



# Azure Data Factory by Example

Practical Implementation for  
Data Engineers

—  
Richard Swinbank

Apress®

## Contenido

<b>7. Dataflows .....</b>	<b>3</b>
<b>7.1. Construir un Data Flow .....</b>	<b>3</b>
7.1.1. Enable Data Flow Debugging.....	4
7.1.2. Añadir una transformación de data flow .....	6
7.1.3. Utilizar la transformación Filter .....	11
7.1.4. Utilizar la transformación Lookup .....	14
7.1.5. Utilizar la transformación de Derived Column .....	19
7.1.6. Utilizar la transformación Select .....	24
7.1.7. Utilizar la transformación Sink .....	25
7.1.8. Ejecutar el Data Flow.....	29
<b>7.2. Actualizar una dimensión de producto .....</b>	<b>36</b>
7.2.1. Crear una tabla de dimensión .....	36
7.2.2. Create Supporting Datasets .....	36
7.2.3. Construya el Data Flow de mantenimiento de productos .....	37
7.2.4. Execute the Dimension Data Flow .....	42
<b>Revisión del capítulo .....</b>	<b>44</b>
Conceptos clave .....	44

## 7. Dataflows

---

La actividad Copiar datos es una potente herramienta para mover datos entre sistemas de almacenamiento de datos, pero tiene un soporte limitado para la transformación de datos. Se pueden añadir columnas a la configuración de origen de la actividad, o eliminarlas excluyéndolas de la asignación de origen a destino (source to sink mapping), pero la actividad no admite la manipulación de filas individuales ni permite combinar o separar orígenes de datos.

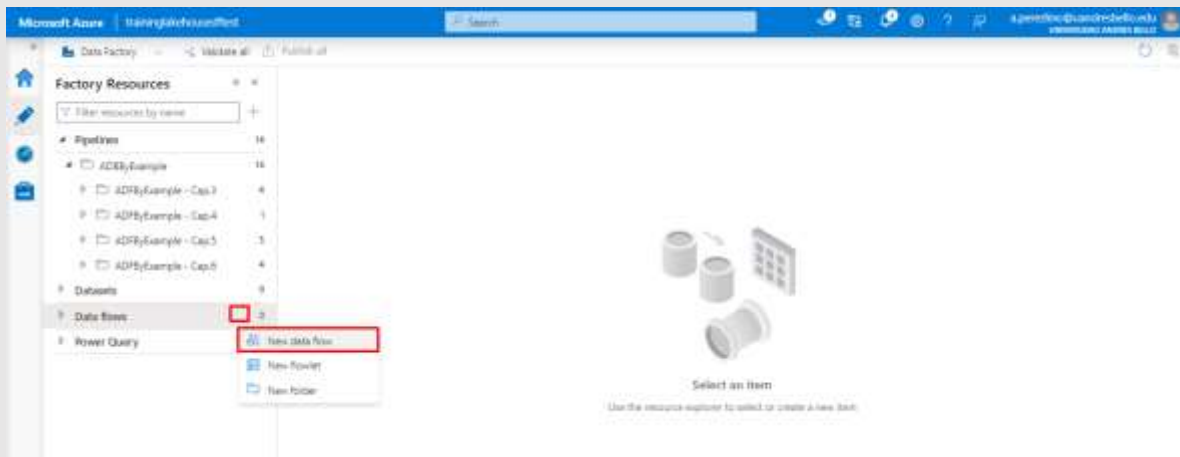
La plataforma Azure admite una serie de herramientas capaces de transformar datos a gran escala, una de las cuales es Azure Databricks. Databricks es una plataforma de análisis en la nube basada en Apache Spark, un framework de código abierto que distribuye automáticamente las cargas de trabajo de procesamiento de datos en un clúster de servidores (denominados nodos) para permitir una ejecución altamente paralela. Los procesos de transformación se implementan utilizando código escrito en uno de los varios lenguajes admitidos, como Scala, R o Python.

Puedes ejecutar los procesos de Azure Databricks desde ADF utilizando una de las actividades de Databricks que se encuentran en la caja de herramientas de actividades, pero esto requiere que implementes las transformaciones de datos de Databricks en el lenguaje de código de tu elección y que crees y gesticiones un espacio de trabajo de Databricks separado en Azure. Azure Data Factory data flows proporciona una ruta alternativa, de bajo código, a la potencia de Azure Databricks. Los data flows se implementan utilizando un editor visual y son convertidos automáticamente por ADF en código Scala para su ejecución en un clúster de Databricks gestionado. Este capítulo le presenta la creación de ADF data flows.

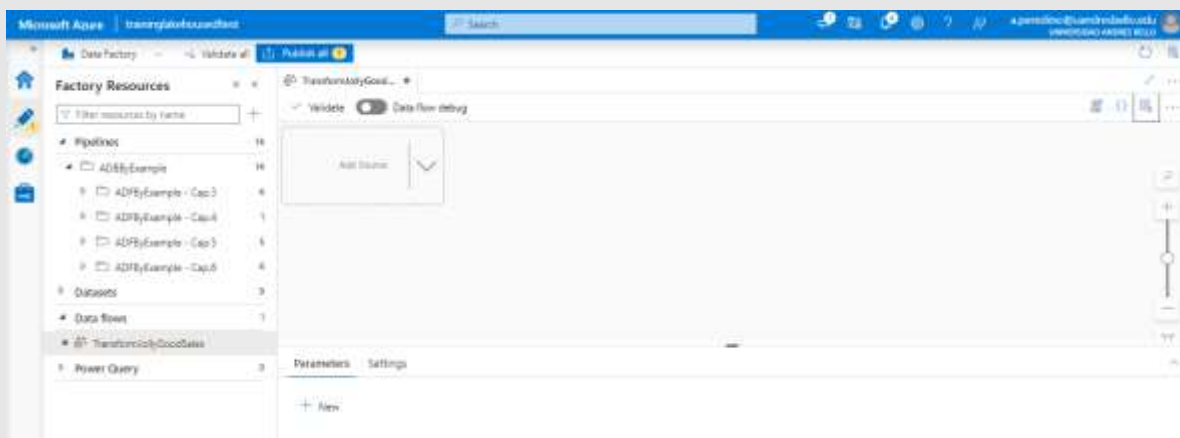
### 7.1. Construir un Data Flow

Los data flows de Azure Data Factory son recursos de ADF independientes y reutilizables, implementados en la UX de ADF mediante un lienzo de flujo de datos visual. La ejecución de un data flow ocurre en un pipeline de ADF usando la actividad Data flow - en esta sección, crearás un data flow, crearás un pipeline para ejecutarlo, y ejecutarás ese pipeline.

1. Abra el espacio de trabajo de creación de ADF UX. En el explorador de Recursos de la Fábrica, encontrará el tipo de recurso Data flows (debajo de los conocidos tipos de recursos Pipelines y Datasets). Cree una nueva carpeta "Capítulo7" utilizando el menú Acciones de Data flows.
2. En el menú Acciones de la carpeta "Capítulo7", haga clic en New data flow (Nuevo flujo de datos).



3. Se abre el lienzo del data flow. Si se trata del primer data flow de su fábrica, se mostrará el mensaje de llamada Comenzar añadiendo una fuente al data flow - desactívelo utilizando su botón de cierre. En la hoja de propiedades, cambie el nombre del data flow a "TransformJollyGoodSales", y luego cierre la hoja de la manera habitual.



Este data flow se utilizará para cargar los datos de ventas ABC de un proveedor de confitería del Reino Unido llamado "Jolly Good Ltd".

#### 7.1.1. Enable Data Flow Debugging

Aunque se desencadena desde una actividad de pipeline, la ejecución de los data flows de ADF Azure Data Factory tiene lugar en un cluster de Databricks. Se debe aprovisionar un clúster antes de que se pueda ejecutar un data flow, ya sea en un entorno publicado o al depurar en la UX de ADF. El aprovisionamiento de un clúster lleva varios minutos, por lo que su primera tarea al desarrollar un data flow es hacer girar un clúster para una depuración conveniente.

1. Ponga el selector de debug de data flow en "On". Cuando está en la posición "Off", como en la Figura 7-1, el color de fondo del conmutador es gris.

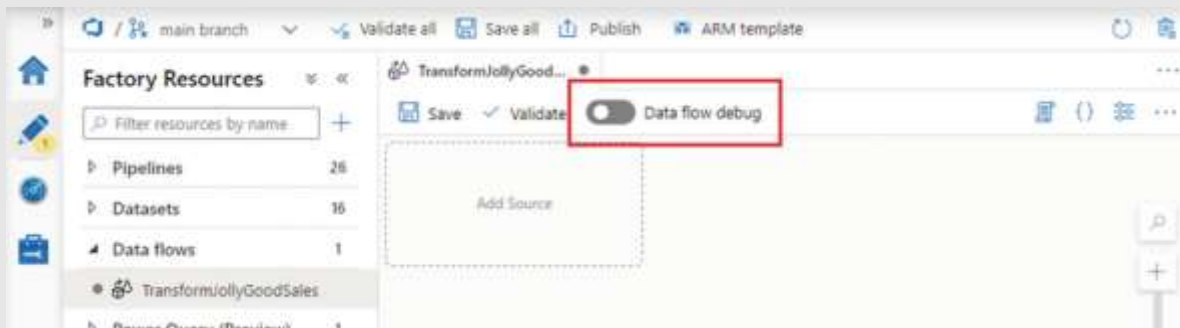


Figura 7-1 Debug de data flow desactivado

2. Aparecerá el cuadro de diálogo Activar data flow debug, que le pedirá que seleccione un Integration runtime y un Time to live (TTL). Deje seleccionados los valores por defecto y haga clic en Aceptar. Los Integration runtime se tratan con más detalle en el capítulo 8.

El color de fondo del selector de Data flow debug cambia a azul y, a su derecha, una rueda giratoria indica que el cluster se está iniciando. Después de unos minutos, la rueda giratoria se sustituye por una marca de verificación en un círculo verde, como se muestra en la Figura 7-2, lo que indica que el clúster está listo para su uso. Cuando el clúster está listo, puede utilizarlo para ejecutar data flows en modo debug desde la UX del ADF.

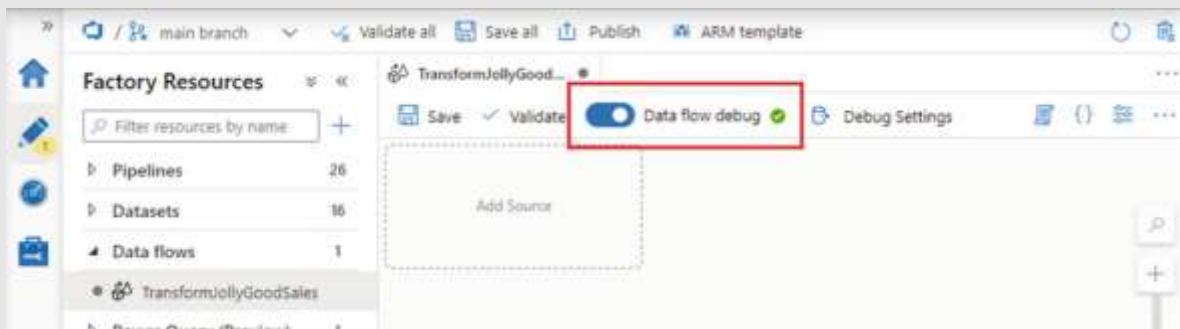


Figura 7-2 Data flow debug habilitado

El tiempo de vida (TTL) por defecto del cluster de debug es de una hora - después de ese tiempo, si el cluster no se utiliza, se apaga automáticamente. Cuando el tiempo de vida del clúster se aproxima, la UX del ADF emite una advertencia, y el icono de la marca verde se sustituye por un círculo naranja relleno. Si desea seguir trabajando con el cluster - y no quiere tener que esperar a que se aprovisiona uno nuevo - debe ejecutar el data flow o realizar otra acción que utilice el cluster.

Cuando haya terminado de desarrollar y depurar, desactive el clúster de depuración - aunque se apagará automáticamente cuando se alcance su TTL, puede ahorrarse el coste de ejecutar el clúster durante el tiempo restante.

### 7.1.2. Añadir una transformación de data flow

Ahora está listo para comenzar el desarrollo del data flow. Un data flow se compone de una secuencia de transformaciones conectadas. Conceptualmente, cada transformación en la secuencia

- ❖ Recibe un flujo de filas de datos de la transformación anterior
- ❖ Modifica las filas en el flujo a medida que pasan
- ❖ Emite las filas modificadas a la siguiente transformación

Para los desarrolladores de SSIS En el Capítulo 3, describí la actividad de datos Copy como una funcionalidad similar a una tarea básica de data flow de SSIS. Los Data Flows de ADF son mucho más parecidos: el comportamiento del lienzo de data flow y las transformaciones es muy parecido al de la superficie y los componentes de data flow de SSIS.

Una transformación especial de Origen lee los datos de una fuente externa y los emite como un flujo de filas para ser transformados por el data flow. Cada data flow contiene al menos una transformación Source.

1. El lienzo del data flow vacío para su nuevo data flow contiene un mosaico de Add Source (añadir fuente) con un contorno punteado. Haga clic en el mosaico para añadir un origen a su data flow. Debe añadir una fuente antes de poder hacer cualquier otra cosa.
2. Si se trata de su primer data flow, se mostrará una llamada de tres pasos, con consejos sobre cómo interactuar con el lienzo del data flow. Siga estos pasos y, en el paso 3, haga clic en **Finalizar**.
3. Al igual que las actividades del ADF en el lienzo de creación de pipelines, las transformaciones del data flow se configuran utilizando un panel de configuración con pestañas debajo del lienzo del data flow. Cada transformación tiene un nombre único dentro del data flow llamado **output stream name** - establezca este valor adecuadamente en la pestaña **Source settings** (mostrado en la Figura 7-3).

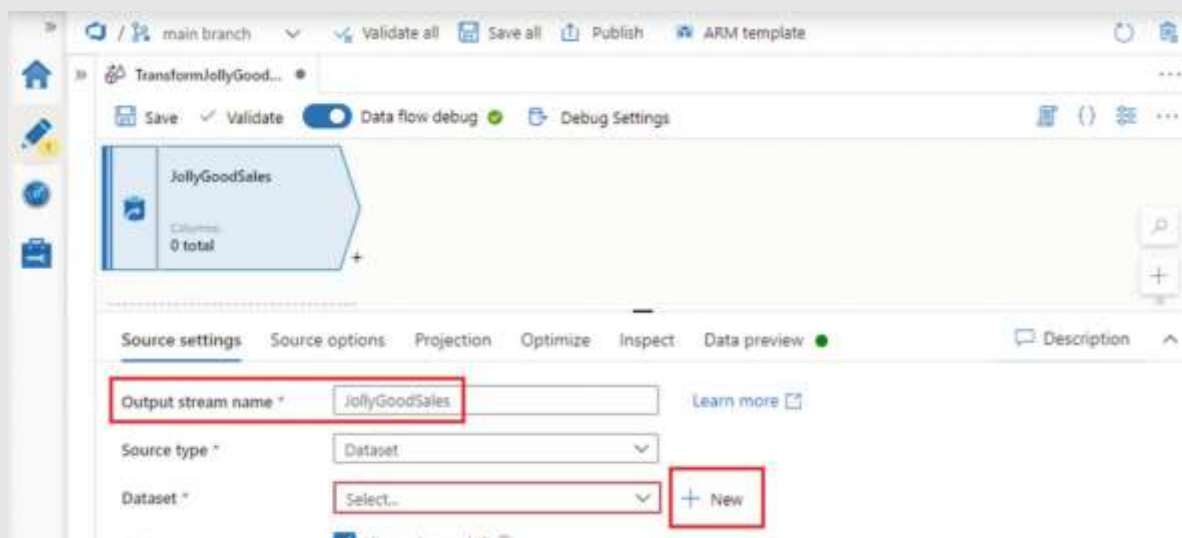
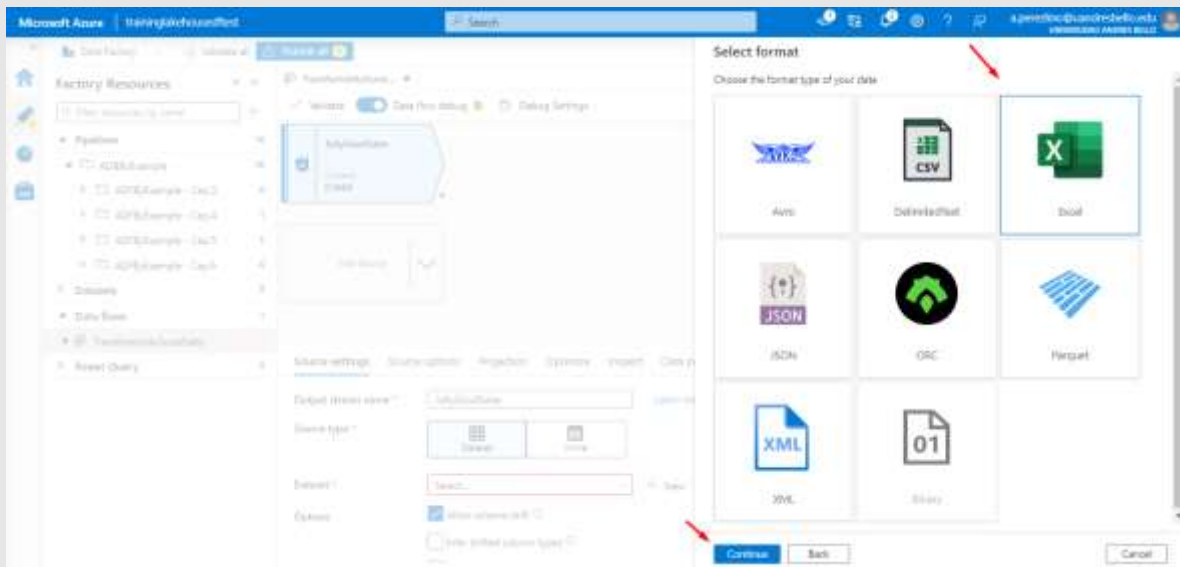
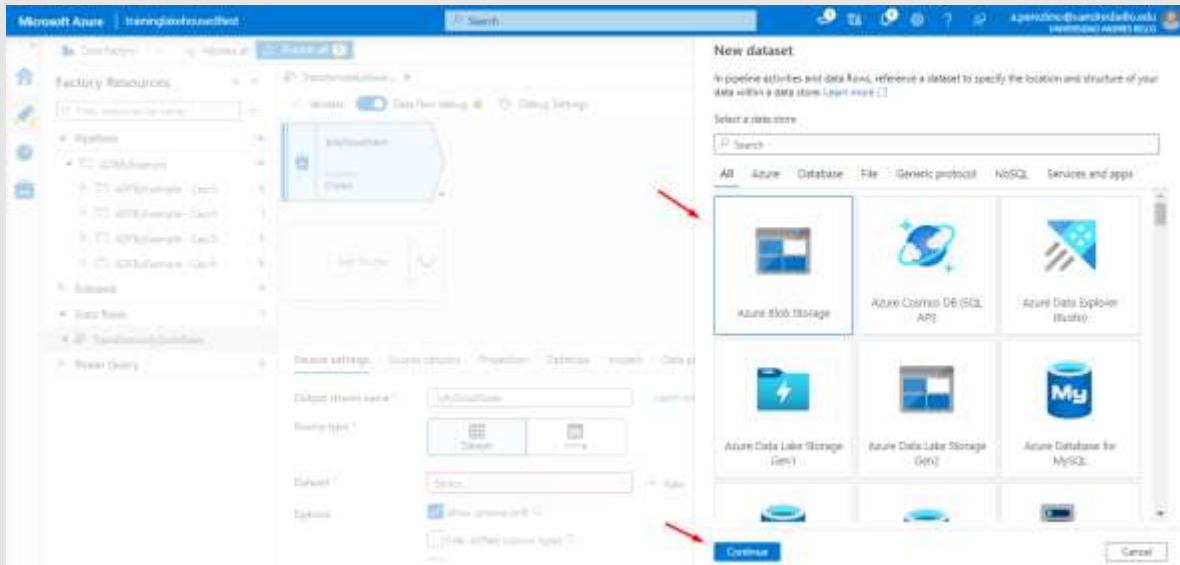


Figura 7-3 Data flow source transformation's Source settings tab

4. Asegúrese de que el Source type de la transformación está configurado como "Dataset" y, a continuación, a la derecha de la lista desplegable Dataset, haga clic en el botón + New.

5. Se abrirá la hoja de New dataset, ya conocida en capítulos anteriores. Seleccione el almacén de datos Azure Blob Storage, haga clic en Continuar y, a continuación, elija el formato de archivo Excel (los datos de ventas de Jolly Good se suministran en archivos de hoja de cálculo Excel). Haga clic en Continuar.



6. Nombra el dataset, luego elige tu original blob storage linked service (un servicio vinculado que no define parámetros). Busque y seleccione el archivo "Sales Apr-Sep 2020.xlsx" de la carpeta "JollyGood" del contenedor "sampledata". Seleccione el nombre de la hoja "SALES" y asegúrese de que la primera fila como cabecera está marcada. La hoja completada se muestra en la Figura 7-4 - haga clic en OK para crear el dataset.



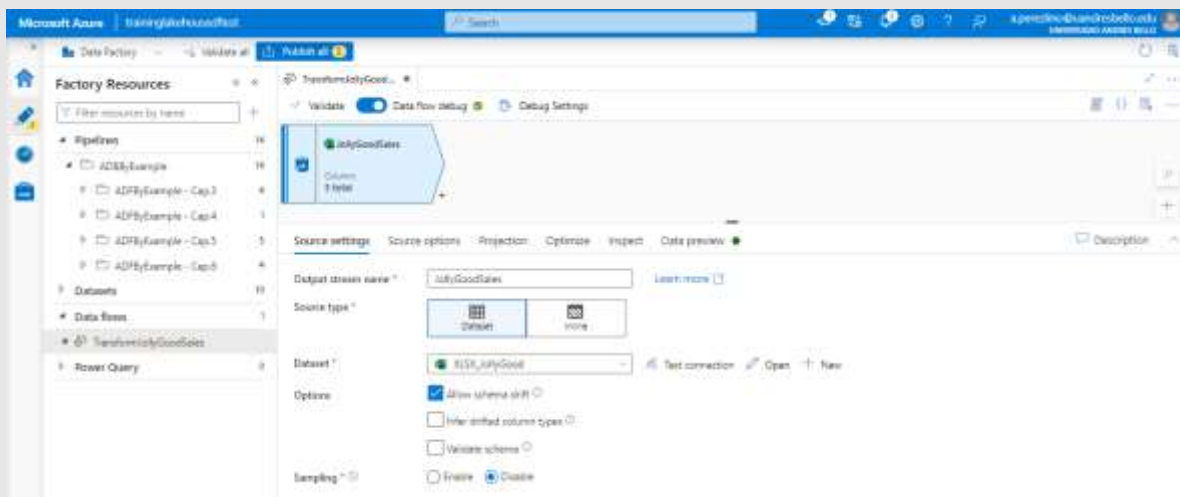
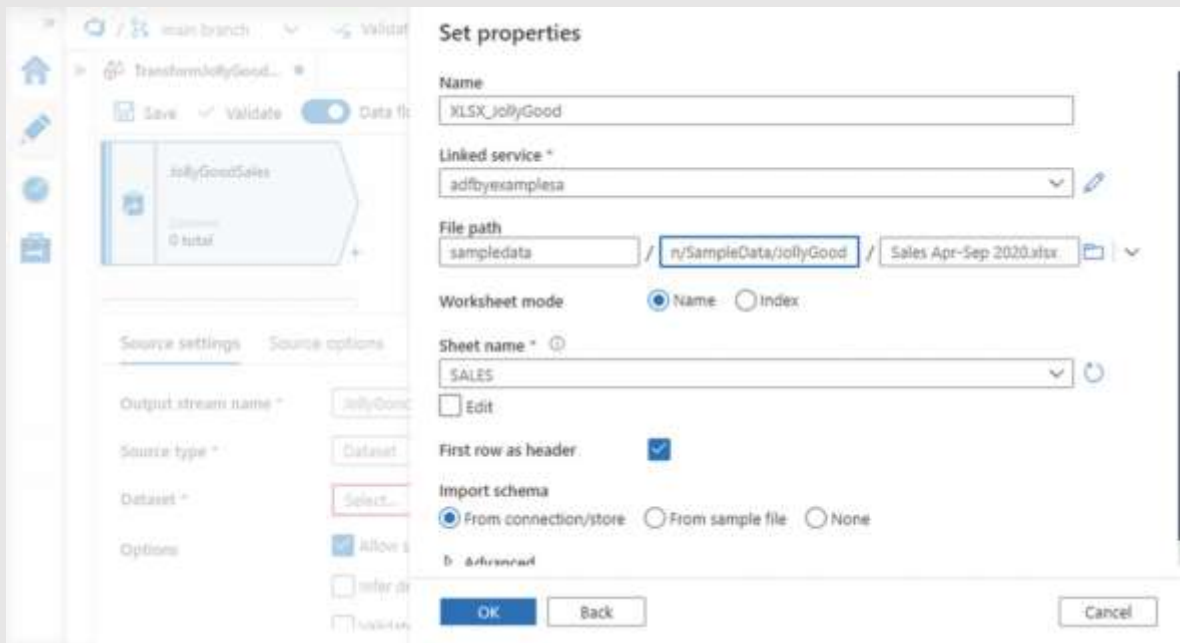
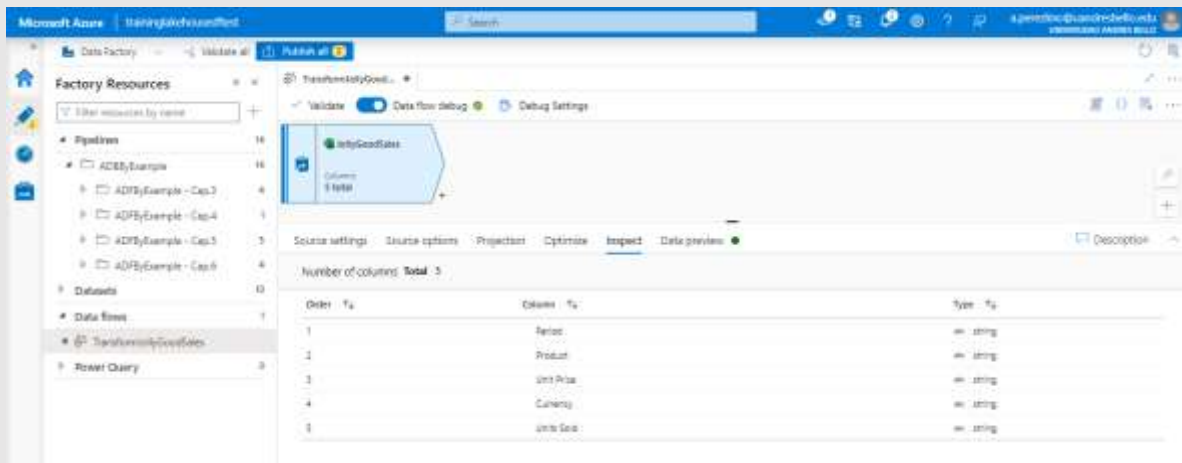


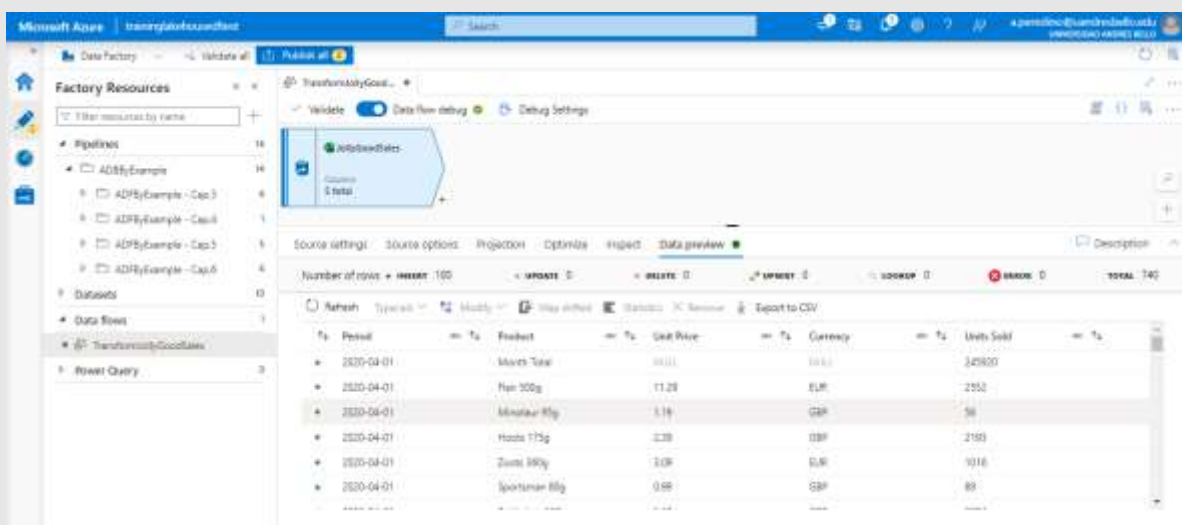
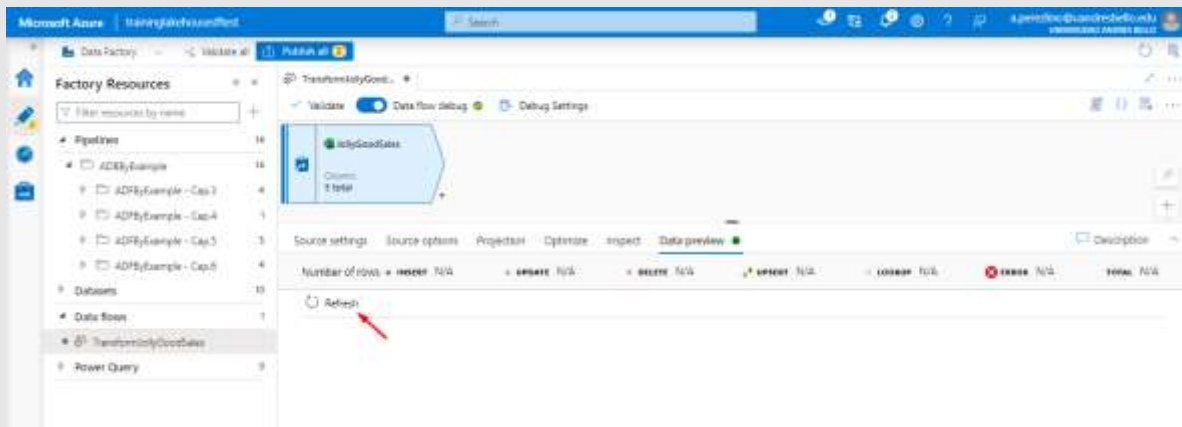
Figura 7-4 Propiedades del dataset de Excel de Jolly Good

7. Seleccione la pestaña Inspect de la transformación de origen. Esta pestaña - que también aparece para cualquier otro tipo de transformación - proporciona detalles del esquema de entrada y/o salida de una transformación. (Una transformación Origen no tiene esquema de entrada). El esquema que se muestra aquí es el que se ha importado al nuevo dataset.



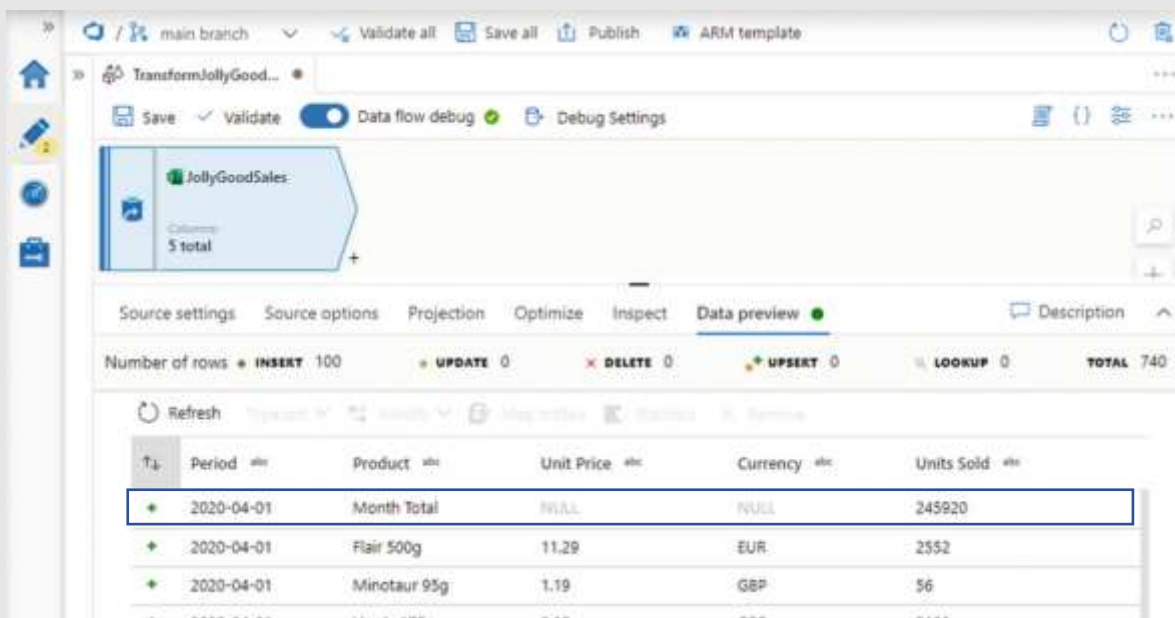


8. Seleccione la pestaña Data preview de la transformación de origen. Esta pestaña, también compartida por todos los tipos de transformación, le permite previsualizar los datos emitidos por una transformación. Haga clic en **Refresh** para cargar la vista previa.



**Consejo** La vista previa de datos sólo está disponible con el modo de depuración activado, es decir, cuando se está ejecutando un cluster de depuración. Si la UX de ADF le advierte que su sesión de cluster de debug está a punto de agotarse, la vista previa de la salida de una transformación es una forma conveniente de extender el TTL del cluster.

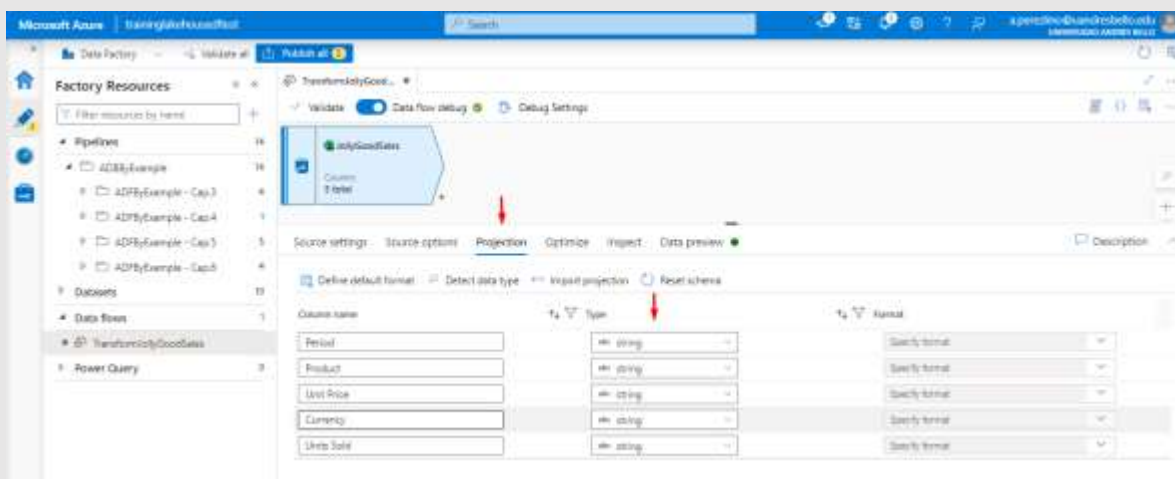
9. En la vista previa (mostrada en la Figura 7-5), observe las características de los datos de ventas de Jolly Good que serían difíciles de manejar utilizando una actividad de copia de datos. El valor de las ventas de productos se informa como una combinación de unidades vendidas (units sold) y precio unitario (unit price) en lugar de un único total de ventas. El precio unitario se proporciona en una mezcla de monedas. El archivo incluye una fila total para cada mes que debe ser excluida. Puede transformar todo esto mediante un data flow.



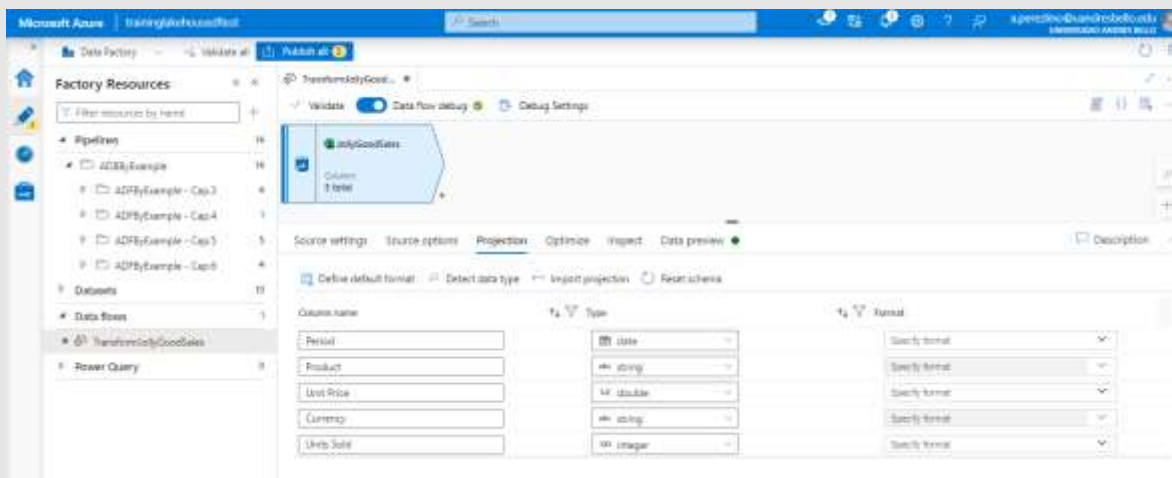
Period	Product	Unit Price	Currency	Units Sold
2020-04-01	Month Total	NULL	NULL	245920
2020-04-01	Flair 500g	11.29	EUR	2552
2020-04-01	Minotaur 95g	1.19	GBP	56

Figura 7-5 Vista previa de los datos de ventas de Jolly Good

10. Seleccione la pestaña Projection de la transformación Source. Esta pestaña es específica de la transformación de origen y muestra las cinco columnas presentes en la hoja "SALES" del archivo de Excel. Todos los tipos de columnas se presentan como "string" - utilice el menú desplegable Type de cada columna para refinar los tipos. Como sugieren los valores de datos previsualizados en la Figura 7-5, la columna "Period" es del tipo "date", "Unit Price" es del tipo "double", y "Units Sold" es una columna "integer".



Column name	Type	Format
Period	string	Specify format
Product	string	Specify format
Unit Price	string	Specify format
Currency	string	Specify format
Units Sold	string	Specify format



11. Haga clic en Guardar todo en la barra de herramientas del ADF UX para guardar su trabajo hasta el momento (incluyendo el nuevo dataset de Excel).

Al igual que otros recursos de fábrica, los data flows se guardan en su repositorio Git como archivos JSON. La secuencia de transformaciones de los data flows se guarda como Script de Data Flow, incrustado en el atributo `properties.typeProperties.script` del archivo JSON del data flow. Puede utilizar el botón de Code (icono de llaves) en la parte superior derecha del lienzo de data flow para ver el JSON de data flow y puede inspeccionar el Script de Data Flow formateado directamente utilizando el botón de Script (al lado del botón de Code).

### 7.1.3. Utilizar la transformación Filter

El formato de datos de ventas de Jolly Good incluye filas de totales mensuales intercaladas con datos de ventas específicos del producto - el primero de ellos es visible en la vista previa de datos que se muestra en la Figura 7-5. Excluya estas filas utilizando la transformación Filter del data flow.

Consejo Es posible que haya observado una actividad llamada Filter en la caja de herramientas de las actividades de canalización. La actividad de pipeline tiene un propósito diferente, ya que le permite seleccionar un subconjunto de elementos de un array de entrada.

1. Para añadir una transformación al data flow, haga clic en el pequeño botón "+" situado en la parte inferior derecha de la transformación Origen en el lienzo del data flow. Este botón aparece en la misma posición para todas las transformaciones, excepto en la de Sink.
2. Aparece un menú emergente de transformaciones disponibles, como se muestra en la Figura 7-6. Busque y seleccione la transformación Filter (hacia la parte inferior de la lista). Cuando la transformación se haya añadido al lienzo del data flow, establezca su Output stream name en la pestaña de configuración de los ajustes de Filter.

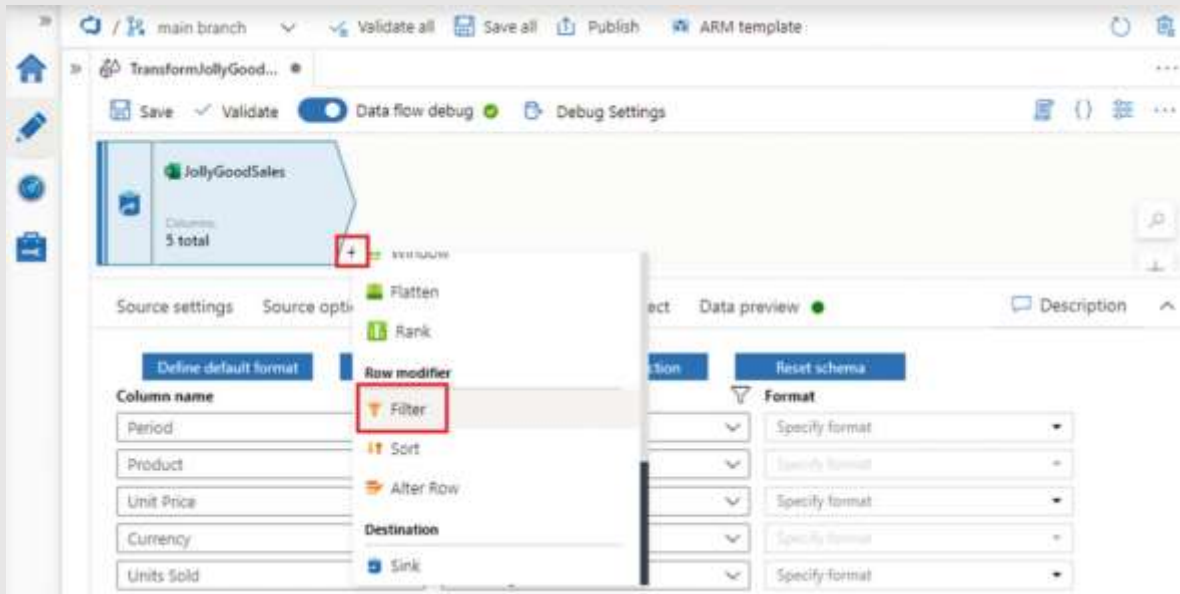
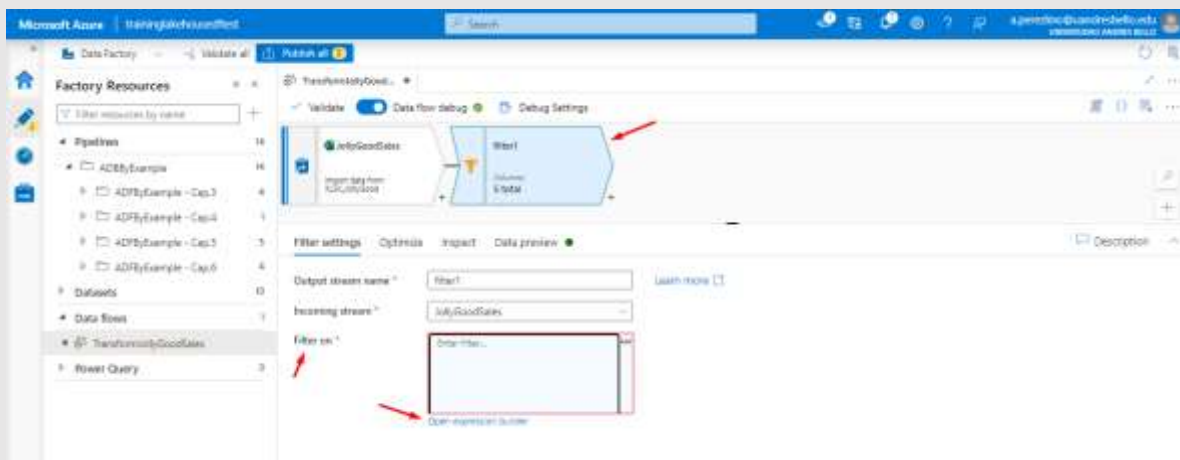


Figura 7-6 Conectar la transformación Filter a la transformación Source

3. También se encuentra en la pestaña Filter settings un área de texto Filter on. Esto almacena una expresión de data flow utilizada para seleccionar filas para la salida de la transformación. Haga clic en el área de texto para abrir el constructor de expresiones visuales de data flow.



4. La Figura 7-7 muestra el generador de expresiones visuales para la transformación de filtro, que incluye un panel de expresiones sobre una barra de herramientas de operadores y un menú de elementos de expresiones. La lista de valores de Expresión corresponde al tipo de elemento seleccionado. Estas características - junto con el panel de vista previa de datos en la parte inferior de la ventana del constructor - están siempre disponibles en el constructor de expresiones.

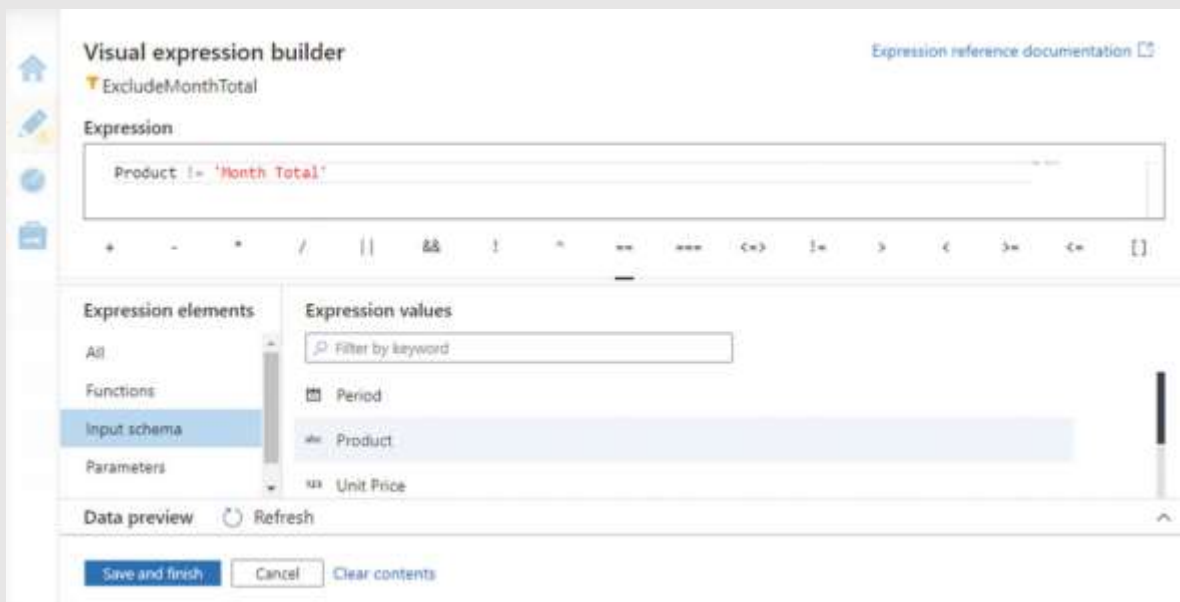
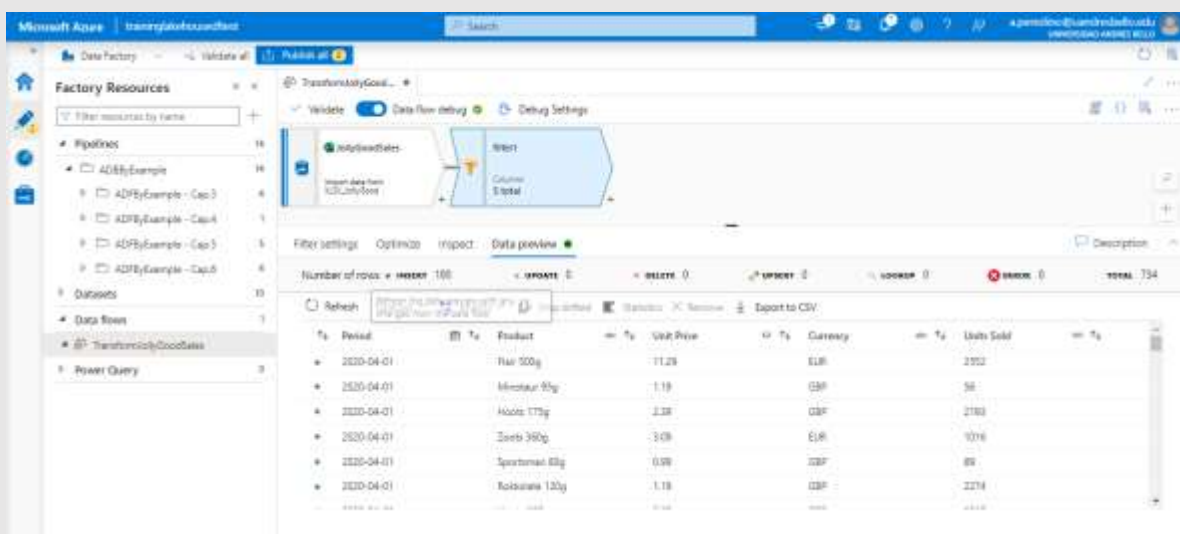


Figura 7-7 Constructor de expresiones visuales para la transformación Filter

Para construir la expresión que se muestra en la Figura 7-7, seleccione la columna "Product" de la lista de valores de expresión en el esquema de entrada, utilice la barra de herramientas de operadores para añadir el operador No igual (!=) y, a continuación, añada a mano 'Month Total' al panel de expresiones. Haga clic en Guardar y termine.

**Nota** El lenguaje de expresión de data flow es diferente al utilizado en las expresiones de pipeline de ADF y es mucho más rico. Todas las comparaciones de texto distinguen entre mayúsculas y minúsculas, a menos que utilice una opción que no distinga explícitamente entre mayúsculas y minúsculas (como el operador <=> o su función equivalente equalsIgnoreCase). A diferencia de las expresiones pipeline, el lenguaje de expresiones data flow incluye una variedad de operadores infijos, mostrados en la barra de herramientas de operadores de la Figura 7-7.

5. Seleccione la pestaña Data preview de la transformación Filter y haga clic en Refresh para cargar la vista previa. Observe que la fila inicial de **Month Total** ya no aparece.



6. Guarde el data flow actualizado.

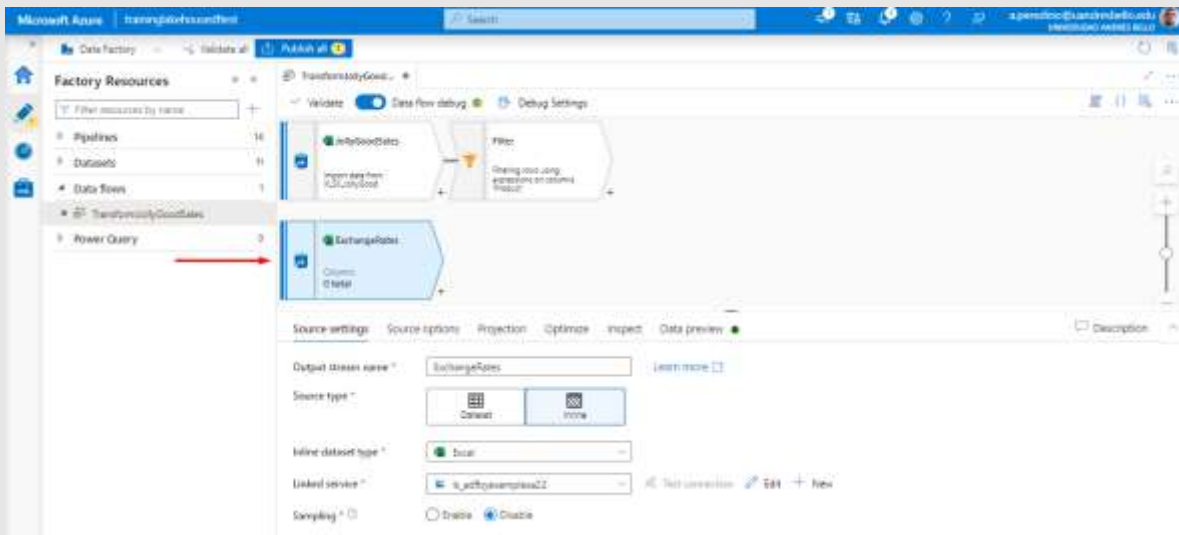
#### 7.1.4. Utilizar la transformación Lookup

La transformación Lookup le permite utilizar valores de un flujo de datos para buscar filas coincidentes en otro flujo de datos. (Esto es diferente de la actividad Lookup de ADF, que carga un dataset en un pipeline, permitiéndole referirse a elementos dentro de él). En esta sección, utilizará la transformación Lookup para obtener información sobre el tipo de cambio de las monedas en el archivo de datos de ventas de Jolly Good.

##### Add a Lookup Data Stream

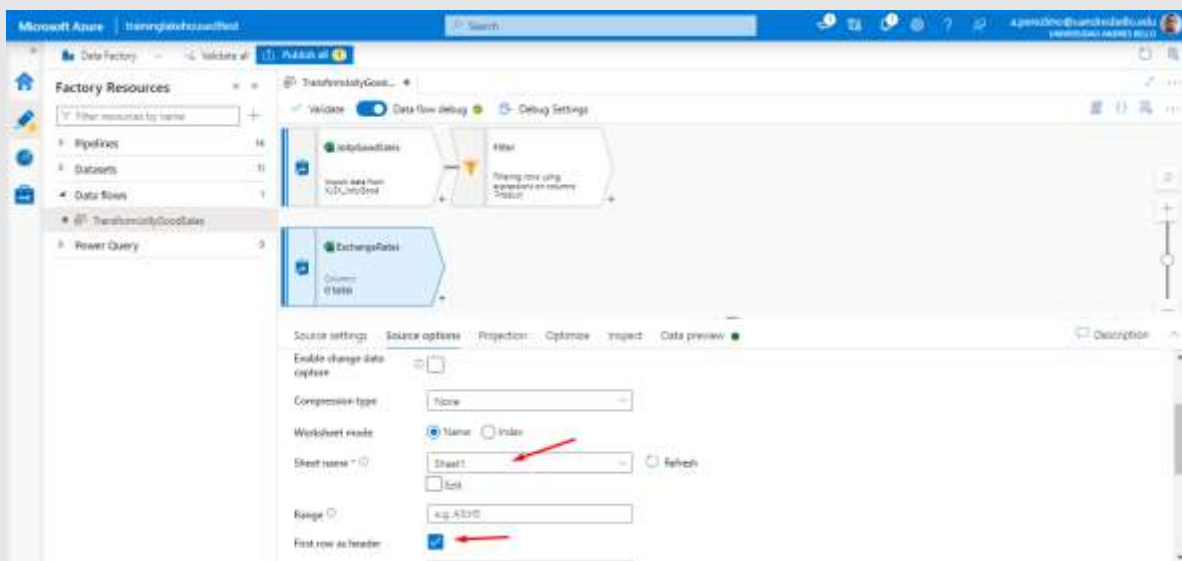
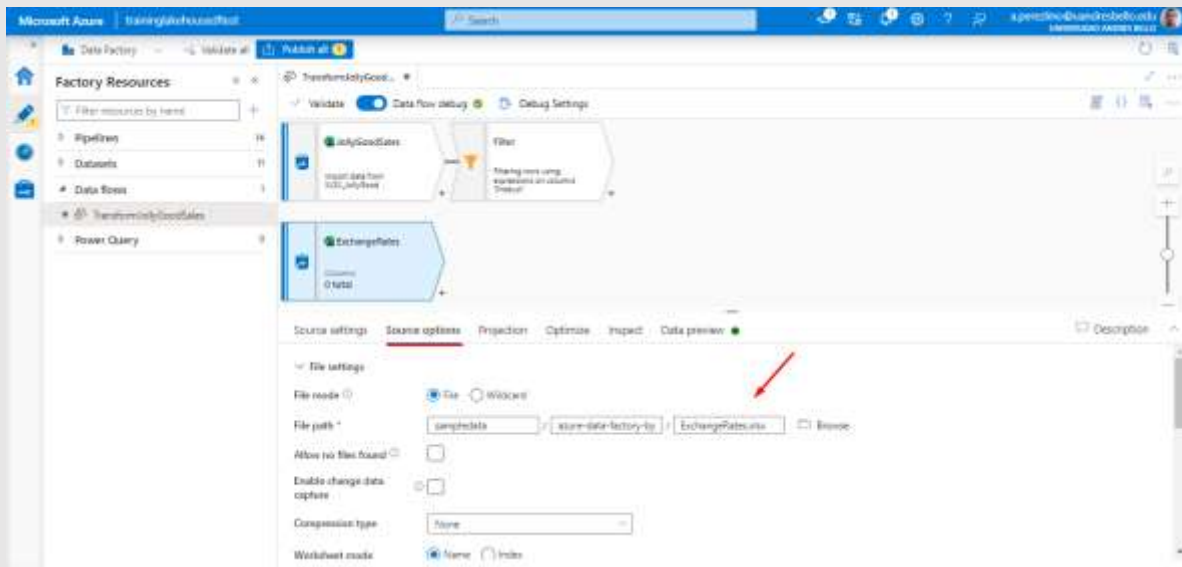
Se requiere un segundo data stream que contenga información sobre el tipo de cambio para la actividad Lookup.

1. Añada otra fuente de datos a su data flow haciendo clic en el mosaico Add Source (añadir fuente) que se muestra debajo de su transformación de fuente de datos de ventas de Jolly Good. Nómbrelo "**ExchangeRates**".
2. Además de los datasets del ADF, las transformaciones de Source del data flow admiten una serie de dataset en línea (inline datasets). El conjunto de formatos admitidos por los datasets en línea no es el mismo que el admitido por los objetos dataset de ADF, aunque hay cierto solapamiento. Establezca la opción Source type de la transformación en "**Inline**". En Inline dataset type seleccione la opción "**Excel**".
3. Aparece un desplegable de Linked service debajo de Source type - elige el servicio vinculado para tu cuenta de blob storage.

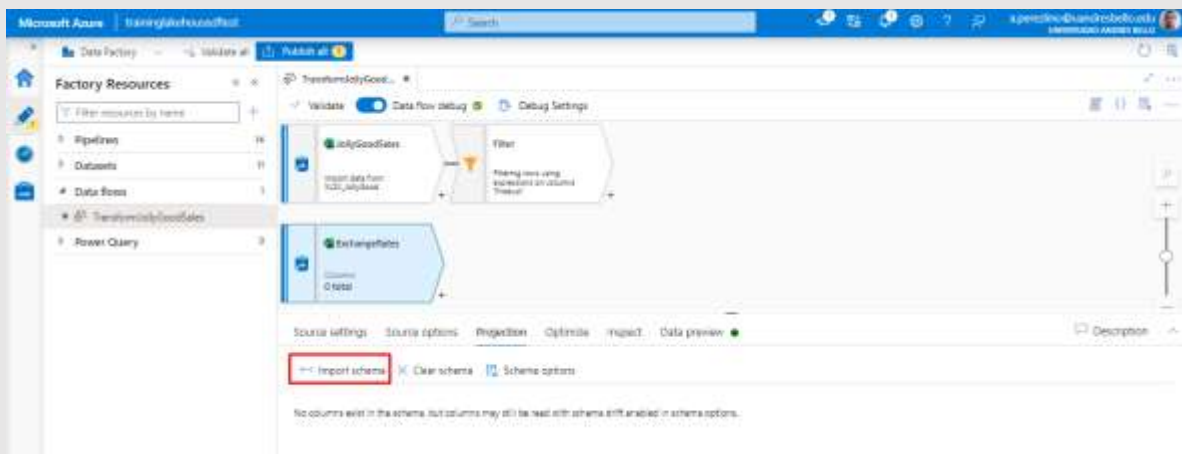


4. Seleccione la pestaña Source options: ahora contendrá opciones específicas para los datasets de blob storage. Utilice el botón Browse (Examinar) situado a la derecha de los campos File path (Ruta del archivo) para seleccionar el archivo "ExchangeRates.xlsx" de la carpeta "SampleData" del contenedor "sampledata". Seleccione el nombre de la hoja "Sheet1" y asegúrese de que First row as header está marcada.





5. Seleccione la pestaña Projection (Proyección). Está vacía porque todavía no se ha importado ninguna información del esquema: el dataset inline no tiene ningún objeto de dataset preexistente al que referirse. Haga clic en Import schema. En la hoja de Import schema que aparece, haga clic en Importar para aceptar las opciones por defecto y continuar. Aparecerán cuatro columnas debidamente escritas, como se muestra en la Figura 7-8.





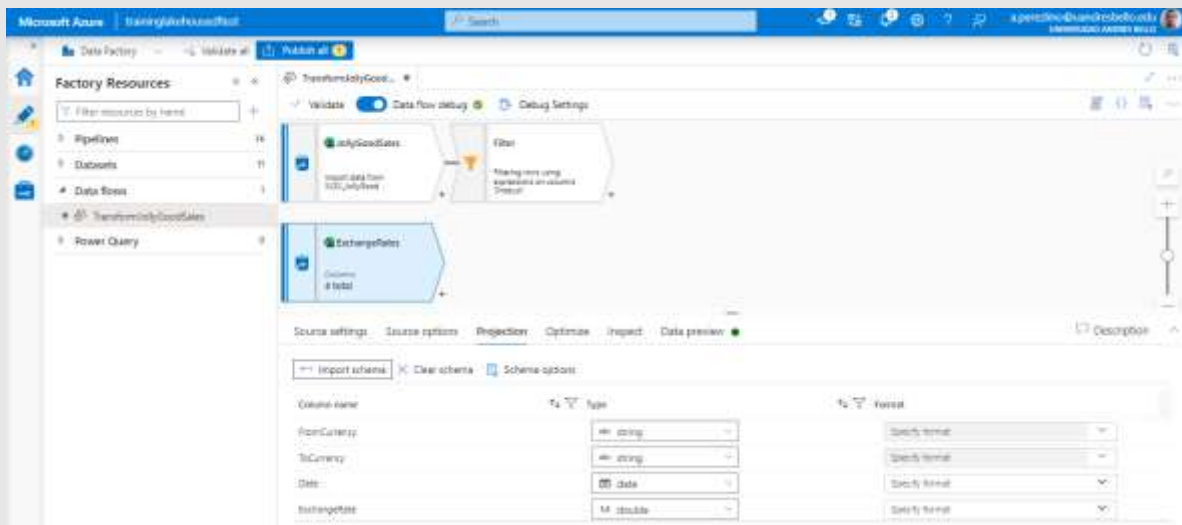
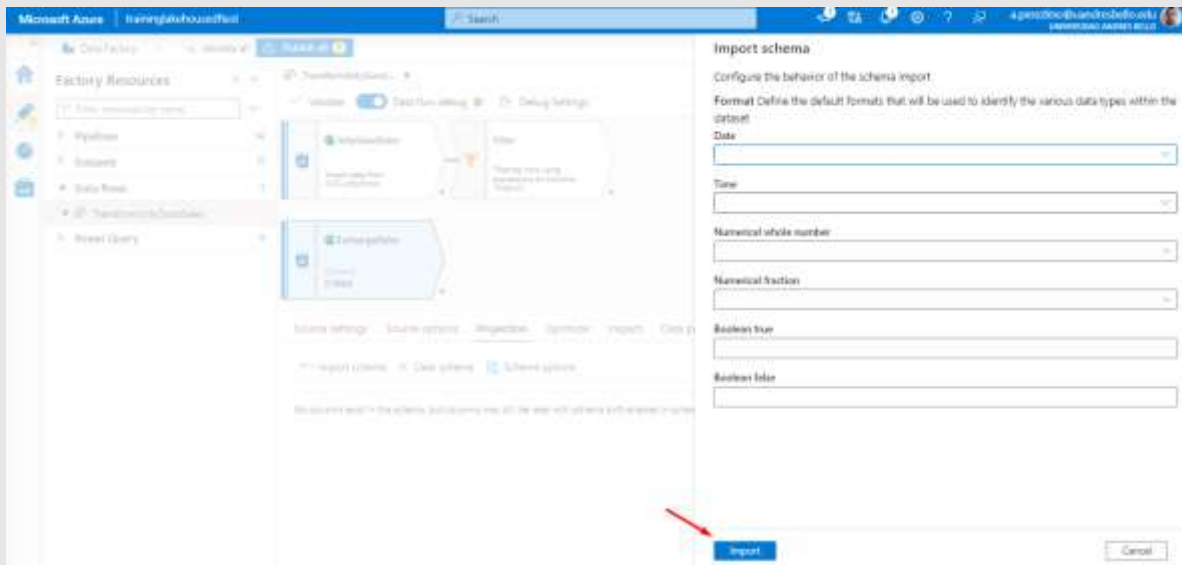
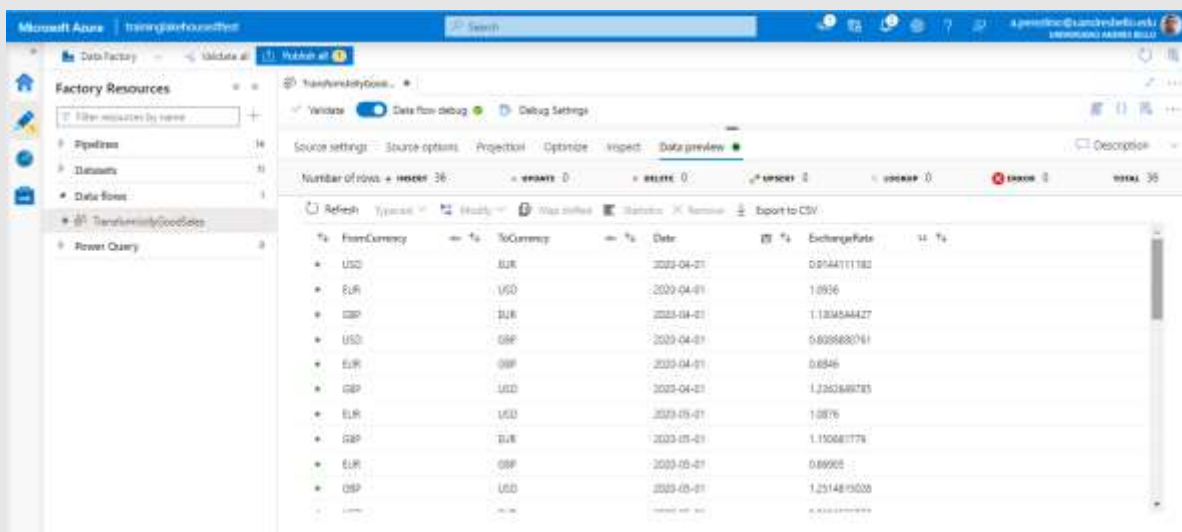
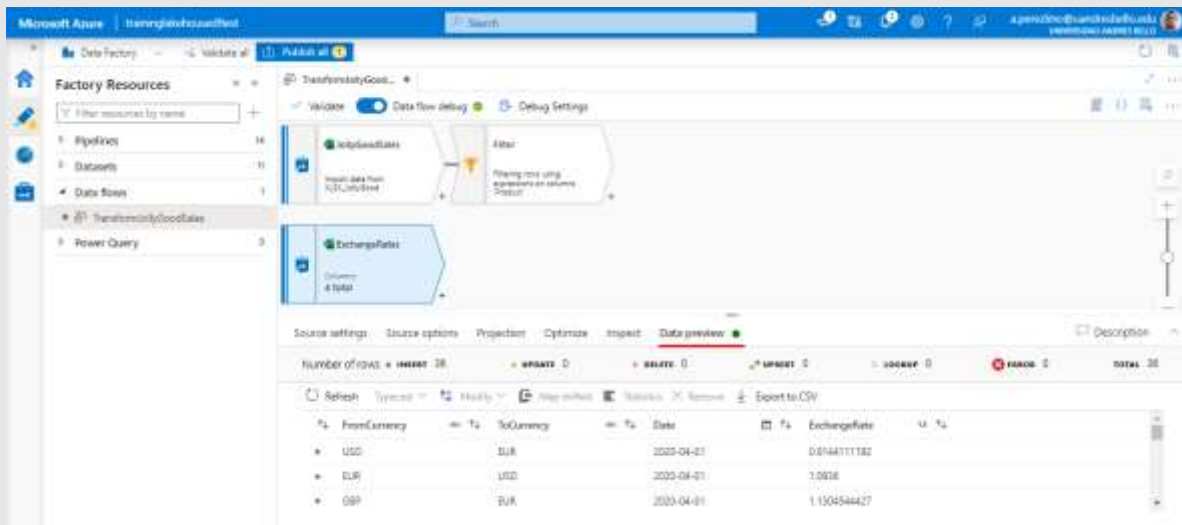


Figura 7-8 Esquema de exchange rate importado en la pestaña de Projection configuration

6. Seleccione la pestaña Data preview y haga clic en Refresh para inspeccionar los datos del archivo. Los datos incluyen los tipos de conversión entre tres divisas (USD, GBP y EUR) el primer día de cada mes del período comprendido entre abril y septiembre de 2020.



7. La transformación Lookup se comporta de forma similar a un join SQL. Para asegurarse de que el lookup se une a la fila correcta, filtre el source de exchange rate para excluir las conversiones a monedas distintas del USD. La Figura 7-9 muestra una transformación Filter configurada para conseguirlo.

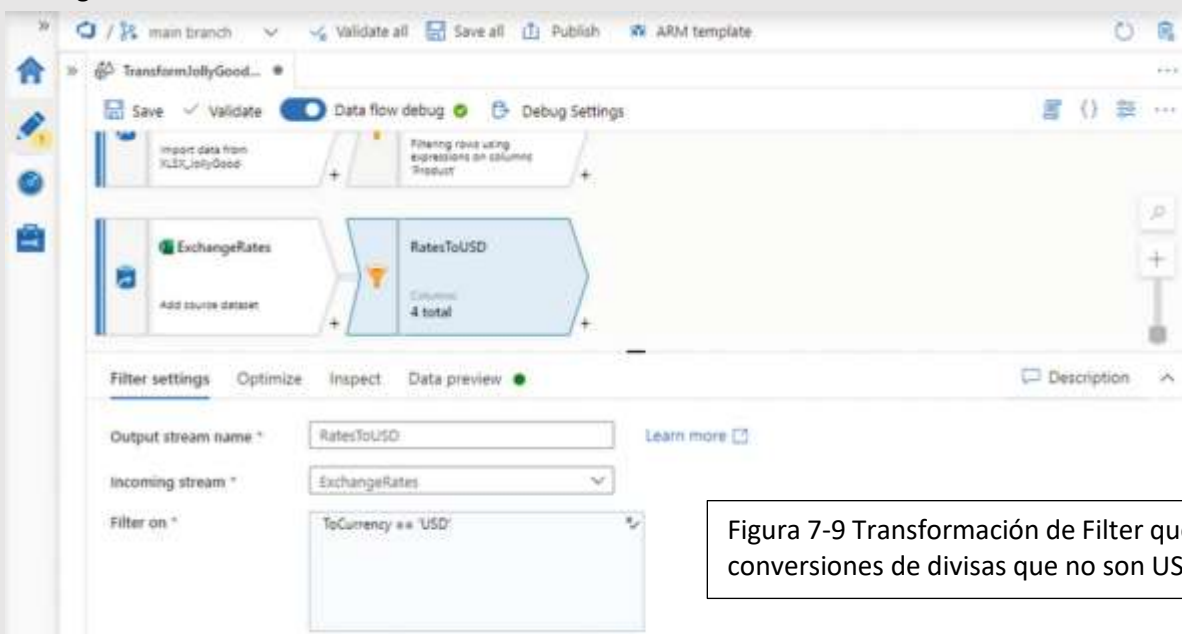


Figura 7-9 Transformación de Filter que excluye las conversiones de divisas que no son USD

## Añadir la transformación Lookup

El exchange rate stream que ha preparado está ahora listo para ser utilizado como un rate lookup.

1. Utilice el botón "+" en la transformación Filter del data stream Jolly Good para conectar una transformación Lookup. En la pestaña Lookup settings, establezca el Output stream name adecuadamente.
2. El campo Primary stream se rellena previamente con el nombre de la transformación ascendente a la que ha conectado el Lookup. Establezca el valor de Lookup stream al output stream name de la transformación Filter de exchange rate - tenga cuidado de seleccionar la transformación correcta en el exchange rate stream.
3. Match multiple rows (Hacer coincidir varias filas) y Match on (Hacer coincidir) el control del comportamiento del join. Marcando la casilla de verificación Match multiple rows crea un efecto como el de un join SQL, donde cada par de registros coincidentes se emite en el flujo de salida - sin esta opción seleccionada, Match on determina qué par coincidente se emite. En este ejemplo, se requiere el par coincidente en el que el valor de "ToCurrency" es "USD" - este requisito es más sofisticado de lo que Match on puede soportar y es la razón por la que filtró el exchange rate stream por adelantado.
4. Especifique dos condiciones en el Lookup. En primer lugar, haga coincidir el campo "Currency" de la izquierda con el campo "FromCurrency" de la derecha del Lookup. Por defecto, las condiciones del Lookup contienen sólo un criterio - añada un segundo haciendo clic en el botón "+" a la derecha de la primera condición. Haga coincidir el campo "Period" de la izquierda con el campo "Date" de la derecha. Ambas condiciones deben utilizar el operador de igualdad == para la coincidencia. La Figura 7-10 muestra el lookup correctamente configurado.

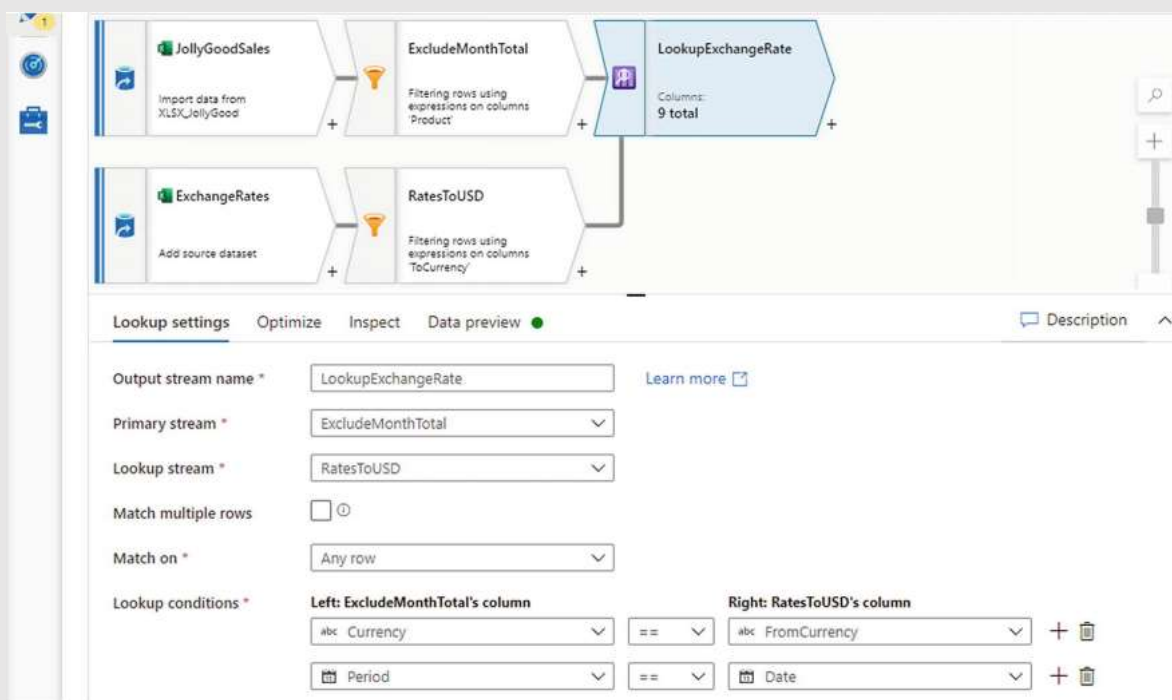


Figura 7-10 Configured exchange rate Lookup transformation

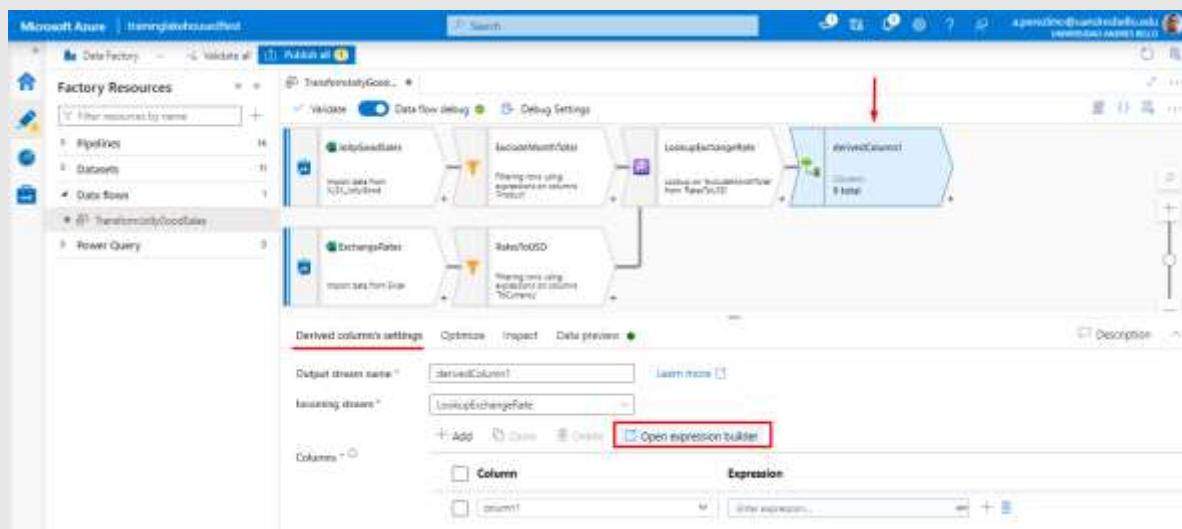
5. Seleccione la pestaña Data preview y haga clic en Refresh para inspeccionar la salida de la transformación. Guarde los cambios.

#### 7.1.5. Utilizar la transformación de Derived Column

La transformación Columna derivada le permite añadir nuevas columnas al data flow. En esta sección, utilizará la transformación para añadir las columnas necesarias para la tabla [dbo].[Sales\_LOAD]: total sales value (en USD), retailer, y el número de secuencia de ejecución (run sequence number) utilizado para el logging de la ejecución del pipeline (pipeline execution logging).

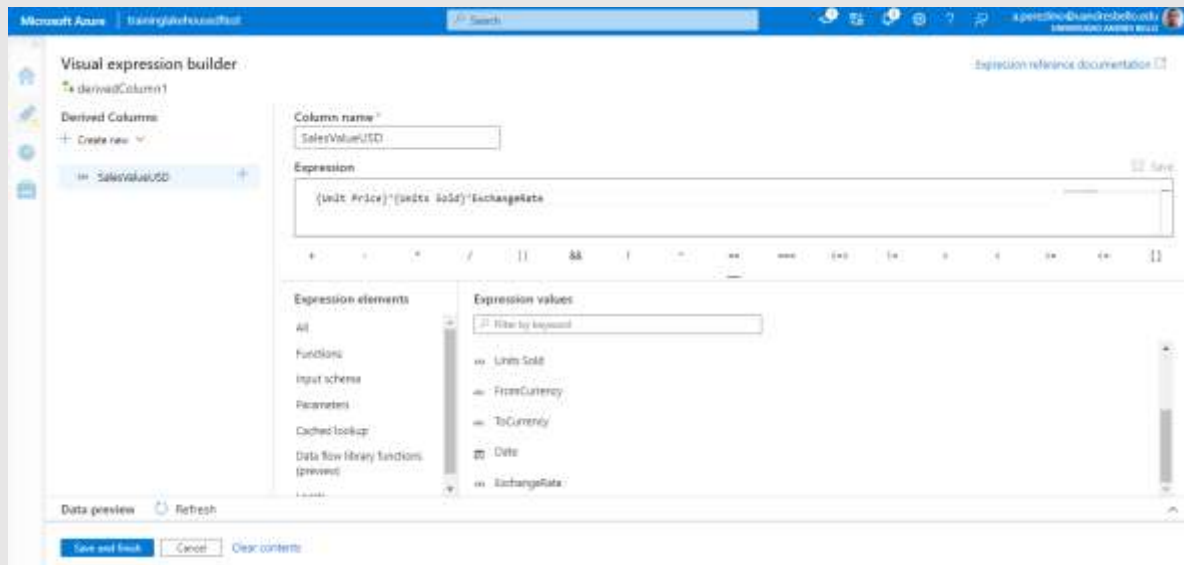
1. Conecte una transformación Derived Column a la transformación Lookup en el lienzo del data flow.

2. Puede añadir columnas en la pestaña Derived column's settings directamente o utilizando el expression builder. Por ahora, haga clic en Open expression builder (Abrir generador de expresiones).

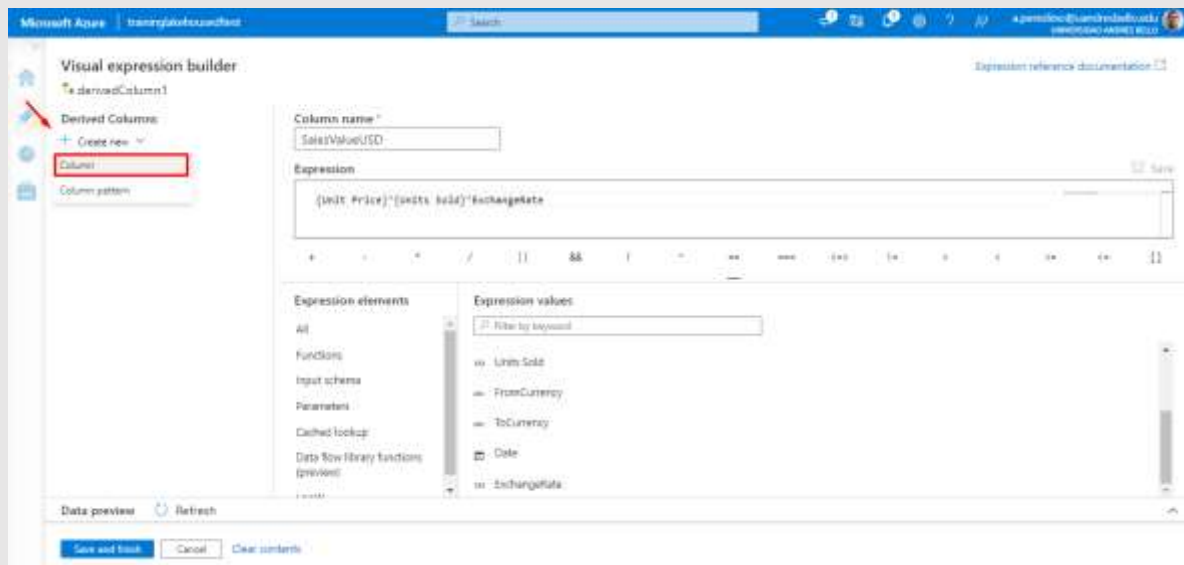


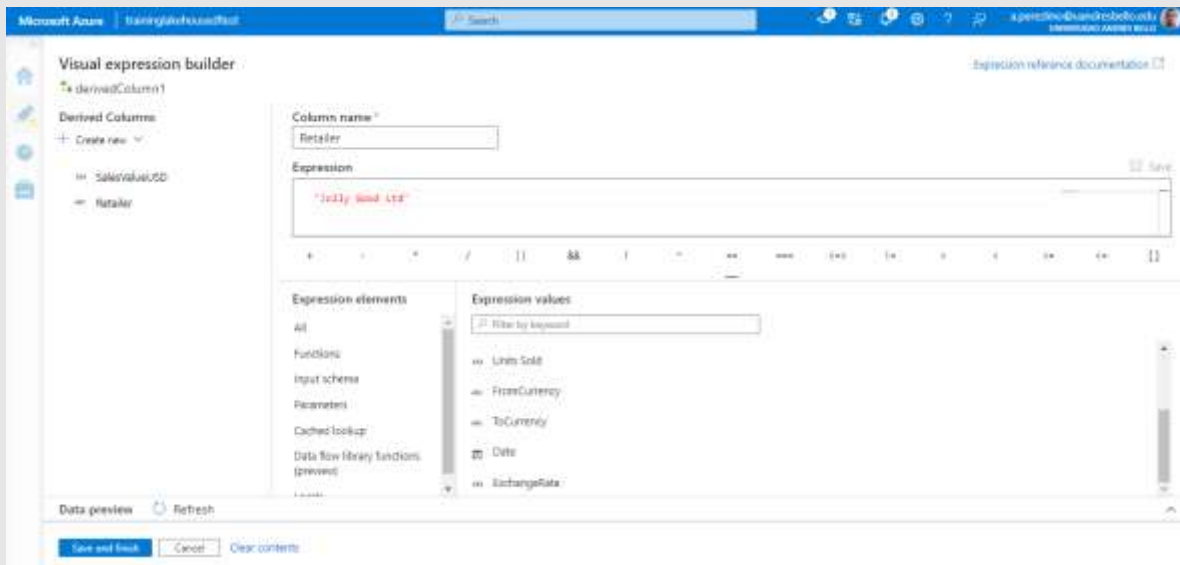
3. El Visual expression builder para la transformación Derived Column contiene características específicas de la transformación (además de las que vio para la transformación Filter) incluyendo un campo Column name y una barra lateral Derived Columns que enumera las columnas derivadas en esta transformación. Establezca el nombre de la columna como "SalesValueUSD".

4. El valor de total USD sales es el producto de unit price, el número de units sold, y el exchange rate. Utilice Input schema expression elements y el operador \* para construir esta expresión. La expresión completada aparece en la Figura 7-11.

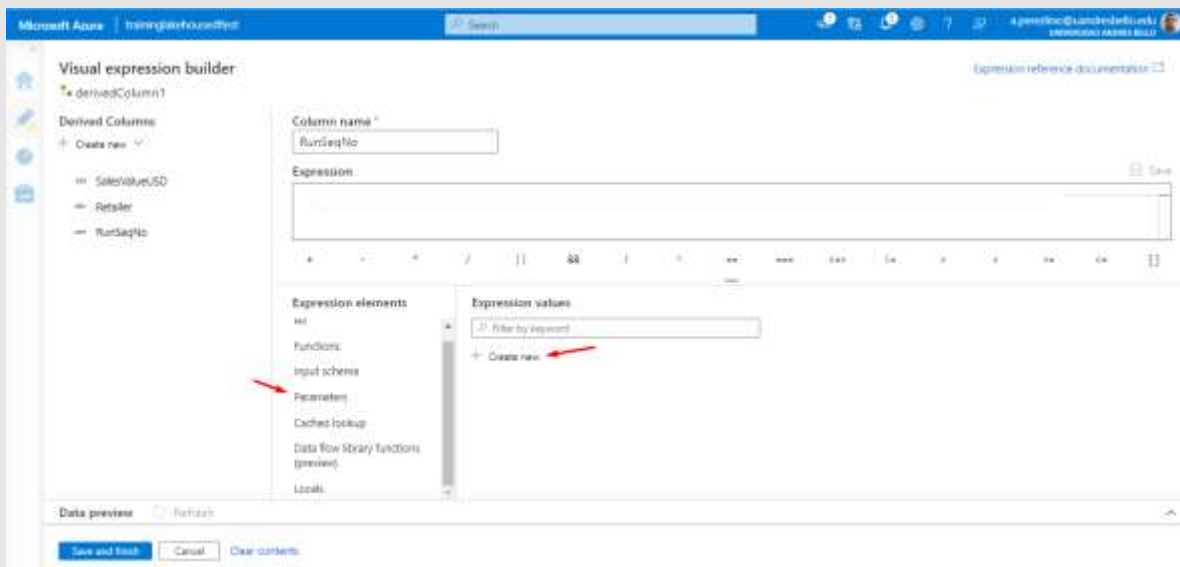


- En la barra lateral de Derived Columns, haga clic en el botón de menú + Create new y seleccione Column para añadir otra columna. Nombra la nueva columna "Retailer" y dale el valor literal de cadena 'Jolly Good Ltd'.

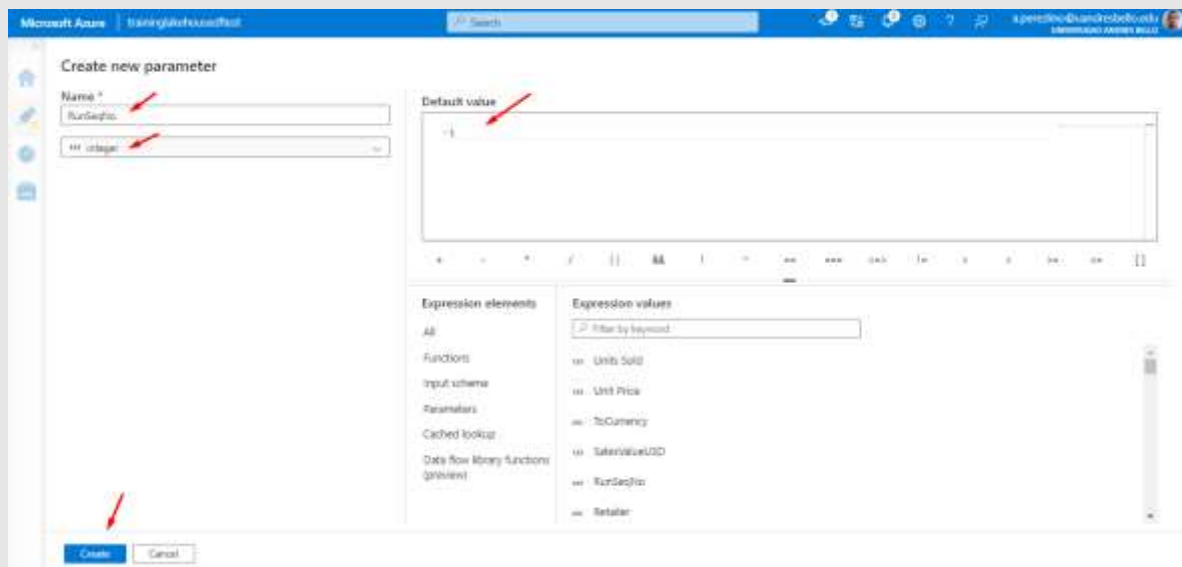




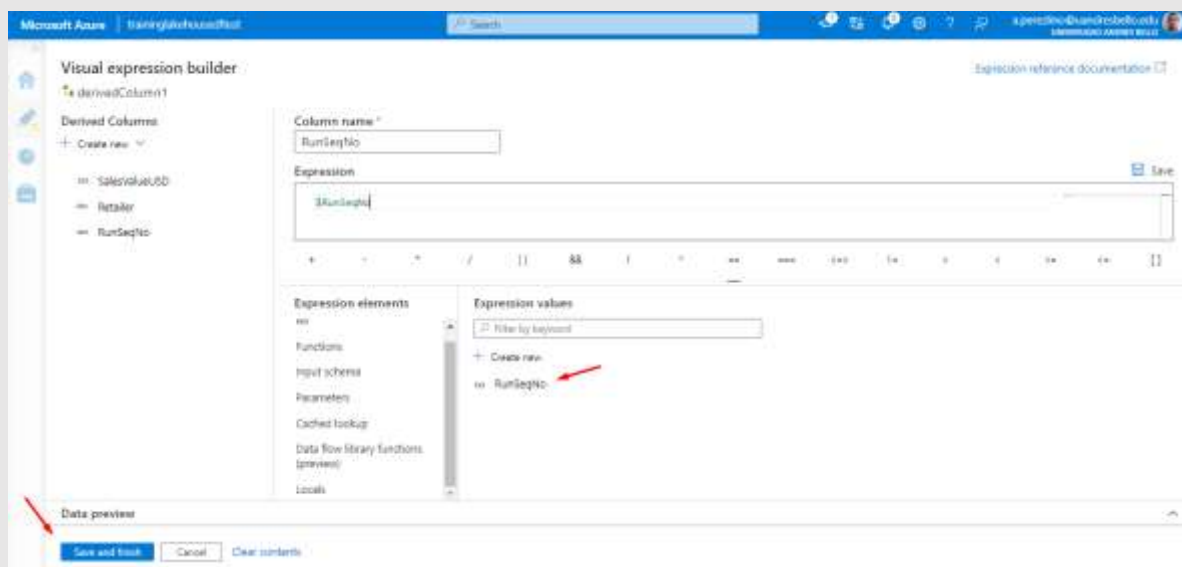
6. Añada una tercera columna de la misma manera, esta vez llamada "RunSeqNo" - esta columna contendrá el número de la secuencia de ejecución del pipeline. Su valor se obtendrá de la misma forma que antes, utilizando la actividad Lookup del pipeline ADF, y se pasará al data flow como parámetro.



7. Puede crear el parámetro "RunSeqNo" aquí mismo, en el constructor de expresiones, seleccionando Parameters en la lista Expression elements. Al hacer clic en + Create new bajo Expression values, la hoja Create new parameter se desliza sobre el constructor de expresiones. En la parte superior izquierda, nombre el parámetro "RunSeqNo" y establezca su tipo como "entero". En el panel Default value, proporcione un valor de -1. Haga clic en Create para crear el parámetro.



8. El constructor de expresiones se reanuda con el nuevo parámetro mostrado en la Expression values list. Haga clic en el nombre del parámetro para seleccionarlo como expresión de la columna derivada. Haga clic en Save (Guardar) y finalice.



La Figura 7-11 muestra la transformación de columna derivada configurada, incluyendo las expresiones de columna construidas en el generador de expresiones. Si desea editar alguna de las expresiones, puede hacer clic en el campo de la expresión para revelar un enlace del constructor de expresiones abierto (como se muestra en la figura para la columna "Retailer"), o simplemente puede editar la expresión in situ.



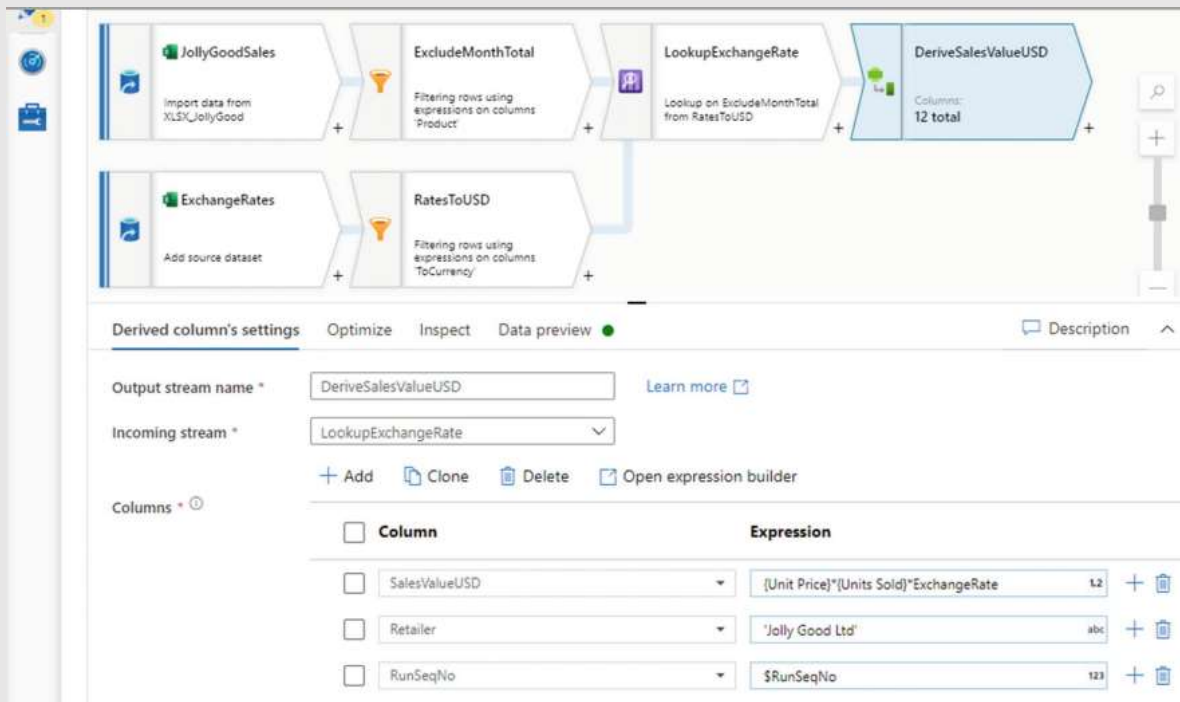
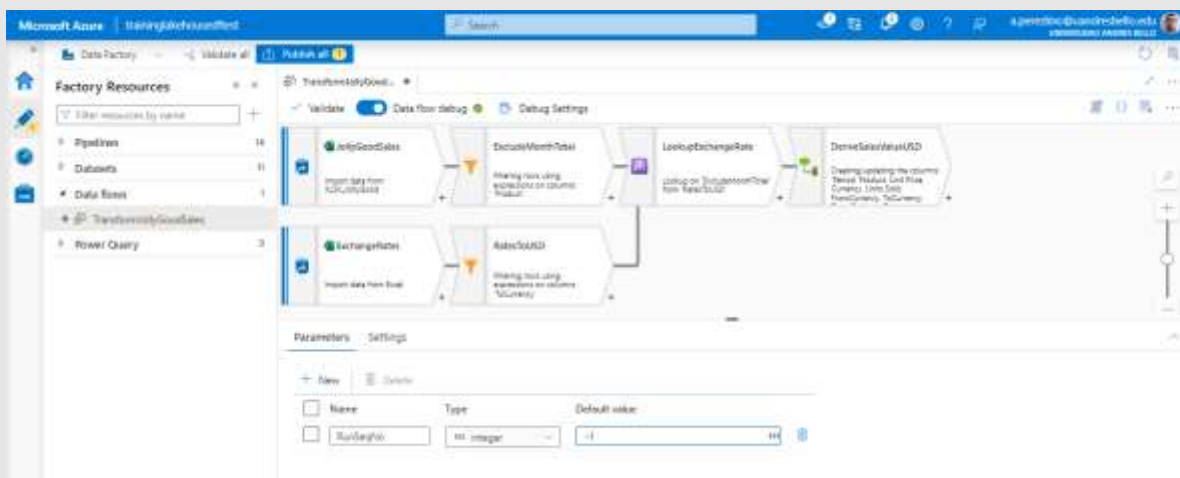


Figura 7-11 Configured Derived Column transformation

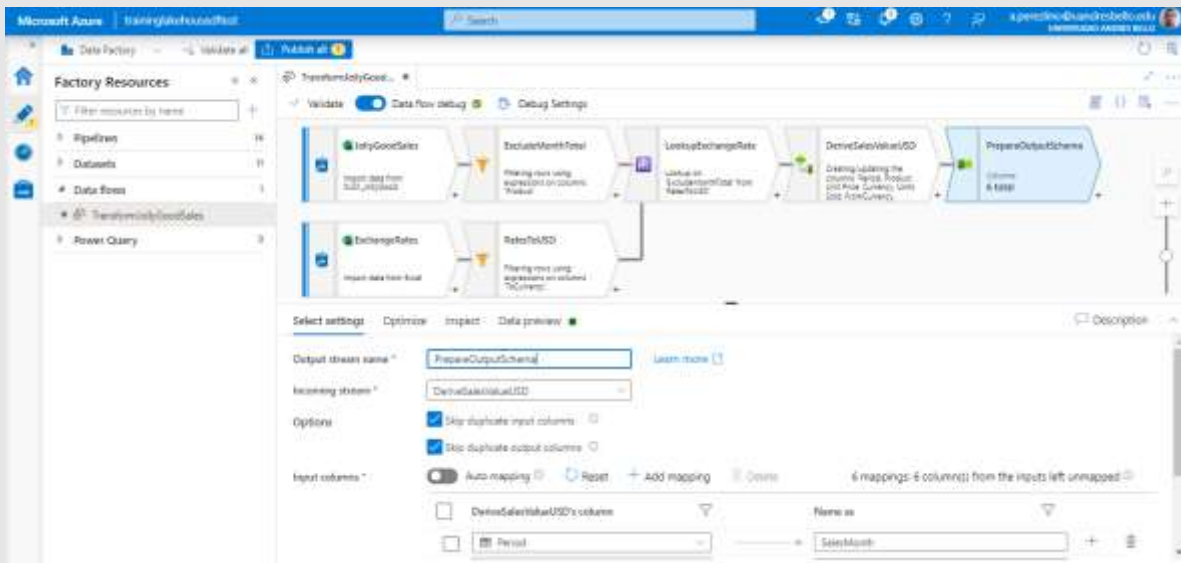
El parámetro "RunSeqNo" que creó mediante el generador de expresiones es un parámetro de entrada (input parameter) para el propio data flow. Haga clic en algún espacio en blanco del lienzo del data flow para ver las opciones de configuración en el nivel del data flow y, a continuación, seleccione la pestaña de configuración **Parameters** del flow para ver la definición del parámetro que ha creado. Si no es visible inmediatamente, cierre el data flow y vuelva a abrirlo para actualizar la interfaz de usuario.



### 7.1.6. Utilizar la transformación Select

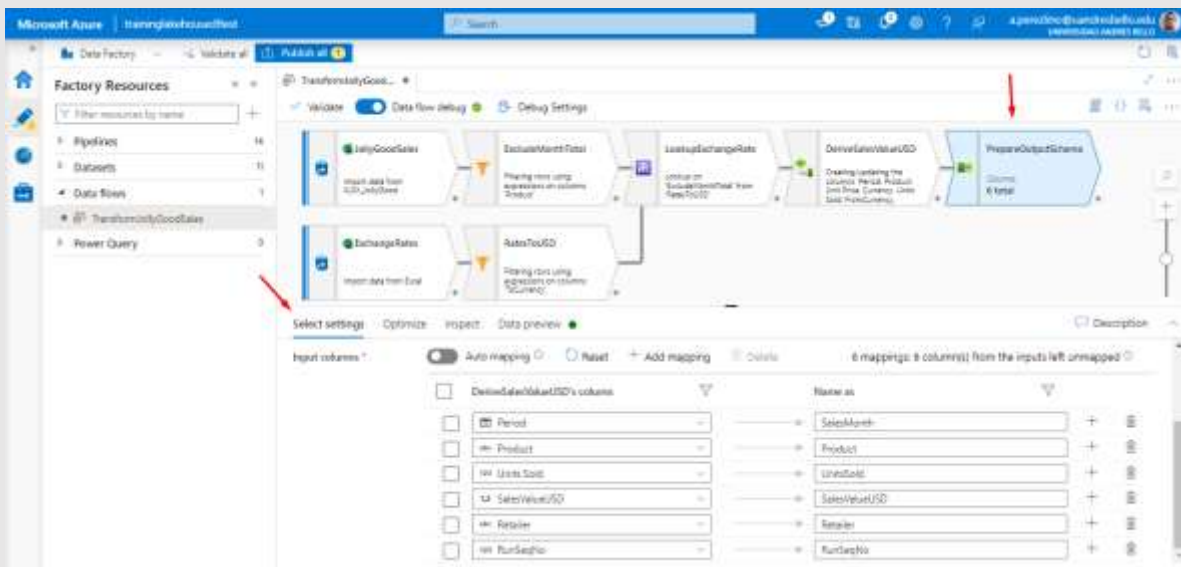
La transformación Select de los data flow permite cambiar el nombre de las columnas del data stream o eliminarlas. La utilizará aquí para alinear el esquema de salida (output schema) con el de la tabla de destino.

1. Conecte una transformación Select a la transformación Derived Column configurada en la sección anterior.

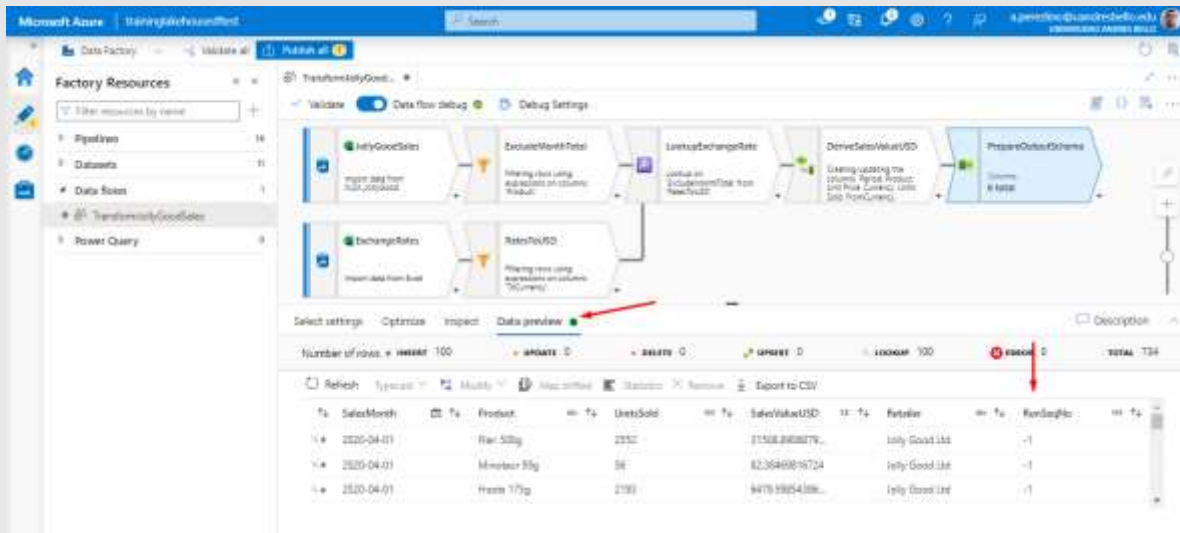


2. La pestaña Select settings contiene una lista de mapeos de columnas entre los esquemas de entrada y salida de la transformación. A la derecha de cada mapeo hay un botón de papelera Eliminar mapeo. Utilícelo para eliminar los mapeos de "Unit Price", "Currency", "FromCurrency", "ToCurrency", "Date" y "ExchangeRate". Esto elimina esas columnas del flujo de salida de la transformación.

3. Utilice el campo Name as (a la derecha de cada asignación) para cambiar el nombre de "Period" a "SalesMonth" y para eliminar el carácter de espacio del medio de "Units Sold".



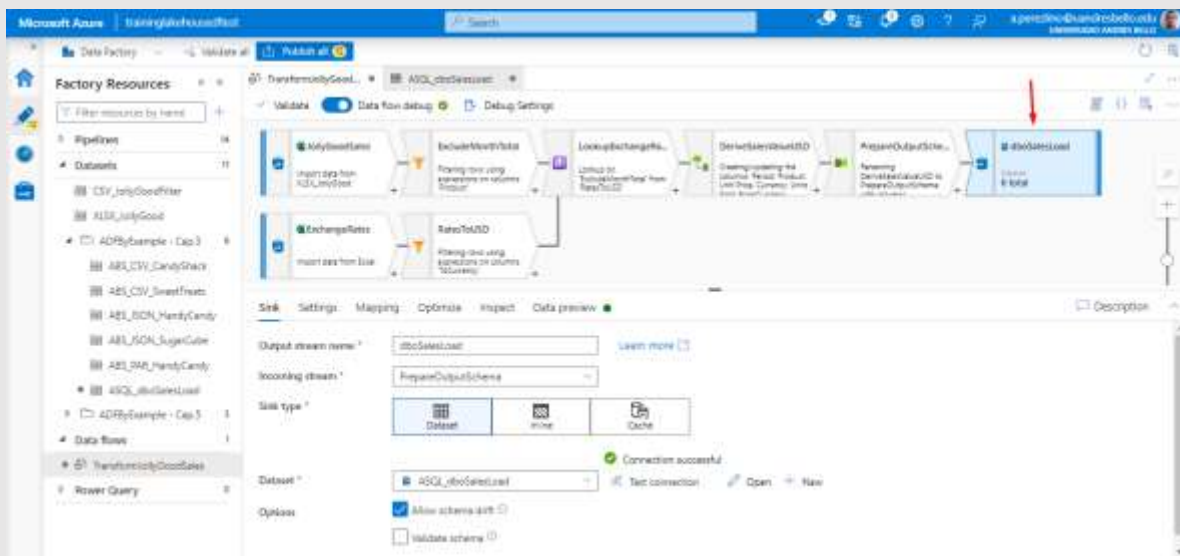
- Utilice la pestaña Data preview para verificar el efecto de la transformación. El valor de la columna "RunSeqNo" será -1, el valor por defecto de su parámetro "RunSeqNo". Este será reemplazado por un número de secuencia de ejecución real en tiempo de ejecución.



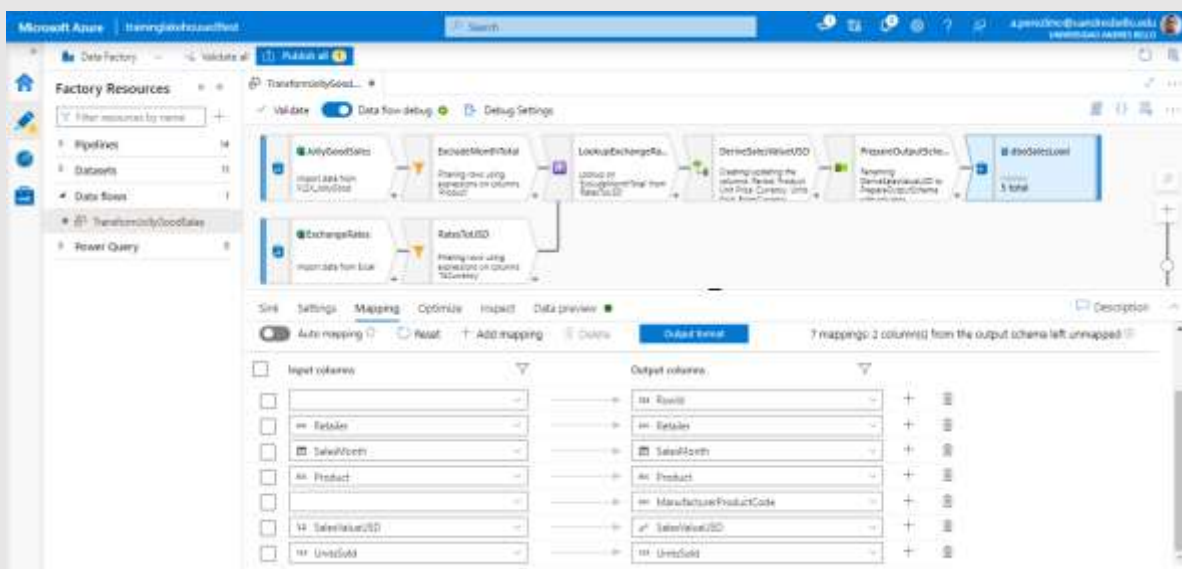
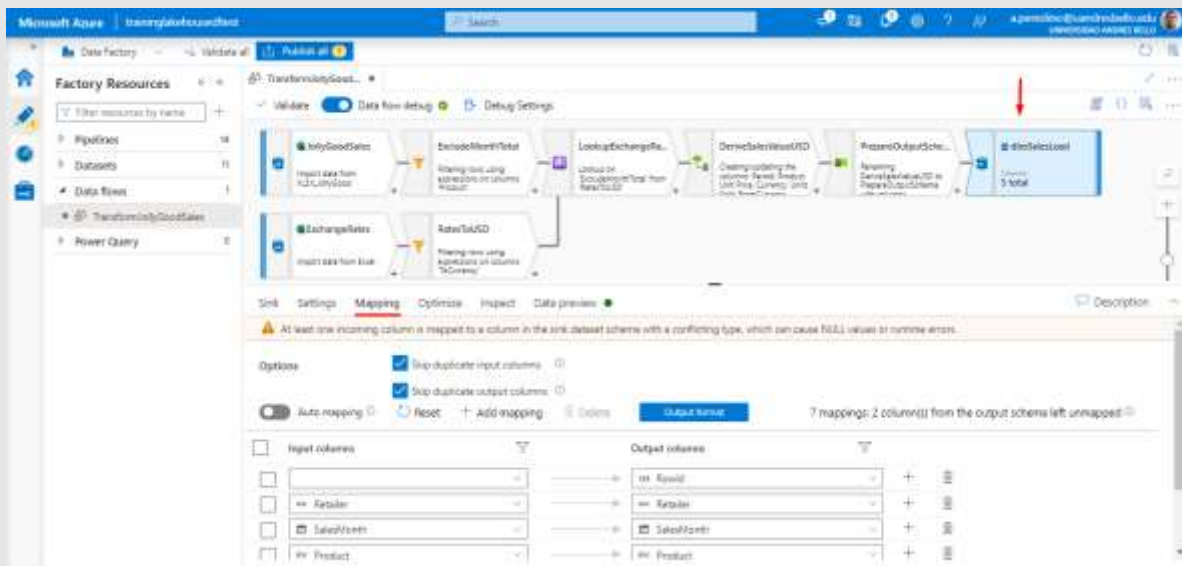
#### 7.1.7. Utilizar la transformación Sink

Una transformación Sink se utiliza para persistir las salidas del data flow en el external data storage. Un data flow válido requiere al menos una transformación Sink - puede guardar flujos incompletos mientras trabaja, pero no podrá ejecutarlos. En esta sección, añadirá una transformación Sink para escribir los datos de ventas de Jolly Good transformados en su tabla de base de datos [dbo].[Sales\_LOAD].

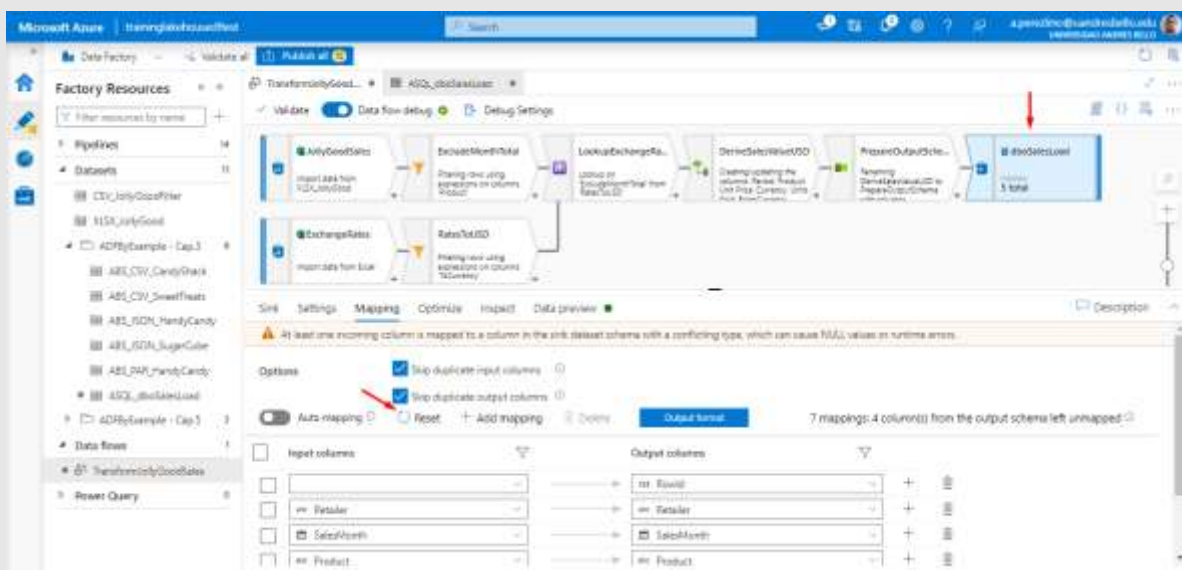
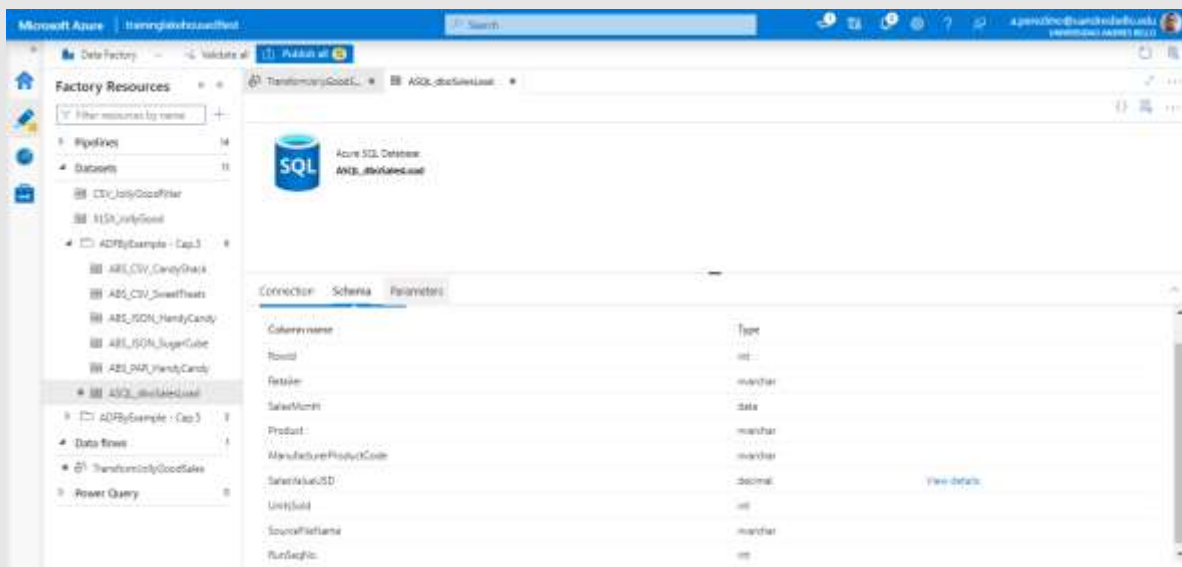
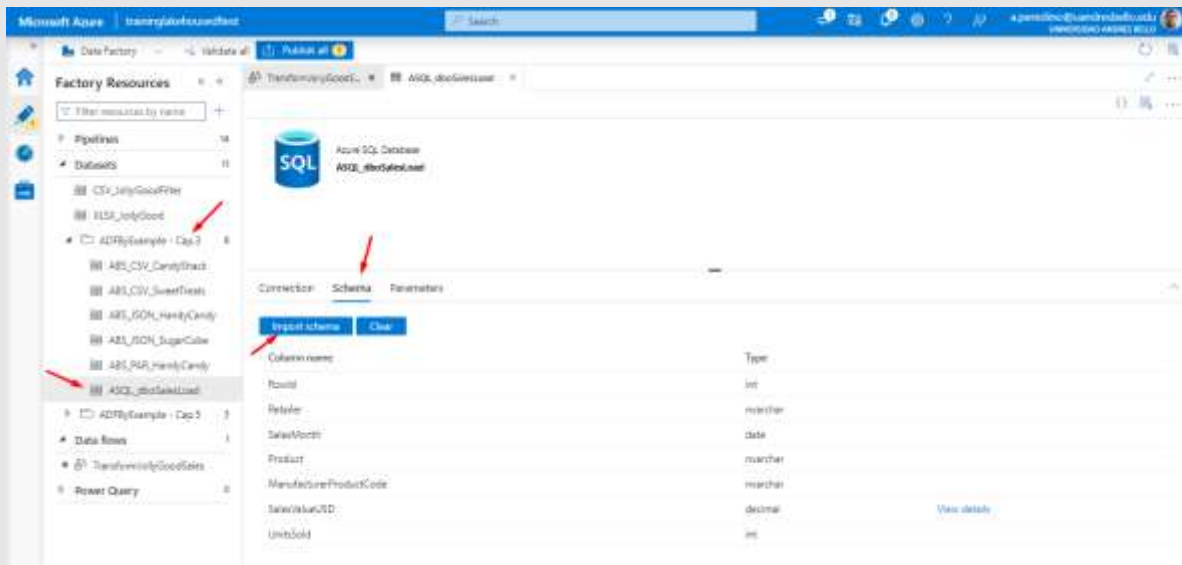
- Conecte una transformación Sink a la transformación Select que añadió anteriormente.
- En la pestaña Sink, asigne un nombre a la transformación y asegúrese de que el tipo de Sink esté configurado como "Dataset". Elija el dataset "ASQL\_dboSalesLoad" en el desplegable Dataset.



3. En la pestaña Mapping, desactive el mapeo automático utilizando el conmutador Auto mapping. Esto tiene el efecto de mostrar el conjunto existente de mapeos creados automáticamente.

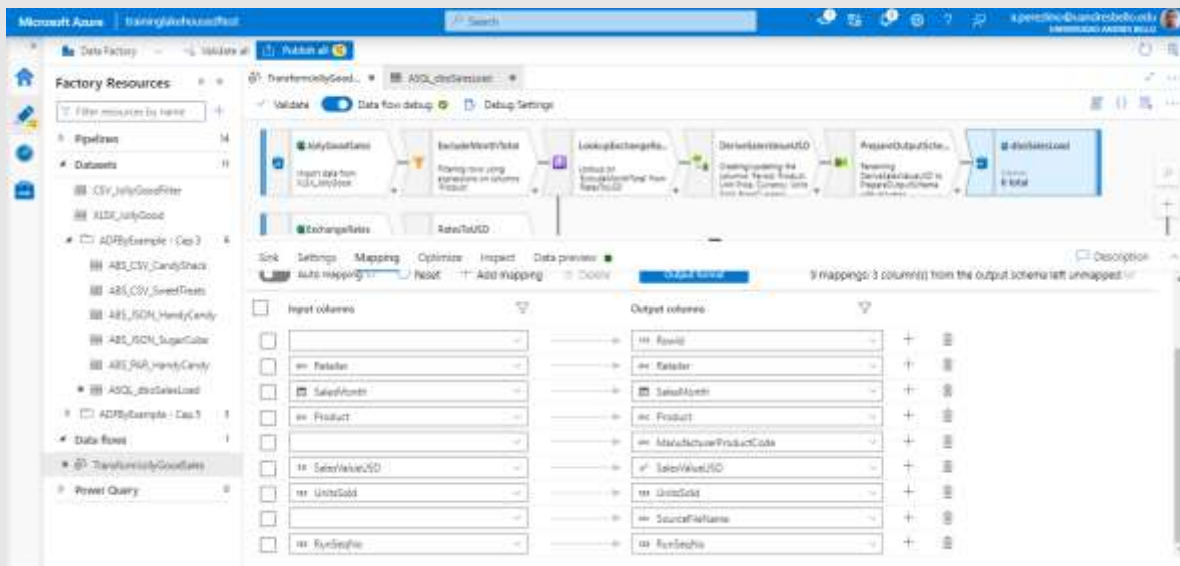


4. Es posible que las columnas de salida [RunSeqNo] y [SourceFileName] no aparezcan, ya que no estaban presentes en la tabla [dbo].[Sales\_LOAD] cuando se creó el conjunto de datos del ADF correspondiente. Para solucionarlo, vuelva a importar el esquema del dataset abriendo directamente el dataset - debe estar en su carpeta de datasets "Chapter3" - y luego utilice el botón Import schema en su pestaña Schema configuration. Vuelva a la pestaña de Mapping de la transformación Sink en el lienzo de data flow y haga clic en Reset para sincronizar la transformación con el dataset actualizado.

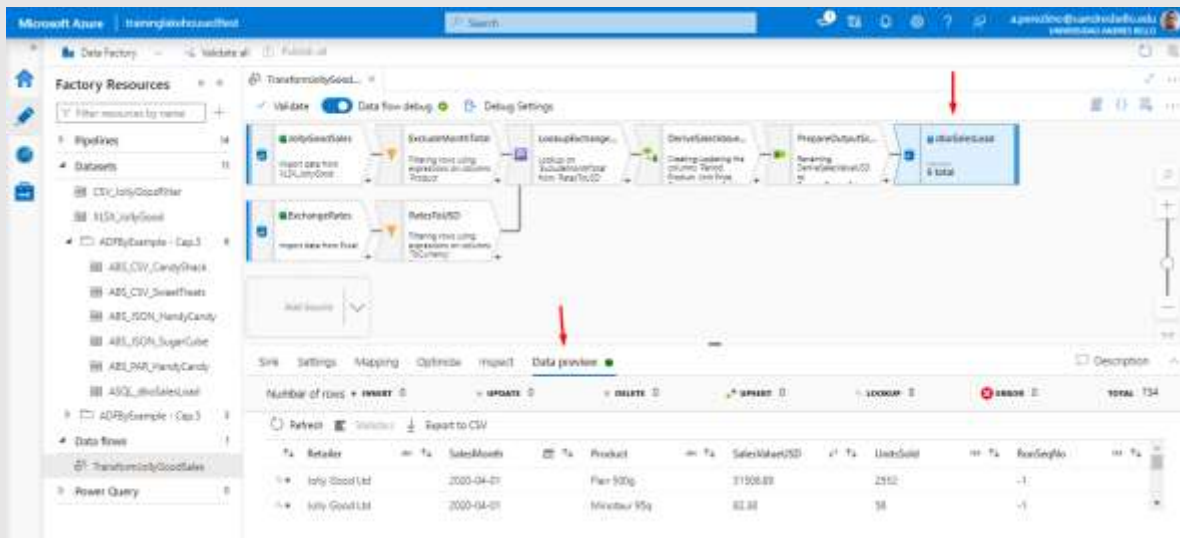




5. Compruebe que las seis columnas de entrada se han asignado correctamente. Las columnas de salida [RowId], [ManufacturerProductCode] y [SourceFileName] no tienen entradas correspondientes y pueden dejarse sin asignar.



6. Inspeccione la salida de la transformación utilizando la pestaña Data preview, luego guarde los cambios. La vista previa de datos en la transformación Sink indica los datos que se escribirían en el sink en tiempo de ejecución, pero no se escribe ningún dato realmente.



### 7.1.8. Ejecutar el Data Flow

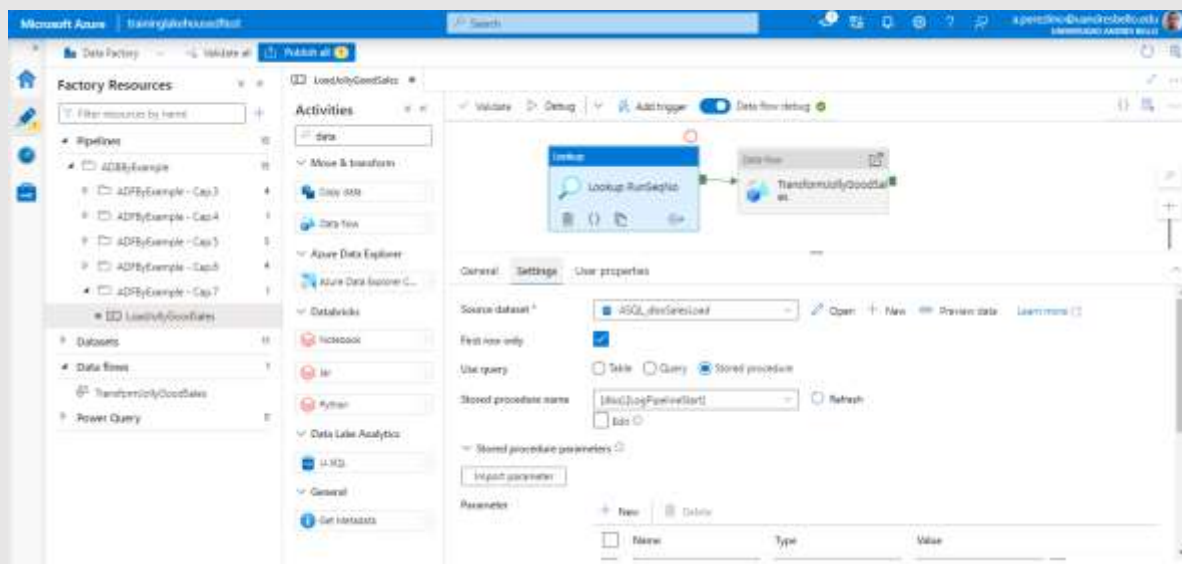
Con la adición de la transformación final de Sink, el data flow está listo para su ejecución. El flujo completo se muestra en la Figura 7-12.

Figura 7-12 Data flow completo para cargar los datos de ventas de Jolly Good Ltd

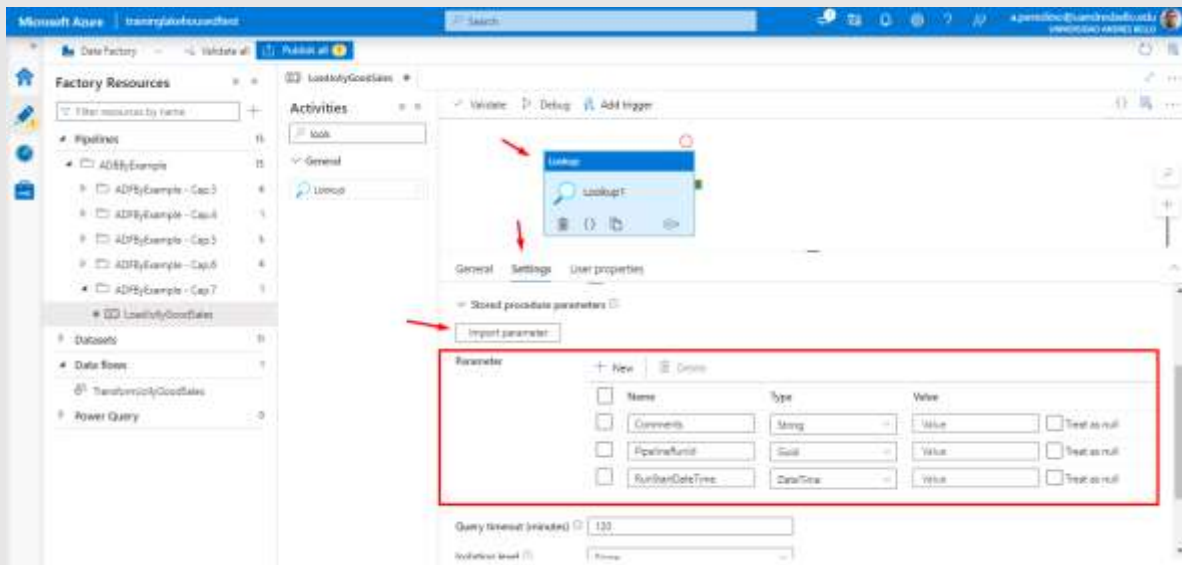
#### Creación de un pipeline para ejecutar el Data Flow

El data flow se ejecutará utilizando un pipeline de ADF similar a los pipelines que creó anteriormente. En este caso, en lugar de utilizar la actividad Copy data, se utilizará la actividad Data flow para mover los datos ejecutando el data flow.

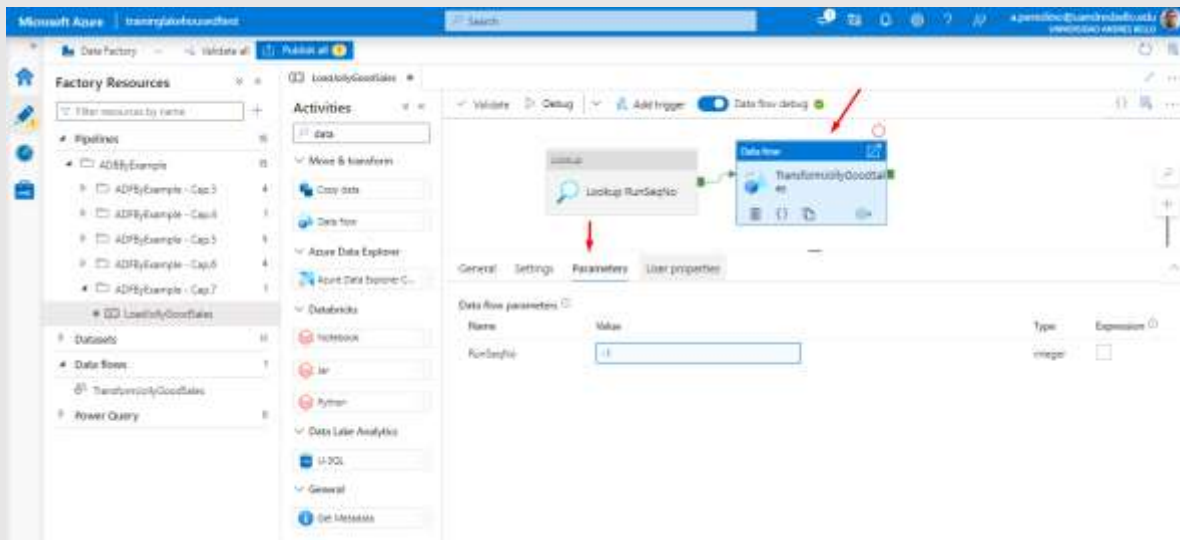
1. En el panel de Recursos de la Fábrica UX del ADF, cree una carpeta de pipelines "Chapter7", y luego cree un nuevo pipeline dentro de ella. Nombre el nuevo pipeline "LoadJollyGoodSales". Es posible que también desee mover su dataset de Excel a una carpeta para este capítulo.
2. Añada una actividad Lookup al pipeline. En su pestaña de configuración Settings, seleccione Source dataset "ASQL\_dboSalesLoad". Establezca Use query como "Stored procedure" y seleccione el Stored procedure name "[dbo].[LogPipelineStart]". Haga clic en Import parameter y, a continuación, proporcione las expresiones adecuadas para los tres parámetros del procedimiento almacenado.



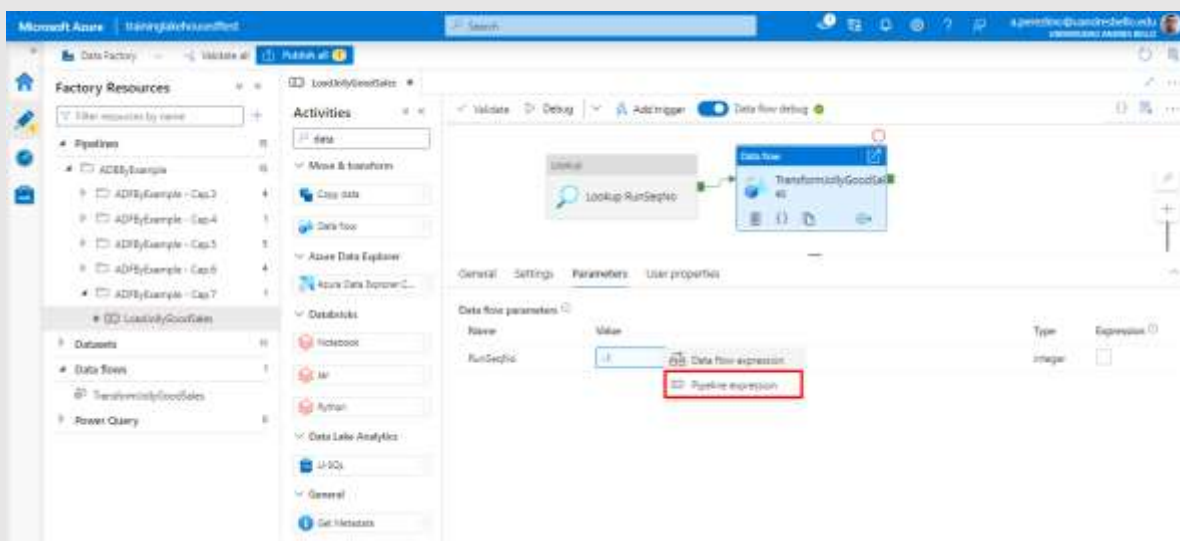




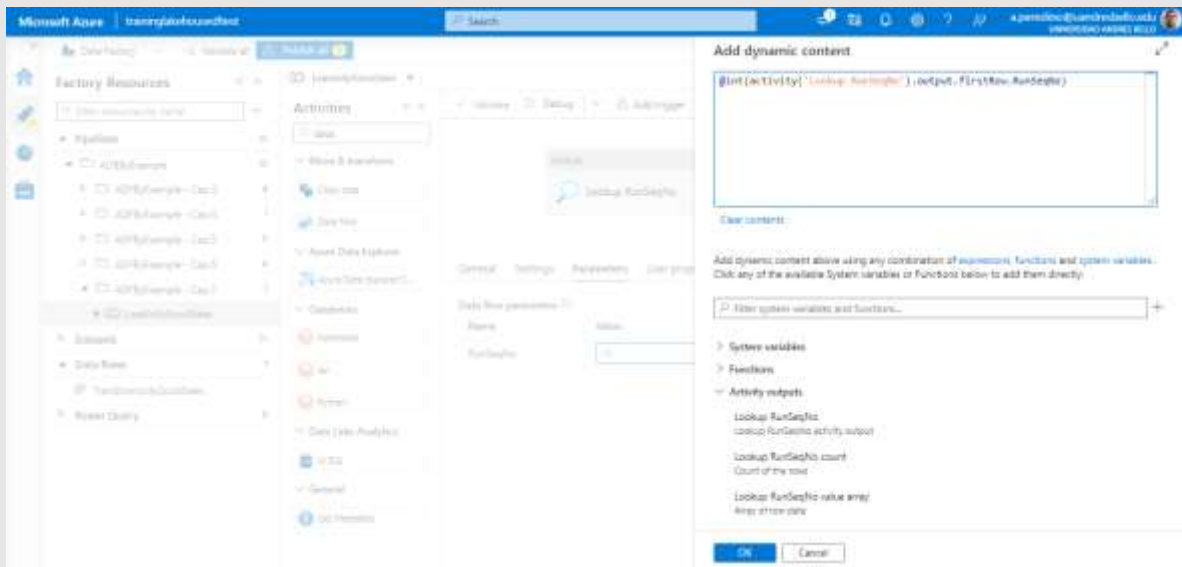
3. Arrastre una actividad de data flow desde el grupo Move & transform de la caja de herramientas de actividades al lienzo de creación. Conecte la nueva actividad como dependiente de la actividad Lookup.
4. En la pestaña Settings de la actividad, seleccione el data flow "TransformJollyGoodSales" del desplegable Data flow. En la pestaña Parameters, encontrará el parámetro "RunSeqNo" que creó para el data flow, con su valor por defecto de -1.



5. Los parámetros del data flow pueden especificarse utilizando el lenguaje de expresión del pipeline o el lenguaje de expresión del data flow. Haga clic en el campo VALUE para editar el valor del parámetro y, a continuación, seleccione Pipeline expression en la ventana emergente que aparece.



6. En el constructor de expresiones del pipeline, introduzca una expresión para devolver la propiedad firstRow.RunSeqNo de la salida de la actividad Lookup. Tenga cuidado de envolver la expresión en la función de conversión int, para evitar conflictos de tipo cuando la actividad intente iniciar el data flow.



```
@int(activity('Lookup RunSeqNo').output.firstRow.RunSeqNo)
```

7. Ejecute el pipeline de la forma habitual haciendo clic en Debug. Los pipelines que contienen actividades de data flow pueden ser ejecutados usando una sesión de depuración de data flow - el comportamiento por defecto cuando se hace clic en Debug - o usando un cluster de Databricks just-in-time (JIT), aprovisionado automáticamente cuando se inicia la ejecución del pipeline. Un menú desplegable a la derecha del botón de Debug le permite elegir entre estos enfoques.

Un clúster JIT tarda unos cinco minutos en ser aprovisionado. Incluso cuando se utiliza una sesión de depuración activa, el pipeline tardará unos instantes más en iniciarse de lo que está acostumbrado: hay un breve retraso mientras se adquieren los recursos informáticos del clúster de depuración.

Inspeccionar la salida de la ejecución

A medida que el pipeline se ejecuta, la información de la ejecución de la actividad se muestra en su pestaña de salida, como se muestra en la Figura 7-13.

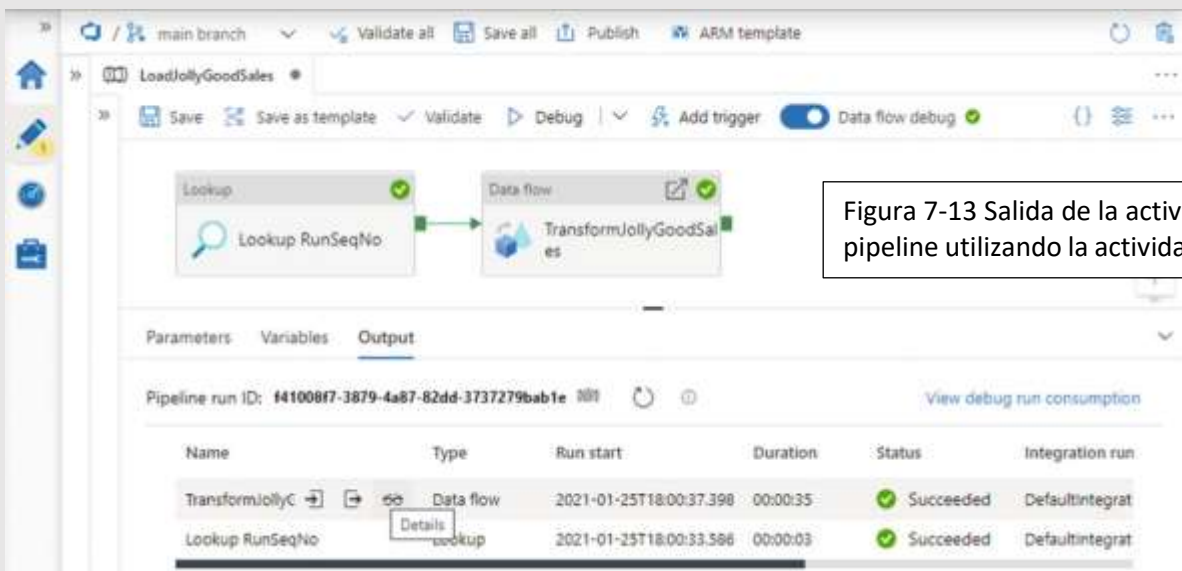
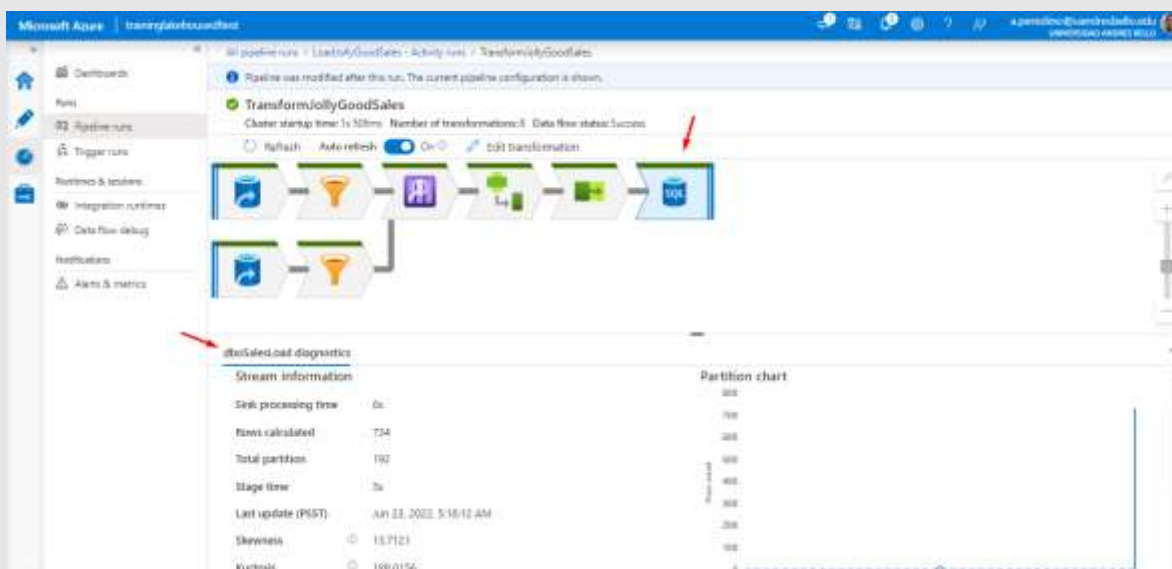
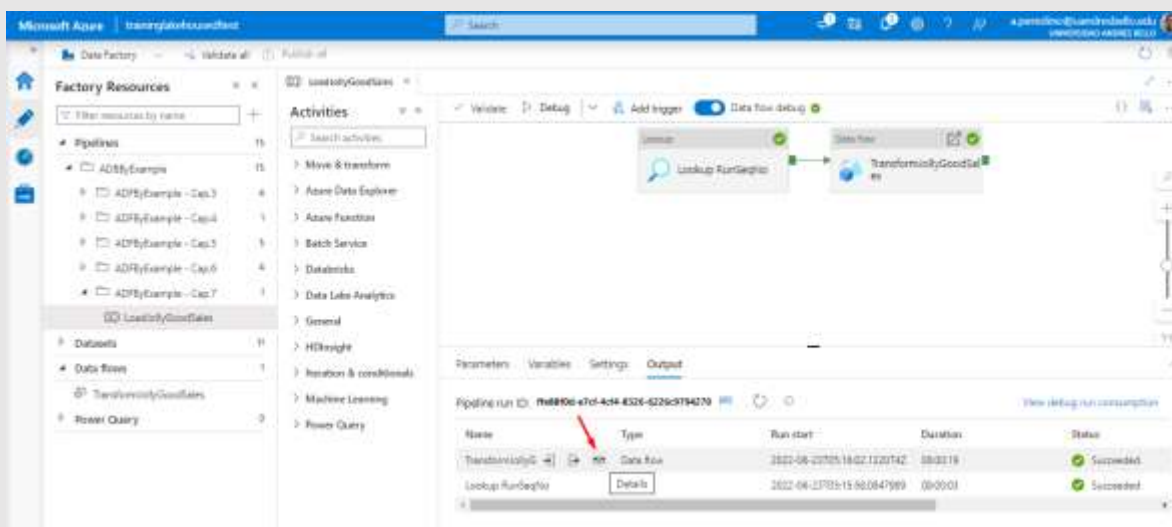


Figura 7-13 Salida de la actividad para el pipeline utilizando la actividad Data flow

El registro de ejecución de la actividad Data flow, al igual que el de la actividad Copiar datos, presenta un botón de detalles, visible en la Figura 7-13 a la derecha del botón de salida de la actividad. Haga clic en el icono Detalles para abrir una vista de supervisión gráfica del data flow.

La vista gráfica de supervisión (mostrada en la Figura 7-14) proporciona información detallada sobre el rendimiento del data flow. La mitad inferior de la figura desglosa el tiempo de ejecución del flujo en grupos de transformaciones que se ejecutaron conjuntamente. La mitad superior contiene una vista simplificada de la estructura del data flow, similar a la disposición en el lienzo del data flow. Al hacer clic en un nodo del lienzo se muestra la información de ejecución de esa transformación en una hoja a la derecha - la figura 7-14 muestra el resultado de seleccionar la transformación "LookupExchangeRate".



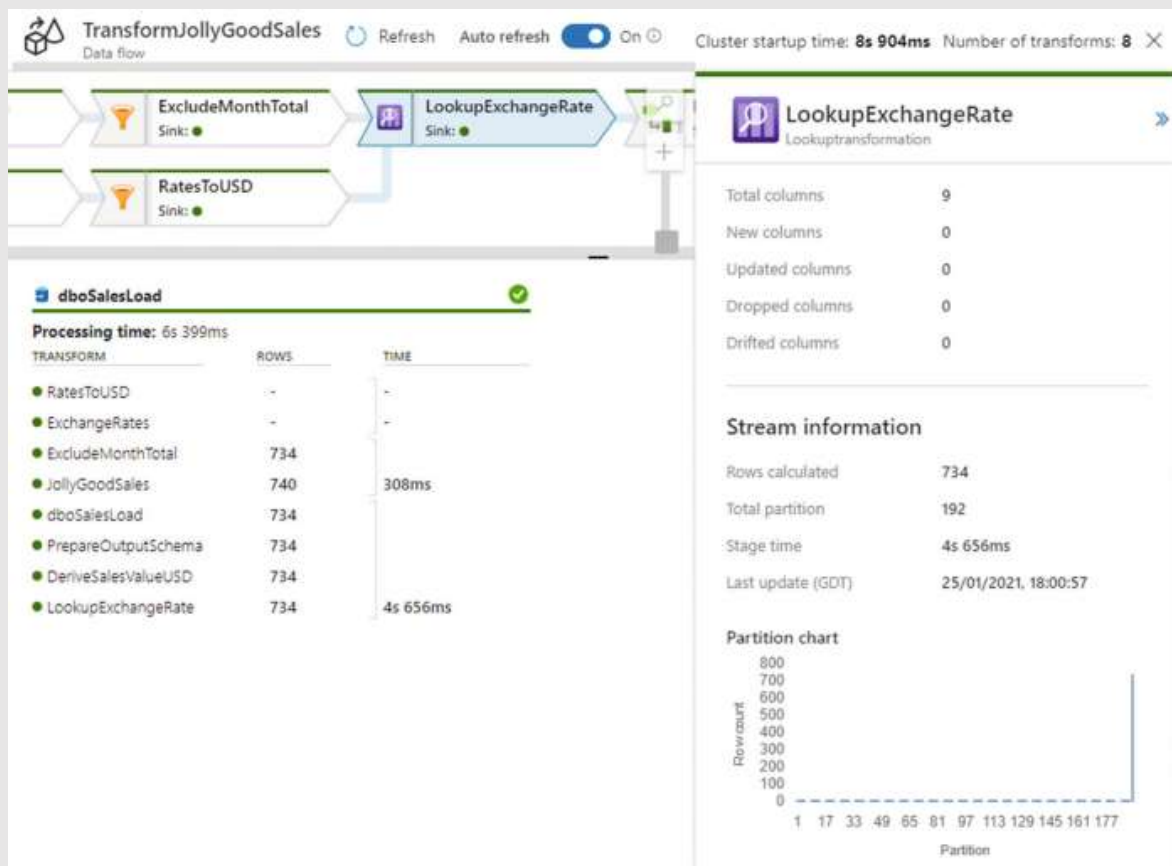


Figura 7-14 Vista de supervisión gráfica del data flow

La información presentada en la hoja es específica de la transformación seleccionada, pero a menudo incluye información sobre cómo se particionan los datos durante la ejecución. Cada transformación presenta una pestaña de configuración de optimización en el lienzo de data flow - puede utilizarla para realizar cambios en la forma de particionar los datos, pero en la mayoría de las situaciones se recomienda el comportamiento de partición por defecto.

#### Persist Loaded Data and Log Completion

Para persistir los datos cargados por la actividad de data flow, llame al procedimiento almacenado de la base de datos [dbo].[PersistLoadedSales].

1. Añada una actividad de Stored procedure a su pipeline, configurada para ejecutarse después de que la actividad Data flow haya finalizado con éxito. En su pestaña Settings, establezca su Linked service (servicio vinculado) a su base de datos Azure SQL vinculada.
2. Elija "[dbo].[PersistLoadedSales]" en el menú desplegable del nombre del Stored procedure, luego debajo de Stored procedure parameters, haga clic en el botón Import. Proporcione una expresión adecuada para el parámetro "RunSeqNo" del procedimiento almacenado (utilizando la salida de la actividad de búsqueda inicial de la canalización).

3. Añada una segunda actividad de procedimiento almacenado, también a continuación de la actividad de flujo de datos. Configure esta para llamar al procedimiento almacenado "[dbo].[LogPipelineEnd]", importando sus parámetros de la misma manera que antes. Configure las expresiones para "RunEndTime", "RunSeqNo" y "RunStatus".
4. Las rutas de propiedades de salida de la actividad Copiar datos que utilizó anteriormente para los parámetros "FilesRead", "RowsRead" y "RowsCopied" no son válidas para la actividad Data flow. Puede tratar estos campos como nulos o encontrar rutas alternativas en la salida de la actividad de Data flow. Los valores correspondientes a "RowsRead" y "RowsCopied" se pueden encontrar, pero es posible que no tenga otra opción que tratar "FilesRead" como nulo.
5. Vuelva a ejecutar el pipeline, verificando que los datos de ventas de Jolly Good han sido persistidos con éxito en la tabla de la base de datos [dbo].[Sales].

## 7.2. Actualizar una dimensión de producto

Los datos almacenados en la tabla [dbo].[Sales] no siempre son fáciles de analizar, porque la columna [Product] combina dos atributos. Cada valor incluye el peso del producto, por lo que agrupar diferentes formatos del mismo producto es difícil. Calcular el peso del producto vendido es difícil, porque los pesos de los productos no se almacenan numéricamente y son una mezcla de onzas y gramos.

Permitir que los registros individuales se agrupen y agreguen utilizando características compartidas es una aplicación clásica para una tabla de dimensiones. En esta sección, construirá un data flow para mantener una dimensión de producto para apoyar el análisis de las ventas de productos.

---

**Nota** A estas alturas, debería haber acumulado datos de ventas para varios minoristas en [dbo].[Sales]. Si no es así, utilice el pipeline "ImportSTFormatFiles" que desarrolló en el Capítulo 6 para recargar los datos de las ventas de Desserts4All y Naughty but Nice.

---

### 7.2.1. Crear una tabla de dimensión

El listado 7-1 proporciona el código SQL para crear una tabla de dimensión llamada [dbo].[Product]. No tiene una integer key, pero sería adecuada para su uso en un modelo tabular de SQL Server Analysis Services (SSAS) o para su uso en un análisis basado en SQL.

Utilice su cliente SQL para ejecutar el script y crear la tabla en su base de datos Azure SQL.

```
CREATE TABLE dbo.Product (  
    Product NVARCHAR(255) PRIMARY KEY  
    , ProductName NVARCHAR(255) NOT NULL  
    , WeightInOunces DECIMAL(19,2) NOT NULL  
    , WeightInGrams DECIMAL(19,2) NOT NULL  
);
```

Listado 7-1 Creación de una tabla [dbo].[Product]

### 7.2.2. Create Supporting Datasets

Si creó un dataset de Azure SQL Database parametrizado como se sugiere en el Capítulo 6, puede utilizarlo aquí en el data flow. Alternativamente, cree datasets ADF para cada una de las dos tablas que utilizará en esta sección:

1. Cree un dataset para representar la tabla existente [dbo].[Sales].
2. Cree un segundo dataset para representar la nueva tabla [dbo].[Product].



### 7.2.3. Construya el Data Flow de mantenimiento de productos

En esta sección, utilizará un data flow para leer los detalles de los productos de la tabla [dbo].[Sales], extraer los nombres y pesos de los productos y, a continuación, añadir esa información a [dbo].[Product].

1. Cree un nuevo data flow en su carpeta de data flow "Capítulo7" y nómbrelo "UpdateProduct".
2. Añada una Source transformation utilizando el dataset de la tabla [dbo].[Sales].

---

**Consejo** Cuando se utilizan datasets parametrizados, los valores de los parámetros en tiempo de ejecución son suministrados por la actividad Data flow del pipeline en ejecución. En tiempo de desarrollo, especifique los valores en el lienzo de la hoja de Debug Settings (Figura 7-15), al que se accede mediante el botón situado encima del data flow. El botón sólo es visible cuando el clúster de depuración se está ejecutando.

---

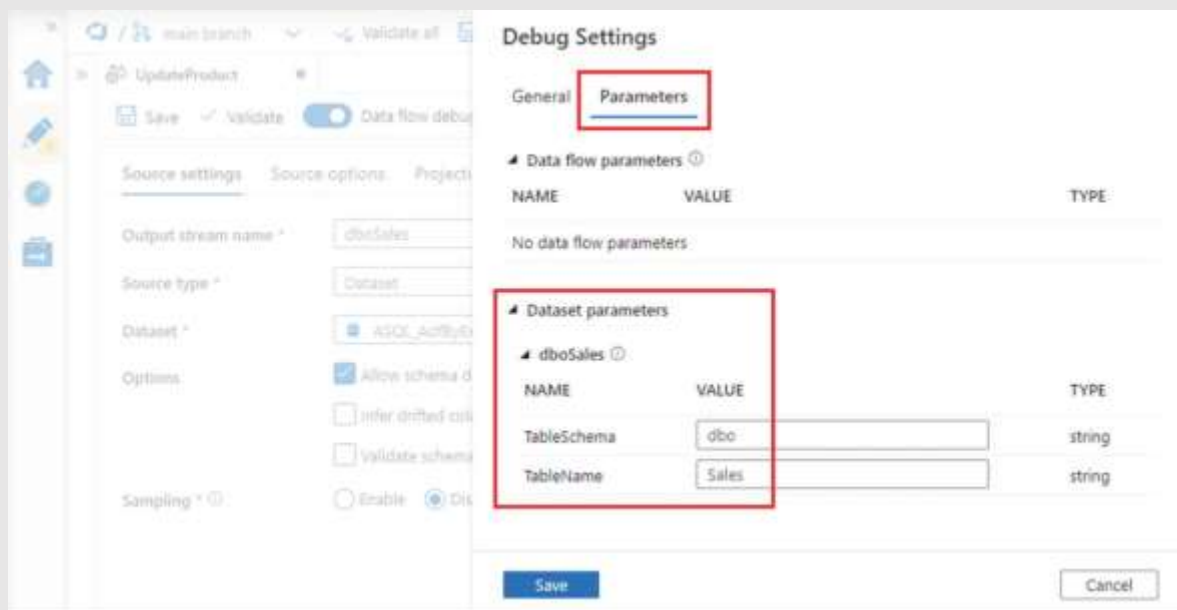


Figura 7-15 Utilizar Debug Settings para establecer los valores de los parámetros del dataset de desarrollo

3. Si está utilizando un dataset parametrizado, importe el esquema de la tabla utilizando el botón Import projection en la pestaña Projection de la transformación Source.
4. Conecte una transformación Derived Column a la salida de la transformación Source, luego abra el constructor de expresiones.

## Use Locals

En la última sección, vio cómo añadir nuevas columnas a un data flow utilizando la transformación Derived Column. A veces, es conveniente poder almacenar expresiones intermedias para reutilizarlas en definiciones de columnas derivadas, sin añadir sus resultados al data flow stream. Esto se puede conseguir mediante las expresiones locales.

El valor derivado por una expresión local puede ser utilizado por otras expresiones dentro de la misma transformación de columna derivada, pero no se incluye en la salida de la transformación. Las descripciones de productos almacenadas en [dbo].[Sales] contienen un nombre de producto y un peso, por ejemplo, "Chocolatey Nougat 3.52oz". Si se elimina la cadena de peso "3.52oz" de la descripción del producto, se obtiene el nombre del producto, y la misma cadena puede reutilizarse como una fuente más sencilla de pesos numéricos.

1. En la lista de Expression elements list del Visual expression builder, seleccione el elemento Locals. En Valores de expresión, haga clic en + Create new para abrir la hoja Create new local.
2. Llame al nuevo local "WeightString", y cree una expresión para devolver sólo esa parte de la descripción del producto. La cadena de peso es la parte de la descripción que sigue al carácter de espacio final - tenga en cuenta que algunas descripciones contienen varios espacios.
3. La figura 7-16 muestra la hoja local Create new que contiene una expresión adecuada. La expresión utiliza tres funciones: right, locate y reverse. Si pasa el ratón por encima de una función en la lista de valores de Expresión (como se muestra en la figura), o por encima de su nombre en el panel de Expresión, se muestra una información sobre herramientas que describe la función. Haga clic en Crear para guardar el nuevo local.

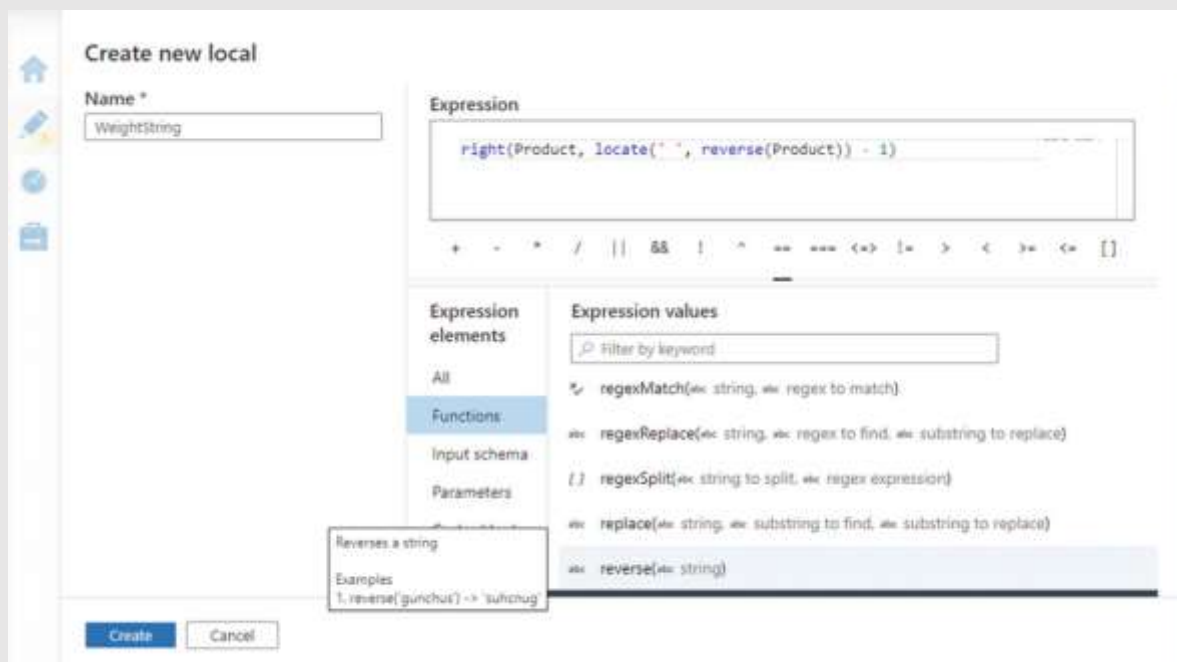


Figura 7-16 Creación de una expresión local en la transformación Derived Column

4. El nuevo local aparece en la lista de Expression values cuando se selecciona el elemento de expresión Locals - como se muestra en la Figura 7-17 - y puede ser seleccionado para su uso en una expresión como cualquier otro valor de expresión. Cree otro local, este llamado "WeightUnit", utilizando la expresión `iif(endsWith(Product, 'oz'), 'oz', 'g')`. La expresión devuelve la unidad de peso que se está utilizando en la descripción del producto.

5. De vuelta en el constructor de expresiones para columnas derivadas, nombre la nueva columna "ProductName". Utilice el local "WeightString" en una expresión que devuelva sólo el nombre del producto. La Figura 7-17 proporciona una expresión de este tipo - el elemento `:WeightString` es una referencia al local.

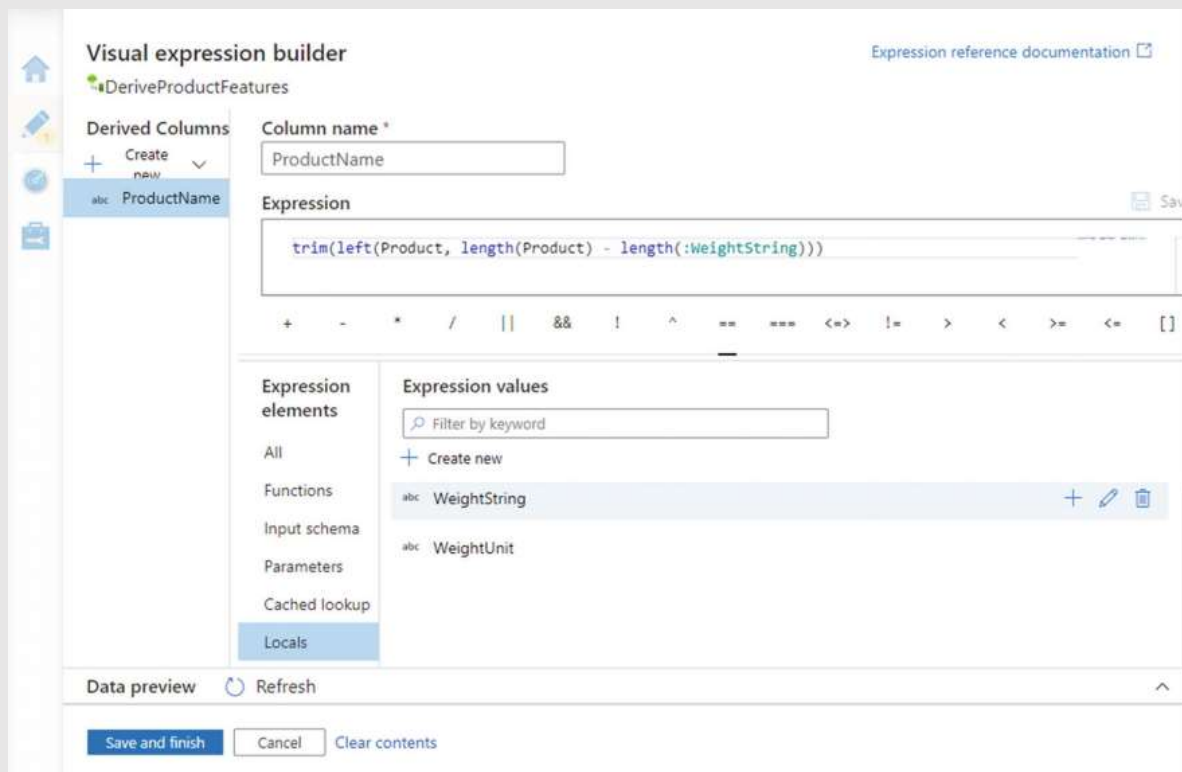


Figura 7-17 Uso de locales en una expresión de columna derivada

6. Añada una segunda columna derivada llamada "WeightInOunces" - esta debería extraer la parte numérica de la cadena de peso y convertirla de gramos si es necesario. Una posible expresión para conseguirlo es `toFloat(replace(:WeightString, :WeightUnit, '')) / iif(:WeightUnit=='g', 28.3495, 1.0)`.

7. Añada una tercera columna derivada llamada "WeightInGrams". Cree una expresión para realizar una función similar a la de "WeightInOunces", esta vez convirtiendo el peso dado de cada producto a gramos.

8. Utilice la pestaña Data preview para comprobar los resultados de la transformación. Verifique que sus columnas derivadas aparecen como se espera, y observe que sus expresiones locales no lo hacen.

Nota Los números resultantes de las conversiones de peso de los productos no se redondearán de forma ordenada a unos pocos decimales. Esto se debe a que los valores de entrada utilizados en las descripciones de los productos ya han sido redondeados.

### Use the Aggregate Transformation

Derived Column transformation's output stream contiene una fila por cada vez que aparece un producto en la tabla [dbo].[Sales]. Para hacer que el campo "Product" sea único - como lo requiere la clave primaria de [dbo].[Product] - debe eliminar los duplicados. Hágalo utilizando la transformación Aggregate.

1. Conecte una transformación Aggregate a la salida de la transformación Derived Column y nómbrela adecuadamente.
2. En la pestaña de configuración de Aggregate, asegúrese de que el conmutador que se encuentra debajo de Incoming stream (flujo de entrada) esté establecido en Group by (Agrupar por), y luego elija "Product" (Producto) en el menú desplegable de Columnas.
3. Cambie el interruptor a Aggregates. En Columna, seleccione "ProductName" en el desplegable. La función agregada requerida en la Expresión es first - puede abrir el constructor de expresiones para construir esto o simplemente introducir first(ProductName) directamente.
4. En muchos lugares, la UX del data flow soporta la especificación de múltiples columnas utilizando un patrón de Columna. Haga clic en el botón + Add encima de la lista Aggregates column y seleccione Add column pattern.
5. Un column pattern utiliza una expresión para identificar un conjunto de columnas. En Cada columna que coincida, introduzca la expresión startsWith(name, 'Weight'). Puede hacerlo directamente o utilizando el constructor de expresiones. La expresión identifica las dos columnas cuyos nombres empiezan por "Weight".
6. Debajo de column pattern expression hay campos para el nombre de la columna agregada y la expresión. Cuando se utiliza un patrón de columna, la cadena \$\$ es un marcador de posición para cada una de las columnas coincidentes - establezca el nombre de la columna como \$\$ y la expresión como first(\$\$) como se muestra en la Figura 7-18.

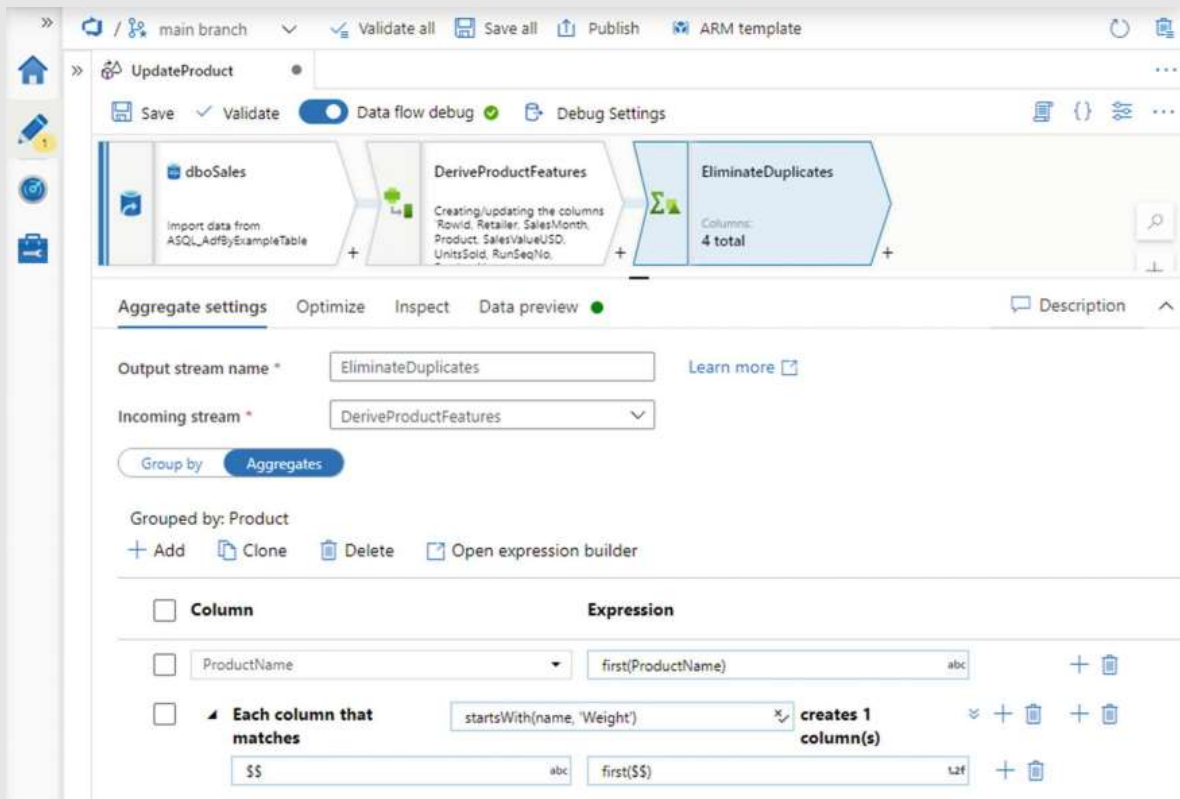


Figura 7-18 Uso de un patrón de columna para especificar múltiples columnas agregadas

7. Utilice la pestaña de vista previa de datos para comprobar los resultados de la transformación.

#### Utilice la transformación Exists

En este momento, la tabla [dbo].[Product] está vacía, pero las ejecuciones repetidas del data flow deben añadir sólo filas nuevas - intentar volver a añadir filas existentes provocará una violación de la clave primaria en la tabla de la base de datos, y el data flow fallará.

La transformación Exists permite filtrar valores en un data stream en función de si existen -o no- en otro data stream. Este data flow debe emitir filas sólo si no están ya presentes en la tabla [dbo].[Product].

1. Añada un segundo Source transformation al data flow, utilizando un dataset para la tabla [dbo].[Product]. Si utiliza un dataset parametrizado, recuerde importar el esquema de la tabla en la pestaña Projection de la transformación.
2. Conecte una transformación Exists a la transformación Aggregate. Establezca su flujo derecho al nuevo Source transformation [dbo].[Product] y su tipo Exist a "Doesn't exist".

3. Las expresiones utilizadas para definir la existencia se especifican en las condiciones Exists. Seleccione la columna "Product" tanto a la izquierda como a la derecha en la condición.

Utilice la pestaña Data preview para comprobar los resultados de la transformación si lo desea. Como la tabla [dbo].[Product] está actualmente vacía, los resultados serán los mismos que los devueltos por la vista previa de la transformación Aggregate.

Dé salida a los datos transformados con una transformación Sink que utilice el dataset de la tabla [dbo].[Product], comprobando que el auto-mapping de las columnas de salida es correcto. La Figura 7-19 muestra el data flow completado.



Figura 7-19 [dbo].[Product] maintenance data flow

#### 7.2.4. Execute the Dimension Data Flow

El data flow de mantenimiento de dimensiones ya está listo para ser ejecutado.

1. Cree un nuevo pipeline llamado "UpdateProductDimension" en su carpeta de pipelines "Chapter7".
2. Añada una actividad de data flow al lienzo, seleccionando su nuevo data flow. Si está utilizando un dataset parametrizado, la pestaña Settings configuration de la actividad se llenará con campos de entrada para los valores de los parámetros - rellene los campos con los valores correctos que se utilizarán cuando se ejecute el pipeline.
3. Ejecute el pipeline. Cuando se complete la ejecución, compruebe que la tabla [dbo].[Product] se ha rellenado correctamente.
4. La transformación Exists del data flow asegura que la transformación Sink sólo añade nuevos registros a [dbo].[Product]. Vuelva a ejecutar el pipeline para comprobar que es así.
5. Como prueba final, elimine algunos de los registros de [dbo].[Product] (pero no todos). Vuelva a ejecutar el pipeline para demostrar que los registros eliminados se restauran.

La tabla [dbo].[Product] poblada aquí es una tabla de estilo de dimensión simple destinada a soportar un análisis más flexible y ad hoc de los datos de la tabla [dbo].[Sales]. Por ejemplo, la consulta del Listado 7-2 devuelve el number, total value, y total kilogram weight de cada producto



vendido entre abril y septiembre de 2020 - este cálculo es posible porque se ha utilizado el data flow para convertir la descripción no estructurada del producto en atributos estructurados del mismo.

```
SELECT
    p.ProductName
, SUM(s.UnitsSold) AS UnitsSold
, SUM(s.SalesValueUSD) AS SalesValueUSD
, SUM(p.WeightInGrams * s.UnitsSold)/1000 AS KgSold
FROM dbo.Sales s
    INNER JOIN dbo.Product p ON p.Product = s.Product
GROUP BY p.ProductName
ORDER BY p.ProductName;
```

Listado 7-2 Análisis de ventas utilizando [dbo].[Product]

---

**Consejo** El data flow que ha creado aquí implementa un patrón básico de mantenimiento de Slowly Changing Dimension (SCD), en este caso, para un SCD de tipo 0. Las plantillas de ADF UX proporcionan patrones de pipeline y data flow reutilizables y están disponibles en la galería de plantillas de ADF. Para acceder a la galería, haga clic en la burbuja Create pipeline a partir de una plantilla en la página de resumen de ADF UX Data Factory (icono de inicio en la barra lateral de navegación).

---

## Revisión del capítulo

En los capítulos 3 a 6 se utilizó la actividad Copiar datos para mover los datos entre los conjuntos de datos de origen y los de destino sin mucha capacidad para modificar los datos en vuelo. Introducidos en este capítulo, los data flows de ADF cierran esa brecha, presentando poderosas capacidades de Databricks en una interfaz gráfica de usuario fácil de usar.

El requisito de un clúster de Databricks puede parecer oneroso dado el tiempo necesario para poner en marcha uno bajo demanda - un clúster JIT tarda alrededor de cinco minutos en ser aprovisionado, ampliando el tiempo de ejecución de los pipelines de ADF que hacen uso de las actividades de data flow. En el entorno publicado, los clústeres de Databricks se crean siempre justo a tiempo, cuando comienza la ejecución de una actividad de Data flow.

La recompensa es que Azure Data Factory es capaz de transformar incluso conjuntos de datos extremadamente grandes de forma eficiente, ahorrando tiempo y escalando bien al distribuir el procesamiento de los conjuntos de datos entre los nodos del clúster. Además, el aprovisionamiento y desmantelamiento automático del clúster reduce los costes operativos del mismo, permitiéndole pagar sólo por lo que utiliza.

## Conceptos clave

Los conceptos que se encuentran en este capítulo incluyen

- ❖ **Apache Spark**: Motor de procesamiento de datos de código abierto que distribuye automáticamente las cargas de trabajo de procesamiento en un clúster de servidores (denominados nodos) para permitir una ejecución altamente paralela.
- ❖ **Databricks**: Plataforma de procesamiento y análisis de datos construida sobre Apache Spark y que añade una variedad de características empresariales.
- ❖ **Data flows**: La herramienta de transformación visual de datos de ADF, construida sobre Azure Databricks.
- ❖ **Data flow debug**: El modo debug de data flow aprovisiona un cluster de Databricks en el que se pueden ejecutar data flows desde la UX de ADF.
- ❖ **Tiempo de vida (TTL)**: El clúster de depuración de data flow tiene un TTL por defecto de una hora, después del cual - si no se está utilizando - se apaga automáticamente.
- ❖ **Actividad de data flow**: Actividad del pipeline del ADF utilizada para ejecutar un data flow.
- ❖ **Parámetros**: Los parámetros del flujo de datos se especifican en la Configuración de depuración durante el desarrollo y se sustituyen por los valores suministrados por la actividad de flujo de datos que llama en tiempo de ejecución.
- ❖ **Data flow canvas**: Entorno de desarrollo visual para los data flows.

- ❖ **Transformación:** Un data flow está formado por una secuencia de transformaciones conectadas, cada una de las cuales modifica un flujo de datos de alguna manera.
- ❖ **Output stream name:** Nombre que identifica de forma exclusiva cada transformación de un data flow.
- ❖ **Pestaña Inspect:** Utilice la pestaña Inspeccionar de una transformación para ver la información del esquema de entrada y salida.
- ❖ **Pestaña Data preview:** Utilice la pestaña Vista previa de datos de una transformación para obtener una vista previa de los datos emitidos por la transformación. El uso de la vista previa de datos requiere que la depuración del flujo de datos esté activada y puede utilizarse para retrasar el tiempo de espera del clúster.
- ❖ **Pestaña Optimize:** Utilice la pestaña Optimizar de una transformación para influir en el particionamiento de datos en Spark cuando se ejecuta la transformación.
- ❖ **Source transformation:** Lee los datos de entrada de una fuente externa. Cada data flow comienza con una o más transformaciones de origen.
- ❖ **Sink transformation:** Escribe los datos transformados a una fuente externa. Cada data flow termina con una o más transformaciones Sink.
- ❖ **Lenguaje de expresión de data flow:** Las expresiones de data flow tienen su propio lenguaje y constructor de expresiones, diferentes a los de las expresiones de ADF pipeline.
- ❖ **Data Flow Script:** Lenguaje en el que se almacenan las transformaciones del flujo de datos, incrustado en el archivo JSON de un flujo de datos.
- ❖ **Column patterns:** Cuando se admiten, se utilizan patrones de columnas para especificar varias columnas que deben manejarse de la misma manera. Las columnas se especifican mediante expresiones de data flow para que coincidan con los metadatos de la columna.
- ❖ **Filter transformation:** Selecciona filas de su flujo de datos de entrada para incluirlas en su flujo de salida, sobre la base de criterios especificados como una expresión de flujo de datos. Las demás filas se descartan.
- ❖ **Lookup transformation:** Conceptualmente similar a una unión SQL entre dos flujos de datos. Admite diversos estilos y criterios de unión.
- ❖ **Derived Column transformation:** Utiliza expresiones de flujo de datos para derivar nuevas columnas para su inclusión en un data flow.
- ❖ **Locals:** Derivaciones intermedias con nombre en una transformación de columna derivada. Se utilizan para simplificar las expresiones y eliminar la redundancia.

- ❖ **Select transformation**: Se utiliza para cambiar el nombre de las columnas en un data flow o para eliminarlas.
- ❖ **Aggregate transformation**: Agrega una o más columnas en un data flow, opcionalmente agrupando por otras columnas especificadas.
- ❖ **Exists transformation**: Selecciona filas de su flujo de datos de entrada para incluirlas en su flujo de salida, basándose en la existencia (o no) de filas coincidentes en un segundo flujo de datos. Las demás filas se descartan.
- ❖ **Templates** (plantillas): Implementaciones reutilizables de patrones comunes de pipeline y data flow.
- ❖ **Template gallery**: Fuente de plantillas proporcionadas, a las que se accede mediante la burbuja Crear canalización a partir de una plantilla en la página de descripción general de Data Factory.