**Mapas, Gráficos de dispersión & Dashboard**

## **Relaciones vs Uniones**

### **Importancia de los diagramas lógicos y físicos**

Si usted viene de aprender SQL, que es lo ideal, recordará que los conceptos de *relaciones* y *uniones* tienen su origen en los diagramas de flujo de datos lógicos y físicos, siendo el último una consecuencia del primero.

Se había planteado sobre los diagramas lógicos como aquellos que se enfocan en el negocio y en las actividades (relaciones) del negocio en cuestión, mientras que en un Diagrama físico se analiza las formas en las que se implementa un sistema. En resumen, el diagrama lógico proporciona el ‘qué’ y el físico el ‘cómo’.

Sin embargo, también se aclaró que en los diagramas lógicos (previos a los *físicos*) es donde ocurren o se ***asignan relaciones*** entre entidades, y que en los diagramas físicos, ya conformados, es donde se pueden ***unir*** las tablas o entidades en cuestión; pues bien, en Tableau, esto no es la excepción.

**En Tableau, sus *relaciones* se dan en un diagrama lógico y sus *uniones* se dan en un diagrama físico.**

### **Configurar relaciones y uniones en Tableau**

En esta sección nos centraremos en las *uniones*,sin embargo, vamos a dejar claro de forma anticipada las diferencias que hay a la hora de configurar relaciones y/o uniones en Tableau.

Convengamos que hemos importado en Tableau el *dataset* ‘P1-AmazingMartEU2’, que corresponde también con el dataset con el que estaremos trabajando por el resto de esta sección. La pantalla inicial que ve, inmediatamente después de importado el archivo, es la siguiente:

Texto

Descripción generada automáticamente con confianza baja

He aquí donde usted debe seleccionar qué fichero, *tabla/entidad,* arrastrar para analizar visualmente en el panel. Supongamos que deseo arrastrar la tabla: ‘ListOfOrders’. Verá algo así:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

#### **Configuración de Relaciones**

Por defecto, usted actualmente se encuentra en el espacio que corresponde al diagrama lógico; es decir, es este el espacio apropiado para crear *relaciones* con otras tablas en Tableau; por ejemplo, extendamos una *relación* entre la tabla ‘ListOfOrders’ y la tabla ‘OrderBreakdown’ (debe arrastrar esta última al panel), tal que así:

Diagrama

Descripción generada automáticamente

Usted aquí ha creado una *relación* entre la tabla ‘ListOfOrders’ y la tabla ‘OrderBreakdown’. Hasta ahí bien.

#### **Configuración de Uniones**

Ahora, supongamos que lo que realmente queríamos configurar era una Unión y no una Relación. Para configurar una Unión entre, por ejemplo, la tabla ‘ListOfOrders’ y la tabla ‘OrderBreakdown’; tenemos que hacer doble clic sobre la primera tabla arrastrada, es decir, sobre ‘ListOfOrders’.

De hecho, si uno flota el ratón sobre la tabla ‘ListOfOrders’ verá un *tooltip* que le avisa explícitamente que se encuentra, de momento, sobre el entorno de un diagrama lógico y que si desea estar en un entorno de diagrama físico debe hacer doble clic sobre su tabla arrastrada (la de interés para hacer una unión con otra). Veamos.

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente, *hagamos doble clic entonces para situarnos sobre un entorno de diagrama físico y, poder así, tener la posibilidad de crear* ***uniones****.* Observe:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja, arrastremos de nuevo la tabla ‘OrderBreakdown’ para crear una unión con la tabla ‘ListOfOrders’.

Diagrama

Descripción generada automáticamente

Por defecto, se crea una unión de tipo *‘Inner Join’ (Unión interior)*, sin embargo, puede cambiar esta configuración haciendo clic sobre la unión misma, … vería la siguiente ventana emergente:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Los tipos de *Unión* ya los conoce desde SQL, revise el documento *‘4. Introducción a Consultas y su importancia (Complejidad; Intermedia)’* para eventual recordación.

Ahora, si cierra el entorno de su diagrama físico, regresando al entorno de su diagrama lógico, podrá ver que la unión entre ambas tablas sigue estando vigente, tal que así:



## **Uniendo Datos en Tableau**

### **Join de SQL = Unión en Tableau**

Si usted recuerda el concepto de ***From + Join de SQL***, éste fue introducido así: … ‘Como usted ya sabe, por cuestiones de *normalización,* siempre se busca que cada entidad se estudie por separado; esto, al asignarles tablas independientes, aunque relacionadas, a cada una de ellas. Sin embargo, a la hora de presentar un informe, a la hora de traer información que sea valiosa, conviene unir dichas tablas; pues, nunca dejan de relacionarse y nos ayudaría, justamente, a *extraer más conclusiones e información adicional* que respondan a nuestras dudas.

**From + Join** es el primer puntapié para darle fundamento práctico a la teoría que esconde el concepto *Bases de datos* ***relacionales;*** lo que trata sobre, justamente, las ***relaciones.*** Es importante relacionar entidades entre sí para poder generar información y darles un sentido práctico a los datos que tenemos al ser contextualizados al mundo real, a las consultas reales que se pretenden responder en una organización, con o sin ánimo de lucro. Las *uniones* nos permiten, con mayor facilidad, convertir el código y los datos en información útil y diciente.

**Por ejemplo**, pongamos esto en contexto, si tuviéramos un negocio de compra y venta de libros, no es muy práctico saber todo acerca de la entidad ‘*libros’* solamente, … definitivamente no si no se puede ***unir***dicha entidad con la entidad ‘*clientes’* y la entidad ‘*autores’*; pues, no podríamos relacionar un libro vendido X con un autor Y que lo escribió ni con un cliente Z que lo adquirió; entonces, para extraer la información completa sobre esta consulta, usted debe necesariamente unir 3 tablas, dado que estos datos no sólo se encuentran en una entidad, sino en 3: ‘libros’, ‘clientes’ & ‘autores’. Dicho lo anterior, en SQL (y también en Tableau),**la información se crea cruzando las tablas.**

Si tenemos, por ejemplo, que queremos cruzar la tabla de “autores” con la de “libros” y tenemos en cuenta que la cardinalidad es 1:M respectivamente (porque se sabe que un autor puede *escribir* cuantos libros quiera, pero un libro puede ser escrito una sola vez por un mismo autor o grupo de autores), y que la relación es “escribir” justamente; entonces, estaríamos extrayendo información (*insights*) tipo **“El libro ABC fue escrito por el autor XYZ en la fecha 00/11/2222”. Tenga algo en cuenta: Si dos tablas se unen (o más), puede llamar (SELECT) cualquier columna de ambas tablas! (o más)**’… Pues bien, bajo la misma estructura lógica funciona Tableau.

Si ahora ponemos todo lo anterior en contexto, desde Tableau, con relación a nuestro *dataset* de trabajo‘P1-AmazingMartEU2’, podríamos generar las mismas *Uniones* dentro de esta tecnología también.

Las tablas con las que estaremos trabajando son: ‘ListOfOrders’ & ‘OrderBreakdown’… Si bien son dos tablas diferentes, necesariamente se relacionan entre sí para poder crear información al cruzar sus datos. ‘ListOfOrders’ posee datos sobre *pedidos realizados*, mientras que ‘OrderBreakdown’ posee datos sobre *los artículos para cada pedido realizado.*

Parta de lo siguiente:

**No ganamos mucho, o no generamos información relevante, si sólo nos quedamos con los datos que la tabla ‘ListOfOrders’ conoce: el *dónde*, *cuándo* y *quién* realiza un pedido, … sin poder saber o conectar dichos datos con los datos que la tabla ‘OrderBreakdown’ ofrece: los artículos que cada cliente pide en un pedido (o viceversa). He aquí donde nace la importancia de las *Uniones,* tanto en SQL como en Tableau*,* porque nos permiten crear información al cruzar tablas relacionadas.**

Unamos en Tableau ambas tablas, ‘ListOfOrders’ & ‘OrderBreakdown’, ya sabe hacerlo.

Diagrama

Descripción generada automáticamente

Estando unidas ambas tablas podemos determinar la Cardinalidad y llaves entre ellas.

### **Cardinalidad y llaves entre List of Orders & OrderBreakdown**

Como bien sabe, ***List of Orders*** contiene los datos sobre las órdenes realizadas y ***OrderBreakdown*** contiene los datos de los artículos para cada orden realizada. Si se fija, una orden es única e irrepetible, pero dentro de una orden puede haber la compra de uno o más de un artículo. Si hay más de un articulo para una misma orden, sepa que todos esos artículos pertenecen al mismo ID de su orden.

Como ambas entidades se componen de un campo que las relaciona entre sí, el cual sería esta vez *Ardera* (ambas entidades lo tienen), entonces podrían unirse. De hecho, ahora, la entidad de *Primary key*, si esta discusión fuera necesaria, vendría de la tabla *List of Orders*; pues, la relación que se guarda con la entidad *OrderBreakdown* es de 1: N; es decir, una orden puede o no tener muchos artículos vendidos.