Índice

Objetivo de la práctica

En esta práctica el alumno utilizará las funcionalidades de PLSQL para automatizar algunas operaciones y para realizar comprobaciones sobre los datos. Estas operaciones y comprobaciones no pueden realizarse solo con SQL.

La última versión de esta práctica está disponible en este enlace.

Descripción del problema

Una compañía necesita automatizar su almacén

- De cada producto se almacena su identificador (máximo 10 dígitos), su nombre (como máximo, 30 caracteres) y su *stock*.
- En cada entrada de producto al almacén, se apunta
  - El producto
  - La cantidad de producto (como máximo, 9999 unidades)
  - El precio pagado al proveedor por unidad de producto (como máximo, 99999€)
- De cada salida de producto del almacén se apunta
  - El producto, cantidad de producto y precio por unidad que paga el cliente
- Será necesario saber a qué precio se realizó la última compra y la última venta de cada producto

Ejercicio 1 : Definir el modelo de datos

Crea las tablas y funciones necesarias para soportar la descripción del problema. Las tablas y atributos concretos no son importantes, ya que el profesor consultará los datos a partir de las vistas definidas más adelante.

Ejercicio 2 : Procedimiento para crear un producto (2 puntos)

Crea un procedimiento de nombre `CREAR_PRODUCTO` (listado 1), que inserte un nuevo producto en la base de datos. El identificador de los productos podrá basarse en la secuencia `SECUENCIA_PRODUCTO_ID`, o usar otro método.

```
1 create or replace procedure CREAR_PRODUCTO(nombreproducto IN varchar, idproducto
    ↪ OUT number)
2 as
3   -- VARIABLES QUE HAGAN FALTA
4 begin
5   -- CONSIGUE EL NUEVO ID
6   -- INSERTA EL PRODUCTO CON ESE ID
7   -- DEVUELVO EL NUEVO ID EN idproducto
8 end;
9 /
```

Listado 1: Creación del procedimiento `CREAR_PRODUCTO`

El procedimiento devolverá en su segundo parámetro el identificador del producto recién creado.

Crea una vista de nombre V_PRODUCTOS para que el profesor pueda consultar los productos en el catálogo (listado 2)

```
1  create or replace view V_PRODUCTOS(nombreproducto,idproducto) as
2  ...
```

Listado 2: Creación de la vista V_PRODUCTOS

Ejercicio 3 : Entrada y salida de productos (3 puntos)

- Crea un procedimiento ENTRADA_PRODUCTO que actualice la base de datos cuando llegue un producto.
- Crea un procedimiento SALIDA_PRODUCTO que actualice la base de datos cuando se envie un producto.
- Si el producto no existe, se lanzará el error -20102.
- Los precios pagados y cobrados son por unidad de producto.

```
1  create or replace procedure ENTRADA_PRODUCTO(
2    idproducto IN number,
3    cantidad IN number,
4    preciopagadoporunidad IN number)
5  as
6    -- VARIABLES QUE HAGAN FALTA
7  begin
8    ...
9  end;
10
11 create or replace procedure SALIDA_PRODUCTO(
12   idproducto IN number,
13   cantidad IN number,
14   preciocobradoporunidad IN number)
15 as
16   -- VARIABLES QUE HAGAN FALTA
17 begin
18   ...
19 end;
```

Listado 3: Entrada y salida de productos

Para poder corregirse, debe existir la función EXISTENCIAS_PRODUCTO que informe del *stock* de un producto. Un producto que no ha tenido ni entradas ni salidas tiene un *stock* de cero. Un producto que no existe tiene *stock* -1.

```
1  create or replace function EXISTENCIAS_PRODUCTO(idproducto IN number) return number
2  as
3  begin
4    ...
5  end;
```

Listado 4: Función de existencias

Ejercicio 4 : Vista de existencias (3 puntos)

Crea la vista `V_EXISTENCIAS`. En esta vista se listan todos los productos existentes y su stock. Un producto que nunca ha tenido entradas o salidas debe tener un *stock* de cero. Un producto que no se ha comprado nunca a un proveedor tiene un `ultimopreciocompra` a `NULL`. Un producto que nunca se ha vendido a un cliente tiene `ultimoprecioventa` a `NULL`. Los precios son por unidad de producto.

```
1 create or replace view V_EXISTENCIAS(idproducto,existencias,ultimopreciocompra,
    ↪ ultimoprecioventa) as
2 ...
```

Listado 5: Vista de existencias

Ejercicio 5 : Función de precio medio de venta (3 puntos)

Crea la función `PRECIO_MEDIO_VENTA`. Esta función devolverá:

- `null` si el producto no se ha vendido nunca
- El precio medio de venta, que se calcula como la suma de los precios de cada una de las unidades vendidas, dividido por el número total de las unidades vendidas. El precio medio tendrá dos decimales.
  - Ejemplo: si se han vendido 3 unidades a 10€ y 2 unidades a 5€ el precio medio es de 8€
- Lanzará el error `-20102` si el producto no existe

```
1 create or replace function PRECIO_MEDIO_VENTA(idproducto number) return number
2 as
3 ...
```

Listado 6: Precio medio de venta

Ejercicio 6 : Salida del almacén respetando existencias (2 puntos)

Crea un procedimento `SALIDA_PRODUCTO_CON_STOCK`. Realizará el mismo proceso que `SALIDA_PRODUCTO`, pero en el caso de que no haya existencias suficientes lanzará un error con el mensaje `Rotura de stock` y número `-20101`. Si el producto no existe, se lanzará el error `-20102`.

```
1  create or replace procedure SALIDA_PRODUCTO_CON_STOCK(
2    idproducto IN number,
3    cantidad IN number,
4    preciocobrado IN number
5  as
6  begin
7    ...
8    RAISE_APPLICATION_ERROR(-20101,'Rotura de stock');
9    ...
10 end;
```

Listado 7: Entrada y salida de productos

Entorno de pruebas

Los procedimientos y funciones se crearán sobre el servidor del instituto, con el usuario de cada alumno. El profesor pasará unas pruebas automáticas, que dejarán el resultado en http://10.1.33.201:80. Para acceder a los resultados desde casa, se puede utilizar un túnel ssh como sigue, y conectar un navegador a http://localhost:8080 (y otro al servidor de Oracle por el puerto 1521)

```
ssh -L 8080:localhost:80 -L 1521:localhost:1521 USUARIO@213.97.191.51 -p 10133
```

Instrucciones de entrega

La entrega se realizará en el servidor de pruebas.

Si no estuviera operativo, se entregará un único fichero SQL para todos los apartados con las sentencias SQL necesarias para crear las tablas, secuencias, procedimientos, funciones y vistas que el alumno necesite.

- Este fichero se corregirá de forma semiautomática, por lo que es necesario seguir la nomenclatura propuesta en el ejercicio.
- El fichero se cargará en un usuario recién creado con permisos necesarios para crear todos los elementos necesarios (tablas, vistas, funciones, secuencias...)
- Si tiene **errores** de compilación podría no corregirse. Si no se siguen los **nombres de objetos** pedidos podría no corregirse.

Sube el documento a la tarea correspondiente en el aula virtual. Presta atención al plazo de entrega (con fecha y hora).