

Depuración de PLSQL con SQLDeveloper

Álvaro González Sotillo

20 de enero de 2022

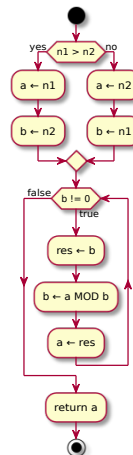
Índice

1. Ejemplo	1
2. Habilitar el <i>debug</i>	4
3. Alternativas a depurar	4
4. Referencias	5

1. Ejemplo

- **Algoritmo de Euclides** para el máximo común divisor de dos números

1.1. Diagrama de flujo



1.2. Código PLSQL

```
create or replace function MCD(n1 number,n2 number) return number
is
  a number;
  b number;
```

```

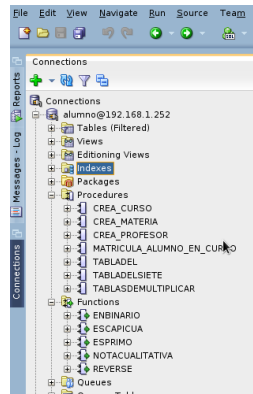
    res number;
begin
    if (n1>n2) then
        a := n1;
        b := n2;
    else
        a := n2;
        b := n1;
    end if;

    while( b!=0 ) loop
        res := b;
        b := mod(a,b);
        a := res;
    end loop;

    return res;
end;
/

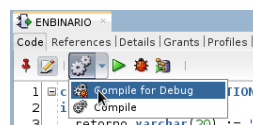
```

1.3. Lista de procedimientos y funciones



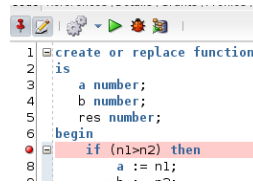
1.4. Compilar para depurar

- Para poder depurar, debe compilarse para depurar
- Si no, los *breakpoints* no funcionarán



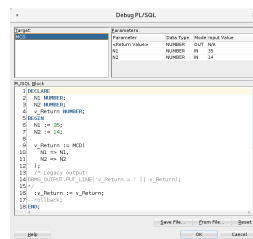
1.5. Poner un *breakpoint*

- Un *breakpoint* es una línea donde la ejecución parará temporalmente para poder inspeccionar el estado de las variables
- f5 o botón derecho sobre los números de línea



1.6. Bloque de invocación

- Es un bloque anónimo que invocará a la función/procedimiento
- Permite introducir valor a los parámetros de entrada, y recoger parámetros de salida y retorno de la función



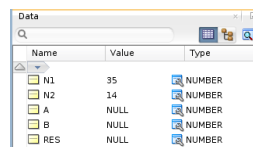
1.7. Avance del programa



- Abortar la ejecución
- Colocar el cursor en la siguiente instrucción a ejecutar
- *Step over* F8: Ejecutar la siguiente línea, sin entrar en funciones
- *Step into* F7: Ejecutar la siguiente línea, entrando en funciones
- *Step out*: Ejecutar hasta acabar la función actual
- Continuar hasta siguiente *breakpoint* F9
- Parar la ejecución y comenzar a debuguear
- Deshabilitar todos los *breakpoints*

1.8. Ventana Data

- Contiene el valor de las variables de la función/procedimiento
- Su valor se irá actualizando cuando avance el programa



2. Habilitar el *debug*

- Privilegios
- Habilitar puestos de depuración

2.1. Privilegios

```
Executing PL/SQL: ALTER SESSION SET PLSQL_DEBUG=TRUE
Executing PL/SQL: CALL DBMS_DEBUG_JDWP.CONNECT_TCP( '192.168.1.119', '34685' )
ORA-01031: insufficient privileges
ORA-06512: at "SYS.DBMS_DEBUG_JDWP", line 68
ORA-06512: at line 1
This session requires DEBUG CONNECT SESSION and DEBUG ANY PROCEDURE user privileges.
```

```
grant debug connect session to USUARIOQUEDEBUEA;
grant debug any procedure to USUARIOQUEDEBUEA;
```

2.2. Habilitar puestos de depuración

- La instancia de Oracle puede limitar los ordenadores que pueden debuguear

```
begin
  dbms_network_acl_admin.append_host_ace
    (host=>'IP DEL CLIENTE QUE DEBUEA',
     ace=> sys.xs$ace_type( privilege_list=>sys.XS$NAME_LIST('JDWP') ,
                           principal_name=>'USUARIO QUE DEBUEA',
                           principal_type=>sys.XS_ACL.PTYPE_DB) );
end;
/

grant execute on DBMS_DEBUG_JDWP to USUARIOQUEDEBUEA;
```

3. Alternativas a depurar

- No siempre es fácil depurar
- No se puede volver atrás al depurar
- Con algo de práctica, las **trazas** pueden ser más útiles

3.1. Ejemplo con trazas

```
create or replace function MCD(n1 number,n2 number) return number
is
  a number;
  b number;
  res number;
begin
  if (n1>n2) then
    dbms_output.put_line( 'n1 es mayor' );
    a := n1;
    b := n2;
  else
    dbms_output.put_line( 'n2 es mayor' );
```

```

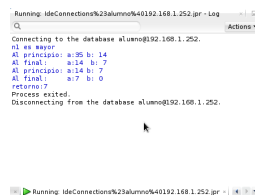
    a := n2;
    b := n1;
end if;

while( b!=0 ) loop
    dbms_output.put_line( 'Al principio: a:' || a || ' b: ' || b );
    res := b;
    b := mod(a,b);
    a := res;
    dbms_output.put_line( 'Al final:      a:' || a || ' b: ' || b );
end loop;

dbms_output.put_line( 'retorno:' || res );

return res;
end;
```

3.2. Ejemplo con trazas



4. Referencias

- Formatos:
 - [Transparencias](#)
 - [PDF](#)
 - [EPUB](#)
- Creado con:
 - [Emacs](#)
 - [org-re-reveal](#)
 - [Latex](#)
- Alojado en [Github](#)