

PRÁCTICA P3. PROGRAMACIÓN CON PL/SQL

FAQ. PREGUNTAS MÁS FRECUENTES

0.GENERALES (afectan a cualquier ejercicio).-

0.1.

PREGUNTA.

¿Podemos usar los RECORDS?

Ejemplo:

```
TYPE datosOng IS RECORD( codigo ONG.codigo%TYPE, nombre ONG.nombre%TYPE);  
TYPE datosTrab IS RECORD(dni TRABAJADOR.dni%TYPE ....etc);
```

RESPUESTA.

No es necesario utilizar los TYPE. Interesa que utilicéis CURSORES y variables (individuales, de tipo fila o de tipo cursor) para volcar en ellas los datos extraídos de la BD y usarlas para procesar los datos durante el recorrido de los cursores.

El principal cometido de esta práctica es que aprendáis y comprendáis lo más básico del acceso a bases de datos mediante programas, esto es, lo que es un resultset, un cursor, cómo aprovecharse de las excepciones, etc.

0.2.

PREGUNTA

- ¿Cómo puedo ver los **errores de compilación** de una Función, un Procedimiento o un Trigger?

RESPUESTA

En el panel de Conexiones de SQL Developer (a la izquierda), que muestra los objetos de base de datos (tablas, vistas, etc.) que cada usuario conectado ha creado, haz click sobre la carpeta "Procedimientos" (o en el de "Funciones", si es el caso).

Ahí aparecerá un elemento con el nombre de tu procedimiento (o función). Si tiene errores de compilación tendrá un icono redondo y rojo con una equis. Si está correcto, el icono será verde.

Si haces click sobre el elemento, aparecerá una nueva pestaña a la derecha (junto a las hojas de trabajo que tuvieras abiertas). Es un editor de código. Ahí se muestra el código y quizá alguna línea subrayada en rojo (el primer error de compilación).

Hay una barra de herramientas que contiene un icono con dos ruedas en engranaje: permite compilar y obtener información de depuración si abris sus opciones pinchando en la flechita que tiene a la derecha.

Os recomiendo solucionar primero el primer error que aparece, pues no es raro que algunos de los demás sean consecuencia del primero.

EJERCICIO 1.-

1.1.

PREGUNTA

En el ejercicio 1, ¿cómo puedo conseguir que las columnas ocupen lo mismo independientemente de la longitud de cada nombre que hemos de imprimir?

RESPUESTA

Se puede utilizar las funciones LPAD y RPAD, que tenéis explicadas en el documento

"https://aulavirtual.um.es/access/content/group/1903_G_2015_N_N/pRACTICAS/oracle-documentacion/funciones-sql.pdf".

EJERCICIO 2

2.1.

PREGUNTA

En el ejercicio 2, para mostrar el número de trabajadores, número de proyectos en los que participa y número de acciones, he pensado en recorrer cada cursor 2 veces (2 bucles), la primera para contar el número y la segunda para imprimir los nombres para mantener el formato que se pide ya que no se pueden crear funciones.

¿Esta forma es correcta o hay otra forma?

RESPUESTA

Hay otra forma mejor (más eficiente y más correcta desde el punto de vista de la programación con bases de datos): calcularlo directamente con un "SELECT COUNT(...) INTO variable..." en el código PL/SQL.

Cada vez que se abre un cursor (OPEN) se ejecuta la SELECT para traer un resultset al programa, que luego vas a procesar cuando recorres el cursor...

¿Qué sentido tiene, por tanto, 1) ejecutar una SELECT,. 2) Recorrer el resultado, ejecutando FETCHs, con el simple propósito de contar filas...3) Cerrar el cursor... , cuando se puede hacer todo de una con una simple ejecución de una SELECT?

El "SELECT COUNT(...) INTO" es mucho más económico ("cuesta" lo mismo que simplemente abrir el cursor!)... además de más correcto desde el punto de vista del estilo de programación PL/SQL: hay que aprovecharse de que estamos en un sistema de bases de datos. Todo lo que podamos conseguir que trabaje o ejecute directamente Oracle (con sentencias SELECT, INSERT, UPDATE, DELETE), no debemos hacerlo nosotros en el código.

IMPORTANTE: los cursores sólo tienen sentido usarlos cuando a cada una de las filas que han obtenido al abrirse hay que aplicarle un procesamiento que no podemos realizar directamente con sentencias SQL.

EJERCICIO 3.- PAQUETE, FUNCIÓN, PROCEDIMIENTO.

3.1.

PREGUNTA

- ¿Cómo consigo realizar la función simplemente utilizando excepciones?
- Si pongo varias SELECT en el código, en la parte de excepciones no sabe distinguir qué SELECT ha sido la que ha provocado la excepción. ¿Cómo lo hago?

RESPUESTA

Antes de nada: No es muy correcto utilizar variables "flag" que nos informen de cuál ha sido la SELECT que ha generado la excepción...

Hay que utilizar en nuestro beneficio la manera de proceder de Oracle cuando ejecuta una SELECT en un bloque de código y manejarla con un cursor implícito.

Se puede preguntar si existe determinada fila en una tabla mediante una SELECT, cuyo resultado sólo puede ser de tres tipos: una fila (sí existe), ninguna fila (no existe) o muchas filas (sí existe, varias veces).

Sólo cuando se obtiene una fila el flujo de ejecución continúa por la siguiente línea de código tras la SELECT.

Los otros dos casos hay que procesarlos capturando las excepciones predefinidas (NO_DATA_FOUND, TOO_MANY_ROWS).

Y al tratarlas, hay que tener en cuenta que un WHEN <excepcion> THEN puede incluir un bloque PL/SQL completo (DECLARE, BEGIN, EXCEPCION, END;) donde se puede incluir, por tanto, otra SELECT y capturar otras excepciones surgidas de la ejecución de la misma.

```
DECLARE
BEGIN
    ...
    SELECT ... INTO...
    FROM...
    WHERE;
    ...
EXCEPTION
    WHEN <excepcion> THEN
        DECLARE
        BEGIN
            SELECT ... INTO...
            FROM...
            WHERE...;
            ...
        EXCEPTION
            WHEN <excepcion> THEN
                END;
        ...
    END;
```

3.2.

PREGUNTA

- ¿Mi función/procedimiento puede invocar a otras funciones o procedimientos?

RESPUESTA

NO. Es posible programar la función y el procedimiento sin la necesidad de que invoquen a ningún otro subprograma PL/SQL (procedimiento ni función).

EJERCICIO 4.- TRIGGERS

4.1.

PREGUNTA

- ¿Mi trigger puede invocar funciones o procedimientos?

RESPUESTA

NO. El propósito de este ejercicio es conseguir programar el cuerpo de los triggers sin necesidad de invocar procedimientos o funciones.