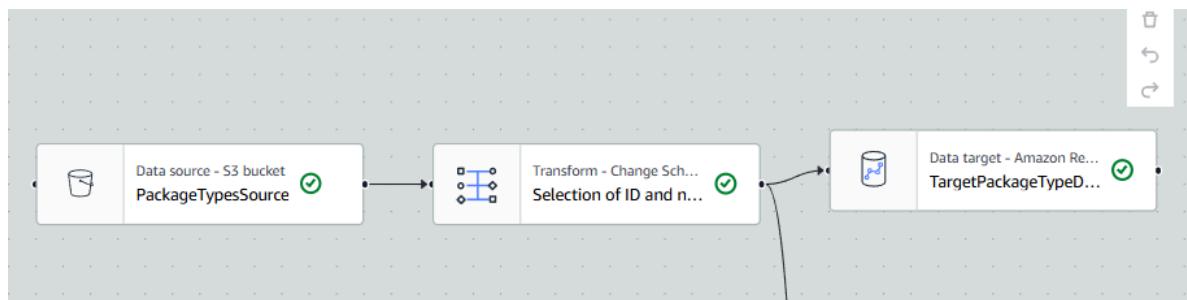


# Laboratorio 3

## 1. Descripción del proceso ETL realizado

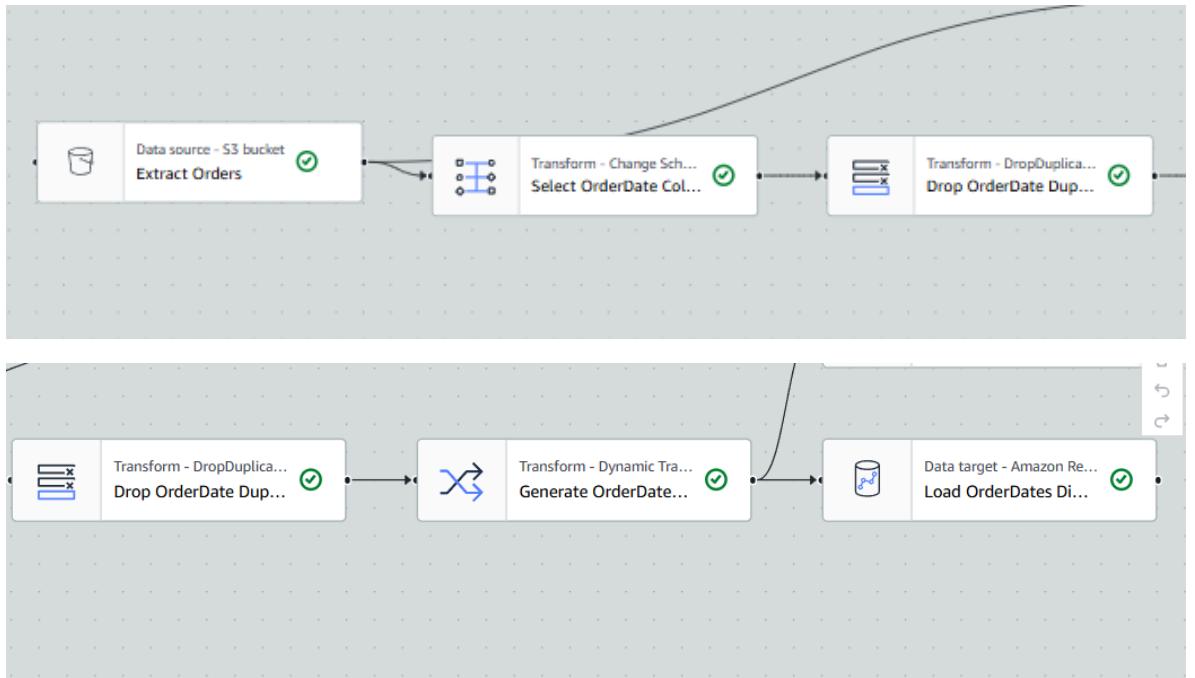
dim\_packagetype

- Extracción: Se toma la información desde el archivo data\_PackageTypes.csv, almacenado en el bucket de S3.
- Transformación:
  - Se seleccionan únicamente las columnas PackageTypeID y PackageTypeName.
  - Se genera un UUID único por fila para identificar el registro de forma universal.
- Carga: La dimensión es cargada en Redshift con los campos limpios y estructurados, para servir como referencia en la tabla de hechos (dim\_packagetype).



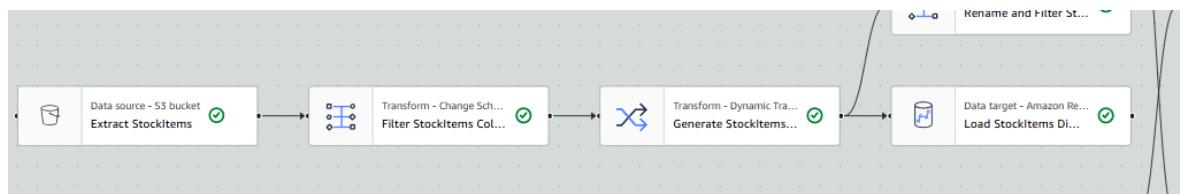
### ◆ dim\_orderdates

- Extracción: Se obtiene la columna OrderDate desde el archivo data\_Orders.csv.
- Transformación:
  - Se seleccionó únicamente la columna orderdate.
  - Se eliminaron fechas duplicadas para obtener valores únicos.
  - Se generó un UUID para cada fecha (orderdateuuid).
- Carga: Se cargan las fechas únicas en la dimensión dim\_orderdates, permitiendo agrupar y analizar métricas por fecha.



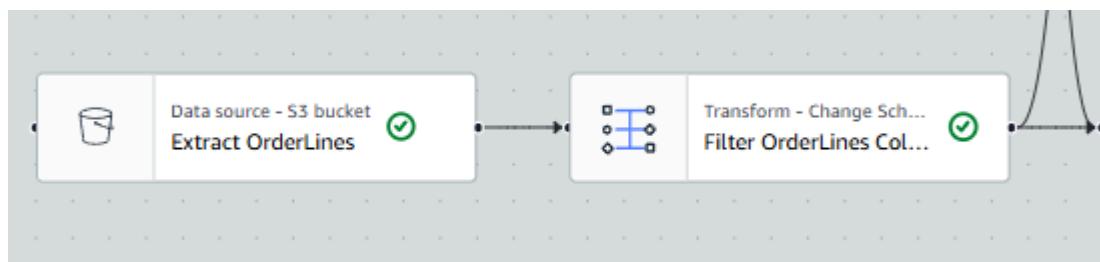
#### ◆ dim\_stockitems

- Extracción: Se toma la información del archivo data\_StockItems.csv.
- Transformación:
  - Se conservaron las columnas stockitemid, stockitemname, colorid, size y unitprice.
  - Se eliminaron campos como tags, photo, validfrom, entre otros que no aportaban valor.
  - Se generó un UUID único para cada producto (stockitemuuid).
- Carga: Se cargan los productos únicos en la dimensión stockitemsdim, que describe los ítems presentes en las órdenes.

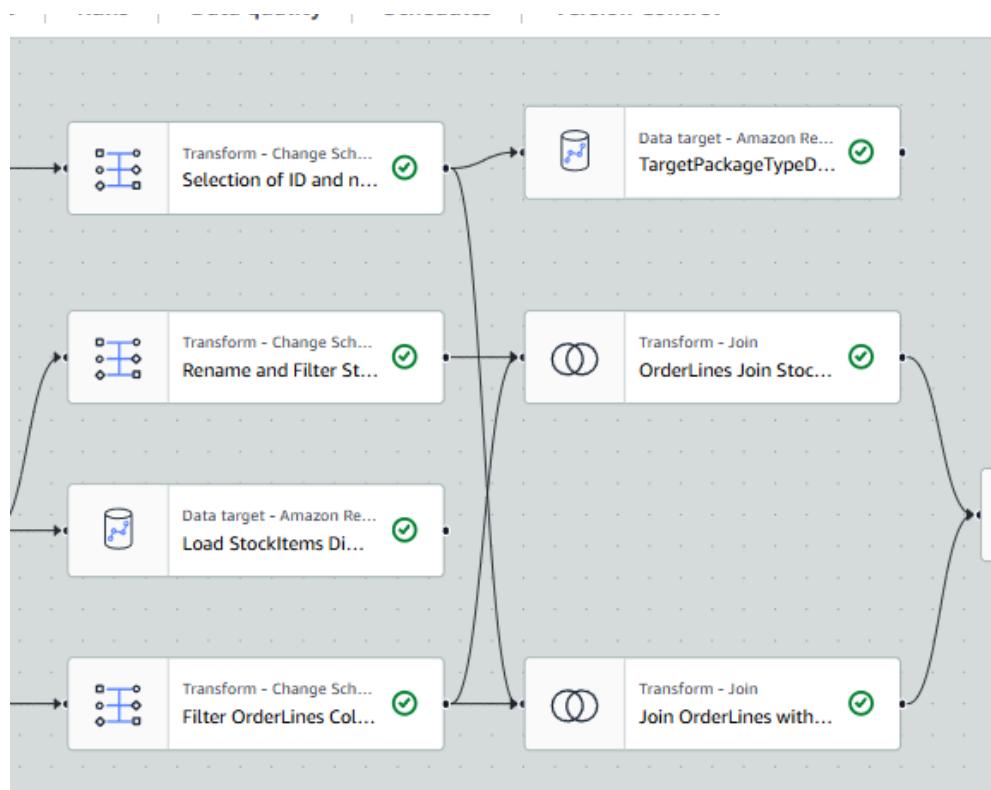


#### ◆ fact\_orders

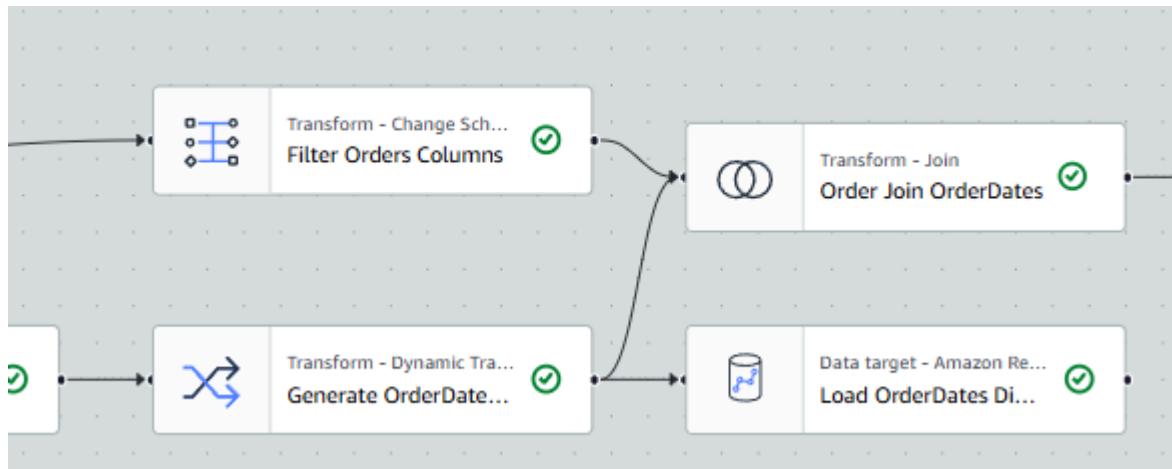
- Extracción: Se combinan datos de los archivos data\_Orders.csv y data\_OrderLines.csv.
- Transformación:
  - Se integraron los archivos data\_OrderLines.csv, data\_Orders.csv, data\_StockItems.csv y data\_PackageTypes.csv.
  - Se depuró OrderLines, eliminando orderlineid, description y pickedquantity, y conservando orderid, stockitemid, packagetypeid, quantity, unitprice y taxrate.



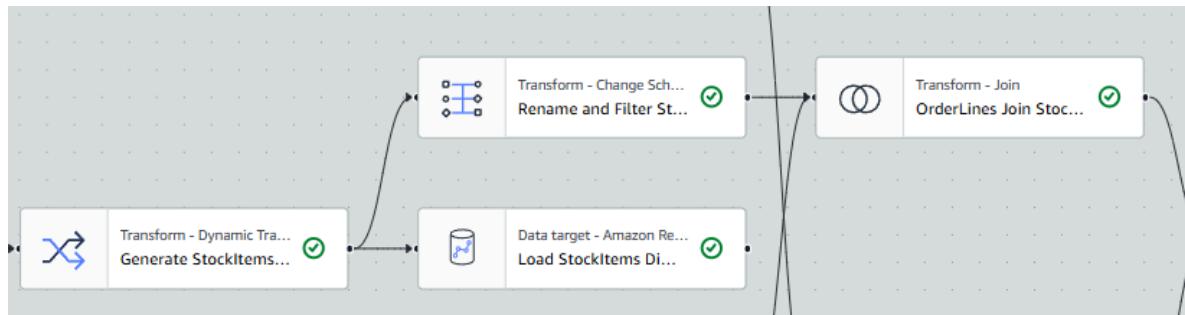
- Se realizó un join con los tipos de empaque utilizando la columna packagetypeid.



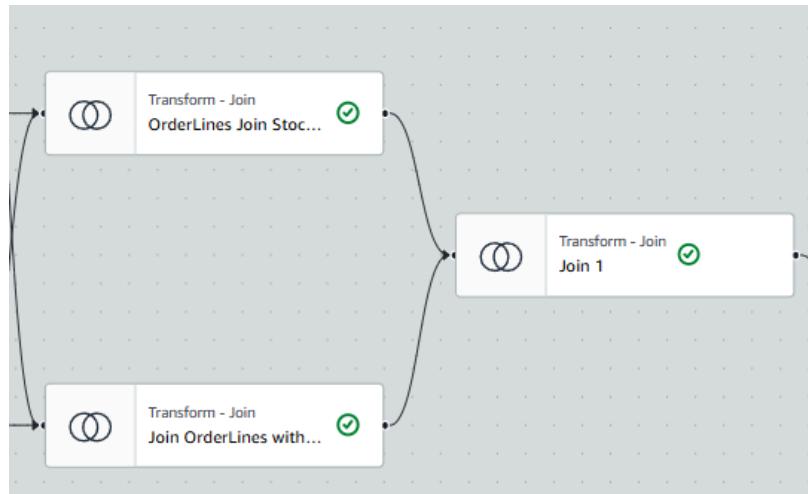
- Se filtró el archivo Orders para conservar solo orderid y orderdate.
- Se integró esta información con dim\_orderdates usando la fecha como llave orderdate



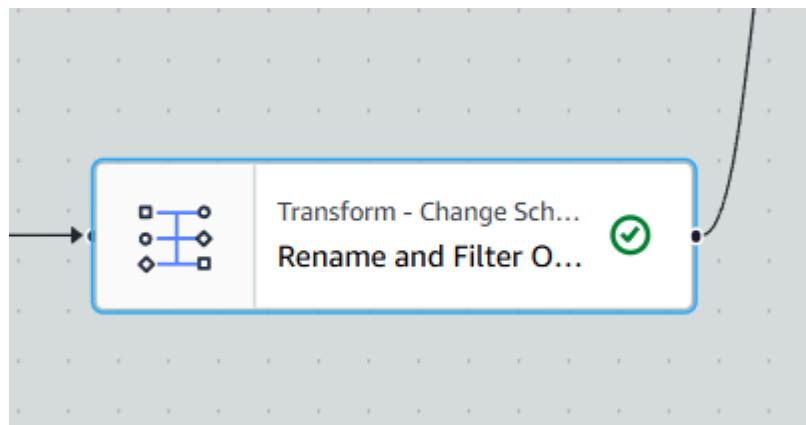
- Se renombró stockitemid a stockitemid\_r para evitar conflictos en el join.
- Se hizo una unión con dim\_stockitems para traer stockitemname, colorid, size y stockitemuid.



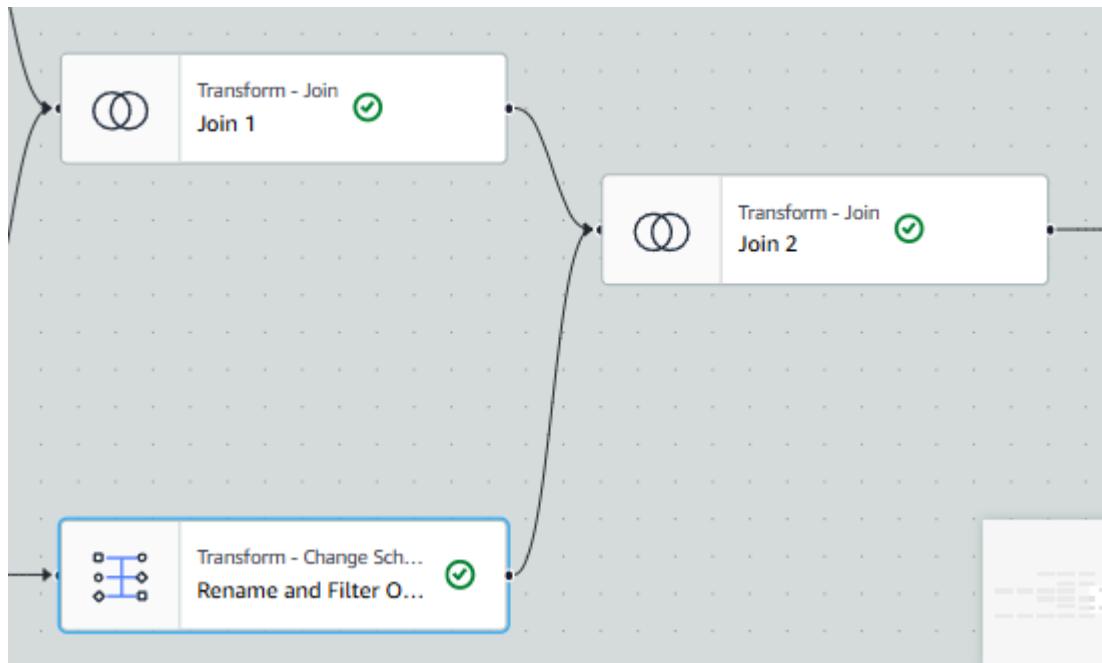
- Se unieron todos los conjuntos utilizando orderid y stockitemid.



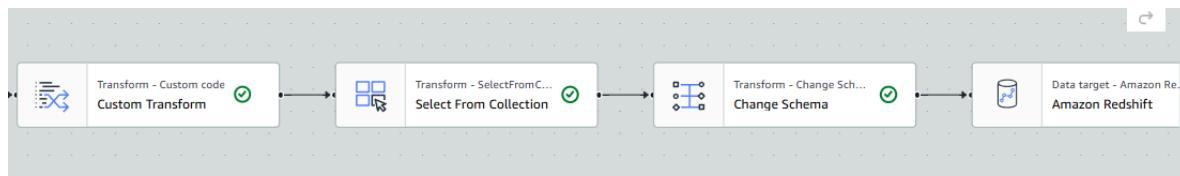
- Se renombraron las filas de orderdate por orderdate\_r



- Se realizó un join entre estas ultimas dos transformaciones pro la llave orderid



- Se creó una columna totalprice multiplicando unitprice por quantity.
- Se eliminaron columnas auxiliares como orderid, unitprice, stockitemid\_r y orderid\_r.



- Carga: Se cargan los datos integrados y transformados en la tabla de hechos ordersfact, base para análisis de ventas, productos, empaques y temporalidad.

Este proceso fue desarrollado, probado e implementado utilizando AWS Glue Studio, conectando orígenes en S3 con Redshift, utilizando transformaciones como Change Schema, Join, Filter y Apply Mapping, así como generadores de UUID para claves primarias y foráneas.

## 2. Consultas SQL

```

1 SELECT colorid, COUNT(*) AS cantidad_productos
2 FROM "dev"."public"."dim_stockitems"
3 GROUP BY colorid
4 ORDER BY cantidad_productos DESC;
5

```

```

1 SELECT
2   column_name
3   FROM
4     information_schema.columns
5   WHERE
6     table_schema = 'public'
7     AND table_name = 'fact_orders';
8

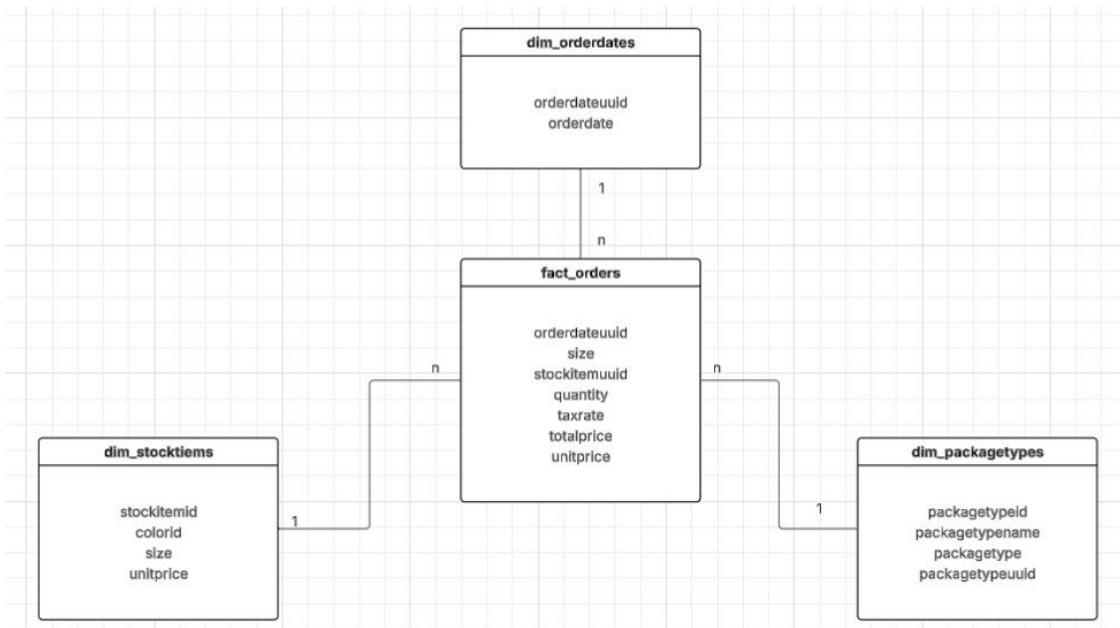
```

```

1 SELECT column_name
2   FROM information_schema.columns
3  WHERE table_name = 'dim_packagetypes';
4

```

### 3. Modelo Multidimensional:



### 4. Análisis y Conclusiones

#### a. Ventajas y desventajas de implementar un ETL con AWS Glue frente a Python/Pandas

Durante este laboratorio se utilizó AWS Glue como herramienta principal para el desarrollo de un proceso ETL. A continuación, se describen algunas de las ventajas y desventajas

observadas al comparar esta implementación con una basada en herramientas como Python, Pandas y demás componentes estudiados previamente en el curso.

Ventajas de AWS Glue:

- Interfaz visual e intuitiva.
- Escalabilidad con Apache Spark administrado.
- Integración con servicios AWS como Redshift y S3.
- Facilidad para orquestar procesos ETL programados.

Desventajas frente a Python/Pandas:

- Menor flexibilidad para transformaciones complejas.
- Curva de aprendizaje inicial.
- Demorada la ejecución
- Diagnóstico de errores menos directo.
- Costos asociados al uso de Glue.

**b. Análisis adicionales posibles y cómo se reflejan en el modelo dimensional**

Análisis propuesto	Dimensiones involucradas	Medidas sugeridas
Ventas por tipo de empaque	Dimensión PackageType	SUM(totalprice)
Comportamiento de ventas por fecha	Dimensión OrderDate	COUNT(orderid), SUM(totalprice)
Segmentación por color o tamaño del ítem	Dimensiones colorid, size	AVG(unitprice), SUM(quantity)
Comparación de productos por categoría	Dimensión StockItem	SUM(totalprice)
Identificación de pedidos con mayor volumen	Dimensión OrderID	SUM(quantity)

**c. Principales errores encontrados y soluciones aplicadas**

- Problema Custom Transform y DynamicFrameCollection

Al usar Custom Transform, el nodo devolvía una colección de DynamicFrames, lo que impedía su uso en pasos posteriores. Se utilizó un nodo SelectFromCollection para extraer el DynamicFrame correcto y continuar el flujo.

- Campo totalprice no disponible en Redshift

Aunque la columna fue generada correctamente, no se visualizaba en la tabla final. Se actualizó el nodo Change Schema final para incluir el campo y se ajustó el nodo Redshift para recrear la tabla.

- Incompatibilidad por nombres de columnas (PackageTypeID)

El mapeo de columnas entre OrderLines y PackageTypes fallaba porque previamente se había eliminado la llave foránea. Se restauró la columna eliminada, se hizo el join correctamente y se validó su propagación en el flujo completo.

- Redshift no mostraba nuevas columnas

A pesar de que el job terminaba exitosamente, al consultar en Query Editor V2 no aparecían las nuevas columnas. Se revisó que el job estuviera marcado como SUCCEEDED y se usó una consulta al information\_schema.columns para validar el campo.