

Laboratorio 3 – ETL

Santiago Martínez Novoa – 202112020

Marilyn Stephany Joven Fonseca – 202021346

María Alejandra Estrada García – 202021060

Tabla de contenido

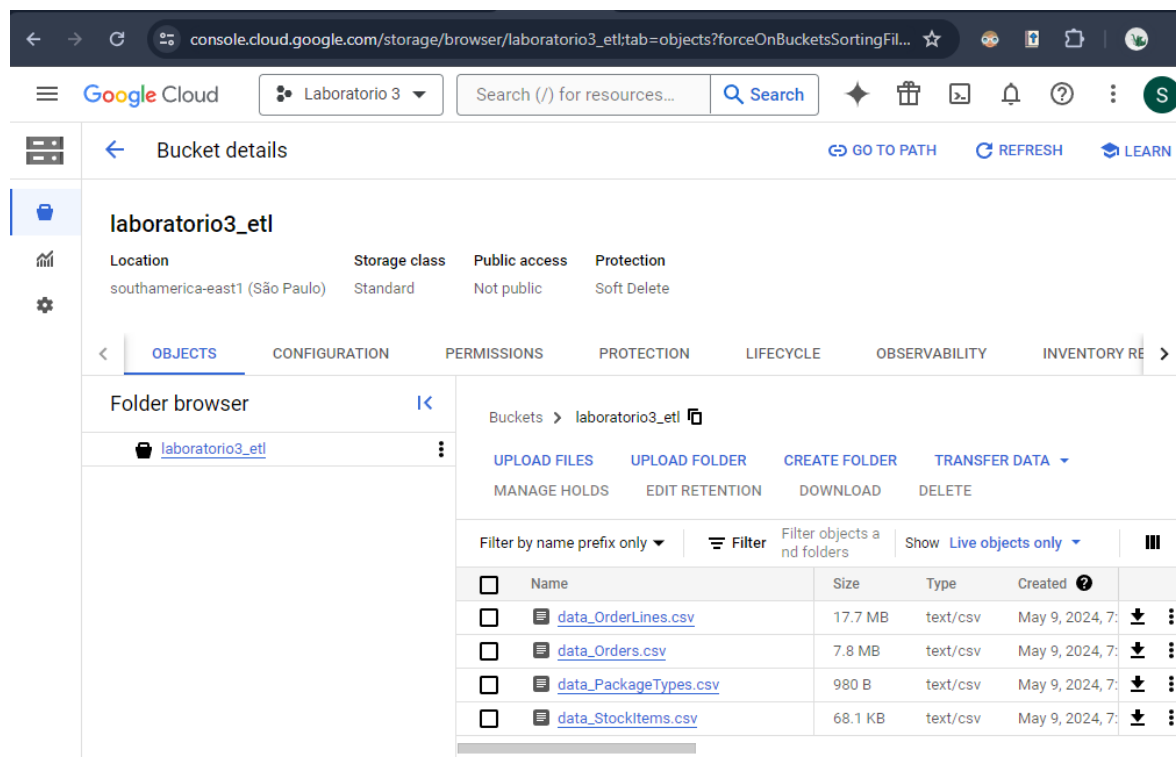
Creación de la Arquitectura	2
Google Cloud Storage:.....	2
Google Cloud DataPrep:.....	2
GoogleCloudBigQuery:	3
Descripción Paso a Paso del ETL inicial	3
Orders.....	3
StockItems.....	4
OrderLines.....	4
Diseño e implementación de extensión del ETL	4
JSON de ETL final y Excel de transformación.....	7
Modelo multidimensional	9
Consultas SQL.....	10
Respuestas a las preguntas	15
Referencias	18

Creación de la Arquitectura

Tras haber completado la etapa 1 y 2 del laboratorio se puede evidenciar la correcta construcción de la arquitectura planteada dentro de las instrucciones. A continuación, se mostrarán las capturas de pantalla de cada una de las aplicaciones utilizadas en GCP.

Google Cloud Storage:

Se configuró el data lake y se creó el Bucket donde se guardaron todos los archivos csv.

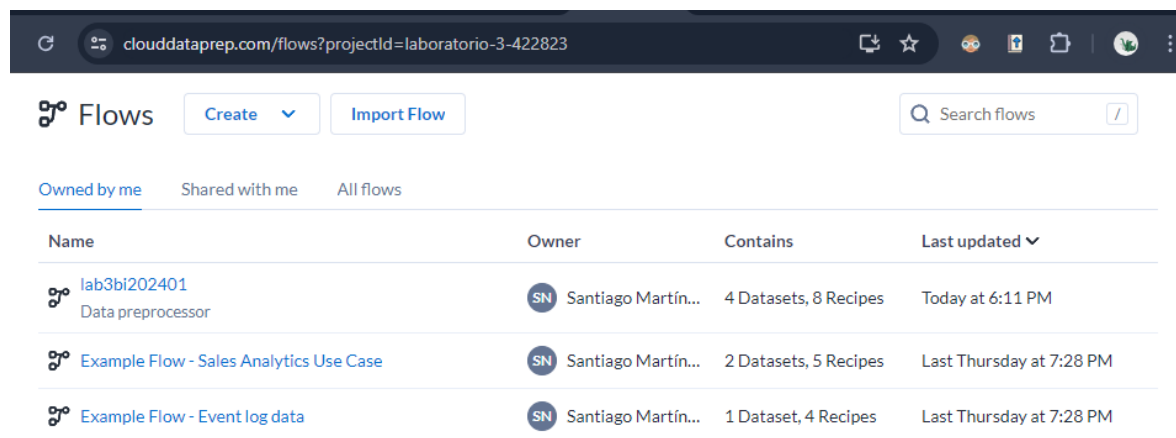


The screenshot shows the Google Cloud Storage console interface. The top navigation bar includes the Google Cloud logo, a project selector for 'Laboratorio 3', a search bar, and various utility icons. The main content area is titled 'Bucket details' for the bucket named 'laboratorio3_etl'. It displays metadata: Location (southamerica-east1 (São Paulo)), Storage class (Standard), Public access (Not public), and Protection (Soft Delete). Below this, a tabbed interface shows 'OBJECTS' selected. On the left, a 'Folder browser' shows the bucket path. On the right, there are buttons for 'UPLOAD FILES', 'UPLOAD FOLDER', 'CREATE FOLDER', 'TRANSFER DATA', 'MANAGE HOLDS', 'EDIT RETENTION', 'DOWNLOAD', and 'DELETE'. A table lists the objects in the bucket, filtered by name prefix.

<input type="checkbox"/>	Name	Size	Type	Created	
<input type="checkbox"/>	data_OrderLines.csv	17.7 MB	text/csv	May 9, 2024, 7:	
<input type="checkbox"/>	data_Orders.csv	7.8 MB	text/csv	May 9, 2024, 7:	
<input type="checkbox"/>	data_PackageTypes.csv	980 B	text/csv	May 9, 2024, 7:	
<input type="checkbox"/>	data_StockItems.csv	68.1 KB	text/csv	May 9, 2024, 7:	

Google Cloud DataPrep:

Se sube a DataPrep el flow inicial provisto en el enunciado y se configuran los datasets.

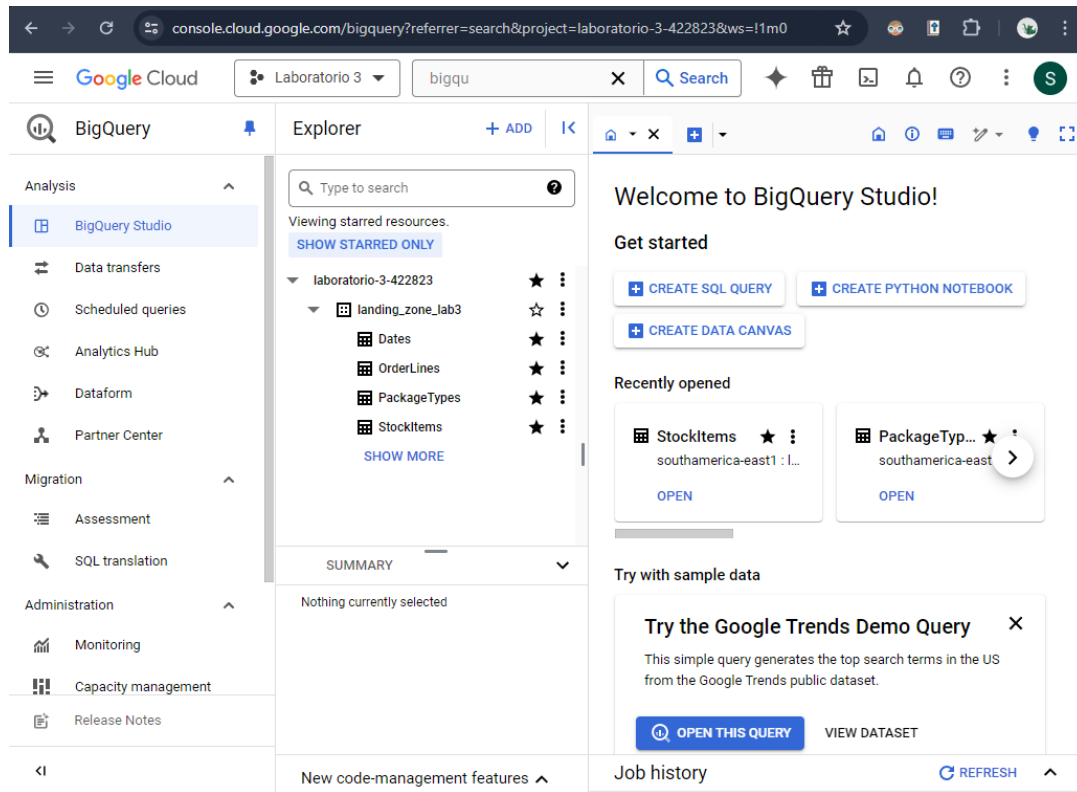


The screenshot shows the Google Cloud DataPrep console. The top navigation bar includes the Google Cloud logo, a project selector for 'Laboratorio 3', a search bar, and various utility icons. The main content area is titled 'Flows'. It displays a list of flows, including 'lab3bi202401' (Data preprocessor), 'Example Flow - Sales Analytics Use Case', and 'Example Flow - Event log data'. Each flow entry shows the owner (Santiago Martín...), the number of datasets and recipes, and the last updated time.

Name	Owner	Contains	Last updated
lab3bi202401 Data preprocessor	Santiago Martín...	4 Datasets, 8 Recipes	Today at 6:11 PM
Example Flow - Sales Analytics Use Case	Santiago Martín...	2 Datasets, 5 Recipes	Last Thursday at 7:28 PM
Example Flow - Event log data	Santiago Martín...	1 Dataset, 4 Recipes	Last Thursday at 7:28 PM

GoogleCloudBigQuery:

Se crea el conjunto de datos bajo el nombre de *landing_zone_lab3*.



Descripción Paso a Paso del ETL inicial

El ETL recibe originalmente tres conjuntos diferentes de datos, Orders, StockItems y OrderLines. De acuerdo con esto se separan las transformaciones para cada archivo.

Orders

En primer lugar, se utiliza la columna order date para crear otras tres columnas, la primera consiste en utilizar la función Year (para hallar el año de la orden), luego, para la nueva columna de mes, se comprueba si es menor a 10 y en caso de serlo se une el número del mes con un cero, de lo contrario, simplemente se aplica la función month(); lo mismo sucede con day(), se comprueba que sea un solo dígito y de ser necesario se agrega un cero a la izquierda.

El siguiente paso recibe estas tres columnas nuevas y la columna original, posteriormente, se borran los duplicados. El resultado de esta operación resulta en todas las fechas en las que se realizó una orden, sin que se repitan.

En el siguiente paso se crea un identificador único para cada combinación año, mes y día. Este resultado se guarda en la tabla Dates. Esta tabla hará referencia a la dimensión Date, esto es importante tenerlo en cuenta cuando se vaya a realizar o diseñar el modelo multidimensional.

Luego de estas transformaciones, se seleccionan dos columnas del conjunto de datos original (OrderID, OrderDate) y se hace un inner join con la tabla generada anteriormente. Esto resulta en que cada orden va a tener asociada una llave foránea a la tabla de fechas. Finalmente, lo que se pasa a la siguiente transformación son solo estas dos columnas (OrderID, DateID).

StockItems

Se utilizan 5 columnas: StockItemID, StockItemName, ColorID, SizeLast, PriceLast.

La primera transformación es que el StockItemID pasa a ser una Natural Key (StockItemNK), debido a que no es capaz de identificar de manera única a las filas de esta tabla, por lo cual es necesario generar una nueva columna de IDs únicos para cada fila. El resultado de la transformación se guarda en la tabla StockItems.

OrderLines

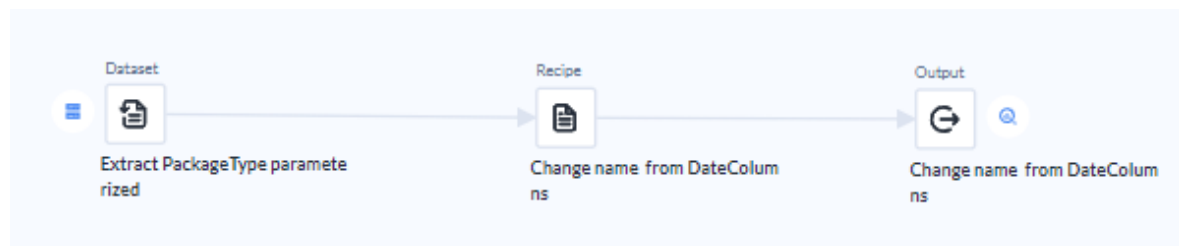
Lo primero que se hace es seleccionar las 5 columnas: OrderID, OrderLineID, StockItemId, Quantity, UnitPrice y TaxRate de los datos OrderLines. En el paso final se realiza un inner join entre StockItemID y StockItemNK de la tabla StockItem.

Luego se realiza un inner join entre OrderID y OrderID de las tablas OrderLines y Dates, esto nos permite añadir DateID a OrderLines. Además, se redondean todos los valores que no siguen el formato de float al entero más cercano. Finalmente quedan las 7 columnas: OrderID, OrderLineID, StockItemId, Quantity, UnitPrice, TaxRate y DateID.

Diseño e implementación de extensión del ETL

- De acuerdo con los requerimientos planteados, es claro que hay que realizar dos modificaciones al Flow que se tiene. En primer lugar, es necesario importar y parametrizar el dataset de PackageTypes. Como en el enunciado no se especifican las necesidades para este nuevo dataset se asume que todas las columnas que trae son relevantes para el negocio. Sin embargo, se recomienda al negocio replantear la existencia de la columna de ValueTo, pues si todos los PackageTypes van a ser válidos hasta el final de los tiempos lo mejor es eliminar esta columna. Así mismo, la columna ValueFrom también podría ser eliminado si el negocio no va a añadir más PackageTypes o no requiere saber desde cual fecha es válido cada tipo de paquete.

De esta manera el procesamiento de los datos para PackageTypes queda planteado en el Flow de la siguiente manera:



Como se mencionó anteriormente, la única transformación de los datos fue cambiarle el nombre de las columnas ValidFrom y ValidTo por nombres más dicentes. Esta información extraída queda guardada en la tabla con el nombre de PackageTypes.

Laboratorio 3 bigqu Search

Explorer + ADD

Type to search

Viewing starred resources.

SHOW STARRED ONLY

laboratorio-3-422823

landing_zone_lab3

Dates

OrderLines

PackageTypes

StockItems

SHOW MORE

SUMMARY

PackageTypes

laboratorio-3-422823.landing_zone_lab3

Last modified May 11, 2024, 6:12:15 PM UTC-5

Data location southamerica-east1

Description

Labels

PackageTypes

SCHEMA DETAILS PREVIEW LINEAGE DATA PROFILE

Filter Enter property name or value

Field name	Type	Mode	Key	Collation	Default Value
PackageTypeID	INTEGER	NULLABLE	-	-	-
PackageTypeName	STRING	NULLABLE	-	-	-
LastEditedBy	INTEGER	NULLABLE	-	-	-
ValidFromDate	STRING	NULLABLE	-	-	-
ValidUntilDate	STRING	NULLABLE	-	-	-

EDIT SCHEMA VIEW ROW ACCESS POLICIES

- Otras pequeñas modificaciones que se hicieron sobre el Flow original fue cambiar el tipo de la columna de year a String en la tabla de Dates, porque estaba siendo tomado como Datetime y generaba información que no era cierta: También se cambió el orden de las columnas para que el identificador de esa tabla quedara de primeras:

Formula

PARSESTRING(YEAR(OrderDate))

Column name

Year

Laboratorio 3

Explorer

Viewing starred resources.

laboratorio-3-422823

landing_zone_lab3

Dates

OrderLines

PackageTypes

StockItems

Dates

< SCHEMA DETAILS PREVIEW LINEAGE DATA PROFILE >

Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode	Key	Collation	Default Value	Pol
<input type="checkbox"/>	DateID	INTEGER	NULLABLE	-	-	-	-
<input type="checkbox"/>	Date	DATETIME	NULLABLE	-	-	-	-
<input type="checkbox"/>	Year	STRING	NULLABLE	-	-	-	-
<input type="checkbox"/>	Month	STRING	NULLABLE	-	-	-	-
<input type="checkbox"/>	Day	STRING	NULLABLE	-	-	-	-

- Por último, se implementó el cálculo de TotalPrice que se realizó por medio de la multiplicación de las columnas Quantity y UnitPrice de OrderLines, redondeando a la centésima más cercana.

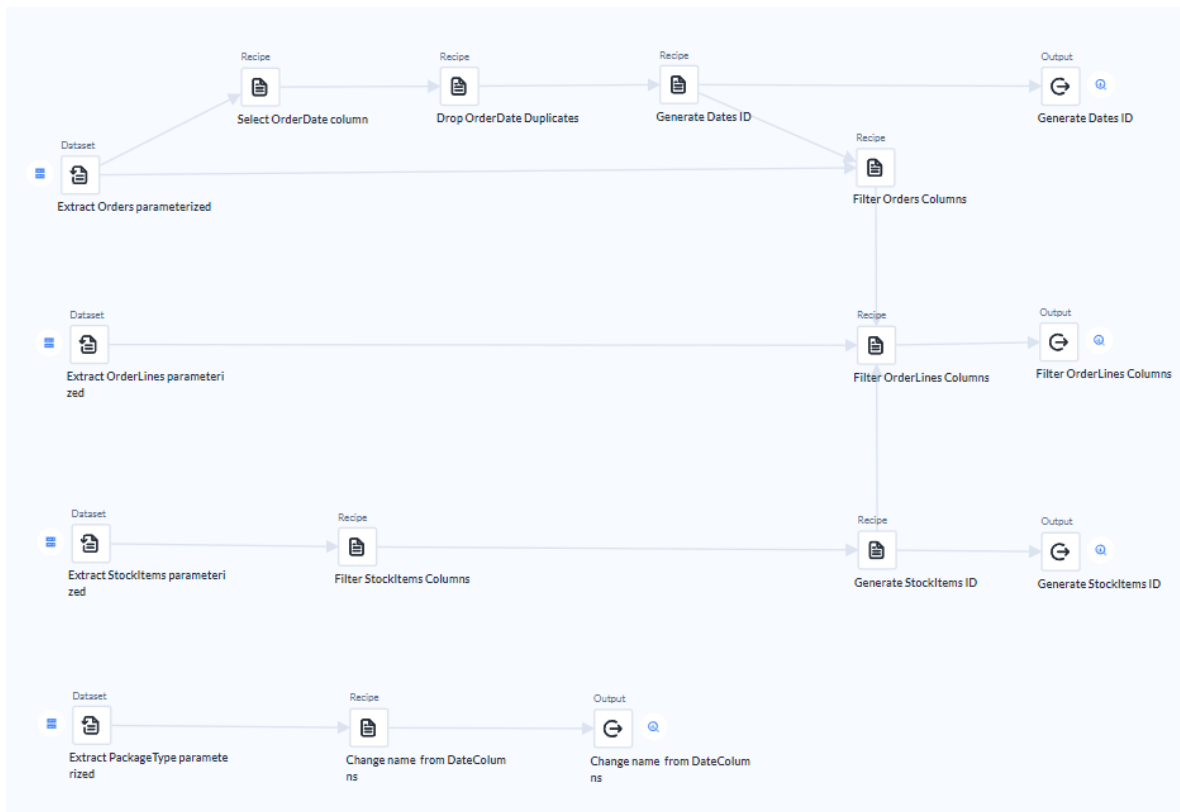
Formula

`ROUND(UnitPrice * Quantity, 2)`

Column name

`TotalPrice`

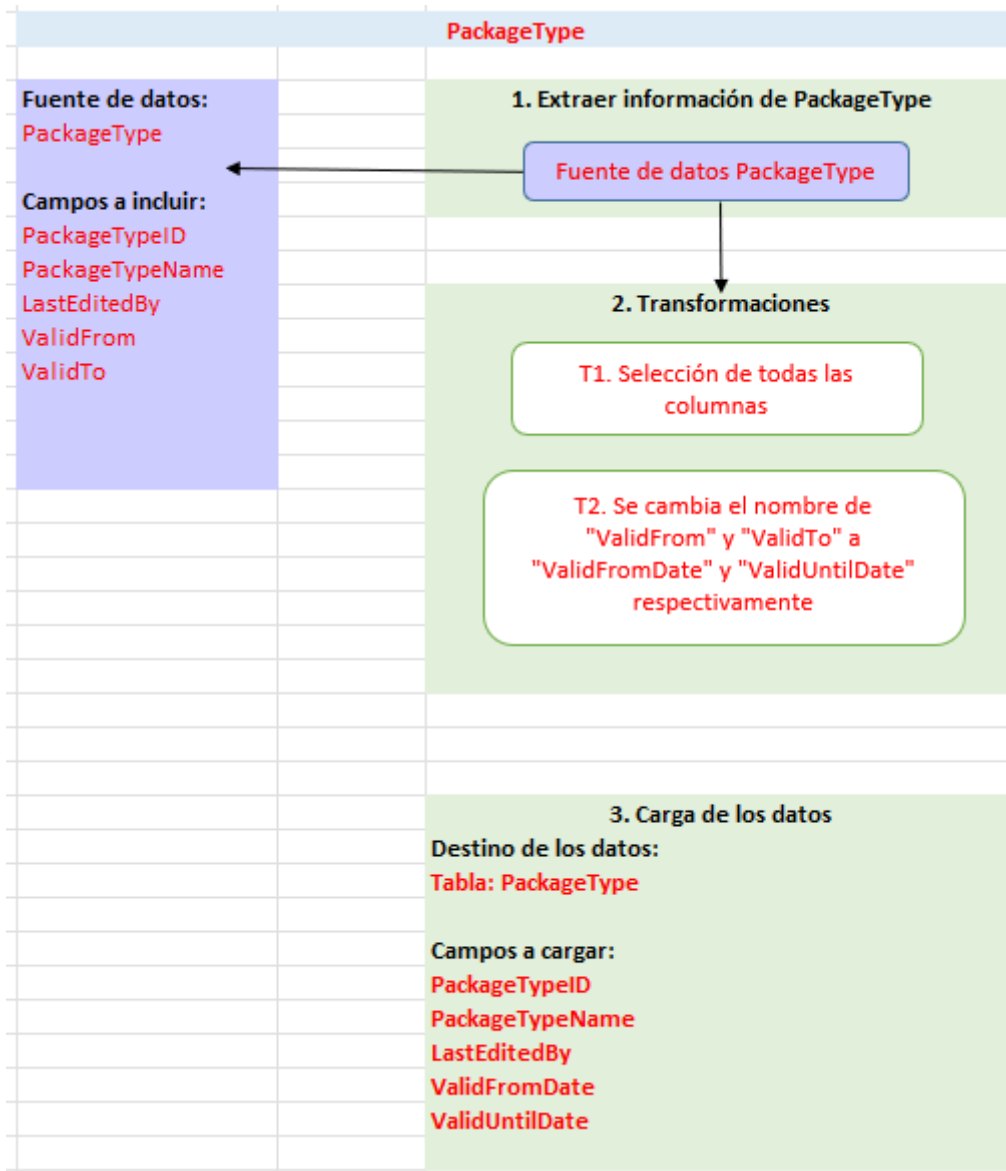
El ETL final queda entonces de la siguiente manera:



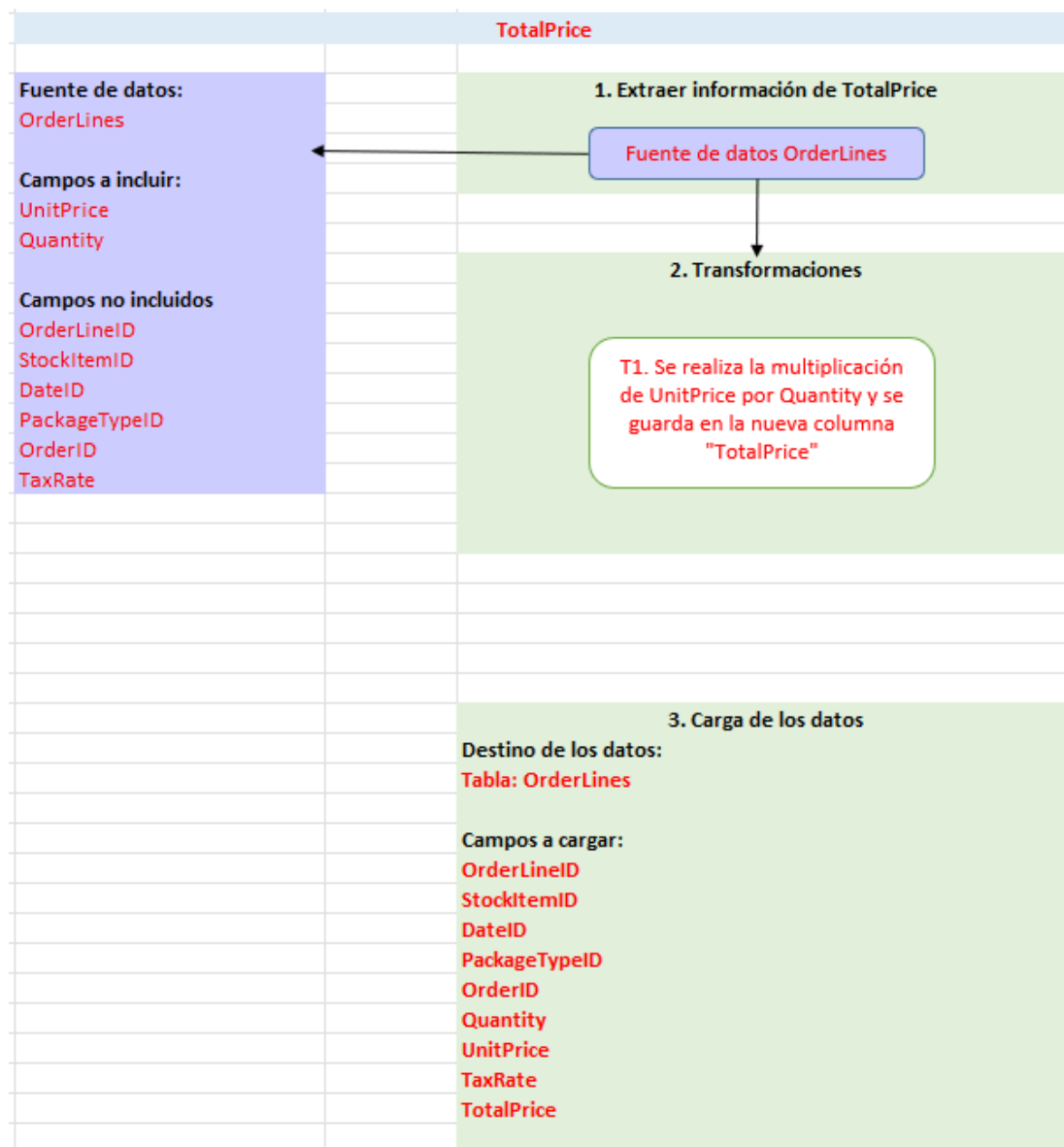
JSON de ETL final y Excel de transformación

El JSON resultado de la extensión del ETL se adjunta a esta entrega, así como la plantilla donde se documentó el diseño del ETL final (únicamente las partes modificadas).

A continuación, se evidencia el diseño ETL para la extracción de los datos PackageType, siguiendo las transformaciones explicadas anteriormente.

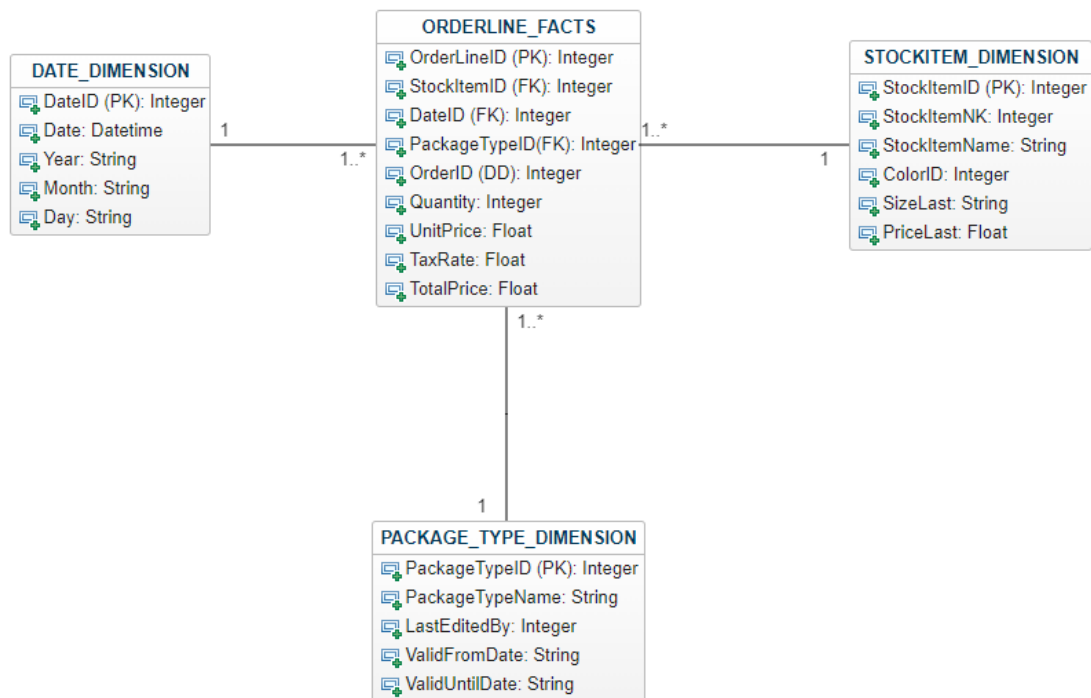


Seguidamente, se realizó el siguiente diseño ETL para el cálculo de TotalPrice, realizando la única transformación que implica el cálculo y la agregación de esta columna.



Modelo multidimensional

A continuación, se muestra el modelo multidimensional construido gracias al análisis de los procesos que se llevan a cabo en el ETL:



Como se puede observar, la tabla de hechos es OrderLine, donde se referencia mediante tres llaves foráneas todas las demás tablas (dimensiones). Gracias a este diagrama de estrella es posible establecer como nivel de granularidad que cada fila de este modelo representa una orden individual a la empresa distribuidora asociado a un elemento en bodega, para una fecha específica y en un tipo de paquete.

Consultas SQL

Para resolver el enunciado y satisfacer las necesidades de WWI, vamos a realizar las tres consultas necesarias:

Consulta 1: Ventas diarias por categoría de producto

```

1 SELECT
2     d.DateID,
3     pt.PackageTypeName AS Categoria,
4     SUM(ol.Quantity * ol.UnitPrice) AS Ventas
5 FROM
6     landing_zone_lab3.OrderLines ol
7 JOIN
8     landing_zone_lab3.StockItems si ON ol.StockItemID = si.StockItemID
9 JOIN
10    landing_zone_lab3.Dates d ON ol.DateID = d.DateID
11 JOIN
12    landing_zone_lab3.PackageTypes pt ON ol.PackageTypeID = pt.PackageTypeID
13 GROUP BY
14     d.DateID, Categoria
15 ORDER BY
16     d.DateID ASC;
17

```

Esta consulta es importante para WWI porque les brinda una visión detallada de cómo se desempeñan sus diferentes categorías de productos en términos de ventas diarias. Primero elegimos la fecha de la orden y la categoría del producto, luego calculamos las ventas totales para cada combinación de esta fecha y categoría. Para lograr esto, unimos las tablas de órdenes, los productos en stock, las fechas y los tipos de paquete para asegurarnos de que estén relacionadas correctamente utilizando identificadores únicos. Luego agrupamos los resultados por fecha y categoría para obtener la cantidad total de ventas para cada categoría en cada día. Finalmente, organizamos los resultados por fecha para dar una idea de cómo las ventas cambian con el tiempo para cada categoría de producto.

Se muestra la correcta ejecución de la consulta:

```

1 SELECT
2     d.DateID,
3     pt.PackageTypeName AS Categoria,
4     SUM(ol.Quantity * ol.UnitPrice) AS Ventas
5 FROM
6     landing_zone_lab3.OrderLines ol
7 JOIN
8     landing_zone_lab3.StockItems si ON ol.StockItemID = si.StockItemID
9 JOIN
10    landing_zone_lab3.Dates d ON ol.DateID = d.DateID
11 JOIN
12    landing_zone_lab3.PackageTypes nt ON ol.PackageTypeID = nt.PackageTypeID

```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO		RESULTADOS	GRÁFICO	JSON	DETALLES DE LA EJ
Fila	DateID	Categoria	Ventas		
1	20130101	Each	119921.05		
2	20130101	Packet	2352.0		
3	20130101	Pair	1560.0		
4	20130102	Each	99778.6		
5	20130102	Packet	2384.0		
6	20130102	Pair	420.0		
7	20130103	Each	128734.0000000...		
8	20130103	Packet	2400.0		

Consulta 2: Total de ordenes por día

```


1 SELECT
2     d.DateID,
3     COUNT(DISTINCT ol.OrderID) AS TotalOrdenes,
4     SUM(ol.Quantity) AS TotalProductos,
5     COUNT(DISTINCT ol.PackageTypeID) AS TiposEmpaque
6 FROM
7     landing_zone_lab3.OrderLines ol
8 JOIN
9     landing_zone_lab3.StockItems si ON ol.StockItemID = si.StockItemID
10 JOIN
11    landing_zone_lab3.Dates d ON ol.DateID = d.DateID
12 GROUP BY
13     d.DateID
14 ORDER BY
15     d.DateID ASC;
16






```

Al proporcionar el número total de órdenes únicas, el número total de productos vendidos y el número total de tipos de empaque únicos agregados por día, la consulta ajustada proporciona a WWI una imagen completa de su actividad diaria. Primero, calcula el número total de órdenes únicas registradas por día. Luego, calcula la cantidad total de productos vendidos cada día, lo que ayuda a comprender la cantidad de productos que el negocio movió a través en un período de tiempo determinado. Además, determina la cantidad de tipos de empaque únicos utilizados cada día, lo que permite a WWI comprender la variedad de

productos empaquetados y las preferencias de empaque de los clientes. A través de la combinación de estos datos, WWI puede evaluar la demanda, administrar el inventario y modificar las estrategias comerciales para aumentar la satisfacción del cliente y optimizar el rendimiento operativo.

Se muestra la correcta ejecución de la consulta:


Consulta sin título

 **EJECUTAR**
 **GUARDAR**
 **DESCARGAR**
 **COMPARTIR**
 **PROCESAR**

```

1 SELECT
2   d.DateID,
3   COUNT(DISTINCT ol.OrderID) AS TotalOrdenes,
4   SUM(ol.Quantity) AS TotalProductos,
5   COUNT(DISTINCT ol.PackageTypeID) AS TiposEmpaque
6 FROM
7   landing_zone_lab3.OrderLines ol

```

Resultados de la consulta

INFORMACIÓN DEL TRABAJO		RESULTADOS	GRÁFICO	JSON	DETALLES DE LA EJECUCIÓN	GF
Fila	DateID	TotalOrdenes	TotalProductos	TiposEmpaque		
1	20130101	79	5059	3		
2	20130102	74	6240	3		
3	20130103	83	7518	3		
4	20130104	42	5627	3		
5	20130105	50	5137	3		
6	20130107	98	11867	3		
7	20130108	32	5187	3		
8	20130109	55	7749	3		
9	20130110	79	9915	3		
10	20130111	77	11408	3		
11	20130112	28	3096	3		
12	20130114	70	8494	3		
13	20130115	99	12284	3		
14	20130116	36	4881	3		
15	20130117	56	6161	3		

Consulta 3: Valor Bruto de la Mercancía (GMV) por día

```
1 SELECT
2     d.DateID,
3     SUM(ol.Quantity * ol.UnitPrice) AS GMV
4 FROM
5     landing_zone_lab3.OrderLines ol
6 JOIN
7     landing_zone_lab3.Dates d ON ol.DateID = d.DateID
8 GROUP BY
9     d.DateID
10 ORDER BY
11     d.DateID ASC;
12
```

Para calcular el Valor Bruto de la Mercancía (GMV) diario, la consulta selecciona datos importantes de la base de datos de WWI. Primero, se utilizan los identificadores de fecha correspondientes para conectar la tabla de líneas de órdenes con la tabla de fechas. Luego se suma el producto de la cantidad y el precio unitario de cada artículo en cada orden para cada día para calcular el GMV. Posteriormente, los resultados se agrupan por fecha para obtener el GMV total de cada día. Por último, los resultados se ordenan de forma ascendente por fecha, lo que proporciona una representación cronológica clara del GMV diario a lo largo del tiempo. WWI necesita esta información para evaluar el rendimiento financiero diario de su empresa y tomar decisiones estratégicas informadas para optimizar sus operaciones.

Se muestra la correcta ejecución de la consulta:

```
1 SELECT
2     d.DateID,
3     SUM(ol.Quantity * ol.UnitPrice) AS GMV
4 FROM
5     landing_zone_lab3.OrderLines ol
6 JOIN
7     landing_zone_lab3.Dates d ON ol.DateID = d.DateID
8 GROUP BY
9     d.DateID
10 ORDER BY
11     d.DateID ASC;
```

Resultados de la consulta		
INFORMACIÓN DEL TRABAJO		RESULTADOS
Fila	DateID	GMV
1	20130101	123833.05
2	20130102	102582.6
3	20130103	131674.0
4	20130104	98701.75000000...
5	20130105	86426.15000000...
6	20130107	221107.15000000...
7	20130108	93731.29999999...
8	20130109	144587.59999999...
9	20130110	153972.10000000...
10	20130111	218000.00000000...
11	20130112	92807.0
12	20130114	175832.5
13	20130115	276660.05000000...

Respuestas a las preguntas

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

Los dos métodos diferentes para administrar y analizar los datos de una empresa son arquitecturas para almacenar los datos. Los datos de un Data Warehouse se organizan y estructuran previamente en un formato predeterminado, lo que facilita su análisis y consulta. Este método funciona mejor con datos estructurados y bien definidos, como los de las bases de datos relacionales convencionales. Por el contrario, un Data Lakehouse combina características de un Lago de Datos y un Almacén de Datos, lo que permite almacenar datos estructurados y no estructurados en su forma original sin necesidad de transformarlos. Esto aumenta la flexibilidad y la capacidad de análisis al manejar una amplia gama de datos, incluidos los que no encajan fácilmente en un esquema predefinido (Koo, 2021).

Para WWI que busca aumentar su infraestructura existente con la capacidad de procesar fuentes de datos no estructurados y realizar análisis en tiempo real de datos estructurados y no estructurados simultáneamente, se recomienda adoptar una arquitectura Data Lakehouse. Este enfoque permite a WWI almacenar datos en su forma original, lo que facilita la integración de fuentes de datos no estructurados, como datos de redes sociales, registros de servidores, imágenes y videos. Además, Data Lakehouse ofrece capacidades de análisis avanzadas, incluida la capacidad de realizar análisis en tiempo real, lo que permite a WWI tomar decisiones más rápidas basadas en datos en respuesta a cambios en el mercado o en el comportamiento de los clientes (Koo, 2021). Esta arquitectura unificada proporciona a WWI la flexibilidad y escalabilidad necesarias para afrontar los desafíos de análisis de datos actuales y futuros.

- 2) ¿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?

Al realizar ETL (Extract, Transform, Load) utilizando herramientas de procesamiento de datos específicas como Google Cloud DataPrep, BigQuery y otras, puede beneficiarse de su facilidad de uso y escalabilidad. Estas herramientas generalmente proporcionan interfaces de usuario intuitivas que simplifican la creación de canalizaciones ETL, lo que reduce la necesidad de escribir código a mano y acelera el desarrollo. Además, al estar basados en la nube, ofrecen una escalabilidad casi ilimitada, lo que permite el procesamiento eficiente de grandes volúmenes de datos. La integración nativa con otros servicios en la nube facilita la implementación de soluciones completas que simplifican la gestión y el mantenimiento de la infraestructura. Sin embargo, estos beneficios pueden verse compensados por los costos asociados, ya que el uso intensivo de los recursos de la nube puede generar costos significativos, especialmente cuando aumentan el volumen de datos y la complejidad de los canales de ETL. Además, estas herramientas pueden tener limitaciones de personalización y la dependencia de un proveedor puede dificultar la portabilidad de soluciones y la migración

entre diferentes plataformas en el futuro (*Principales Ventajas de Utilizar la Herramienta ETL de Informatica*, s. f.).

En comparación, el desarrollo ETL con Python, Pandas y otras herramientas ofrece más flexibilidad y potencia para programar y transformar datos. Esto le permite implementar una lógica de transformación altamente personalizada adaptada a necesidades comerciales específicas. Hay muchos recursos y comunidades de soporte disponibles porque son lenguajes de programación ampliamente utilizados. Sin embargo, esta flexibilidad conlleva una curva de aprendizaje más pronunciada, ya que requiere un conocimiento más profundo de programación y análisis de datos. Además, puede ser necesario gestionar manualmente la escalabilidad y eficiencia de los procesos ETL, lo que puede requerir más esfuerzo y experiencia técnica por parte del equipo de desarrollo. Es decir, el desarrollo en Python y Pandas puede requerir un conocimiento más profundo de programación, lo que limita la accesibilidad a un grupo más especializado. Además de la facilidad de uso, simplifica el diseño de procesos ETL al proporcionar herramientas específicas orientadas a esta tarea. Esto evita la necesidad de escribir código extenso y permite una implementación más rápida y eficiente en comparación con la creación manual utilizando Python (Khan, 2024).

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

El modelo multidimensional construido en este laboratorio representa un modelo de tipo estrella. Esto lo demuestra la tabla de hechos central (OrderLines) en el centro del modelo, que está directamente relacionada con varias tablas de dimensiones (Dates, StockItems y PackageTypes) a través de claves externas. En un gráfico de estrellas, una tabla de hechos contiene dimensiones numéricas y está vinculada a varias tablas de dimensiones, cada una de las cuales representa una dimensión diferente del negocio. Las tablas de dimensiones contienen atributos descriptivos que proporcionan contexto para el tamaño de la tabla de datos. En este caso, las dimensiones incluyen información como fechas, datos de almacenamiento y paquetes. Además, la estructura de relaciones del modelo es simple y directa, y las tablas de dimensiones están conectadas directamente a la tabla de hechos específica del esquema en estrella (Navarro, 2024). Este diseño permite consultas analíticas rápidas y eficientes, lo que lo hace adecuado para inteligencia empresarial y análisis de datos. Por tanto, el modelo multidimensional construido en este laboratorio corresponde a un modelo de tipo estrella en términos de disposición y estructura.

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

En este modelo multidimensional, la tabla de hechos central es la tabla "OrderLines", que almacena las medidas relacionadas con los pedidos de ventas, como el número de productos vendidos, el precio unitario y las ventas totales. Esta tabla es la parte central del modelo y contiene información detallada de cada transacción realizada por la empresa. Además, las tablas de dimensiones ("Dates", "StockItems" y "PackageTypes") proporcionan contexto adicional para las dimensiones de la tabla de datos. Por ejemplo, la dimensión "Dates" proporciona información temporal, como la fecha del pedido, mientras que las dimensiones

"StockItems" y "PackageTypes" proporcionan información sobre los productos y tipos de paquetes asociados con cada pedido. Juntas, estas dimensiones enriquecen la tabla de datos al agregar información descriptiva que permite un análisis más completo y detallado de las transacciones comerciales. Por lo tanto, en este modelo multidimensional, la tabla "OrderLines" actúa como una tabla de hechos y las tablas "Dates", "StockItems" y "PackageTypes" actúan como tablas de dimensiones.

- 5) 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?

Cuando la dimensión "StockItemDim" cambia su tratamiento histórico al tipo 2 (Slowly Changing Dimension), se requieren ajustes significativos en varias áreas para garantizar que los datos se carguen correctamente y mantener la integridad del modelo multidimensional. Primero, necesitamos agregar más columnas en la dimensión "StockItemDim" para rastrear el historial de cambios de precio y tamaño del producto. Estas nuevas columnas contendrían la versión o el identificador de rendimiento de cada producto y las fechas de inicio y finalización de la versión. Además, se debe implementar una lógica especial para manejar SCD tipo 2, lo que significa que cuando se introduce una nueva versión del producto, el registro existente se marca como histórico y se crea un nuevo registro con valores actualizados y fechas correspondientes. En la tabla de datos "OrderLines", las claves externas que hacen referencia a la dimensión "StockItemDim" se deben verificar y actualizar para reflejar los cambios en la dimensión y garantizar que las nuevas versiones del producto estén asociadas correctamente con los eventos. Con respecto al proceso ETL, es importante agregar una lógica específica del SCD Tipo 2, que incluye la detección del tamaño y los precios del producto y la actualización de las dimensiones durante la carga de datos. Además, las transformaciones ETL deben ajustarse para tener en cuenta las nuevas columnas del historial en la dimensión "StockItemDim" y para garantizar que los cambios se procesen y almacenen correctamente en la tabla de dimensiones final. En resumen, la transición al SCD Tipo 2 requiere una planificación y adaptación cuidadosas de la estructura de dimensiones, la tabla de datos y el proceso ETL para mantener la integridad y precisión de los datos del modelo multidimensional (*Dimensiones Lentamente Cambiantes* | Dataprix, s. f.).

- 6) ¿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 incorporar los nuevos datos informados por WWI en el proceso ETL creado en esta práctica de laboratorio, se deben realizar algunos ajustes para permitir la carga incremental de los datos. Este enfoque implica cargar datos nuevos o modificados solo después de la última ejecución del proceso ETL, en lugar de reprocesar todo el conjunto de datos desde cero. Primero, el proceso ETL debe implementar lógica para identificar y capturar registros

nuevos o actualizados de la fuente de datos. Esto se puede lograr comparando marcas de tiempo o usando un identificador de versión para cada registro (Schneider, 2023).

Por otro lado, se requerirían cambios en las transformaciones ETL para implementar correctamente la lógica de carga incremental. Esto requiere agregar operaciones que identifiquen y filtren solo registros nuevos o modificados, en lugar de procesar todo el conjunto de datos (Haider, 2024). Para las tablas de hechos y las dimensiones, las actualizaciones incrementales se deben manejar correctamente para evitar duplicados o inconsistencias en los datos cargados. Esto puede incluir la actualización de claves primarias y externas y el historial de versiones en dimensiones SCD de tipo 2 (*Dimensiones Lentamente Cambiantes* | Dataprix, s. f.).

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

Al comienzo de la práctica se obtuvo un error al intentar ejecutar los jobs, pues el data lake y el bucket se habían creado en regiones diferentes. Se pudo observar que esto es un problema en términos de latencia, costos y compatibilidad de servicios por las siguientes razones. En primer lugar, al estar estos dos en regiones diferentes se presenta latencia debido a la distancia geográfica pues esto implica un efecto en las operaciones de lectura y escritura. En cuanto a costos, en la transferencia de datos entre regiones, algunos proveedores de servicios en la nube pueden tener tarifas diferentes por región y al incurrir en regiones diferentes se presenta un costo diferente. Por último, las diferentes regiones pueden prestar diferentes servicios, lo que afecta en caso de requerir un servicio específico y que la región no lo soporte. Luego de entender todos aquellos factores que afectan, se configuraron ambos para la misma región y no se volvieron a presentar problemas.

Otro error que se presentó fue al realizar la modificación del flow en la que se debía importar y parametrizar PackageTypes. Lo que sucedió fue que el conjunto de datos se importó, al hacer esto se presentaba otro error de conflicto con las regiones. Teniendo en cuenta todo lo mencionado en el error anterior se procedió a verificar en qué parte específicamente se presentaban regiones diferentes. Se encontró que la configuración del data flow tiene un endpoint regional, y este estaba por defecto a una región diferente a la utilizada en el bucket y data lake, al modificarlo se arregló el error y se pudo ejecutar correctamente el job con esta nueva configuración.

Referencias

Koo, B. (2021, 8 enero). *Data Lake vs Data Warehouse*. Kaits Consulting.

<https://www.kaitsconsulting.com/data-lake-vs-data-warehouse-sabes-la-diferencia/#:~:text=Un%20data%20lake%20almacena%20datos,datos%20estructura dos%20y%20previamente%20procesados.>

Principales ventajas de utilizar la herramienta ETL de Informatica. (s. f.).

<https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/407134/principales-ventajas-de-utilizar-la-herramienta-etl-de-informatica>

Khan, F. (2024, 20 marzo). ETL Using Python: Pros vs. Cons | Astera. *Astera*.

<https://www.astera.com/es/type/blog/etl-using-python/>

Navarro, S. (2024, 18 abril). Modelos estrella y copo de nieve | KeepCoding Bootcamps.

KeepCoding Bootcamps. <https://keepcoding.io/blog/modelos-estrella-y-copo-de-nieve/#:~:text=Los%20modelos%20estrella%20y%20copo,bases%20de%20datos%20a%20comprenderlos.>

Dimensiones lentamente cambiantes | Dataprix. (s. f.). [https://www.dataprix.com/es/blog-](https://www.dataprix.com/es/blog-it/bernabeudario/dimensiones-lentamente-cambiantes)

[it/bernabeudario/dimensiones-lentamente-cambiantes](https://www.dataprix.com/es/blog-it/bernabeudario/dimensiones-lentamente-cambiantes)

Schneider, F. (2023, 18 diciembre). Procesos ETL: Qué son, fases, beneficios y herramientas.

OpenWebinars.net. <https://openwebinars.net/blog/procesos-etl-que-son-fases-beneficios-y-herramientas/>

Haider, K. (2024, 25 abril). What is ETL? Definition, Process, Best Practices, Use Cases.

Astera. <https://www.astera.com/es/type/blog/etl/>