

1. Realice lo siguiente
 - 1.1 CREAR una tabla PEDIDOS con los siguientes campos:
 - ID artículo
 - Cantidad Pedida,
 - Fecha_pedido
 - 1.2 Desarrollar los triggers que se indican a continuación:
 - Un trigger que se disparará DESPUÉS de INSERTAR, MODIFICAR o BORRAR en la tabla B_DETALLE_VENTAS. El trigger deberá:
 - Actualizar el STOCK_ACTUAL en la tabla B_ARTICULOS (restar del stock cuando se inserta una venta, sumar al stock cuando se borra una venta, y modificar el stock al actualizar). Puede utilizar el procedimiento P_INCREMENTAR ARTICULO incluido en el paquete PACK_ART desarrollado en el ejercicio de paquetes.
 - Actualizar el MONTO_TOTAL en la tabla B_VENTAS (sumar al monto cuando se inserta una venta, restar del monto cuando se borra, y modificar el monto al actualizar).
 - Un trigger que se disparará ANTES de modificar el campo STOCK_ACTUAL sobre la tabla B_ARTICULOS. El trigger deberá
 - Validar que el nuevo valor de STOCK_ACTUAL no sea ≤ 0 . Si es así el trigger debe dar el mensaje apropiado y levantar un error.
 - Validar que el STOCK_ACTUAL no sea menor que el STOCK_MINIMO. Si es así, debe insertar un registro en la tabla PEDIDOS. La cantidad pedida debe ser igual al $\text{STOCK_MINIMO} + 25\%$. La fecha del pedido debe ser la fecha del sistema.
2. Programe a través de triggers las siguientes reglas de negocio sobre la tabla B_POSICION_ACTUAL:
 - Al INSERTAR la nueva posición de un empleado, debe verificar que la categoría que se asigna no tenga una asignación MENOR a la que tenía anteriormente
 - No debe permitir el borrado de un registro.
3. Programe un trigger que se dispare al INSERTAR un registro en la tabla B_DIARIO_DETALLE, que deberá ACTUALIZAR la tabla B_MAYOR (prever que el registro correspondiente al año y mes no esté aún creado, en cuyo caso deberá insertar el registro). El trigger deberá actualizar el acumulado de débito o el acumulado del crédito de la cuenta correspondiente con el importe, dependiendo si se trata de DEBE o HABER.
4. En la tabla B_CUENTAS debe controlar que una cuenta **NO pueda ser padre de otra cuenta** si ya es imputable ($\text{IMPUTABLE} = 'S'$). Del mismo modo, cuenta NO puede volverse imputable si ya tiene hijos.
Codifique el trigger de manera a evitar que se produzca el error de tabla mutante
5. Cree la vista V_CUENTAS, que deberá tener las siguientes columnas de las tablas B_CUENTAS y B_TIPO_CUENTA:

- CODIGO_CTA
- NOMBRE_CTA
- **NOMBRE_CATEGORIA**
- NIVEL
- CTA_SUPERIOR
- ORDEN
- FECHA_APERTURA
- IMPUTABLE

Prevea la operación de INSERCIÓN a través de la vista en la tabla B_CUENTAS. (El tipo de categoría debe existir). Debe impedir la actualización a través de la vista con un trigger-

6. Con el script proporcionado más abajo cree la tabla **B_AUDITORIA_ANULADOS** y luego programe un trigger que se dispare antes de eliminar una venta, el mismo debe realizar las siguientes acciones:
 - Actualizar el stock de los artículos a eliminar (restar del stock).
 - Eliminar los artículos de la tabla detalle.
 - Si es una venta a crédito, eliminar primero las cuotas generadas.
 - Por último, insertar los datos de la venta eliminada en la tabla: B_AUDITORIA_ANULADOS.

```
CREATE TABLE B_AUDITORIA_ANULADOS (
  ID_VENTA  NUMBER(8) NOT NULL,
  NUMERO_FACTURA  NUMBER(8) NOT NULL,
  MONTO_TOTAL  NUMBER(9) NOT NULL,
  FECHA_VENTA  DATE NOT NULL,
  ID_CLIENTE  NUMBER(8) NOT NULL,
  CONDICION_VENTA  VARCHAR2(2) NOT NULL,
  CANTIDAD_ARTICULOS  NUMBER(7) NOT NULL,
  USUARIO_ANULACION  VARCHAR2(30) NOT NULL,
  FECHA_ANULACION  DATE NOT NULL
);
```