

| GUÍA 1.2.2:

Cursores Explícitos Complejos en PL/SQL

Sigla	Asignatura	Experiencia de Aprendizaje
BDY1103	Taller de Base de Datos	EA1: Desarrolla bloques PL/SQL para procesar datos y generar información relevante para el negocio.
Tiempo	Modalidad de Trabajo	Indicadores de logro
4h	Individual	IL1.2



Antecedentes generales

En esta guía encontrarás los contenidos asociados a cursores explícitos complejos en PL/SQL, junto con ejemplos y actividades prácticos a desarrollar.



Requerimientos para esta actividad

En esta actividad, los y las estudiantes deberán utilizar SQL Developer y seguir las instrucciones indicadas por el/la docente.



Sesión 1: CURSORES EXPLÍCITOS COMPLEJOS

Objetivos de Aprendizaje

1. Comprender el uso de cursores explícitos complejos con parámetros.
2. Desarrollar bloques PL/SQL que procesen datos y generen información relevante para el negocio.
3. Implementar ejemplos prácticos y realizar actividades para consolidar los conceptos aprendidos.

1. Introducción a Cursores Explícitos Complejos

1.1 Definición

- **Cursores Explícitos:** Son cursores definidos por el usuario para gestionar consultas SELECT que devuelven más de una fila.
- **Cursores complejos con Parámetros:** Cursores que aceptan parámetros para filtrar datos y proporcionar flexibilidad en el procesamiento de datos.

1.2 Ventajas

- **Control sobre el procesamiento de datos:** Permiten manipular y procesar filas individualmente.
- **Flexibilidad:** Los parámetros permiten reutilizar cursores en diferentes contextos.

1.3 Escenarios de Aplicación

- **Procesamiento de datos:** Generación de informes, análisis de datos en tiempo real.
- **Automatización:** Procesos repetitivos que necesitan manipulación de filas específicas.

2. Declaración y Uso de Cursores Explícitos con Parámetros

2.1 Sintaxis de Declaración

Declaración de un cursor con parámetros:

```
CURSOR cursor_name (parameter_name datatype) IS  
    SELECT statement WHERE condition;
```

Apertura de un cursor con parámetros:

```
OPEN cursor_name(parameter_value);
```

Bucle de FETCH:

```
FETCH cursor_name INTO variable1, variable2, ...;
```

Cierre del cursor:

```
CLOSE cursor_name;
```

2.2 Ejemplo Básico

Cursor con parámetros para empleados de un departamento específico:

```
DECLARE  
    CURSOR emp_cursor (p_dept_id dept.deptno%TYPE) IS  
        SELECT empno, ename, sal  
        FROM emp  
        WHERE deptno = p_dept_id;  
    v_emp_id emp.empno%TYPE;  
    v_emp_name emp.ename%TYPE;  
    v_salary emp.sal%TYPE;  
BEGIN  
    OPEN emp_cursor(10);  
    LOOP  
        FETCH emp_cursor INTO v_emp_id, v_emp_name, v_salary;  
        EXIT WHEN emp_cursor%NOTFOUND;  
        DBMS_OUTPUT.PUT_LINE('ID: ' || v_emp_id || ', Name: ' || v_emp_name  
|| ', Salary: ' || v_salary);  
    END LOOP;  
    CLOSE emp_cursor;  
END;
```

/

3. Ejemplos Prácticos

3.1 Ejemplo: Cursores con Parámetros para Filtrar Datos

Cursores que filtran empleados por departamento y salario mínimo:

```
DECLARE
  CURSOR emp_cursor (p_dept_id dept.deptno%TYPE, p_min_salary NUMBER) IS
    SELECT empno, ename, sal
    FROM emp
    WHERE deptno= p_dept_id AND sal >= p_min_salary;
  v_emp_id emp.empno%TYPE;
  v_emp_name emp.ename%TYPE;
  v_salary emp.sal%TYPE;
BEGIN
  OPEN emp_cursor(20, 3000);
  LOOP
    FETCH emp_cursor INTO v_emp_id, v_emp_name, v_salary;
    EXIT WHEN emp_cursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('ID: ' || v_emp_id || ', Name: ' || v_emp_name
|| ', Salary: ' || v_salary);
  END LOOP;
  CLOSE emp_cursor;
END;
/
```

3.2 Ejemplo: Uso de Cursores sin Parámetros para Generar Reportes

Cursor para generar un reporte de empleados y sus departamentos:

```
DECLARE
  CURSOR emp_dept_cursor IS
    SELECT e.empno, e.ename, d.dname
    FROM emp e
    JOIN dept d ON e.deptno = d.deptno;
  v_emp_id emp.empno%TYPE;
  v_emp_name emp.ename%TYPE;
  v_dept_name dept.dname%TYPE;
BEGIN
  OPEN emp_dept_cursor;
  LOOP
    FETCH emp_dept_cursor INTO v_emp_id, v_emp_name, v_dept_name;
    EXIT WHEN emp_dept_cursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('ID: ' || v_emp_id || ', Name: ' || v_emp_name
|| ', Department: ' || v_dept_name);
  END LOOP;
  CLOSE emp_dept_cursor;
END;
/
```

4. Actividades Prácticas

Actividad 1: Declarar y Usar Cursores con Parámetros

1. Objetivo: Crear un cursor con parámetros para seleccionar empleados por departamento y salario mínimo.
2. Pasos:
 - Define un cursor con parámetros dept_id y min_salary.
 - Abre el cursor para un departamento específico y donde los salarios sean mayores o iguales al salario mínimo pasado como parámetro.
 - Muestra los resultados en la consola.

Actividad 2: Generar Reportes con Cursores

1. Objetivo: Crear un cursor sin parámetros para generar un reporte de empleados y sus salarios.
2. Pasos:
 - Define un cursor que seleccione empno, ename, y sal de la tabla emp.
 - Abre el cursor y recorre todas las filas, mostrando los resultados.

Conclusión de la Sesión 1

- Los cursores con parámetros proporcionan flexibilidad y control en la selección de datos.
- Facilitan el procesamiento dinámico de datos según criterios específicos.
- Actividades prácticas para implementar y utilizar cursores explícitos con parámetros en PL/SQL.

Recomendaciones

1. **Revisar la documentación oficial de Oracle sobre cursores** para más detalles.
2. **Practicar el uso de cursores en diferentes escenarios** para comprender mejor sus aplicaciones.
3. **Aplicar cursores complejos en proyectos reales** para mejorar la eficiencia del procesamiento de datos.

Recursos Adicionales

Orfali, C. M. (s/f). *Aprendiendo sobre los Cursores en PL/SQL*. Aprenderdesarrolloapex.com. Recuperado el 25 de julio de 2024, de <https://aprenderdesarrolloapex.com/aprendiendo-sobre-los-cursos-en-pl-sql/>

Informáticas, B. L. •. [@bilbaolabs]. (2024, mayo 6). 15. *CURSORES con PARÁMETROS en PL/SQL - ORACLE*. Youtube. https://www.youtube.com/watch?v=SV_9CaOYtZ4