**Informe inteligencia de negocios**

**José Vargas**

Profesor

Mauricio Sepúlveda

Curso

Minería de datos y big data

**11 de mayo del 2022**

# Índice

[**Índice**](#_pnpytn34wp0t) **2**

[**Introducción**](#_1b85lldc9mau) **3**

[Objetivo](#_6r82pasc3cha) 3

[**Marco teórico**](#_1jj692uo82xx) **3**

[¿Qué es Business Intelligence (BI)?](#_5pkli51drc94) 3

[¿Qué es un datawarehouse?](#_yvp6p47wwjsx) 3

[**Entorno del negocio**](#_5cq9f6bywmdh) **4**

[Estado actual](#_mxl8wot4fgyz) 4

[Área a elegir de la empresa](#_x31liaah686u) 4

[**Diseño de un Data Warehouse**](#_tzy2g1jp8xwv) **4**

[Requerimientos](#_xskdcwg7qlt8) 4

[Preguntas de gestión](#_uqum0squ4rfz) 5

[Identificación de KPIs y dimensiones](#_n3bxnwd6y7iy) 5

[Análisis de procesos OLTP](#_x9wzf9mtlp) 6

[Determinación de KPIs](#_phcq4nvm94k7) 6

[Pasos para el diseño del DW](#_ybhnvss91m20) 6

[Paso 1. Elección del proceso para modelar](#_6kcjv6nkftbp) 7

[Paso 2. Decidir el nivel de granularidad](#_vxn5vnup6shz) 7

[Paso 3. Identificar las dimensiones](#_fsonz64zc53m) 8

[Paso 4. Decidir la información a almacenar](#_wbkqglaa4lwz) 11

[**Procesos ETL**](#_uc5vpu2krmaj) **12**

[Tabla de dimensión “PRODUCTOS”](#_bv8hnqo7dyi5) 14

[Tabla dimensión “STORE”](#_zcc2mkshsz8m) 16

[Tabla dimensión “CUSTOMER”](#_yf4fecyn739s) 18

[Tabla dimensión “STAFF”](#_y4izp67oltoz) 19

[Tabla dimensión “DATE”](#_vika32vccbfe) 21

[Tabla de hechos “SALES”](#_dr85yd8zy6gv) 22

[Tabla de hechos “Staff sale accumulative”](#_gy0yd6vqbw0r) 23

[Procedimiento de carga](#_xw0zve5wyij6) 24

[**Power BI**](#_a1v8t3yx8s76) **27**

[**Conclusiones**](#_43shaopkupp7) **31**

[**Referencias**](#_rbhy0sj9fzzc) **31**

# **Introducción**

En el presente informe se expone un caso de implementación de inteligencia de negocios para una tienda de bicicletas. La cual cuenta con una base de datos relacional (OLTP), con información de ventas, detalles de las ventas. la tienda, los clientes y los productos primeramente. Además se mostrará el proceso de creación de un modelo multidimensional que pueda ser de ayuda para la toma de decisiones.

## **Objetivo**

El objetivo principal es la creación de un modelo multidimensional basado en la inteligencia de negocios que agrupe los datos y se aprovechen para convertirlos en información, y de esa manera usarlos para el crecimiento del negocio.

# **Marco teórico**

## *¿Qué es Business Intelligence (BI)?*

Business Intelligence (BI) o inteligencia de negocios se define como la habilidad corporativa para tomar decisiones. Esto se logra mediante el uso de metodologías, aplicaciones y tecnologías que permiten reunir, depurar, transformar datos, y aplicar en ellos técnicas analíticas de extracción de conocimiento (Parr, 2000).

BI se puede ver como el proceso en el que intervienen personas y sistemas con la meta de obtener, recopilar, analizar y presentar la información que soporte de mejor forma la toma de decisiones de negocio. El proceso se puede dividir en 4 etapas: extracción, consolidación, explotación y visualización (Dávila, 2006).

## *¿Qué es un datawarehouse?*

“Datawarehouse es una colección de datos orientados al tema, integrados no volátiles e históricos cuyo objetivo es servir de apoyo en el proceso de toma de decisiones gerenciales” (Inmon, 2005) .

“La Bodega de Datos es un colección de datos en forma de una base de datos que guarda y ordena información que se extrae directamente de los sistemas operacionales (ventas, producción, finanzas, marketing, etc.) y de datos externos'' (Ross & Kimball, 2013).

# **Entorno del negocio**

En esta sección se presenta la empresa de bicicletas, definiendo las áreas relacionadas con el Datawarehouse, abordando la situación actual, para especificar las necesidades reales y actuales que presenta la empresa respecto a una problemática. Incluso ver la implementación del sistema y señalar una metodología a emplear.

## **Estado actual**

Está ocasión se trata de una empresa ficticia o una emulación que se encuentra muy adecuada a la realidad. Se puede decir que es una cadena de bicicletas que cuenta con 3 sucursales y que vende variadas marcas de bicicletas de alta calidad. Se sabe que estas sucursales cuentan con un equipo de empleados administrados por un manager.

Además cuenta con una base de datos relacional que gestiona las compras de los clientes. No se sabe con certeza las áreas que maneja la empresa internamente, pueden ser por ejemplo: Gestión comercial, Gestión de Ventas, Gestión de Logística, etc.

## **Área a elegir de la empresa**

En este informe se usará como ejemplo el área de soporte de información que es la sección que se encarga de proveer información a las distintas áreas de la empresa. Viendo como se encuentra el estado de la empresa, es posible asumir que no tenga la generación de informes o si los tiene es de forma manual y poco eficiente. Se desconoce en caso de tener informes o reportes si se cargan con cierta frecuencia, y estos pueden llegar a demorar el sistema con una simple consulta sql.

Los datos que la empresa tiene se almacenan en formato de tablas, y no cuentan con un registro histórico diario o mensual. Además cabe mencionar que algunos datos no se pueden obtener con solo extraerlos directamente de la base de datos, por lo que debe de crearse un área de soporte o la ya existente encargarse de ello, con tal de que la información solicitada llegue al usuario final.

Por ello el análisis se hace sencillo por medio de herramientas sobre un ambiente de un DW. De tal manera que se permita un mejor manejo de los datos y entreguen la información requerida y llegué a ser conocimiento.

# **Diseño de un Data Warehouse**

En esta sección se hablará del diseño de un modelo multidimensional y los pasos para la construcción de un DW. Desde el análisis de los requerimientos hasta la generación de los cubos de información.

## **Requerimientos**

* El principal requerimiento es la entrega de información, la cual permita generar la carga automática de los informes del área correspondiente.

Para este caso práctico se pueden usar informes mensuales como un requerimiento.

* Además se asume que el DW estará conectado a la base de datos actual, permitiendo una actualización de los datos en ambos lados.
* Otro requerimiento es que permita a los usuarios finales realizar consultas complejas.

## **Preguntas de gestión**

En este apartado se busca la recopilación de las necesidades de información, y esto se suele realizar por medio de entrevistas, conferencias, cuestionarios, etc.

Se tiene como objetivo lograr identificar las necesidades de información precisa de un alto nivel, para llegar a las metas y cumplir con lo que dice la empresa.

En este caso práctico se tratara del área de soporte de información orientado al proceso de ventas, ya que contiene un gran volumen de información. Es importante conocer las ganancias, montos de ventas concretadas y los costos asociados, incluso la cantidad de bicicletas vendidas a un cliente.

Cabe mencionar, que el modelado multidimensional corresponde a un esquema multidimensional, que representa una actividad que es objeto de análisis (hecho) y las dimensiones que caracterizan la actividad (dimensiones).

La información relevante sobre el hecho (actividad) se representa por un conjunto de indicadores (medidas o atributos de hecho).

La información descriptiva de cada dimensión se representa por un conjunto de atributos (atributos de dimensión).

El modelado multidimensional se puede aplicar utilizando distintos modelos de datos (conceptuales o lógicos).

Dentro del área de ventas podemos deducir que:

* Se desea conocer cómo fluctúan los descuentos en los productos.
* Se desea conocer el monto de ventas de un vendedor.
* Se desea conocer el nivel o tamaño de órdenes que maneja un vendedor y como este cumple.
* La media de descuentos que un vendedor puede acumular.

## **Identificación de KPIs y dimensiones**

En este sector del informe se utilizan las respuestas de preguntas de gestión para identificación de los indicadores que se usarán para el análisis.

**Área de ventas:**

* Ingreso del artículo sin descuento.
* Venta con descuento.
* Monto del descuento

Incluso se desea obtener información del desempeño de los trabajadores y esto se estudiará a partir de los ingresos de sus ventas concretadas.

* Venta total del empleado
* Total de ítems vendidos
* Total de órdenes registradas
* Promedio de descuentos

Lo subrayado corresponde a un indicador, pero se asociará un nombre para diferenciarlo y la continuación del requerimiento son las dimensiones.

Las dimensiones que se encuentran respecto de lo anterior son:

* Sucursal
* Producto
* Empleado
* Cliente

## **Análisis de procesos OLTP**

Para alimentar el DW se debe identificar las fuentes de los datos y para ello se usan como referencia los procesos OLTP (On Line Transactional Processing), es decir, aquella información transaccional que se produce en una empresa en un periodo de tiempo y las fuentes puedes llegar a variar.

Para este caso se ha determinado que existe un modelo relacional que cuenta con 9 tablas de datos.

### **Determinación de KPIs**

Este paso es para determinar el cálculo que se llevará a cabo con tal de obtener el indicador.

Para el cálculo de la primera tabla de hechos “Sales”:

* list\_price \* quantity
* discount \* (list\_price \* quantity)
* list\_price \* quantity - discount \* (list\_price \* quantity)

Para el cálculo de la segunda tabla de hechos “Staff\_sale\_accumulative”:

* sum(oi.list\_price \* quantity - discount \* (oi.list\_price \* quantity))
* sum(quantity)
* count(o.order\_id)
* avg(discount)

## **Pasos para el diseño del DW**

A continuación se presentan los pasos para el diseño de un DW:

* Paso 1: Elegir un “proceso” de la organización para modelar.
* Paso 2: Decidir el nivel de granularidad de representación del proceso.
* Paso 3: Identificar las dimensiones que caracterizan el proceso.
* Paso 4: Decidir la información a almacenar sobre el proceso.

### **Paso 1. Elección del proceso para modelar**

*Proceso*: Corresponde a la actividad de una organización soportada por un sistema OLTP del cual podemos extraer información con el fin de construir un almacén de datos.

Para esta ocasión se basa en una**cadena de ventas de bicicletas**.

La actividad que corresponde a modelar es las **ventas** de las bicicletas del vendedor por medio de las ventas concretadas. Además se desea conocer que tanto influyen los descuentos en los productos.

### **Paso 2. Decidir el nivel de granularidad**

*Granularidad*: Es el nivel de detalle, al que se quiere almacenar información. Tiene que ser a un nivel atómico.

La granularidad viene a ser:

“**Se desea almacenar información sobre las ventas mensuales de cada bicicleta por cada sucursal de la cadena**”.

**Granularidad Producto:**

* **Id de producto**: clave primaria que identifica a un producto único.
* **Nombre del producto**: nombre asociado al producto.
* **id de la marca**: código de la marca.
* **marca**: nombre de la marca.
* **id de la categoría**: código de la categoría.
* **categoría**: nombre de la categoría.
* **año del modelo**: año en que se hizo el producto.
* **el precio lista**: precio del producto.
* **id de stock:** código correspondiente al stock.
* **stock**: cantidad del producto.

**Granularidad Fecha:**

* **Date key:** código de la fecha.
* **Fecha**: fecha de la orden del producto.
* **Descripción de la fecha**: fecha en formato de palabras.
* **día de la semana**: número del día de la semana.
* **número del mes**: número del mes que corresponde.
* **número del año**: número del año que se encuentra asociado.

**Granularidad Sucursal:**

* **Id de la sucursal**: código de la sucursal.
* **nombre**: nombre de la sucursal.
* **teléfono**: teléfono asociado a la sucursal.
* **email**: dirección de correo electrónico de la sucursal.
* **dirección**: dirección de la sucursal.
* **ciudad** : nombre de la ciudad donde se encuentra la sucursal.
* **estado**: estado donde se encuentra la ciudad.
* **código postal**: código postal del estado.

**Granularidad Cliente:**

* **Id del cliente**: código del cliente.
* **nombres**: nombres del cliente.
* **apellidos**: apellidos del cliente.
* **teléfono**: teléfono del cliente.
* **email**: correo electrónico del cliente.
* **dirección**: dirección de residencia del cliente.
* **ciudad**: ciudad del cliente.
* **estado**: estado en que vive el cliente.
* **código postal**: código postal del estado del cliente.

**Granularidad Empleado:**

* **Id del personal**: código asignado a un empleado.
* **nombres**: nombres del empleado.
* **apellidos**: apellidos del empleado.
* **email**: correo electrónico del empleado.
* **teléfono**: número de teléfono del empleado.
* **actividad**: si el empleado sigue activo o se retiró.
* **sucursal** **id**: código de la sucursal a la que pertenece el empleado.
* **manager id**: código del manager al cual está asociado el empleado.

### **Paso 3. Identificar las dimensiones**

*Dimensión*: Es la caracterización de la actividad al nivel de detalle que se ha elegido.

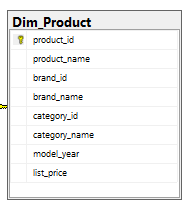
Las dimensiones a seleccionar son:

* Fecha (dimensión temporal: ¿Cuándo se produce la actividad?)
* Producto (¿Cuál es el objeto de la actividad?)
* Sucursal(¿Dónde se produce la actividad?)
* Cliente (¿Quién es el destinatario de la actividad?)
* Empleado (¿Quién es el responsable de la actividad?)

Cada dimensión está asociada a los atributos relevantes para el análisis de la actividad.

**Dimensión Producto:**

* **Id de producto**
* **Nombre del producto**
* **id de la marca**
* **marca**
* **id de la categoría**
* **categoría**
* **año del modelo**
* **el precio lista**
* **id de stock**
* **stock**

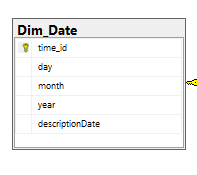
****

**Figura 1. Tabla dimensión “PRODUCT”.**

**Dimensión Fecha:**

Las fechas se generan a partir del campo “order\_date” de la tabla orders.

* **Date key**
* **Fecha**
* **Descripción de la fecha**
* **día de la semana**
* **número del mes**
* **número del año**

****

**Figura 2. Tabla dimensión “DATE”.**

**Dimensión Sucursal:**

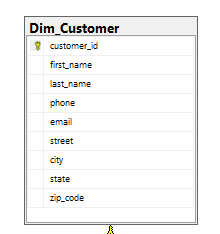
* **Id de la sucursal**
* **nombre**
* **teléfono**
* **email**
* **dirección**
* **ciudad**
* **estado**
* **código postal**

****

**Figura 3. Tabla dimensión “STORE”.**

**Dimensión Cliente:**

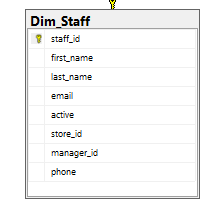
* **Id del cliente**
* **nombres**
* **apellidos**
* **teléfono**
* **email**
* **dirección**
* **ciudad**
* **estado**
* **código postal**

****

**Figura 4. Tabla dimensión “CUSTOMER”.**

**Dimensión Empleado:**

* **Id del personal**
* **nombres**
* **apellidos**
* **email**
* **teléfono**
* **actividad**
* **sucursal** **id**
* **manager id**

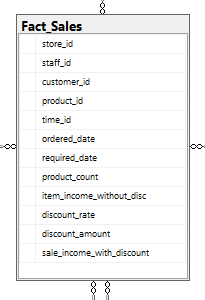
****

**Figura 5. Tabla dimensión “STAFF”.**

### **Paso 4. Decidir la información a almacenar**

*Hechos*: Información que se desea almacenar en cada conjunto de la tabla y la cuál se analizará.

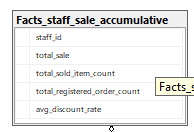
* Ingreso de artículo sin descuento.
* Venta con el descuento aplicado.



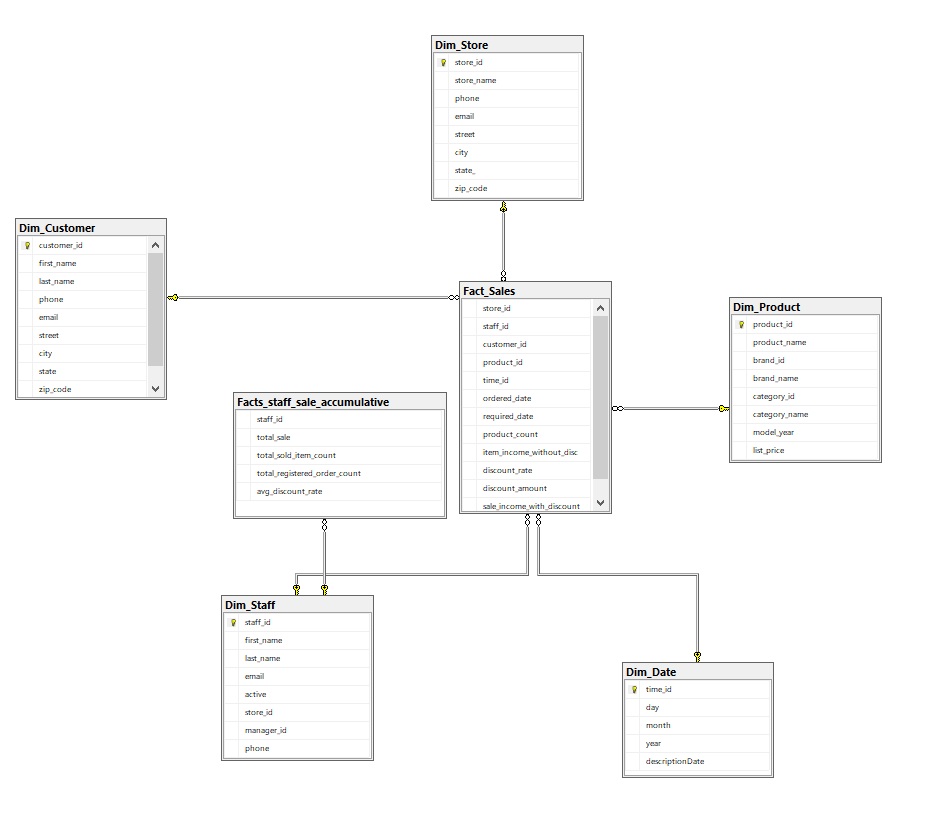
**Figura 6. Tabla de Hechos “SALES”.**

A continuación, se muestra otra tabla de hechos para poder mostrar otros datos de interés:

* Venta total del empleado
* Total de ítems vendidos
* Total de órdenes registradas
* Promedio de descuentos



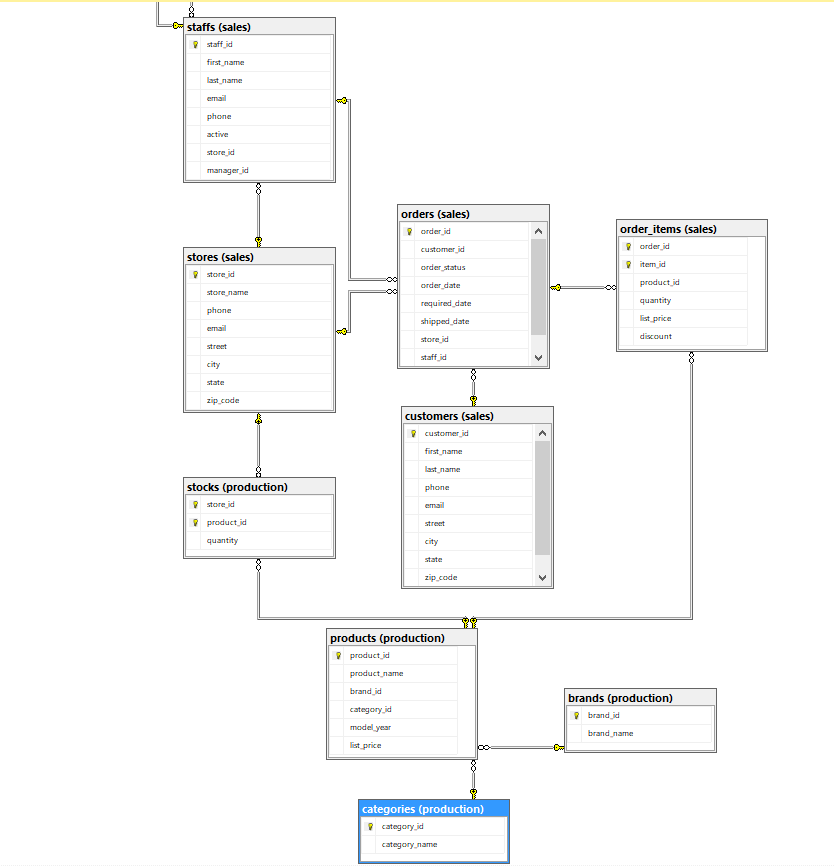
**Figura 7. Tabla de Hechos “Facts\_staff\_sale\_accumulative”.**

**Figura 8. Modelo tipo estrella del Data Warehouse.**

# **Procesos ETL**

En está sección se definen los procesos ETL, es decir, extracción de los datos, para luego transformarlos y limpiarlos. Para ello se utilizarán sentencias SQL.

La información actualmente se encuentra almacenada en una base de datos llamada “BikeStores”. Como se muestra en la siguiente figura.



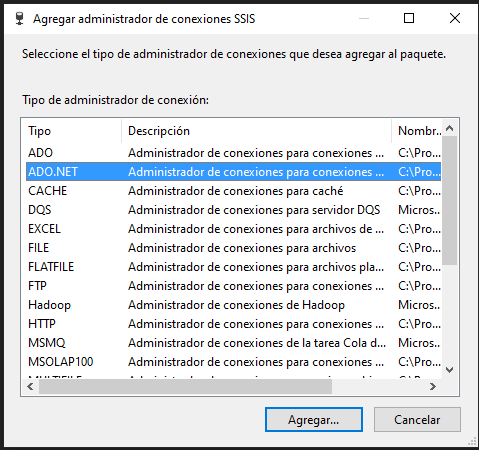
**Figura 9. Base de datos relacional.**

Como sumario cabe decir que, en primer lugar se cargarán los datos de las dimensiones y luego los de la tabla de hechos. Posterior a la carga se establecen las políticas para la actualización de los datos.

Se presentan las sentencias SQL para las diferentes tablas de dimensiones y tablas de hechos.

Para el proceso ETL se hará uso de la herramienta Visual Studio con tal de emplear SSIS.

En la siguiente imagen se crean las conexiones SSIS que se tendrán en consideración. Se debe tener una conexión de la base de datos de origen (BikeStores) y una conexión de destino del datawarehouse (DW\_BikeStore).



**Figura 10. Conexión de la base de datos destino.**

## **Tabla de dimensión “PRODUCTOS”**

La entrada de esta tabla corresponde a la tabla “production.products” unida a la tabla “production.categories” y a la tabla “production.brands” provenientes de la base de datos.

A continuación se encuentra la consulta SQL para la creación de la dimensión Dim\_Product:

USE [DW\_BikeStore]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Dim\_Product] Script Date: 02/05/2022 22:59:25 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Dim\_Product](

[product\_id] [int] NOT NULL,

[product\_name] [varchar](255) NULL,

[brand\_id] [int] NULL,

[brand\_name] [varchar](255) NULL,

[category\_id] [int] NULL,

[category\_name] [varchar](255) NULL,

[model\_year] [smallint] NULL,

[list\_price] [decimal](10, 2) NULL,

CONSTRAINT [PK\_Dim\_Products] PRIMARY KEY CLUSTERED

(

[product\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

Y mediante la siguiente sentencia SQL se usará para llenar la dimensión producto.

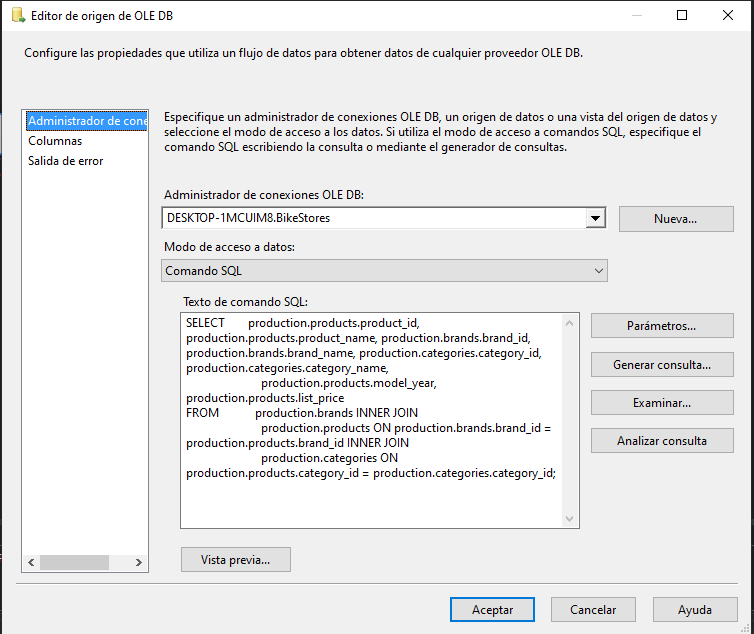
SELECT production.products.product\_id, production.products.product\_name, production.brands.brand\_id, production.brands.brand\_name, production.categories.category\_id, production.categories.category\_name,

production.products.model\_year, production.products.list\_price

FROM production.brands INNER JOIN

production.products ON production.brands.brand\_id = production.products.brand\_id INNER JOIN

production.categories ON production.products.category\_id = production.categories.category\_id;



**Figura 11. Carga de datos mediante SQL a dim\_product.**

## **Tabla dimensión “STORE”**

La entrada de esta tabla corresponde a la tabla “sales.stores” proveniente de la base de datos.

A continuación se encuentra la consulta SQL para la creación de la dimensión Dim\_Store:

USE [DW\_BikeStore]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Dim\_Store] Script Date: 02/05/2022 23:02:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Dim\_Store](

[store\_id] [int] NOT NULL,

[store\_name] [varbinary](255) NULL,

[phone] [varchar](25) NULL,

[email] [varchar](255) NULL,

[street] [varchar](255) NULL,

[city] [varchar](255) NULL,

[state] [varchar](10) NULL,

[zip\_code] [varchar](5) NULL,

CONSTRAINT [PK\_Dim\_Store] PRIMARY KEY CLUSTERED

(

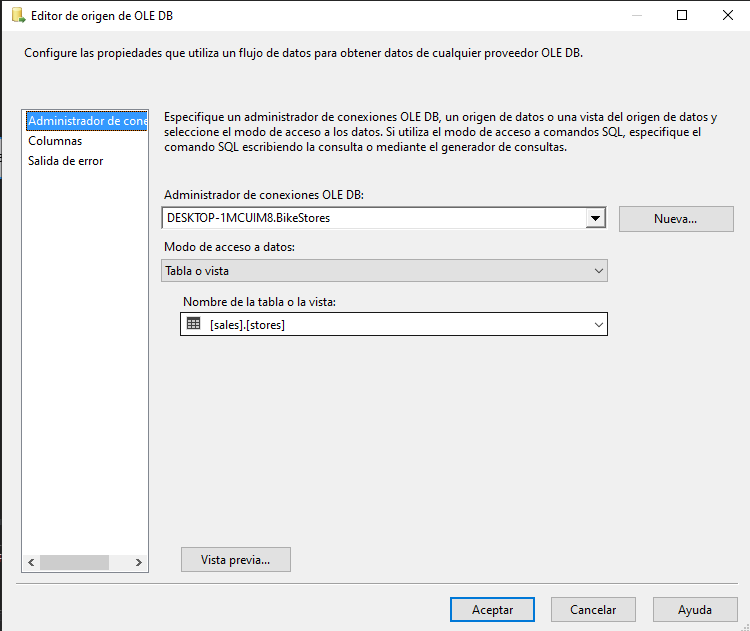
[store\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

Y para está tabla no es necesaria la sentencia, basta con solo pasar la tabla “sales.stores”.



**Figura 12. Carga de datos mediante tabla a dim\_store.**

## **Tabla dimensión “CUSTOMER”**

La entrada de esta tabla corresponde a la tabla “sales.customers” proveniente de la base de datos.

A continuación se encuentra la consulta SQL para la creación de la dimensión Dim\_Customer:

USE [DW\_BikeStore]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Dim\_Customer] Script Date: 02/05/2022 23:08:16 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Dim\_Customer](

[customer\_id] [int] NOT NULL,

[first\_name] [varchar](255) NULL,

[last\_name] [varchar](255) NULL,

[phone] [varchar](25) NULL,

[email] [varchar](255) NULL,

[street] [varchar](255) NULL,

[city] [varchar](50) NULL,

[state] [varchar](25) NULL,

[zip\_code] [varchar](5) NULL,

CONSTRAINT [PK\_Dim\_Customer] PRIMARY KEY CLUSTERED

(

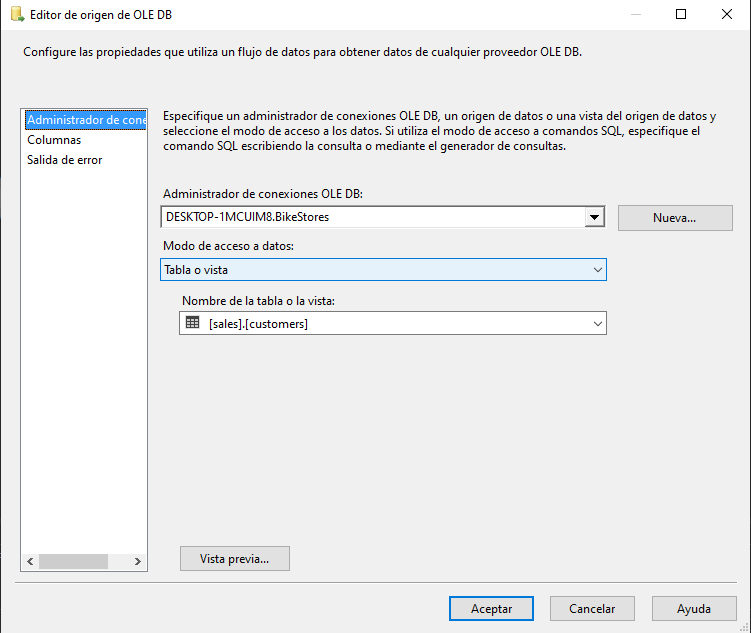
[customer\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

Y para está tabla tampoco es necesaria la sentencia, basta con solo pasar la tabla “sales.customers”.



**Figura 13. Carga de datos mediante tabla a dim\_customer.**

## **Tabla dimensión “STAFF”**

La entrada de esta tabla corresponde a la tabla “sales.staffs” proveniente de la base de datos.

A continuación se encuentra la consulta SQL para la creación de la dimensión Dim\_Staff:

USE [DW\_BikeStore]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Dim\_Staff] Script Date: 02/05/2022 23:13:19 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Dim\_Staff](

[staff\_id] [int] NOT NULL,

[first\_name] [varchar](50) NULL,

[last\_name] [varchar](50) NULL,

[email] [varchar](255) NULL,

[active] [tinyint] NULL,

[store\_id] [int] NULL,

[manager\_id] [int] NULL,

[phone] [varchar](50) NULL,

CONSTRAINT [PK\_Dim\_Staff] PRIMARY KEY CLUSTERED

(

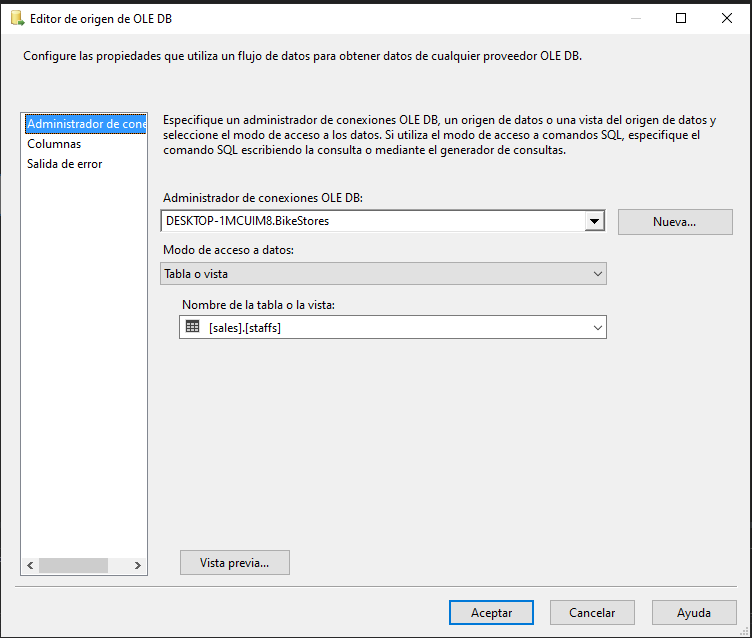
[staff\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

Y para está tabla tampoco es necesaria la sentencia, basta con solo pasar la tabla “sales.staffs”.



**Figura 14. Carga de datos mediante tabla a dim\_staff.**

## **Tabla dimensión “DATE”**

La entrada de esta tabla corresponde a la tabla “sales.orders” proveniente de la base de datos.

A continuación se encuentra la consulta SQL para la creación de la dimensión Dim\_Dater:

USE [DW\_BikeStore]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Dim\_Date] Script Date: 02/05/2022 23:33:11 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Dim\_Date](

[time\_id] [int] NOT NULL,

[day] [int] NULL,

[month] [int] NULL,

[year] [int] NULL,

[descriptionDate] [varchar](50) NULL,

CONSTRAINT [PK\_Dim\_Date] PRIMARY KEY CLUSTERED

(

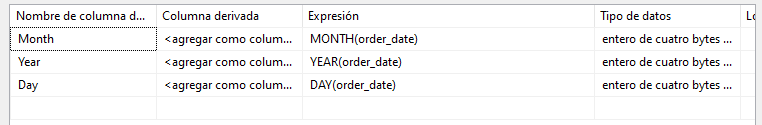
[time\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

Y para está tabla se poblara mediante una columna derivada.



**Figura 15. Carga de datos mediante columna derivada.**

## **Tabla de hechos “SALES”**

La entrada de esta tabla corresponde a la tabla “sales.orders” unida a “sales.orders\_items” proveniente de la base de datos.

A continuación se encuentra la consulta SQL para la creación de la tabla de hechos Fact\_Sales:

create table Fact\_Sales

(

time\_id int,

sales\_id int,

store\_id int,

staff\_id int,

customer\_id int,

product\_id int,

ordered\_date date,

required\_date date,

product\_count int,

item\_income\_without\_disc decimal(10, 2),

discount\_rate decimal(4, 2),

discount\_amount decimal(10, 2),

sale\_with\_discount decimal(10, 2)

)

Para poblar la tabla de Hechos se crea una tabla logs que permita tener registro de las acciones realizadas y posterior la creación de un procedimiento para la población de dicha tabla.

create table logs

(

date datetime,

table\_name varchar(50),

status tinyint,

text varchar(500)

)

Cabe mencionar que se debió usar una función que devolviera un formato correcto de fecha correspondiente a la mínima.

create function dbo.MIN\_DATE() returns date as

begin

return cast('2016-1-1' as date)

end

go

---------------------------------------------------------------------------------------------------------------------------

El siguiente procedimiento se encarga de cargar en la tabla de hechos los datos hasta la fecha indicada.

create procedure insert\_fact\_sales @end\_date date as

begin

declare @cur\_date as date

insert into logs

values (current\_timestamp, 'Fact\_Sales', 2, 'insert\_fact\_sales ')

select @cur\_date = IIF(max(ordered\_date) is null, dbo.MIN\_DATE(), dateadd(day, 1, max(ordered\_date)))

from Fact\_orders

while @cur\_date < @end\_date

begin

insert into Fact\_orders

select

dd.time\_id,

store\_id,

staff\_id,

customer\_id,

dp.product\_id,

order\_date,

required\_date,

quantity,

oi.list\_price \* quantity,

discount,

discount \* (oi.list\_price \* quantity),

oi.list\_price \* quantity - discount \* (oi.list\_price \* quantity)

from BikeStores.sales.orders o

full outer join BikeStores.sales.order\_items oi

on o.order\_id = oi.order\_id and o.order\_id is not null and oi.order\_id is not null

left join Dim\_Product dp on oi.product\_id = dp.product\_id left join Dim\_Date dd on oi.order\_id = dd.time\_id

where order\_date = @cur\_date

set @cur\_date = dateadd(day, 1, @cur\_date)

end

insert into logs

values (current\_timestamp, 'Facts\_Sales', 3, 'insert\_facts\_sales ')

end

go

Para ejecutar el procedimiento basta con ejecutarlo mediante la siguiente sentencia, la fecha puede ser cualquiera, pero se empleo la actual:

exec insert\_facts\_sales '2022-4-05'

go

## **Tabla de hechos “Staff sale accumulative”**

Se crea una tabla de hechos de ventas del staff, con el motivo de mostrar información que no puede mostrarse con la otra tabla, ya que los datos son de gran densidad, no conviene hacer los cálculos dentro del DW.

Esta tabla estará conformada por las tablas de “sales.orders” y “sales.order\_items” con funciones de cálculo previamente realizadas.

create table Facts\_staff\_sale\_accumulative

(

staff\_id int,

total\_sale decimal(20, 2),

total\_sold\_item\_count int,

total\_registered\_order\_count int,

avg\_discount\_rate decimal(4, 2)

)

Para la selección de los datos se utilizará la siguiente secuencia SQL:

select staff\_id,

sum(oi.list\_price \* quantity - discount \* (oi.list\_price \* quantity)) as total\_sale ,

sum(quantity) as total\_sold\_item\_count,

count(o.order\_id) as total\_registered\_order\_count,

avg(discount) as avg\_discount\_rate

from BikeStores.sales.orders o

full outer join BikeStores.sales.order\_items oi

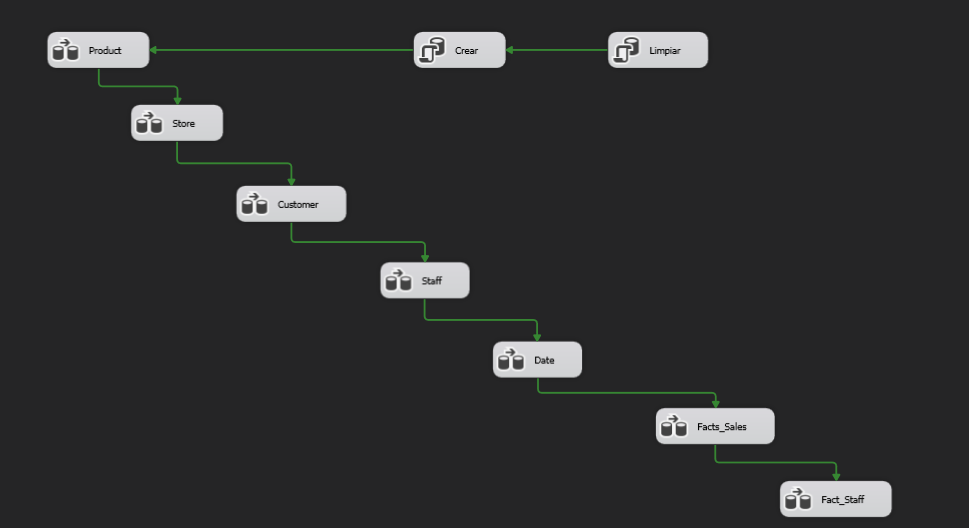
on o.order\_id = oi.order\_id and o.order\_id is not null and oi.order\_id is not null

group by staff\_id;

## **Procedimiento de carga**

En este apartado se habla sobre un procedimiento de carga realizado en Visual Studio, utilizando SSIS (SQL Server Integration Services), la cual es una plataforma que permite generar soluciones de integración de datos de alto rendimiento, y donde se está desarrollando los procesos ETL.

En primer lugar, se crea un paquete para los procesos que se realizan. El siguiente paso es crear un **flujo de control**, que lleve todas las cargas y limpiezas del proceso.



**Figura 16. Proceso ETL para DW.**

A continuación se muestra el procedimiento de carga según la fecha dada. Podemos usar el año 2016 o 2017 para referencia.

USE [DW\_BikeStore]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[runner] Script Date: 10/05/2022 21:54:08 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create procedure [dbo].[runner] @end\_date date as

begin

exec insert\_facts\_sales '2020-1-20'

exec insert\_fact\_staff\_sale\_accumulative @end\_date

end

GO

USE [DW\_BikeStore]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[insert\_fact\_staff\_sale\_accumulative] Script Date: 10/05/2022 21:52:13 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create procedure [dbo].[insert\_fact\_staff\_sale\_accumulative] @end\_date date as

begin

declare @cur\_date as date

declare @last\_run\_status as tinyint

select top 1 @cur\_date = date, @last\_run\_status = status

from logs

where table\_name = 'Facts\_staff\_sale\_accumulative'

order by date desc, status desc

if @last\_run\_status != 3 and @last\_run\_status is not null

begin

print 'último intento no fue exitoso en Facts\_staff\_accumulative'

return

end

if @last\_run\_status is null

delete from TEMP\_staff\_accumulative where 1 > 0

while @cur\_date < @end\_date

begin

insert into TEMP\_staff\_accumulative

select staff\_id,

sum(oi.list\_price \* quantity - discount \* (oi.list\_price \* quantity)),

sum(quantity),

count(o.order\_id),

avg(discount)

from BikeStores.sales.orders o

full outer join BikeStores.sales.order\_items oi

on o.order\_id = oi.order\_id and o.order\_id is not null and oi.order\_id is not null

where o.order\_date = @cur\_date

group by staff\_id

set @cur\_date = dateadd(day, 1, @cur\_date)

end

insert into TEMP\_staff\_accumulative

select staff\_id, total\_sale, total\_sold\_item\_count, total\_registered\_order\_count, avg\_discount\_rate

from Facts\_staff\_sale\_accumulative

insert into logs

values (@end\_date, 'fact\_staff\_sale\_accumulative', 2, 'start deleting and replace table ')

delete from Facts\_staff\_sale\_accumulative where 1 > 0

insert into Facts\_staff\_sale\_accumulative

select staff\_id,

sum(total\_sale),

sum(total\_sold\_item\_count),

sum(total\_registered\_order\_count),

avg(avg\_discount\_rate \* total\_registered\_order\_count) / sum(total\_registered\_order\_count)

from TEMP\_staff\_accumulative

group by staff\_id

insert into logs

values (@end\_date, 'fact\_staff\_sale\_accumulative', 3, 'finish deleting and replace table ')

end

GO

# **Power BI**

En esta sección se hablará de la herramienta power bi para la consulta y análisis de la información almacenada en el DW.

Para cada KPI se tendrán 2 vistas que muestran datos importantes en relación al negocio.

* Ingreso del artículo sin descuento: Se muestran los datos de ingresos por cada producto sin el descuento y al lado se tiene un gráfico circular que implica las ventas por año de todos los artículos. Además cabe mencionar que en la zona inferior se tiene un filtro que permite ver las ventas por año.



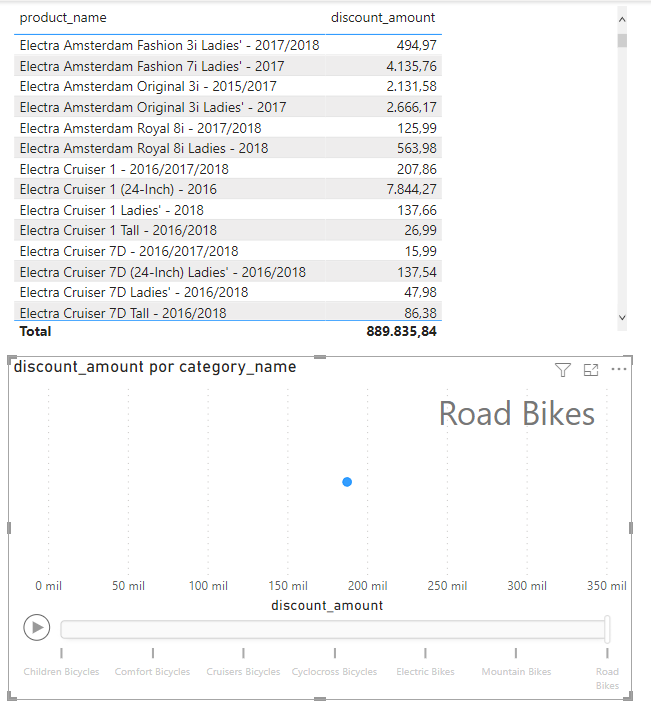
**Figura 17. Vistas KPI “Ingreso sin descuento”**

* Venta con descuento: Similar a lo anterior, pero solo que se consideran los productos con el descuento agregado.



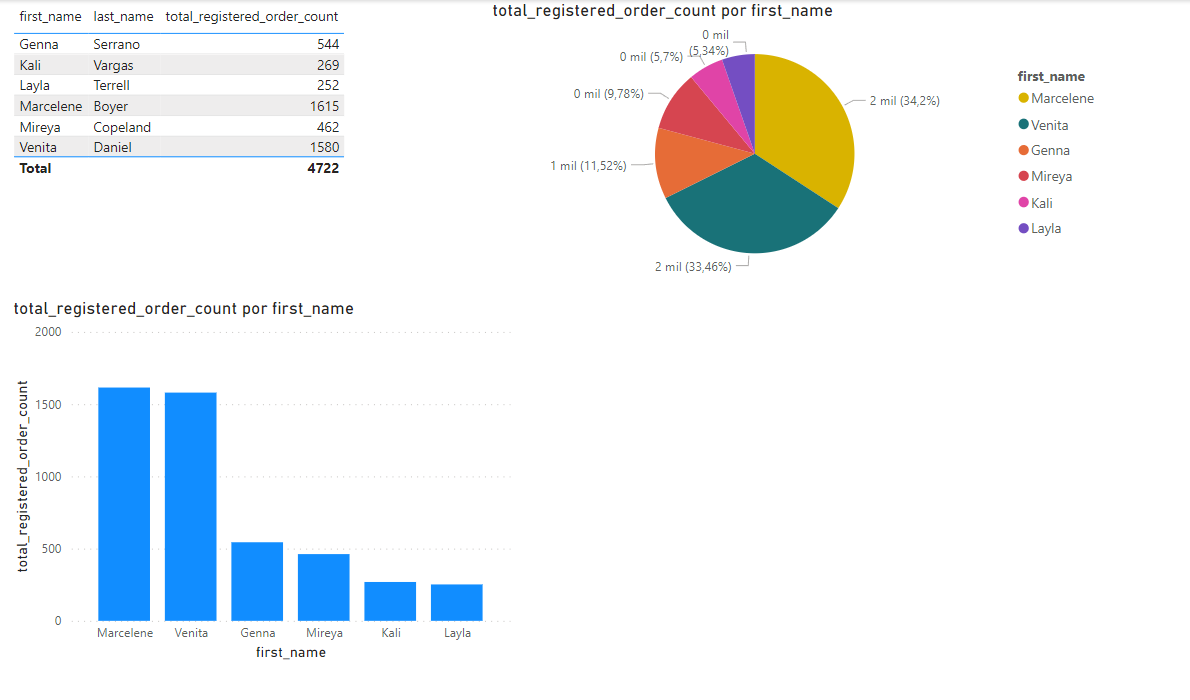
**Figura 17. Vistas KPI “Ingreso con descuento”**

* Monto del descuento



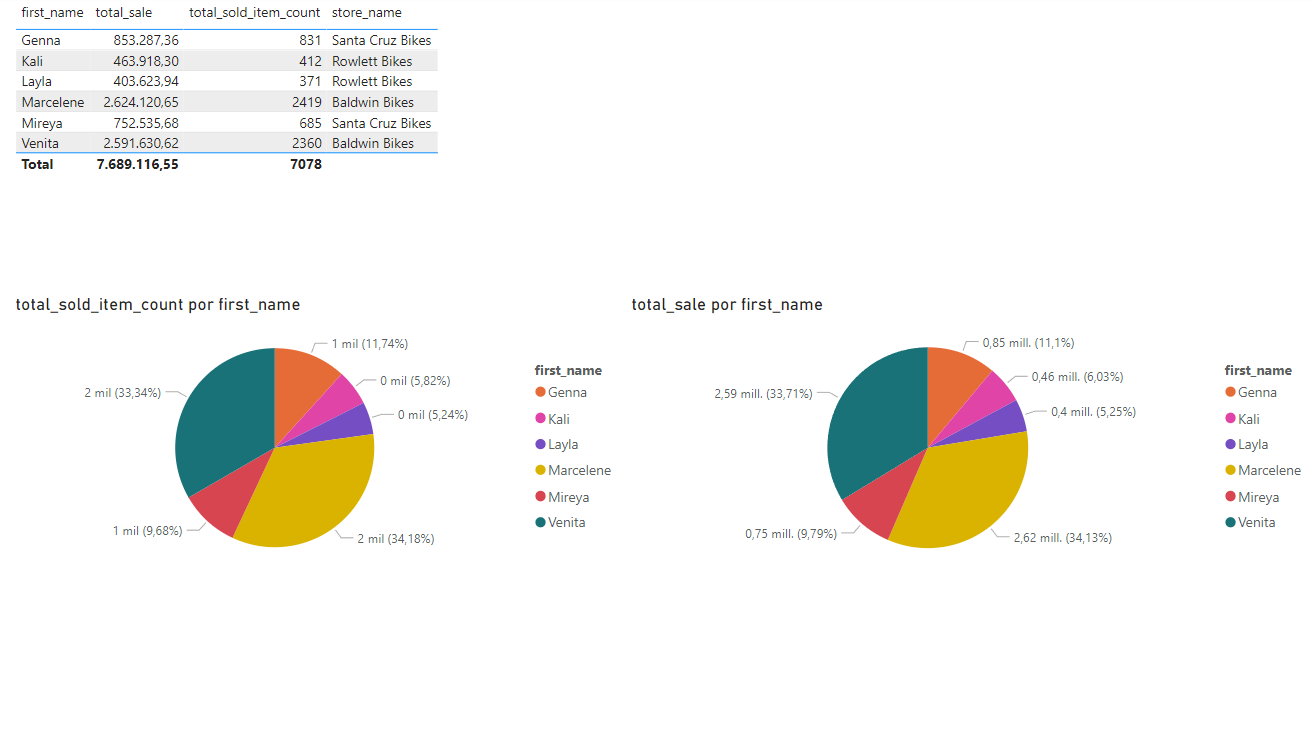
**Figura 18. Vistas KPI “Monto del descuento”**

* Total de órdenes registradas por empleado



**Figura 19. Vistas KPI “Ventas totales del empleado”**

* Total de ítems vendidos y Total de venta por vendedor.



**Figura 20. Vistas KPI “Totales por empleado”.**

No se tomarán en cuenta el resto de KPIs, ya que con esos presentados son suficientes para explicar lo que busca el cliente a primera vista.

# Conclusiones

Posterior a la realización de esta investigación y profundo análisis acerca de los Data Warehouse, se puede notar que es un área bastante amplia dentro de la inteligencia de negocios. En este informe se trató de abarcar paso a paso la creación de un modelo multidimensional, la implementación de un DW y el análisis por medio de power BI, el cual se puede observar visualmente. El conocer el negocio, profundizar los requerimientos, comprender lo que desea el cliente, es complejo de analizar y que tenga resultados inmediatos, por ende es recomendable ir iterando sobre el modelo con tal de lograr lo esperado por el cliente.

Otro aspecto a mencionar es acerca de los procesos ETL, ya que estos son necesarios para la implementación del DW, en un área de soporte TI, radica la dificultad de definir las extracciones de datos de las tablas que contienen una gran cantidad de información y luego el tener que procesarlas al modelo lleva tiempo.

Por ende, este trabajo permitió un aprendizaje y desarrollo de un modelo de negocios que es común dentro del mercado actual, quedando abierto para futuras implementaciones y mejoras que puedan expandir el modelo.

# Referencias

Dávila, F. (2006). *LA INTELIGENCIA DEL NEGOCIO*. https://docplayer.es/627131-La-inteligencia-del-negocio-business-intelligence-fernando-davila-politecnico-grancolombiano-fdavila-poligran-edu-co.html

Inmon, W. (2005). *Building the Data Warehouse.* (Fourth Edition ed., Vol. 4). Jhon Wiley & Sons. https://ia800202.us.archive.org/9/items/2005BuildingTheDataWarehouse4thEditionWilliamH.Inmon/2005%20-%20Building%20The%20Data%20Warehouse%20%284th%20Edition%29%20%28William%20H.%20Inmon%29.pdf

Parr, O. (2000). *Data Mining Cookbook*. John Wiley & Sons. https://books.google.com.co/books?id=L3w0loZrcU0C&printsec=front

Ross, M., & Kimball, R. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley. https://aatinegar.com/wp-content/uploads/2016/05/Kimball\_The-Data-Warehouse-Toolkit-3rd-Edition.pdf