

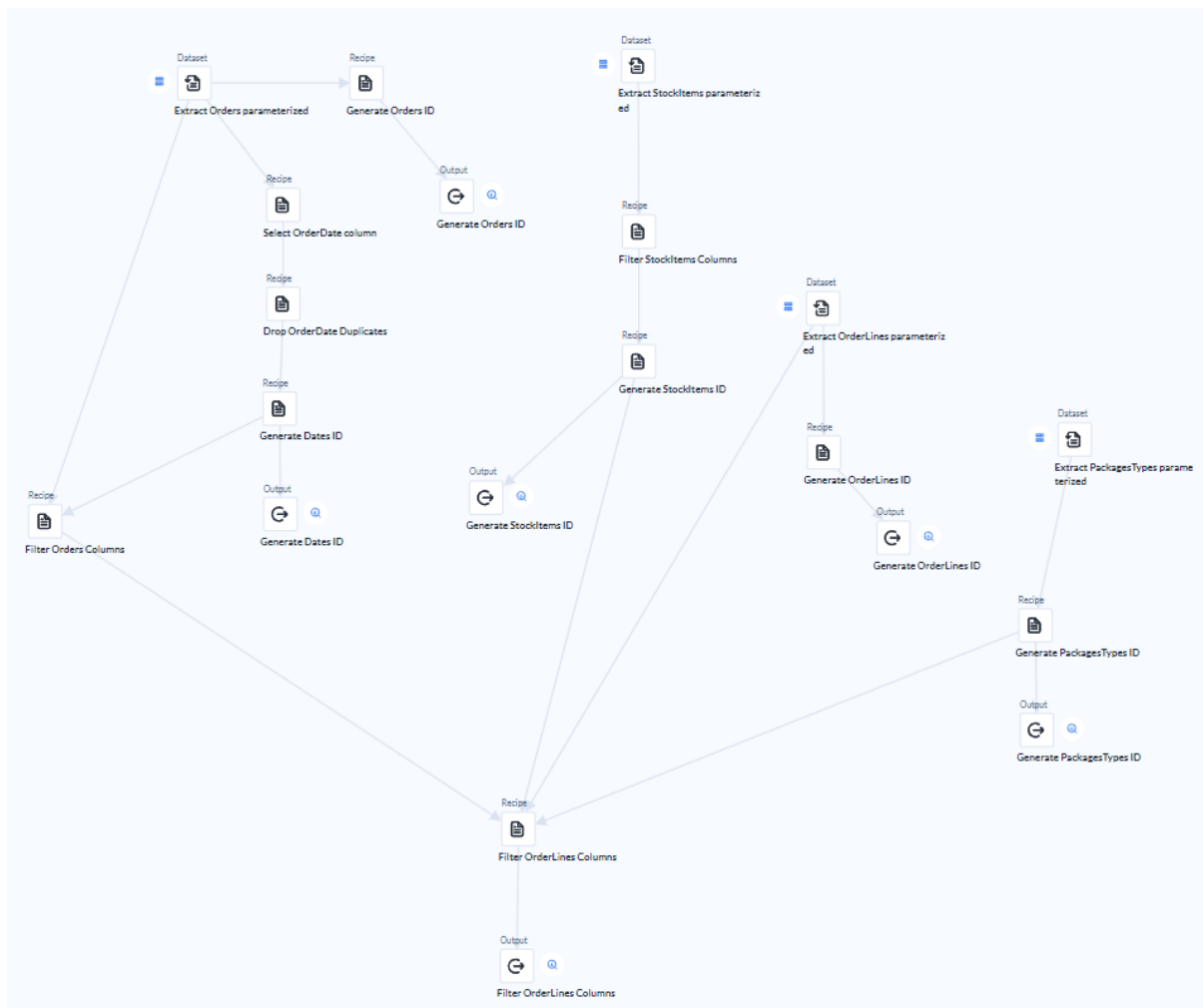
Laboratorio 3 – Inteligencia de Negocios

Estudiante 1: Ingrith Barbosa - 201712085

Estudiante 2: Daniela Camacho - 202110974

Estudiante 3: Daniel Acevedo - 201910941

1. Diseño del ETL Final



2. Descripción del ETL

El proceso del ETL involucra la extracción de datos de las cuatro diferentes fuentes de datos que tenemos, la transformación de estos datos en un formato compatible y la carga de estos datos transformados en un almacén de datos.

Extracción:

El proceso de extracción comienza con la extracción de cuatro fuentes diferentes:

1. **Dataset Orders** que tiene las columnas OrderID, CustomerID, SalespersonPersonID, PickedByPersonID, ContactPersonID,

BackorderOrderID, OrderDate, ExpectedDeliveryDate, CustomerPurchaseOrderNumber, IsUndersupplyBackordered, Comments, DeliveryInstructions, InternalComments, PickingCompletedWhen, LastEditedBy, LastEditedWhen.

2. **Dataset StockItem** que tiene las columnas StockItemID, StockItemName, SupplierID, ColorID, UnitPackageID, OuterPackageID, Brand, Size, LeadTimeDays, QuantityPerOuter, IsChillerStock, Barcode, TaxRate, UnitPrice, RecommendedRetailPrice, TypicalWeightPerUnit, MarketingComments, InternalComments, Photo, CustomFields, Tags, SearchDetails, LastEditedBy, ValidFrom, ValidTo.
3. **Dataset OrderLines** que tiene las columnas OrderLineID, OrderID, StockItemID, Description, PackageTypeID, Quantity, UnitPrice, TaxRate, PickedQuantity.
4. **Dataset PackagesTypes** que tiene las columnas PackageTypeID, PackageTypeName, LastEditedBy, ValidFrom, ValidTo.

Transformación:

Para cada uno de los datasets se aplican determinadas recetas, para el dataset de orders se le aplica la receta de generar los ordersID, esto se hace con la intención de crear una tabla que solamente tenga tres columnas, el orderID, el customerPurchaseOrderNumber y el IsUndersupplyBackordered. Otra receta que se le aplica a este dataset es para crear una tabla OrderDate con cuatro columnas: Date, Year, Month y Day. A esta tabla OrderDate se le aplica otra receta para eliminar los duplicados para conservar la integridad de los datos y luego, una última receta es agregar a esta tabla OrderDate la columna DateID para poder identificar cada date de las orders. La última receta que se le aplica al dataset Orders es un inner join entre el orderID y el dateID creado anteriormente para relacionar cada orden con una fecha.

Para el dataset StockItems se aplica primero una receta que permita crear una tabla de 5 columnas: StockItemID, StockItemName, ColorID, SizeLast, PriceLast. La siguiente receta para esta nueva tabla de StockItems filtrada se cambia el nombre StockItemID a StockItemNK (Natural Key) y se crea la columna StockItemID por el número de fila para manejar mejor la información.

Para el dataset OrderLines solo se le aplicó la receta para crear una tabla OrderLinesID donde están dos columnas: OrderLineID y Description.

Para el dataset de PackagesTypes se realizó de igual manera una sola receta, esta receta crea una tabla PackagesTypesID que tiene dos columnas: PackageTypeID y PackageTypeName.

Por último, ya teniendo las diferentes tablas creadas y modificadas se crea una tabla FilterOrderLines con los siguientes pasos:

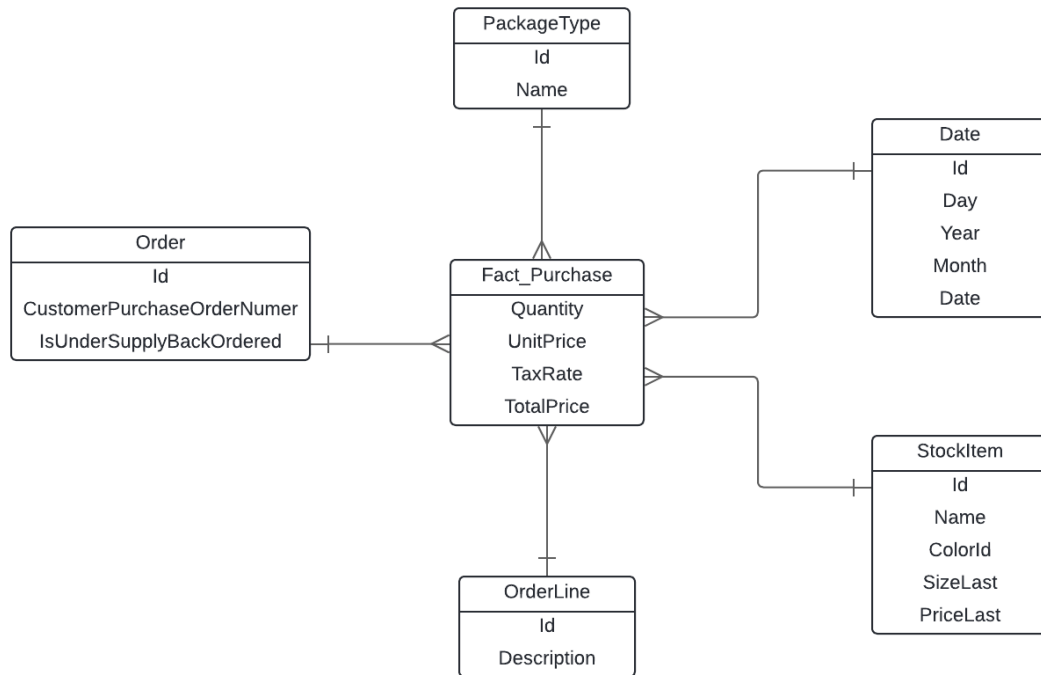
1. Crear una tabla con 7 columnas.
2. Realizar una unión interna (INNER JOIN) con la tabla “Generate StockItems” utilizando la columna “StockItemID” igual a “StockItemNK”.
3. Realizar otra unión interna con la tabla “Orders Columns” utilizando la columna “OrderID” igual a “OrderID”.
4. Calcular el precio total (TotalPrice) como el producto de la cantidad (Quantity) y el precio unitario original (UnitPrice). Este valor se redondea a dos decimales.
5. Finalmente, realizar una unión interna con la tabla “Generate PackagesTypes” utilizando la columna “PackageTypeID” igual a “PackageTypeID”.

En resumen, la tabla queda con las columnas OrderID, OrderLineID, StockItemID, PackageTypeID, DateID, Quantity, UnitPrice, TaxRate y TotalPrice.

Carga:


Por último, los datos transformados se cargan a BigQuery corriendo cada “Job” de los diferentes outputs creados y las tablas cargadas finalmente son: Fliter_OrderLines_Columns, Generate_Dates_ID, Generate_OrderLines_ID, Generate_Orders_ID, Generate_PackagesTypes_ID, Generate_StockItems_ID.

3. Modelo multidimensional





El modelo multidimensional resultante sería una estructura estrella típica, donde la tabla de hechos (**Fact_Purchase**) está rodeada por varias dimensiones (**Dim_Date**, **Dim_StockItem**, **Dim_PackageType**, **Dim_Order** y **Dim_OrderLine**). Este modelo proporciona una estructura analítica eficaz para analizar las ventas y las transacciones de pedidos en función de diferentes dimensiones como fecha, artículo y tipo de paquete.

4. Consultas SQL



Los diez productos más vendidos

 EJECUTAR

 GUARDAR CONSULTA

```
1 #Los diez productos más vendidos
2 SELECT f.StockItemID, s.StockItemName, SUM(f.Quantity) AS TotalQuantity
3 FROM `lab3bi.DatosLab3.Filter_OrderLines_Columns` f
4 JOIN `lab3bi.DatosLab3.Generate_StockItems_ID` s ON f.StockItemID = s.StockItemID
5 GROUP BY f.StockItemID, s.StockItemName
6 ORDER BY TotalQuantity DESC
7 LIMIT 10;
8
```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO		RESULTADOS	GRÁFICO	JSON	DETALLES DE L
Fila	StockItemID	StockItemName	TotalQuantity		
1	191	Black and orange fragile despatch tape 48mmx75m	207324		
2	192	Black and orange fragile despatch tape 48mmx10...	193680		
3	189	Clear packaging tape 48mmx75m	158626		
4	188	3 kg Courier post bag (White) 300x190x95mm	152375		
5	185	Shipping carton (Brown) 356x356x279mm	152125		
6	184	Shipping carton (Brown) 305x305x305mm	151875		
7	187	Express post box 5kg (White) 350x280x130mm	149825		
8	177	Shipping carton (Brown) 413x285x187mm	147675		
9	179	Shipping carton (Brown) 229x229x229mm	146375		
10	186	Shipping carton (Brown) 457x457x457mm	144950		

Mes con mayor cantidad de productos vendidos
EJECUTAR

```

1 #Mes con mayor cantidad de productos vendidos
2 SELECT s.Month
3 FROM `lab3bi.DatosLab3.Filter_OrderLines_Columns` f
4 JOIN `lab3bi.DatosLab3.Generate_Dates_ID` s ON f.DateID = s.DateID
5 GROUP BY s.Month
6 ORDER BY SUM(f.Quantity) DESC
7 LIMIT 1;

```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO	RESULTADOS	GRÁFICO	JSON	DETALL
Fila	Month			
1	05			

Tipo de paquete más pedido por los clientes
EJECUTAR
GUARDAR C

```

1 #Tipo de paquete más pedido por los clientes
2 SELECT s.PackageTypeID, s.PackageTypeName
3 FROM `lab3bi.DatosLab3.Filter_OrderLines_Columns` f
4 JOIN `lab3bi.DatosLab3.Generate_PackagesTypes_ID` s ON f.PackageTypeID = s.PackageTypeID
5 GROUP BY s.PackageTypeID, s.PackageTypeName
6 ORDER BY SUM(f.Quantity) DESC
7 LIMIT 1;

```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO	RESULTADOS	GRÁFICO	JSON	DETALLES DE LA EJECU
Fila	PackageTypeID	PackageTypeName		
1	7	Each		

5. Preguntas

- ¿Qué diferencia existe entre una arquitectura Data warehouse y un Data lakehouse?

Aunque ambos cuentan con la funcionalidad de almacenar y gestionar datos, el data warehouse trabaja con estructuras de datos estructuradas, por lo que tiene un esquema muy bien definido. Mientras que el data lakehouse da más flexibilidad con respecto a la estructura de los datos por lo que soporta los datos sin procesar y procesados, este es una combinación entre el data warehouse y el data lake.

- ¿Qué tipo de arquitectura le recomendaría a WWI para complementar lo desarrollado en este laboratorio incluyendo fuentes de datos no estructuradas, análisis en tiempo real que incluyan simultáneamente tanto datos estructurados como no estructurados?

La arquitectura recomendada para la empresa es el data lakehouse, ya que su flexibilidad permite el ingreso de datos en tiempo real. Esta arquitectura facilita que los datos entren al lake en tiempo real, independientemente de su estructura original. Esto permite que el análisis de estos datos sea más eficiente lo que conlleva un beneficio para el negocio

Además del ingreso de datos en tiempo real, el data lakehouse ofrece la ventaja adicional de poder incorporar datos estructurados en el mismo entorno. Esto significa que la empresa puede almacenar tanto datos no estructurados como estructurados en un único repositorio de datos. Esto permite tener un entorno de datos más completo para el negocio lo que le permite tener una visión mejor de la información ingresada, teniendo así un mejor análisis.

- ¿Qué ventajas y desventajas observa al momento de implementar un ETL utilizando este tipo de herramientas respecto a desarrollarlo utilizando Python, Pandas y demás herramientas vistas durante la primera parte del curso?

Ventajas:

La creación del preprocesamiento es mucho más fácil y eficiente en herramientas como DataPrep. No se requiere de múltiples líneas de código para elaborarlo, se observa el resultado de cada paso del procesamiento según se realiza y poder ver una representación gráfica del modelo también hace más fácil entenderlo. Además, esta herramienta permite ver más fácilmente los temas de calidad de datos como la cantidad de nulos, distribución de datos y el tipo de cada dato.

Desventajas:

Cada vez que se modifica la base de datos o que se quiere agregar un paso a la transformación de los datos se deben eliminar las tablas creadas anteriormente y volver a correr el job. Además, en algunos casos en los que había que modificar una recipe era necesario modificar más de una, por ejemplo, para que el id necesario para hacer un join aparezca en la tabla necesaria.

- ¿Qué tipo de esquema, estrella o copo de nieve, representa el modelo multidimensional construido en este laboratorio? Justifica tu respuesta.

El esquema que hemos construido se asemeja más a un esquema estrella que a un esquema copo de nieve. En un esquema estrella, una tabla de hechos central está rodeada de tablas de dimensiones, y cada dimensión está conectada directamente a la tabla de hechos. En nuestro caso, la tabla de hechos es Fact_OrderLines, y las dimensiones son Dim_Order, Dim_Date,

Dim_StockItem y Dim_PackageType, todas conectadas directamente a Fact_OrderLines.

- ¿Qué tipo de tablas de hechos y de medidas se identifican en el modelo multidimensional dado? Justifica tu respuesta.

En el modelo multidimensional proporcionado, la tabla central de hechos es Fact_OrderLines, que encapsula las transacciones individuales de pedidos. Esta tabla almacena información detallada sobre cada línea de pedido, incluyendo cantidades, precios unitarios, tasas impositivas y el precio total. Estas métricas cuantitativas son esenciales para analizar el desempeño de las ventas y realizar cálculos relevantes en el contexto del negocio.

Las medidas clave como Quantity, UnitPrice, TaxRate y TotalPrice representan las unidades vendidas, los precios por unidad, las tasas impositivas aplicadas y el precio total de cada línea de pedido, respectivamente. Estas medidas proporcionan una visión detallada de las transacciones de pedidos y son fundamentales para comprender el rendimiento y las tendencias comerciales.

Por lo tanto, en el modelo multidimensional dado, Fact_OrderLines sirve como el punto focal para analizar las ventas y las transacciones de pedidos, mientras que las medidas identificadas ofrecen una comprensión cuantitativa profunda de estas transacciones, facilitando análisis significativos y toma de decisiones informada.

- Suponga que la dimensión StockItemDim cambia el manejo de la historia de tamaño y precio del producto a un tipo 2 (Slow Change Dimension). ¿Qué ajustes a la dimensión relacionada con el producto, a la tabla de hechos y al proceso ETL se deben realizar para que al cargar la información se incluya este manejo de historia?

Dimensión StockItemDim:

Agregar campos de versión como StartDate y EndDate para rastrear el historial de cambios.

Incluir un campo de estado para marcar la versión actual y las anteriores del producto.

Tabla de Hechos Fact_OrderLines:

Actualizar las llaves foráneas para reflejar el cambio en la dimensión, posiblemente usando llaves compuestas que incluyan el ID del producto y la fecha efectiva.

Agregar campos de referencia como ProductVersionID para identificar la versión específica del producto en el momento de la transacción.

Proceso ETL:

Implementar lógica para detectar cambios en los atributos del producto y crear nuevas versiones en la dimensión con fechas de inicio y fin actualizadas.

Modificar las consultas de carga para incluir la lógica de historificación y asegurar la correcta relación entre las nuevas versiones de los productos y la tabla de hechos.

Ajustar las consultas de informes para considerar el historial de cambios en la dimensión del producto, permitiendo análisis tanto de la versión actual como de versiones anteriores.

- ¿Qué ajustes al proceso ETL construido en este laboratorio hay que realizar para cargar nueva información que sea reportada por WWI? ¿Esto se considera en la literatura una carga incremental?

Para agregar nueva información al etl hay que modificar los csv originales y volver a correr los jobs que utilizan la nueva información. Dado solo se están cargando los datos nuevos o modificados en lugar de volver a cargar todo el conjunto de datos, se puede considerar que es carga incremental.

- ¿Qué errores se le presentaron en el desarrollo del laboratorio y qué solución plantearon? Haga énfasis en los que fueron más difíciles de solucionar.

Para agregar la nueva base de datos al etl fue necesario modificar más de una recipe del proceso. Esto con el fin de que el id de PackageType apareciera en la tabla final y poder hacer el join. Adicionalmente, para poder usar todas las dimensiones en BigQuery fue necesario agregar los outputs correspondientes. En este proceso no hubo errores, pero hace el etl más robusto.

Bibliografía:

- <https://aws.amazon.com/es/what-is/data-lake/#:~:text=Data%20lakes%20allow%20you%20to,lake%20in%20its%20original%20format.>
- <https://www.montecarlodata.com/blog-data-warehouse-vs-data-lake-vs-data-lakehouse-definitions-similarities-and-differences/>