

Módulo: Bases de Datos

Unidad 6: Programación de Bases de Datos

Sesión 4: Procedimientos en PL/SQL

Descripción:

Durante esta unidad se ha estudiado como implementar procedimientos como bloques PL/SQL que realizan una tarea concreta u acción sobre los datos almacenados en las tablas de un esquema.

Para practicar todos estos conceptos se propone la realización de los siguientes casos prácticos.

Lista de operaciones:

- Cree un procedimiento denominado ADD_JOB para insertar un nuevo trabajo en la tabla JOBS. Se deben proporcionar el identificador del trabajo (JOB_ID) y el título del trabajo (JOB_TITLE) por medio de parámetros. Pruebe el procedimiento implementado anteriormente mediante la sentencia EXECUTE y mediante un bloque anónimo
- Cree un procedimiento denominado UP_JOB para actualizar el título asignado a un trabajo (JOB_TITLE) a partir de su identificador de trabajo (JOB_ID), esta información debe ser enviada a través de parámetros. Una vez compilado el procedimiento UP_JOB, pruebe dicho procedimiento a través del paso de parámetro por nombre.
- Cree un procedimiento denominado info_emp que obtenga el nombre y el salario de un empleado a través de su id. Cree tantos parámetros como crea necesario y establezca si son parámetros de tipo IN o de tipo OUT en función de sus necesidades.

Criterios de Evaluación:

- RA05_a: Se han identificado las diversas formas de automatizar tareas.
- RA05_b: Se han reconocido los métodos de ejecución de guiones.
- RA05_c: Se han identificado las herramientas disponibles para editar guiones.
- RA05_d: Se han definido y utilizado guiones para automatizar tareas.
- RA05_f: Se han definido procedimientos y funciones de usuario.

Objetivos:

- Identificar las diversas formas de automatizar tareas.
- Utilizar estructuras de control de flujo.
- Hacer uso de las funciones proporcionadas por el sistema gestor.
- Definir procedimientos.

Recursos:

- Acceso a Internet.
- Oracle Express.
- SQL Developer

Conceptos a revisar previamente:

- Realizar el estudio de los apartados de la unidad:
 - Procedimientos almacenados. Funciones de usuario.
- Realizar Ejercicios Resueltos:
 - Procedimientos

Resolución de la práctica:

- Cree un procedimiento denominado ADD_JOB para insertar un nuevo trabajo en la tabla JOBS. Se deben proporcionar el identificador del trabajo (JOB_ID) y el título del trabajo (JOB_TITLE) por medio de parámetros.

```
CREATE OR REPLACE PROCEDURE ADD_JOB (
  j_id IN jobs.job_id%TYPE,
  j_tit IN jobs.job_title%TYPE
) IS
BEGIN
  INSERT INTO jobs (job_id,job_title) VALUES (j_id,j_tit);
  DBMS_OUTPUT.PUT_LINE('Se ha insertado el nuevo trabajo cuyo id es ' ||
j_id
  || ' y su nombre es: ' || j_tit);
END ADD_JOB;
```

Pruebe el procedimiento implementado anteriormente mediante la sentencia EXECUTE y mediante un bloque anónimo

```
EXECUTE ADD_JOB('Prof','Profesor');

DECLARE
  v_job_id jobs.job_id%TYPE := 'Alm';
  v_job_title jobs.job_title%TYPE := 'Alumno';
BEGIN
  ADD_JOB(v_job_id,v_job_title);
END;

EXECUTE ADD_JOB('ST_MAN','Stock Manager');
```

- Cree un procedimiento denominado UP_JOB para actualizar el título asignado a un trabajo (JOB_TITLE) a partir de su identificador de trabajo (JOB_ID), esta información debe ser enviada a través de parámetros. Una vez compilado el procedimiento UP_JOB, pruebe dicho procedimiento a través del paso de parámetro por nombre.

```
CREATE OR REPLACE PROCEDURE UP_JOB (
  j_id IN jobs.job_id%TYPE,
  j_tit IN jobs.job_title%TYPE
) IS
BEGIN
  UPDATE jobs SET job_title=j_tit WHERE job_id=j_id;
  DBMS_OUTPUT.PUT_LINE('Se ha actualizado trabajo cuyo id es ' || j_id
  || ' y su nuevo nombre es: ' || j_tit);
END UP_JOB;
```

```
EXECUTE up_job ('Prof', 'Profesor 2')
```

```
EXECUTE up_job(j_tit=>'Alumnos', j_id=>'Alm')
```

- Cree un procedimiento denominado info_emp que obtenga el nombre y el salario de un empleado a través de su id. Cree tantos parámetros como crea necesario y establezca si son parámetros de tipo IN o de tipo OUT en función de sus necesidades

```
CREATE OR REPLACE PROCEDURE info_emp (  
e_id IN EMPLOYEES.EMPLOYEE_ID%TYPE,  
nombre OUT EMPLOYEES.FIRST_NAME%TYPE,  
salario OUT EMPLOYEES.SALARY%TYPE  
) is  
BEGIN  
  
SELECT first_name, salary INTO nombre, salario FROM employees WHERE  
EMPLOYEE_ID=e_id;  
  
END info_emp;
```

```
SET SERVEROUTPUT ON;  
--Bloque anonimo  
DECLARE  
nom EMPLOYEES.FIRST_NAME%TYPE;  
sal EMPLOYEES.SALARY%TYPE;  
BEGIN  
info_emp(100,nom,sal);  
SYS.DBMS_OUTPUT.PUT_LINE(nom || ' gana ' || sal || ' $ ' );  
END;
```