

PRÁCTICA P3. Programación en PL/SQL**Objetivos**

- Comprender los conceptos y estructuras de programación de PL/SQL (extensión procedural del SQL de Oracle): bloques, cursores, excepciones, estructuras de selección e iteración, etc.
- Construir, ejecutar y depurar código PL/SQL: bloques anónimos, procedimientos, funciones, paquetes (*packages*), disparadores (*triggers*), etc.

Contenidos

Esta práctica consiste en la resolución de una serie de ejercicios de programación en PL/SQL de Oracle, sobre el **esquema de bases de datos** que cada grupo de prácticas haya **creado durante la práctica P2. Definición y modificación de datos en SQL.**

Normas importantes y obligatorias para resolver los ejercicios

- ❑ Puesto que para realizar esta práctica es necesario utilizar el **esquema** elaborado en la práctica **P2**, se debe adjuntar el script completo de creación (y carga) del esquema. Hay que entregarlo SIEMPRE: incluirlo en el informe y entregar el fichero `.sql`.
 - Si **se entregó la práctica P2**, se debe trabajar con el esquema creado en la misma.
 - Da igual si se trabaja con el esquema y datos correspondientes a los ejercicios 1 y 2, o bien con los que quedan tras ejecutar todos los ejercicios de la práctica **P2**. Debe indicarse **cuál** de ellos se ha utilizado.
 - Si se ha cambiado el esquema que se entregó (corregido algún fallo o errata, ampliado o completado algo), el script entregado debe incluir comentarios o anotaciones para **indicar** cuáles han sido las **correcciones o mejoras** realizadas en el esquema.
 - Si **NO se entregó la práctica P2**, y puesto que no se puede probar el código PL/SQL sin la existencia del esquema y sus datos, es obligatorio crear (y rellenar con datos) dicho esquema. Es imprescindible **indicar esta circunstancia y entregar el script** de creación y carga.
- ❑ Es importante cuidar el **estilo de programación** PL/SQL. Por ejemplo, es una mala costumbre (aunque funcione) hacer un FETCH de un cursor que ya está vacío; no es necesario usar un CURSOR explícito para hacer una consulta que devuelve una única fila; no es necesario hacer un SELECT COUNT para obtener el número total de filas de un CURSOR cuando dicho total se puede obtener con un atributo %ROWCOUNT o incrementando una variable en cada iteración; etc.
- ❑ También es importante elegir el **algoritmo** más adecuado para resolver cada ejercicio. No es suficiente con que el ejercicio funcione (aunque sea al estilo “fuerza bruta”).
- ❑ **No se puede usar tablas temporales** para almacenar resultados intermedios.
- ❑ Se debe **insertar** en las tablas los **datos necesarios** para que se obtengan resultados que permitan **comprobar** que los ejercicios **funcionan** adecuadamente.
- ❑ No dedicar demasiado esfuerzo en formatear o tabular la salida por pantalla; es suficiente con que los resultados sean **legibles**.
- ❑ Cumplida la fecha límite de entrega, **todos los objetos** (funciones, procedimientos, disparadores y paquetes) han de estar creados en la base de datos de prácticas `Oracle11g` de la Facultad, en la cuenta del grupo de prácticas (`bdgxxxx`, donde se debe sustituir `xxxx` por el número de grupo asignado).

Fecha límite de entrega

La práctica se puede entregar **hasta el jueves 21 de mayo a las 23:55pm**. Es decir, en el Aula Virtual la Tarea correspondiente, denominada de forma similar a “Entrega de Práctica P3. Programación en PL/SQL”, se cerrará en ese momento.

Antes de intentar resolver los ejercicios que hay que entregar, se recomienda resolver los incluidos a continuación, cuya dificultad es baja y media. Le resultará útil para comprender mejor el estilo de programación PL/SQL y para adquirir destreza con dicho lenguaje.

1. Escribir una **función** llamada `nombre_cliente` que dado un dni devuelva el nombre del cliente que corresponda. Si no existe el dni, debe devolver 'CLIENTE INEXISTENTE'.
2. Escribir un **procedimiento** llamado `subalternos_vendedor` tal que dado un dni de un vendedor, muestre por pantalla su nombre, y el dni y nombre de los vendedores de los que es jefe. Si alguno de éstos, a su vez, es jefe de otros, no hay que listarlos. El formato de la salida sería este:

```
<Nombre del vendedor D> ES JEFE DE <número de> VENEDORES:
- <dni. Nombre vendedor_1>.
- <dni. Nombre vendedor_2>.
...
- <dni. Nombre vendedor_n>.
```

3. Escribir un **trigger** de base de datos `T_PVP` tal que, cuando se introduzca una nueva venta de un coche, o se modifique el `pvp` de una venta antigua, se compruebe que *el precio de venta de un coche no supere un 15% al precio recomendado por la marca para ese modelo de coche*.

Ejercicios

1. **Listar** por pantalla, en dos columnas, los nombres de todos los vendedores y de todos los clientes. Ambas columnas han de estar ordenadas alfabéticamente y mostrarse de la forma que se indica a continuación. La última línea ha de mostrar el número total de vendedores y de clientes. Tenga en cuenta que, lógicamente, puede haber distinto número de vendedores que de clientes. No hay que escribir procedimientos ni funciones.

VENDEDORES	CLIENTES
-----	-----
<nombre vendedor_1>	<nombre cliente_1>
...	...
<nombre vendedor_n>	...
	<nombre cliente_m>

TOTAL: <n> VENDEDORES Y <m> CLIENTES	

2. Hacer un **listado** que para cada (nombre de) marca muestre sus concesionarios, indicando cuál de ellos es el concesionario principal de la marca, y para cada concesionario muestre los modelos de coche que tiene en su catálogo, indicando el recuento de coches de ese modelo que tiene el concesionario (vendidos o no, da igual).

Siga el formato que se indica a continuación. Note que el listado debe estar ordenado alfabéticamente por nombre de marca y modelo, y que también debe incluir todos los cálculos que se indican (número de concesionarios de cada marca, cuántos modelos de cada marca en cada concesionario, recuento de coches de cada marca/modelo...).

En este ejercicio obligatoriamente hay que usar cursores parametrizados. No hay que escribir procedimientos ni funciones.

```
MARCA <nombre marca_1>
--- <número de> CONCESIONARIOS:
----- <nombre concesionario_1>
----- <número de> MODELOS:
----- <modelo_1>. COCHES: <recuento>
      ...
----- <modelo_n>. COCHES: <recuento>
      ...
----- <nombre concesionario_i> ES EL PRINCIPAL
----- <número de> MODELOS:
----- <modelo_1>. COCHES: <recuento>
      ...
----- <modelo_p>. COCHES: <recuento>
      ...
----- <nombre concesionario_c>
      ...
...
MARCA <nombre marca_m>
...
```

3. Escribir un **package** llamado **P_CON** con el siguiente contenido:

- Una **función** llamada **compras_cliente**, que dado un dni de un cliente devuelva 1) '**CLIENTE SIN COCHES**' si ese dni corresponde a un cliente que no ha adquirido ningún coche, 2) la matrícula del coche comprado, si el cliente ha adquirido sólo un coche, 3) '**CLIENTE CON <número de> COCHES**' si el cliente ha comprado más de un coche (sustituyendo <numero de> por el recuento de los coches que ha adquirido), y 4) '**CLIENTE INEXISTENTE**' si no existe dicho cliente.
Importante: la función hay que programarla manejando excepciones. No se puede utilizar cursores explícitos ni sentencias SELECT COUNT que eviten el manejo de excepciones (sólo se puede usar para calcular el número de coches una vez se sabe que el cliente ha comprado más de uno).
- Un **procedimiento** **ventas_vendedor** tal que dado un dni de un vendedor, muestre por pantalla el nombre del vendedor, su concesionario y la marca que representa, y el listado de las ventas de coches en las que ha participado, indicando la fecha de venta, el modelo del coche y el pvp. Siga el formato indicado a continuación para la salida por pantalla.

```
VENDEDOR <nombre vendedor>
CONCESIONARIO <nombre concesionario>
MARCA <nombre marca>
<número de> VENTAS:
--FECHA:<fecha>; MODELO:<nombre modelo>. PVP: <euros> €.
...
--FECHA:<fecha>; MODELO:<nombre modelo>. PVP: <euros> €.
```

4. Escribir los **triggers** de base de datos necesarios para imponer que un vendedor sólo puede participar en las ventas de coches expuestos en el concesionario en el que está contratado. Tenga en cuenta que un vendedor podría cambiar de concesionario (de entre los que representan la misma marca). Considere también que podría modificarse el coche y/o el vendedor de una venta existente.

Documentación que se debe entregar

La entrega se realizará mediante el **Aula Virtual** (SAKAI), antes de la fecha límite indicada en la **Tarea** correspondiente. Importante: debe hacerlo cada uno de los miembros de cada grupo de prácticas.

Hay que entregar el **informe** de realización de la práctica (memoria), así como el conjunto de **guiones** (*scripts*) SQL elaborados.

Todos estos ficheros deben ser incluidos en un **fichero comprimido** con el nombre **bdgxxxx-p3** (.zip o .rar).

El **nombre y formato** de cada uno de los ficheros será el siguiente:

(1) **Informe** de la práctica. Documento llamado (en minúsculas) **bdgxxxx-p3.ext** (sustituir **bdgxxxx** por el nombre de su grupo de prácticas). La extensión (**.ext**), es decir, el formato del documento, puede ser **.pdf** (recomendado) o bien **.doc**, **.docx**, **.rtf**, **.odt**.

El informe debe tener las páginas numeradas y debe incluir lo siguiente;

- ☐ **Portada**, que muestre estos datos:
 - Asignatura (**Bases de Datos**), curso académico (**2014/15**) y convocatoria (**junio, julio, febrero**).
 - Identificador (**P3**) y nombre de la **práctica** (**Programación en PL/SQL**).
 - Nombre del **grupo** de prácticas (**bdgxxxx**), **nombre** y **apellidos** de cada componente.
 - Dirección de correo electrónico de cada integrante del grupo de prácticas.
 - Nombre del **profesor** al que se realiza la entrega.
- ☐ Sentencias de **creación y carga** del esquema elaborado en la práctica **P2** (ver 1ª norma de realización).
- ☐ Para **cada uno** de los **ejercicios** deberá aparecer lo siguiente:
 - *Número* de ejercicio y *enunciado* del ejercicio.
 - *Comentarios* y aclaraciones que se consideren necesarios.
 - **Código PL/SQL**, con indentaciones adecuadas y fácilmente legible.
 - **Pruebas** para comprobar el correcto funcionamiento del ejercicio. Concretamente...
 - Para los ejercicios **1**, y **2**, mostrar lo que sale por pantalla con la ejecución del código.
 - Para el ejercicio **3**, probar todos los posibles resultados de la función y también probar el procedimiento, al menos, con un vendedor que tenga varias ventas, con otro que no tenga ninguna, con un dni que no exista en la tabla vendedor, etc.
 - Para el ejercicio **4**, probar todos los eventos de cada uno de los *triggers* creados.

Nota: se puede presentar varias **soluciones alternativas** de un mismo ejercicio, en cuyo caso debería de indicarse la alternativa que se considere más adecuada o eficiente.

(2) **Guión** (*script*) de **creación y carga** del esquema de base de datos utilizado en la práctica **P2** (ver la 1ª norma de realización). El fichero debe llamarse **bdgxxxx-p3-esquema.sql**.

(3) **Guiones SQL** (*scripts*) **Oracle**. Ha de crearse un guión por cada ejercicio, llamado **bdgXXXX-p3-nn.sql** (nn es el número de ejercicio), que contendrá...

- El *número* de ejercicio y (quizá resumido) el *enunciado* del ejercicio.
- *Comentarios* y aclaraciones que se consideren oportunos.
- **Código PL/SQL** con las indentaciones adecuadas y fácilmente.
- Ejemplos que se consideren oportunos para comprobar el correcto funcionamiento del ejercicio.
- **NO DEBE APARECER** el resultado de la ejecución de las sentencias.

Criterios de evaluación

- Es obligatorio entregar vía **Aula Virtual** el informe de la práctica y todos los guiones (*scripts* SQL). No se corregirá la práctica si no se ha presentado la documentación completa (por ejemplo, si sólo se entregan algunos ejercicios y otros se dejan en blanco, o si se entregan sólo algunos *scripts*).
- La lectura del **informe** ha de ser **suficiente** para la comprensión total del trabajo realizado en la práctica (sin necesidad de recurrir a los guiones).
- Se valorará la inclusión de soluciones alternativas cuando haya lugar a ello y la justificación de las mismas.
- Se valorará el estilo en la resolución de cada ejercicio, las explicaciones dadas y la organización de la documentación explicativa (informe) de la práctica.

Es obligatorio respetar estrictamente las normas y el formato de presentación de la práctica, detallados en este documento