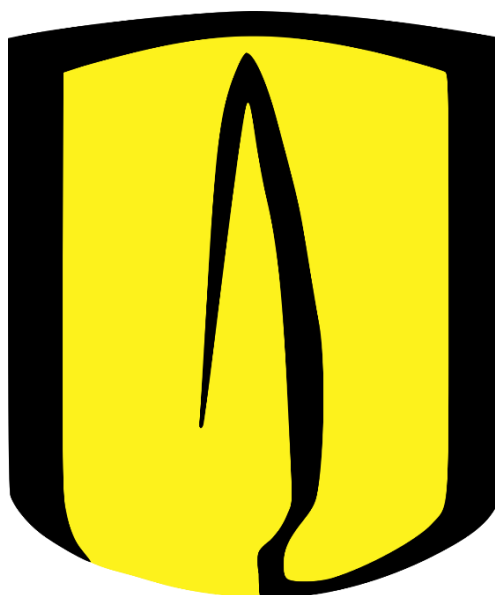


Informe de laboratorio # 3 Inteligencia de Negocios

Bogotá DC

Universidad de Los Andes



Integrantes:

Juan Andrés Bernal – ja.bernalg1@uniandes.edu.co - 202110848

Santiago Diaz - s.diazm1@uniandes.edu.co - 201912247

Juan Felipe Serrano – j.serrano@uniandes.edu.co - 201921654

1.

Inicialmente se realizan los pasos del tutorial, donde se crea todo lo necesario para la correcta ejecución del ETL.

Primero se crea el bucket en S3 donde se cargan todos los archivos:

The screenshot shows the Amazon S3 console interface for a bucket named 'uniandes-bi-lab3-source-g15'. The bucket is publicly accessible. The 'Objetos' (Objects) tab is selected, displaying a list of four CSV files. The table below summarizes the objects shown in the screenshot.

	Nombre	Tipo	Última modificación	Tamaño	Clase de almacenamiento
<input type="checkbox"/>	OrderLines.csv	csv	16 Nov 2023 9:49:19 AM -05	30.3 MB	Estándar
<input type="checkbox"/>	Orders.csv	csv	16 Nov 2023 9:49:23 AM -05	7.8 MB	Estándar
<input type="checkbox"/>	PackageTypes.csv	csv	16 Nov 2023 9:49:24 AM -05	980.0 B	Estándar
<input type="checkbox"/>	StockItems.csv	csv	16 Nov 2023 9:49:24 AM -05	64.6 KB	Estándar

Se decide trabajar de forma pública para evitar problemas con los permisos

Posteriormente se crea tanto la conexión como la base de datos en RedShift.

The screenshot shows the Amazon RedShift console interface for a cluster named 'uniandes-bi-lab3-destination-g15'. The cluster is in an 'Available' state. The 'Información general acerca del clúster' (General cluster information) section is visible, showing the cluster name and its state. The 'Métricas del clúster' (Cluster metrics) section is also visible, showing various performance metrics.

Cluster en RedShift

Teniendo en cuenta esto, nos dirigimos a AWS GLUE, donde crearemos una prueba de conexión y cargaremos el ETL.

uniandes-bi-lab3-connection

Connection details

Connector type	JDBC	Connection URL	jdbc:redshift://uniandes-bi-lab3-destination-g15.c1wjo8ohlog.us-east-1.redshift.amazonaws.com:5439/dev
Driver class name	-	Driver path	-
Username	awsuser	Require SSL connection	false
Subnet	subnet-0cad5c9fbfbd7c58	Security groups	sg-039dd47ef191f2283
Description	-	Created on	2023-11-16 10:01:30.320000
Last modified	2023-11-16 10:01:30.320000	Class name	-

Your jobs (1)

Job name	Type	Last modified	AWS Glue version
uniandes-bi-lab3-processor-final-g15	Glue ETL	11/18/2023, 11:50:21 AM	4.0

Creación de la conexión.

uniandes-bi-lab3-processor

Last modified on 11/18/2023, 12:00:12 PM Try new UI Actions Save Run

Visual Script Job details Runs Data quality Schedules Version Control

Source Action Target Undo Redo Remove

Data source properties - S3

Name: Extract StockItems

S3 source type: S3 location

S3 URL: s3://uniandes-bi-lab3-source-g15/StockItems.csv

Data format: CSV

Delimiter: Semicolon (;)

Escape character - optional: Enter a character to use for escaping

Quote character: Double quote (")

First line of source file contains column headers

Choose the button below to infer or reinfer the schema of your data in S3.

Infer schema

ETL cargado exitosamente

Funcionamiento correcto del run del mismo ETL:

uniandes-bi-lab3-processor

Last modified on 11/18/2023, 12:00:12 PM Try new UI Actions Save Run

Visual Script Job details Runs Data quality Schedules Version Control

Job runs (1/1) info Last updated (UTC) November 18, 2023 at 17:12:45 View details Stop job run Table View Card View

Filter job runs by property

Run status	Retries	Start time (UTC)	End time (UTC)	Duration	Capacity (DPUs)	Worker type	Glue version
Succeeded	0	2023/11/18 17:00:30	2023/11/18 17:03:38	2 m 41 s	10 DPUs	G.1X	4.0

2023/11/18 17:00:30

Job name: uniandes-bi-lab3-processor Id: jr_69f623a15c5ed436cab4b73bfe033036c0c6d6c3ad93b5bd6bca8b8cc452b58b Run status: Succeeded Glue version: 4.0

Retry attempt number: Initial run Start time (UTC): 18 de noviembre de 2023 17:00:30 End time (UTC): 18 de noviembre de 2023 17:03:38 Start-up time: 27 seconds

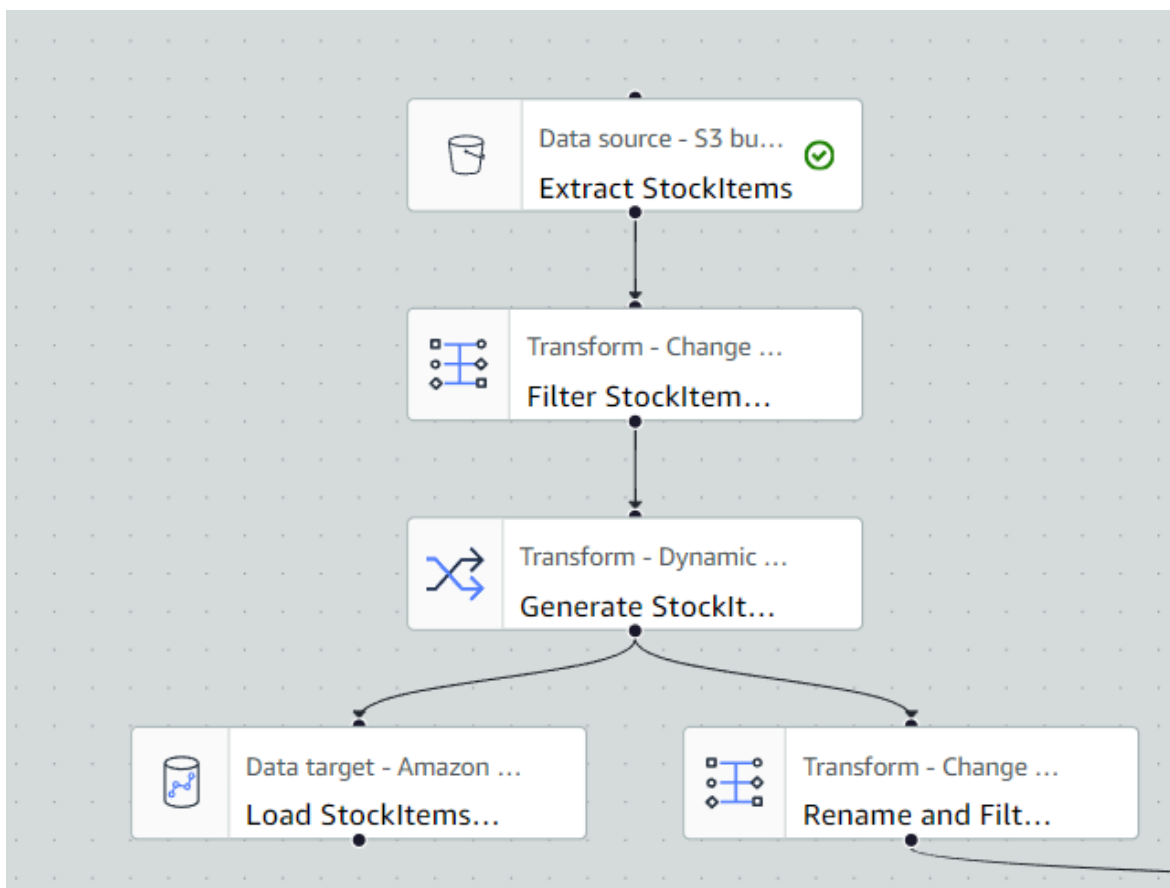
Execution time: 2 minutes 41 seconds Last modified on (UTC): 18 de noviembre de 2023 17:03:38 Trigger name: - Security configuration: -

Timeout: 2880 minutes Max capacity: 10 DPUs Number of workers: 10 Worker type: G.1X

Execution class: Standard Log group name: /aws-glue/jobs Cloudwatch logs: All logs, Output logs, Error logs Performance and debugging recommendations: View in CloudWatch

2.1 Análisis de los subprocessos del TL dado:

Primer subprocesso asociado a *StockItems* :



En este subproceso, se cargan los datos de nuestro bucket en S3, con respecto al csv de StockItems, posteriormente, se requiere un cuadro de Transform, que básicamente realiza el filtrado de las características que nos interesan, que, en este caso, son el Id y nombre del Objeto:

Transform | Output schema | Data preview

Name: Filter StockItems Columns

Node parents: Choose which nodes will provide inputs for this one. Choose one or more parent node. Extract StockItems X SS - DataSource

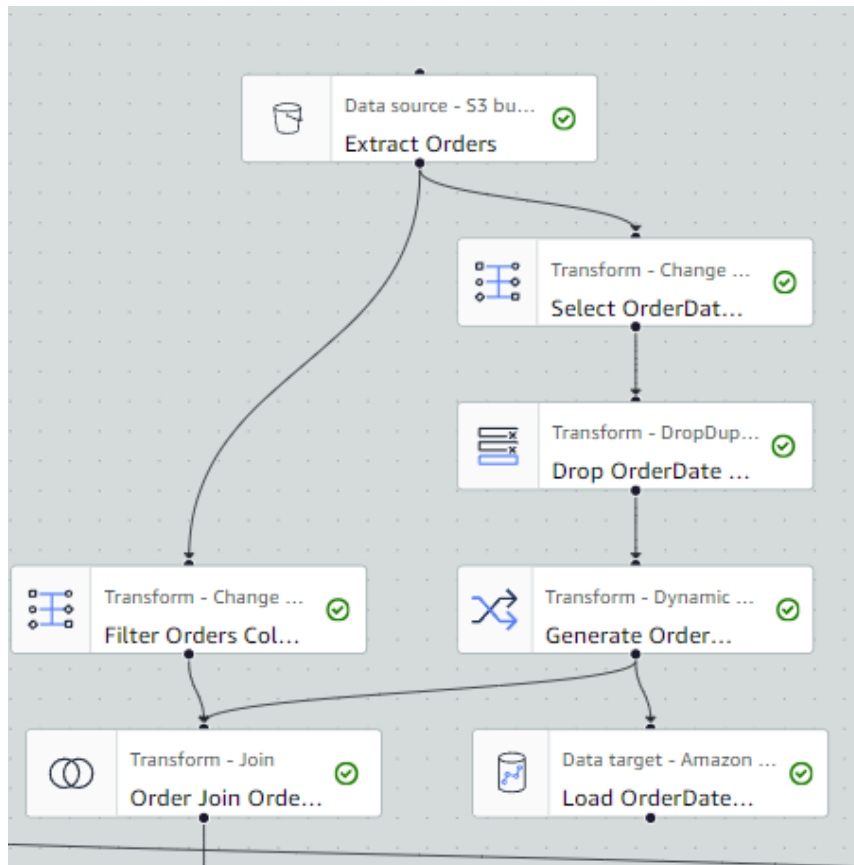
Change Schema (Apply mapping)

Source key	Target key	Data type	Drop
StockItemID	StockItemID	int	<input type="checkbox"/>
StockItemName	StockItemName	string	<input type="checkbox"/>
SupplierID			<input checked="" type="checkbox"/>
ColorID			<input checked="" type="checkbox"/>
UnitPackageID			<input checked="" type="checkbox"/>
OuterPackageID			<input checked="" type="checkbox"/>
Brand			<input checked="" type="checkbox"/>

Filtrado de atributos

Y finalmente, esta tabla obtenida tras el filtrado es subida a RedShift, esta tabla será usada más Adelante para otro subproceso

Segundo subprocesso: Orders:



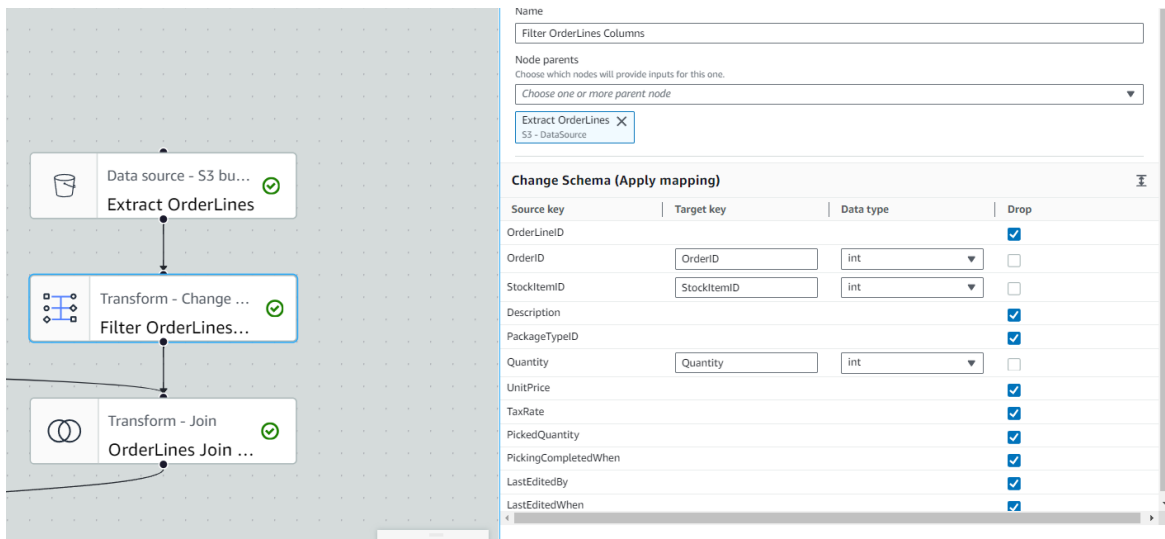
Se realiza un proceso similar al de los StockItems donde se filtra por medio de los atributos que nos interesan, en este caso, solamente los orderDate de las ordenes, Posteriormente, se aplica un noDuplicates a la columna, obteniendo las fechas únicas:

La imagen muestra la interfaz de usuario de 'uniandes-bi-lab3-processor'. En la parte superior, hay una barra de navegación con pestañas como 'Visual', 'Script', 'Job details', 'Runs', 'Data quality', 'Schedules' y 'Version Control'. Debajo de esto, hay una barra de herramientas con iconos para acciones como 'Source', 'Action', 'Target', 'Undo', 'Redo', 'Remove', etc. El área principal de la izquierda muestra un diagrama de flujo similar al anterior, pero con menos transformadores visibles. A la derecha, se encuentra un panel 'Data preview' que muestra una lista de fechas: 'orderdate', '2013-01-01', '2013-01-02' y '2013-01-03'. El panel también incluye botones como 'Filter sample dataset', 'End session' y 'Previewing 1 of 1 fields'.

Obtención de fechas únicas.

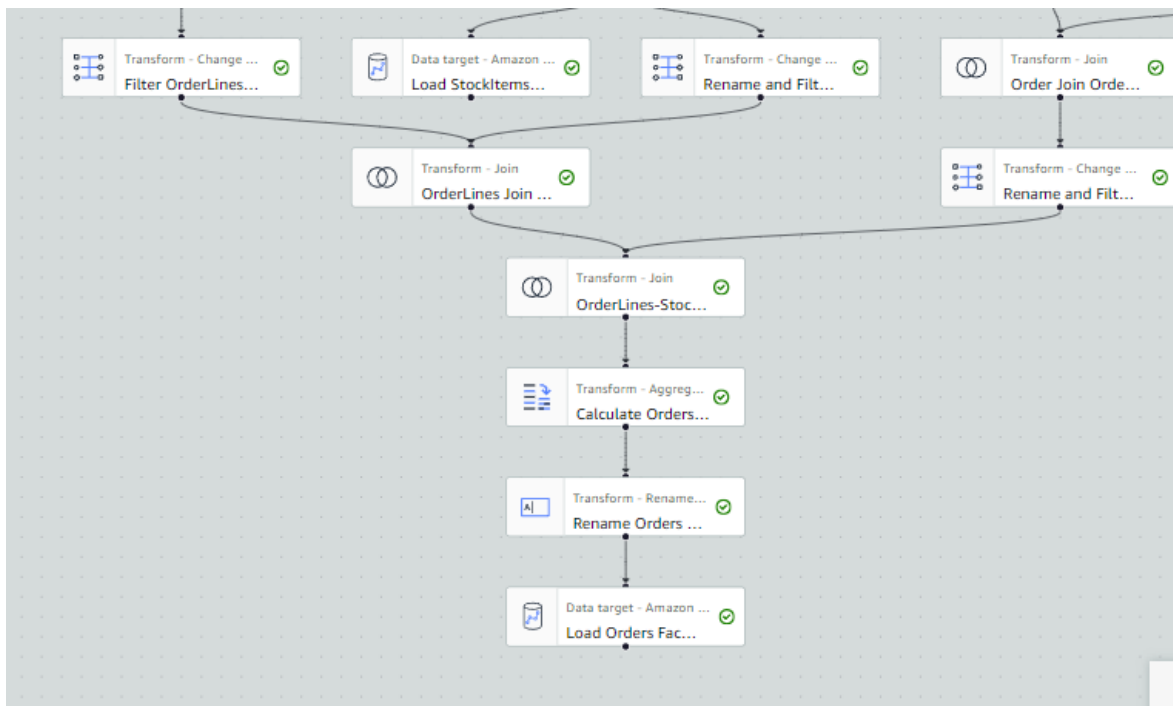
Finalmente, se asigna un ID aleatorio, con el fin de identificar cada fecha y cargarla al ETL.

Tercer subprocesso:



Se cargan las orderLines del respectivo CSV

Con base a esto, se implementa un Join con la columna del id de los Stock Items, con el id del Stock Item de los Order lines. Teniendo en cuenta esto, se realizar otro Join con base a las órdenes de las fechas, obtenidas. Con base a esto se genera una columna de agregación para sumar la cantidad total de cantidades de producto por orden. Y se renombra la columna, esto se evidencia mejor en el caso mostrado en lo siguiente:

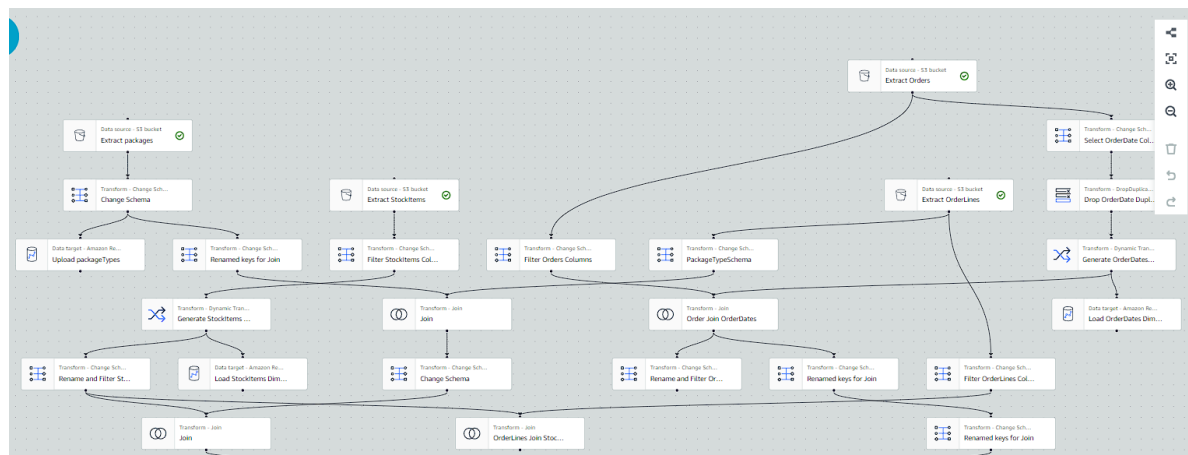


Con base a esto, se genera la tabla de hechos inicial, la cual está compuesta por el ID del stockitem, el ID de la fecha correspondiente y la cantidad de producto que fue adquirido. Esta columna relaciona.

2.2 Para la realización de este requerimiento, se implementan los siguientes subprocesos:

1. Se crea una carga del archivoCSV packageTypes, el cual tendrá los datos de los tipos de paquetes.
2. Se crea una transformación a la carga, la cual solamente tendrá el Id del paquete y Nombre correspondiente.
3. Con base a ello, relacionamos las columnas PackageId de las tablas OrderLines y la dimensión de packageType que fue creada previamente.
4. Con base a ello, creamos una nueva relación con la tabla Order con base al orderID de la tabla creada y la tabla órdenes.
5. Con base a ello, ya tenemos nuestra tabla de hechos creada y empezamos a realizar funciones de agregación y de filtrado de las columnas para mostrar en la tabla de hechos lo requerido al negocio. Así, que agrupamos con base al ID del stockItem, el Id de la OrderDate, la cantidad y precio de cada uno de los registros de la tabla de hechos.
6. Adicionalmente, agregamos la columna TotalPrice la cual se calcula entre la cantidad multiplicada por el precio unitario de cada StockItem en el inventario. Con base a ello se obtiene el siguiente resultado:

Modelo ETL extendido de bloques:



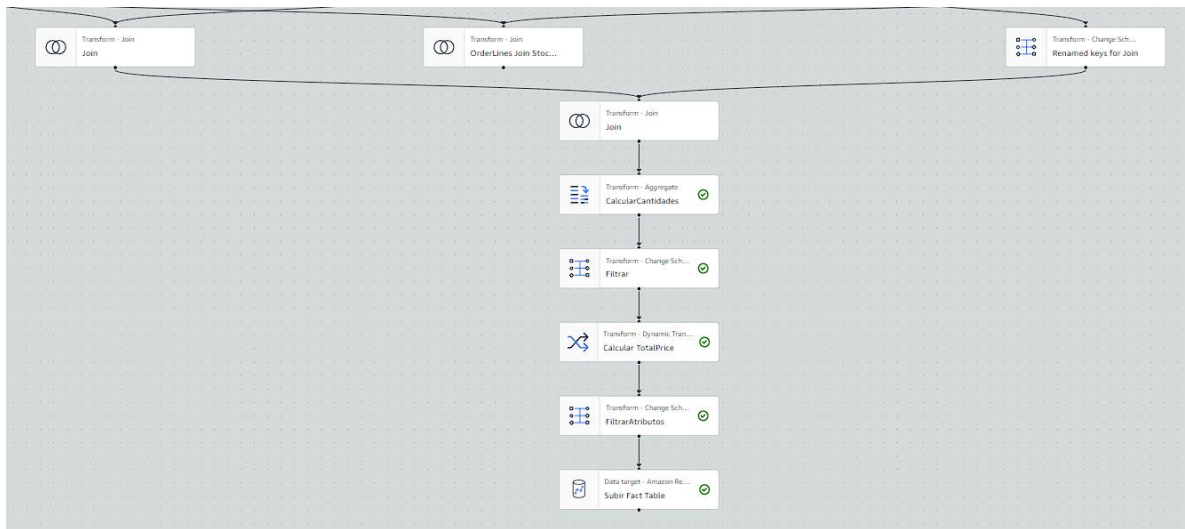


Tabla de hechos en RedShift:

```
1 SELECT * FROM "dev"."public"."facttableorders";
```

stockitemuuid	orderdateuuid	packagetypeid	totalprice
fa41a033-f45b-463f-8fea...	765f2634-fb12-46cc-bb48...	7	350
4580eb0e-bef2-4541-869...	bff7c9d1-d390-4723-8f19...	7	225
3b6514c0-f158-4abe-a6d...	f2c9bf34-af16-45dd-9298...	7	432
49d1917a-a84d-4fa8-921...	5ef0c7a3-87d3-4806-87fd...	7	720
c1c2bbc4-0961-4521-9e7...	8860607c-5a27-441b-884...	7	66
5bfac59a-287c-4981-947...	efd164ee-31b6-4f2e-a721...	7	1425
89abceca-346e-45a1-a44...	2eb599a2-0566-43e3-b59...	7	50
9ef2a8f3-a622-4877-9909...	0138024d-ef81-4954-8bd...	7	91
d041cd1a-3474-432c-907...	ed0d3bda-d178-4b37-8d1...	7	104
6142dd70-3c8e-455e-94d...	f1a1f358-847f-4b9c-9563...	7	2160
e0470106-d6d6-4a2d-9c0...	c655ae96-a917-4ad1-96c...	7	64
0bf960be-7e90-4357-829...	efb0a4d3-44f1-47ac-9cea...	7	32
7512a72f-03f0-4328-990a...	1e81dfd2-ab37-41af-b5b4...	7	288
b412812a-038a-441e-a0c...	1dea2ffa-60b1-4c54-8038...	7	875
826acaa2-6925-422d-bd2...	283a4d33-ed54-459e-bb3...	7	78
be7b398d-1b0a-458d-b3f...	14bf859d-524f-477a-9c73...	9	720
9c931cce-17ff-456d-8211...	34bc445b-13d2-4128-8f2...	7	324
bc3bb7b8-3e16-4c0f-8fbb...	4bf93363-a737-41da-8cf1...	7	160
75c6c586-280d-4c9b-987...	504200ae-6798-4203-92a...	7	216
826acaa2-6925-422d-bd2...	2503e144-7975-4175-994...	7	78
db6a66ab-9e3a-44ae-ad9...	86025a80-014c-4123-a86...	7	130

Tabla de hechos de las ordenes

Cabe aclarar que esta tabla está sujeta las decisiones e intereses del negocio, es decir, de la granularidad que este quiera tener, en este caso y dado los requerimientos que se nos estaba pidiendo el total Price de cada una de las ordenes, se decide mostrar solamente el precio total de cada una de las ordenes, sin embargo, no hay problema con mostrar las 2 columnas del quantity y unitPrice, para tener una mayor granularidad.

Dimensión StockItems:

```
1 SELECT * FROM "dev"."public"."stockitemsdim";
```

stockitemid	stockitemname	stockitemuuid
1	USB missile launcher (Gr...	9b3a6df6-3122-4edc-9c6...
2	USB rocket launcher (Gray)	32b75179-a2ed-4968-857...
3	Office cube periscope (Bl...	df6c05b8-950e-42b1-8aa...
4	USB food flash drive - sus...	0bf960be-7e90-4357-829...
5	USB food flash drive - ha...	2c7f18c1-4caf-4b5d-8d17...
6	USB food flash drive - hot...	8fd15b7d-5c1e-45f4-a84d...
7	USB food flash drive - piz...	0c3215c4-a52e-4880-8da...
8	USB food flash drive - di...	be7b398d-1b0a-458d-b3f...
9	USB food flash drive - ba...	d21111ec-4b65-4736-aae...
10	USB food flash drive - ch...	c37fb644-e717-4cae-bfc8...
11	USB food flash drive - co...	bc3bb7b8-3e16-4c0f-8fbb...
12	USB food flash drive - donut	eb3ac554-674c-4225-ba7...
13	USB food flash drive - shr...	f5b33102-a2de-48f9-ba07...
14	USB food flash drive - fort...	964083b3-7201-42be-99c...
15	USB food flash drive - de...	816ec71c-6e25-4c02-b8ff...

Dimensión de los StockItems

Dimensión OrderDate:

```
1 SELECT * FROM "dev"."public"."orderdatesdim";
```

orderdateuid	orderdate
8bb9a9b8-11a5-452b-9e4...	2013-02-14
b0566d6b-afed-4ba6-855...	2013-03-20
347d97d5-dcf7-468c-b74...	2013-07-30
f7a9a5d4-a2c8-441e-868...	2013-10-11
0bda5bd3-3e75-4993-af4...	2013-11-14
e59914b8-42a0-4def-ab5...	2013-11-23
abaed7f3-2eda-4dce-883...	2013-12-10
a18b9d76-a2d5-4a2e-aa6...	2014-01-30
47907663-66bb-47cc-ad5...	2014-03-01
42871800-c17a-47b4-908...	2014-03-14
11334f4f-2719-4822-a4e3...	2014-06-23
ee46f609-1ee9-41ed-80e...	2014-06-30
50e73f92-e8f6-4cbe-88d6...	2014-08-20
ac382d37-43ca-4aee-8e3...	2014-09-30
591b5c50-f360-41cd-a8e...	2014-10-07
4bf9aa73-174a-4c06-b23...	2015-02-16

Dimensión de OrderDate

Dimensión PackageType:

```
1 SELECT * FROM "dev"."public"."packagetyperedim";
```

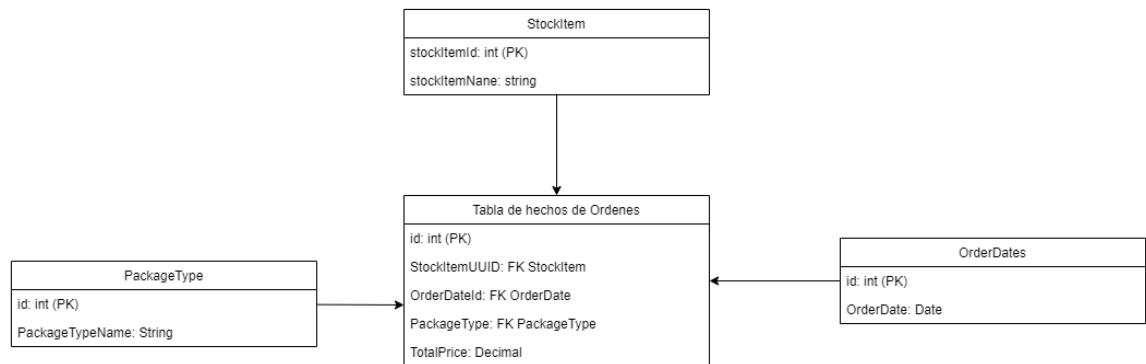
	packagetypeid	packagetypername
<input type="checkbox"/>	1	Bag
<input type="checkbox"/>	2	Block
<input type="checkbox"/>	3	Bottle
<input type="checkbox"/>	4	Box
<input type="checkbox"/>	5	Can
<input type="checkbox"/>	6	Carton
<input type="checkbox"/>	7	Each
<input type="checkbox"/>	8	Kg
<input type="checkbox"/>	9	Packet
<input type="checkbox"/>	10	Pair
<input type="checkbox"/>	11	Pallet
<input type="checkbox"/>	12	Tray
<input type="checkbox"/>	13	Tub
<input type="checkbox"/>	14	Tube

Dimensión de PackageType

Modelo de bloques creado en RedShift.

3.

3.1



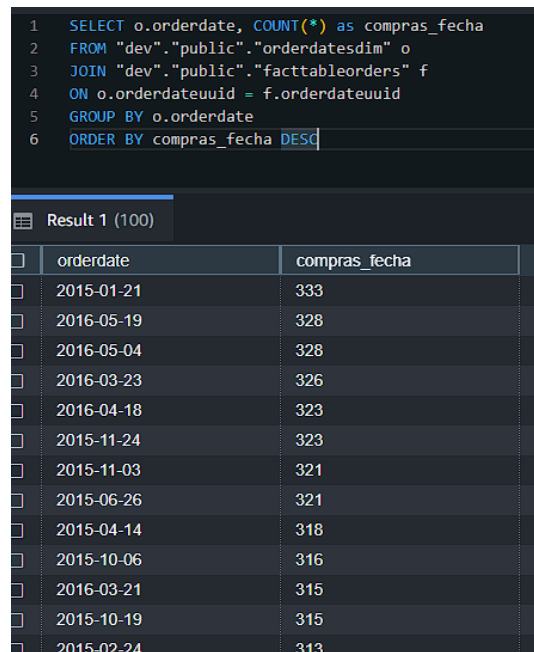
Esto se recrea teniendo en cuenta exclusivamente lo mostrado en las tablas de RedShift y la granularidad manejada, tanto las PK's de StockItem como de OrderDates son aquellas subrogadas, es decir aquella generada por el sistema ETL, es decir aquellas 'UUID'.

3.2

Con base al modelo creado en RedShift, se crean las siguientes consultas para ser ejecutadas en el entorno:

Fechas con mayor cantidad de compras realizadas:

```
SELECT o.orderdate, COUNT(*) as compras_fecha
FROM "dev"."public"."orderdatesdim" o
JOIN "dev"."public"."facttableorders" f
ON o.orderdateuuid = f.orderdateuuid
GROUP BY o.orderdate
ORDER BY compras_fecha DESC
```



The screenshot shows a SQL query being executed in a database tool. The query is the same as the one in the previous block. Below the query, the results are displayed in a table with two columns: 'orderdate' and 'compras_fecha'. The results are ordered by 'compras_fecha' in descending order. The first result shows '2015-01-21' with 333 purchases.

orderdate	compras_fecha
2015-01-21	333
2016-05-19	328
2016-05-04	328
2016-03-23	326
2016-04-18	323
2015-11-24	323
2015-11-03	321
2015-06-26	321
2015-04-14	318
2015-10-06	316
2016-03-21	315
2015-10-19	315
2015-02-24	313

Output de la consulta

Los días con mayor numero de recaudo, es decir, totalPrice:

```
SELECT o.orderdate, SUM(f.totalprice) as recaudo
FROM "dev"."public"."orderdatesdim" o
JOIN "dev"."public"."facttableorders" f
ON o.orderdateuuid = f.orderdateuuid
GROUP BY o.orderdate
ORDER BY recaudo DESC
```

1	SELECT o.orderdate, SUM(f.totalprice) as recaudo
2	FROM "dev"."public"."orderdatesdim" o
3	JOIN "dev"."public"."facttableorders" f
4	ON o.orderdateuuid = f.orderdateuuid
5	GROUP BY o.orderdate
6	ORDER BY recaudo DESC

Result 1 (100)	
orderdate	recaudo
2016-04-18	253145
2015-10-06	252898
2016-05-06	252897
2013-06-20	250782
2016-03-23	249985
2013-07-11	249218
2015-02-24	244684
2015-11-03	237927
2015-01-21	237399
2015-03-10	235655
2013-11-18	233913
2015-09-08	232742
2015-12-03	232481
2015-07-23	231764
2016-03-22	231683
2016-05-19	231148
2014-05-20	228876

Output de la consulta

Se puede evidenciar que los días donde más ventas hay, no necesariamente son los días donde hay más recaudo, a comparación de la consulta anterior.

3.3

¿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?

Las principales ventajas de usar herramientas especializadas de AWS para ETL, en comparación con el desarrollo en Python utilizando librerías de inteligencia de negocios se pueden agrupar en:

1. Integración fácil con otros servicios de AWS, escalabilidad automática, y una gestión de infraestructura simplificada. Debido a que están diseñadas para funcionar juntas y estar en la nube
2. Usabilidad. Servicios como AWS Glue ofrecen una interfaz fácil de usar. Además, la gestión de datos y seguridad se maneja por debajo lo cual facilita mucho su configuración

En contraste, las desventajas de usar herramientas de AWS para ETL frente a soluciones en Python con librerías de inteligencia de negocios se centran en la flexibilidad limitada y costos asociados a hostear servicios en la nube. Debido a la facilidad de configuración por AWS se puede dar que no se aprenda del todo a hacer los procesos por uno mismo y en cambio que se vuelva dependiente de AWS para estos proyectos. Por otro lado, el uso de Python y librerías especializadas permite un mayor control y personalización, pero requiere una gestión de infraestructura más intensiva, puede enfrentar desafíos de escalabilidad y mantenimiento, y podría no ser óptimo para volúmenes de datos muy grandes.

A partir del resto de información contenida en los archivos CSV proporcionados, ¿qué otros análisis consideran que se pueden soportar y cómo se traducirían a dimensiones y medidas sobre el modelo actual?

Análisis de Optimización de Precios: se debería expandir la tabla de hechos para incluir medidas de precio de venta y cantidad vendida, que permitirían evaluar cómo los cambios de precios afectan las ventas. También se podría añadir una dimensión de Producto/Artículo detallada utilizando la tabla StockItems. Una medida calculada de elasticidad de precio también podría ser interesante para este análisis, ya que indicaría la sensibilidad de la demanda frente a cambios en el precio.

Previsión de Ventas: Una dimensión de Tiempo bien estructurada podría utilizarse para analizar las ventas a lo largo de diferentes períodos. Se podrían utilizar datos históricos de ventas como medida en la tabla de hechos para identificar tendencias y patrones, como estacionalidad o crecimiento año tras año. Esto con el fin de desarrollar modelos predictivos y ayudar en la planificación futura del inventario y recursos.

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

Primero tuvimos unos problemas de conexión con AWS, además de que nos sacaba constantemente de las cuentas y después no nos dejaba hacer login. Además, durante el desarrollo del lab había un poco de diferencias entre las interfaces actuales de AWS y las mostradas en los tutoriales. También tuvimos dificultades con los nombres sugeridos en los tutoriales ya que para algunos componentes, como los buckets, estos ya estaban ocupados (no entendimos muy bien como esto era posible) lo que nos llevó a seleccionar nombres alternativos para evitar conflictos.

Particularmente el problema relacionado con AWS Glue fue el más complicado. Al cambiar los nombres de las transformaciones en Glue, nos topamos con un fallo del sistema completo. Lo investigamos un poco y concluimos que esto podría deberse a cómo AWS Glue gestiona las dependencias y referencias entre los componentes. Cualquier cambio en los nombres podría haber roto estas referencias, causando errores en la ejecución. Finalmente lo logramos resolver usando la opción de fix transformation que se ofrece en la descripción aunque nos demoramos un poco encontrándola.