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.

Modelo de datos

Los ejercicios se realizarán sobre las tablas y datos del ejemplo de jardineria

Ejercicio 1 : Procedimiento para contratar un becario (3 puntos)

Crea un procedimiento de nombre `CONTRATA_BECARIO` (listado 1). El procedimiento recibirá el identificador de un empleado ya existente que será el jefe del nuevo becario, y el nombre y apellidos del nuevo becario.

- La extensión, email, oficina... se copiarán del jefe del becario, ya que al ser becario, no tiene ventajas como un email propio (50 %)
- El código de empleado se calculará dentro del procedimiento, como el `nextval` de una secuencia, y se devolverá en un parámetro de salida (50 %). Si esto no funciona no se podrá corregir el ejercicio.

```
create sequence SECUENCIA_EMPLEADOS;
create or replace procedure CONTRATA_BECARIO(nombre IN varchar, apellido1 IN
    ↪ varchar, apellido2 IN varchar, codigojefe IN integer, codigobecario OUT
    ↪ integer)
as
  -- VARIABLES QUE HAGAN FALTA
begin
  -- CONSIGUE EL NUEVO ID
  -- INSERTA EL BECARIO CON ESE ID
  -- DEVUELVO EL NUEVO ID EN codigobecario
end;
/
```

Listado 1: Creación del procedimiento `CONTRATA_BECARIO`

Ejercicio 2 : Venta de productos (30 %)

- Crea un procedimiento `AGREGA_DETALLE_PEDIDO` que añada una línea de pedido a un pedido ya existente.
- El precio del producto será el precio de venta en la tabla `PRODUCTOS` (20 %)
- El número de línea no es importante, se puede poner siempre a `1`.
- El procedimiento se encargará de eliminar del *stock* del producto las cantidades vendidas (30 %)
- Si el *stock* no es suficiente, lanzará el error `-20001` (50 %)

Aviso

Este sistema de control de *stock* no funcionaría si más de un usuario estuviera conectado a la vez a la base de datos. Se necesitaría un *isolation level* adecuado, pero queda fuera de esta práctica.

```
create or replace procedure AGREGA_DETALLE_PEDIDO(
  codigopedido IN integer
  codigoproducto IN varchar,
  cantidad IN number)
as
  -- VARIABLES QUE HAGAN FALTA
begin
  ...
  if ..... then
    RAISE_APPLICATION_ERROR(-20001,'Rotura de stock');
  end if;
  ...
end;
```

Listado 2: Añadir un producto a un pedido

Ejercicio 3 : Registro de pagos (4 puntos)

Crea un procedimiento para registrar los pagos de los clientes

- La fecha del pago será el momento de la ejecución (`SYSDATE`) (20 %)
- El identificador de la transacción será `PAGOAUTOMATICO-XXXX`, siendo `XXXX` una secuencia. El identificador se devolverá en un parámetro `out` (30 %). Si no funciona este parámetro no se podrá corregir el ejercicio.
- Si el pago no es necesario, porque el cliente no debe tanto dinero, se lanzará el error `-20002`. Para saber el saldo de un cliente se calculará su facturación y se restará el total de pagos. (50 %)

```
create sequence SECUENCIA_PAGOS;
create or replace procedure REGISTRA_PAGO(
  codigocliente IN integer,
  formapago IN varchar,
  cantidad IN number,
  idtransaccion OUT varchar
)
as
  -- VARIABLES QUE HAGAN FALTA
begin
  ...
  if ..... then
    RAISE_APPLICATION_ERROR(-20002,'Pago innecesario');
  end if;
  ...
end;
```

Listado 3: Registro de pagos

Instrucciones de entrega

Se creará un único fichero SQL para todos los apartados.

- 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 las tablas y datos de jardinería, y con permisos necesarios para crear todos los elementos necesarios (tablas, vistas, funciones, secuencias...)
- Además de los procedimientos y secuencias pedidos, pueden crearse vistas o funciones adicionales para facilitar los ejercicios.
- 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).