**Uniendo y Combinando Datos & Gráficos de Doble Eje**

## **Profundización a Joins: Múltiples Campos**

### **Uniendo tablas con Múltiples Campos**

Al grano. Realmente Tableau guarda los mismos principios que SQL a la hora de hacer *Joins,* por lo que se recomienda leer sobre *Joins* desde la documentación que ya hay de SQL. Ahora, una novedad particular. Si usted bien recuerda, en SQL sólo bastaba con que las tablas estuviesen lo suficientemente *normalizadas* como para garantizar que sólo fuera necesario el uso de un solo *campo,* de tipo *llave* y por cada tabla,para poder unir dos tablas relacionadas entre sí. Es decir, con solo un campo *llave* que estuviese en ambas tablas relacionadas, que no fuera *null* ni se repitiera (que fuera *unique* por cada registro), era posible hacer una unión correcta entre dos o más de dos tablas (y de hecho esto es lo ideal, no se conoce otra forma en SQL para hacer *Joins*).

Sin embargo, en Tableau, las tablas no siempre están lo suficientemente *normalizadas* y se corre el riesgo de que ninguna tabla tenga un campo con un identificador único por cada registro (al menos no lo tenga el campo *primario* con el que nos interese, propiamente, unir las tablas en sí), es decir, se corre el riesgo de que se repitan valores en los registros de los campos o columnas *llaves* de las tablas de interés (las que nos interesa unir); por lo que a veces es necesario no sólo tomar un campo como parámetro (como *llave* única) para hacer una unión correcta sin generar registros erróneos; sino, tomar **múltiples campos** para garantizar que los registros de cada una de las tablas unidas **sí** sean correspondidos entre sí, *los datos sí estén limpios*. **Observe el siguiente ejemplo para mayor comprensión. Se evidenciará porqué, para este caso, será necesario unir las tablas con múltiples campos.**

#### **Ejemplo**

Trataremos de unir dos tablas, A & B, considerando solamente al campo ***Order #*** como campo *llave* para lograr la unión.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Con las especificaciones previamente dadas, nuestra unión quedaría así:

Interfaz de usuario gráfica

Descripción generada automáticamente  
**Esto es incorrecto.** El cruce de datos está mal. Si se percata, **no coinciden los datos de cada uno de los registros al ser unidos, puntualmente son errados los registros 3, 5 & 6 (no hacen falta)**. Planteado lo anterior, lo que corresponde sería considerar un campo adicional, como parámetro, para lograr una unión correcta y *limpia* entre ambas tablas. El parámetro *llave* adicional será ***Store.*** Veamos. **Observe la nueva sentencia de la unión (que ya incluye al campo *Store*).**

**Captura de pantalla de un celular

Descripción generada automáticamente  
 *Resultado***

**Tabla

Descripción generada automáticamente**

## **Joins de datos VS Blend de datos**

Trataremos de explicar las diferencias que hay entre un *Joins de datos* vs un *Blend de datos* y, adicionalmente, en qué escenario uno podría ser más apropiado que otro. Sin embargo, dejemos clara la primera diferencia: **Los *Joins* ocurren en la capa física y los *Blends* ocurren en la capa lógica.**

Esta vez continuaremos trabajando con el *dataset* ‘P1-AmazingMartEU2’; en esta ocasión nos interesa saber, puntualmente, **cuál fue el rendimiento de cada uno de los departamentos para la misma tienda de Europa que estuvimos analizando en la sección anterior:** *‘Mapas, Gráficos de dispersión & Dashboard’*. El rendimiento, **según *departamento****,* pretende ser evaluado tomando en consideración el desempeño **mensual** para todos los años de estudio propios al *dataset*; es decir, **se pretende evaluar el rendimiento (beneficio) de cada departamento** **mes a mes y compararlo con el objetivo de ventas que tienen trazado para ese mismo marco temporal, es decir, para cada mes**.

Adelanto. Será necesaria una ***unión*** entre tablas debido a que los datos de cada una de las ventas por departamento lo posee una tabla, mientras que los datos relacionados a las fechas de estas ventas fueron registrados en otra. Dicho lo anterior, se requiere unir una tabla con otra para poder tener datos completos que responda a nuestra consulta: **evaluar el rendimiento de cada departamento mes a mes y compararlo con el objetivo de ventas que tienen trazado para ese mismo marco temporal, es decir, para cada mes**.

Concretamente, la tabla que posee la información de cada uno de los registros de venta, por cada artículo, para cada departamento es la tabla **‘OrderBreakDown’** y la tabla que posee la información de cada una de las fechas de las ordenes concretadas, que pueden o no poseer muchos artículos dentro de cada orden, es la tabla: **‘ListOfOrders’**. Por lo pronto, esta será nuestra unión. Veamos.



### **Joins con Registros Duplicados**

Esta unión resultante es de tipo: ***Joins con Registros Duplicados***. Explico. Si bien la tabla *ListOfOrders* no depende de campos múltiples para hacer una unión a partir de ella, sino que, cuenta con su propio campo, tipo *llave primaria,* que es **Order ID** donde no existe registro repetido alguno; si bien esto es así, la tabla *OrderBreakDown*, al contrario, como registra las ventas de cada uno de los artículos vendidos, uno por uno sin importar si pertenecen o no a la misma orden, sí es posible que encuentre que varios **Order ID** se repitan en sus registros (porque hay ordenes con más de un artículo y los registros se hacen en términos de *artículos,* no *ordenes*). Entonces, como el campo **Order ID**es quien puede unir ambas tablas (es el campo que tienen en común, el que los relaciona) para que haya una equivalencia entre el número de los registros de cada tabla y queden a la par, al ser unidas, se deben incrementar los registros de la tabla *ListOfOrders* para cada uno de los **Order ID** que se repitan en la tabla *OrderBreakDown*; así hasta que haya una equivalencia en el número de registros de ambas tablas y los registros, de cada tabla, se correspondan y sean consistentes entre sí. Esto es, efectivamente, un ejemplo perfecto del tipo de ***Join con Registros Duplicados***; pues, a la tabla *ListOfOrders* le toca duplicar sus registros al punto de lograr ser equivalente a los registros de la tabla *OrderBreakDown* al ser unidas.

### **Join de tablas con Múltiples Campos**

Ahora, necesitaremos unir nuestra tercera tabla ***SalesTargets*** a la unión de la tabla resultante entre ***ListOfOrders*** + ***OrderBreakDown***, pues es en *SalesTargets* donde se encuentra nuestro *Target por mes* para cada una de nuestras *Categorías*; en consecuencia, **necesitaremos esta última tabla para poder comparar los rendimientos de cada departamento, mes a mes, con el *target* que ellos se fijaron para cada uno de esos meses***.* Luego, es muy probable que esta tercera tabla necesite de *múltiples campos* para lograr una *unión* con las otras; es decir, podría ser esto un ejemplo perfecto de ***Unir tablas con Múltiples Campos.*** Lo explico.

Diferente a la tabla *ListOfOrders,* la tabla *SalesTargets* no cuenta con un campo en específico que haga de *llave primaria*, **esto es así porque ninguno de sus campos podría presumir de que NO posea ningún registro repetido**: en todos sus campos, si los evaluamos uno por uno, verá que hay registros repetidos.

He aquí donde nace la necesidad de valerse de **múltiples campos** de una tabla para hacer una unión con otra; esto para lograr una combinación de valores/datos provenientes de diferentes campos, por cada registro de la misma tabla, donde sí resulte ser único, o haga inigualables, cada registro de la tabla en cuestión. En este caso puntual, para cada registro por separado, se podrían unir los datos del campo ***Month of Order Date*** con los datos del campo ***Category***, convirtiendo así a cada registro de la tabla ***SalesTargets*** en un registro único que conserva, en una eventual unión (*Join*) con otra tabla, una unión consistente y limpia.

Explicado todo esto, si procedemos con la unión de la tercera tabla, sería así:



Sin embargo para este caso puntual, este tipo de unión, el *Join* convencional, haría que las demás tablas se rijan a la granularidad de la última, no por ser la última tabla unida; si no, por ser la tabla con la granularidad más alta de todas las tres; es decir, los datos de la unión de las tablas *ListOfOrders* & *OrderBreakDown* ahora se van a estructurar en términos de los datos de la tabla *SalesTargets,* más puntualmente en términos de los campos ***Month of Order Date & Category***… Esto no es conveniente, se suprimiría la información relacionada a los números de órdenes y, más allá de eso, la información de cada uno de los artículos.

De hecho, la unión se logra mediante el campo ***Category*** (ni siquiera considera a ***Month of Order Date***).



Vamos al grano. Para evitar el anterior conflicto, lo siguiente: toda vez que pretenda unir de manera correcta una tabla con otra, por medio de **múltiples campos**, haga mejor un ***Blending*** en vez de un ***Join*** propiamente***.***

### **Blend de datos**

Por lo pronto tenemos que nuestra unión, desde la actual fuente de datos, quedaría así:

Diagrama

Descripción generada automáticamente

El *Blend* o fundido de datos, para este caso puntual, consiste en ‘unir’ nuestro fichero ***SalesTargets*** a la unión ***ListOfOrders + OrderBreakdown*** sin realmente hacer un ***Join.*** No sería un ***Join*** porque la tabla ***SalesTargets*** necesita unirse **valiéndose de múltiples campos;** con lo cual, así, no sería recomendado hacer un ***Join*** para lograr la *unión* entre las tres tablas en cuestión.

**Cuando una tabla se vale de múltiplos campos para hacer una unión con otra u otras, lo ideal es hacer un Blend de datos.**

Entonces, esta tabla que se vale de múltiplos campos para unirse con otra, necesitamos abrirla desde una fuente de datos completamente nueva desde el mismo Tableau. Veamos.

Vamos a nuestra ***Worksheet*** y hacemos clic en el *cilindro* que hace alusión a las bases de datos ; dado el clic, se nos pide insertar una fuente de datos adicional completamente nueva. La fuente de datos que va a insertar, como se imagina, debe contener dentro de sí al fichero ***SalesTargets*** (para este caso puntual continuamos hablando de ‘P1-AmazingMartEU2’) para poder lograr una unión con las otras tablas por medio del *Blending.* Así quedaría nuestra segunda fuente de datos donde tan sólo se vería arrastrado el campo ***SalesTargets***:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

El total de las diferentes fuentes de datos insertadas se verían así: Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En **conclusión**, y antes de abordar a profundidad el alcance de un recurso tan poderoso como lo es el *Blending* y su forma de trabajo*,* conciba a esta herramienta como un ***Join*** inteligente que puede ocurrir sobre la marcha, no sobre la capa física, sino en base a la lógica (al código); de lo cual, tomaremos ventaja: vamos a crear visualizaciones requeridas usando el *Blending* en lugar del *Joining* para no machacar la estructura física que tienen los datos*.*

En resumidas cuentas no se puede proceder con un ***Join***,en la unión de estas tres tablas, sino a un ***Blend***; para este caso particular no se pudiera hacer un ***Join*** debido a que hubiéramos tenido que *agregar*, o llevar la *Granularidad*,nuestro conjunto de datos resultante ***ListOfOrders + OrderBreakdown*** a la granularidad de la tabla ***SalesTargets***, ya queno puede ser correspondida (son *granularidades* diferentes); en consecuencia, se perdería una gran cantidad de información de ***ListOfOrders + OrderBreakdown*** (éstas en primer lugar sí se correspondieron, la granularidad de ambas sí compaginaron porque contaban con la información necesaria para hacerlo por medio del campo ***Order ID***).

Especialmente le tocará, sí o sí, hacer un ***Blend*** cuando los conjuntos de datos que pretenda unir ni siquiera provengan de una misma fuente de datos; es decir, cuando sepa de entrada que son diferentes y que no necesariamente deban relacionarse. Si los conjuntos de datos son diferentes probablemente también tendrán diferentes granularidades; por lo cual no se recomendaría nunca hacer un ***Join***,si pretende unirlos, sino un ***Blend***.