

## Enunciado de la práctica juvenil de PL/SQL

### Ejercicio 1

Partiendo del siguiente esquema relacional, crea las tablas e introduce algunos registros de prueba:

SOCIOS		
<u>DNI</u>	VARCHAR2, tamaño 10	No Nulo
Nombre	VARCHAR2, tamaño 20	No Nulo
Direccion	VARCHAR2, tamaño 20	
Penalizaciones	NUMBER de 2	Por defecto, 0

LIBROS		
<u>RefLibro</u>	VARCHAR2, tamaño 10	No Nulo
Nombre	VARCHAR2, tamaño 30	No Nulo
Autor	VARCHAR2, tamaño 20	No Nulo
Genero	VARCHAR2, tamaño 10	
AñoPublicación	NUMBER	
Editorial	VARCHAR2, tamaño 10	

PRESTAMOS		
<u>DNI</u>	VARCHAR2, tamaño 10	No Nulo
<u>RefLibro</u>	VARCHAR2, tamaño 10	No Nulo
<u>FechaPrestamo</u>	FECHA	No Nulo
Duración	NUMERICO de 2	Por defecto, 24

Realiza un procedimiento llamado *listadocuatromasprestados* que nos muestre por pantalla un listado de los cuatro libros más prestados y los socios a los que han sido prestados con el siguiente formato:

```
NombreLibro1 NumPrestamosLibro1 GeneroLibro1
      DNISocio1      FechaPrestamoalSocio1
...
      DNISocion      FechaPrestamoalSocion

NombreLibro2 NumPrestamosLibro2      GeneroLibro2
      DNISocio1      FechaPrestamoalSocio1
...
      DNISocion      FechaPrestamoalSocion

NombreLibro3 NumPrestamosLibro3      GeneroLibro3
      DNISocio1      FechaPrestamoalSocio1
...
      DNISocion      FechaPrestamoalSocion

NombreLibro4 NumPrestamosLibro4      GeneroLibro4
      DNISocio1      FechaPrestamoalSocio1
...
      DNISocion      FechaPrestamoalSocion
```

El procedimiento debe gestionar adecuadamente las siguientes excepciones:

- La tabla Libros está vacía.
- La tabla Socios está vacía.
- Hay menos de cuatro libros que hayan sido prestados.

## **Ejercicio 2**

Partiendo del siguiente script, crea la BD correspondiente a los alumnos matriculados en algunos de los módulos de 1º y 2º curso del CFS y sus correspondientes notas:

```
REM ***** TABLAS ALUMNOS, ASIGNATURAS, NOTAS: *****

DROP TABLE ALUMNOS cascade constraints;

CREATE TABLE ALUMNOS
(
  DNI VARCHAR2(10) NOT NULL,
  APENOM VARCHAR2(30),
  DIREC VARCHAR2(30),
  POBLA VARCHAR2(15),
  TELEF VARCHAR2(10)
) TABLESPACE USER_DATA;

DROP TABLE ASIGNATURAS cascade constraints;

CREATE TABLE ASIGNATURAS
(
  COD NUMBER(2) NOT NULL,
  NOMBRE VARCHAR2(25)
) TABLESPACE USER_DATA;

DROP TABLE NOTAS cascade constraints;

CREATE TABLE NOTAS
(
  DNI VARCHAR2(10) NOT NULL,
  COD NUMBER(2) NOT NULL,
  NOTA NUMBER(2)
) TABLESPACE USER_DATA;

INSERT INTO ASIGNATURAS VALUES (1,'Prog. Leng. Estr. ');
INSERT INTO ASIGNATURAS VALUES (2,'Sist. Informáticos ');
INSERT INTO ASIGNATURAS VALUES (3,'Análisis ');
INSERT INTO ASIGNATURAS VALUES (4,'FOL ');
INSERT INTO ASIGNATURAS VALUES (5,'RET ');
INSERT INTO ASIGNATURAS VALUES (6,'Entornos Gráficos ');
INSERT INTO ASIGNATURAS VALUES (7,'Aplic. Entornos 4ªGen ');

INSERT INTO ALUMNOS VALUES
('12344345','Alcalde García, Elena', 'C/Las Matas, 24','Madrid','917766545');

INSERT INTO ALUMNOS VALUES
('4448242','Cerrato Vela, Luis', 'C/Mina 28 - 3A', 'Madrid','916566545');

INSERT INTO ALUMNOS VALUES
('56882942','Díaz Fernández, María', 'C/Luis Vives 25',
'Móstoles','915577545');

INSERT INTO NOTAS VALUES('12344345', 1,6);
INSERT INTO NOTAS VALUES('12344345', 2,5);
INSERT INTO NOTAS VALUES('12344345', 3,6);

INSERT INTO NOTAS VALUES('4448242', 4,6);
INSERT INTO NOTAS VALUES('4448242', 5,8);
INSERT INTO NOTAS VALUES('4448242', 6,4);
INSERT INTO NOTAS VALUES('4448242', 7,5);

INSERT INTO NOTAS VALUES('56882942', 5,7);
INSERT INTO NOTAS VALUES('56882942', 6,8);
INSERT INTO NOTAS VALUES('56882942', 7,9);
```

COMMIT;

Diseña un procedimiento al que pasemos como parámetro de entrada el nombre de uno de los módulos existentes en la BD y visualice el nombre de los alumnos que lo han cursado junto a su nota.

Al final del listado debe aparecer el nº de suspensos, aprobados, notables y sobresalientes.

Asimismo, deben aparecer al final los nombres y notas de los alumnos que tengan la nota más alta y la más baja.

Debes comprobar que las tablas tengan almacenada información y que exista el módulo cuyo nombre pasamos como parámetro al procedimiento.

### **Ejercicio 3**

A partir de las tablas creadas con el siguiente script:

```
DROP TABLE productos CASCADE CONSTRAINTS;
```

```
CREATE TABLE productos
(
    CodProducto  VARCHAR2(10) CONSTRAINT p_cod_no_nulo NOT NULL,
    Nombre       VARCHAR2(20) CONSTRAINT p_nom_no_nulo NOT NULL,
    LineaProducto VARCHAR2(10),
    PrecioUnitario NUMBER(6),
    Stock        NUMBER(5),
    PRIMARY KEY (CodProducto)
);
```

```
DROP TABLE ventas CASCADE CONSTRAINTS;
```

```
CREATE TABLE ventas
(
    CodVenta      VARCHAR2(10) CONSTRAINT cod_no_nula NOT NULL,
    CodProducto   VARCHAR2(10) CONSTRAINT pro_no_nulo NOT NULL,
    FechaVenta    DATE,
    UnidadesVendidas NUMBER(3),
    PRIMARY KEY (CodVenta)
);
```

```
INSERT INTO productos VALUES ('1','Procesador P133', 'Proc',15000,20);
INSERT INTO productos VALUES ('2','Placa base VX', 'PB', 18000,15);
INSERT INTO productos VALUES ('3','Simm EDO 16Mb', 'Memo', 7000,30);
INSERT INTO productos VALUES ('4','Disco SCSI 4Gb', 'Disc',38000, 5);
INSERT INTO productos VALUES ('5','Procesador K6-2', 'Proc',18500,10);
INSERT INTO productos VALUES ('6','Disco IDE 2.5Gb', 'Disc',20000,25);
INSERT INTO productos VALUES ('7','Procesador MMX', 'Proc',15000, 5);
INSERT INTO productos VALUES ('8','Placa Base Atlas', 'PB', 12000, 3);
INSERT INTO productos VALUES ('9','DIMM SDRAM 32Mb', 'Memo',17000,12);
```

```
INSERT INTO ventas VALUES('V1', '2', '22/09/97',2);
INSERT INTO ventas VALUES('V2', '4', '22/09/97',1);
INSERT INTO ventas VALUES('V3', '6', '23/09/97',3);
INSERT INTO ventas VALUES('V4', '5', '26/09/97',5);
INSERT INTO ventas VALUES('V5', '9', '28/09/97',3);
INSERT INTO ventas VALUES('V6', '4', '28/09/97',1);
INSERT INTO ventas VALUES('V7', '6', '02/10/97',2);
INSERT INTO ventas VALUES('V8', '6', '02/10/97',1);
INSERT INTO ventas VALUES('V9', '2', '04/10/97',4);
INSERT INTO ventas VALUES('V10', '9', '04/10/97',4);
INSERT INTO ventas VALUES('V11', '6', '05/10/97',2);
```

```

INSERT INTO ventas VALUES('V12','7', '07/10/97',1);
INSERT INTO ventas VALUES('V13','4', '10/10/97',3);
INSERT INTO ventas VALUES('V14','4', '16/10/97',2);
INSERT INTO ventas VALUES('V15','3', '18/10/97',3);
INSERT INTO ventas VALUES('V16','4', '18/10/97',5);
INSERT INTO ventas VALUES('V17','6', '22/10/97',2);
INSERT INTO ventas VALUES('V18','6', '02/11/97',2);
INSERT INTO ventas VALUES('V19','2', '04/11/97',3);
INSERT INTO ventas VALUES('V20','9', '04/12/97',3);

```

a) Realiza un procedimiento que actualice la columna Stock de la tabla Productos a partir de los registros de la tabla Ventas.

El citado procedimiento debe informarnos por pantalla si alguna de las tablas está vacía o si el stock de un producto pasa a ser negativo, en cuyo caso se parará la actualización.

b) Realiza un procedimiento que presente por pantalla un listado de las ventas con el siguiente formato:

Linea Producto: *NombreLinea1*

<i>Prod11</i>	<i>UnidadesTotales1</i>	<i>ImporteTotal1</i>
<i>Prod12</i>	<i>UnidadesTotales2</i>	<i>ImporteTotal2</i>
...		
<i>Prod1n</i>	<i>UnidadesTotalesn</i>	<i>ImporteTotaln</i>

Importe total *NombreLinea1*: ImporteLinea1

Linea Producto: *NombreLinea2*

<i>Prod21</i>	<i>UnidadesTotales1</i>	<i>ImporteTotal1</i>
<i>Prod22</i>	<i>UnidadesTotales2</i>	<i>ImporteTotal2</i>
...		
<i>Prod2n</i>	<i>UnidadesTotalesn</i>	<i>ImporteTotaln</i>

Importe total *NombreLinea2*: ImporteLinea2

.  
.  
.

Total Ventas: *Importedetodaslaslineas*