

Práctica 7: PL-SQL Básico y Disparadores

Compilación de bloques PL/SQL

Todos los bloques PL/SQL deben incluir al final una fila con el carácter '/' para compilar y ejecutar. Si no se introduce el carácter '/' el editor no identifica que se haya finalizado el bloque PL/SQL y por tanto no lo compila ni ejecuta.

SQL Developer y SQL*Plus en el caso de que el proceso de compilación falle muestran los errores de compilación, incluyendo información acerca del problema y la posición del mismo. En el caso de subprogramas almacenados, SQL*Plus no muestra los errores directamente. Para examinar los errores de compilación y ejecución de un programa almacenado en SQL*Plus se utiliza la orden:

```
SHOW ERRORS;
```

Salida por consola en SQL*Plus

Para poder utilizar la consola en SQL*Plus es necesario activar el “eco” del buffer ejecutando la siguiente orden:

```
SET SERVEROUTPUT ON [FORMAT WRAPPED] [SIZE<tamaño>]
```

Enunciado

Siguiendo con la base de datos descrita en prácticas anteriores, se pide codificar, utilizando PL/SQL, los bloques de código descritos a continuación.

1. Procedimiento almacenado llamado **pedidosCliente** que reciba como parámetro el id de un cliente (*customer_id*) y muestre por pantalla sus datos personales (*customer_id*, *cust_first_name*, *cust_last_name*), junto con un listado con los datos de los pedidos que ha realizado (*order_id*, *order_date*, *order_status* y *order_total*), ordenados crecientemente por fecha. En caso de error (el id del cliente no existe o no hay pedidos para ese cliente), deberá mostrarse por pantalla un mensaje de advertencia explicando el error. Al finalizar el listado se deberá mostrar la suma de los importes de todos los pedidos del cliente.

Incluye un bloque de código anónimo para probar el procedimiento. Ejemplo de salida tras llamar al procedimiento **pedidosCliente** con 4 ids distintos:

```
-----
Id: 101
Nombre: Constantin
Apellido: Welles

Pedidos:
  Código: 2458
  Fecha:  16/08/99 14:34:12
  Estado: 0
```

Importe: 78367,6

Código: 2447

Fecha: 27/07/00 07:59:10

Estado: 8

Importe: 33893,6

Código: 2413

Fecha: 29/03/00 12:34:04

Estado: 5

Importe: 48552

Código: 2430

Fecha: 02/10/99 05:18:36

Estado: 8

Importe: 29669,9

Total Pedidos: 190483,1

Id: 102

Nombre: Harrison

Apellido: Pacino

Pedidos:

Código: 2397

Fecha: 19/11/99 15:41:54

Estado: 1

Importe: 42283,2

Código: 2432

Fecha: 14/09/99 08:53:40

Estado: 10

Importe: 10523

Código: 2414

Fecha: 29/03/99 13:22:40

Estado: 8

Importe: 10794,6

```

Código: 2431
Fecha: 14/09/98 06:03:04
Estado: 1
Importe: 5610,6

Total Pedidos: 69211,4
-----
ERROR: ID 1 no encontrado
-----

Id: 981
Nombre: Daniel
Apellido: Gueney

No hay pedidos para el cliente con id:981

```

2. Procedimiento almacenado llamado **revisaPedidos** cuya misión es comprobar la consistencia de los datos de todos los pedidos. El campo *order_total* de la tabla *ORDERS* debe almacenar la suma del precio todos los elementos del pedido (*unit_price * quantity*). El procedimiento debe verificar estos datos para todos los pedidos, y cuando el campo *order_total* no se corresponda con la suma de los elementos deberá mostrar un mensaje indicando el número de pedido y el total actual, el total calculado a partir de los datos de la tabla *ORDER_ITEMS*. Si todos los pedidos están correctos se mostrará un mensaje indicando que la comprobación ha sido un éxito.

Incluye un bloque anónimo de prueba, al ejecutarlo se debería de mostrar lo siguiente:

```

----- Pedidos incorrectos: -----
Pedido: 2388
totalOrder: 282826,3
totalOrderItem: 282738,3

```

Modifica los pedidos inconsistentes y vuelve a ejecutar el procedimiento *revisaPedidos* para comprobar que funciona bien cuando la información de la tabla es correcta.

3. Nos dicen que es conveniente reflejar en la tabla *PRODUCT_INFORMATION* la cantidad total disponible de cada producto en todos los almacenes de la empresa. Para cumplir con este nuevo requisito debes de hacer lo siguiente:
 - a. Modificar la tabla *PRODUCT_INFORMATION* y añadir el campo *quantity*.
 - b. Crear un bloque de PL/SQL anónimo que actualice automáticamente todas las filas de la tabla para reflejar su cantidad en los almacenes. Si algún

producto está sin existencias la cantidad de la tabla *PRODUCT_INFORMATION* debe de estar a 0.

-
- c. Crea un procedimiento almacenado llamado *actualizaPIQuantity* que reciba como parámetros un identificador de producto y una cantidad y actualice en la tabla *PRODUCT_INFORMATION* la cantidad del producto sumándole la cantidad recibida por parámetro.

-
- d. Crea un disparador que cada vez que se actualice la cantidad de un producto en la tabla *INVENTORIES*, actualice la cantidad total en la tabla *PRODUCT_INFORMATION*. Debes de hacer uso del procedimiento creado en el apartado anterior.

-
- e. Comprueba que el disparador creado funciona correctamente.