

# Informe Taller 1: Inteligencia de negocios

Integrantes: Samuel Borraccini - Felipe Figueroa

Profesor: Mauricio Sepúlveda

# Índice

<b>Introducción</b>	<b>3</b>
<b>Requerimientos de gestión</b>	<b>3</b>
<b>KPI</b>	<b>4</b>
KPI-1: Total bruto de ventas con filtro de profundidad	4
Explicación	4
Requerimiento al que responde	4
KPI-2: Productos más populares por estado con filtro de profundidad	4
Explicación	4
Requerimiento al que responde	4
KPI-3: Categorías de productos más populares con filtro de profundidad	5
Explicación	5
Requerimiento al que responde	5
KPI-4: Unidades en descuento adquiridas por producto con filtro de profundidad	5
Explicación	5
Requerimiento al que responde	5
<b>Data Warehouse</b>	<b>5</b>
Granularidad Elegida	6
Modelo de BBDD Multidimensional	6
Dimensiones	7
Dimensión Orders	7
Dimensión Products	7
Dimensión Stores	8
Dimension Date	8
Tabla de hechos	8
<b>Explicación procesos ETL</b>	<b>9</b>
Construcción del staging area	10
Implementación del staging area	10
Carga de dimensiones	10
Dimensión date	10
Dimensiones orders, product y stores	10
Carga tabla de hechos en el staging area	11
Migración del staging area al data warehouse	11
Procesos de carga mensual	12
<b>Cuadro de mando</b>	<b>12</b>
Carga y preparación de herramienta	13
KPI-1	15
Mapa con filtro de profundidad	15
Gráfico de barras apiladas con filtro de profundidad	16

KPI-2	17
TreeMap en profundidad	17
Gráfico de columnas apiladas	19
KPI-3	20
TreeMap	20
Gráfico de barras apiladas con filtro de profundidad	21
KPI-4	23
Mapa	23
Gráfico de barras 100% apiladas	24
<b>Conclusión</b>	<b>26</b>
<b>Anexos</b>	<b>26</b>
Link dashboard PowerBI	26
Backup de data staging y DW multidimensional	27
Repositorio GitHub	27
Guía de ejecución	27
<b>Referencias</b>	<b>27</b>

# Introducción

El siguiente trabajo tiene cómo objetivo simular un proyecto de inteligencia de negocios, mediante el cual aplicar conocimientos adquiridos en clases en un entorno cercano a la realidad.

Un proyecto de inteligencia de negocios tiene busca la generación de valor, información y/o conocimiento, que permitan la toma de decisiones corporativas de manera más informada, con base en datos reales y no corazonadas o juicios personales, permitiendo entregar competencias a una organización.

El taller a desarrollar se basa en el caso de una empresa dedicada a las ventas de bicicletas en Estados Unidos. Para el cual se les ha entregado a los miembros del equipo, una base de datos con la información necesaria para la realización de un diagnóstico de necesidades e implementación BI.

El proyecto BI se generará en base a requerimientos de gestión, que serán definidos por el equipo de trabajo, el desarrollo culminará en la generación de cuadros de mando que permitan obtener información valiosa para la empresa.

## Requerimientos de gestión

Dada la naturaleza formativa que genera el origen de la investigación, no existe un cliente real que demande el estudio de sus datos ni exponga requerimientos para generar información sobre los mismos. Por esto, para lograr simular y establecer requerimientos de gestión que se ajusten lo máximo posible al caso expuesto, se usará el contexto de la empresa que se puede intuir a partir del documento de especificación del taller entregado por el docente y de los mismos datos presentes en el respaldo de la base de datos de la empresa. En resumen la empresa “cliente” que requiere esta investigación se podría considerar una franquicia de venta de bicicletas la cual cuenta con sedes distribuidas dentro de los estados de Texas, California y New York en Estados Unidos, donde comercializa gran variedad de productos relacionados al ciclismo, con distintos descuentos y fabricantes. Habiendo contextualizado un poco sobre la empresa y conociendo a groso modo los datos expuestos por la misma, se decidió generar los siguientes requerimientos de gestión que influyen en la selección de KPI's más adelante:

- Conocer las ventas de las distintas sedes
- Validar efectividad de los descuentos ofrecidos por la compañía
- Definir los gustos de sus clientes
- Comprender los productos de la empresa

# KPI

A continuación se detallarán los KPI's a construir y argumenta el cómo cada uno de estos responden a los requerimientos del cliente. Como punto a tener en cuenta en este apartado, está el comentar que cada KPI será nombrado bajo un código numérico, es decir, cada indicador tendrá el nombre KPI-(algun numero), esta nomenclatura se mantendrá a lo largo de todo el informe, todo esto con la finalidad de evitar nombrar el KPI por su nombre completo el cual suele ser muy extenso y brindaría excesiva longitud al informe de manera innecesaria.

## KPI-1: Total bruto de ventas con filtro de profundidad

### Explicación

Este KPI permitirá conocer el estado actual de la distribución de ventas a lo largo de las sedes establecidas en las distintas locaciones de las cuales es dueña la empresa cliente. Además permitirá identificar rápidamente la tendencia de acuerdo a las tiendas que venden más productos. Apoyando a la administración en obtener conocimiento valioso que permita adoptar medidas organizacionales.

### Requerimiento al que responde

Este KPI permite satisfacer el requerimiento de gestión denominado cómo *“Conocer las ventas de las distintas sedes”*.

## KPI-2: Productos más populares por estado con filtro de profundidad

### Explicación

Cómo equipo desarrollador del proyecto BI creemos que no solo es importante conocer la distribución de acuerdo al monto total de productos vendidos por tienda, sino que para tomar medidas más ajustadas a la realidad puede ser necesario conocer con más detalle cada una de las ventas realizadas. Es por esto que se define este kpi que permitirá conocer cómo se distribuyen las ventas según un producto específico. Para la administración permitirá tener un conocimiento relevante que permita tomar medidas de acuerdo, por ejemplo, a la gestión de distribución de productos, o publicidad que se genera para ciertos productos en algunas locaciones, ya que el KPI permitirá entender qué productos son más relevantes para los clientes en los diferentes estados.

### Requerimiento al que responde

Este KPI permite satisfacer los requerimientos de gestión denominados cómo *“Definir los gustos de sus clientes”* y *“Comprender los productos de la empresa”*.

## KPI-3: Categorías de productos más populares con filtro de profundidad

### Explicación

Cómo complemento del KPI anterior creemos que es necesario ofrecer información más agregada en caso de que sea necesario potenciar las ventas, publicidad o procesos de distribución de productos asociados a una categoría en particular.

### Requerimiento al que responde

Este KPI permite satisfacer los requerimientos de gestión denominados cómo *“Definir los gustos de sus clientes”* y *“Comprender los productos de la empresa”*.

## KPI-4: Unidades en descuento adquiridas por producto con filtro de profundidad

### Explicación

Es importante definir un indicador que permita obtener conocimiento acerca de los descuentos efectuados por producto, además de saber cuánto porcentaje del total de descuento pertenece a cierto producto, cómo la distribución de la cantidad total descontada por estado. Esto puede ser favorable en caso de que la administración quiera tomar medidas con respecto a campañas de descuento o promociones, o simplemente entender cómo se distribuyen los descuentos a lo largo de las tiendas y por producto.

### Requerimiento al que responde

Este KPI permite satisfacer los requerimientos de gestión denominados cómo *“Validar efectividad de los descuentos ofrecidos por la compañía”* y *“Comprender los productos de la empresa”*.

## Data Warehouse

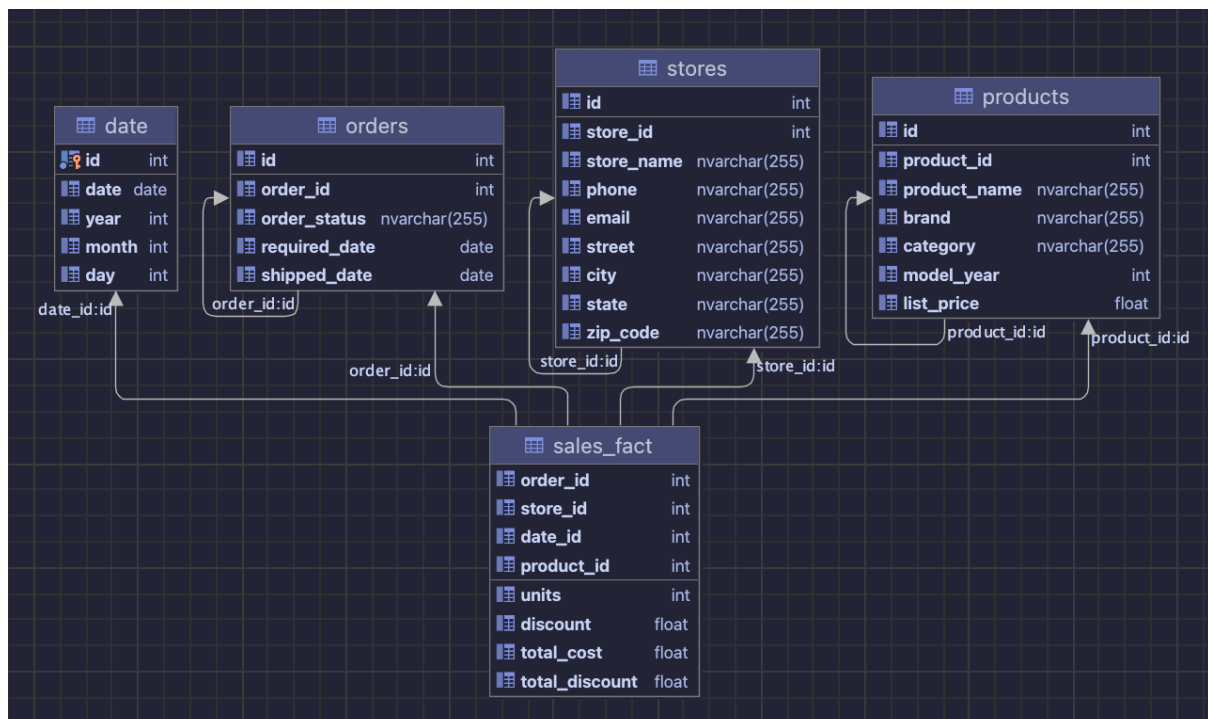
En el apartado que se presenta a continuación se detalla el proceso de razonamiento y resultado generado por el equipo de trabajo al diseñar el modelo multidimensional que contendrá la información necesaria para dar sustento a los KPI's descritos en la sección anterior. Previa a la profundización en el diseño del data warehouse, es pertinente detallar el nivel de granularidad elegido y el modelo multidimensional que se generó como resultado, para tener un punto de referencia visual al momento de explicar el porqué de sus dimensiones y tabla de hechos más adelante, por lo tanto se procederá a detallar esos aspecto previa a la profundización del DW.

## Granularidad Elegida

Antes de detallar la granularidad escogida, se considera importante definir el término en pocas palabras, por lo tanto podemos decir que: La granularidad representa el nivel de detalle presente en los registros de la tabla de hechos. Comprendido el concepto de que es granularidad en el contexto que se está trabajando, se quiere iniciar mencionando que durante este trabajo de BI, el equipo de desarrollo se vio en la necesidad de decidirse por un nivel de detalle sumamente específico, esto debido principalmente al KPI detallado anteriormente que lleva por nombre “Unidades en descuento adquiridas por producto con filtro de profundidad”. Debido a que este KPI requería conocer el detalle de las unidades de cada producto que fueron adquiridas bajo los distintos descuentos disponibilizados por las diversas tiendas de bicicletas. Suponiendo además que los descuentos sobre un producto pueden llegar a variar mensual, semanal o hasta diariamente; Era necesario conocer el detalle del descuento aplicado y cantidades adquiridas con dicho descuento de cada producto en cada una de las órdenes generadas por los distintos establecimientos diariamente. Concluyendo así que la granularidad única viable y escogida por el equipo es que cada registro representará el artículo adquirido y sus cantidades por orden de compra (diaria), para de esta manera controlar y asegurar la veracidad del análisis, ya que los descuentos aplicados pueden variar dentro de distintos lapsos de tiempo para un mismo producto.

## Modelo de BBDD Multidimensional

En este apartado como se mencionó anteriormente únicamente se quiere presentar el modelo diseñado por el equipo de trabajo de manera genérica para tener una figura a la cual referirse en apartados posteriores donde se explicara a detalle cada dimensión y tabla de hechos presentes en el mismo.



**Figura 1: “Modelo multidimensional”**

## Dimensiones

Las dimensiones seleccionadas para formar parte del modelo multidimensional son un total de 4, las cuales serán explicadas de manera individual a continuación.

### Dimensión Orders

Esta dimensión presente en el data warehouse como se puede ver en la figura 1 (imagen del modelo md) posee atributos respectivos a las órdenes de venta generadas por la tienda. Es indispensable para lograr satisfacer el nivel de detalle deseado por el KPI-1, ya que habilita la búsqueda en profundidad por órdenes. Los atributos presentes en esta dimensión son:

- **order\_id**: Este atributo permitirá a la futura tabla de hechos que consuma esta dimensión mantener una relación referencial hacia la misma, esencial para extraer la información.
- **order\_status**: Es un atributo que permitirá conocer el estado de la orden, en la documentación no se especifica el significado del código numérico utilizado por el cliente, pero se consideró importante colocar esta característica de las órdenes en la dimensión, debido al valor que podría generar conocer el estado de procesamiento presente en la orden (completada, atrasada, etc).
- **required\_date**: Atributo destinado a exponer la fecha en que se pidió la entrega del producto, se decidió colocar el mismo debido a que permitirá conocer la cantidad de envíos realizados en distintas fechas, posibles periodos de saturación, etc.
- **shipped\_date**: Similar al atributo anterior, detalla la fecha en la que se generó efectivamente el envío de la orden. Permite expandir el conocimiento generado por el atributo order\_status, ya que uniendolo con el required\_date se pueden conocer días de atraso en pedidos, pedidos que nunca se procesaron (no tienen fecha de envío), etc.

### Dimensión Products

Dimensión la cual aloja toda la información respectiva a los productos que comercializan las distintas tiendas (marca, categoría, modelo, etc). Esta dimension es utilizada para nutrir todos los KPI's definidos (KPI-1 a 4) ya que se utiliza informacion del producto ya sea para asignar codigos de color, agrupar datos cuantitativos, etc. Sus atributos son:

- **product\_id**: Campo necesario para que la tala de hechos pueda mantener relación referencial con la presente dimensión
- **product\_name**: Atributo en cual se especifica el nombre detallado del artículo que represente dicho registro. Habilita filtros por artículos.
- **Brand**: Atributo que permite conocer la marca de cada uno de los artículos. Habilita filtros por distintas marcas
- **category**: Campo que expresa la categoría a la cual pertenece cada unos de los artículos registrados. Habilita filtros por categoría.



- **model\_year**: Atributo que detalla el año correspondiente al modelo del producto. Puede ser utilizado para filtrar productos que sean modelos del mismo año y realizar comparaciones de popularidad entre los mismos, ver cuales modelos de distintos años son más comprados, etc. Habilita filtros por año de modelo.

## Dimensión Stores

Esta dimensión contiene información relevante a las distintas tiendas de las cuales se tiene información y publican ventas en la tabla orders. Dimensión transversal a todos los KPI, ya que esta habilita las búsquedas en profundidad geográfica para todos los KPI definidos (1 al 4) . Sus atributos son:

- **store\_id**: Campo necesario para que la tabla de hechos pueda mantener relación referencial con la presente dimensión
- **store\_name**: Detalla el nombre completo de las distintas tiendas. Habilita filtro por tienda.
- **phone**: Expresa el número de contacto de cada una de las tiendas. Permite generar mayor información y contexto sobre las distintas tiendas.
- **email**: Similar al atributo phone, este campo permite generar mayor contexto y dar más información sobre los registros de la dimensión Stores.
- **street, city, state y zipcode**: Estos tres campos agrupan funcionalidad, al cual se divide en dos: En primer lugar poder habilitar ubicación geográfica de las tiendas, para un potencial uso de mapa interactivo en una herramienta de visualización y en segundo lugar filtros para agrupar información por estado, ciudad, etc.

## Dimension Date

Esta dimensión está totalmente dedicada a contener registros de fechas. Esta dimensión es principalmente utilizada y necesaria por el KPI-1, ya que este expone las ventas (diarias) realizadas por las distintas tiendas, que vienen dados por sus atributos, los cuales son:

- **id**: Campo necesario para que la tabla de hechos pueda mantener relación referencial con la presente dimensión
- **date**: Detalla la fecha completa en formato (YYYY-MM-DD)
- **year, month y day**: Estos tres atributos apuntan a detallar en distintos niveles de granularidad aspectos correspondientes a la fecha de cada uno de los registros de la dimensión: año, mes y día del mes al que representan. Este atributo habilita filtros de agrupación sobre las fechas de órdenes con distintos niveles de granularidad (por mes, día, año).

## Tabla de hechos

Para finalizar este apartado se explicaran los atributos presentes en la tabla de hechos, para evitar entregar información redundante solo se hablara de los atributos acumulados y no de los atributos correspondientes a las dimensiones (FK) ya que estos cumplen la función de habilitar el acceso a cada una de las dimensiones y sus beneficios especificados particularmente en el apartado anterior.

- **units:** Atributo que expresa la cantidad de unidades adquiridas por cada registro (producto) presente en la tabla de hechos. Es indispensable obtener unidades automatizadas haciendo uso del visualizado (PowerBI) y generar campos acumulados presentes en la tabla de hechos que se detallarán más adelante (**total\_cost** y **total\_discount**)
- **discount:** corresponde al descuento asociado a una unidad de producto correspondiente a un registro de la tabla de hechos, se deja expresado para aportar información en caso de ser necesario en futuros *KPI*.
- **total\_cost:** Campo calculado obtenido mediante la operación sobre el precio de lista del producto y las unidades del mismo. Este atributo permite realizar cálculos automatizados dentro del visualizador para acompañar las agrupaciones geográficas presentes en el KPI-1, dotándolas de un valor dependiendo del filtro aplicado (estado, tienda, etc).
- **total\_discount:** Similar al atributo anterior, es obtenido mediante una operación sobre las unidades, precio de lista y descuento de cada producto. Este atributo no es utilizado por ningún KPI actualmente pero se quiso colocar para nutrir de mejor manera la tabla de hechos y dejar propuesta la posibilidad de ser utilizado en un futuro hipotético por la empresa cliente para generar indicadores cuantitativos respecto a la cantidad de dinero descontada en cada venta.

## Explicación procesos ETL

Los procesos de extracción, transformación y carga se realizan de manera manual sin hacer uso de alguna herramienta automática. El equipo de trabajo será el encargado de definir y realizar las consultas *SQL* que permitirán llevar a cabo el proceso. Se decidió hacer de esta manera para generar un mayor conocimiento con respecto a este proceso particular de *Business Intelligence*.

Para llevar a cabo las tareas es necesario primero definir un área de trabajo o *staging area*, la cual permita realizar todas las pruebas de las consultas que se estén construyendo, además de la carga preliminar de las dimensiones y la tabla de hechos, tabla en la cual el equipo deberá probar las consultas y verificar que los procedimientos construidos se ejecutan correctamente. De esta manera se genera un trabajo más limpio y modularizado, que permita separar los ambientes de trabajo y producción. Luego de las pruebas realizadas y construcción de los procedimientos, dimensiones y tabla de hechos, se dará paso a la migración del modelo al data warehouse en producción, a través del cual se podrá habilitar la información que será consumida por una herramienta de *Business Intelligence*, para este trabajo en particular *PowerBI*.

Cabe destacar, que dada una decisión técnica el equipo determina que las áreas de trabajo (*staging area*) y el propio *data warehouse* se implementarán como *schemas* independientes que pertenecen a una misma base de datos.

## Construcción del staging area

### Implementación del staging area

Una vez se tiene diseñado el modelo, se da paso a la implementación de este en el staging área.

Se desarrolló un script encargado de la implementación del modelo, a través del cual se crean los schemas, dimensiones y tabla de hechos, añadiendo las relaciones a través de una clave foránea, la cual será la encargada de asociar un registro en una dimensión, con un registro en la tabla de hechos, esta relación es la que permitirá generar filtros en base a otros atributos de la dimensión para las consultas que se deseen generar en el *data warehouse*.

**Primera ejecución de script →**

[https://github.com/FelipeAFV/taller1-bigdata/blob/master/create\\_stagging\\_area.sql](https://github.com/FelipeAFV/taller1-bigdata/blob/master/create_stagging_area.sql)

### Carga de dimensiones

#### Dimensión *date*

Luego de la implementación del modelo se da paso a la creación de las dimensiones. La primera dimensión cargada es la dimensión *date*, para este paso se creó un procedimiento parametrizado que permite la carga de datos en la dimensión *date*, el procedimiento recibe como parámetros la fecha de inicio y fin del periodo en el cual se desean cargar los datos.

**Segunda ejecución de script →**

[https://github.com/FelipeAFV/taller1-bigdata/blob/master/dates\\_load2.sql](https://github.com/FelipeAFV/taller1-bigdata/blob/master/dates_load2.sql)

El *script* genera un procedimiento que debe ser ejecutado con el siguiente comando, para la carga de la dimensión fechas para el periodo 2016.

1. 

```
exec staging area.dateLoad @StartDate = '2016-01-01', @EndDate = '2016-12-31'
```

#### Dimensiones *orders*, *product* y *stores*

Para la carga de las demás dimensiones se crearon procedimientos almacenados, encargados de llevar los datos de la base de datos original hacia el *staging area*. Para este caso particular el equipo de trabajo determinó que no era necesario un proceso de limpieza de los datos ya que al revisarlos se encontró que solo dos campos de la base de datos original poseían campos nulos y para el equipo estos campos no eran relevantes para etapas posteriores, por lo que determina dejar los campos nulos ya que no afectarían las consultas que se realizarán posteriormente en el *data warehouse*. Cabe destacar que el proceso de carga de las dimensiones se corresponde con la etapa de extracción, ya que se deben obtener los datos desde la base de datos original.

**Tercera ejecución de script →**

<https://github.com/FelipeAFV/taller1-bigdata/blob/master/loadProcedures.sql>

El *script* crea los procedimientos de carga de dimensiones, los cuales deben ser ejecutados con los siguientes comandos.

1. `exec staging_area.loadStoresData`
2. `exec staging_area.loadProductsData`
3. `exec staging_area.loadOrdersData`

**Carga tabla de hechos en el *staging area***

Una vez cargadas las dimensiones se puede proceder a la carga de la tabla de hechos. Se generó una consulta *sql* para la carga inicial, la cual contempla la carga de todas las órdenes actuales cargadas en la dimensión *orders*, además de la creación de un procedimiento para la carga mensual de los datos, el cual contempla la creación de los nuevos registros en la dimensión *date* además de la carga de los registros a la tabla de hechos. El proceso de carga de datos a la tabla de hechos involucra el cálculo de datos acumulados, específicamente los campos *total\_cost* y *total\_discount*, este proceso de cálculo se corresponde con el proceso de transformación, ya que estamos modificando los datos originales, preparando para consultas optimizadas por parte de la herramienta de *business intelligence*.

**Cuarta ejecución de script - script 4 →**

[https://github.com/FelipeAFV/taller1-bigdata/blob/master/dw\\_load\\_2.sql](https://github.com/FelipeAFV/taller1-bigdata/blob/master/dw_load_2.sql)

El *script* crea los procedimientos que deben ser ejecutados con los siguientes comandos.

1. Se cargan los datos iniciales a la tabla de hechos del *staging area*, datos año 2016.  
`exec staging_area.loadInitialDwData`
2. Ejecución de ejemplo para la carga de datos mensuales con el procedimiento creado en el script 4.  
`exec staging_area.loadMonthlyData @StartDate = '2017-01-01', @EndDate = '2017-01-31'`

**Migración del *staging area* al data warehouse**

Una vez cargadas las dimensiones y la tabla de hechos, se corrobora el correcto funcionamiento de las queries y los procedimientos asociados al proceso de extracción y transformación, por lo que se puede dar paso al proceso de carga, el cual para este caso particular consiste en la migración inicial del esquema multidimensional del *staging area* al *data warehouse*.

**Quinta ejecución de script - script 5 →**

[https://github.com/FelipeAFV/taller1-bigdata/blob/master/initial\\_dw\\_load.sql](https://github.com/FelipeAFV/taller1-bigdata/blob/master/initial_dw_load.sql)

## Procesos de carga mensual

Para la mantención se tienen contemplados dos procedimientos almacenados, el mencionado anteriormente encargado de carga mensual de datos en el *staging area* y un nuevo procedimiento de carga de esos datos al *data warehouse*, el procedimiento de carga al *data warehouse* contempla la creación de nuevos registros en las dimensiones *date* y *orders*.

### Sexta ejecución de script - script 6 →

[https://github.com/FelipeAFV/taller1-bigdata/blob/master/monthly\\_dw\\_load.sql](https://github.com/FelipeAFV/taller1-bigdata/blob/master/monthly_dw_load.sql)

1. Primero se procede con la carga de los datos de la base de datos original al *staging area* en el periodo mensual seleccionado, esto permite segregar los procesos evitando errores en el *data warehouse*.

```
exec staging_area.loadMonthlyData @StartDate = '2017-02-01', @EndDate = '2017-02-28'
```

2. Luego, cuando el proceso de carga haya sido exitoso se procede a ejecutar el procedimiento de migración de los datos al *data warehouse*, para el mismo periodo con el cual se cargaron los datos, esto permite asegurar que los datos nuevos para las dimensiones *orders* y *date* ya fueron cargados y están listos para ser migrados de ambiente.

```
exec dw.loadMonthlyData @StartDate = '2017-02-01', @EndDate = '2017-02-28'
```

## Cuadro de mando

Durante este apartado se explicará a detalle las secciones que componen el cuadro de mando, agrupando las vistas presentes en el mismo con el KPI que representan (se llamarán a los KPI por sus códigos identificadores definidos en secciones anteriores). Para finalmente exponer el resultado generado en la plataforma de visualización. Cabe acotar que el equipo de investigación decidió utilizar la herramienta PoweBI de microsoft para la visualización, principalmente por tener conocimientos previos sobre el uso de la misma y su facilidad de interacción con bases de datos SQL server, siendo esta última donde se genero el DW de la solución.

## Carga y preparación de herramienta

Como etapa inicial al proceso de creación de cuadro de mando, se debió llevar toda la información contenida en el DataWarehouse creado hacia la herramienta PowerBI y realizar configuraciones dentro de la misma para el correcto uso de la data.

Para generar la carga de la información hacia la herramienta se hizo uso de una query ejecutada dentro de la herramienta mediante conexión local con la base de datos. Dicha query permitirá obtener los registros de la tabla de hechos y las relaciones pertinentes necesarias para la creación de los indicadores de rendimiento. La query ejecutada fue la siguiente:

```
select orders.order_id ,date.date, prod.product_name, prod.category, salesf.units
,salesf.discount, stores.store_name, stores.street, stores.city, stores.state,
salesf.total_cost, salesf.total_discount from dw.sales_fact as salesf inner join
dw.orders as orders on salesf.order_id=orders.order_id inner join dw.date as date on
salesf.date_id=date.id inner join dw.products as prod on salesf.product_id=prod.id
inner join dw.stores as stores on salesf.store_id=stores.id where year(date.date) =
2016
```

Una vez ejecutada dicha consulta los datos se cargaron exitosamente en la herramienta, generando el siguiente dataset:

order_id	date	product_name	category	units	discount	store_name	street	city	state	total_cost	total_discount
2	01-01-2016 0:00:00	Electra Townie Original 7D EQ - 2016	Cruisers Bicycles	2	0,05	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	1139,981	59,999
2	01-01-2016 0:00:00	Electra Townie Original 7D EQ - Women's - 2016	Cruisers Bicycles	1	0,07	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	557,9907	41,9993
3	02-01-2016 0:00:00	Electra Townie Original 7D EQ - Women's - 2016	Cruisers Bicycles	1	0,05	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	569,9905	29,9995
5	03-01-2016 0:00:00	Pure Cycles Vine 8-Speed - 2016	Cruisers Bicycles	1	0,07	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	398,97	30,03
6	04-01-2016 0:00:00	Electra Townie Original 21D - 2016	Cruisers Bicycles	2	0,05	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	1044,981	54,999
6	04-01-2016 0:00:00	Pure Cycles Western 3-Speed - Women's - 2015/2016	Cruisers Bicycles	1	0,07	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	417,57	31,43
6	04-01-2016 0:00:00	Electra Townie Original 7D EQ - Women's - 2016	Cruisers Bicycles	1	0,1	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	539,991	59,999
7	04-01-2016 0:00:00	Electra Moto 1 - 2016	Cruisers Bicycles	1	0,07	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	492,8907	37,0993
7	04-01-2016 0:00:00	Pure Cycles Vine 8-Speed - 2016	Cruisers Bicycles	2	0,1	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	772,2	85,8
8	04-01-2016 0:00:00	Electra Townie Original 7D EQ - Women's - 2016	Cruisers Bicycles	2	0,07	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	1115,9814	83,9986
10	05-01-2016 0:00:00	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	Cruisers Bicycles	1	0,1	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	242,991	26,999
11	05-01-2016 0:00:00	Electra Townie Original 7D EQ - 2016	Cruisers Bicycles	2	0,2	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	959,984	239,996
13	08-01-2016 0:00:00	Electra Cruiser 1 (24-inch) - 2016	Cruisers Bicycles	1	0,1	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	242,991	26,999
13	08-01-2016 0:00:00	Electra Townie Original 7D EQ - 2016	Cruisers Bicycles	2	0,05	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	1139,981	59,999
13	08-01-2016 0:00:00	Pure Cycles Vine 8-Speed - 2016	Cruisers Bicycles	2	0,05	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	815,1	42,9
13	08-01-2016 0:00:00	Electra Townie Original 7D EQ - Women's - 2016	Cruisers Bicycles	2	0,1	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	1079,982	119,998
15	09-01-2016 0:00:00	Electra Townie Original 21D - 2016	Cruisers Bicycles	2	0,07	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	1022,9814	76,9986
15	09-01-2016 0:00:00	Pure Cycles Western 3-Speed - Women's - 2015/2016	Cruisers Bicycles	2	0,05	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	853,1	44,9
21	15-01-2016 0:00:00	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	Cruisers Bicycles	1	0,05	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	256,4905	13,4995
21	15-01-2016 0:00:00	Pure Cycles Vine 8-Speed - 2016	Cruisers Bicycles	1	0,1	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	386,1	42,9
24	18-01-2016 0:00:00	Pure Cycles Western 3-Speed - Women's - 2015/2016	Cruisers Bicycles	2	0,07	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	835,14	62,86
25	18-01-2016 0:00:00	Electra Girl's Hawaii 1 (16-inch) - 2015/2016	Cruisers Bicycles	1	0,2	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	215,992	53,998
26	18-01-2016 0:00:00	Electra Townie Original 21D - 2016	Cruisers Bicycles	1	0,2	Baldwin Bikes	4200 Chestnut Lane	Baldwin	NY	439,992	109,998

Una vez el dataset se encuentra generado en la herramienta, fue necesario ajustar parámetros respecto a cómo la PowerBI interpretaba los valores presentes en los registros, los cambios realizados sobre las distintas columnas fueron:

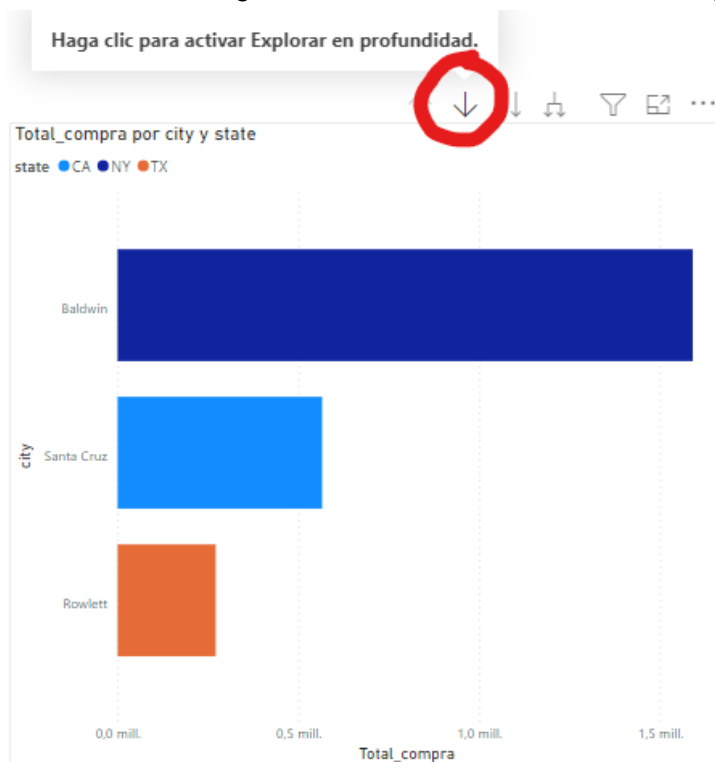
- Order\_id: Se debió eliminar el resumen por suma que configurar automáticamente la herramienta, para permitir ocupar el campo como una categoría y no un valor.
- Discount: Similar a order\_id, se debió eliminar el resumen por suma definido automáticamente por PowerBI, para poder usar este campo como filtro y no como valor acumulativo.

- Street, City y State: Estos tres campos debieron ser modificados con el tipo de dato dirección, ciudad y estado respectivamente, para habilitar a la herramienta el uso de estos campos como direcciones reales en mapas.

Con estos ajustes concluyen todos los pasos preliminares realizados para preparar y ajustar la data dentro de la herramienta de visualización.

Antes de iniciar con el detalle del cuadro de mando, el equipo encuentra pertinente aclarar dos aspectos que están presentes en la mayoría de las vistas del cuadro de mando.

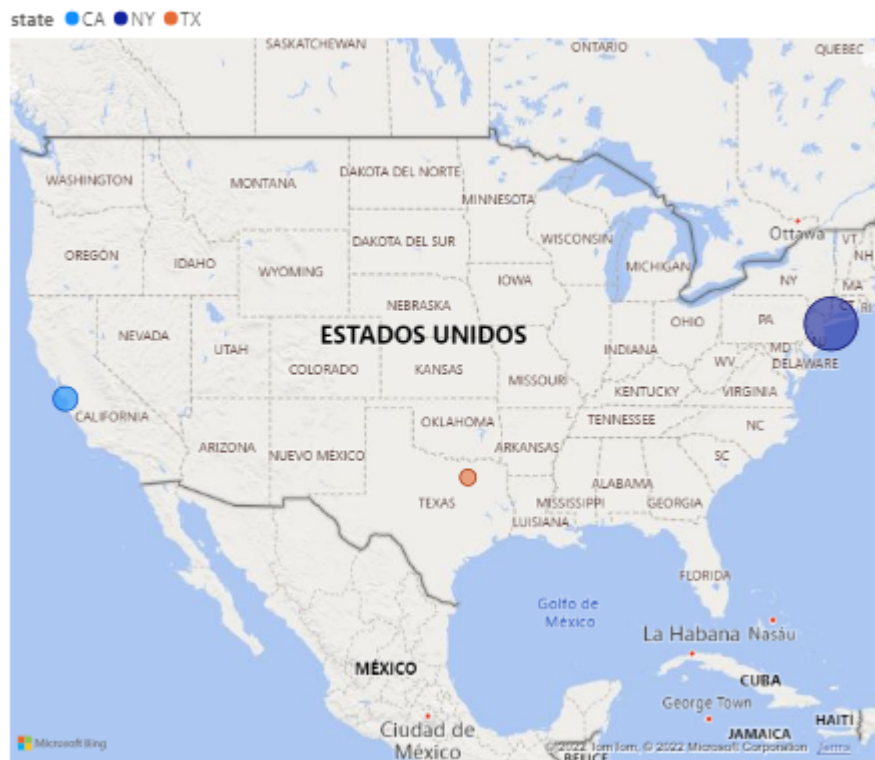
La primera es informar al lector que las vistas que se definan con las palabras “**claves con filtro de profundidad**” notifican la existencia de interactividad clickeable más allá de un simple focus en el objeto, estas vistas permiten obtener mayor especificidad de detalle a medida se interactúa con la misma, para activar este modo de búsqueda en profundidad se debe clicar el siguiente boton dentro del tablero compartido de PoweBI:



Y el segundo aspecto a aclarar es que debido a esta característica de búsqueda en profundidad, cada vista genera muchas sub-vistas por lo que para mantener el presente informe lo más conciso posible al momento de explicar cada vista, solo se añadirá captura de pantalla de la vista sin filtro de profundidad aplicado y la sub-vista generada por el primer nivel de profundidad.

## KPI-1

### Mapa con filtro de profundidad



Como se puede ver en la imagen esta vista presenta al usuario un mapa con tres ubicaciones que cuentan con un código de color respectivo a su estado, el tamaño de las burbujas presente en cada localización expresa la relación sobre la cantidad de ingresos brutos por ventas entre las distintas ubicaciones (mayor tamaño más ventas y viceversa). A continuación se presenta el primer nivel de profundidad.



Total\_cost por city, state y state

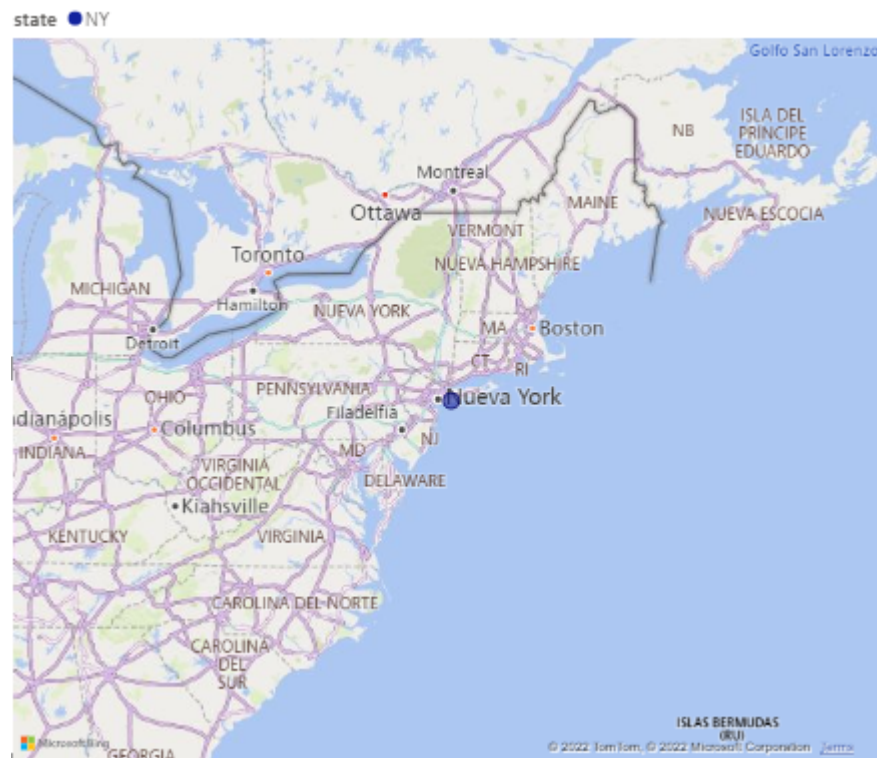
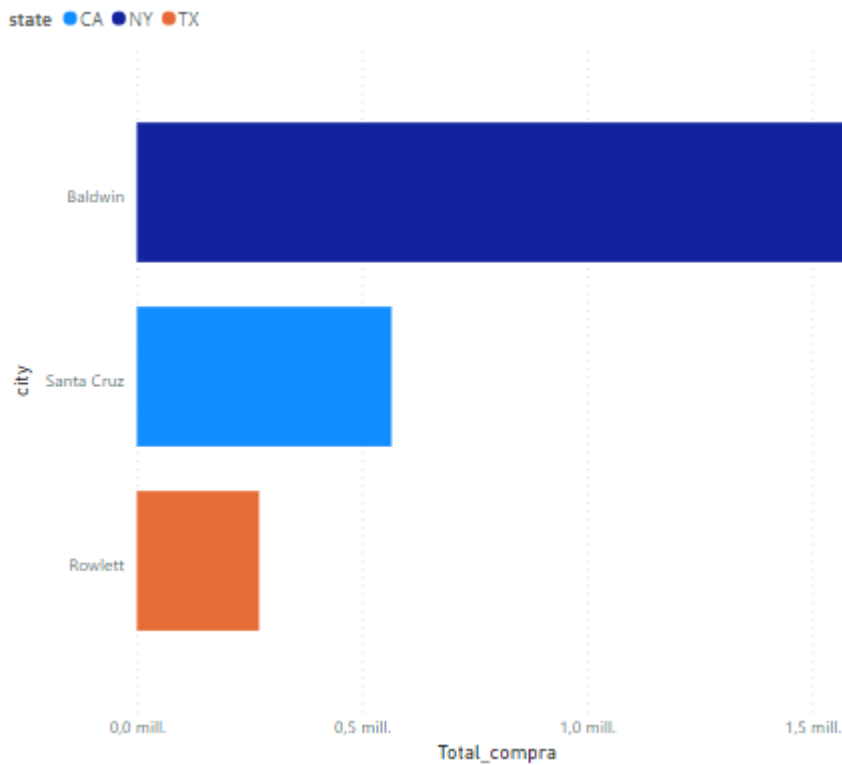
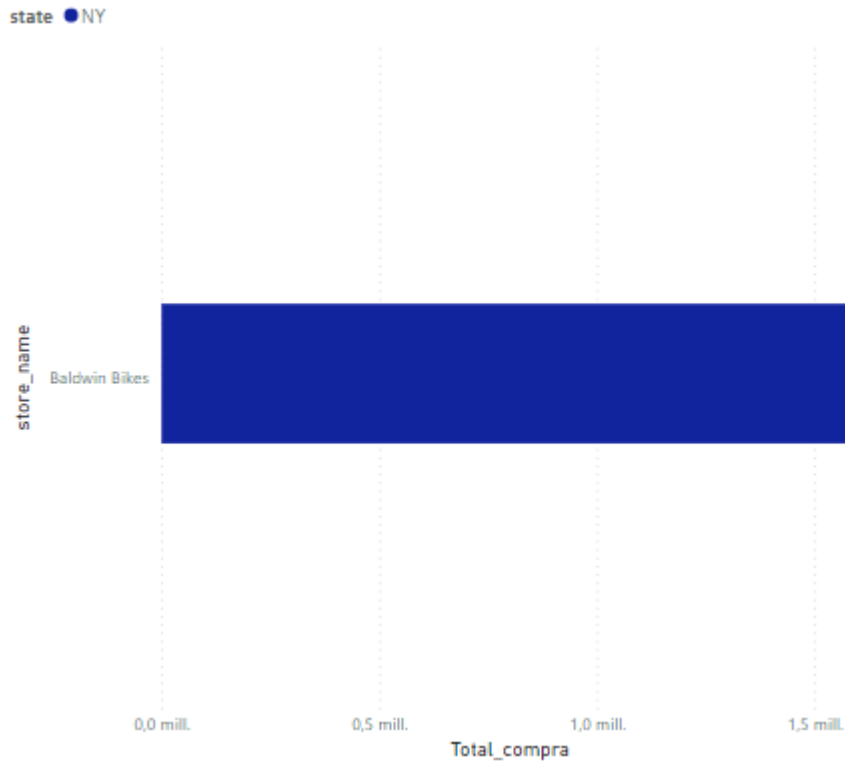


Gráfico de barras apiladas con filtro de profundidad

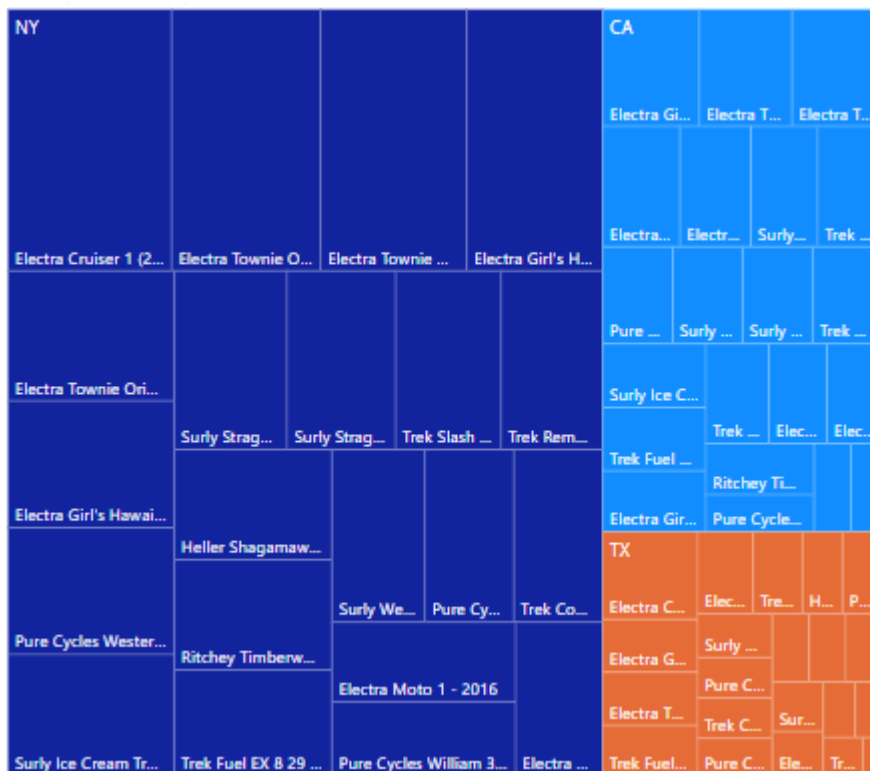


La vista representa un grafico de barras agrupado por ciudad (eje y) y con codigo de color segun el estado (leyenda), donde el progreso de cada barra repreenta los ingresos brutos de cada una. El primer nivel de profundidad asociada a esta vista es el siguiente.



## KPI-2

### TreeMap en profundidad



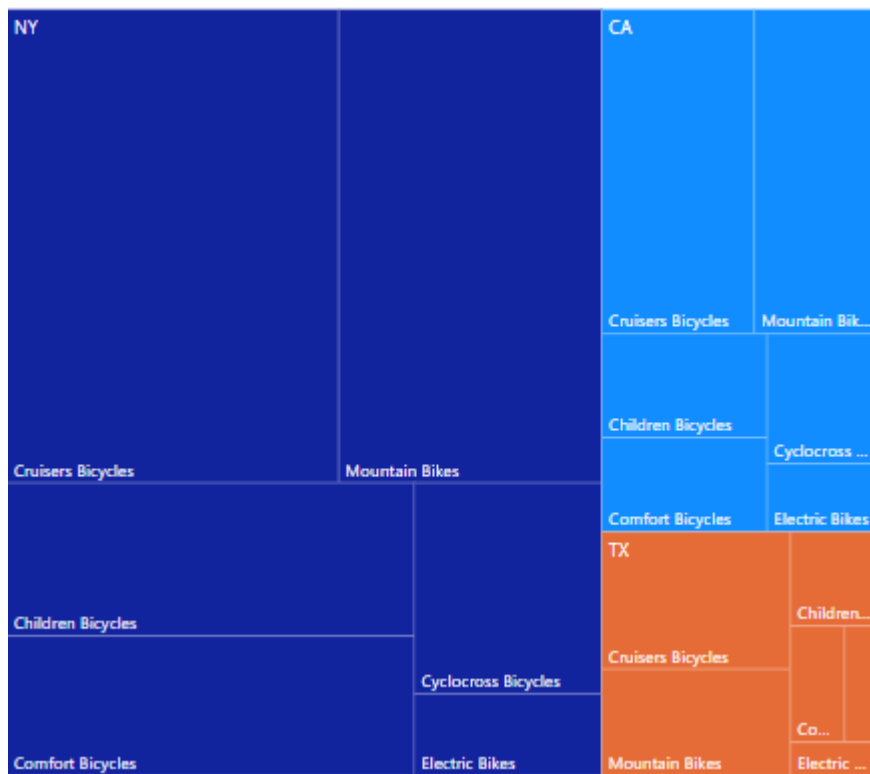
Esta vista presenta un mapa de arbol en el esta compuesto por rectangulos de distintos tamaños los cuales representan la cantidad de cada articulo comprado (mayor tamaño mayor cantidad con respecto a el resto de articulos), ademeas se agrupan por codigo de color dependiendo del estado al que pertenzcan. El primer nivel de profundidad del mapa es el siguiente.

Baldwin				
Electra Cruiser 1 (24-Inch) - 2016	Electra Townie O...	Electra Girl's Ha...	Pure Cycles We...	Surly Ice Crea...
Electra Townie Original 21D - 2016	Surly Straggler - 2016			
Electra Townie Original 7D EQ - 2016	Surly Straggler 650b - 2...	Heller Shaga...	Ritchey Timb...	Trek Fuel EX ...
		Surly Wednesday Fram...		
	Trek Slash 8 27.5 - 2016	Pure Cycles Vine 8-Spe...	Electra ...	Pure Cy...
Electra Girl's Hawaii 1 (16-inch) - 201...	Trek Remedy 29 Carbon...	Trek Conduit+ - 2016	Electra Townie Origi...	



## KPI-3

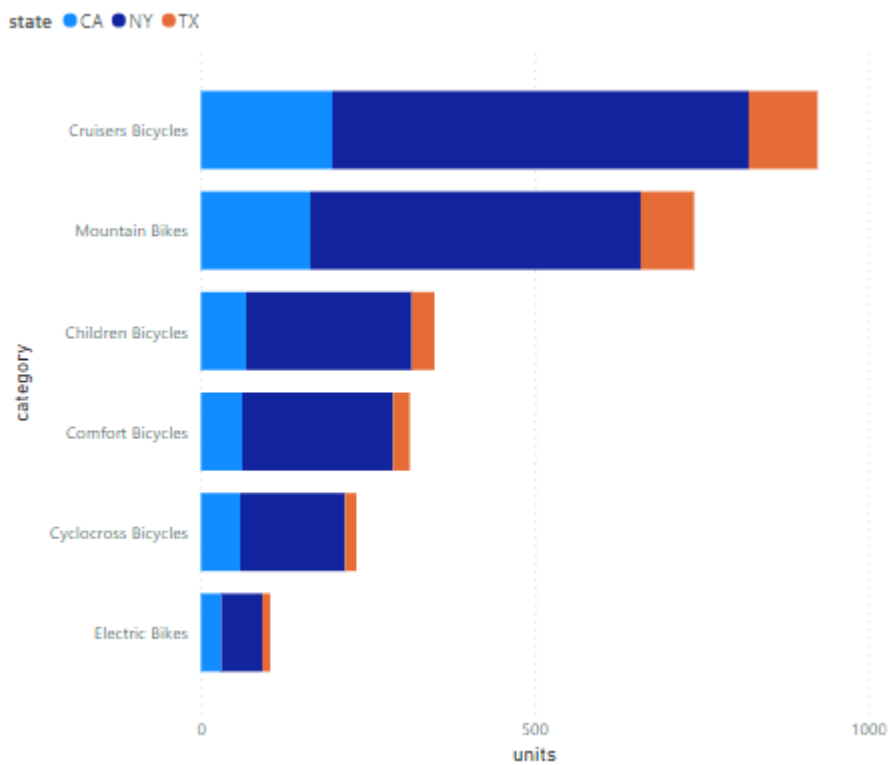
### TreeMap



Esta vista es similar al treemap presente en el KPI-2, pero tiene distinto foco. El presente mapa de árbol está compuesto por rectángulos que representan categorías de productos, los cuales varían en tamaño según las unidades de dicha categoría adquiridas. La agrupación del mapa está ejecutada por estados. El primer nivel de profundidad de la vista es el siguiente.

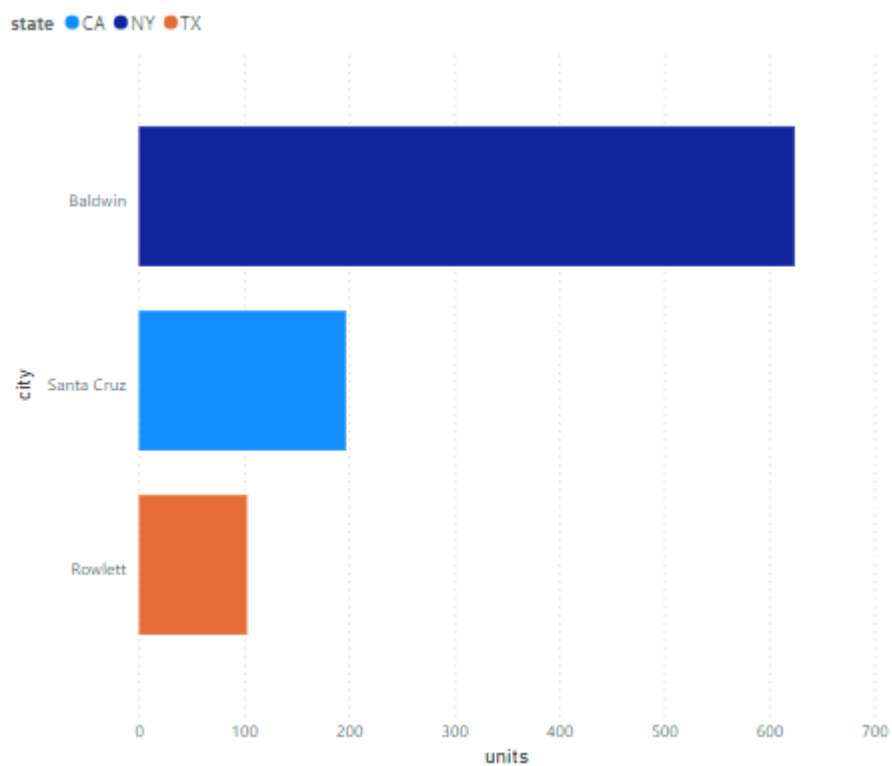


Gráfico de barras apiladas con filtro de profundidad



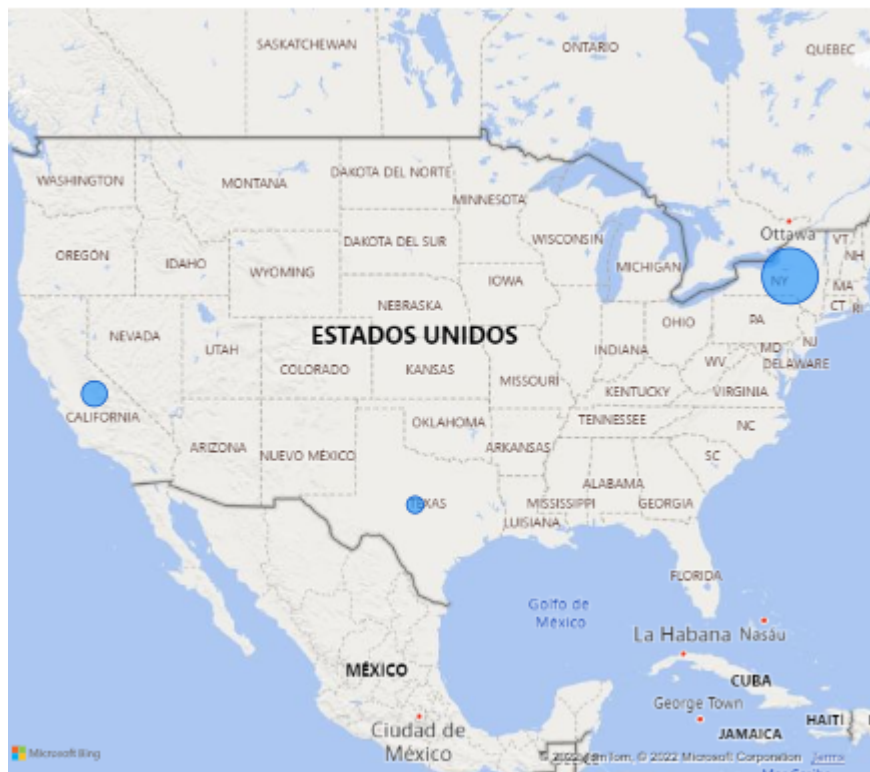
La vista expone un grafico de barras agrupado por categorias de producto (eje y) y con codigo de color segun el estado (leyenda), donde el progreso de cada barra repreenta las

unidades totales vendidas de cada una. El primer nivel de profundidad asociada a esta vista es el siguiente.



## KPI-4

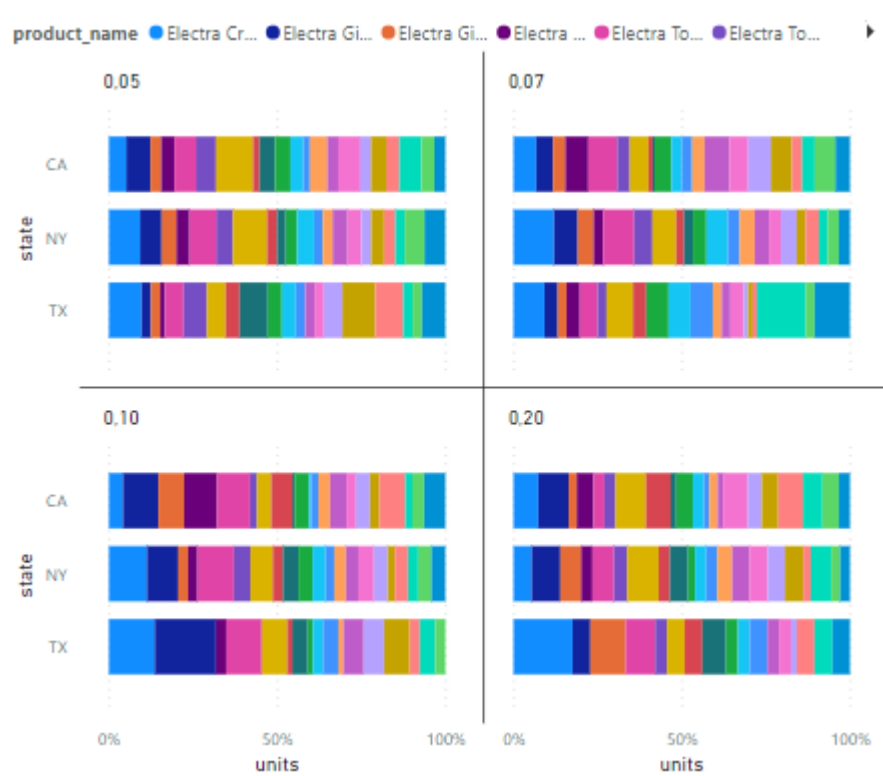
### Mapa



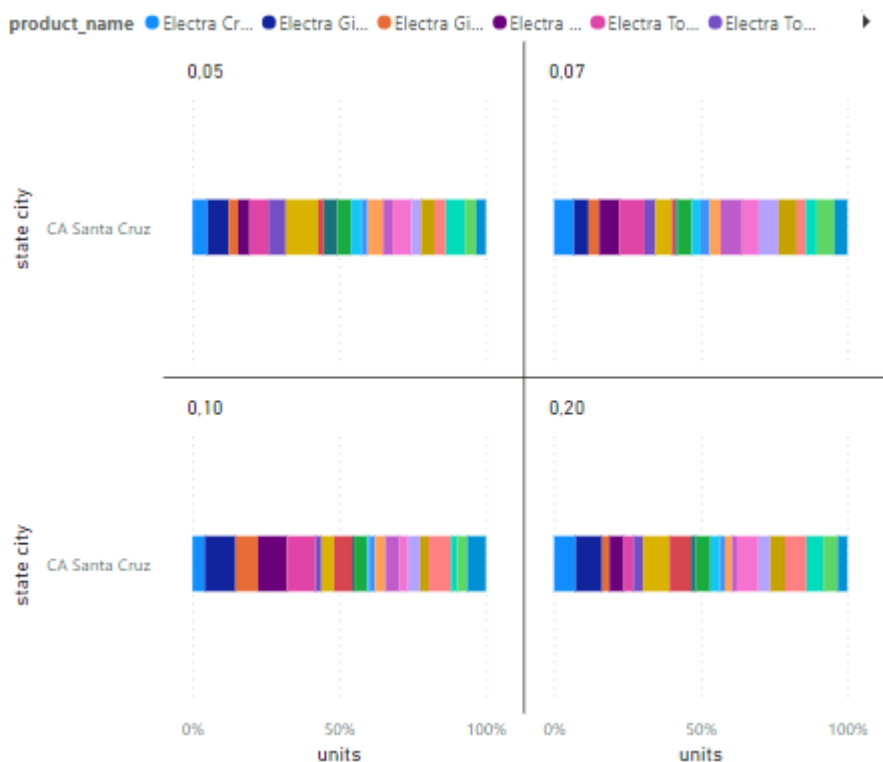
Esta vista presenta al usuario un mapa con tres ubicaciones, en las cuales el tamaño de las burbujas presente en cada localización expresa la relación sobre la cantidad de dinero descontado por ventas (mayor tamaño más dinero descontado y viceversa).



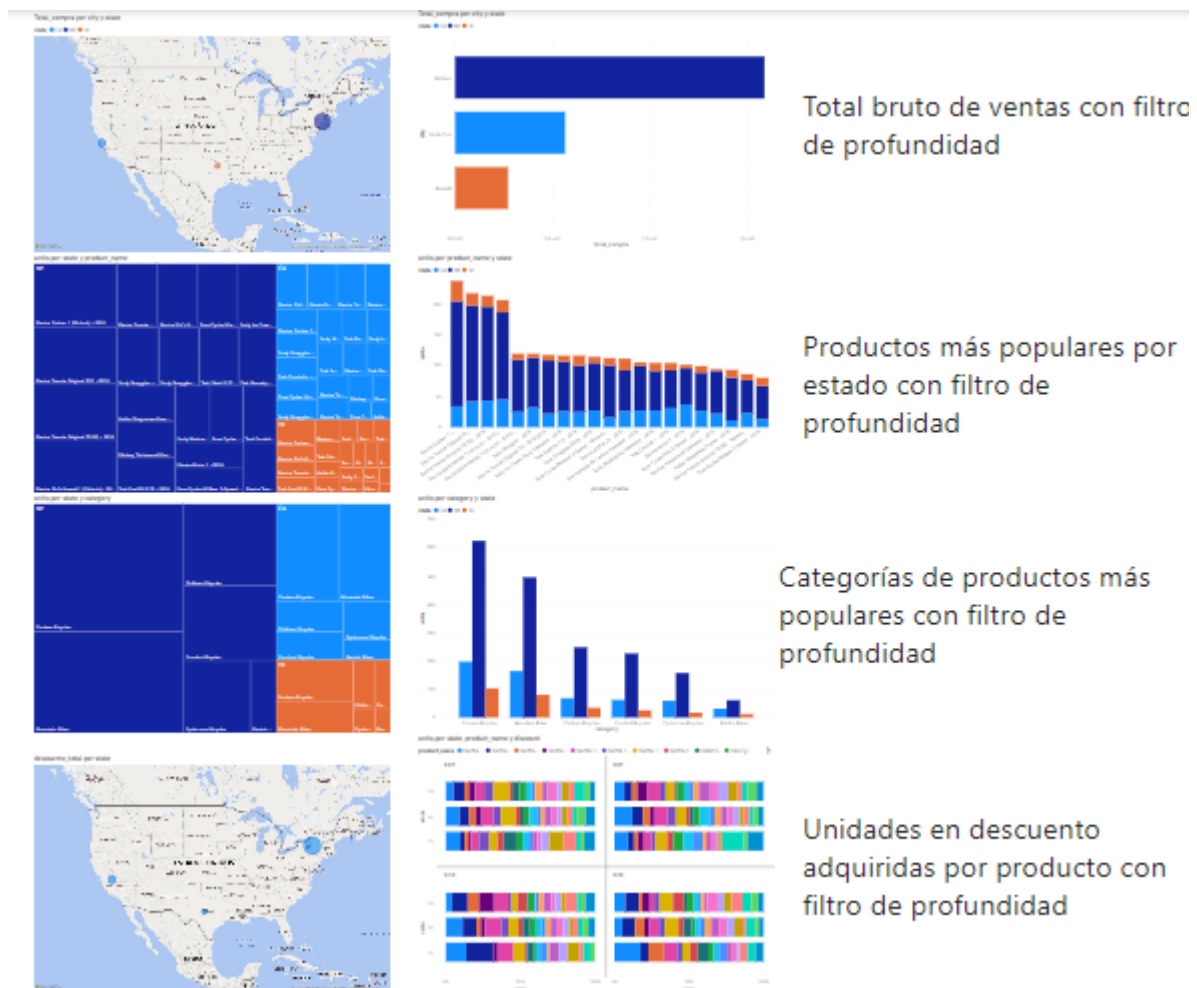
## Gráfico de barras 100% apiladas



Esta visualización representa por cuadrantes dependiendo del descuento aplicado, la cantidad de unidades (progreso horizontal de cada barra) vendidas por cada artículo (código de color) haciendo uso de un descuento, con un nivel de agrupación por estado. El primer nivel de profundidad generado por la vista es el siguiente.



Para finalizar el apartado de cuadro de mando se quiere comentar un aspecto que igualmente será visible por el docente una vez interactúe con la solución publicada en PowerBI, el cual es que todas las visualizaciones detalladas anteriormente y por ende el cuadro de mando interactúa de manera enlazada, es decir al clicar en una visualización las demás responden de manera sincrónica ajustando su nivel de especificidad. Esto es gracias a la forma en que fueron construidas las vistas por parte del equipo investigador, manteniendo consistencia en especificidad a lo largo de las mismas. A continuación se presenta la vista integral generada o cuadro de mando, debido a la gran cantidad de indicadores ajustados en una unica pagina, causa que el tamaño no sea optimo de visualizar, para esto existen dos soluciones: La primera hacer uso de la función pantalla focus ofrecida por PowerBI, la cual permite aumentar el tamaño de una visualización y la segunda es dirigirse a los distintos apartados que fueron dejados en el documento como pestañas extra, donde se exponen únicamente dos vistas por página, siendo mucho más fácil de visualizar.



## Conclusión

Como palabras finales para el presente informe en primer lugar se quiere afirmar con seguridad que los objetivos planteados por el docente para la realización del mismo fueron logrados, bajo la opinión del equipo de desarrollo. Se generó un modelo de datos multidimensional acorde a los requerimientos organizacionales planteados, el cual no quedó meramente en el plano teórico, sino que fue cargado de información compilada y transformada, haciendo una correcta ejecución de los métodos vistos en clases. Para posteriormente cerrar el taller de manera exitosa generando visualizaciones que cumplieran con las buenas prácticas estudiadas, entre las cuales destacan interactividad, colores limitados, uso eficiente del espacio ,etc. En pocas palabras, el proyecto de BI establecido para apoyar a la franquicia de tiendas de bicicletas fue completado con éxito. De igual forma los investigadores miembros del equipo de trabajo quieren hacer uso de este apartado para reflexionar sobre los aspectos que resultaron más resaltantes a los investigadores del equipo de trabajo y los aprendizajes obtenidos a lo largo de todo el desarrollo del proceso de BI. En primer lugar se logró conocer de primera mano lo exhaustivo y representativo de todo el proyecto BI que es la ETL, los investigadores lograron comprender y experimentar cómo este proceso efectivamente representa el 80% de cualquier proyecto BI, y más importante que esto se comprende que es un proceso crítico para el éxito del proyecto ya que un error en cualquier paso del mismo puede generar la carga de datos erróneos al DW, que posteriormente dificulten o aún peor detengan por completo cualquier tipo de visualización o análisis que se quiera hacer sobre los mismos, ya sea por datos faltantes, erróneos o inconsistentes. Otro aspecto que generó aprendizaje importante para el equipo de trabajo es el comprender la potencia de las herramientas de visualización, en el caso específico de este proyecto PowerBI, a pesar de contar con una pequeña experiencia previa e idea sobre el funcionamiento del software, a lo largo del desarrollo los investigadores pudieron darse cuenta de el nivel de automatización y reconocimiento de datos presente en esta herramienta; Está no solo es capaz de permitir visualizar los datos, sino que también los interpreta, y utiliza de distintas maneras dependiendo de esto último. Por ejemplo automáticamente reconoce direcciones del mundo real y es capaz de ubicarlas vía GPS, resume columnas de manera automática para facilitar el filtrado y creación de vistas, y muchas otros aspectos más que resaltan la capacidad de esta tecnología. Es decir este proyecto fue sumamente nutritivo para ambos miembros del equipo investigador, generando y reforzando conocimiento en los mismos, el cual sin lugar a dudas será de importante en su futuro como profesionales del área informática.

## Anexos

### Link dashboard PowerBI

<https://app.powerbi.com/links/-PYDPNiWfV?ctid=7a599002-001c-432c-846e-1ddca9f6b299&pbisource=linkShare&bookmarkGuid=ec9702e3-7b43-4a16-a3d6-571fc25c7fbd>

## Backup de data staging y DW multidimensional

[https://github.com/FelipeAFV/taller1-bigdata/blob/master/taller1\\_BI.bak](https://github.com/FelipeAFV/taller1-bigdata/blob/master/taller1_BI.bak)

## Repositorio GitHub

<https://github.com/FelipeAFV/taller1-bigdata>

## Guía de ejecución

Para poder simular los procesos ETL a continuación se explicarán los pasos necesarios para recrear estos procedimientos y simular el trabajo realizado en este informe.

1. Restaurar la base de datos original desde el siguiente archivo  
<https://github.com/FelipeAFV/taller1-bigdata/blob/master/BikeStores.bak>
2. Ejecutar los scripts definidos en la sección “Explicación procesos ETL”:
  - a. Ejecutar paso a paso cada uno de los scripts de creación de procedimientos linkeados a Github, en conjunto con los comandos de ejecución de estos procedimientos descritos en la misma sección del informe (**se recalca la importancia de que estos deben ser ejecutados en el orden en que están descritos, de otra forma la creación no se ejecutará de manera correcta**).

**NOTA:** En caso de querer acceder directamente al resultado final obtenido como resultado de seguir la ejecución, se incluye en el github el siguiente archivo  
[https://github.com/FelipeAFV/taller1-bigdata/blob/4edbae77bb4374f9b3b7d5070e5746a9c6efce4a/taller1\\_BI.bak](https://github.com/FelipeAFV/taller1-bigdata/blob/4edbae77bb4374f9b3b7d5070e5746a9c6efce4a/taller1_BI.bak). El cual contiene el backup de la base de datos resultado de los procesos anteriores, únicamente es necesario restaurarla.

## Referencias

- M. (2021, 18 diciembre). Tutorials for SQL Server - SQL Server. Microsoft Docs.  
<https://docs.microsoft.com/en-us/sql/sql-server/tutorials-for-sql-server-2016?view=sql-server-ver15>
- Sherif, A. S. (2016). Practical Business Intelligence (1st edition). Van Haren Publishing.
- W. (2022, 28 januari). *Create a Stored Procedure - SQL Server*. Microsoft Docs. Geraadpleegd op 4 mei 2022, van

<https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/create-a-stored-procedure?view=sql-server-ver15>