

Según el modelo dado resuelva:

1. Hacer una función que dado un artículo y un depósito devuelva un string que indique el estado del depósito según el artículo. Si la cantidad almacenada es menor al límite retornar "OCUPACION DEL DEPOSITO XX %" siendo XX el % de ocupación. Si la cantidad almacenada es mayor o igual al límite retornar "DEPOSITO COMPLETO".
2. Realizar una función que dado un artículo y una fecha, retorne el stock que existía a esa fecha.
3. Cree el/los objetos de base de datos necesarios para corregir la tabla empleado en caso que sea necesario.  
Se sabe que debería existir un único gerente general (debería ser el único empleado sin jefe). Si detecta que hay más de un empleado sin jefe deberá elegir entre ellos el gerente general, el cual será seleccionado por mayor salario. Si hay más de uno se seleccionará el de mayor antigüedad en la empresa.  
Al finalizar la ejecución del objeto la tabla deberá cumplir con la regla de un único empleado sin jefe (el gerente general) y deberá retornar la cantidad de empleados que había sin jefe antes de la ejecución.
4. Cree el/los objetos de base de datos necesarios para actualizar la columna de empleado empl\_comision con la sumatoria del total de lo vendido por ese empleado a lo largo del último año. Se deberá retornar el código del vendedor que más vendió (en monto) a lo largo del último año.
5. Realizar un procedimiento que complete con los datos existentes en el modelo provisto la tabla de hechos denominada Fact\_table tiene la siguiente definición:

**Create table Fact\_table**

**( anio char(4),  
mes char(2),  
familia char(3),  
rubro char(4),  
zona char(3),  
cliente char(6),  
producto char(8),  
cantidad decimal(12,2),  
monto decimal(12,2)  
)**

**Alter table Fact\_table**

**Add constraint primary key(anio,mes,familia,rubro,zona,cliente,producto)**

6. Realizar un procedimiento que si en alguna factura se facturaron componentes que conforman un combo determinado (o sea que juntos componen otro producto de mayor nivel), en cuyo caso deberá reemplazar las filas correspondientes a dichos

productos por una sola fila con el producto que componen con la cantidad de dicho producto que corresponda.

7. Hacer un procedimiento que dadas dos fechas complete la tabla Ventas. Debe insertar una línea por cada artículo con los movimientos de stock realizados entre esas fechas. La tabla se encuentra creada y vacía.

#### VENTAS

Código	Detalle	Cant. Mov.	Precio de Venta	Renglón	Ganancia
Código del articulo	Detalle del articulo	Cantidad de movimientos de ventas (Item factura)	Precio promedio de venta	Nro. de línea de la tabla	Precio de venta – cantidad * costo actual

8. Realizar un procedimiento que complete la tabla Diferencias de precios, para los productos facturados que tengan composición y en los cuales el precio de facturación sea diferente al precio del cálculo de los precios unitarios por cantidad de sus componentes, se aclara que un producto que compone a otro, también puede estar compuesto por otros y así sucesivamente, la tabla se debe crear y esta formada por las siguientes columnas:

#### DIFERENCIAS

Código	Detalle	Cantidad	Precio_generado	Precio_facturado
Código del articulo	Detalle del articulo	Cantidad de productos que conforman el combo	Precio que se compone a través de sus componentes	Precio del producto

9. Hacer un trigger que ante alguna modificación de un ítem de factura de un artículo con composición realice el movimiento de sus correspondientes componentes.
10. Hacer un trigger que ante el intento de borrar un artículo verifique que no exista stock y si es así lo borre en caso contrario que emita un mensaje de error.
11. Cree el/los objetos de base de datos necesarios para que dado un código de empleado se retorne la cantidad de empleados que este tiene a su cargo (directa o indirectamente). Solo contar aquellos empleados (directos o indirectos) que sean menores que su jefe directo.
12. Cree el/los objetos de base de datos necesarios para implantar la siguiente regla “Ningún jefe puede tener a su cargo más de 50 empleados en total (directos + indirectos)”. Se sabe que en la actualidad dicha regla se cumple y que la base de datos es accedida por n aplicaciones de diferentes tipos y tecnologías.

13. Cree el/los objetos de base de datos necesarios para que nunca un producto pueda ser compuesto por sí mismo.

Se sabe que en la actualidad dicha regla se cumple y que la base de datos es accedida por n aplicaciones de diferentes tipos y tecnologías.

No se conoce la cantidad de niveles de composición existentes.

14. Cree el/los objetos de base de datos necesarios para implantar la siguiente regla

“Ningún jefe puede tener un salario mayor al 20% de la suma de los salarios de sus empleados totales (directos + indirectos)”. Se sabe que en la actualidad dicha regla se cumple y que la base de datos es accedida por n aplicaciones de diferentes tipos y tecnologías