

# **“BIKE Z”**

## Examen Transversal

**Asignatura:** INTELIGENCIA DE NEGOCIOS  
**Sección:** 002D  
**Profesor/a:** ALICIA ELENA RAMOS DE PERAZA

**Integrantes:**

- Cristóbal Acosta Flores
- Julio Fierro Astete
- Cesar Retamal Jara
- Javier Sabando Ruiz
- Andres Cortez

# Índice

Índice	2
Introducción	3
1. Factores Para Desarrollar	4
1.1. KPIS a identificar.	4
1.2. Elección de dimensiones.	5
1.3 Modelo dimensional y lógico	6
2 Tabla Dimensional	7
2.1 Drop y Clean	11
Drop	11
Clean	11
3.- Proceso ETL.	12
3.1 ETL Carga Incremental Producto	15
3.2 ETL Incremental FactVentas	15
3.3 ETL Detalle	16
3.4 ETL Empleado	16
3.5 ETL Flow	17
3.6 ETL País	17
3.7 ETL Producto	17
3.8 ETL Territorio	18
4.-Cubo OLAP	19
4.1 Productos más vendidos	19
4.2 Territorios con más ventas	21
4.3 Fechas con mayores ventas	21
5.- Conclusión	23

## Introducción

Ya para finalizar recordaremos los procesos que como primer proceso de la empresa “BIKE Z “ realizado donde fue necesario identificar KPIS y generar un modelo dimensional para poder identificar de mejor manera lo que haremos es generar distintos ETL respecto a la empresa “BIKE Z” ,donde ya con el modelo dimensional generado es necesario extraer , transformar y cargar los distintos datos que se relacionan con la base de datos y con las herramientas ya generadas podemos crear los ETL mediante Visual Studios y SQL Server, una vez teniendo todos los procesos listos y dispuestos procederemos a generar el cubo olap ya que de esta forma podemos darle una respuesta más “visual” a nuestros KPI’s y darles mejor descripción y detalles.

# 1. Factores Para Desarrollar

El equipo de IT, analizando el modelo de negocio, se ha podido reconocer los elementos claves que se necesitan, para que la empresa incorpore esta “Inteligencia de Negocios” y realizar más su crecimiento, con ello, el equipo IT a determinado los siguientes puntos:

## 1.1. KPIS a identificar.

- **Productos más vendidos**

Esto no ayudará para determinar si el negocio puede llegar a especializarse en un rubro específico dentro del mundo de la venta de bicicletas y refacciones.

- **Territorios con más ventas**

Con esta información podrán saber en qué país y territorio conviene más realizar una expansión o invertir en más y mejores sucursales.

- **Fechas con mayores ventas**

En muchas industrias existen épocas de año especialmente buenas en ventas, con este KPI podremos identificarlas y adelantarnos a los pedidos con un stock más preparado.

- **Empleados con mayores ventas por territorio**

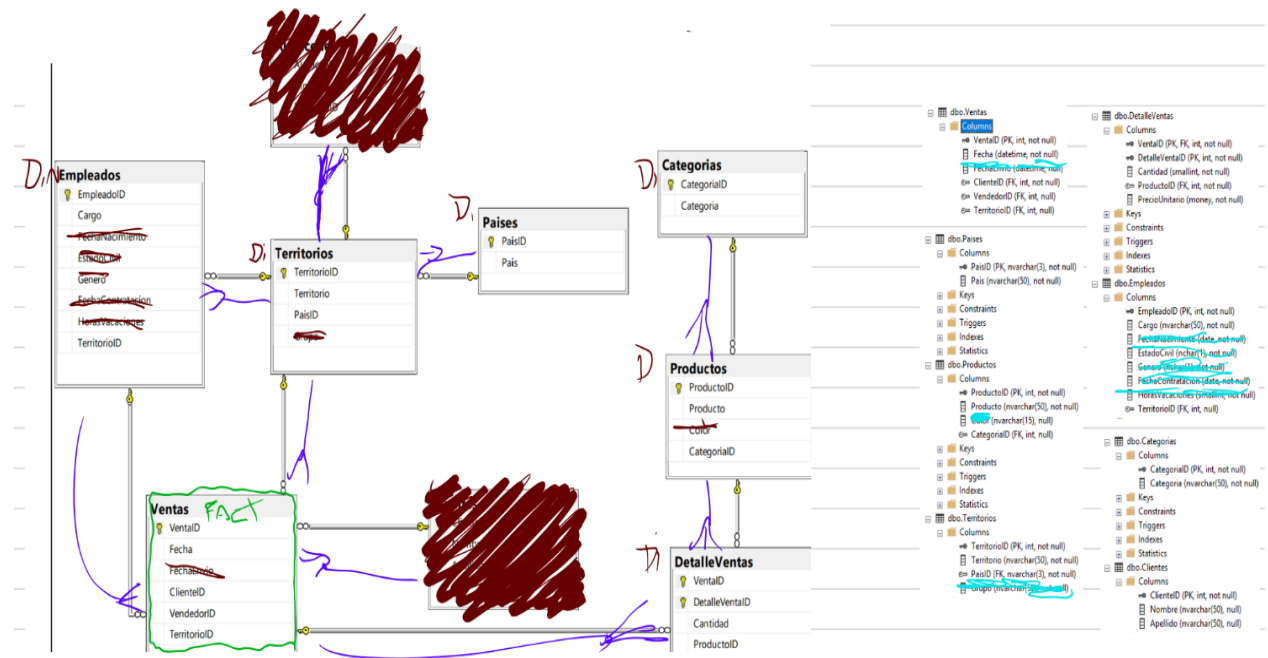
Este es un indicador que no solo nos permitirá reconocer y premiar a los trabajadores con mejores resultados, sino que también podremos usar esta información para saber que hacen la gerencias para dar estos resultados y así generar una cultura de mejora continua.

- **Clientes frecuentes por territorio**

Este punto, al igual que en el anterior, podremos medir algo más humano, como lo es la retención de cliente, si los clientes son fieles, significa que la planta está haciendo un buen trabajo e identificar que se está haciendo, replicándose en otras sucursales puede ser muy beneficioso para el negocio.

## 1.2. Elección de dimensiones.

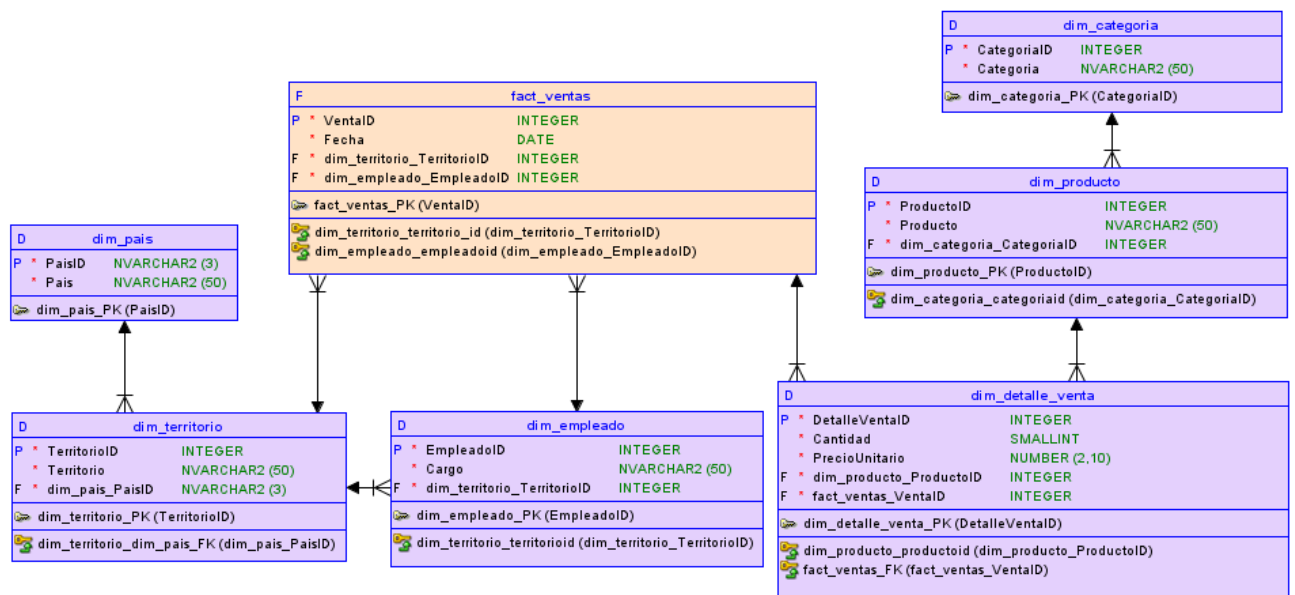
Luego de definir cuáles eran los indicadores que seguiríamos, hicimos la elección de información de utilidad para nuestro caso, identificando como estaba hecho el modelo de base de datos y el tipo de dato que manejaba.

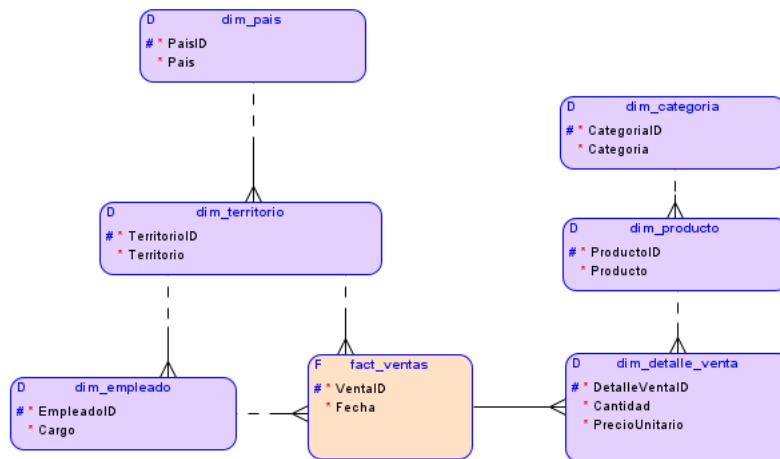


## 1.3 Modelo dimensional y lógico

Después de las definiciones concluidas del KPI, encontradas en el modelo de negocio de la empresa “**BIKE Z**”, el equipo IT a conformado el siguiente modelo dimensional, con los puntos claves encontrados, definiéndolo en un modelo tipo “Copo de Nieve”:

\*Cabe destacar que cambiamos los nvarchar por varchar y el number por money para gestionar de mejor manera la base de datos.\*





## 2 Tabla Dimensional

```
CREATE TABLE dim_categoria (
    categoriaid INTEGER NOT NULL,
    categoria NVARCHAR(50) NOT NULL
);
```

```
ALTER TABLE dim_categoria ADD CONSTRAINT dim_categoria_pk PRIMARY
KEY ( categoriaid );
```

```
CREATE TABLE dim_detalle_venta (
    detalleventaid INTEGER NOT NULL,
    cantidad SMALLINT NOT NULL,
    preciounitario MONEY NOT NULL,
    dim_producto_productoid INTEGER,
    fact_ventas_ventaid INTEGER NOT NULL
);
```

```
ALTER TABLE dim_detalle_venta ADD CONSTRAINT dim_detalle_venta_pk
PRIMARY KEY ( detalleventaid );
```

```
CREATE TABLE dim_empleado (  
    empleadoid          INTEGER NOT NULL,  
    cargo                NVARCHAR(50) NOT NULL,  
    dim_territorio_territorioid INTEGER  
);
```

```
ALTER TABLE dim_empleado ADD CONSTRAINT dim_empleado_pk PRIMARY  
KEY ( empleadoid );
```

```
CREATE TABLE dim_pais (  
    paisid NVARCHAR(3) NOT NULL,  
    pais   NVARCHAR(50) NOT NULL  
);
```

```
ALTER TABLE dim_pais ADD CONSTRAINT dim_pais_pk PRIMARY KEY ( paisid  
);
```

```
CREATE TABLE dim_producto (  
    productoid          INTEGER NOT NULL,  
    producto            NVARCHAR(50) NOT NULL,  
    dim_categoria_categoriaid INTEGER  
);
```

```
ALTER TABLE dim_producto ADD CONSTRAINT dim_producto_pk PRIMARY  
KEY ( productoid );
```

```
CREATE TABLE dim_territorio (  
    territoriod   INTEGER NOT NULL,  
    territorio    NVARCHAR(50) NOT NULL,  
    dim_pais_paisid NVARCHAR(3)
```



);

```
ALTER TABLE dim_territorio ADD CONSTRAINT dim_territorio_pk PRIMARY KEY  
( territoriod );
```

```
CREATE TABLE fact_ventas (  
    ventaid          INTEGER NOT NULL,  
    fecha            DATE NOT NULL,  
    dim_territorio_territorioid INTEGER,  
    dim_empleado_empleadoid  INTEGER  
);
```

```
ALTER TABLE fact_ventas ADD CONSTRAINT fact_ventas_pk PRIMARY KEY (  
ventaid );
```

```
ALTER TABLE dim_producto  
    ADD CONSTRAINT dim_categoria_categoriaid FOREIGN KEY (  
dim_categoria_categoriaid )  
    REFERENCES dim_categoria ( categoriaid );
```

```
ALTER TABLE fact_ventas  
    ADD CONSTRAINT dim_empleado_empleadoid FOREIGN KEY (  
dim_empleado_empleadoid )  
    REFERENCES dim_empleado ( empleadoid );
```

```
ALTER TABLE dim_detalle_venta  
    ADD CONSTRAINT dim_producto_productoid FOREIGN KEY (  
dim_producto_productoid )  
    REFERENCES dim_producto ( productoid );
```

```
ALTER TABLE dim_territorio
```

```
ADD CONSTRAINT dim_territorio_dim_pais_fk FOREIGN KEY (
dim_pais_paisid )
REFERENCES dim_pais ( paisid );
```

```
ALTER TABLE fact_ventas
```

```
ADD CONSTRAINT dim_territorio_territorio_id FOREIGN KEY (
dim_territorio_territorioid )
REFERENCES dim_territorio ( territoriod );
```

```
ALTER TABLE dim_empleado
```

```
ADD CONSTRAINT dim_territorio_territorioid FOREIGN KEY (
dim_territorio_territorioid )
REFERENCES dim_territorio ( territoriod );
```

```
ALTER TABLE dim_detalle_venta
```

```
ADD CONSTRAINT fact_ventas_fk FOREIGN KEY ( fact_ventas_ventaid )
REFERENCES fact_ventas ( ventaidd );;fk FOREIGN KEY (
fact_ventas_ventaid )
REFERENCES fact_ventas ( ventaidd );
```

## 2.1 Drop y Clean

### Drop

```
Drop Table dim_pais;  
Drop Table dim_territorio;  
Drop Table dim_categoria;  
Drop Table dim_producto;  
Drop Table dim_detalle_venta;  
Drop Table dim_empleado;  
Drop Table fact_ventas;
```

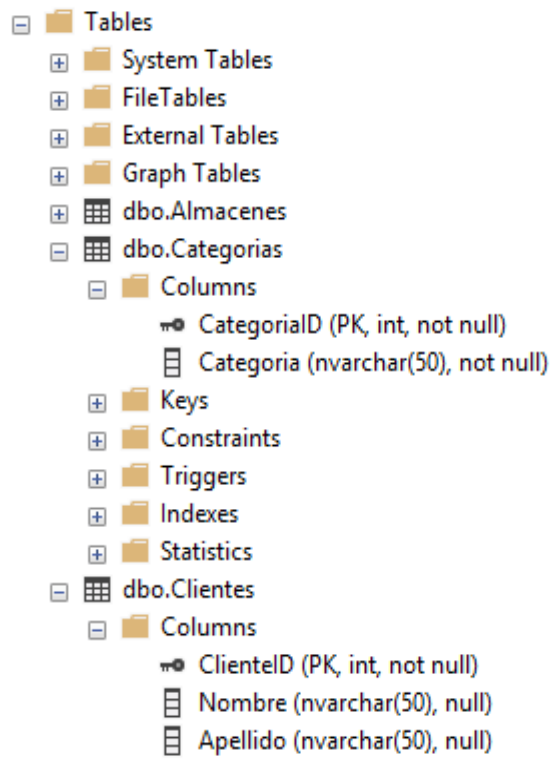
### Clean























```
DELETE FROM dim_pais;  
DELETE FROM dim_territorio;  
DELETE FROM dim_categoria;  
DELETE FROM dim_producto;  
DELETE FROM dim_detalle_venta;  
DELETE FROM dim_empleado;  
DELETE FROM fact_ventas;
```



































### 3.- Proceso ETL.

Al tener el “Modelo Dimensional” listo con todos sus componentes en sus respectivas casillas, solo cabía conectar el modelo nuevo de “**BIKE Z**” al SQL SERVER, a través de la máquina virtual de Amazon, extrayendo, transformándolo y cargar cuyos datos usando la aplicación “*Visual Studio*”, con ello creando el componente SSIS-SQL SERVER.

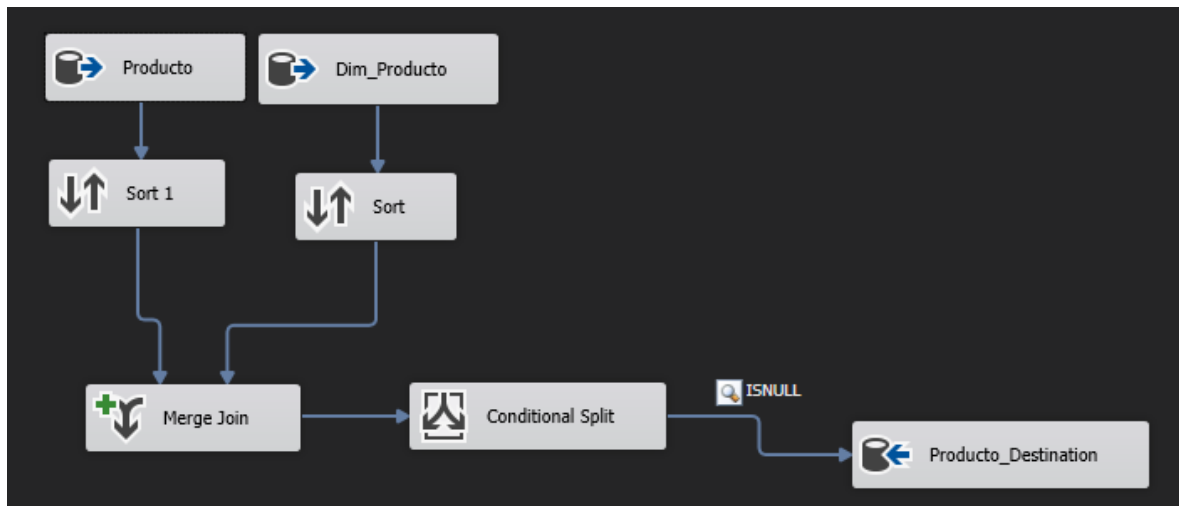
Al finalizar, los datos de mostrarían de la siguiente manera:



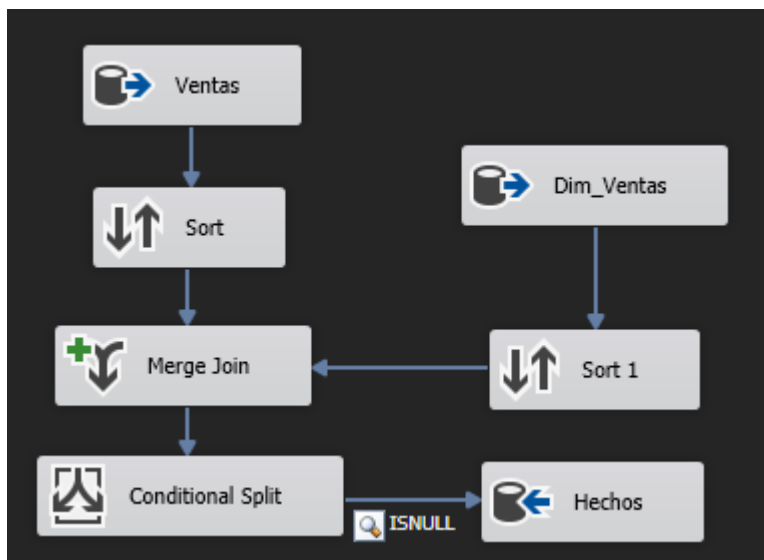
- [-]  **dbo.DetalleVentas**
  - [-]  **Columns**
    -  VentalID (PK, FK, int, not null)
    -  DetalleVentasID (PK, int, not null)
    -  Cantidad (smallint, not null)
    -  ProductID (FK, int, not null)
    -  PrecioUnitario (money, not null)
  - [+]  **Keys**
  - [+]  **Constraints**
  - [+]  **Triggers**
  - [+]  **Indexes**
  - [+]  **Statistics**
- [-]  **dbo.Empleados**
  - [-]  **Columns**
    -  EmpleadoID (PK, int, not null)
    -  Cargo (nvarchar(50), not null)
    -  FechaNacimiento (date, not null)
    -  EstadoCivil (nchar(1), not null)
    -  Genero (nchar(1), not null)
    -  FechaContratacion (date, not null)
    -  HorasVacaciones (smallint, not null)
    -  TerritorioID (FK, int, null)

- [-]  **dbo.Paises**
  - [-]  **Columns**
    -  PaisID (PK, nvarchar(3), not null)
    -  Pais (nvarchar(50), not null)
  - [+]  Keys
  - [+]  Constraints
  - [+]  Triggers
  - [+]  Indexes
  - [+]  Statistics
- [-]  **dbo.Productos**
  - [-]  **Columns**
    -  ProductID (PK, int, not null)
    -  Producto (nvarchar(50), not null)
    -  Color (nvarchar(15), null)
    -  Categoriad (FK, int, null)
  - [+]  Keys
  - [+]  Constraints
  - [+]  Triggers
  - [+]  Indexes
  - [+]  Statistics
- [-]  **dbo.Territorios**
  - [-]  **Columns**
    -  TerritoriID (PK, int, not null)
    -  Territorio (nvarchar(50), not null)
    -  PaisID (FK, nvarchar(3), not null)
    -  Grupo (nvarchar(50), not null)
- [-]  **dbo.Ventas**
  - [-]  **Columns**
    -  VentalID (PK, int, not null)
    -  Fecha (datetime, not null)
    -  FechaEnvio (datetime, null)
    -  ClientelD (FK, int, not null)
    -  VendedorID (FK, int, null)
    -  TerritoriID (FK, int, null)

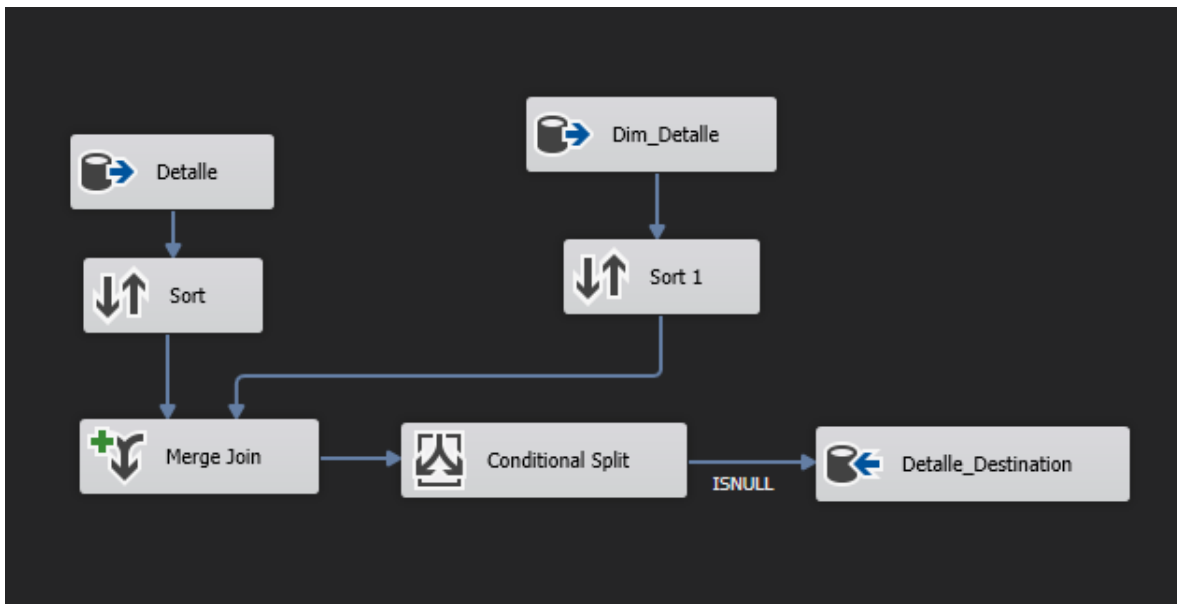
### 3.1 ETL Cargo Incremental Producto



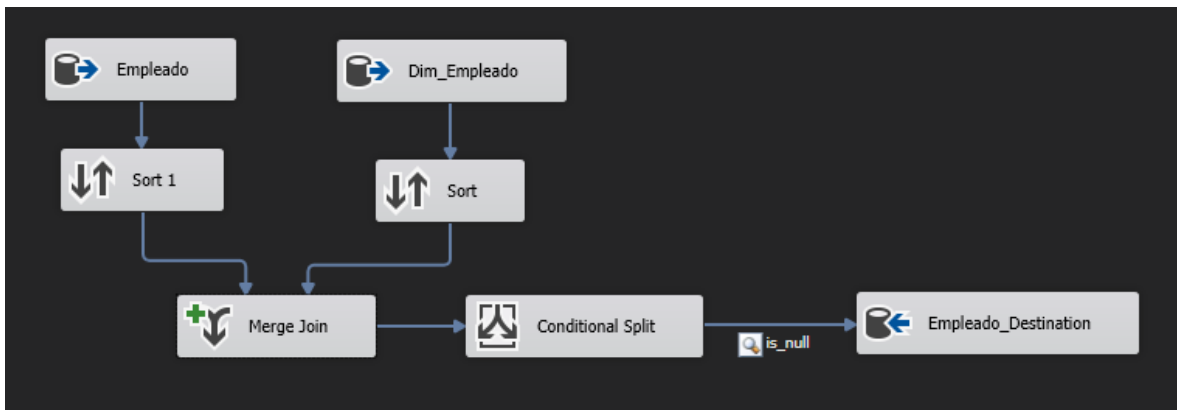
### 3.2 ETL Incremental FactVentas



### 3.3 ETL Detalle

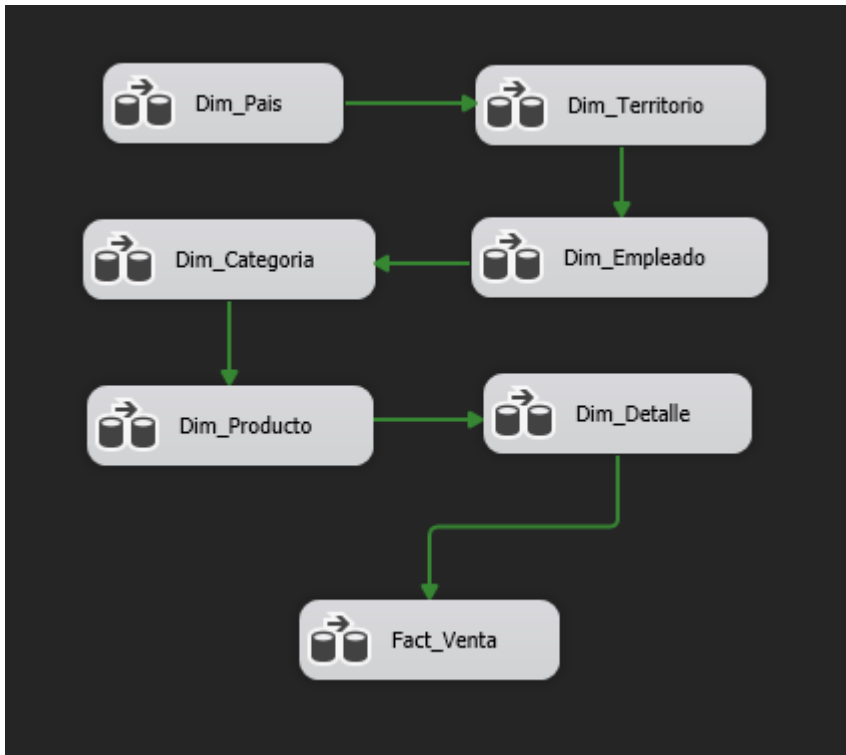


### 3.4 ETL Empleado

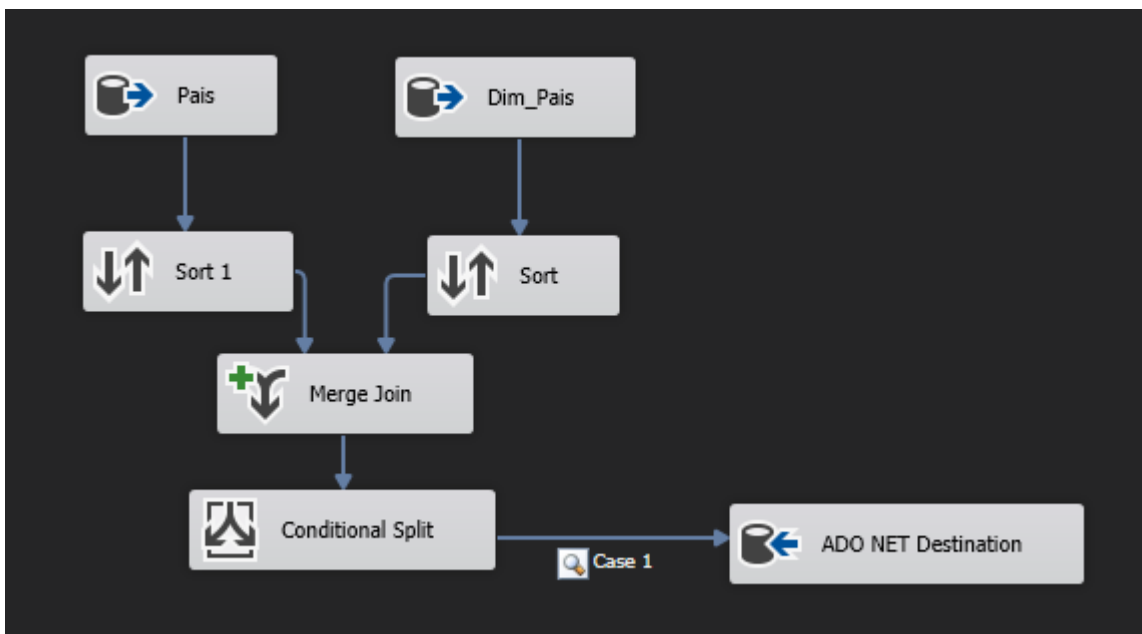




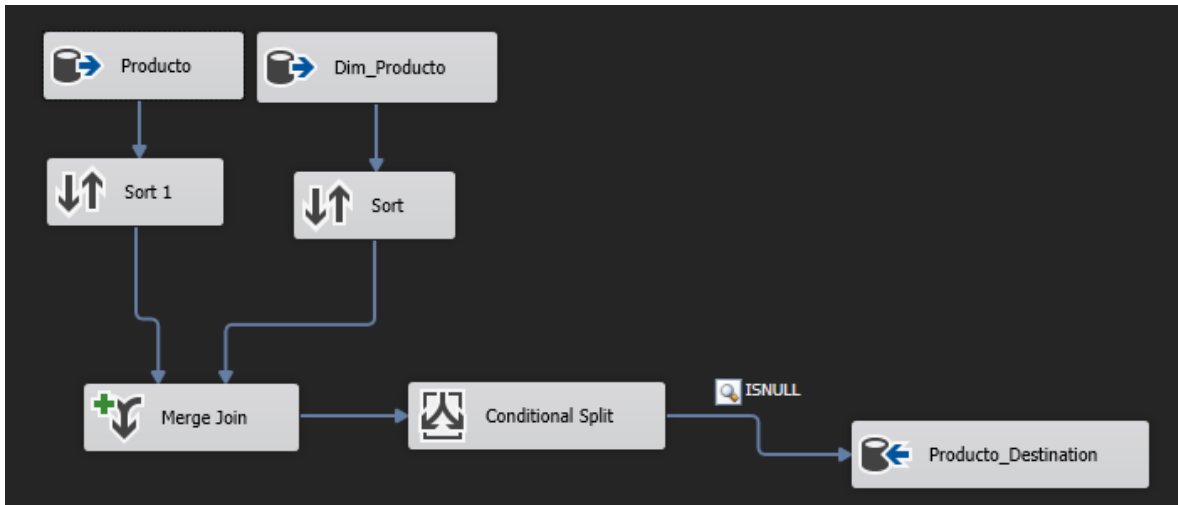
### 3.5 ETL Flow



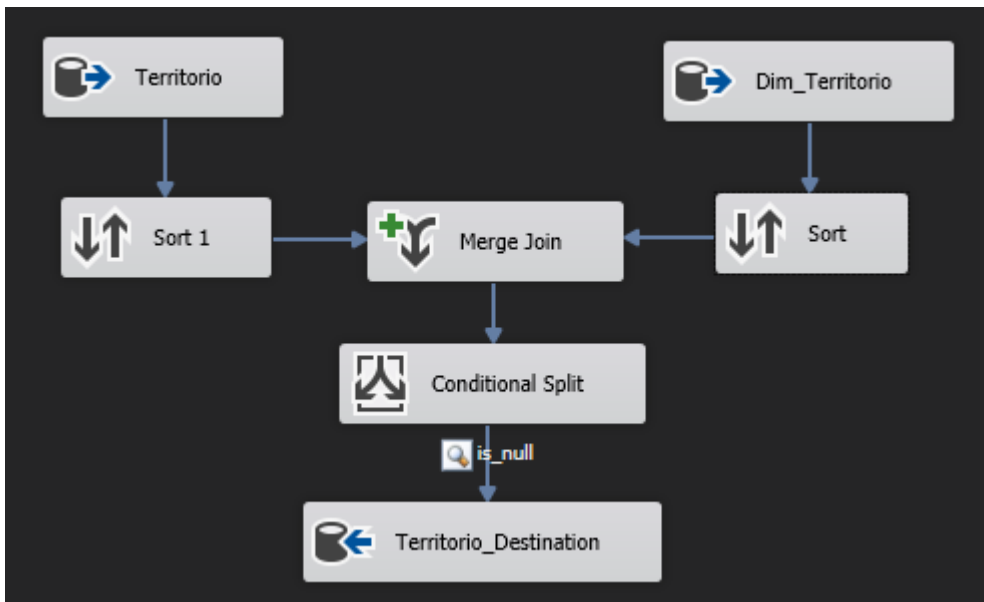
### 3.6 ETL País



### 3.7 ETL Producto



### 3.8 ETL Territorio



## 4.-Cubo OLAP

Un cubo OLAP es una estructura de datos que supera las limitaciones de las bases de datos relacionales y proporciona un análisis rápido de datos. Los cubos pueden mostrar y sumar grandes cantidades de datos, a la vez que proporcionan a los usuarios acceso mediante búsqueda a los puntos de datos.

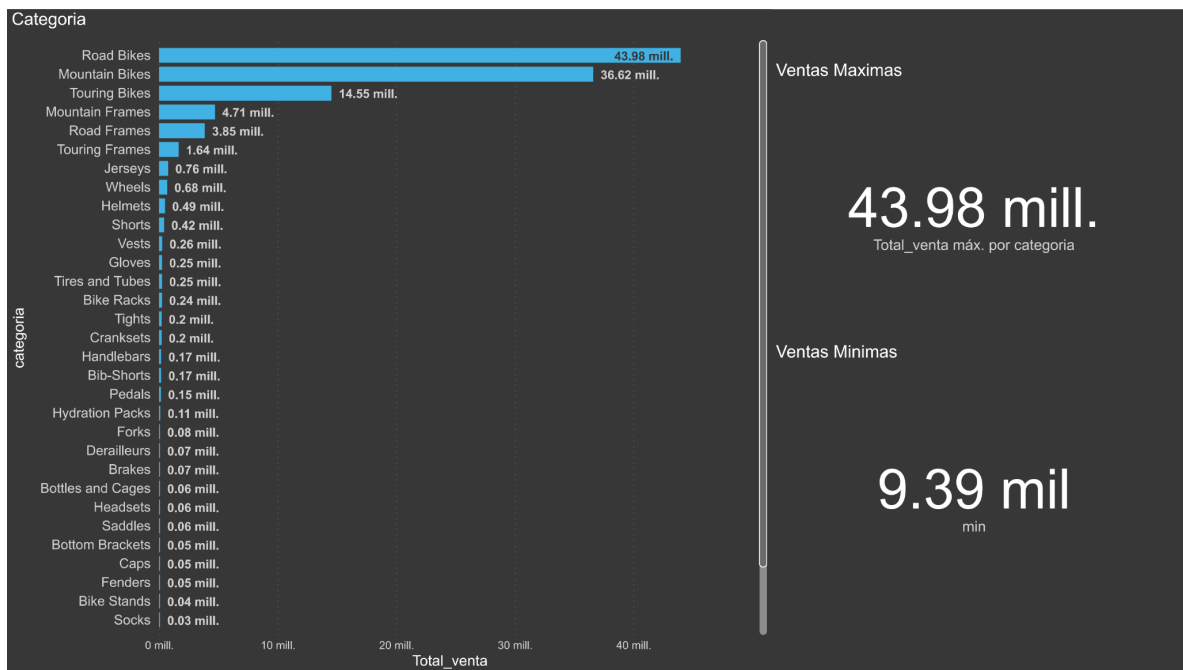
Para poder responder los KPI's referentes a nuestro proyecto tomamos los siguientes puntos:

### 4.1 Productos más vendidos

Esto no ayudará para determinar si el negocio puede llegar a especializarse en un rubro específico dentro del mundo de la venta de bicicletas y refacciones.

La categoría más vendida de la empresa Bike Z son las Mountain Bikes , mientras que la categoría que le dio más ventas a la empresa fue Road Bikes .

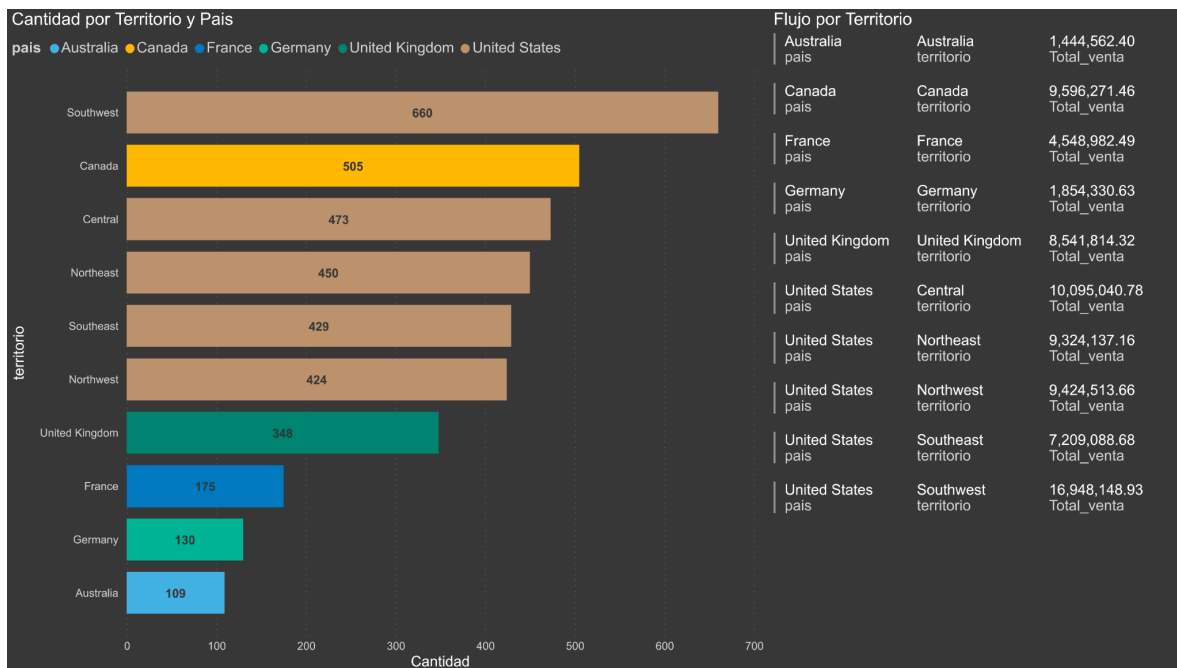




## 4.2 Territorios con más ventas

Con esta información podrán saber en qué país y territorio conviene más realizar una expansión o invertir en más y mejores sucursales.

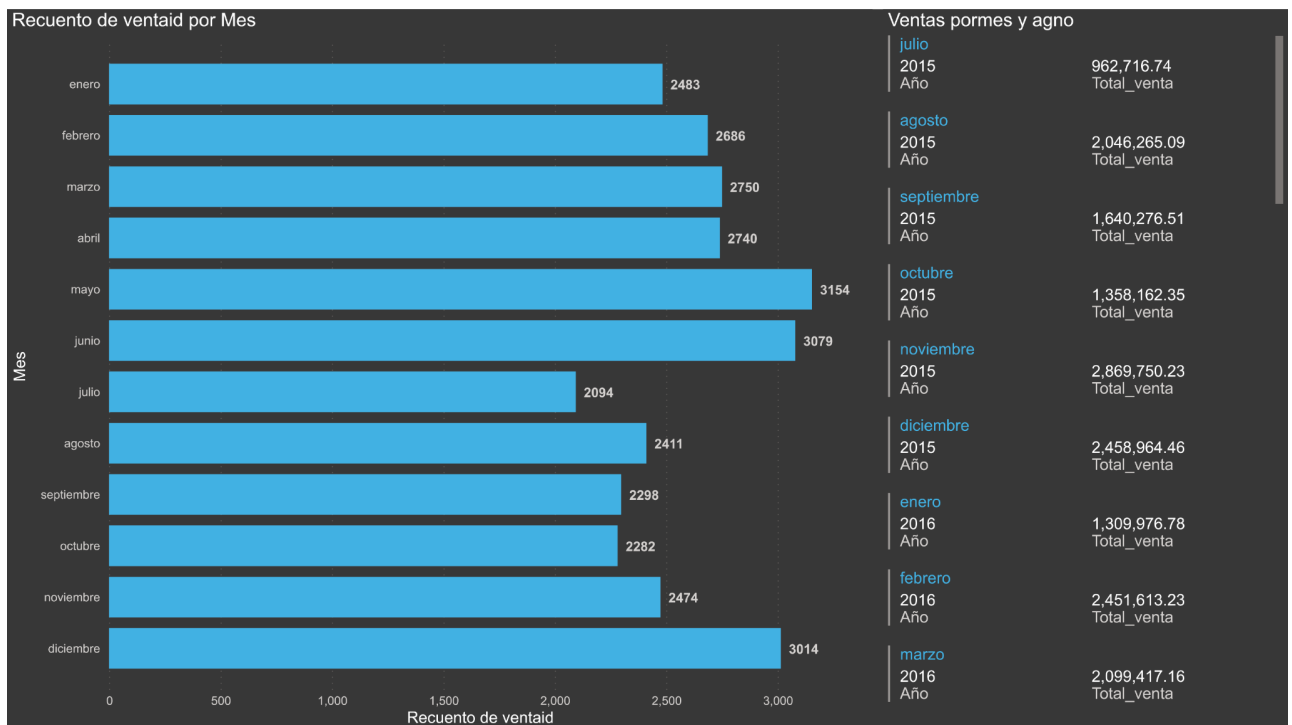
**El territorio con más venta corresponde a United States con 5 sucursales en la que más conviene invertir sería en la sucursal de SouthWest.**



### 4.3 Fechas con mayores ventas

En muchas industrias existen épocas de año especialmente buenas en ventas, con este KPI podremos identificarlas y adelantarnos a los pedidos con un stock más preparado.

**El mes que mejor le fue a la empresa Bike Z fue el mes mayo y el mes con menos ventas fue julio.**



## 5.- Conclusión

En este informe final se logra demostrar los aspectos desde el modelo de negocio de un sistema establecido a una reorganización estructural para un mayor rendimiento, analizando la gestión de los KPI's inculcados en el modelo de negocio, como la estructuración y tipo del Modelo Dimensional, y el cómo se lleva a cabo los datos cuando se llevan a cabo las funciones en los Procesos ETL para lograr así un mejor funcionamiento a la hora de extraer, cargar , y modificar los datos de la BD. y poniendo en práctica el CUBO OLAP para así una mejor gestión y visualización de los KPI's

Link de los archivos de github actualizado entrega examen transversal :

<https://github.com/Ratapan/Inteligencia-de-negocios>