

**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 viernes 20 de mayo a las 23:55pm**. La **Memoria en PAPEL** se entregará en el casillero del profesor de prácticas. El **fichero comprimido con la Memoria y los Scripts SQL** a través de la Tarea del Aula Virtual denominada de forma similar a “Entrega de Práctica P3. Programación en PL/SQL”.

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** denominada `nombre_socio` que dado un DNI devuelva el nombre del socio que corresponda. Si no existe el DNI, debe devolver 'SOCIO INEXISTENTE'.
2. Escribir un **procedimiento** llamado `equipo_trabajador` tal que dado un DNI de un trabajador, muestre por pantalla su nombre, así como el conjunto de trabajadores contratados por la ONG de la que es responsable, indicando el DNI, nombre y si son voluntarios o no. Si el DNI no corresponde a ningún trabajador debe devolver 'TRABAJADOR INEXISTENTE'.

El formato de la salida sería este:

```
<Nombre del trabajador> ES RESPONSABLE DE <número de> TRABAJADORES:  
- <dni_1. Nombre vendedor_1. <si/no> ES VOLUNTARIO>.  
- <dni_2. Nombre vendedor_2. <si/no> ES VOLUNTARIO>.  
...  
- <dni_n. Nombre vendedor_n. <si/no> ES VOLUNTARIO>.
```

3. Escribir un **trigger** de base de datos denominado `T_ONG_0` tal que, cuando se introduzca un nuevo trabajador en una ONG, o se modifique la ONG de un trabajador existente, se compruebe que el trabajador no es socio de esa misma ONG. Nótese que es la implementación de una regla de integridad de la práctica P2 (RI2. Una persona no puede ser socio y trabajador de la misma ONG).

## Ejercicios

1. **Listar** por pantalla, en dos columnas, los nombres de todos los trabajadores y de todos los socios. 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 trabajadores y de socios. Tenga en cuenta que, lógicamente, puede haber distinto número de socios que de trabajadores. No hay que escribir procedimientos ni funciones.

TRABAJADORES	SOCIOS
-----	-----
<nombre trabajador_1>	<nombre socio_1>
...	...
<nombre trabajador_n>	...
	<nombre socio_m>
-----	-----
TOTAL: <n> TRABAJADORES Y <m> SOCIOS	

2. Hacer un **listado** que para cada (nombre de) ONG muestre sus trabajadores (nombre), *indicando cuál de ellos es el responsable* de la ONG, y para cada trabajador muestre los (identificadores de los) proyectos en los que coopera, indicando el recuento de acciones diferentes dentro de cada proyecto en las que participa.

Siga el formato que se indica a continuación. Note que el listado debe estar *ordenado alfabéticamente por nombres de ONG y trabajador*, y que también debe incluir todos los cálculos que se indican (número de trabajadores de cada ONG, en cuántos proyectos colabora cada trabajador en cada ONG, recuento de acciones proyecto/trabajador...).

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

```
ONG <nombre ong_1>
--- <número de> TRABAJADORES:
----- <nombre trabajador_1>
----- <número de> PROYECTOS:
----- <idproyecto_1>. ACCIONES: <recuento>
      ...
----- <idproyecto_n>. ACCIONES: <recuento>
      ...
----- <nombre trabajador_i> ES EL RESPONSABLE
----- <número de> PROYECTOS:
----- <idproyecto_1>. ACCIONES: <recuento>
      ...
----- <idproyecto_p>. ACCIONES: <recuento>
      ...
----- <nombre trabajador_t>
      ...
...
ONG <nombre ong_n>
...
```

3. Escribir un **package** llamado **P\_ONG** con el siguiente contenido:

- Una **función** llamada **quien\_es**, que dado un DNI devuelva
  - 1) '**TRABAJADOR/SOCIO**', si el DNI pertenece a un trabajador que también es socio de alguna ONG,
  - 2) '**TRABAJADOR DE** <nombre de su ONG>', si es un trabajador que no es socio de ninguna ONG,
  - 3) '**SOCIO DE** <tantas> **ONG**', si el DNI es de un socio que no es trabajador (sustituir <tantas> por el recuento de las ONG con las que colabora), y
  - 4) '**PERSONA NO ENCONTRADA**', si el DNI no corresponde a ningún socio ni trabajador.

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 con cuántas ONG colabora un socio en el caso 3).

- Un **procedimiento** llamado **cooperacion** tal que dado un DNI muestre por pantalla lo siguiente:  
Si es socio, el nombre del socio y un listado de las ONG con las que colabora.  
Si es trabajador, su nombre, la ONG, su responsable, en cuántos proyectos coopera, y si es o no voluntario.  
Siga el formato indicado a continuación para la salida por pantalla.

```
TRABAJADOR <nombre trabajador>
ONG <nombre ong>.
CON RESPONSABLE <nombre responsable de la ONG>.
PARTICIPA EN <número de> PROYECTOS.
<ES/NO ES> VOLUNTARIO.

SOCIO <nombre socio>
DE <número de> ONG:
- ONG:<nombre>; CUOTA:<cuota> €.
...
- ONG:<nombre>; CUOTA:<cuota> €.
```

4. Escribir los **triggers** de base de datos necesarios para imponer que un trabajador sólo puede participar en las acciones correspondientes a proyectos puestos en marcha por la ONG en la que está contratado.

Tenga en cuenta que un trabajador podría cambiar de ONG, lo que es aceptable sólo si no participa ya en ninguna acción/proyecto.

También podría modificarse el trabajador en una participación ya registrada.

Importante: Asegúrese de programar el código de cada trigger de modo que cubra todas las posibilidades de ejecución.

Denomine los disparadores siguiendo este formato: T\_ONG\_1, T\_ONG\_2, ... T\_ONG\_n.

Nótese que este ejercicio es la implementación de una regla de integridad de la práctica P2 (RI1. Un trabajador sólo puede participar en proyectos de su ONG).

## Documentación que se debe entregar

La entrega se realizará antes de la fecha límite indicada. Hay que entregarlo por dos vías:

- 1) **Memoria en PAPEL**, en el casillero de Conserjería del profesor de prácticas (grapada, **no encuadernada**).
- 2) Un **FICHERO comprimido con todos los documentos (Memoria + Scripts SQL)**, con el nombre **bdgxxxx-p3 (.zip o .rar)**, a través de una **TAREA** del Aula Virtual.

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 (**20nn/nn**) 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 previa **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 (memoria)** 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**