

DEPURACION DE PL/SQL EN EL SQLDEVELOPER

Profesor: Héctor Gómez Gauchía

Preparación para la ejecución

Vamos a probar cómo funciona la depuración en el PL/SQL. Usamos los scripts del Campus Virtual:

\Ejercicios-y-examenes-CV\PLSQLejemplosEjecutablesTeoria con la BDEjemplo

Preparamos todos los elementos siguiendo estos pasos:

- La ejecución del *trigger* utiliza una *secuencia* que es una variable global especial que actúa como un contador. Para crearla necesitamos darnos permiso para hacerlo, entrando desde el ADMINUSER(pide clave al profesor):
`GRANT create sequence TO tuUsuario;`
- Después, en nuestro usuario personal:
`drop sequence cuentapremio; -- por si existía antes`
`create sequence cuentapremio; -- crea la secuencia`
`select cuentapremio.nextval from dual; -- vemos su valor`
`commit; -- confirmamos los cambios`
- Ejecuta los siguientes scripts en orden:
 - Crear-BDejemplo.sql - Inicializaremos nuestra base de datos
 - proc3.sql - Crea un procedimiento, pero NO lo ejecuta
 - proc3plus2.sql
 - trigger3.sql
 - cursor3.sql
- Abre los procedimientos y el trigger en el editor (están en la carpeta Procedimientos y Disparadores)
 - Repasa en el editor de cada uno, el contenido de las **Tabs**:
Dependencias (tablas y procs que usa), *Errores* (compilación) y *Referencias* (otros procs que lo llaman)

Información útil para uso de PL/SQL en el SQLDeveloper

- PL/SQL es un lenguaje *compilado*: No olvides de compilar de nuevo cada vez que modificas algo.
- Para ver el número de línea en el editor:
 - En el margen izquierdo del panel del editor: botón dcho "Conmutar num. línea"
- Hacer click en la carpeta "Procedimientos", abre el tuyo
 - Si tiene una X roja es que tiene errores de compilación
 - Pestaña "Código" (en el editor del proc.)
- Al compilar puede dar errores:
 - Ver ventana "Compilador - Log"
 - Doble click en el error: se posiciona en la línea de código
- Dos modos de trabajar:
 - a.- Ejecutar para encontrar errores:
 - "Compilar para depurar": pone sombra verde sobre el nombre
 - Ejecutar para Depurar (insecto): te pide parámetros para ejecutar
 - Click en el "valor de Entrada" para cambiarlo
 - Hay dos enfoques complementarios de Depuración: ver más abajo.
 - b.- Para ejecutar normal, si ya está sin errores:
 - "Compilar"
 - Ejecutar: Triángulo verde

Depuración Básica en Ejecución

Una forma básica de encontrar errores, es probar trozos del código:

- Comentando trozos de líneas de código:
 - un conjunto de líneas entre `/*` y `*/`
 - una línea poniendo `--`: la parte dcha de la línea queda comentada
- Poniendo mensajes que dicen por donde pasa y contenido de variables
 - hay que activar el buffer de salida para DBMS_OUTPUT en el sqldeveloper para que el put_line funcione:
`set serveroutput on`
- Inconveniente de este método: solo ves los mensajes cuando termina el programa, no paso a paso.

Depuración Profesional en Ejecución: PL/SQL DEBUG

Depurar tu procedimiento usando el Depurador del SQL DEVELOPER, complementa el método anterior. Puedes: ir instrucción a instrucción, parar en una línea de código (breakpoint) y consultar variables en cada paso.

Para más detalles tienes este tutorial:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/prod/appdev/sqldev/plsql_debug/plsql_debug_otn.htm

En el apartado "Creating and Compiling a PL/SQL Procedure"

PASOS:

- Entro con el ADMINUSER y doy privilegios a mi usuario:
`GRANT DEBUG CONNECT SESSION, DEBUG ANY PROCEDURE TO miUsuario`
- Se asume que has hecho los pasos del apartado "Preparación para la ejecución"
- Para ver los mensajes del put_line set serveroutput on
- Antes de ejecutar, hay que "Compilar para depuración" en icono ruedas:
 - Hazlo para proc3.sql, proc3plus2.sql, trigger3.sql y cursor3.sql.
 - Después, cierras los ficheros que ha abierto.
 - Abres desde la carpetas (disparadores, procedimientos, funciones) los cuatro en el editor.
→ No olvides compilar de nuevo cada vez que modificas algo, ej: un breakpoint de los que se indica abajo.
- Ejecuta ahora Cursor3 y verás la ejecución hasta el final: nada interesante sucede.
 - Entra en menú Herramientas + Preferencias
 - Busca "Depurador" en la lista de la izquierda:
 - En "Opción de Inicio de Depuración" hay varias posibilidades:
 - Escogemos : Ir a 1ª línea ejecutable de Código
 - Otra opción para más detalle : Ir siguiente línea de Código
- Ejecutar depurando es solo interesante si has puesto breakpoints, y usas el inspector:
 - Para poner en Cursor3 breakpoints (punto de Ruptura) donde se parará la ejecución:
 - (Editor) En el margen izq: click en el núm.línea. Aparece un punto rojo. Coloca estos:
 - `OPEN cursor_ricos`
 - `FETCH cursor_ricos`
- Para Ejecutar depurando pulsa el insecto (NO el triángulo)
 - Aparece un cuadro para pedir datos de ejecución:
 - No tenemos parámetros en este ejemplo: Aceptar.
 - Si da error -> ORA-01031: privilegios insuficientes
 - Vuelve a conceder privilegios a tu cuenta.
 - Después cierra y abre la conexión con tu usuario.
- (panel Depuración) En la Tab "depurando. . ." se ve cómo se queda parado al principio del bloque:
 - Se puede usar los iconos de ese panel para avanzar
 - Avanza paso a paso con F8 (siguiente línea de código): observa por donde va pasando.
 - y comprueba en la Tab "Datos" los valores que vas teniendo cada variable.
- Vuelve a ejecutar el procedimiento cursor3 con el insecto y avanza con F9 hasta llegar al `FETCH cursor_ricos`:
 - En panel Depuración: Reanudar (F9) es para avanzar seguido, parando solo en los check points
 - Dentro del editor del procedimiento:
 - Marca la línea del primer " update invierte . . " con el ratón
 - pulsa b.dcho y escoge "ir a cursor":
 - ejecuta todo y se para en la línea donde marcaste con el ratón (el cursor)
 - En el Tab "Datos" pulsa b.dcho sobre TDNCL :tiene '00000002'
 - escoge "Modificar valor" : coloca '00000004'
 - Avanza paso a paso y comprueba que entra en el IF del cliente '00000004'.

Repetimos ejecución de Cursor3 siguiendo estos pasos:

- Vamos a poner otros breakpoints en Cursor3 en estas líneas:
 - `proc3plus2('0000`

- `proc3('0000 . . .`
 - `proc3('0000. . .`
- Vamos a poner breakpoints en Proc3Plus2 en estas líneas:


```
update cliente;
DBMS_output.put_line('--- proc3plus, no encuentra '|| dnibusca);
update invierte;
update invierte;
Y compila para depurar
```
- Vamos a poner breakpoints en Proc3 en estas líneas:


```
RAISE cliente_listillo;
TDNICL := 'PROC3: -- ELSE ';
Tcoderror:= SQLCODE;          -- en la excepción de cliente_listillo
Y compila para depurar
```
- Vamos a poner breakpoints en trigger3 en estas líneas:


```
DBMS_output.put_line('--- trigger3, CREAM: '|| . . . .;
DBMS_output.put_line('--- trigger3, CREADO');
Y compila para depurar
```
- Además vamos a inspeccionar continuamente la variable DNIBUSCA antes y después de la llamada:
 - Ejecutar `cursor3` para Depuración. Cuando se detenga:
 - (en editor) Click B.dcho encima de la declaración de dicha variable en el código:
 - Inspeccionar, (aparece ventana Inspector)
 - Puedes escoger también Observar
 - En la ventana Inspector: Doble click sobre el valor concreto de DNIBUSCA
 - Escribe el valor '00000001' en la ventana “modificar Valor”, que se ha abierto
 - Sigue ejecutando con F9 y observa cómo cambia ese valor.

Puede aparecer otro panel con la "Pila" de procedimientos que participan: El primero es un bloque anónimo que es la ejecución que hemos empezado nosotros con `Cursor3`.

- Continúa la ejecución de `Cursosr3` con Reanudar (F9) como antes, pasa por todos los procedimientos
 - Observa: cuántas veces se dispara el trigger? Y qué valores va tomando DNIBUSCA
- Si consultas la tabla cliente: no observas cambios hasta hacer commit y refrescar. Pero en Tab “Datos” ves cambios
- NOTA: no olvides de compilar normal cuando ya funcione bien, porque ralentiza la ejecución y eso es peligroso en funcionamiento real, el llamado "en Producción".
- Para ver más detalles:

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/prod/appdev/sqldev/plsql_debug/plsql_debug_otn.htm

Tengo dos breakpoint en Proc3Plus2: líneas 21 y 29. Debajo se ve el panel “Depurando” con la ejecución completa del cursor que llama a proc3plus2

The screenshot shows an IDE window with a PL/SQL procedure named `proc3plus2`. The code is as follows:

```

20 BEGIN
21
22   DBMS_output.put_line('--- proc3plus, begin '|| dnibusca);
23   update cliente
24       set NombreC = NombreCL
25       WHERE DNI = dnibusca;
26
27
28   IF SQL%NOTFOUND THEN
29       DBMS_output.put_line('--- proc3plus, no encuentra '|| dnibusca);
30       insert into cliente
31           values (dnibusca,NombreCL,DirCL,TelCL);
32   END IF;
33   TDNICL := dnibusca;
34   IF TDNICL = '00000005' THEN
35       DBMS_output.put_line('--- proc3plus, en el IF: '|| TDNICL);
36   ELSE
37       TDNICL := ':ELSE ';
38       DBMS_output.put_line('--- proc3plus, otro cliente: '|| TDNICL);
39   END IF;
40
41 END proc3plus2;

```

Below the code editor, the "Depurando" (Debugging) panel is visible, showing the execution log for the cursor `IDEConnections%23ABD-14-15-HECTOR.jpr`. The log includes the following messages:

```

Conectando a la base de datos ABD-14-15-HECTOR.
Ejecutando PL/SQL: ALTER SESSION SET PLSQL_DEBUG=TRUE
Ejecutando PL/SQL: CALL DBMS_DEBUG_JDWP.CONNECT_TCP( [redacted], '56275' )
El depurador ha aceptado la conexión de la base de datos en el puerto 56275.
Punto de ruptura de origen: CURSOR3.pls:52
Punto de ruptura de origen: PROC3PLUS2.pls:23
Ejecutando PL/SQL: CALL DBMS_DEBUG_JDWP.DISCONNECT()
---- cursor3: para depuración
---- cursor3: sumando , DNI: 00000002
---- cursor3: restando , DNI: 00000004
--> cursor3: proceso acabado por EXIT loop
--- proc3plus, begin 00000005
--- proc3plus, en el IF: 00000005
El proceso ha terminado.
Desconectando de la base de datos ABD-14-15-HECTOR.
Depurador desconectado de la base de datos.

```

Ahora ejecución incluyendo el disparo del Trigger3

...OR
TRIGGER3
CURSOR3
PROC3PLUS2
ANONYMOUS_BLOCK
CLIENTE

Código
Referencias
Permisos
Detalles
Errores
Dependencias
Perfiles

Buscar
ABD-14-15-HECTOR

```

10 DECLARE
11
12 -- vars de trabajo
13
14
15     Tcoderror NUMBER;
16     Ttexterror VARCHAR2(100);
17     Tnumfact compas.NumF$Type;
18
19
20 BEGIN
21     select cuentapremio.nextval into Tnumfact from dual;
22
23
24     DBMS_output.put_line('--- trigger3, CREAR: '|| :new.DNI || '-'|| Tnumfact || '-'|| :new.N
25
26     INSERT into compas
27         VALUES (:new.DNI, 10000001,Tnumfact,999,'REGALO', 100);
28
29     DBMS_output.put_line('--- trigger3, CREADO');
30

```

trigger: trigger3
38

Depurando: IdeConnections%23ABD-14-15-HECTOR.jpr - Log

Conectando a la base de datos ABD-14-15-HECTOR.

Ejecutando PL/SQL: ALTER SESSION SET PLSQL_DEBUG=TRUE

Ejecutando PL/SQL: CALL DBMS_DEBUG_JDWP.CONNECT_TCP('58101')

El depurador ha aceptado la conexión de la base de datos en el puerto 58101.

Punto de ruptura de origen: CURSOR3.pls:52

Punto de ruptura de origen: PROC3PLUS2.pls:23

Punto de ruptura de origen: PROC3PLUS2.pls:43

Punto de ruptura de origen: TRIGGER3.pls:24

Punto de ruptura de origen: TRIGGER3.pls:29

Punto de ruptura de origen: TRIGGER3.pls:24

Punto de ruptura de origen: TRIGGER3.pls:29

Punto de ruptura de origen: PROC3PLUS2.pls:49

Ejecutando PL/SQL: CALL DBMS_DEBUG_JDWP.DISCONNECT()

```

---- cursor3: para depuración
---- cursor3: sumando , DNI: 00000002
---- cursor3: restando , DNI: 00000004
--> cursor3: proceso acabado por EXIT loop
--- proc3plus, begin 00000005
--- proc3plus, en el IF: 00000005
--- trigger3, CREAR: 00000003-2-Empresa 55      100
--- trigger3, CREADO
--- trigger3, CREAR: 00000003-3-Empresa 33      100
--- trigger3, CREADO

```

El proceso ha terminado.

Desconectando de la base de datos ABD-14-15-HECTOR.

Depurador desconectado de la base de datos.