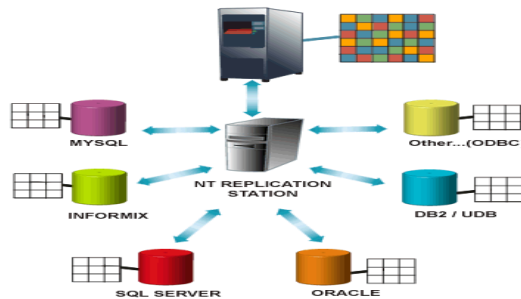


UNIVERSIDAD TECNOLÓGICA DE PANAMA  
FACULTAD DE INGENIERIA DE SISTEMAS  
LICENCIATURA EN INGENIERIA EN SISTEMAS DE INFORMACION.

## SISTEMAS DE BASE DE DATOS II ORACLE PROGRAMACION PL/SQL

Procedimiento -PL/SQL ORACLE



Por. Ing. Henry Lezcano Sistemas  
de Base de Datos II II Semestre del  
2022

{ 1 }

1

## CONTENIDO

### Capítulo IV. Procedimientos

- **Fundamentos de Lenguaje PL/SQL**
- **Cursores**
- **Procedimientos**



Por. Ing. Henry Lezcano Sistemas  
de Base de Datos II II Semestre del  
2022

{ 2 }

2

### 4.3 PROCEDIMIENTOS



- Los procedimientos y funciones de PL/SQL se comportan de manera similar a los procedimiento y funciones de otros lenguajes, comparten muchas de sus propiedades.
- También son conocidos como subprogramas.

```
CREATE OR REPLACE PROCEDURE AddNewStudent (  
  p_StudentID      students.id%TYPE,  
  p_FirstName      students.first_name%TYPE,  
  p_LastName       students.last_name%TYPE  
  p_Major          students.major%TYPE ) AS  
BEGIN  
  -- Inserta una nueva fila en la tabla students. Usa  
  -- student_sequence para generar el nuevo ID del estudiante y  
  -- asigna el valor 0 a current_credits.  
  
  INSERT INTO students (ID, first_name, last_name , major, current_credits)  
    VALUES ( student_sequence.next, p_FirstName, p_LastName, p_Major, 0 );  
  
  COMMIT;  
END AddNewStudent;
```

3

### 4.3 PROCEDIMIENTOS



Una vez creado el procedimiento, puede ser invocado desde otro bloque PL/SQL, por ejemplo

```
BEGIN  
  AddNewStudent (50, 'David', 'Dinsmore', 'Music');  
END;
```

Este ejemplo ilustra varios puntos de importancia:

- En primer lugar se crea el procedimiento **AddNewStudent**, con la orden **CREATE OR REPLACE PROCEDURE**. Al crear el procedimiento, este se compila y se almacena en la Base de Datos en forma compilada.
- Cuando es invocado, puede pasársele parámetros. Para el ejemplo, se pasa en tiempo de ejecución el nombre, apellido y especialidad del nuevo estudiante. Dentro del procedimiento el parámetro **p\_FirstName** tendrá el valor de "David", **p\_LastName** tendrá el valor de 'Dinsmore' y **p\_Major**, tendrá el valor de 'Music', ya que se pasan estas literales al procedimiento en el momento de invocarlo.
- Una llamada a un procedimiento es una orden PL/SQL por si misma. La llamada no se produce como parte de la expresión. Cuando se llama a un procedimiento, el control pasa a la primera orden ejecutable dentro de el. Cuando procedimiento termina, se devuelve el control a la orden que sigue a la llamada al procedimiento.
- Un procedimiento es un bloque PL/SQL, con una sección declarativa ejecutable y una sección de manejo de excepciones. Al igual que los bloques anónimos, la única sección obligatoria es la sección ejecutable. **AddNewStudent** solo consta de una sección ejecutable.

```
CREATE OR REPLACE PROCEDURE AddNewStudent (  
  p_StudentID      students.id%TYPE,  
  p_FirstName      students.first_name%TYPE,  
  p_LastName       students.last_name%TYPE  
  p_Major          students.major%TYPE ) AS  
BEGIN  
  -- Inserta una nueva fila en la tabla students. Usa  
  -- student_sequence para generar el nuevo ID del estudiante y  
  -- asigna el valor 0 a current_credits.  
  
  INSERT INTO students (ID, first_name, last_name , major, current_credits)  
    VALUES ( student_sequence.next, p_FirstName, p_LastName, p_Major, 0 );  
  
  COMMIT;  
END AddNewStudent;
```

4

## 4.3 PROCEDIMIENTOS

### Creación de un Procedimiento



La sintaxis para la orden CREATE OR REPLACE PROCEDURE es

```
CREATE [OR REPLACE] PROCEDURE nombre_procedimiento (
  [ ( argumento [{IN | OUT | IN OUT}] tipo,
    ....
    argumento [{IN | OUT | IN OUT}] tipo ) ] IS | AS
  cuerpo_procedimiento
```

Donde **nombre\_procedimiento** es el nombre del procedimiento que se quiere crear, **argumento** es el nombre de un parámetro, tipo es el tipo del parámetro asociado y el **cuerpo\_procedimiento** es un bloque PL/SQL que contiene el código del procedimiento.

- Para poder cambiar el código de un procedimiento, es necesario eliminarlo y volverlo a crear.
- La palabra clave OR REPLACE permite hacer esto, en una única operación.
- Si queremos eliminar un procedimiento usamos la orden DROP PROCEDURE **nombre\_procedimiento**.
- Si el procedimiento existe y no se le incluyo la palabra clave OR REPLACE la orden CREATE mandara un error de Oracle. Ora-00955 name is already used by an existing object.
- Al igual que sucede con la orden CREATE, la creación de un procedimiento es una orden DDL, así que el COMMIT es implícito. Se puede usar IS o AS ambas son equivalentes.

Por. Ing. Henry Lezcano Sistemas  
de Base de Datos II II Semestre del  
2022

{ 5 }

5

## 4.3 PROCEDIMIENTOS

### Parámetros y Modos



Dado el procedimiento AddNewStudent , podemos invocarlo desde el siguiente bloque PL/SQL anónimo:

```
DECLARE
  -- Variables que describen al nuevo estudiante
  v_NewFirstName      students.first_name%TYPE := 'Margaret';
  v_NewLastName       students.last_name%TYPE  := 'Mason';
  v_NewMajor          students.major%TYPE      := 'History';
BEGIN
  -- Añade Margaret Mason a la Base de Datos
  AddNewStudent(v_NewFirstName, v_NewLastName, v_NewMajor);
END;
```

- Las variables declaradas en el bloque precedente (v\_NewFirstName, v\_NewLastName, v\_NewMajor) se pasan como argumentos a **AddNewStudent** .
- Bajo esta perspectiva dichas variables reciben el nombre de **parametros reales**, mientras que los parametros en la declaracion del procedimiento (p\_FirstName, p\_LastName, p\_Major) se denominan **parametros formales**.

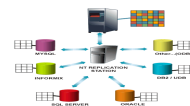
Por. Ing. Henry Lezcano Sistemas  
de Base de Datos II II Semestre del  
2022

{ 6 }

6

# 4.3 PROCEDIMIENTOS

## Parámetros y Modos



- Los parámetros reales contienen los valores que se pasan al procedimiento cuando este es invocado y reciben el resultado del procedimiento cuando este termina.
- Los valores de los parámetros reales son los que usan dentro del procedimiento.
- Los valores de los parámetros formales son meros contenedores para los valores de los parámetros reales.
- Cuando se llama al procedimiento se asigna el valor de los parámetros reales a los parámetros formales.
- Dentro del procedimiento, se hace referencia a dichos valores mediante los parámetros formales.
- Cuando el procedimiento termina, se asigna el valor de los parámetros formales a los parámetros reales siguiendo las reglas de asignación y conversión del PL/SQL.
- Los parámetros formales pueden tener 3 modos: IN, OUT o IN OUT .
- Sino se especifica el modo de un parámetro formal se adopta por defecto IN.

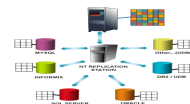
Por Ing. Henry Lezcano Sistemas de Base de Datos II II Semestre del 2022

{ 7 }

7

# 4.3 PROCEDIMIENTOS

## Parámetros y Modos



Modo	Descripcion
IN	El valor del parámetro real se pasa al procedimiento cuando este es invocado. Dentro del procedimiento, el parámetro formal se considera como de <i>sólo lectura</i> y no puede ser cambiado. Cuando se termina el procedimiento, y se devuelve el control al entorno que realizó la invocación, el parámetro real no sufre cambio.
OUT	Se ignora cualquier valor que tenga el parámetro real cuando se llama el procedimiento. Dentro del procedimiento, el parámetro formal se considera como de <i>sólo escritura</i> no puede ser leído, sino que tan solo puede asignársele valores. Cuando termina el procedimiento y se devuelve al entorno que realizó la llamada, los contenidos del parámetro formal se asignará al parámetro real.
IN OUT	Este modo es una combinación de IN y OUT. El valor del parámetro real se pasa al procedimiento cuando este es invocado. Dentro del procedimiento, el parámetro formal puede ser tanto <i>leído como escrito</i> . Cuando termina el procedimiento y se devuelve el control al entorno que realizó la llamada, los contenidos del parámetro forma se asignan al parámetro real.

Por Ing. Henry Lezcano Sistemas de Base de Datos II II Semestre del 2022

{ 8 }

8

4.3 PROCEDIMIENTOS

Parámetros y Modos



Cuales de las asignaciones de los variables es correcta?

```
CREATE OR REPLACE PROCEDURE ModeTest (  
  p_InParameter IN NUMBER,  
  p_OutParameter OUT NUMBER,  
  p_InOutParamter IN OUT NUMBER) IS  
  v_LocalVariable Number ;  
BEGIN  
  v_LocalVariable := p_InParameter;  
  p_InParameter := 7;  
  p_OutParameter := 7;  
  v_LocalVariable := p_OutParameter;  
  v_LocalVariable := p_InOutParameter;  
  p_InOutParameter := 7;  
END ModeTest;
```

Por Ing. Henry Lezcano Sistemas de Base de Datos II II Semestre del 2022

4.3 PROCEDIMIENTOS

Parámetros y Modos



Cual de la siguientes invocaciones es correcta?

```
CREATE OR REPLACE PROCEDURE ModeTest (  
  p_InParameter IN NUMBER,  
  p_OutParameter OUT NUMBER,  
  p_InOutParamter IN OUT NUMBER) IS  
  v_LocalVariable Number ;  
BEGIN  
  v_LocalVariable := p_InParameter;  
  p_InParameter := 7;  
  p_OutParameter := 7;  
  v_LocalVariable := p_OutParameter;  
  v_LocalVariable := p_InOutParameter;  
  p_InOutParameter := 7;  
END ModeTest;
```

```
-- Invocación No.1  
DECLARE  
  v_Variable1 NUMBER,  
  v_Variable2 NUMBER;  
BEGIN  
  ModeTest(12, v_Variable1, v_variable2);  
END;
```

```
Invocación No. 2  
DECLARE  
  v_Variable1 NUMBER,  
  v_Variable2 NUMBER;  
BEGIN  
  ModeTest(12, v_Variable1, 11);  
END;
```

Por Ing. Henry Lezcano Sistemas de Base de Datos II II Semestre del 2022

## 4.3 PROCEDIMIENTOS

### El Cuerpo del Procedimiento



- El cuerpo del procedimiento es un bloque PL/SQL, con sus acciones declarativa, ejecutable y de manejo de excepciones.
- La sección declarativa se sitúa entre la palabra clave IS o As y la palabra BEGIN.
- La ejecutable (la única obligatoria) esta comprendida entre las palabras clave BEGIN y EXCEPTION.
- La sección de excepciones, por su parte esta delimitada por las palabras clave EXCEPTION y END.
- La estructura de un procedimiento tendra la forma siguiente:

```
CREATE OR REPLACE PROCEDURE nombre_procedimiento AS
    /* Section declarativa */
BEGIN
    /* Seccion Ejecutable */
EXCEPTION
    /* Seccion de Excepciones */
END [nombre_procedimiento];
```

Por. Ing. Henry Lezcano Sistemas  
de Base de Datos II II Semestre del  
2022

[ 11 ]

11

## 4.3 PROCEDIMIENTOS

### Restricciones sobre los Parametros Formales



- En una declaración de procedimiento, es **illegal** restringir un parámetro **CHAR o VARCHAR2** con determinada longitud, o un parámetro NUMBER con un valor de precisión y/o escala
- La declaración del procedimiento es ilegal:

```
CREATE OR REPLACE PROCEDURE ParameterLength (
    p_Parameter1 IN OUT VARCHAR2(10),
    p_Parameter2 IN OUT NUMBER(3, 2) ) AS
BEGIN
    p_Parameter1 := 'abcdefghijklm';
    p_Parameter2 := 12.3;
END ParameterLength;
```

- La declaración del procedimiento es legal:

```
CREATE OR REPLACE PROCEDURE ParameterLength (
    p_Parameter1 IN OUT VARCHAR2,
    p_Parameter2 IN OUT NUMBER) AS
BEGIN
    p_Parameter1 := 'abcdefghijklm';
    p_Parameter2 := 12.3;
END ParameterLength;
```

Por. Ing. Henry Lezcano Sistemas  
de Base de Datos II II Semestre del  
2022

[ 12 ]

12

### 4.3 PROCEDIMIENTOS

#### Restricciones sobre los Parametros Formales



- En una declaración de procedimiento, es **illegal** restringir un parámetro **CHAR o VARCHAR2** con determinada longitud, o un parámetro **NUMBER** con un valor de precisión y/o escala
- La declaración del procedimiento es legal:

```
CREATE OR REPLACE PROCEDURE ParameterLength (  
    p_Parameter1 IN OUT VARCHAR2,  
    p_Parameter2 IN OUT NUMBER) AS  
BEGIN  
    p_Parameter1 := 'abcdefghijklm;  
    p_Parameter2 := 12.3;  
END ParameterLength;
```

-- Invocación No.1

```
DECLARE  
    v_Variable1 VARCHAR2(40);  
    v_Variable2 NUMBER(3,4);  
BEGIN  
    ParameterLength(v_Variable1, v_variable2);  
END;
```

-- Invocación No.2

```
DECLARE  
    v_Variable1 VARCHAR2(10);  
    v_Variable2 NUMBER(3,4);  
BEGIN  
    ParameterLength(v_Variable1, v_variable2);  
END;
```

*Cual seria la situación presentada aqui?*

Por. Ing. Henry Lezcano Sistemas  
de Base de Datos II II Semestre del  
2022

[ 13 ]

13

### 4.3 PROCEDIMIENTOS

#### Valores predeterminados de los parametros



- Al igual que con las declaraciones de las variables, los parámetros formales de un procedimiento o función pueden tener valores predeterminados.
- Si un parámetro tiene un valor predeterminado, no tiene por que ser pasado desde el entorno que realizo la llamada. Si es pasado, se usara el valor real en lugar del valor predeterminado.
- La sintaxis es:

*nombre\_parámetro [modo] tipo\_parámetro { := | DEFAULT } valor\_inicial*

Ejemplo:

```
CREATE OR REPLACE PROCEDURE AddNewStudent (  
    p_FirstName      students.first_name%TYPE ,  
    p_LastName       students.last_name%TYPE ,  
    p_Major          students.major%TYPE DEFAULT 'Economic') AS  
BEGIN  
    -- Inserta una nueva fila en la tabla students. Usa student_sequence  
    -- para generar el nuevo valor ID del estudiante y asigna el valor 0  
    -- a current_credits  
    INSERT INTO students VALUES (student_sequence.nextval, p_FirstName, p_LastName, 0);  
END AddNewStudent ;
```

-- Invocación No.1 – Notacion Posicional

```
BEGIN  
    AddNewStudent('Barbara', 'Blues');  
END;
```

-- Invocación No.2 – Con Notacion nominal

```
BEGIN  
    AddNewStudent( p_Firstname => 'Barbara',  
                   p_LastName => 'Blues');  
END;
```

Por. Ing. Henry Lezcano Sistemas  
de Base de Datos II II Semestre del  
2022

[ 14 ]

14