

Estructura y formato de los CSV

Los csv deben tener un formato específico para su correcta subida a la BBDD. Esto se debe a que el código está pensado para insertar los datos de una manera determinada, aunque es posible modificarlo a nivel de código muy fácilmente.

La idea de este formato único tiene el objetivo de que la BBDD se toque lo menos posible y mantenga una estructura fija y coherente.

Los csv deben tener 8 columnas en este orden:

1º referencia (NO PUEDE ESTAR VACIO)

2º id (esta columna DEBE ESTAR VACIA ya que la bbdd inserta un id automáticamente)

3º PROVEEDOR (el id del proveedor que DEBE estar en todas las filas del csv)

4º Talla

5º Color

6º Imagen

7º Descripción

8º Precio (NO PUEDE ESTAR VACIO, debe estar en formato número con 2 decimales y sin símbolo de moneda a poder ser)

*hay que asegurarse de que los csv no tenga más ni menos de 8 columnas y los datos estén en su correcto orden.

En general código de la inserción consiste en lo siguiente, aunque se realizan más operaciones en el proceso:

Un bucle que recorre el csv

```
$handle = fopen( $_FILES['file']['tmp_name'], "r");  
  
while( ($data = fgetcsv($handle, 1000, ";") ) !== FALSE )
```

Se comprueba que cada registro del csv no exista ya en la BBDD para controlar repetidos

```
//busca repetidos en la bbdd  
$query = mysqli_query($conexion,"SELECT * FROM productos WHERE  
REFERENCIA = '$data[0]'  
AND PROVEEDOR = '$data[2]'  
AND PRECIO = '$data[7]'  
");  
  
$result = mysqli_num_rows($query);  
  
//si encuentra repetidos no hace nada  
if($result > 0){  
    $resultado=TRUE;  
  
//si no hay repetidos inserta el nuevo registro  
}elseif($result == 0){
```

Si no existe repetido se monta la query para insertar el registro

```
//si no hay repetidos inserta el nuevo registro
}elseif($result == 0){

    $q = "INSERT INTO productos values (
        '$data[0]',
        '$data[1]',
        '$data[2]',
        '$data[3]',
        '$data[4]',
        '$data[5]',
        '$data[6]',
        '$data[7]'
    )";

    require_once "../conexion.php";

    $resultado=mysqli_query($conexion,$q);
```

A tener en cuenta sobre la BBDD

(las imágenes son orientativas, no es la estructura final)

- Los artículos **se insertan en la tabla productos y a la vez se crea una copia idéntica en la tabla historial_productos.**

central_uniformes productos	central_uniformes historial_productos
REFERENCIA : varchar(20)	FECHA_REGISTRO : datetime
ID : int(15)	REFERENCIA : varchar(20)
PROVEEDOR : int(11)	ID : int(15)
TALLA : varchar(10)	PROVEEDOR : int(11)
COLOR : varchar(20)	TALLA : varchar(10)
IMAGEN : varchar(60)	COLOR : varchar(20)
DESCRIPCION : varchar(60)	IMAGEN : varchar(60)
PRECIO : float	DESCRIPCION : varchar(60)
	PRECIO : float

- **Los pedidos se crean** obteniendo el **id** del artículo de la **tabla productos**, pero **los datos se recogen de su copia en la tabla historial_productos**. **Esto funciona así para** asegurarla **integridad** de los datos. **Véase la tabla productos como una guía** de los artículos actuales y activos para la venta, **y a la tabla historial_productos como la caja fuerte** de donde se sacan esos artículos y que cohabitan con otros que ya no están a la venta pero que en su día se pudieron llegar a vender.
- **Cada vez que se inserta un artículo** este adquiere un **id autoincrementado**. Esto quiere decir que si se borra un artículo con id 1, al **volver a insertar el mismo**, ahora tendrá asignado id 2. **Este problema se soluciona** de la siguiente manera, cuando se inserta un producto **se consulta** a la tabla historial_productos si existe un **artículo similar comparando referencia, proveedor y precio** (se podría añadir más campos en la comparación si fuera necesario) de los dos artículos. **Si sus campos son iguales** entonces el programa le asigna el id del artículo que está en el historial al nuevo artículo insertado en la tabla productos ya que el programa **considera con son el mismo artículo**. De esta manera los ids se mantienen iguales, aunque se borren y se vuelvan a meter los mismos artículos en la tabla productos.
- **Es muy importante mantener la integridad del historial** ya que es realmente de ahí de donde se obtienen los datos de los artículos. **Si se borran los artículos de la tabla historial_productos** no se podría recuperar la información de los artículos que se vendieron en x pedido, y a la hora de leerlos desde la app saltarían errores de lectura (porque está intentado buscar un producto que alguien borro del historial). De todas formas, **los pedidos y sus detalles siempre se mantendrán intactos**, aunque se borre el historial, **pero en el campo de idproducto solo aparecerá el id** que hace referencia a un producto que ya no existe con lo cual ya no se podrá recuperar los datos de ese artículo.

- Hay que tener cuidado al editar productos en la BBDD **ya que no coincidirá con su copia** de la tabla historial, **si se edita uno se debe editar en las dos tablas** y deben ser idénticos.
- **Cualquier problema o error** de lectura en los datos referentes a los productos casi con total seguridad que es motivo de que un artículo no tiene su copia en el historial o de que se borró un artículo cuyo id está asociado a algún pedido.
- **Los únicos productos que se podrían borrar del historial** sin que afecten a la aplicación son aquellos productos que **nunca se llegaron a vender** ya que **sus ids no están asociados** a ningún pedido. Esto se podría saber realizando una sencilla consulta a la tabla detalles.
- **Es recomendable hacer respaldos** de las tablas (historial_productos, pedidos, detalle) ya que son el eje del programa.