

CerveZetas



Por José Luis Martínez Pérez

1º DAW

Modelo y explicación del E/R

Mi proyecto CerveZetas, después de muchos giros de guión va tomando forma tanto el Modelo Entidad relación, como el relacional y la parte final en SQL.

Para esta primera parte contamos con un total de 6 Entidades. Comenzamos con **Usuario**, que tiene los atributos básicos y para minimizar el trabajo y habiendo realizado ya una parte del proyecto en java, añadimos "**TipoUsuario**", que tiene la función de un **Check** en la creación de tablas en el .DDL, y en el proyecto de java sería una clase enum, que tendría en su interior: **Administrador**, **Cliente** y **Visitante**.

A parte en esta entidad tenemos como clave primaria "email", dado que los **Administradores** tendrán un correo electrónico especial para ellos que se distinguirá del de **Clientes**. Los **Visitantes** son aquellos que ni se loguean ni se registran en la web. **Teléfono** puede ser registrado por cada Cliente una o varias veces, debido al uso de fijos, o cambios de número.

Usuario tiene relación con **Producto**, pero estos, solo se podrán relacionar haciendo si son administradores, esto se reflejará como una restricción más adelante.

Producto tiene atributos normales, y luego he añadido **Categoría** y **Tipoproducto** como atributos, estos serán como el ya mencionado **TipoUsuario**, tendrán un check cada uno en el .DDL, que será, para **Categoría**: **Rubia**, **Malta**, **Tostada**. Y para **TipoProducto** será: **Pack Especial**, **Pack Eventual**, **Cubos**, **Cervezas**. Esto se verá reflejado en java con una clase enum.

Usuario a su vez se relacionará con **Pedido** y **Dirección**, dado que un **Usuario** tendrá una dirección asociada a su cuenta y este realizará un pedido o ninguno, pero será el **Usuario** quien tenga ese poder.

Dirección tiene los atributos normales que constan en una **Dirección**, pero he añadido **código dirección** para que haya una clave primaria que con la que se identifique una dirección concreta, ya que poner uno de los demás atributos como clave primaria sería algo muy raro ya que hay calles, municipios, números o códigos postales que se pueden repetir en una base de datos. Este no es el único atributo que he agregado diferente a los normales, estarían en este grupo tanto **TipoDirección**, **TipoVía**, que tendrían checks en sus respectivas tablas en el .DDL, con los atributos, tanto de **TipoDirección**: **Casa**, **Trabajo**, **Lugar Vacacional**, **Campo**. Y en **TipoVía**: **Calle**, **Avenida**, **Plaza**.

Línea pedido, es una relación entre **Pedido** y **Producto**, dónde se almacenará en la relación tanto los atributos de **Pedido**, como los de **Producto** y a su vez la relación tiene dos atributos, **Precio** y **Cantidad**.

Y por último **Producto** tendrá una marca asociada por eso es una relación de existencia, porque sin una **Marca** un producto no existiría.

Los enums antes mencionados pueden dar confusión en algún caso, pero esto lo he hecho de manera que sea más fácil y sencillo implementar el código final, se que esto se podría implementar directamente haciendo objetos de cada tipo y con ello no tener problemas en añadir una que no esté. Pero tanto en **Categorías** como en **TipoProducto** veo necesarias estas "restricciones" por llamarlo de alguna manera.

[Enlace al Lucid para ver el E/R.](#)

Modelo Relacional

Marca (Código, nombre)

PK: código

Dirección (código_dirección, TipoVía, TipoDirección, Número, Provincia, CódigoPostal, Municipio, email_usuario)

PK: código_dirección, email_usuario

FK: email_usuario → **Usuario**.email

Usuario (email, TipoUsuario, Teléfono, Apellidos, Foto, Nombre, Contraseña, Última_conexión, F_Nacimiento)

PK: email

Marca (código, Nombre)

PK: código

Producto (código, Categoría, TipoProducto, Fecha_Modificación, Precio, Descripción, Nombre, Stock, Fecha_Creación, código_marca, admin_crea, admin_modifica)

PK: código

FK: código_marca → **Marca**.código

FK: admin_crea → **Usuario**.email

FK: admin_modifica → **Usuario**.email

Pedido (código, Fecha, Precio_total, Dirección_pedido, email_usuario, usuario)

PK: código

FK: dirección_pedido, usuario → **Dirección**.(código_dirección, email_usuario)

FK: email_usuario → **Usuario**.email

Factura (número, Fecha, Precio_total, Dirección_facturación)

PK: número

FK: Dirección_facturación → **Dirección**.código_dirección

Línea_Pedido (código_pedido, num_factura, código_producto, precio, cantidad)

PK: código_producto, código_pedido

FK: código_pedido → **Pedido**.código

FK: código_producto → **Producto**.código

VNN: (Producto, Pedido)

Pedido_genera_Factura (cod_pedido, num_factura, fecha_factura, dirección, email_cliente, precio_total, métodoPago, tipoUsuario)

PK: cod_pedido, num_factura

FK: dirección, email_cliente → **Dirección**(código_dirección, email_cliente)

FK: tipoUsuario → **Usuario**.email

VNN: (Factura, Pedido)