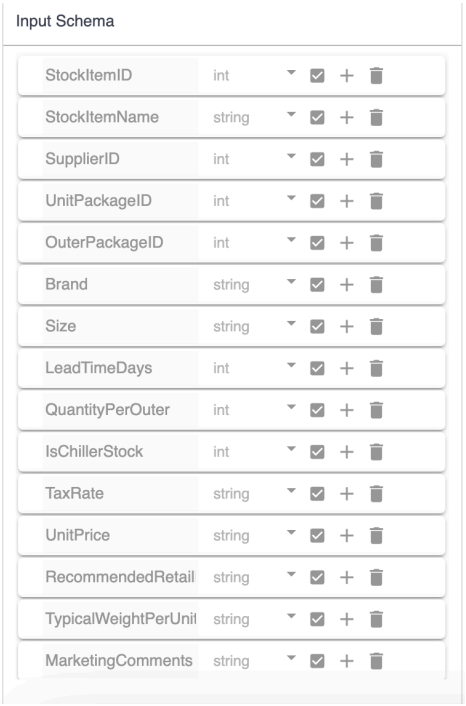
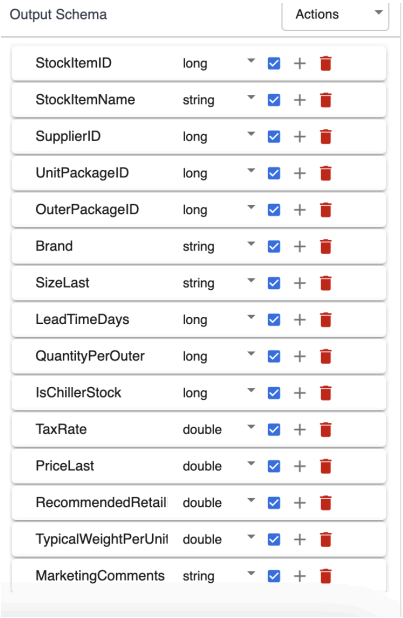
Wrangler:
StockItemsClean

Receta:
rename Size SizeLast
rename UnitPrice PriceLast
drop :ColorID
drop :Barcode
fill-null-or-empty :Brand ' '
fill-null-or-empty :SizeLast ' '
fill-null-or-empty :InternalComments ' '
fill-null-or-empty :Photo ' '
fill-null-or-empty :MarketingComments ' '

Esquema de entrada



Esquema de salida



Explicación y perfilamiento de la receta:

La receta StockItemsClean se encarga de estandarizar y limpiar los datos provenientes del archivo StockItems.csv antes de cargarlos en la dimensión DimStockItem. Esta transformación incluye renombrar columnas para mantener consistencia semántica en el modelo multidimensional —por ejemplo, Size se convierte en SizeLast y UnitPrice en PriceLast, facilitando la distinción entre atributos originales y valores utilizados para cálculos posteriores. También se eliminan columnas irrelevantes para el análisis, como ColorID y Barcode, reduciendo ruido y evitando cargar información que no aporta valor al Data Warehouse. Adicionalmente, se aplican múltiples directivas de imputación mediante fill-null-or-empty, rellenando con valores vacíos aquellos campos que podrían contener nulos, tales como Brand, SizeLast, InternalComments, Photo y MarketingComments. Esto garantiza consistencia en los tipos de datos y evita errores en uniones o cargas posteriores. El perfilamiento de esta receta evidencia que todos los campos numéricos fueron correctamente convertidos a tipos long o double en el Output Schema, y los campos textuales fueron normalizados, lo cual asegura que la dimensión StockItem se alimente con datos limpios, coherentes y aptos para análisis posteriores dentro del modelo multidimensional.

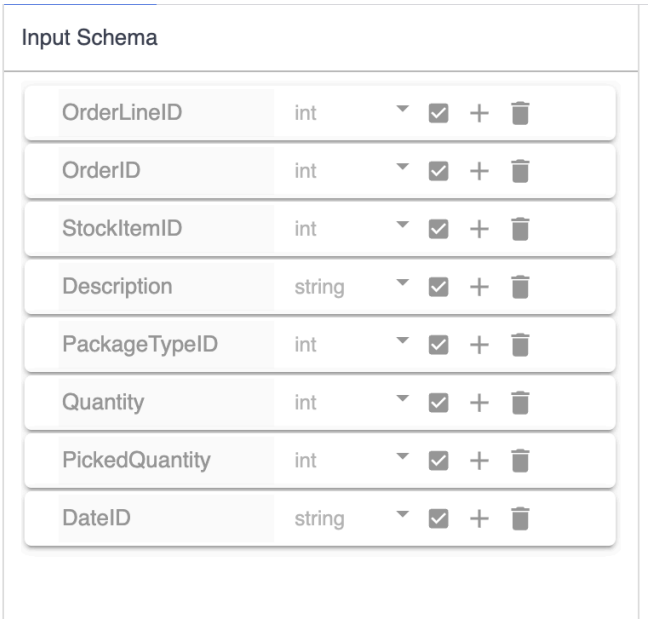
Wrangler:

FormatDateID

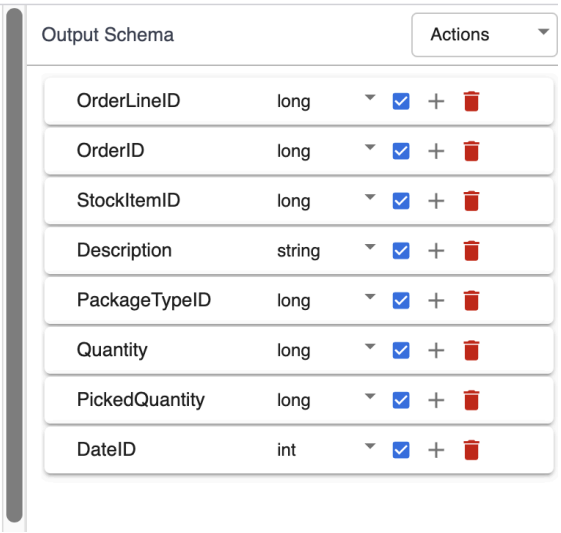
Receta:

```
set-type :DateID string
find-and-replace :DateID s/T.*//g
find-and-replace :DateID s/-//g
set-type :DateID integer
```

Esquema de entrada:



Esquema de salida:



Explicación y perfilamiento de la receta:

La receta FormatDateID tiene como objetivo estandarizar el campo DateID, que inicialmente llega en formato string y con información adicional que impide utilizarlo como una clave numérica consistente dentro del Data Warehouse. El proceso inicia convirtiendo el atributo DateID al tipo string para permitir manipularlo mediante expresiones regulares. Luego, se aplican dos transformaciones find-and-replace: la primera elimina todo el contenido desde la letra “T” hacia la derecha (s/T.*//g), limpiando restos de timestamps provenientes del archivo original; la segunda remueve los guiones del formato de fecha (s/-//g), transformando valores como "2016-05-12" en "20160512". Finalmente, el campo se castea nuevamente a tipo integer, convirtiéndose en un identificador de fecha compacto, consistente y apto para

ser utilizado como llave foránea en la tabla de hechos y como llave primaria en la dimensión DimDates. El perfilamiento confirma que el Output Schema queda limpio, homogéneo y compatible con el modelo multidimensional, mejorando integridad referencial y eficiencia en las uniones posteriores.

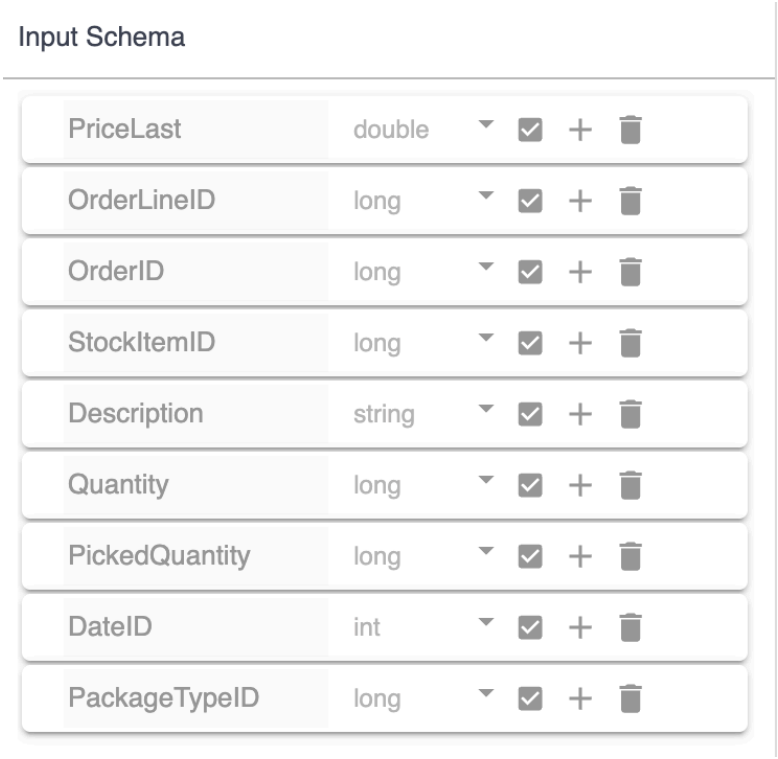
Wrangler:

CreateTotalPrice

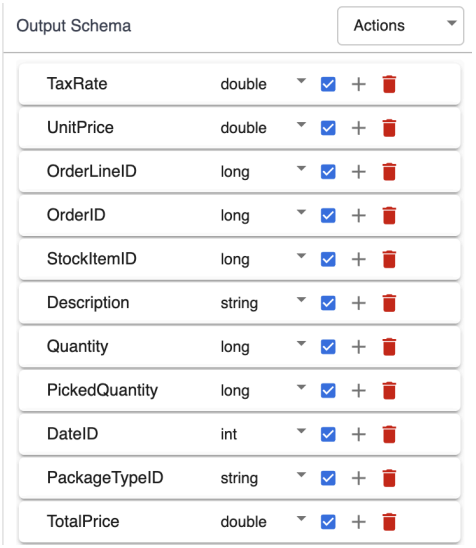
Receta:

set-column TotalPrice Quantity * PriceLast

Esquema de entrada:



Esquema de salida:



Explicación y perfilamiento de la receta:

En el Wrangler CreateTotalPrice, el objetivo principal es generar la nueva medida TotalPrice, necesaria para la tabla de hechos. Para ello, se parte del esquema de entrada que contiene los valores Quantity y PriceLast (precio unitario limpio proveniente de la dimensión de StockItems). La receta implementa una única transformación: `set-column TotalPrice Quantity * PriceLast`, con lo cual se crea un nuevo campo calculado que representa el precio total de cada línea de pedido. Este cálculo sigue la fórmula estándar del negocio TotalPrice = Quantity × PriceLast, garantizando un valor aditivo que posteriormente será usado en análisis de ventas, rentabilidad y volumetría. Durante el perfilamiento se verificó que Quantity y PriceLast estuvieran correctamente tipados como long y double, respectivamente, evitando errores de ejecución en Data Fusion por incompatibilidades de tipos. El resultado es un esquema de salida consistente, donde se conserva la estructura original de OrderLines y se agrega el nuevo campo TotalPrice como una medida lista para análisis.

Wrangler:

Dates

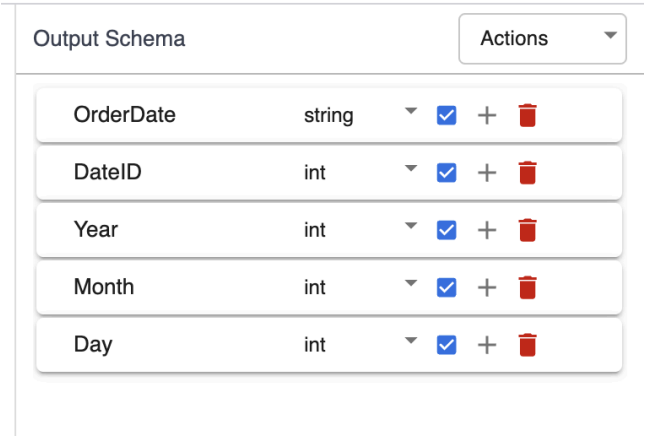
Receta:

drop :CustomerID,:SalespersonPersonID,:PickedByPersonID,:OrderID,:ContactPersonID,:BackorderOrderID,:ExpectedDeliveryDate,:CustomerPurchaseOrderNumber,:IsUndersupplyBackordered,:Comments,:DeliveryInstructions,:InternalComments,:PickingCompletedWhen,:LastEditedBy,:LastEditedWhen
copy :OrderDate :DateID true
split-to-columns :DateID -

find-and-replace :DateID s/-//g
set-type :DateID integer
set-type :DateID_1 integer
set-type :DateID_2 integer
set-type :DateID_3 integer
rename DateID_1 Year
rename DateID_2 Month
rename DateID_3 Day

Esquema de entrada:

Esquema de salida:



Explicación y perfilamiento de la receta:

El wrangler Dates se utiliza para construir la dimensión de tiempo a partir de la tabla Orders, extrayendo y normalizando la información necesaria para generar los atributos DateID, Year, Month y Day. En primer lugar, se realiza un perfilamiento inicial identificando que la tabla contiene una gran cantidad de columnas operativas que no aportan valor a la dimensión de

fechas; por ello, la receta inicia con un drop masivo de atributos como CustomerID, SalespersonPersonID, ContactPersonID, BackorderOrderID, comentarios y metadatos administrativos. Esto reduce ruido y deja únicamente OrderDate como fuente principal de datos temporales. Luego, se duplica OrderDate en una nueva columna DateID, que servirá como clave sustituta. Posteriormente, se aplica split-to-columns para separar el formato AAAA-MM-DD en tres columnas; adicionalmente, se estandarizan los tipos de datos con set-type y se eliminan los caracteres “-” para construir un DateID numérico en formato YYYYMMDD, siguiendo buenas prácticas de modelado dimensional. Finalmente, se renombran las columnas generadas como Year, Month y Day, conformando así una dimensión de fechas consistente, limpia y apta para su uso en un modelo multidimensional.

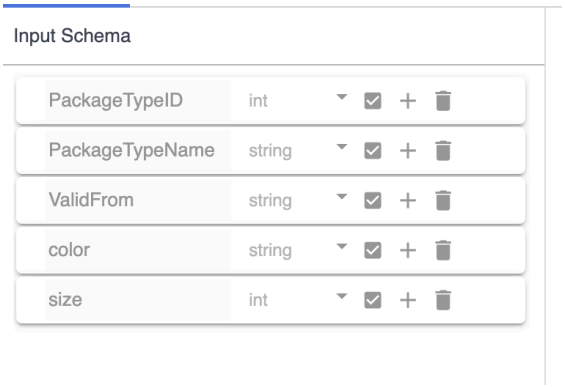
Wrangler:

Generate ID

Receta:

generate-uuid :PackageTypeUuid

Esquema de entrada:



Esquema de salida:

Explicación y perfilamiento de la receta:

En este paso del proceso ETL se utiliza el Wrangler Generate ID para crear un identificador sustituto único (*surrogate key*) para cada tipo de empaque incorporado en la dimensión DimPackageType. Esta transformación aplica la directiva generate-uuid :PackageTypeUuid, con la cual se genera un valor UUID en el nuevo campo *PackageTypeUuid*. El objetivo principal es dotar a la dimensión de un identificador independiente del *PackageTypeID* original proveniente del sistema transaccional, cumpliendo así con las buenas prácticas de modelado dimensional al desacoplar la llave del negocio de la llave técnica del DW.

Desde un punto de vista de perfilamiento, se observa que el Wrangler recibe como *Input Schema* los atributos propios de los tipos de empaque: *PackageTypeID*, *PackageTypeName*, *ValidFrom*, *color* y *size*. Posteriormente, mantiene intactos estos campos y añade la nueva columna *PackageTypeUuid* en el *Output Schema*. Este comportamiento asegura que no se altera la granularidad ni la integridad semántica de los datos, pues el UUID actúa únicamente como una clave técnica adicional sin modificar los atributos originales. Además, la generación automática garantiza unicidad global y elimina riesgos de colisiones que podrían afectar los futuros manejos de historia u operaciones de SCD. En síntesis, este paso prepara la dimensión para integrarse correctamente con el resto del modelo multidimensional y facilita trazabilidad, versionamiento y escalabilidad del almacenamiento analítico.

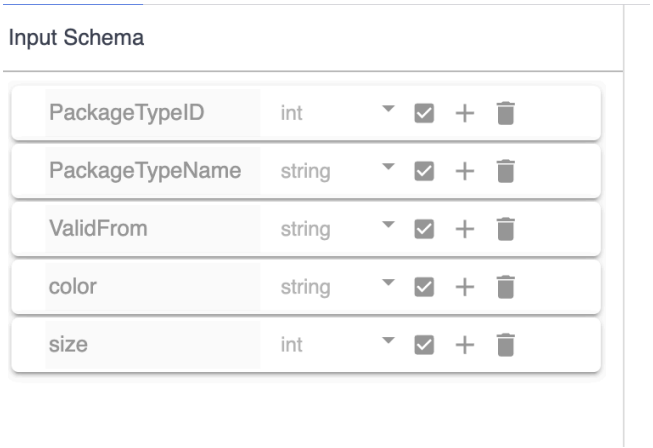
Wrangler:

SCD2

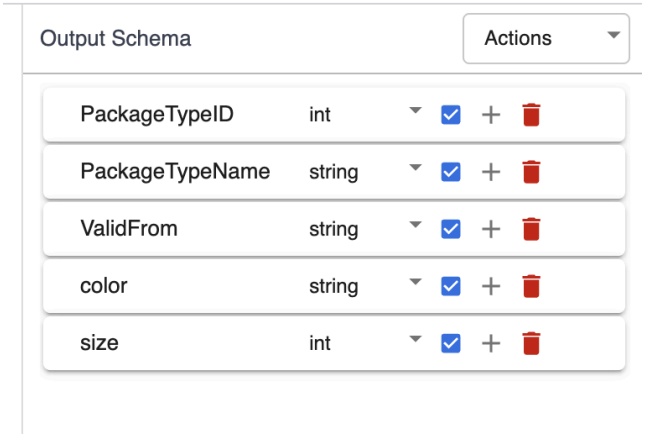
Receta:

```
set-column StartDate CURRENT_DATE()
set-column EndDate ""
set-column IsCurrent "Y"
```

Esquema de entrada:



Esquema de salida:



Explicación y perfilamiento de la receta:

En esta etapa se implementó la lógica de manejo de historia tipo 2 (SCD Type 2) para la dimensión PackageTypes, permitiendo conservar versiones históricas de los atributos cuando cambian en el tiempo. Para ello, se agregaron explícitamente las columnas StartDate, EndDate e IsCurrent, que permiten identificar el periodo de vigencia de cada registro y cuál corresponde a la versión activa. La receta utiliza set-column para inicializar StartDate con la fecha actual usando CURRENT_DATE(), lo cual representa el momento en que la versión del registro entra en vigencia. Posteriormente, se establecen EndDate como vacío ("") para indicar que esta versión sigue activa, y IsCurrent se marca como "Y" para facilitar consultas analíticas que requieran la versión vigente del atributo. Aunque los cambios efectivamente cargados en este laboratorio no incluían múltiples versiones históricas, la estructura generada refleja de manera correcta un diseño SCD2, preparando la dimensión para soportar futuras actualizaciones y permitiendo que el pipeline gestione nuevas versiones si se extiende la funcionalidad.