**El modelo de datos en Power Pivot**

Al trabajar con Power Pivot es inevitable encontrar el término *Modelo de datos* que es algo nuevo para la gran mayoría de usuarios de Excel. Este término proviene de la teoría de bases de datos y es importante explicar algunos conceptos que nos ayudarán a tener un buen fundamento para trabajar con Power Pivot en Excel.

Debo decir que, si eres una persona que tiene experiencia o entrenamiento previo en bases de datos, los conceptos expuestos en este artículo serán muy básicos para ti. Por el contrario, si nunca has trabajado con bases de datos relacionales, es importante leer con detenimiento y comprender los términos que explicaré a continuación.

**¿Qué es un modelo de datos?**

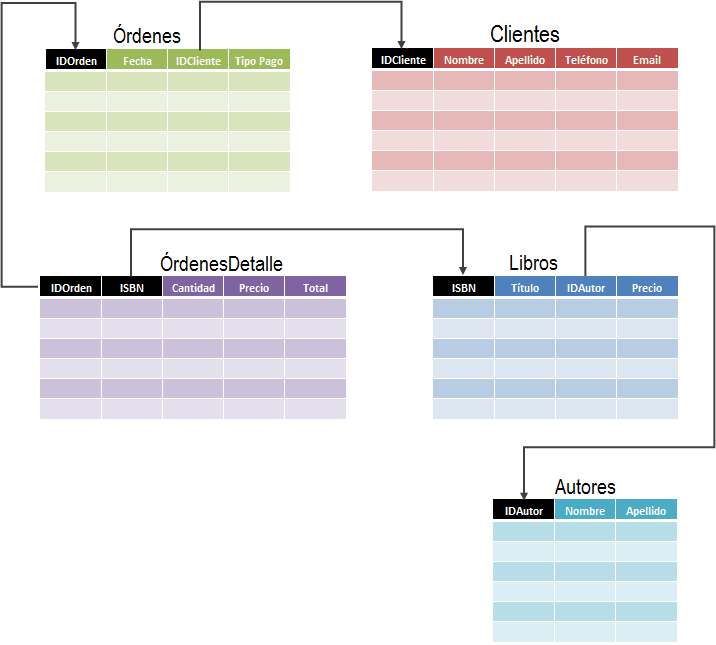
Una de las funciones principales de los sistemas computacionales es la de manejar y almacenar grandes cantidades de información. Un modelo de datos describe la estructura de esos datos que están almacenados en dichos sistemas.

El objetivo principal de un modelo de datos es darnos información sobre la manera en que están almacenados los datos. También podríamos definir el modelo de datos como un mapa que nos ayudará a comprender la forma en que se han organizado y almacenado los datos.

Dentro de la teoría de bases de datos encontrarás diferentes tipos de modelos, pero el que nos interesa y que es relevante para nuestro trabajo con Power Pivot, es el *modelo de datos relacional*.

**El modelo de datos relacional**

Comprender la teoría de datos relacionales podría llevarnos todo un curso, pero para hacer las cosas extremadamente simples, podría decir que este modelo nos dice que los datos pueden representarse por un conjunto de tablas que estarán vinculadas entre sí por un campo en común.



En la práctica, es común referirnos a estos elementos como tablas y relaciones. Una tabla está formada por columnas y filas mientras que una relación nos indica la columna que ha sido utilizada para vincular a dos tablas entre sí.

Pero ¿por qué querría tener muchas tablas cuando toda la información puede estar contenida en una sola? ¿Qué beneficios tiene distribuir la información en múltiples tablas? Esta es una pregunta muy común y totalmente válida para los usuarios que comienzan en el modelado de datos y la razón la explicaré con un ejemplo.

Supondremos el caso de una empresa que se dedica a vender teléfonos celulares y en los últimos días se han realizado varias ventas que han sido registradas de la siguiente manera:

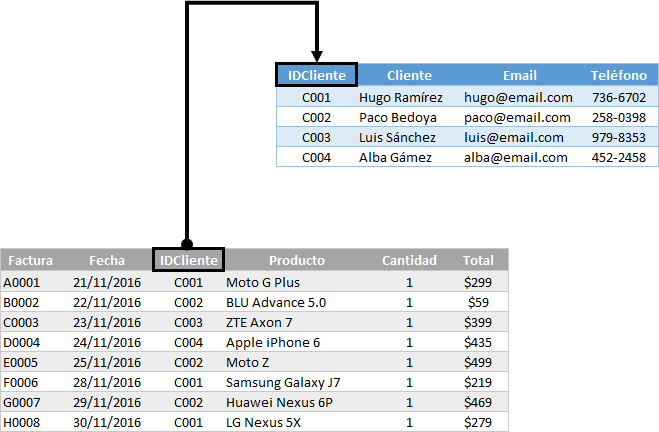


Quiero que pongas atención en las columnas *Cliente*, *Email* y *Teléfono* las cuales tienen información de contacto del cliente que ha hecho la compra. Ya que la empresa tiene clientes recurrentes, la información se vuelve a repetir cada vez que el mismo cliente hace una compra y eso puede traernos algunos problemas. A continuación menciono algunos de esos problemas:

* **El ingreso de nuevos datos**: Si en alguna nueva compra se ingresa de manera incorrecta la información de contacto del cliente, ya no habrá manera de saber cuál de todos los datos es correcto o incorrecto. Por ejemplo, si la próxima compra de *Hugo Ramírez* se ingresa el correo electrónico *hugo2@email.com* no habrá certeza sobre cuál de los dos correos es correcto en el historial de compras de *Hugo*.
* **La actualización de datos**: Si *Hugo Ramírez* cambia de *Teléfono* de contacto y queremos mantener actualizada la tabla anterior, será necesario modificar cada fila con la nueva información de *Hugo*. Siempre existe la posibilidad de omitir el cambio en alguna fila y de nueva cuenta no sabremos cuál de los teléfonos es el correcto.
* **El espacio de almacenamiento**: Cada letra y palabra ingresada en la tabla se traduce en *bytes* de información en el disco del equipo. Entra más grande sea la cantidad de datos repetidos, mayor será el tamaño de la base de datos. No hace mucho sentido guardar información repetida que puede tener algún impacto en el rendimiento de una base de datos.

Estos son algunos de los problemas que te puedes encontrar al colocar todos los datos en una sola tabla. Por supuesto, el impacto será mayor conforme la base de datos crezca, así que es conveniente cuidar estos aspectos de diseño desde el principio.

Para evitar estos problemas en nuestro ejemplo, debemos dividir la tabla original en dos tablas de la siguiente manera:



Las tres columnas *Cliente*, *Email* y *Teléfono*, que hacían referencia a la información del cliente, ahora son una tabla independiente y se ha agregado una columna adicional que funciona como un identificador único (*Llave*) para cada cliente. De esta manera, cuando queremos hacer referencia a un cliente desde otra tabla, podemos utilizar su identificador único.

La gran ventaja de este cambio es que solo debemos enfocarnos en la tabla que contiene la información de los clientes para hacer cualquier modificación. No será necesario hacer actualizaciones en diferentes filas o tablas, ya que ahora toda la información que se refiere a los clientes está en una sola tabla. De esa manera también reducimos el espacio requerido para el almacenamiento de los datos.

De eso se trata el modelado de datos, de adoptar buenas prácticas que hagan más eficientes nuestras bases de datos. A este proceso de “mejora” se le conoce como [Normalización de bases de datos](https://es.wikipedia.org/wiki/Normalizaci%C3%B3n_de_bases_de_datos) y es la aplicación de un conjunto de reglas (*Formas normales*) a una base de datos. Cuando una base de datos cumple con las *formas normales*, se dice que la base de datos ha sido normalizada.

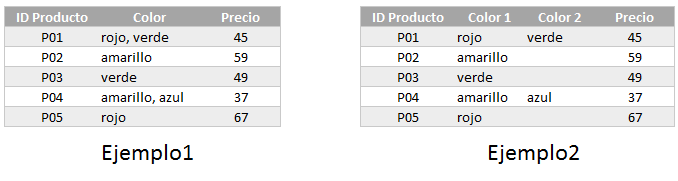
**Aplicando las *formas normales* a tus datos**

El diseño de bases de datos es un tema realmente amplio e imposible de abarcar en un solo artículo. Existen libros enteros dedicados a este tema y solamente el estudio y la práctica te pueden convertir en un experto.

Sin embargo, no quiero dejar pasar la oportunidad para darte algunos consejos rápidos que puedes poner en práctica de inmediato para mejorar el diseño de tus datos. A continuación explicaré brevemente un par de reglas que debes tomar en cuenta.

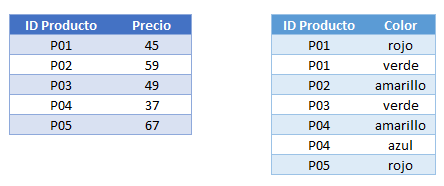
**Primera forma normal**: *Los valores de la base de datos deben ser atómicos es decir, no pueden ser divididos aún más. Además, no deben existir grupos de columnas que repitan datos.*

Los dos errores más comunes que no cumplen la primera forma normal son los siguientes:



En el *Ejemplo1*, la columna *Color* tiene valores que no son atómicos sino que son valores separados por comas que pueden ser divididos aún más. En el *Ejemplo2*, los colores son atómicos pero la tabla tiene dos columnas para guardar el mismo tipo de información.

En ambos casos se tendrá un problema al momento de quitar o agregar un nuevo color para un producto. La solución es crear una nueva tabla que nos permita enlistar todos los colores disponibles para un mismo producto de la siguiente manera.



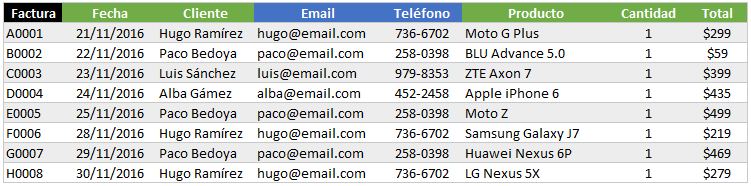
Cuando quieras conocer los colores disponibles para un producto, deberás elegir las filas de la tabla (*derecha*) que tengan la clave del producto indicado.

No solo es necesario cumplir las reglas anteriores para que una tabla se encuentre en la primera forma normal, sino que además es indispensable que la tabla tenga una *clave* o *llave primaria*. Esto quiere decir que, debe existir una o varias columnas que identifiquen de manera única a cada una de las filas.

Si la llave primaria consiste de una sola columna, entonces dicha columna no podrá tener valores repetidos. Si la llave primaria es la combinación de dos columnas o más columnas, entonces habrá valores repetidos de manera individual en cada columna pero jamás se repetirá la misma combinación entre todas las columnas.

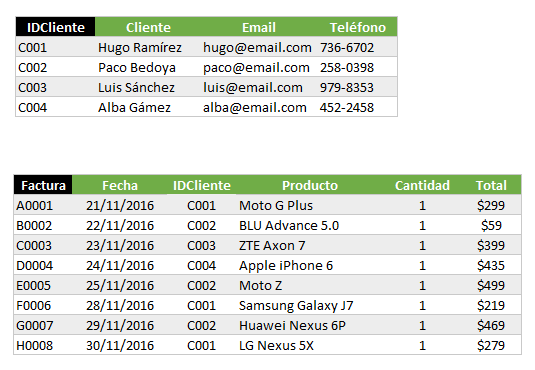
**Segunda forma normal:** *Las columnas de la tabla dependerán completa y únicamente de la llave primaria.*

Este caso lo podemos ejemplificar con el ejemplo que realizamos al comienzo de este artículo. En dicho ejemplo, la tabla original tenía columnas que no dependían de la llave primaria. Observa la tabla con detenimiento:



En color negro está la llave primaria de la tabla y en color verde están las columnas que dependen de dicha llave como la *Fecha* de la factura, el *Cliente* que ha hecho la compra, el *Producto* adquirido, su *Cantidad* y *Precio*.

Por el contrario, las columnas *Email* y *Teléfono* no dependen de la columna *Factura* sino del *Cliente*. Por esa razón deben salir y colocarse en una tabla independiente donde quede establecida su dependencia con el cliente.



De esta manera, cada tabla tiene su llave primaria y el resto de campos dependen solamente de ella. Con estos cambios, los datos ahora cumplen también la segunda forma normal.

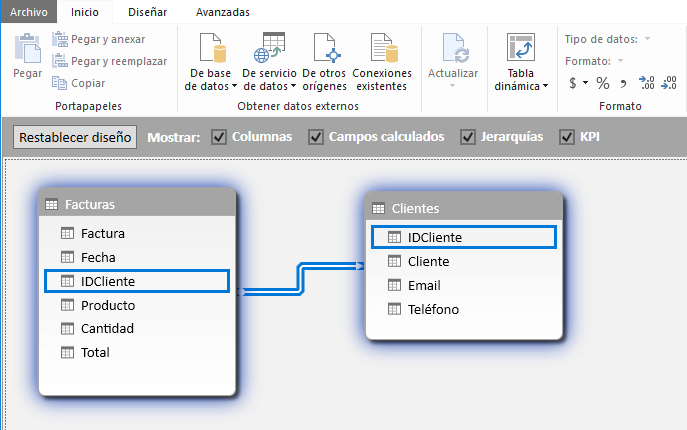
Sería presuntuoso pensar que con la explicación anterior ya se ha cubierto todo el material y casos de uso de la primera y segunda regla normal, pero ahora tienes un par de ideas que puedes comenzar a aplicar desde hoy al trabajar con tus datos.

En los libros encontrarás que existen hasta cinco formas normales que podrás aplicar a una base de datos. Es necesario investigar y leer más sobre este tema para que puedas aplicar todos los consejos posibles al momento de modelar una base de datos.

**El modelo de datos en Power Pivot**

Algo que no debes olvidar después de leer este artículo es que antes de [importar los datos en Power Pivot](https://exceltotal.com/importar-datos-power-pivot/), es altamente recomendable asegurarse que están normalizados. Recuerda que, una vez que importas una tabla al modelo de datos de Power Pivot no podrás modificar su estructura es decir, no podrás modificar sus columnas o filas sino que solamente podrás crear relaciones con las demás tablas del modelo.

Power Pivot importará cualquier base de datos que le pidas, ya sea que tenga los datos normalizados o sin normalizar. Es tu responsabilidad como analista, el cuidar la calidad e integridad de los datos que incluyes en el modelo. Los datos normalizados enriquecerán en gran medida tu modelo de datos y te permitirán crear reportes altamente efectivos.



Para conocer más sobre este tema puedes buscar en internet o hacerte de un buen libro sobre el *Diseño de bases de datos*. Aunque este blog es leído por hispanoparlantes, si sabes un poco de inglés técnico, me gusta recomendar el libro “[Beginning Database Design Solutions](https://www.amazon.com/Beginning-Database-Design-Solutions-Stephens/dp/0470385499/" \t "_blank)” de la editorial Wrox el cual fue traducido al español por la editorial Anaya Multimedia. Otra opción es el libro “[Beginning Database Design: From Novice to Professional](https://www.amazon.com/Beginning-Database-Design-Novice-Professional/dp/1430242094/" \t "_blank)” de la editorial Apress.

No te desanimes si alguno de los conceptos que he compartido en este artículo no ha quedado claro. En ocasiones la teoría de modelado de datos puede ser difícil de entender en la primera lectura, pero con la práctica seguramente dominarás todas las reglas y podrás crear modelos de datos excepcionales en Power Pivot.

Publicado por [Moisés Ortíz](https://exceltotal.com/acerca/).