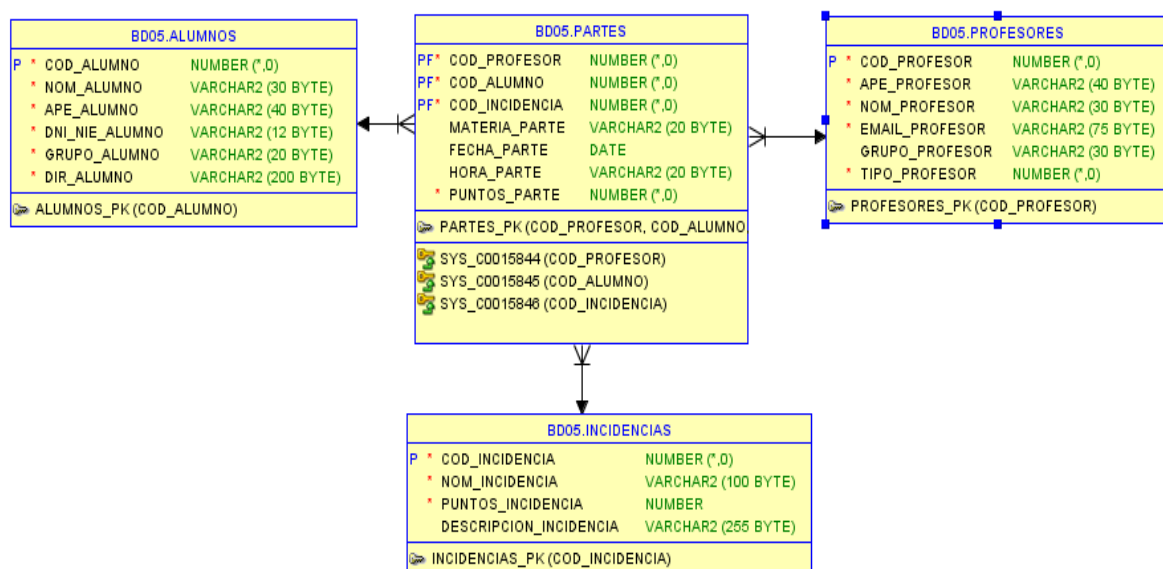


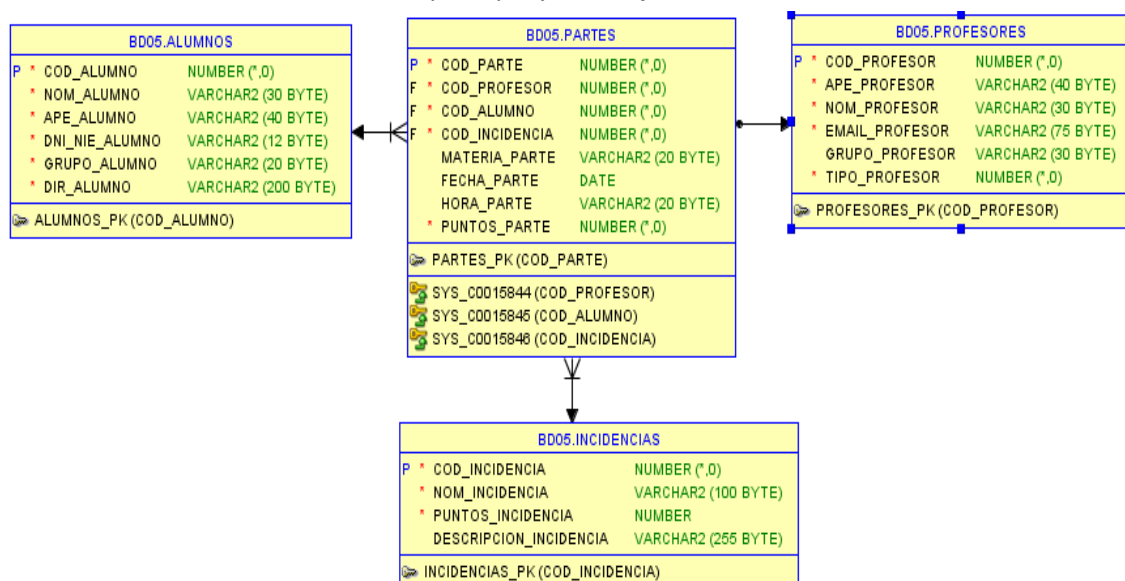
1. Crear todas tablas necesarias para reflejar la siguiente relación. *Las restricciones de claves principal y clave ajena las realizaremos a nivel de tabla.*



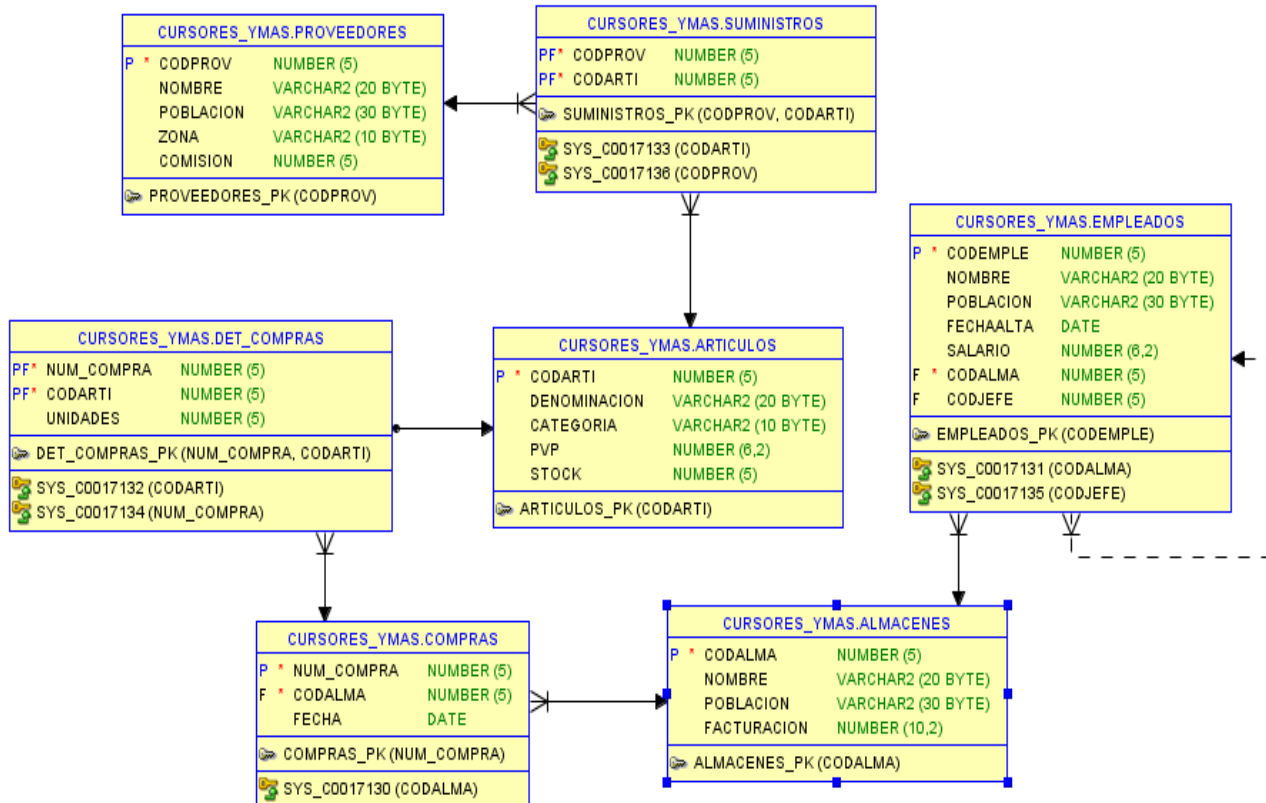
2. Uy!!!! Realmente la relación anterior era de la siguiente forma. Realiza **todas** las modificaciones necesarias, para reflejar los cambios realizados en el modelo. *Las restricciones de claves principal y clave ajena las realizaremos a nivel de tabla.*



3. Uuuuuuuuuuu!!!! Ahora tenemos que añadir un nuevo campo a la tabla de relación, que sea clave, es decir, la clave cambiará y pasará a estar formada sólo por el nuevo campo. Los campos que teníamos anteriormente en la tabla, seguirán existiendo en la tabla, pero tendrás que realizar los cambios necesarios como consecuencia de la inserción del nuevo campo clave. *Las restricciones de claves principal y clave ajena las realizaremos a nivel de tabla.*



## UTILIZA LA BASES DE DATOS BDALMACEN



1. Crear un bloque de código que nos permita controlar si existe el artículo que introduzcamos por teclado. Si el artículo existe, mostraremos todos los datos de sus proveedores por pantalla.
2. Realiza el ejercicio anterior utilizando una solución diferente para comprobar si existe algún proveedor del artículo introducido.
3. Crear un procedimiento al que le pasaremos desde un bloque de código como parámetro un número de compra (que introducirá el usuario) para que nos muestre por pantalla la denominación de cada uno de los artículos de la compra, su categoría, las unidades vendidas en una compra y el precio del artículo.

Tendremos que comprobar que el número de la compra sea correcto, en caso de no ser correcto, no se realizará la llamada al procedimiento y mostraremos un mensaje de error.

La salida se realizará el procedimiento será la siguiente forma:

```

NUMERO_COMPRA: xxxxxxxx          FECHA_COMPRA: xxxxxx
-----
      DENOMINACION      CATEGORIA      UNIDADES      PRECIO      SUBTOTAL
      -----
      xxxxxxxxxxxx      xxxxxxxxx      xxxxxxxxx      xxxxxx      xxxxxxxxx
      xxxxxxxxxxxx      xxxxxxxxx      xxxxxxxxx      xxxxxx      xxxxxxxxx
-----
TOTAL_ARTICULOS:xxxxx      TOTAL_UNIDADES:xxxxx      TOTAL_COMPRA: xxxxx
-----
  
```

4. Crear un procedimiento que a partir de un código de almacén nos muestre todas sus compras y para compra todos sus artículos. El código de almacén se pasará como parámetro al procedimiento. En caso de que no exista el código del almacén, mostraremos un mensaje que así lo indique y se realizará la llamada al procedimiento.

La salida se realizará de la siguiente forma:

NUMERO\_COMPRA: xxxxxxxx

DENOMINACION	UNIDADES	PRECIO
xxxxxxxxxxxxx	xxxxxxxx	xxxxxx
xxxxxxxxxxxxx	xxxxxxxx	xxxxxx

NUMERO\_COMPRA: xxxxxxxx

DENOMINACION	UNIDADES	PRECIO
xxxxxxxxxxxxx	xxxxxxxx	xxxxxx
xxxxxxxxxxxxx	xxxxxxxx	xxxxxx

NUMERO\_COMPRA: xxxxxxxx

DENOMINACION	UNIDADES	PRECIO
xxxxxxxxxxxxx	xxxxxxxx	xxxxxx
xxxxxxxxxxxxx	xxxxxxxx	xxxxxx

5. Crear una función que nos devuelva el código del empleado que tenga el mayor sueldo del almacén que introduzca el usuario. La función será llamada desde un bloque donde se mostrarán los datos del empleado. A la función le pasaremos como parámetro el código del almacén y nos devolverá el código del empleado.

En caso de que el almacén no exista, o se produzca otro error. Se mostrará un mensaje y la función no realizará ninguna acción.

6. Crear una función que devolverá el código del proveedor que introduzca el usuario por teclado, esta información será pasada a un procedimiento que se encargará de mostrar los datos del proveedor introducido o bien un mensaje de error en caso de que no exista el proveedor.

Ni la función, ni el procedimiento se almacenarán en la base de datos.

7. Crear un trigger que nos permita controlar si existe el proveedor y el artículo cuando insertemos un registro en la tabla de suministros.
8. Crear un trigger que almacene en la tabla de HISTORIAL\_EMPLEADOS (que no existe) el usuario, la fecha y la operación que se ha realizado. El trigger se disparará cada vez que se realice una operación sobre la tabla de empleados.
9. Crear un trigger que se ejecute cuando modifiques el stock de un artículo, de tal forma que no pueda tomar valores negativos.

Si al realizar la modificación el campo stock tomara un valor negativo, le asignaremos 10 unidades.

Si al realizar la modificación el campo stock el valor nuevo fuera menor que el valor que tenía anteriormente el campo no se realizará la modificación y se almacenará el stock que había anteriormente.

10. Crear un trigger que se ejecute cuando se realice un cambio de los datos de un empleado de tal forma que realice las siguientes comprobaciones sobre los nuevos datos del empleado:

- a. La fecha de alta del nuevo contrato del empleado debe ser mayor o igual a la fecha de hoy. En caso contrario, se le asignará la fecha de hoy.
- b. El salario debe ser mayor que 0 y mayor que el salario anterior. Si no es así, se le asignará un salario un 10% mayor del salario que tenía anteriormente.
- c. El código de almacén debe existir en la tabla de almacenes. Si el nuevo código de almacén no se existe le asignaremos el código que tenía anteriormente.