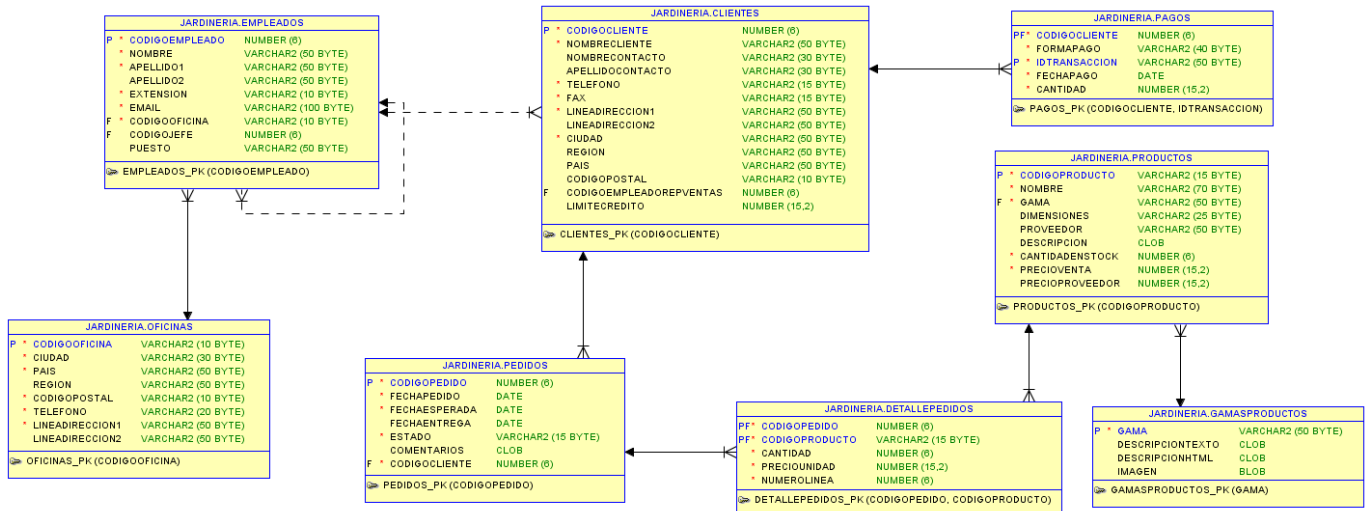


EJERCICIOS JDBC

Dada la BD jardinería, cuyo modelo de datos se representa en la figura:



EJERCICIOS:

- Realiza un **método java** que reciba los datos de un empleado y los inserte en la tabla EMPLEADOS. Los datos a recibir son el nombre, apellido1, apellido2, extensión, email, **codigooficina**, **codigojefe**, y puesto. El **código de empleado será una unidad más**, al código de empleado máximo almacenado en la tabla. Al insertar comprobar que exista el código de oficina y el código de jefe. Visualizar todos los mensajes de error que ocurran, e indicar si el registro se ha insertado o no.
- Realiza un **método java** que reciba un código de cliente, y visualice los pedidos de ese cliente, con los totales por cada pedido, y el pedido con total importe máximo, y el producto más comprado. Visualizar cada pedido ordenado por número de línea. Visualizar si el cliente no tiene pedidos o si el cliente no existe en la BD. La salida a visualizar es la siguiente:

COD-CLIENTE: xxxxxxxx NOMBRE: xxxxxxxxxxxxxxxx
DIRECCIÓN1: xxxxxxxxxxxxxxxx Número de pedidos:xxxxxxx

```

COD-PEDIDO: xxxxxxxx    FECHA PEDIDO: DD/MM/YYYY    ESTADO DEL PEDIDO: xxxxxxxxxxxx
NUM-LINEA    COD-PROD    NOMBRE PRODUCTO    CANTIDAD    PREC-UNID    IMPORTE
-----
          999    xxxxxxxx    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    9.999    9.999,99    99.999,99
          999    xxxxxxxx    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    9.999    9.999,99    99.999,99
-----
TOTALES POR PEDIDO    99.999    99.999,99    999.999,99
    
```

```

COD-PEDIDO: xxxxxxxx    FECHA PEDIDO: DD/MM/YYYY    ESTADO DEL PEDIDO: xxxxxxxxxxxx
NUM-LINEA    COD-PROD    NOMBRE PRODUCTO    CANTIDAD    PREC-UNID    IMPORTE
-----
          999    xxxxxxxx    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    9.999    9.999,99    99.999,99
          999    xxxxxxxx    xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    9.999    9.999,99    99.999,99
-----
TOTALES POR PEDIDO    99.999    99.999,99    999.999,99
    
```

COD de PEDIDO y FECHA PEDIDO CON TOTAL IMPORTE MÁXIMO: xxxxxxxxxxxxxxxxxxxxxxxx
COD PRODUCTO y NOMBRE PERODUCTO, del producto más comprado(Cantidad Máxima): xxxxxxxxxxxxxxxxxxxxxxxx

3. En la tabla CLIENTES hay clientes que no tienen pedidos. Se desea hacer un método para eliminar los clientes que no han realizado pedidos. Los clientes borrados se deben de añadir a una tabla que el método tiene que crear. Esta tabla se llamará **CLIENTESSINPEDIDO**, tendrá la misma estructura que la tabla CLIENTES.

El método a **crear deberá crear la tabla CLIENTESSINPEDIDO**, añadir a esa tabla los clientes sin pedidos, y eliminar de la tabla CLIENTES esos mismos clientes.

El método se podrá ejecutar tantas veces como se desea. Hay que controlar todos los errores posibles.

4. Los empleados de la empresa se encargan de atender a los clientes. Y cada empleado tiene asignado o no, a uno o más clientes.

Se desea almacenar en la tabla EMPLEADOS el número de clientes a los que atiende.

Para ello se pide realizar un método para **añadir la columna NUMCLIENTES** de tipo numérica a la tabla EMPLEADOS. Una vez añadida la columna se debe actualizar esa tabla para almacenar en esa columna el número de clientes que tiene asignado el empleado.

El método se podrá ejecutar tantas veces como se desea. Hay que controlar todos los errores posibles.

5. Se desea actualizar la cantidad en stock de los productos (CANTIDADENSTOCK), **para ello se añadirá la columna STOCKACTUALIZADO a la tabla productos**, con el mismo formato que CANTIDADENSTOCK. En esa columna se almacenará la actualización que será igual a la CANTIDADENSTOCK menos la suma de la CANTIDAD vendida del producto.

Para ello se pide realizar un método que añada la columna STOCKACTUALIZADO, y la actualice.

Obtener por pantalla aquellos productos que se necesiten reponer, que será los que tengan en el STOCKACTUALIZADO una cantidad menor de 5. Visualizar su código, la cantidad en stock, y su stock actualizado.

6. Disponemos de una función almacenada que nos devuelve los datos de la oficina y el número de empleados que tiene. La función recibe el código de la oficina y devuelve el número de empleados, y además devuelve en parámetros de salida la ciudad, el país, la región, y la dirección de la oficina.

La función devuelve -1 si no existe en la BD, y en este caso, además en ciudad y país devuelve NO EXISTE OFICINA.

La función es la siguiente:

```
CREATE OR REPLACE
FUNCTION veroficina( cod VARCHAR2, ciudad OUT VARCHAR2 ,
    pais OUT VARCHAR2, region OUT VARCHAR2 , direcc OUT VARCHAR2)
RETURN NUMBER
AS
    cuenta NUMBER:= -1;
BEGIN
    SELECT COUNT(*) INTO cuenta FROM empleados WHERE codigooficina = cod;
    IF (cuenta=0) THEN
        ciudad := 'NO EXISTE OFICINA';
        pais := 'NO EXISTE OFICINA';
    ELSE
        SELECT ciudad, pais, region, lineadireccion1
        INTO ciudad, pais, region, direcc
        FROM oficinas WHERE codigooficina = cod;
    END IF;
    RETURN CUENTA;
END;
```

Se pide realizar la siguiente salida por consola haciendo uso de la función almacenada:

EJERCICIOS JDBC

COD OFICINA	CIUDAD	PAIS	REGION	DIRECCION1	NUM EMPLES
XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX

7. Haciendo uso del método realizado en el ejercicio 2, realiza un método java para obtener el los datos de los pedidos de todos los clientes. La salida será similar a la que se muestra:

COD-CLIENTE: xxxxxxxx NOMBRE: xxxxxxxxxxxxxxxx
DIRECCIÓN1: xxxxxxxxxxxxxxxx Número de pedidos:xxxxxxx

COD-PEDIDO: xxxxxxxx					
FECHA PEDIDO: xxxxxxxxxxxxxx					
ESTADO DEL PEDIDO: xxxxxxxxxxxxxx					
NUM-LINEA	COD-PROD	NOMBRE PRODUCTO	CANTIDAD	PREC-UNID	IMPORTE
999	xxxxxxxx	xx	9.999	9.999,99	99.999,99
999	xxxxxxxx	xx	9.999	9.999,99	99.999,99
-----			-----	-----	-----
TOTALES POR PEDIDO			99.999	99.999,99	999.999,99

COD-PEDIDO: xxxxxxxx		FECHA PEDIDO: xxxxxxxxxxxx		ESTADO DEL PEDIDO: xxxxxxxxxxxx		
NUM-LINEA	COD-PROD	NOMBRE PRODUCTO	CANTIDAD	PREC-UNID	IMPORTE	
999	xxxxxxxx	xx	9.999	9.999,99	99.999,99	
999	xxxxxxxx	xx	9.999	9.999,99	99.999,99	
TOTALES POR PEDIDO			99.999	99.999,99	999.999,99	

COD de PEDIDO y FECHA PEDIDO CON TOTAL IMPORTE MÁXIMO: xxxxxxxxxxxxxxxxxxxxxxxx
COD PRODUCTO y NOMBRE PRODUCTO, del producto más comprado(Cantidad Máxima): xxxxxxxxxxxxxxxxxxxxxxxx

COD-CLIENTE: xxxxxxxx NOMBRE: xxxxxxxxxxxxxxxx
DIRECCIÓN1: xxxxxxxxxxxxxxxx Número de pedidos:xxxxxxx

COD-PEDIDO: xxxxxxxx		FECHA PEDIDO: xxxxxxxxxxxx		ESTADO DEL PEDIDO: xxxxxxxxxxxx		
NUM-LINEA	COD-PROD	NOMBRE PRODUCTO	CANTIDAD	PREC-UNID	IMPORTE	
999	xxxxxxxx	xx	9.999	9.999,99	99.999,99	
999	xxxxxxxx	xx	9.999	9.999,99	99.999,99	
TOTALES POR PEDIDO			99.999	99.999,99	999.999,99	

COD-PEDIDO:	xxxxxxx	FECHA PEDIDO:	xxxxxxxxxx	ESTADO DEL PEDIDO:	xxxxxxxxxx
NUM-LINEA	COD-PROD	NOMBRE PRODUCTO	CANTIDAD	PREC-UNID	IMPORTE
999	xxxxxxx	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	9.999	9.999,99	99.999,99
999	xxxxxxx	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	9.999	9.999,99	99.999,99
TOTALES POR PEDIDO			99.999	99.999,99	999.999,99

COD de PEDIDO y FECHA PEDIDO CON TOTAL IMPORTE MÁXIMO: xxxxxxxxxxxxxxxxxxxxxxxx
COD PRODUCTO y NOMBRE PRODUCTO, del producto más comprado(Cantidad Máxima): xxxxxxxxxxxxxxxxxxxxxxxx

8. Dada la tabla **NUEVOSEMPLEADOS**, que contiene información acerca de nuevos empleados, y de la actualización de datos de otros que ya existen. Se pide leer los datos de la tabla, comprobar los datos de cada registro y modificar los registros existentes, e insertar los registros nuevos, siempre y cuando no se encuentre ningún error.

Si hay errores en los datos ni se inserta ni se actualiza, y se visualizan los mensajes de error ocurridos.

A la hora de actualizar hay que visualizar TODOS LOS DATOS que se van a actualizar, respecto a la información almacenada en el registro.

A la hora de insertar hay que comprobar que el código de empleado no exista, y que el jefe y la oficina si existan. Hay que visualizar todos los errores.

