL/SQL



CURSOS VIRTUALES

En esta URL tienes cupones de descuento:

<http://gcoronelc.blogspot.com/p/cursos-virtuales.html>

Java Orientado a Objetos



JAVA **ORIENTADO A OBJETOS**

Eric Gustavo Coronel Castillo
www.desarrollasoftware.com
INSTRUCTOR

<https://www.udemy.com/java-orientado-a-objetos/>

Programación de Base de Datos con Java JDBC



PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JAVA JDBC

Eric Gustavo Coronel Castillo
www.desarrollasoftware.com
INSTRUCTOR

[https://www.udemy.com/java-jdbc-oracle /](https://www.udemy.com/java-jdbc-oracle/)