PL/SQL Procedures

Ricardo Abarca t024672
Brianna Fernandez
Paulina De la Garza t023527

Subprogramas

Un subprograma es una unidad/módulo que realiza una tarea. Pueden ser invocados por otro subprograma o un programa que se le denota como 'Calling Program'. Un subprograma puede ser creado:

- A nivel de 'schema'
- Dentro de un paquete
- Dentro de un bloque PL/SQL

Un subprograma PL/SQL se le conoce como bloque PL/SQL y pueden ser invocados con ciertos parámetros. PL/SQL proporciona dos tipos de subprogramas:

- **Functions**: Regresan un solo valor. Se utilizan principalmente para calcular y regresar un valor
- **Procedures:** No regresan un valor directamente, se utilizan para realizar acciones

¿Qué es entonces un ‘Procedure’?

Al igual que en otros tipos de lenguajes, se pueden crear sus propios “Procedures”. Estos son un grupo de declaraciones de PL/SQL que pueden ser llamados por nombre. Se pueden crear, borrar o reemplazar.

- Crear : `CREATE PROCEDURE`
- Eliminar: `DROP PROCEDURE`
- Reemplazar: `REPLACE PROCEDURE`

Estos pueden:

- Aceptar varios parámetros de entrada y obtener como respuesta múltiples valores en forma de parámetros
- Pueden llamar a otros procedimientos

¿Para qué se utilizan?

La ventaja de utilizar los `‘procedures’` o procedimientos son varios:

- Permiten la programación modular
- Permiten una ejecución más rápida
- Pueden reducir el tráfico de red
- Pueden ser utilizados como mecanismos de seguridad (definer’s rights & invoker’s rights)

Sintaxis

Header: sección para definición del procedure

```
CREATE [OR REPLACE] PROCEDURE procedure_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
{IS | AS}
BEGIN
  < procedure_body >
END procedure_name;
```

Lista de parametros

Body: sección para hacer las declaraciones

Declaraciones ejecutables

Sintaxis

- La opción `[OR REPLACE]` permite la modificación de un procedure existente.
- La lista de parámetros contiene el nombre, modo y tipo de los parámetros.
- `IN` representa el valor que será pasado desde fuera.
- `OUT` representa el parámetro que regresará un valor fuera del procedure.
- El keyword `AS` puede ser usado en lugar de `IS` para crear un procedure único.
- Para ejecutarlo se utiliza el comando `EXECUTE proc_name`

Parámetros de entrada/salida

1	`IN` Es un parámetro que te permite pasar un valor al subprograma. Es un parámetro únicamente de lectura. Los parámetros son pasados por medio de referencias.
2	`OUT` Es un parámetro que regresa un valor al programa que está llamando. Dentro del subprograma, este parámetro actúa como una variable. El parámetro actual debe ser una variable y esta es pasada por medio de un valor.
3	`IN OUT` Este parámetro inicialmente pasa un valor al subprograma y posteriormente regresa un valor actualizado al que lo llamó. El parámetro actual debe ser una variable y se le debe asignar un valor. Este es pasado por medio de un valor.

Ejemplo

```
PROCEDURE secure_dml
IS
BEGIN
     IF TO_CHAR (SYSDATE, 'HH24:MI') NOT BETWEEN '08:00' AND '18:00'
          OR TO_CHAR (SYSDATE, 'DY') IN ('SAT', 'SUN') THEN
     RAISE_APPLICATION_ERROR (-20205,
               'You may only make changes during normal office hours');
     END IF;
END secure_dml;
```

Referencias

https://www.tutorialspoint.com/plsql/plsql_procedures.htm

https://www.youtube.com/watch?v=buaSuEMi4lw