



Azure Data Factory by Example

Practical Implementation for
Data Engineers

—
Richard Swinbank

Apress®

Contenido

4. Expresiones	3
4.1. Explorar el Constructor de Expresiones	3
4.2. Utilizar las variables del sistema	5
4.2.1. Habilitar el almacenamiento de información de auditoría	5
4.2.2. Crear un nuevo pipeline	6
4.2.2. Añadir nuevas columnas de origen	6
4.2.3. Ejecutar el pipeline.....	7
4.3. Acceso a las propiedades de ejecución de la actividad	11
4.3.1. Crear objetos de base de datos.....	11
4.3.2. Añadir la actividad de procedimiento almacenado	13
4.3.3. Ejecutar el pipeline.....	16
4.4. Utilizar la actividad de Lookup	18
4.4.1. Crear objetos de la base de datos	18
4.4.2. Configurar la Actividad de Lookup	20
4.4.3. Utilizar Breakpoints (Puntos de interrupción)	23
4.4.4. Utilizar el valor de Lookup.....	24
4.4.5. Actualice la actividad del procedimiento almacenado	26
4.4.6. Ejecutar el pipeline.....	29
4.5. Variables de usuario	32
4.5.1. Crear una variable	32
4.5.2. Establecer una variable	33
4.5.3. Utilizar la variable.....	34
4.5.4. Array Variables	40
4.6. Concatenar Strings	41
4.6.1. Infix Operators.....	44
4.6.2. String Interpolation	44
4.6.3. Escapar @	45
4.7. Repaso del capítulo	46
4.7.1. Conceptos clave	46
4.7.2. Para los desarrolladores de SSIS	48

4. Expresiones

Los pipelines que has creado en el capítulo 3 tienen al menos una cosa en común: los valores de todas sus propiedades son estáticos, es decir, se determinan en tiempo de desarrollo. En muchos lugares, Azure Data Factory admite el uso de valores de propiedad dinámicos -determinados en tiempo de ejecución- mediante el uso de expresiones.

El enfoque principal de este capítulo es introducirte a las expresiones, pero también comenzarás a encontrar más actividades del pipeline. La actividad Copiar datos es lo suficientemente amplia e importante como para haber merecido un capítulo para sí misma, pero a partir de ahora empezará a hacer uso de una mayor variedad de actividades que se encuentran en la caja de herramientas de actividades.

4.1. Explorar el Constructor de Expresiones

Las expresiones pueden utilizarse en la UX del ADF siempre que vea el texto Añadir contenido dinámico [Alt+Mayúsculas+D] - es posible que ya haya notado que este mensaje aparece inmediatamente debajo del texto u otros campos de entrada cuando los edita. La Figura 4-1 muestra el campo Ruta de la carpeta comodín (en la pestaña Origen de la actividad Copiar datos) con el foco para que el mensaje sea visible.

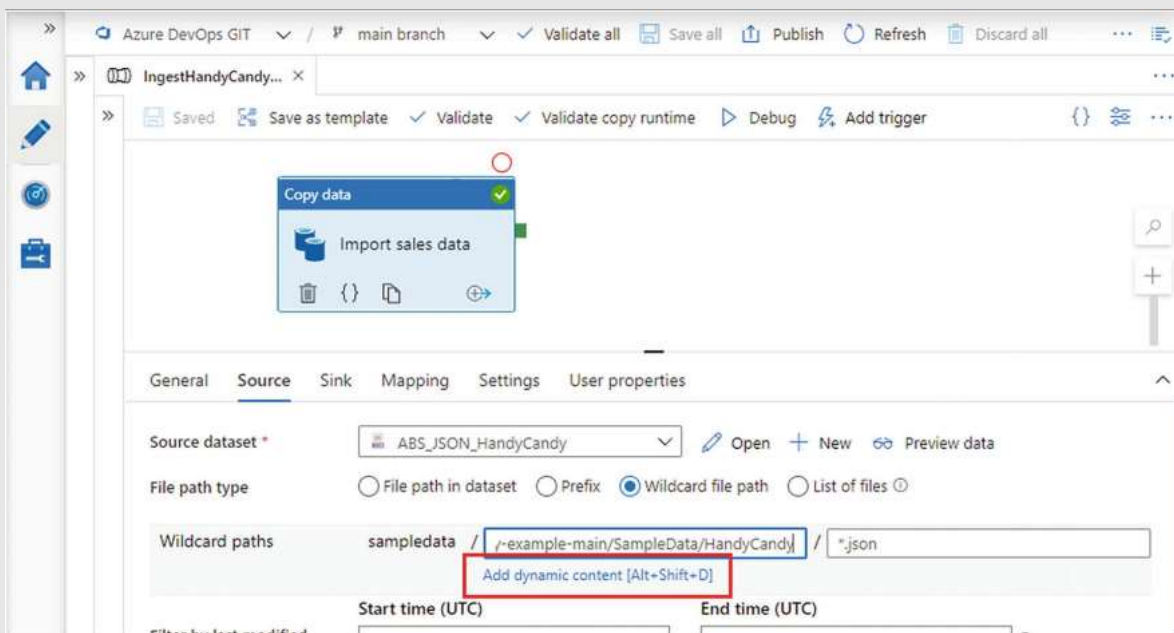


Figura 4-1 Añadir contenido dinámico [Alt+Mayúsculas+D] aparece al editar un campo

El mensaje en sí es un enlace para lanzar el constructor de expresiones del ADF UX. Como sugiere el mensaje, también puede iniciar el generador de expresiones para un campo pulsando Alt+Mayúsculas+D cuando el campo tiene el foco. La Figura 4-2 muestra el constructor de expresiones abierto.

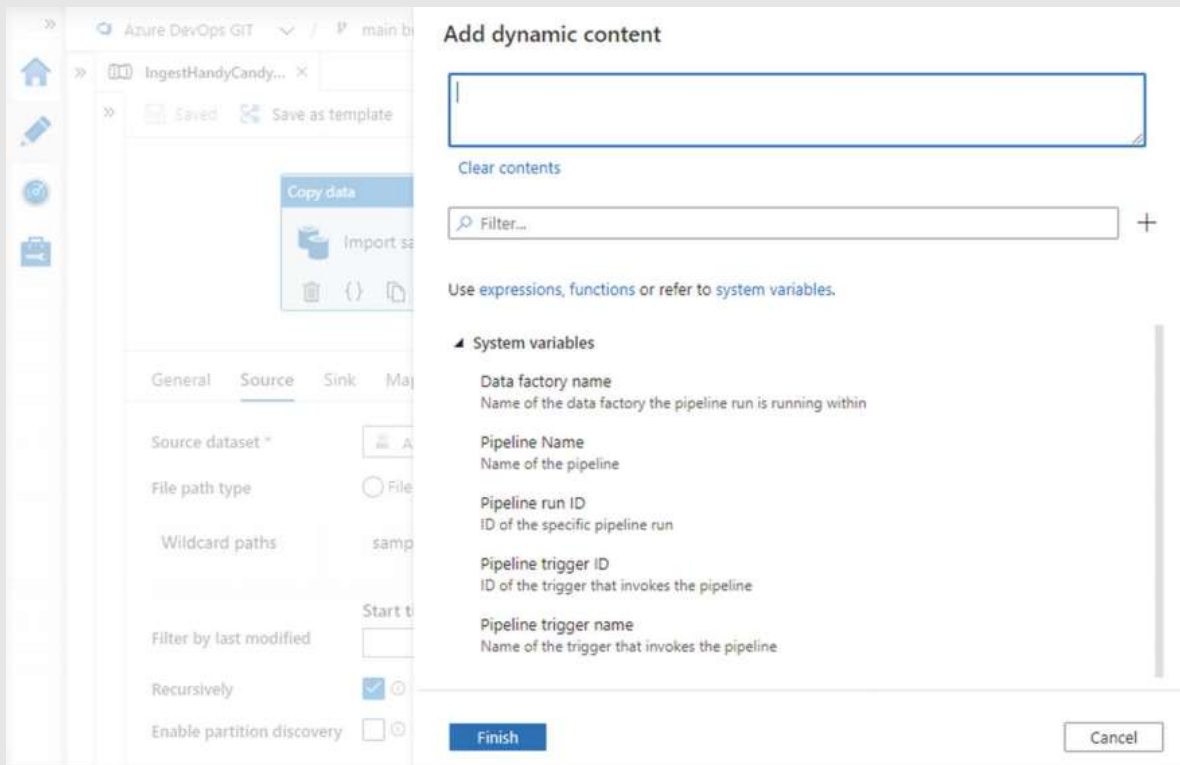


Figura 4-2 Constructor de expresiones del ADF

Las expresiones pueden hacer uso de valores literales, propiedades de recursos de fábrica y una biblioteca de funciones de expresión. La mitad inferior del constructor de expresiones es una lista desplazable de variables de sistema, funciones, variables de usuario y propiedades de recursos disponibles, dispuestas en secciones plegables. El cuadro de búsqueda Filtro... situado encima de la lista permite localizar rápidamente los elementos necesarios.

Una parte posterior de este capítulo se ocupa de las variables de usuario. Antes de eso, empezará a utilizar las variables y funciones del sistema para añadir funcionalidad al pipeline.

Tenga en cuenta que las expresiones del ADF distinguen entre mayúsculas y minúsculas: cuando introduzca expresiones a mano, asegúrese de utilizar las mayúsculas y minúsculas correctas cuando se refiera a una función, variable o propiedad.

4.2. Utilizar las variables del sistema

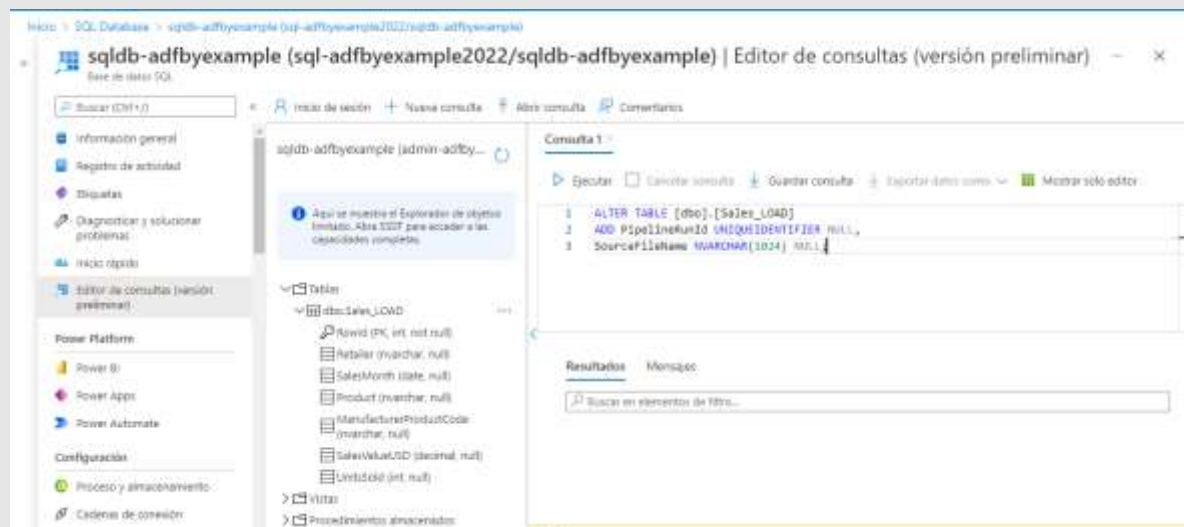
Entre sus opciones de configuración del conjunto de datos de origen, la actividad Copiar datos permite especificar columnas adicionales que pueden copiarse al receptor de la actividad. Puede utilizar esta función para duplicar una columna existente, para incluir un valor estático adicional o para añadir un valor determinado en tiempo de ejecución. Un caso de uso común para los valores dinámicos es añadir información de auditoría a los datos de origen entrantes. En esta sección, ampliará el manejo de los datos de Sweet Treats añadiendo valores de auditoría de esta manera.

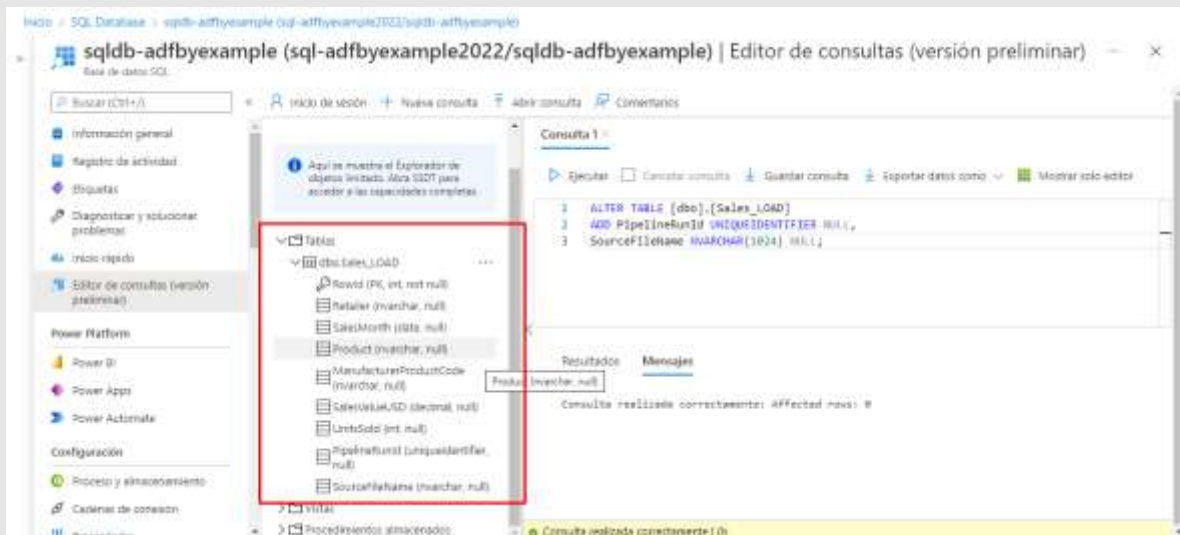
4.2.1. Habilitar el almacenamiento de información de auditoría

Antes de continuar, amplíe la tabla [dbo].[Sales_LOAD] para acomodar la información de auditoría adicional utilizando su cliente SQL para ejecutar el script dado en el Listado 4-1 contra su base de datos Azure SQL.

```
ALTER TABLE [dbo].[Sales_LOAD]
ADD PipelineRunId UNIQUEIDENTIFIER NULL,
SourceFileName NVARCHAR(1024) NULL;
```

Listado 4-1 Añadir columnas a [dbo].[Sales_LOAD]





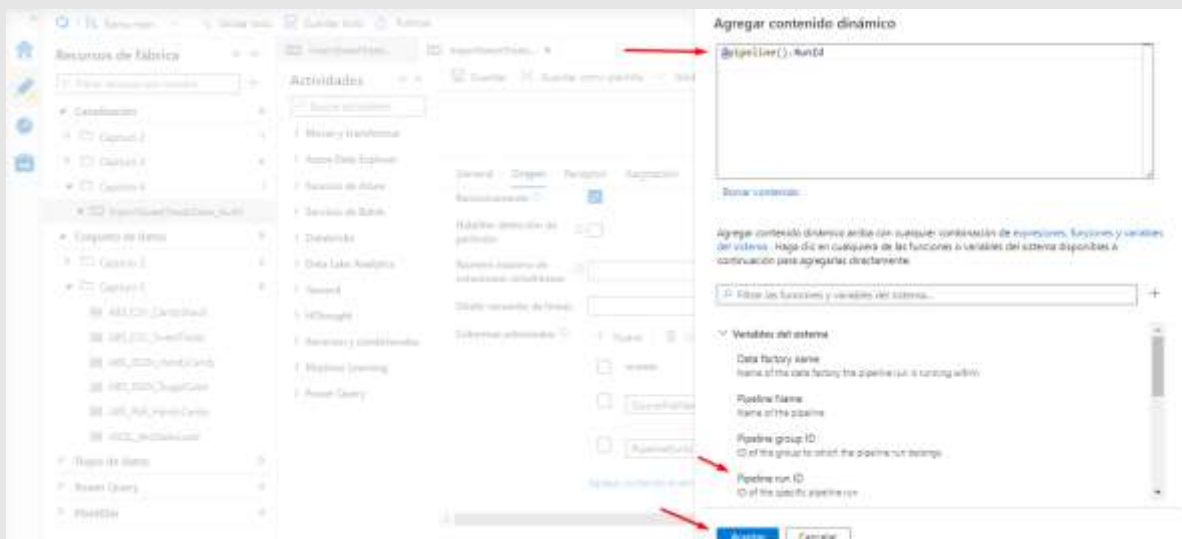
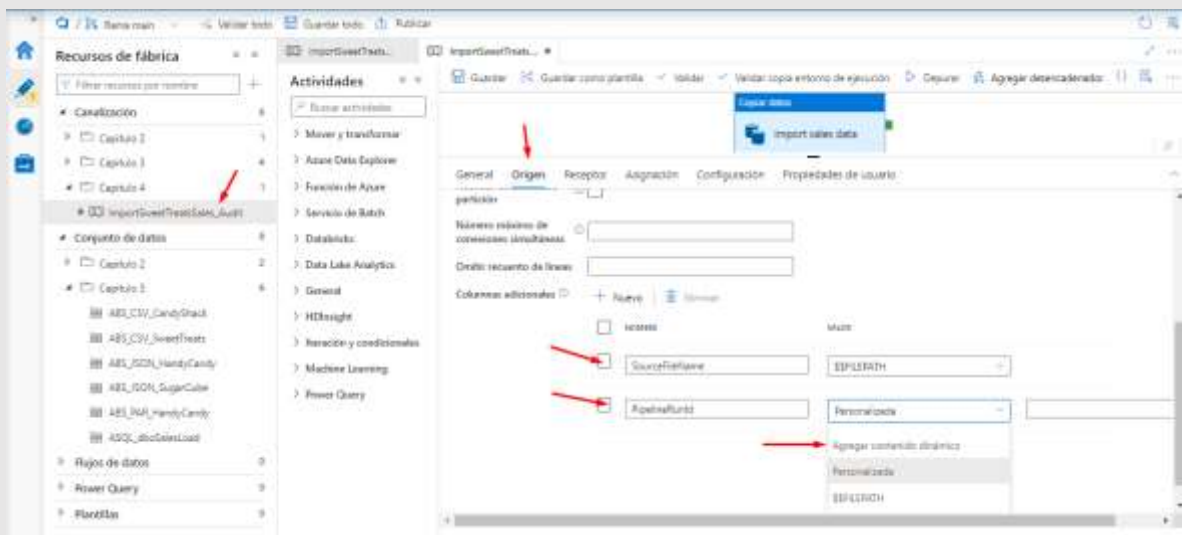
4.2.2. Crear un nuevo pipeline

Cree un nuevo pipeline clonando su pipeline "ImportSweetTreatsSales" del Capítulo 3. Nombre el pipeline clonado "ImportSweetTreatsSales_Audit", luego cree una carpeta "Capítulo 4" para pipelines y arrastre el nuevo pipeline dentro de ella.

4.2.2. Añadir nuevas columnas de origen

Seleccione la actividad de Copiar datos del nuevo pipeline. Añada nuevas columnas de origen así:

1. Navegue hasta la pestaña Origen y desplácese hacia abajo hasta encontrar la sección **Columnas adicionales**.
2. Añada una nueva columna haciendo clic en el botón + Nuevo. Déle el nombre **"SourceFileName"** y deje su valor establecido en el valor predeterminado **\$\$FILEPATH**. Esta variable reservada especial indica al ADF que la ruta del archivo origen en tiempo de ejecución debe ser proporcionada como valor del campo - no es una verdadera expresión del ADF y sólo puede ser utilizada aquí.
3. Añada una segunda columna nueva llamada **"PipelineRunId"**. Despliegue el desplegable del campo VALOR del nuevo campo y seleccione **"Agregar contenido dinámico"**.
4. En el constructor de expresiones, haga clic en Pipeline run ID en la lista de variables del sistema. El texto **@pipeline().RunId** aparece en el panel de expresión en la parte superior del constructor. Haga clic en Finalizar.



Sugerencia Asegúrese de que los nombres de las nuevas columnas sean exactamente los especificados aquí. El pipeline de Sweet Treats se basa en el mapeo del esquema inferido de ADF: los nombres de columna del conjunto de datos de origen y de destino deben coincidir para que el mapeo tenga éxito.

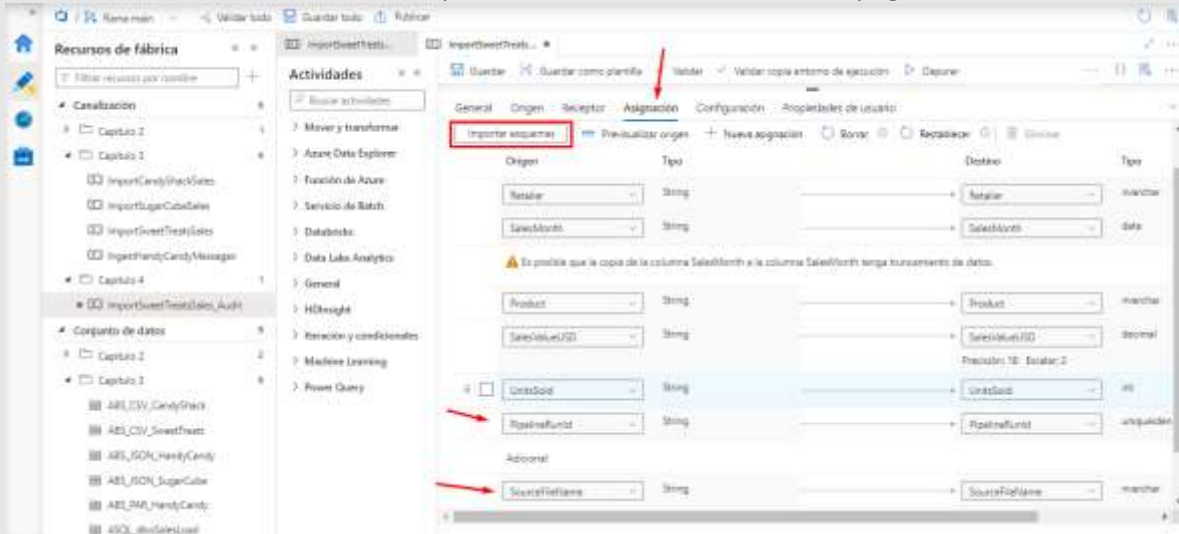
4.2.3. Ejecutar el pipeline

Ejecute el pipeline y espere a que termine de ejecutarse, luego utilice su cliente SQL para inspeccionar el contenido de la tabla [dbo].[Sales_LOAD] - La Figura 4-3 muestra las dos nuevas columnas de la base de datos rellenas con un valor de pipeline run ID y una ruta de archivo de origen. (La ruta del archivo no es la ruta completa y absoluta del archivo en blob storage, sino que

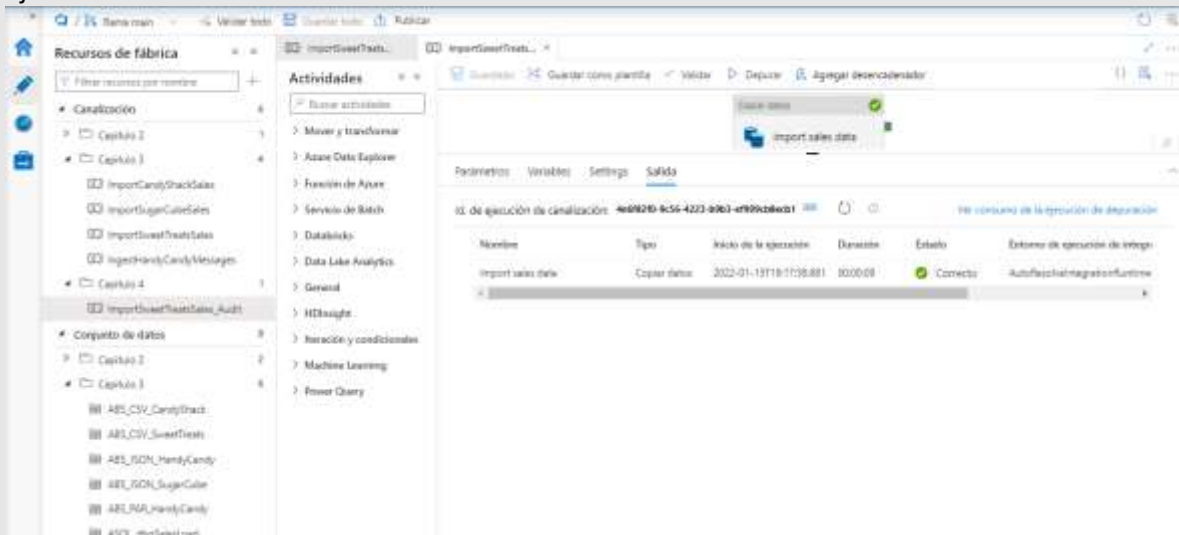
es relativa a la ruta de la carpeta del conjunto de datos de origen: este es el comportamiento de \$\$FILEPATH).

El pipeline run ID insertado en la nueva columna es un identificador único global (GUID). Los GUIDs se utilizan para identificar las ejecuciones del pipeline de la misma manera que se utilizan para identificar casi todo en Azure. El valor del ID de ejecución - "91d506d0-b5a4-4b53-8124-d8d22ea9f46f" en la Figura 4-3 - es el valor devuelto por la expresión @pipeline().RunId.

Antes de ejecutar, importe esquemas para actualizar los esquemas. Es importante, en las columnas adicionales volver a seleccionar las opciones de destino en las listas desplegadas.



Ejecute.



Antes de realizar la consulta.

Inicio > SQL Database > sqldb-adfbyexample (sql-adfbyexample2022/sqldb-adfbyexample)

sqldb-adfbyexample (sql-adfbyexample2022/sqldb-adfbyexample) | Editor de consultas (versión preliminar)

Base de datos SQL

Buscar (Ctrl+F)

Inicio de sesión + Nueva consulta Abrir consulta Comentarios

Consultas Consultas 2

Ejecutar Cancelar consulta Guardar consulta Exportar datos como Mostrar todo

Resultados Mensajes

SalesValueUSD	UnitsSold	PipelineUnitId	SourceFileName
6450.93	887		
3922.21	409		
888.44	818		
271.32	28		
1210.04	718		
1625.28	494		
448.25	75		

Consulta realizada correctamente 0s

Inicio > SQL Database > sqldb-adfbyexample (sql-adfbyexample2022/sqldb-adfbyexample)

sqldb-adfbyexample (sql-adfbyexample2022/sqldb-adfbyexample) | Editor de consultas (versión preliminar)

Base de datos SQL

Buscar (Ctrl+F)

Inicio de sesión + Nueva consulta Abrir consulta Comentarios

Consultas Consultas 2

Ejecutar Cancelar consulta Guardar consulta Exportar datos como Mostrar todo

Resultados Mensajes

Code	SalesValueUSD	UnitsSold	PipelineUnitId	SourceFileName
1210.04	718		1da20947-791e-4470-b0b8-42...	May-20/Sales.csv
6450.93	887		1da20947-791e-4470-b0b8-42...	Jun-20/Sales.csv
3922.21	409		1da20947-791e-4470-b0b8-42...	Apr-20/Sales.csv
888.44	818		1da20947-791e-4470-b0b8-42...	Jul-20/Sales.csv
271.32	28		1da20947-791e-4470-b0b8-42...	Aug-20/Sales.csv
1625.28	494		1da20947-791e-4470-b0b8-42...	Sep-20/Sales.csv
448.25	75		1da20947-791e-4470-b0b8-42...	Jun-20/Sales.csv
973.39	391		1da20947-791e-4470-b0b8-42...	Jun-20/Sales.csv

Consulta realizada correctamente 0s

Home > AdfByExample (adfbyexample-sql/AdfByExample)

AdfByExample (adfbyexample-sql/AdfByExample) | Query editor (preview) ×

SQL database

» 👤 Login ➕ New Query ↕ Open query 💡 Feedback

Query 2 ×

▶ Run ☐ Cancel query ↓ Save query ↓ Export data as 📊 Show only Editor

```
1 SELECT TOP (1000) * FROM [dbo].[Sales_LOAD]
```

>

Results Messages

Product	M...	SalesValueUSD	UnitsSold	PipelineRunId	SourceFileName
Schnoogles 8.81oz		271.32	28	91d506d0-b5a4-4b53-8124-d8d22ea9f46f	Aug-20/Sales.csv
Schnoogles 8.81oz		6450.93	687	91d506d0-b5a4-4b53-8124-d8d22ea9f46f	Jun-20/Sales.csv
Twisters 10.57oz		1625.26	494	91d506d0-b5a4-4b53-8124-d8d22ea9f46f	Sep-20/Sales.csv

Figura 4-3 Resultados de la consulta mostrando las columnas de auditoría rellenas

El propósito del símbolo @ inicial es indicar a ADF que lo que sigue debe ser evaluado como una expresión. Si omite el símbolo @, la expresión se convierte en un valor fijo: la cadena "pipeline().RunId". Dicha omisión provocaría el fallo de este pipeline, ya que la cadena no es un valor válido para el tipo SQL UNIQUEIDENTIFIER de la columna receptor.

Consejo Debe utilizar el editor de expresiones para crear expresiones - si las introduce directamente en un campo, el ADF UX las almacena como valores de cadena. Los campos que contienen valores se muestran con un fondo blanco; los que contienen expresiones son de color azul claro. Este efecto puede verse en la Figura 4-5, más adelante en este capítulo.

4.3. Acceso a las propiedades de ejecución de la actividad

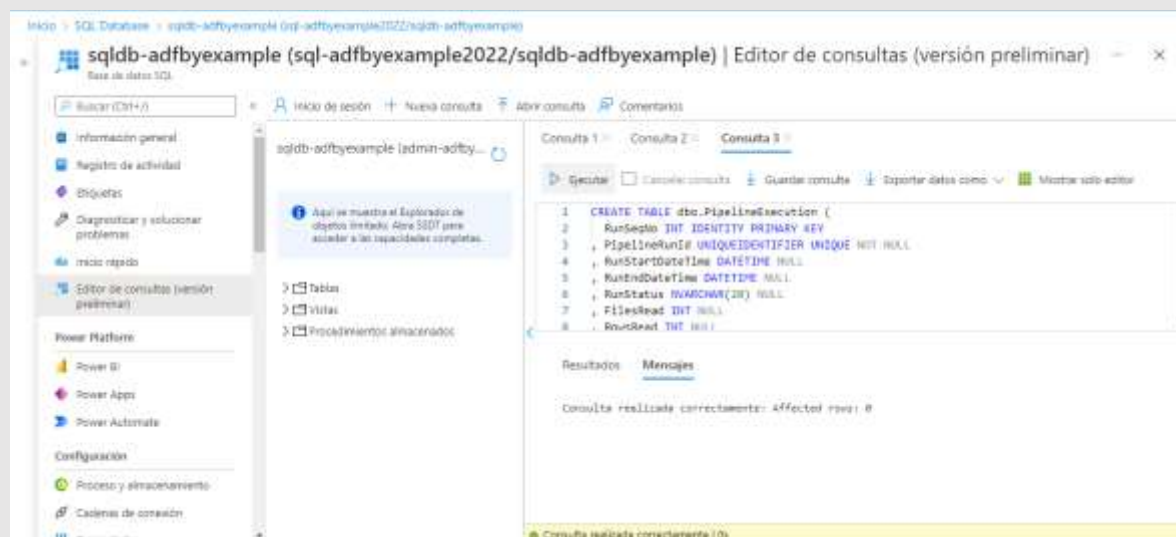
En esta sección, usted extenderá su pipeline para escribir alguna información de auditoría sobre la ejecución del pipeline en su Azure SQL Database. Lo hará utilizando la actividad de procedimientos almacenados de ADF.

4.3.1. Crear objetos de base de datos

Una vez completada la actividad Copiar datos, su pipeline llamará a un procedimiento almacenado para registrar cierta información sobre la ejecución del pipeline en una tabla de registro de la base de datos. Utilice los Listados 4-2 y 4-3 para crear la tabla de registro y el procedimiento almacenado en su base de datos.

```
CREATE TABLE dbo.PipelineExecution (  
    RunSeqNo INT IDENTITY PRIMARY KEY  
    , PipelineRunId UNIQUEIDENTIFIER UNIQUE NOT NULL  
    , RunStartDateTime DATETIME NULL  
    , RunEndDateTime DATETIME NULL  
    , RunStatus NVARCHAR(20) NULL  
    , FilesRead INT NULL  
    , RowsRead INT NULL  
    , RowsCopied INT NULL  
    , Comments NVARCHAR(1024) NULL  
);
```

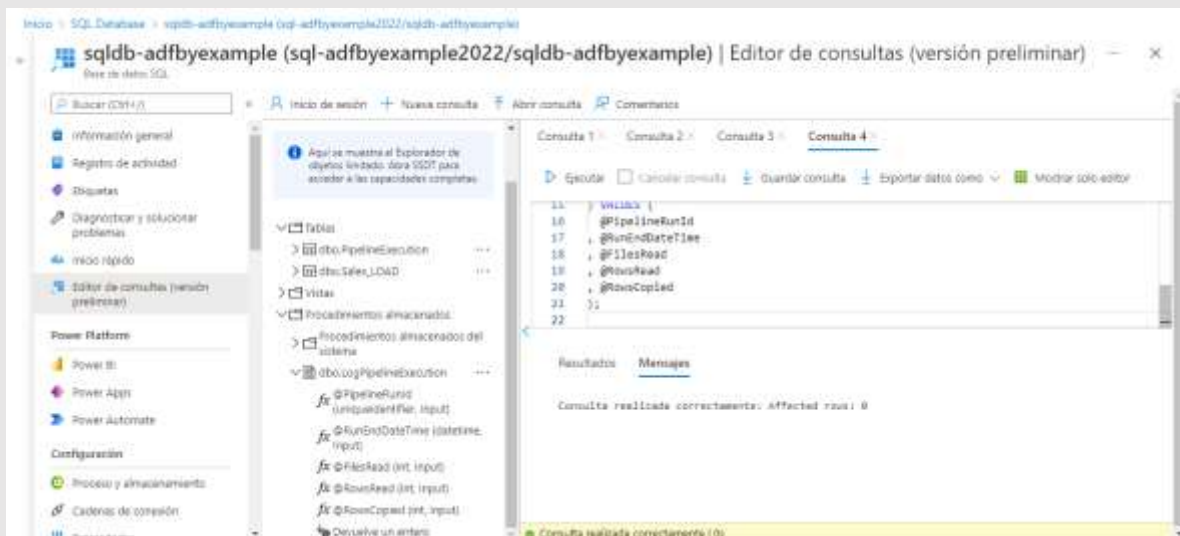
Listado 4-2 Crear una tabla de registro



```
CREATE PROCEDURE dbo.LogPipelineExecution (
    @PipelineRunId UNIQUEIDENTIFIER
, @RunEndDateTime DATETIME
, @FilesRead INT
, @RowsRead INT
, @RowsCopied INT
) AS
```

```
INSERT INTO dbo.PipelineExecution (
    PipelineRunId
, RunEndDateTime
, FilesRead
, RowsRead
, RowsCopied
) VALUES (
    @PipelineRunId
, @RunEndDateTime
, @FilesRead
, @RowsRead
, @RowsCopied
);
```

Listado 4-3 Crear un procedimiento almacenado de registro



4.3.2. Añadir la actividad de procedimiento almacenado

La actividad de procedimiento almacenado se encuentra en el grupo General de la caja de herramientas de actividades.

1. Abra el pipeline "ImportSweetTreatsSales_Audit" en el ADF UX y arrastre una actividad de procedimiento almacenado de la caja de herramientas al lienzo de creación. El panel de configuración debajo del lienzo se expande automáticamente - en la pestaña General de la actividad, nómbrela "Log pipeline outcome".
2. La nueva actividad debe ejecutarse después de que la actividad Copiar datos haya finalizado con éxito. Para ello, mantenga pulsado el botón izquierdo del ratón sobre el "asa" verde del borde derecho de la actividad Copiar datos y, a continuación, arrastre el puntero del ratón sobre la actividad Procedimiento almacenado. Al soltar el botón izquierdo del ratón aquí se crea una dependencia de la actividad. La Figura 4-4 muestra el pipeline configurado con la dependencia correcta.

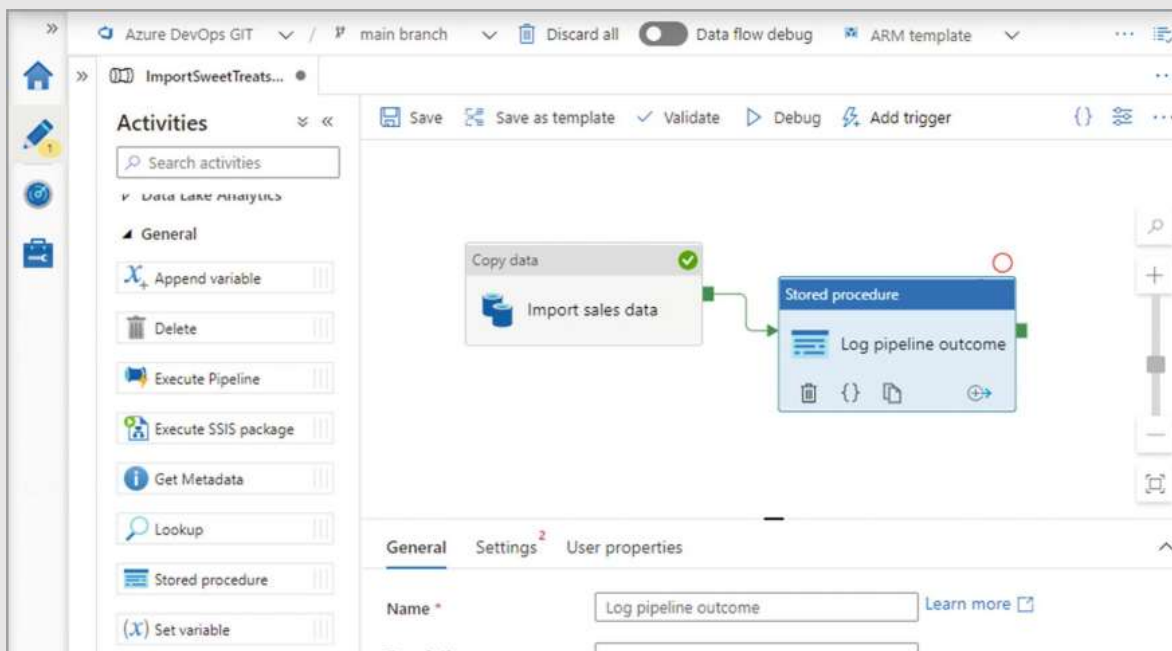
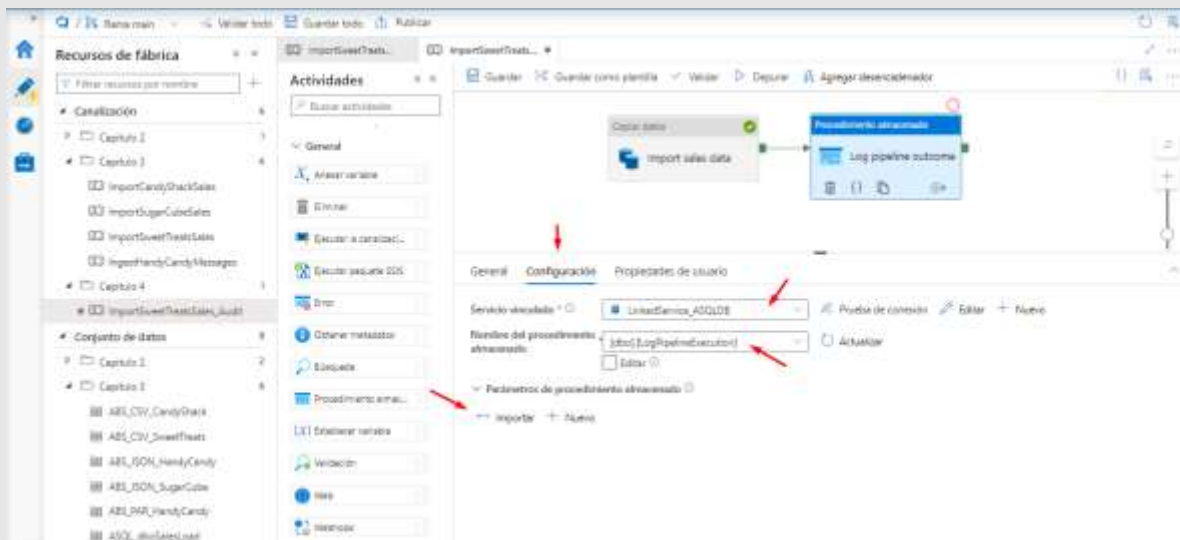


Figura 4-4 Actividad de procedimiento almacenado dependiente de la actividad Copiar datos

Sugerencia Para eliminar una actividad dependiente, haga clic con el botón derecho del ratón en la flecha de la dependencia y seleccione Eliminar en el menú emergente, o haga clic con el botón izquierdo del ratón para seleccionarla y pulse la tecla Eliminar. Estas opciones también están disponibles para las actividades de pipeline, además del botón Eliminar (icono de papelera) visible cuando se selecciona una actividad.

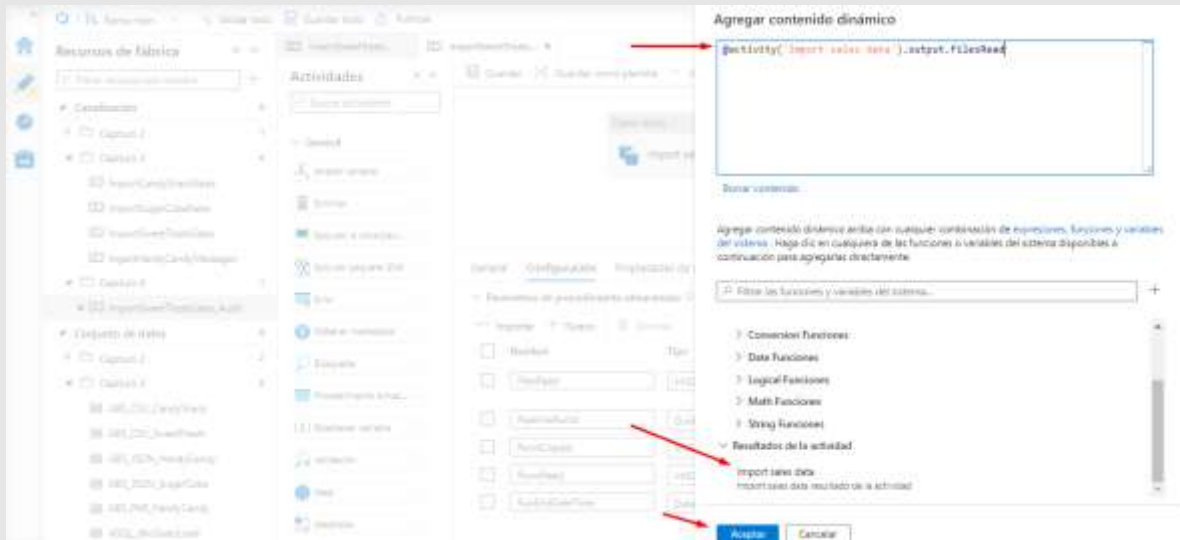
3. Seleccione la pestaña de Configuración de la actividad de procedimiento almacenado. Contiene dos campos de entrada obligatorios: elija su servicio vinculado a Azure SQL Database. Una vez hecho esto, podrá seleccionar el procedimiento almacenado recién creado en el desplegable Stored procedure name.
4. Debajo del encabezado de la sección **Parámetros del procedimiento almacenado** hay un botón **Importar** - haga clic en él para importar los parámetros del procedimiento almacenado desde su definición en la base de datos.



5. Utilice el constructor de expresiones para establecer el valor del parámetro "PipelineRunId" como antes, utilizando la variable del sistema Pipeline run ID.
6. Establezca el parámetro "RunEndDateTime" para utilizar la fecha y hora UTC actuales. Esto está disponible utilizando la función utcnow, que se encuentra en la sección Funciones de fecha de la lista de funciones del constructor de expresiones.

Nota Las funciones de fecha de la expresión devuelven strings, porque el lenguaje de expresión no tiene tipos de fecha u hora. El lenguaje tiene seis tipos: string, integer, float, boolean, arrays y diccionarios. Un diccionario es una colección de elementos con nombre.

7. Los valores de "FilesRead", "RowsRead" y "RowsCopied" serán tomados del objeto JSON de salida de la actividad Copiar datos - recuerde que encontró la salida de la actividad en el Capítulo 3. Abre el constructor de expresiones para el parámetro "FilesRead" y desplázate hasta el final de la lista de funciones.
8. La sección Activity outputs incluye una lista de actividades con salidas disponibles, en este caso, la actividad Copy data del pipeline (mostrada por su nombre, "Import sales data"). Haga clic en la actividad y observe que el texto @activity('Import sales data').output aparece en el panel de expresión.



9. activity('Import sales data') se refiere a la actividad llamada "Import sales data", y el sufijo .output especifica el objeto de salida de la actividad. Para identificar un campo dentro del objeto, debe añadir otro punto, seguido de su expresión de ruta JSON. El campo filesRead está en la raíz del objeto de salida, por lo que su ruta es simplemente el nombre del campo. Añada .filesRead al panel de expresión para que la expresión sea @activity('Import sales data').output.filesRead. Haga clic en Finalizar.

10. Configure los valores de los parámetros "RowsRead" y "RowsCopied" de la misma manera. La configuración completa de los parámetros se muestra en la Figura 4-5.

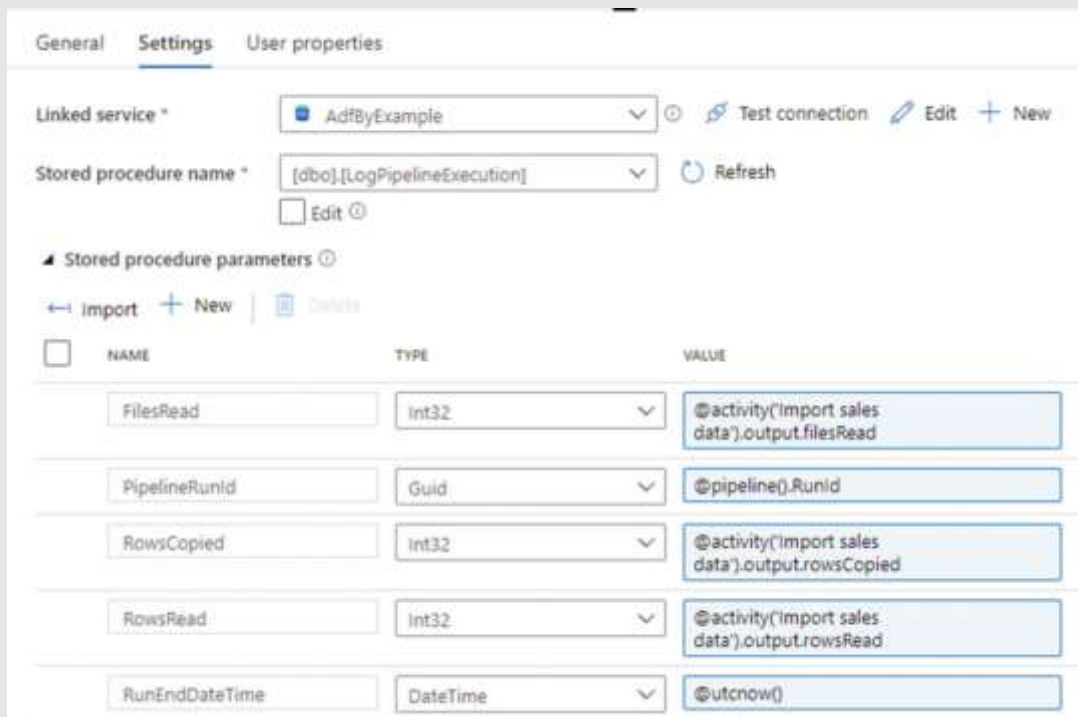


Figura 4-5 Expresiones de parámetros para [dbo].[LogPipelineExecution]

Para los desarrolladores de SSIS: La actividad Stored procedure es conceptualmente similar a la SSIS Execute SQL Task, pero no soporta la ejecución de sentencias SQL arbitrarias, ni puede devolver un conjunto de resultados. Las dependencias de la actividad tienen un claro paralelismo con las restricciones de precedencia de SSIS y se discuten en mayor detalle en el Capítulo 6.

4.3.3. Ejecutar el pipeline

Ejecute el pipeline, luego utilice su cliente SQL para inspeccionar el contenido de las dos tablas [dbo].[PipelineExecution] y [dbo].[Sales_LOAD].

La tabla [dbo].[PipelineExecution] contiene una única fila con información sobre la ejecución de la canalización. Puede verificar los valores de las columnas [PipelineRunId], [FilesRead], [RowsRead] y [RowsCopied] comparándolos con la salida de la actividad de copia de datos (en la pestaña Output del panel de configuración de la canalización). Observe que la pestaña de configuración de salida ahora muestra dos ejecuciones de actividades: la actividad de copia de datos y la actividad de procedimiento almacenado.

Las filas de [dbo].[Sales_LOAD] tienen un valor [PipelineRunId] que coincide con el de [dbo].[PipelineExecution].

The screenshot displays the SQL Server Data Tools (SSDT) interface. On the left, the 'Recursos de fábrica' (Factory Resources) pane shows a project named 'ImportSalesData' with a folder 'Canalizaciones' (Pipelines) containing 'ImportSalesData'. The 'Actividades' (Activities) pane shows the 'Import sales data' and 'Log pipeline outcome' tasks. The 'Salida' (Output) pane shows the execution results for both tasks.

Nombre	Tipo	Inicio de la ejecución	Duración	Estado	Entorno de ejecución de la tarea
Log pipeline outcome	Procedimiento...	2022-01-19T20:12:01.437	00:00:02	Completado	AutoReservadoIntegrationServices
Import sales data	Copiar datos	2022-01-19T20:11:52.118	00:00:05	Completado	AutoReservadoIntegrationServices

Inicio > SQL Database > sqldb-adfbyexample (sql-adfbyexample2022/sqldb-adfbyexample)

sqldb-adfbyexample (sql-adfbyexample2022/sqldb-adfbyexample) | Editor de consultas (versión preliminar)

Base de datos SQL

Buscar (Ctrl+F)

Inicio de sesión Nueva consulta Abrir consulta Comentarios

Información general Registro de actividad Etiquetas Diagnosticar y solucionar problemas Inicio rápido Editor de consultas (versión preliminar)

Power Platform Power BI Power Apps Power Automate Configuración Proceso y almacenamiento Cadenas de conexión Innovaciones

sqldb-adfbyexam...

Aquí se muestra el Explorador de objetos integrado. Para SSOT para acceder a las capacidades completas.

Tablas > dbo.PipelineExecut > dbo.Sales_LOAD > vistas > Procedimientos almacenados

Consulta 1 Consulta 2 Consulta 3 Consulta 4

Ejecutar Cancelar consulta Guardar consulta Exportar datos como Mostrar solo editor

1 SELECT * FROM PipelineExecution

Resultados Mensajes

Buscar en elementos de filtro...

RunId	PipelineRunId	RunStartDateTime	RunEndDateTime	RunStatus
1	db04513-0683-4a5d-82ae-cb9ea6...		2022-01-15T20:12:01.4210000	

Consulta realizada correctamente [0s]

Inicio > SQL Database > sqldb-adfbyexample (sql-adfbyexample2022/sqldb-adfbyexample)

sqldb-adfbyexample (sql-adfbyexample2022/sqldb-adfbyexample) | Editor de consultas (versión preliminar)

Base de datos SQL

Buscar (Ctrl+F)

Inicio de sesión Nueva consulta Abrir consulta Comentarios

Información general Registro de actividad Etiquetas Diagnosticar y solucionar problemas Inicio rápido Editor de consultas (versión preliminar)

Power Platform Power BI Power Apps Power Automate Configuración Proceso y almacenamiento Cadenas de conexión Innovaciones

sqldb-adfbyexam...

Aquí se muestra el Explorador de objetos integrado. Para SSOT para acceder a las capacidades completas.

Tablas > dbo.PipelineExecut > dbo.Sales_LOAD > vistas > Procedimientos almacenados

Consulta 1 Consulta 2 Consulta 3 Consulta 4

Ejecutar Cancelar consulta Guardar consulta Exportar datos como Mostrar solo editor

1 SELECT * FROM PipelineExecution

Resultados Mensajes

Buscar en elementos de filtro...

FileId	RowId	RowCopied	Comments
6	664	664	

Consulta realizada correctamente [0s]

Inicio > SQL Database > sqldb-adfbyexample (sql-adfbyexample2022/sqldb-adfbyexample)

sqldb-adfbyexample (sql-adfbyexample2022/sqldb-adfbyexample) | Editor de consultas (versión preliminar)

Base de datos SQL

Buscar (Ctrl+F)

Inicio de sesión Nueva consulta Abrir consulta Comentarios

Información general Registro de actividad Etiquetas Diagnosticar y solucionar problemas Inicio rápido Editor de consultas (versión preliminar)

Power Platform Power BI Power Apps Power Automate Configuración Proceso y almacenamiento Cadenas de conexión Innovaciones

sqldb-adfbyexam...

Aquí se muestra el Explorador de objetos integrado. Para SSOT para acceder a las capacidades completas.

Tablas > dbo.PipelineExecut > dbo.Sales_LOAD > vistas > Procedimientos almacenados

Consulta 1 Consulta 2 Consulta 3 Consulta 4

Ejecutar Cancelar consulta Guardar consulta Exportar datos como Mostrar solo editor

1 SELECT * FROM Sales_LOAD

Resultados Mensajes

Buscar en elementos de filtro...

UnitID	UnitsSold	PipelineRunId	SourceFileName
818		db04513-0683-4a5d-82ae-cb9ea63d1152	Jul-20/Sales.csv
718		db04513-0683-4a5d-82ae-cb9ea63d1152	May-20/Sales.csv

Consulta realizada correctamente [0s]

4.4. Utilizar la actividad de Lookup

Para los pipelines publicados, la información copiada en [dbo].[PipelineExecution] está disponible en el propio historial de ejecución de pipelines de ADF, pero las tablas log pueden ser más convenientes, particularmente para el monitoreo y análisis de pipelines a largo plazo. En este escenario, la descripción de las ejecuciones de pipeline usando GUIDs puede ser poco manejable o inconveniente. El uso de un identificador entero para el seguimiento del linaje - en lugar del [PipelineRunId] en la tabla [dbo].[Sales_LOAD], por ejemplo - puede tener el efecto deseable de reducir el tamaño de las filas de la base de datos.

La actividad Lookup de ADF se utiliza para obtener pequeñas cantidades de información de un dataset de ADF para su uso en la ejecución de un pipeline. En esta sección, utilizará la actividad Lookup para recuperar el valor entero de identidad de una fila de log (ya asignado en [dbo].[PipelineExecution]) y utilizarlo en lugar del GUID de ejecución del pipeline.

4.4.1. Crear objetos de la base de datos

En la sección anterior, usted insertó una fila en la tabla Log ([dbo].[LogPipelineExecution]) después de que se completara la actividad de Copiar datos. Ahora tendrá que insertar una fila antes de que comience la copia, para generar y devolver su valor de identidad. Posteriormente, actualizará la misma fila para añadir la información de la ejecución del pipeline. Usando su cliente SQL, ejecute el código dado en los Listados 4-4 y 4-5 para crear dos nuevos procedimientos almacenados.

Listado 4-4 Crear [dbo].[LogPipelineStart]

```
CREATE PROCEDURE dbo.LogPipelineStart (  
    @PipelineRunId UNIQUEIDENTIFIER  
, @RunStartDateTime DATETIME  
, @Comments NVARCHAR(1024) = NULL  
) AS  
INSERT INTO dbo.PipelineExecution (  
    PipelineRunId  
, RunStartDateTime  
, Comments  
) VALUES (  
    @PipelineRunId  
, @RunStartDateTime  
, @Comments  
);  
SELECT SCOPE_IDENTITY() AS RunSeqNo;
```

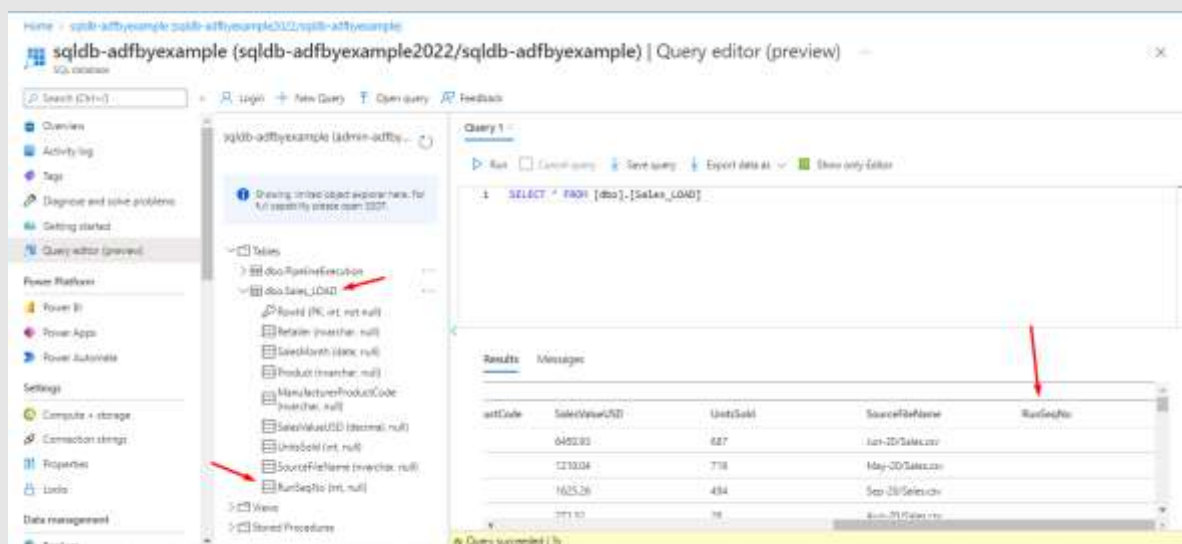
Listado 4-5 Crear [dbo].[LogPipelineEnd]

```
CREATE PROCEDURE dbo.LogPipelineEnd (  
    @RunSeqNo INT  
    , @RunEndTime DATETIME  
    , @RunStatus VARCHAR(20)  
    , @FilesRead INT  
    , @RowsRead INT  
    , @RowsCopied INT  
) AS  
UPDATE dbo.PipelineExecution  
SET RunEndTime = @RunEndTime  
    , RunStatus = @RunStatus  
    , FilesRead = @FilesRead  
    , RowsRead = @RowsRead  
    , RowsCopied = @RowsCopied  
WHERE RunSeqNo = @RunSeqNo;
```

El procedimiento almacenado que se muestra en el Listado 4-4 devuelve un conjunto de resultados compuesto por una fila y una columna, que contiene el valor INT único insertado en la tabla Log. Para utilizar este valor en lugar del pipeline run ID de ADF, es necesario realizar alguna modificación adicional en la tabla [dbo].[Sales_LOAD]. Ejecute el código indicado en el Listado 4-6 para sustituir la columna UNIQUEIDENTIFIER [PipelineRunId] de 16 bytes de la tabla por un campo INT más pequeño de 4 bytes.

Listado 4-6 Alterar la tabla de registro

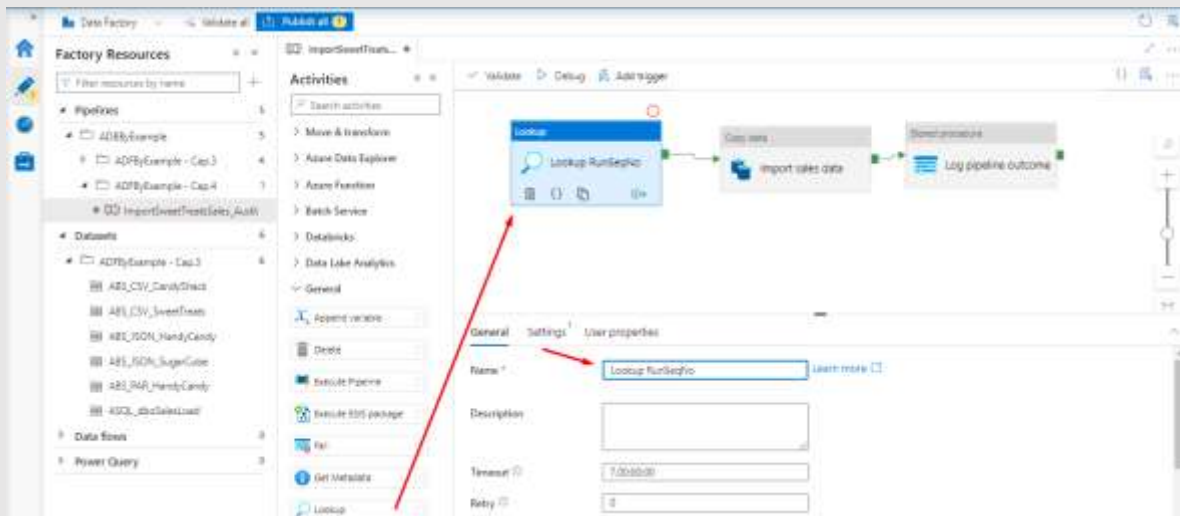
```
ALTER TABLE dbo.Sales_LOAD  
DROP COLUMN PipelineRunId;  
ALTER TABLE dbo.Sales_LOAD  
ADD RunSeqNo INT;
```



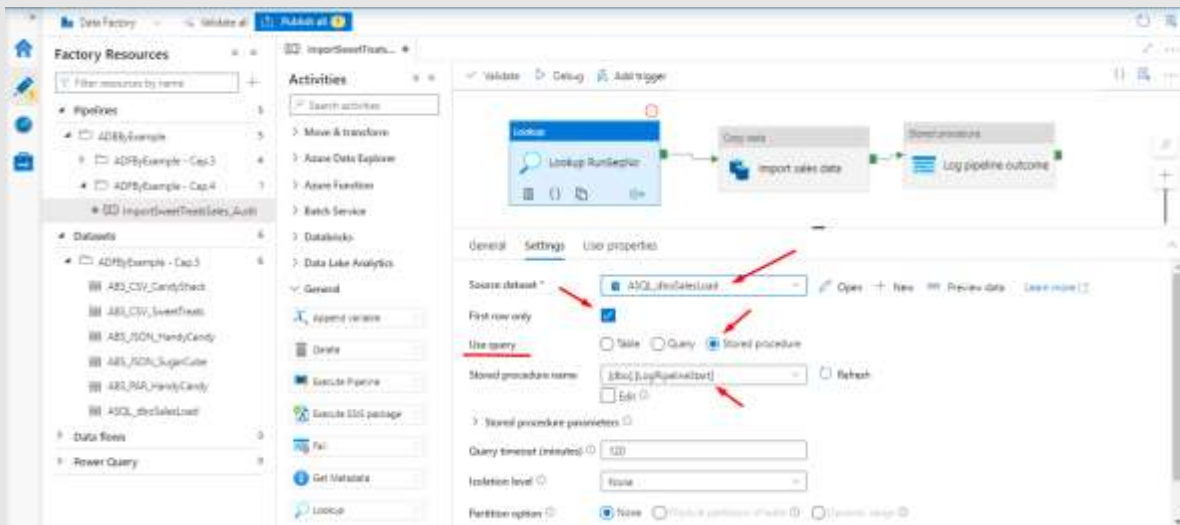
4.4.2. Configurar la Actividad de Lookup

Ahora está listo para revisar su pipeline "ImportSweetTreatsSales_Audit" en el ADF UX.

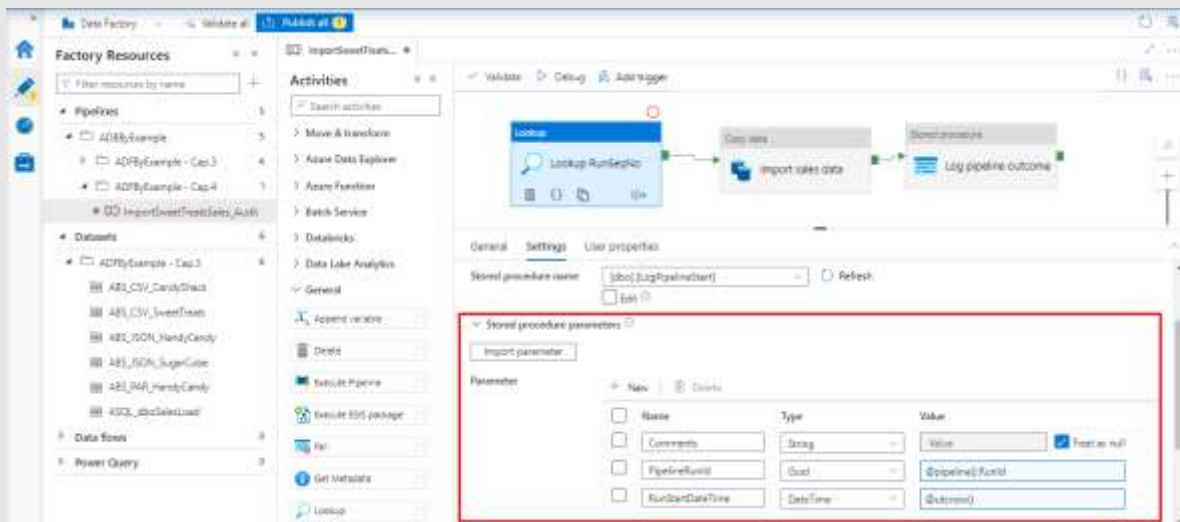
1. Arrastre una actividad Lookup desde la sección General de la caja de herramientas de actividades al lienzo de creación y nómbrela "Lookup RunSeqNo". Cree una dependencia para garantizar que la actividad Copy data se ejecute después de la actividad Lookup. La Figura 4-6 muestra la dependencia correctamente configurada.



2. Selecciona la actividad Lookup para que se muestre su panel de configuración y, a continuación, abre su pestaña Settings (Configuración). Recuerde que la actividad Lookup obtiene datos de un conjunto de datos, por lo que la pestaña le ofrece un menú desplegable de conjunto de datos: elija el conjunto de datos de Azure SQL DB que ha estado utilizando para copiar datos en [dbo].[Sales_LOAD].
3. Anteriormente, se podía anular la ruta de archivo configurada en un conjunto de datos de blob storage especificando una ruta comodín en la actividad de copia de datos - Las actividades de copia de datos que utilizan una conexión de Azure SQL DB pueden anular la configuración de su conjunto de datos de la misma manera. El conjunto de datos se creó para representar la tabla [dbo].[Sales_LOAD], pero puede anular esta configuración para representar otros conjuntos de resultados, como otras tablas (o vistas), consultas SQL o procedimientos almacenados que devuelven un conjunto de resultados. Establezca Usar consulta como "Procedimiento almacenado" para indicar que utilizará un procedimiento almacenado.
4. Cuando establezca Use query en "Stored procedure", el ADF UX le ofrece una lista desplegable de nombres. Elija su nuevo procedimiento [dbo].[LogPipelineStart] y haga clic en el botón Importar parámetros para adquirir su lista de parámetros.



5. Establezca los valores de "PipelineRunId" y "RunStartDateTime" como antes, utilizando las expresiones @pipeline().RunId y @utcnow(). El nuevo procedimiento almacenado también tiene un parámetro "Comments" - déjalo NULL por ahora, marcando la casilla Treat as null.
6. Finalmente, desplácese hacia abajo para comprobar que la casilla de verificación First row está marcada.



La Figura 4-6 muestra el panel de configuración correctamente completado para la actividad Lookup.

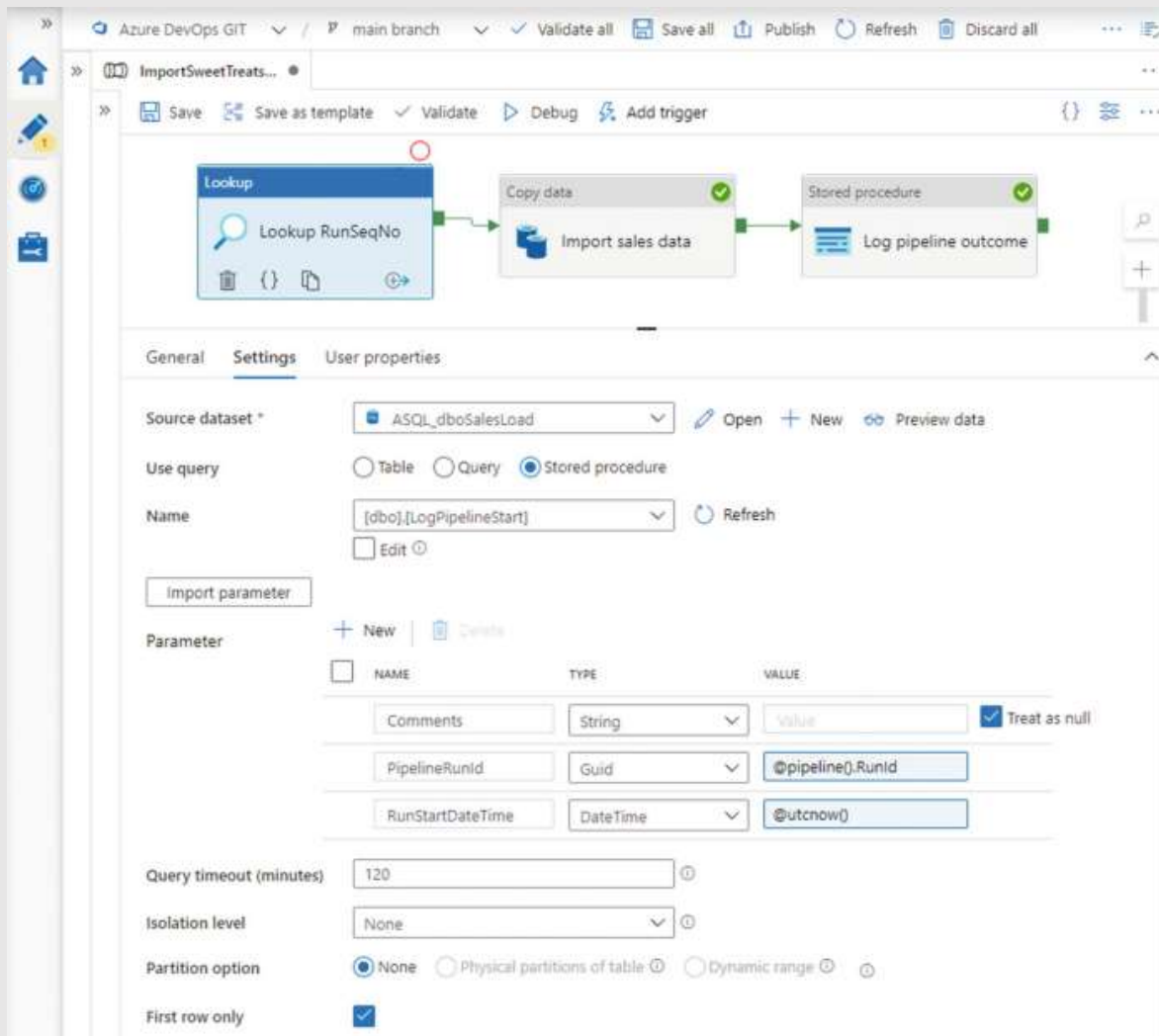


Figura 4-6 Actividad Lookup con la configuración correcta de dependencia y fuente

Para los desarrolladores de SSIS: No confunda la actividad Lookup de ADF con la transformación Lookup en SSIS - el lienzo de autoría es equivalente a la superficie de flujo de control de SSIS, por lo que aquí no se procesa ningún stream de filas. Una comparación más cercana es el uso de una tarea SSIS para rellenar una variable de paquete - por ejemplo, utilizando una tarea Execute SQL para recuperar un conjunto de resultados de una o más filas. La funcionalidad del conjunto de resultados ausente en la actividad de procedimientos almacenados de ADF está presente en cambio en la actividad Lookup.

4.4.3. Utilizar Breakpoints (Puntos de interrupción)

Para hacer uso de la salida de la actividad Lookup, debe leer el valor o valores que necesita de su objeto de salida de ejecución. La ruta JSON exacta requerida depende de la estructura del valor devuelto: qué nombres de campo contiene y si contiene sólo una fila o más. Una forma sencilla de encontrar la ruta JSON en el objeto de salida de la ejecución de la actividad es ejecutar la actividad.

No es posible ejecutar las actividades de ADF de forma aislada, pero la UX de ADF permite detener la ejecución en un punto determinado mediante un punto de interrupción. Cuando se selecciona una actividad en el lienzo de creación - como la actividad Lookup en la Figura 4-6 - se muestra un círculo rojo sobre su esquina superior derecha. Para establecer un punto de interrupción en la actividad, haga clic en el círculo. Cuando se establece el punto de interrupción, el círculo se llena de rojo - para eliminar el punto de interrupción, haga clic en el círculo lleno.

La Figura 4-7 muestra el proceso con un punto de interrupción establecido en la actividad de Lookup. Las otras actividades de la línea de producción están en gris porque el establecimiento del punto de interrupción las deshabilita - la actividad en la que se ha establecido el punto de interrupción se seguirá ejecutando. A diferencia de otros IDEs con los que puede estar familiarizado, la UX del ADF no le permite reanudar la ejecución después de que el punto de interrupción haya sido alcanzado - un pipeline se ejecuta hasta e incluyendo la primera actividad con un punto de interrupción establecido, pero no más allá.

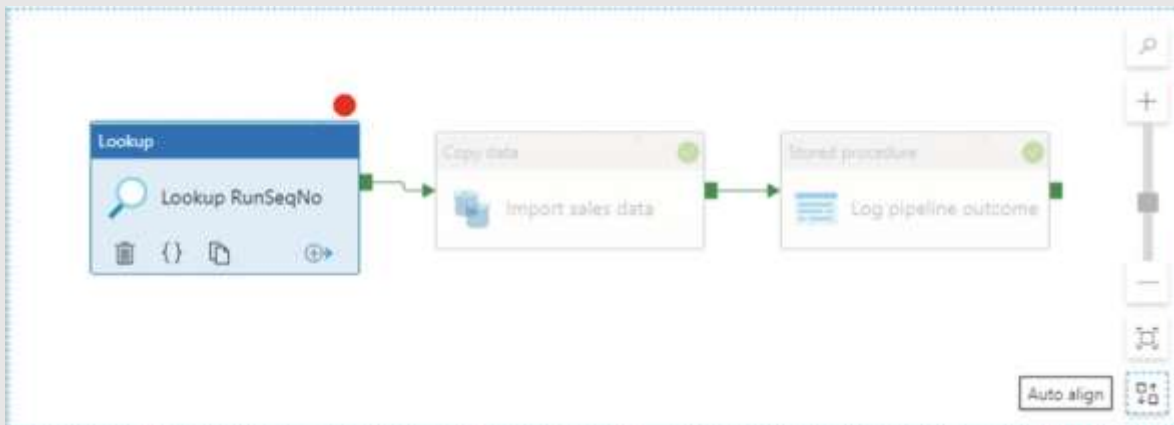


Figura 4-7 Punto de interrupción establecido en Lookup RunSeqNo

Sugerencia Para organizar las actividades de forma ordenada en el lienzo de creación, utilice el botón Auto align, que se encuentra en la parte inferior de la columna de herramientas de visualización en el lado derecho del lienzo (mostrado en la Figura 4-7).

Establezca un punto de interrupción en la actividad Lookup como se muestra en la Figura 4-7, y luego ejecute el pipeline. Cuando la ejecución esté completa, inspeccione el JSON de salida de la actividad en la pestaña de configuración de salida del pipeline. La Figura 4-8 muestra el panel de salida, resaltando el área de interés - la ruta JSON al valor devuelto por el procedimiento almacenado es `firstRow.RunSeqNo`.

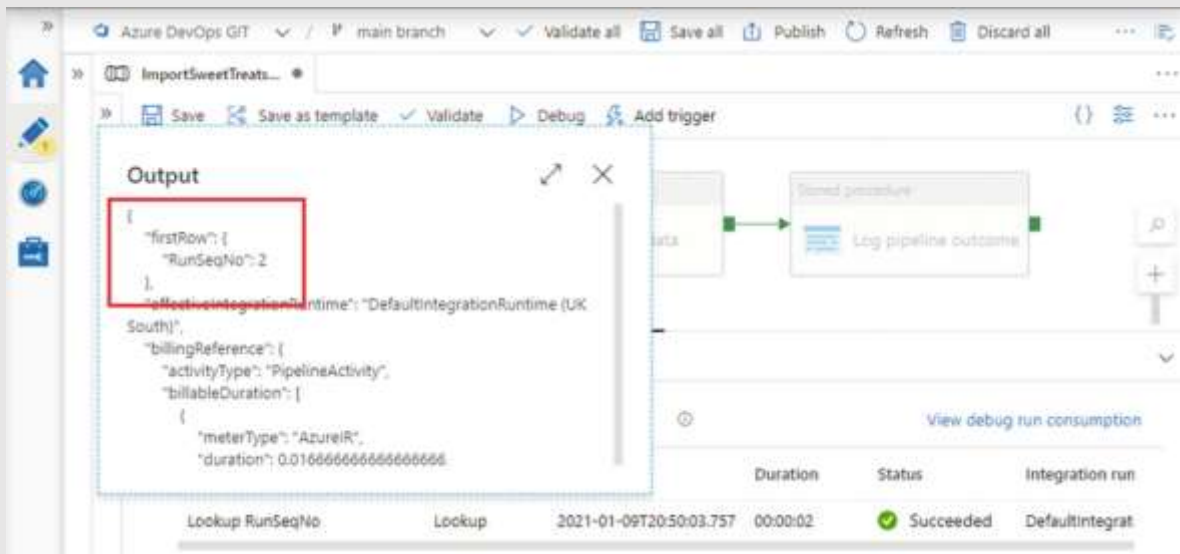
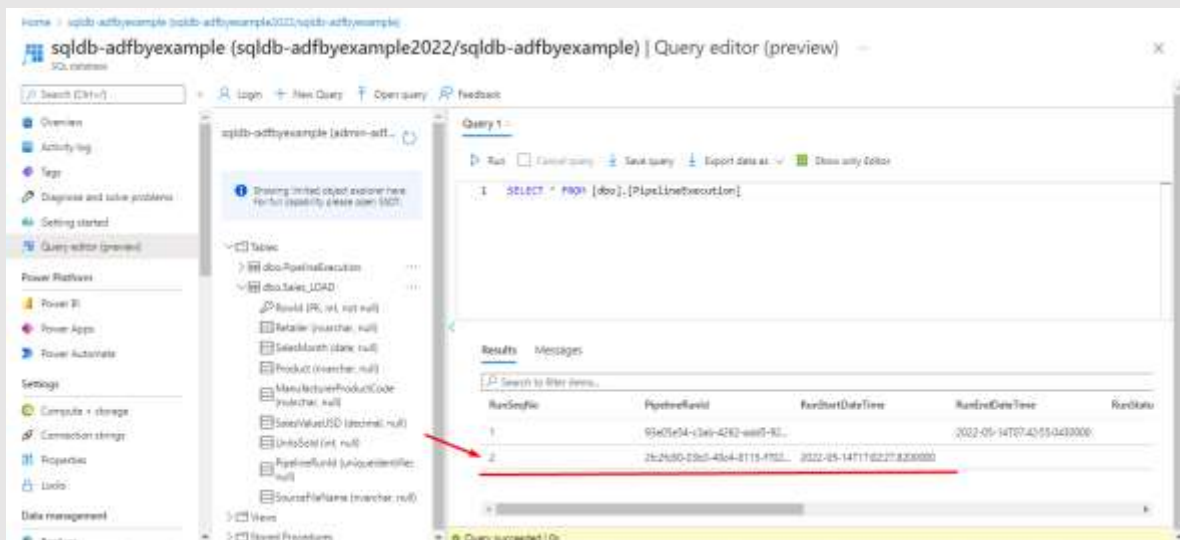


Figura 4-8 Objeto de salida de la actividad Lookup

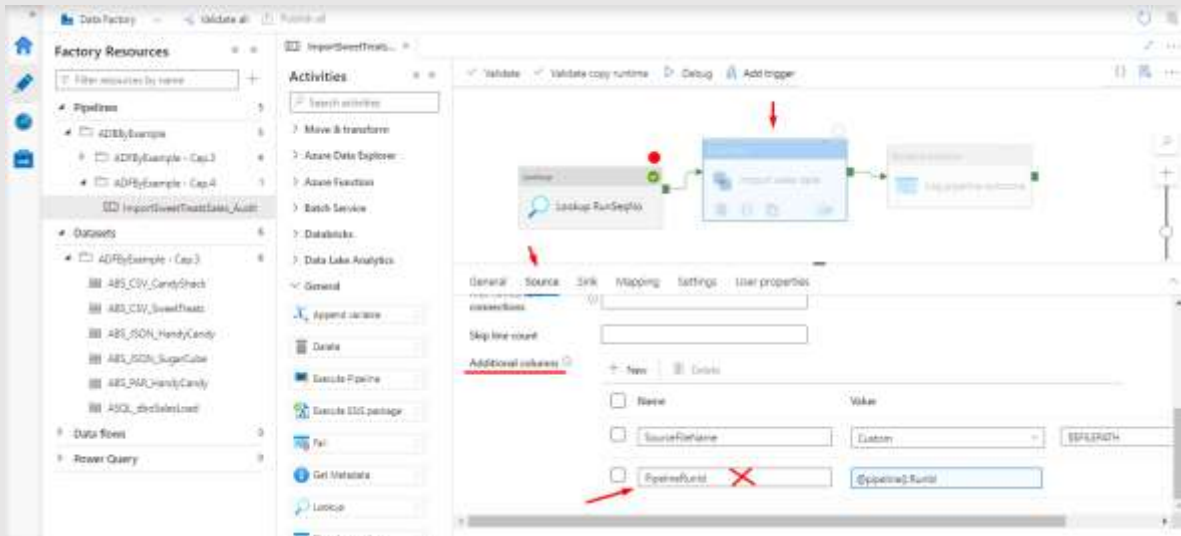
Se registro en la tabla dbo.PipelineExecution el siguiente registro después de haber ejecutado la primera actividad solamente. Se registraron datos solamente en las 3 primeras columnas, según lo indicado en el Stored procedure.



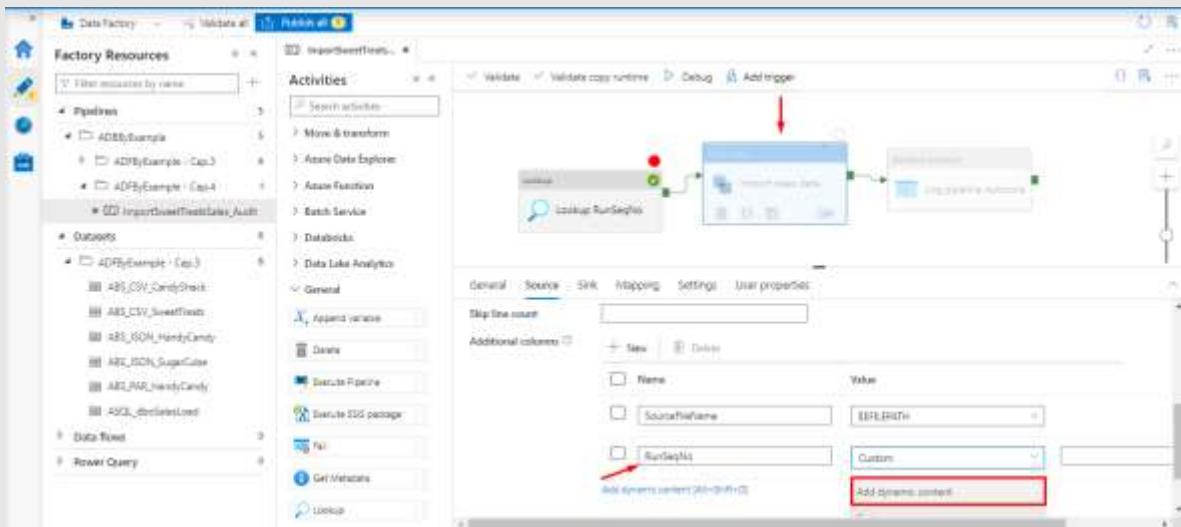
4.4.4. Utilizar el valor de Lookup

Con el lookup value's JSON path, ahora puede utilizar el valor en la actividad Copiar datos.

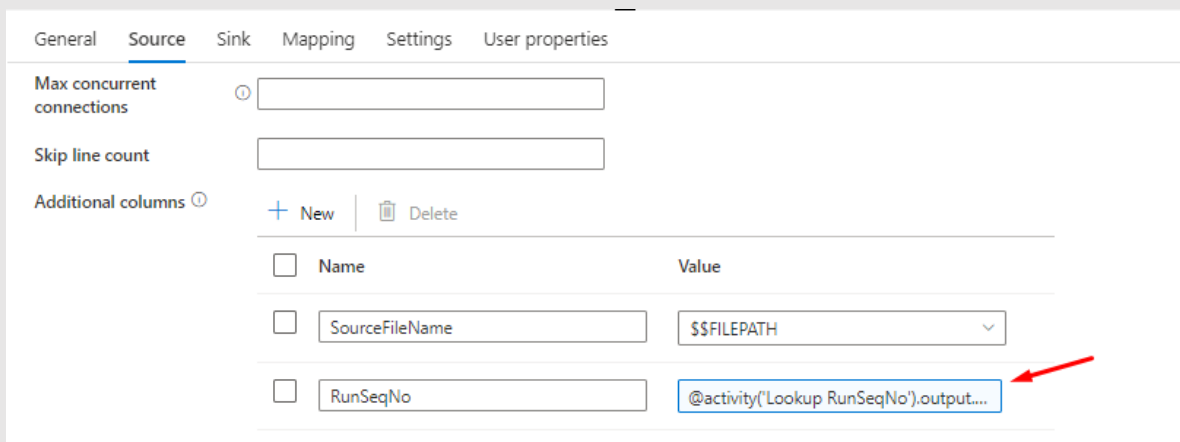
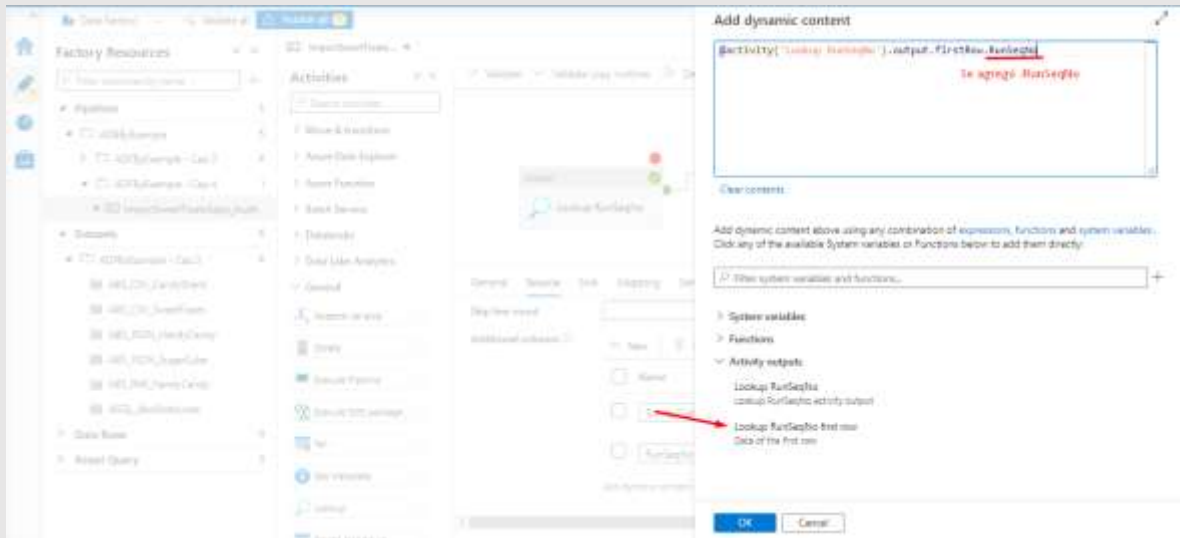
1. Seleccione la actividad de copia de datos del pipeline y abra la pestaña de configuración del origen. Desplácese hacia abajo hasta encontrar la sección Columnas adicionales.



2. Cambie el nombre de la columna "PipelineRunId" por "RunSeqNo". Recuerda que la actividad está utilizando un mapeo inferido, por lo que el nombre de la columna debe coincidir exactamente con el de la nueva columna de la tabla de la base de datos.



3. Abra el generador de expresiones para el campo VALUE de la columna "RunSeqNo". Elimine la expresión -hay un enlace Clear contents debajo del panel de expresión- y desplácese hacia abajo hasta la sección Activity outputs para encontrar el resultado de la actividad Lookup.
4. Haga clic en Lookup activity output para añadirla al panel de expresión, luego añada un punto y la ruta JSON identificada anteriormente. Compruebe que la expresión dice `@activity('Lookup RunSeqNo').output.firstRow.RunSeqNo` y, a continuación, haga clic en Finalizar.

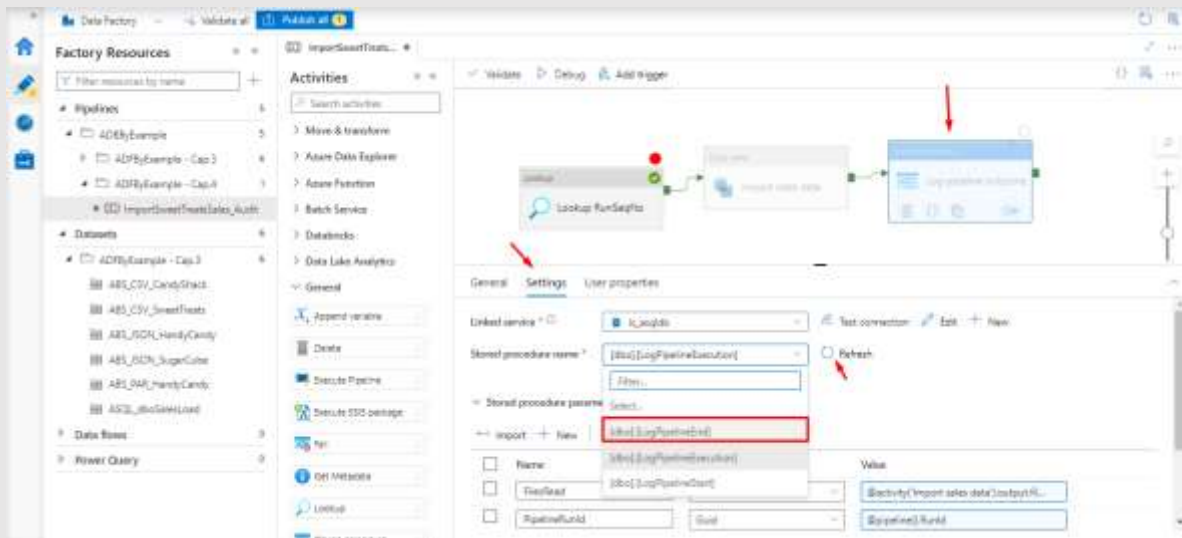


Puedes notar que el constructor de expresiones detecta automáticamente la sintaxis inválida, pero no puede detectar una ruta JSON inválida - esto es porque la ruta no se determina hasta el tiempo de ejecución.

4.4.5. Actualice la actividad del procedimiento almacenado

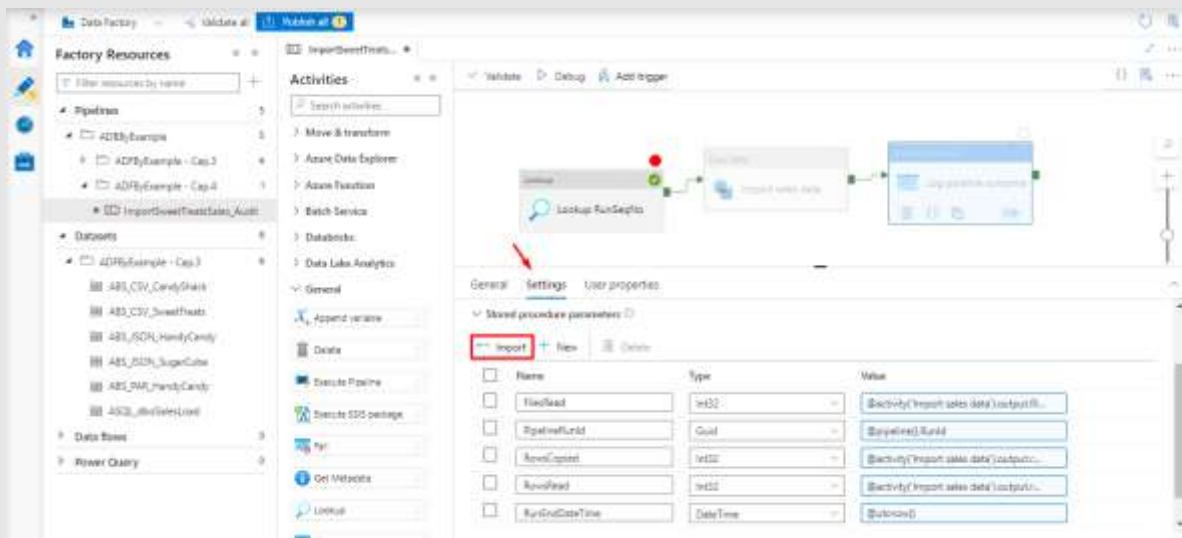
Ahora debe actualizar la actividad de procedimiento almacenado de su canalización para utilizar el nuevo procedimiento almacenado especificado en el Listado 4-5.

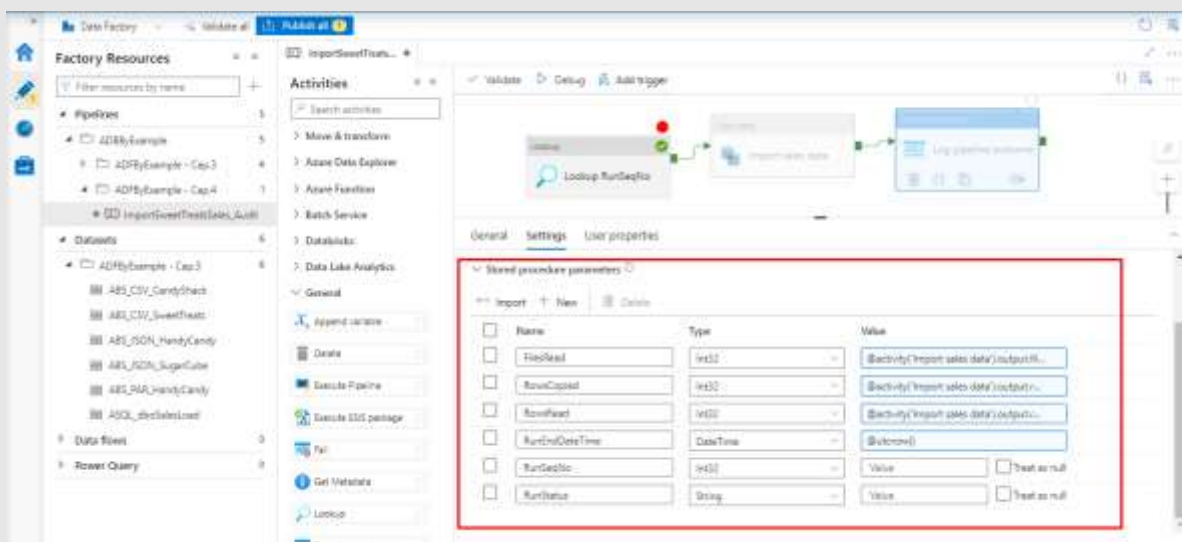
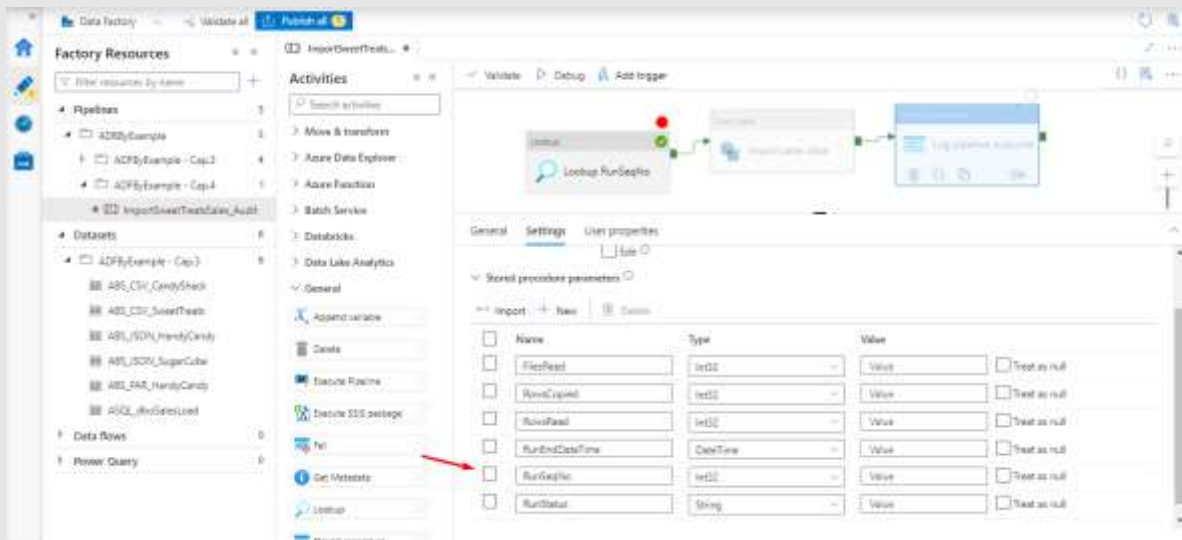
1. Seleccione la actividad de Procedimiento Almacenado y abra su pestaña de configuración de Ajustes. Seleccione "[dbo].[LogPipelineEnd]" de la lista desplegable del nombre del procedimiento almacenado.



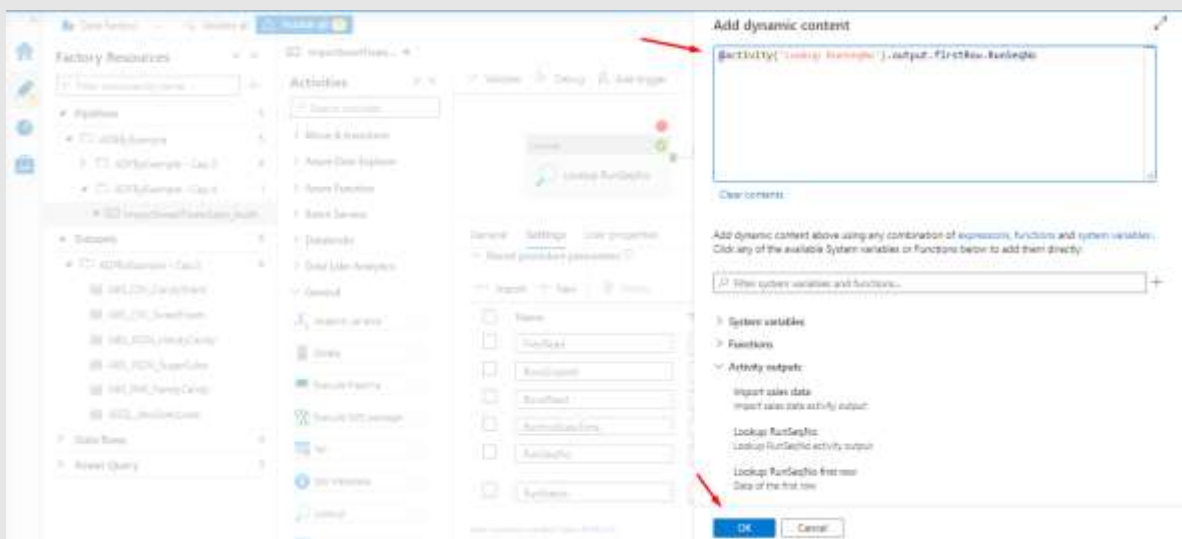
Sugerencia Si no puede ver el procedimiento almacenado en la lista desplegable, haga clic en el botón Actualizar que se encuentra a su derecha - esto recargará la lista desde la base de datos. Si el campo en sí parece ser un cuadro de texto en lugar de una lista desplegable, verifique que la casilla de verificación Editar no esté marcada.

2. En Parámetros del procedimiento almacenado, haga clic en el botón Importar para importar los parámetros del nuevo procedimiento. Establezca las expresiones para los valores de los parámetros "FilesRead", "RowsCopied", "RowsRead" y "RunEndDateTime" como antes.

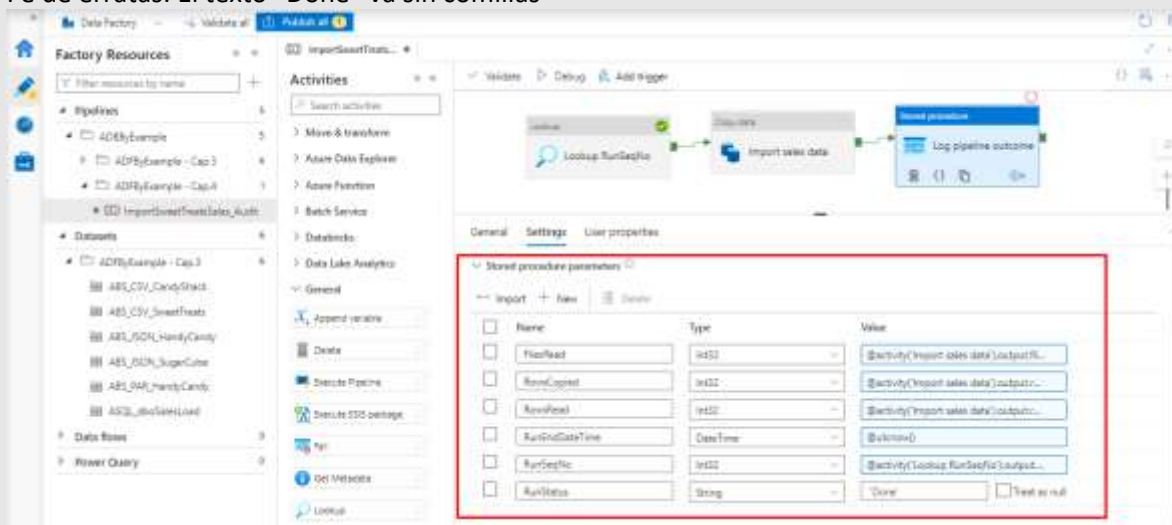




3. Establezca el valor de "RunSeqNo" para utilizar la misma expresión que en la sección anterior (basada en la salida de la actividad Lookup). Establezca el valor de "RunStatus" en "Done" - este es un string literal, por lo que puede introducirlo directamente en el campo sin lanzar el constructor de expresiones.



Fe de erratas: El texto “Done” va sin comillas



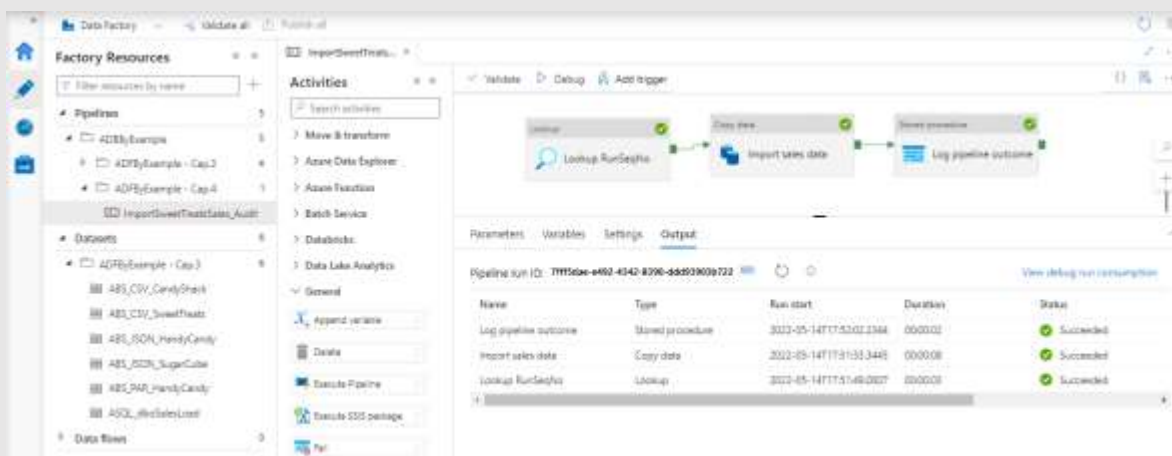
Las propiedades de salida de una actividad están disponibles para ser utilizadas por cualquier actividad que dependa de ella, ya sea directa o indirectamente.

4.4.6. Ejecutar el pipeline

Asegúrese de que ha desactivado el punto de interrupción de la sección anterior y ejecute la canalización. Cuando la ejecución se haya completado con éxito, utilice su cliente SQL para inspeccionar el contenido de la tabla. Debería encontrar que

- ❖ La última fila de [dbo].[PipelineExecution] - la fila con el mayor valor de [RunSeqNo] - contiene valores para [RunStartEndTime] y [RunEndDateTime].
- ❖ El valor de [RunSeqNo] en [dbo].[Sales_LOAD] coincide con el valor de la última fila de [dbo].[PipelineExecution].

El valor de [RunSeqNo] en [dbo].[PipelineExecution] es el valor creado y recuperado por la actividad Lookup, y posteriormente utilizado en las actividades Copy data y Stored procedure del pipeline.



Home > sqldb-adfbyexample > sqldb-adfbyexample2022/sqldb-adfbyexample

sqldb-adfbyexample (sqldb-adfbyexample2022/sqldb-adfbyexample) | Query editor (preview)

SQL database

Search (Ctrl+F) Login New Query Open query Feedback

Overview Activity log Tags Diagnose and solve problems Getting started Query editor (preview)

Power Platform Power BI Power Apps Power Automate

Settings Compute + storage Connection strings Properties Logs Data management Services

sqldb-adfbyexample (admin-adfby...)

Showing limited object explorer view. For full capability please open SSOT.

- Tables
 - dbo.PipelineExecution
 - dbo.Sales_LOAD
 - RowId (PK, int, not null)
 - Retailer (nvarchar, null)
 - SalesMonth (date, null)
 - Product (nvarchar, null)
 - ManufacturerProductCode (nvarchar, null)
 - SalesValueUSD (decimal, null)
 - UnitSold (int, null)
 - SourceFileName (nvarchar, null)
 - RunSeqNo (int, null)
- Views
- Stored Procedures

Query 1

Run Cancel query Save query Export data as Show only Editor

```
1 SELECT * FROM [dbo].[PipelineExecution]
```

Results Messages

Search to filter items...

RunSeqNo	PipelineId	RunStartTime	RunEndTime	RunStatus
1	93a5e54-c5e9-4262-ae05-82...		2022-05-14T07:42:55.0400000	
2	2b2b60-23ed-48a4-8115-f762...	2022-05-14T17:52:27.8200000		
3	78f5de-e602-4342-6290-4d4...	2022-05-14T17:51:45.2050000	2022-05-14T17:52:02.7670000	Done

Query succeeded | 3

Home > sqldb-adfbyexample > sqldb-adfbyexample2022/sqldb-adfbyexample

sqldb-adfbyexample (sqldb-adfbyexample2022/sqldb-adfbyexample) | Query editor (preview)

SQL database

Search (Ctrl+F) Login New Query Open query Feedback

Overview Activity log Tags Diagnose and solve problems Getting started Query editor (preview)

Power Platform Power BI Power Apps Power Automate

Settings Compute + storage Connection strings Properties Logs Data management Services

sqldb-adfbyexample (admin-adfby...)

Showing limited object explorer view. For full capability please open SSOT.

- Tables
 - dbo.PipelineExecution
 - dbo.Sales_LOAD
 - RowId (PK, int, not null)
 - Retailer (nvarchar, null)
 - SalesMonth (date, null)
 - Product (nvarchar, null)
 - ManufacturerProductCode (nvarchar, null)
 - SalesValueUSD (decimal, null)
 - UnitSold (int, null)
 - SourceFileName (nvarchar, null)
 - RunSeqNo (int, null)
- Views
- Stored Procedures

Query 1

Run Cancel query Save query Export data as Show only Editor

```
1 SELECT * FROM [dbo].[Sales_LOAD]
```

Results Messages

Search to filter items...

RowId	Retailer	SalesMonth	Product	Manufacturer
1	Sweet Treats	2020-06-01T00:00:00.0000000	Schmorgles 8.81oz	
2	Sweet Treats	2020-07-01T00:00:00.0000000	Pedsy 1.23oz	
3	Sweet Treats	2020-08-01T00:00:00.0000000	Schmorgles 8.81oz	
4	Sweet Treats	2020-09-01T00:00:00.0000000	Schmorgles 8.81oz	

Query succeeded | 3

Home > sqldb-adfbyexample > sqldb-adfbyexample2022/sqldb-adfbyexample

sqldb-adfbyexample (sqldb-adfbyexample2022/sqldb-adfbyexample) | Query editor (preview)

SQL database

Search (Ctrl+F) Login New Query Open query Feedback

Overview Activity log Tags Diagnose and solve problems Getting started Query editor (preview)

Power Platform Power BI Power Apps Power Automate

Settings Compute + storage Connection strings Properties Logs Data management Services

sqldb-adfbyexample (admin-adfby...)

Showing limited object explorer view. For full capability please open SSOT.

- Tables
 - dbo.PipelineExecution
 - dbo.Sales_LOAD
 - RowId (PK, int, not null)
 - Retailer (nvarchar, null)
 - SalesMonth (date, null)
 - Product (nvarchar, null)
 - ManufacturerProductCode (nvarchar, null)
 - SalesValueUSD (decimal, null)
 - UnitSold (int, null)
 - SourceFileName (nvarchar, null)
 - RunSeqNo (int, null)
- Views
- Stored Procedures

Query 1

Run Cancel query Save query Export data as Show only Editor

```
1 SELECT * FROM [dbo].[Sales_LOAD]
```

Results Messages

Search to filter items...

RowId	Retailer	SalesMonth	Product	Manufacturer
1	Sweet Treats	2020-06-01T00:00:00.0000000	Schmorgles 8.81oz	
2	Sweet Treats	2020-07-01T00:00:00.0000000	Pedsy 1.23oz	
3	Sweet Treats	2020-08-01T00:00:00.0000000	Schmorgles 8.81oz	
4	Sweet Treats	2020-09-01T00:00:00.0000000	Schmorgles 8.81oz	

Query succeeded | 3

Home > sqldb-adfbyexample (sqldb-adfbyexample2022/sqldb-adfbyexample) | Query editor (preview)

SQL database

Search (Ctrl+F) Login New Query Open query Feedback

Overview Activity log Sign Diagnose and solve problems Getting started Query editor (preview)

Power Platform Power BI Power Apps Power Automate Settings Compute + storage Connection strings Properties Data management

sqldb-adfbyexample (admin-adfby...)

Warning: limited object explorer view. for full capability please open SSMS.

Tables

- dbo.PipelineExecution
- dbo.Sales_LOAD
 - RowId (PK, int, not null)
 - RefDate (nvarchar, null)
 - SalesMonth (date, null)
 - Product (nvarchar, null)
 - ManufactureProductCode (nvarchar, null)
 - SalesValueUSD (decimal, null)
 - UnitsSold (int, null)
 - SourceFileName (nvarchar, null)
 - RunSeqNo (int, null)
- Views
- Stored Procedures

Query 1

Run Cancel query Save query Export data as Show all

Results Messages

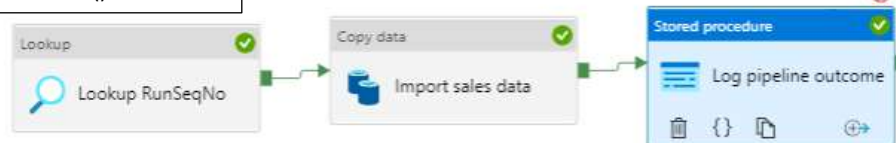
RowId	SalesValueUSD	UnitsSold	SourceFileName	RunSeqNo
6459.93	667	Jun-20/Sales.csv	3	
388.44	619	Jul-20/Sales.csv	3	
271.32	28	Aug-20/Sales.csv	3	
3522.31	408	Apr-20/Sales.csv	3	
1213.04	718	May-20/Sales.csv	3	
1625.26	434	Sep-20/Sales.csv	3	
448.25	75	Jun-20/Sales.csv	3	
973.59	381	Jun-20/Sales.csv	3	
2234.07	722	Jun-20/Sales.csv	3	

Query succeeded (0)

Utiliza el Stored Procedure **[dbo].[LogPipelineStart]** donde **inserta** un registro para las columnas **PipelineRunId**, **RunStartDateTime** y se genera un registro para la columna **RunSeqNo** de la tabla **PipelineExecution** y además se captura el "último" valor de la columna **RunSeqNo**, mediante **SCOPE_IDENTITY()**

Se copian los valores de los archivos CSV de la carpeta **/SweetTreats/*.csv** a la tabla **Sales_LOAD**. Además, se agregan dos columnas adicionales: **SourceFileName** y **RunSeqNo**. A ambas se le otorgan valores dinámicos. Para la primera se utiliza **\$\$FILEPATH** y la segunda toma el valor entregado por la actividad anterior **"Lookup RunSeqNo"**

Utiliza el Stored Procedure **[dbo].[LogPipelineEnd]** donde **actualiza** los registros para las columnas **RunSeqNo** y su valor dinámico lo obtiene de la primera actividad **"Lookup RunSeqNo"**, **RunEndDateTime** su valor lo obtiene de una función dinámica, **RunStatus** su valor es estático, **FilesRead**, **RowsRead**, **RowsCopied** sus valores dinámicos los obtienen de la actividad anterior **"Import sales data"**, de la tabla **PipelineExecution**



Nota También observará una fila incompleta en **[dbo].[PipelineExecution]** donde se inició una ejecución pero no terminó - este es el registro creado cuando se ejecutó el pipeline hasta el punto de interrupción de la actividad **Lookup**.

4.5. Variables de usuario

En la sección anterior, usted utilizó la expresión

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

en dos lugares: una vez en la actividad Copy data y otra vez en la actividad Stored procedure. Tener que utilizar la misma expresión dos veces no es lo ideal, porque si algo cambia (por ejemplo, el nombre de la actividad o el nombre del campo devuelto), tendrás que actualizar la expresión dondequiera que se utilice. Puede eliminar este código redundante utilizando una variable de usuario.

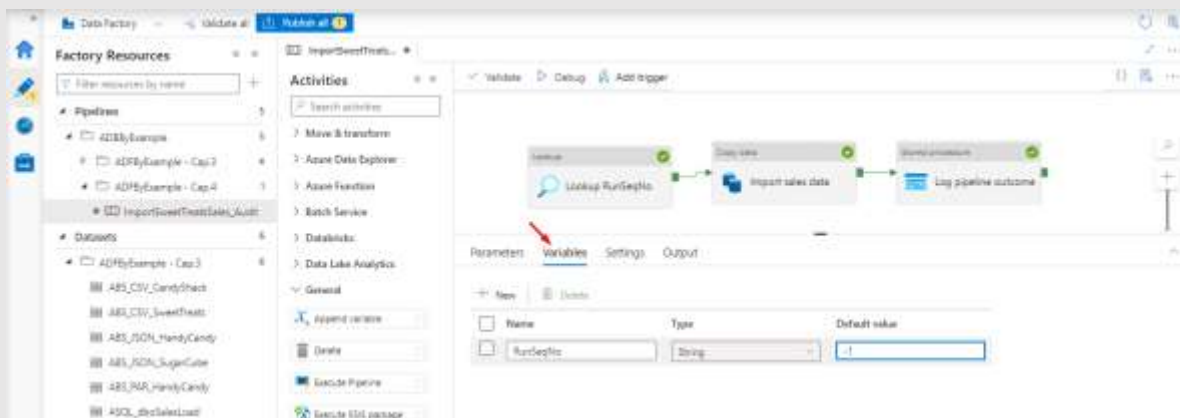
4.5.1. Crear una variable

Las variables de usuario se crean a nivel del pipeline.

1. Haga clic en algún lugar del espacio en blanco del lienzo de creación para abrir el panel de configuración que contiene los ajustes a nivel del pipeline.
2. Seleccione la pestaña Variables y haga clic en el botón + Nuevo para crear una nueva variable.
3. Una variable requiere un nombre, un tipo y, opcionalmente, un valor por defecto. Nombre la variable "RunSeqNo" y establezca su tipo como "String".

Nota El valor de RunSeqNo devuelto por la actividad Lookup es un número entero, pero las variables sólo pueden ser de tres tipos: **String, Boolean o Array**.

Especifica un valor por defecto de "-1". Proporcionar un valor por defecto para "RunSeqNo" no tiene mucho sentido desde la perspectiva del propósito de este pipeline, pero establecer uno facilita la detección de errores en los casos en que la variable no ha sido asignada correctamente.



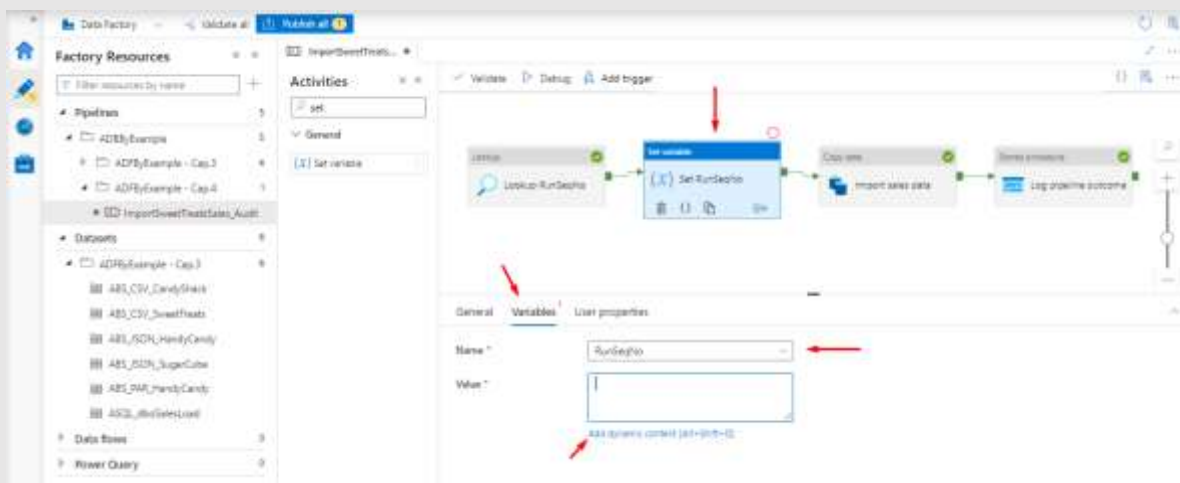
4.5.2. Establecer una variable

La asignación de variables se implementa mediante dos actividades del ADF, ambas se encuentran en la sección General de la caja de herramientas de actividades:

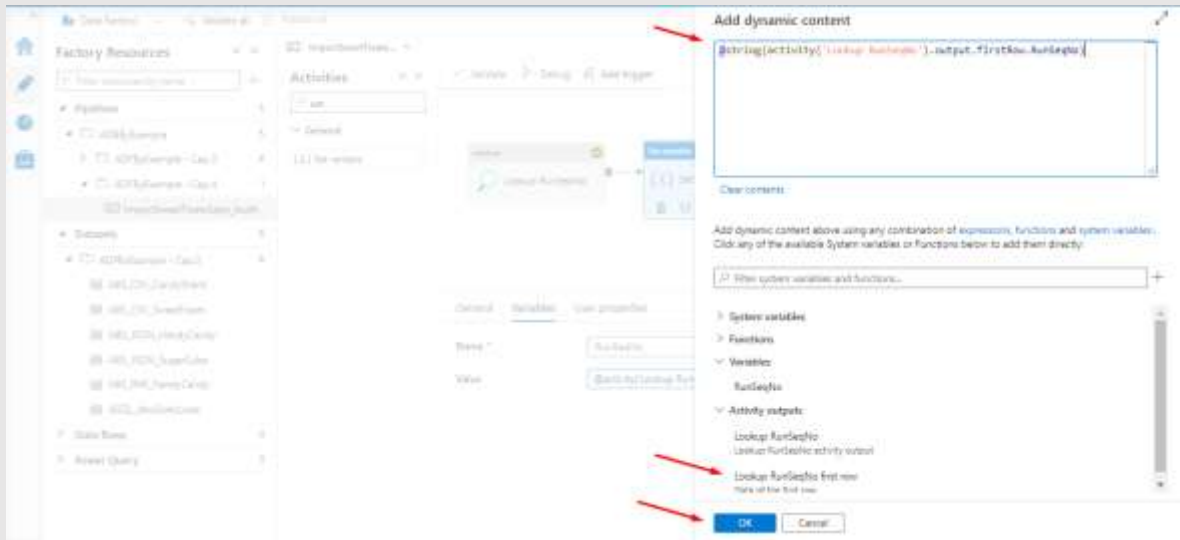
- ❖ La actividad **Set variable** establece el valor de una variable, sobrescribiendo su valor por defecto o cualquier valor anterior.
- ❖ La actividad **Append variable** añade un elemento al final de una variable de tipo Array.

Establezca la variable "RunSeqNo" para usarla en su pipeline como sigue:

1. Arrastre una actividad Set variable al lienzo de autoría y nómbrela "Set RunSeqNo". La variable se establecerá con el valor obtenido por la actividad Lookup, así que cree una dependencia entre las dos actividades.
2. En el panel de configuración de la actividad Set variable, selecciona la pestaña Variables y elige la variable "RunSeqNo" en el menú desplegable Name. Aparecerá un campo Valor debajo del desplegable.
3. Haga clic en el campo Valor para mostrar el enlace del generador de expresiones. Abra el constructor de expresiones.



4. En la sección Activity outputs, haga clic en Lookup RunSeqNo - si la actividad no aparece, verifique que la actividad Set variable depende de la actividad Lookup. Complete la expresión añadiendo .firstRow.RunSeqNo.



5. Aunque su número es limitado, los tipos de variables son estrictos. Sin una modificación adicional, esta asignación de variable fallará, porque el valor obtenido por la actividad Lookup es un número, no un string. **Para convertir el valor en un string, envuélvalo en la función de string** (que se encuentra en la sección de funciones de conversión del constructor de expresiones).

La Figura 4-9 muestra la actividad Set variable, configurada correctamente con la expresión

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

4.5.3. Utilizar la variable

Las variables tienen un alcance a nivel de pipeline, por lo que pueden ser utilizadas por cualquiera de las actividades del pipeline. Debe crear dependencias de actividades según sea necesario para asegurar que una variable no se utilice hasta que se haya establecido.

1. Cree una dependencia entre la actividad Set variable y la actividad Copy data. La Figura 4-9 muestra esta dependencia.

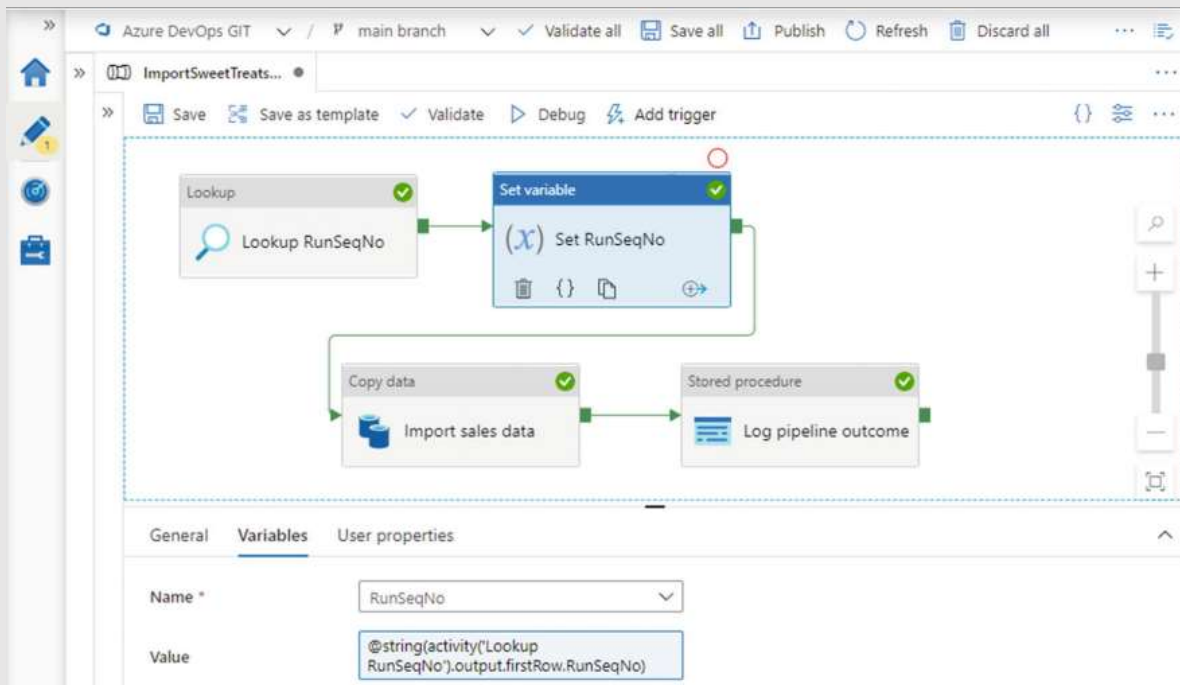
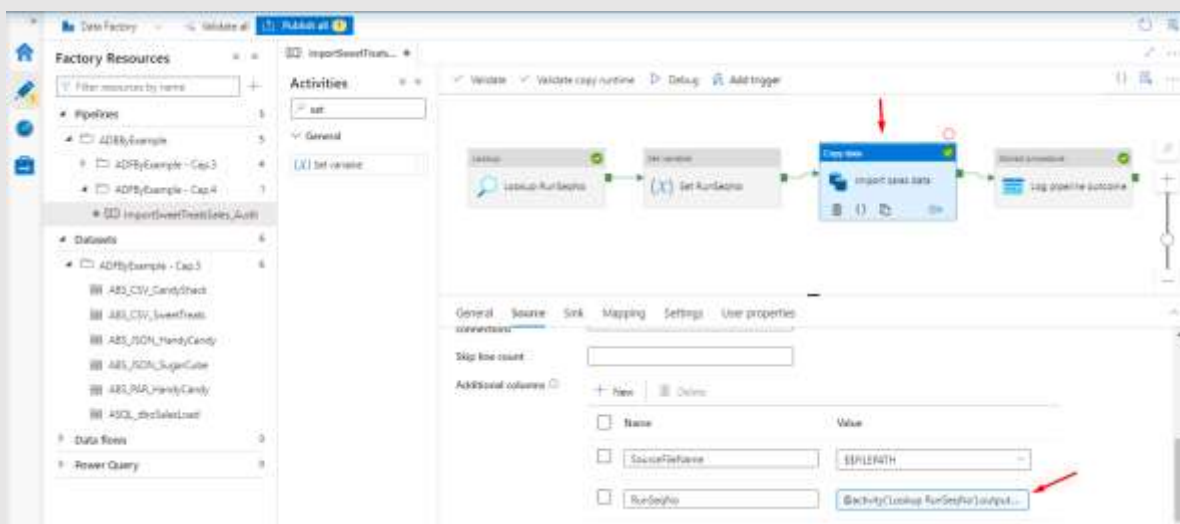
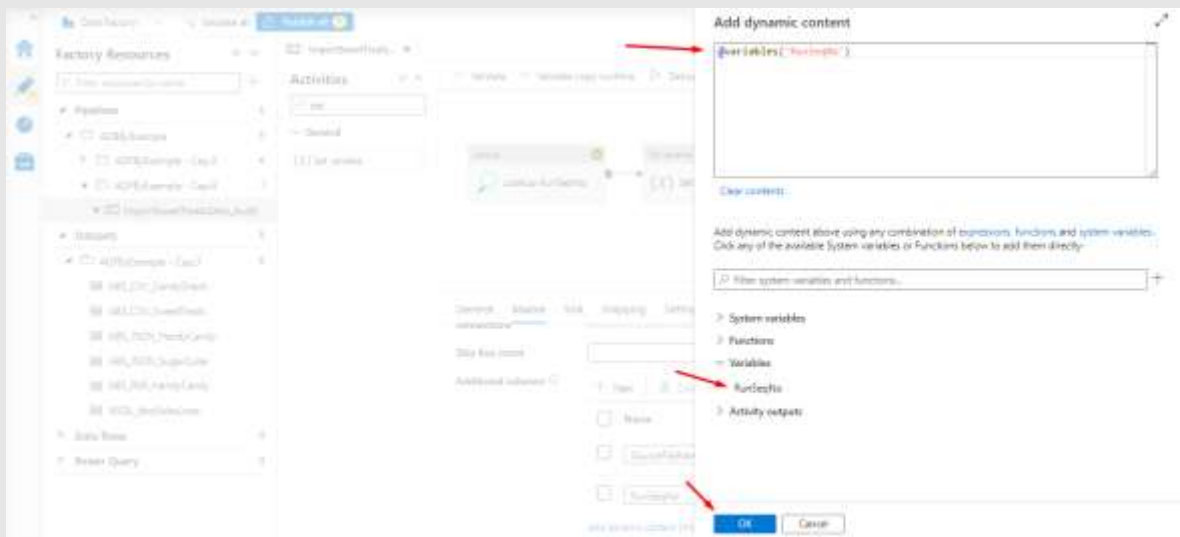


Figura 4-9 Actividad Set variable con expresión de valor y dependencias

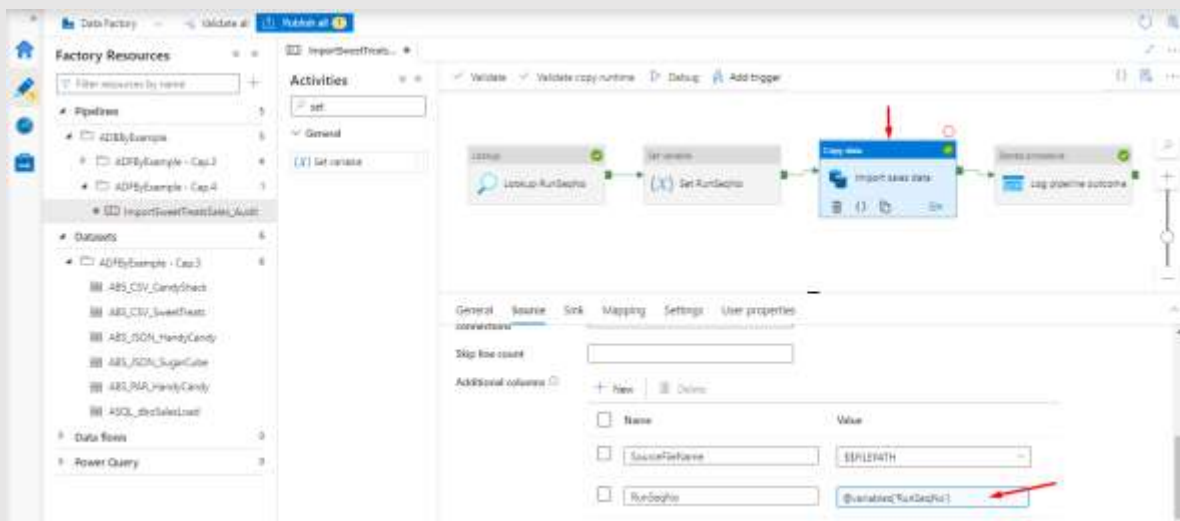
2. La cadena de dependencias significa que la dependencia directa entre las actividades Lookup y Copy data es redundante - **elimínala**.
3. Abra la pestaña Source configuration de la actividad Copy data y abra el generador de expresiones para la columna "RunSeqNo" (en la sección Additional columns).



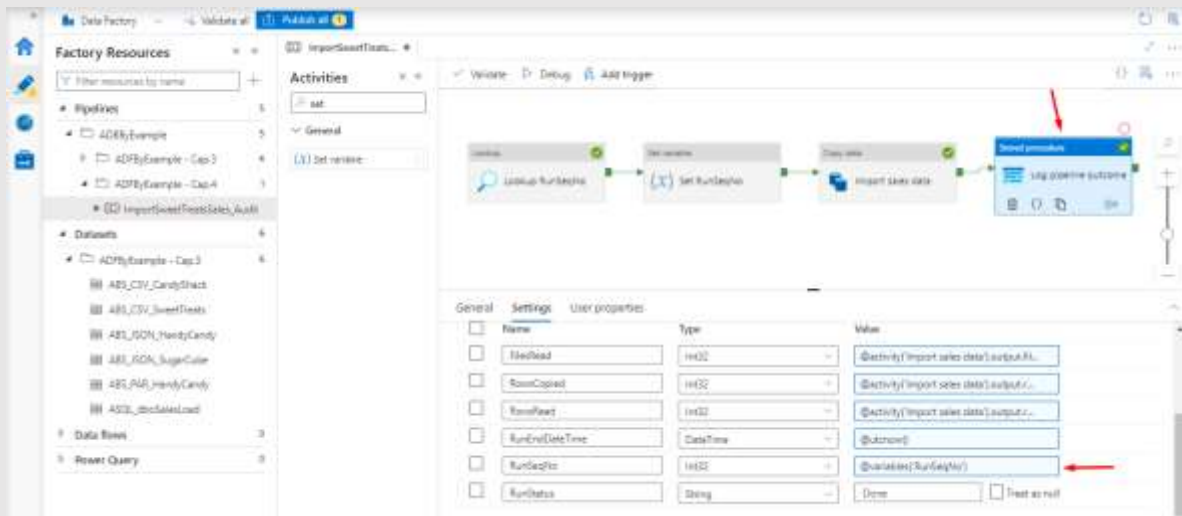
4. Elimine la expresión existente utilizando el enlace Clear contents, luego desplácese hasta la sección Variables que ahora contiene la variable "RunSeqNo". Haz clic en ella para añadirla al panel de expresiones, y luego haz clic en Finalizar.



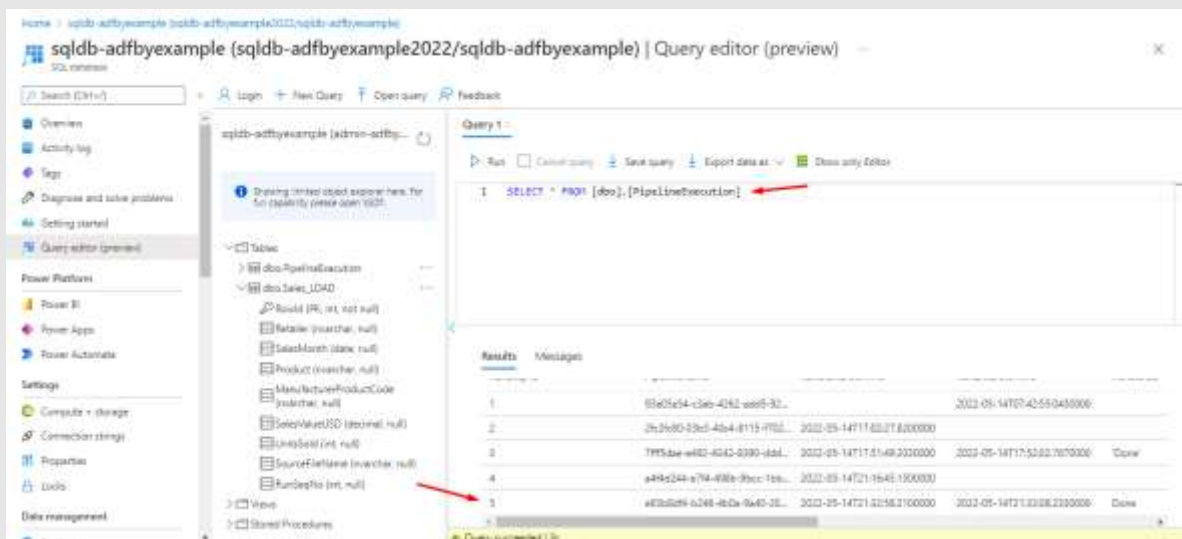
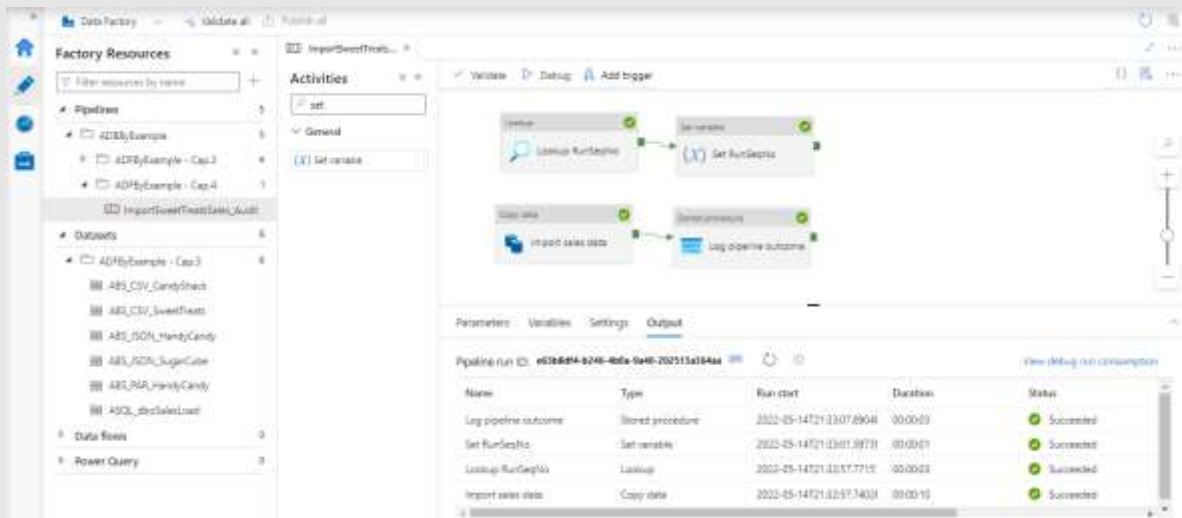
5. A diferencia de las variables del sistema, que utilizan la sintaxis de propiedad pipeline().property, una variable de usuario se refiere utilizando la sintaxis de colección variables('variableName'). El panel de expresión debería contener ahora la expresión @variables('RunSeqNo').



6. Abre la pestaña de configuración de la actividad Stored procedure y actualiza la expresión del parámetro "RunSeqNo" para que también utilice la nueva variable.



7. Por último, ejecute el pipeline y compruebe que sigue funcionando correctamente.



Home > sqldb-adfbyexample (sqldb-adfbyexample2022/sqldb-adfbyexample) | Query editor (preview)

Search (Ctrl+F) Login + New Query Open query Feedback

Overview
Activity log
Tags
Diagnose and solve problems
Getting started
Query editor (preview)

Power Platform
Power BI
Power Apps
Power Automate

Settings
Compute + storage
Connection strings
Properties
Locale
Data management
Environments

sqldb-adfbyexample (admin-adfby...)

Warning: Invalid object specifier 'dbo.Sales_Load'. For full capability please open SSOT.

Tables
dbo Pipeline execution
dbo Sales_Load
Product (nvarchar, null)
SalesMonth (date, null)
SalesValueUSD (decimal, null)
SourceFileName (nvarchar, null)
UnitSold (int, null)
RunSeqNo (int, null)

Views
Stored Procedures

Query 1

Run Cancel query Send query Export data as (View only) Edit

```
SELECT * FROM [dbo].[Sales_Load]
```

Results Messages

unitCode	SalesValueUSD	UnitSold	SourceFileName	RunSeqNo
202201	409	409	Apr-20\Sales.csv	-1
202202	616	616	Jul-20\Sales.csv	-1
202204	718	718	May-20\Sales.csv	-1
202205	78	78	Jun-20\Sales.csv	-1

Query succeeded | 4s

Le borro el valor por defecto que habíamos sugerido anteriormente como “-1” para la variable

Microsoft Azure trainingWebResource

Data Factory Validate all Publish all

Factory Resources

Pipelines
ADFBExample
ADFBExample - Cap.3
ADFBExample - Cap.4
ImportSweetThreatData_Audit

Datasets
ADFBExample - Cap.3
ABS_CSV_CandyShark
ABS_CSV_SweetThreats
ABS_JSON_HandyCandy
ABS_JSON_SuperCandy
ABS_PAW_HandyCandy
ABSQ_WebSalesLoad

Data Store
Power Query

Activities
Set
General
Set variable

Variables
Debug
Add trigger

Lookup RunSeqNo
Set RunSeqNo

Copy data
Import sales data
Log pipeline outcome

Parameters
Variables
Settings
Output

Parameters

Name	Type	Default value
RunSeqNo	String	Value

Data Factory Validate all Publish all

Factory Resources

Pipelines
ADFBExample
ADFBExample - Cap.3
ADFBExample - Cap.4
ImportSweetThreatData_Audit

Datasets
ADFBExample - Cap.3
ABS_CSV_CandyShark
ABS_CSV_SweetThreats
ABS_JSON_HandyCandy
ABS_JSON_SuperCandy
ABS_PAW_HandyCandy
ABSQ_WebSalesLoad

Data Store
Power Query

Activities
Set
General
Set variable

Variables
Debug
Add trigger

Lookup RunSeqNo
Set RunSeqNo

Copy data
Import sales data
Log pipeline outcome

Parameters
Variables
Settings
Output

Pipeline run ID: 35a72652-b4c1-4197-a87c-c6d478c5564c

Name	Type	Run start	Duration	Status	Info
Set RunSeqNo	Set variable	2022-05-14T11:37:42.7677	00:00:01	Succeeded	
Import sales data	Copy data	2022-05-14T11:37:43.6603	00:00:09	Failed	Auto
Lookup RunSeqNo	Lookup	2022-05-14T11:37:43.6603	00:00:00	Succeeded	Auto

Vuelvo a unir la actividad de Set variable con Copy data y se obtiene el resultado esperado.

The screenshot shows the Microsoft Azure Data Factory interface. On the left, the 'Factory Resources' pane lists various pipelines and datasets. The main area displays a pipeline named 'ImportSweetToothSales' with a visual representation of its activities: 'Lookup RunPipeline', 'Set variable', 'Copy data', and 'Log pipeline outcome'. Below the visual, the 'Pipeline run' details are shown for a specific run ID. The 'Output' tab is selected, displaying a table of activity results.

Name	Type	Run start	Duration	Status
Log pipeline outcome	Stored procedure	2022-05-14T15:44:03.4861	00:00:03	Succeeded
Import sales data	Copy data	2022-05-14T15:43:53.3566	00:00:09	Succeeded
Set RunPipeline	Set variable	2022-05-14T15:43:52.7103	00:00:01	Succeeded
Lookup RunPipeline	Lookup	2022-05-14T15:43:49.0676	00:00:03	Succeeded

The screenshot shows the SQL query editor for the 'sqldb-adfbyexample' database. The query is: `SELECT * FROM [dbo].[PipelineExecution]`. The results are displayed in a table with columns: 'id', 'RunID', 'SalesValueUSD', 'UnitSold', 'SourceFileName', and 'RunSeqNo'. A red arrow points to the 'RunSeqNo' column in the results table.

id	RunID	SalesValueUSD	UnitSold	SourceFileName	RunSeqNo
3	7854ee-e802-4542-b990-adb1...	2022-05-14T15:1462000000	2022-05-14T15:02027670000	Done	
4	a44c294-e746-4989-B9cc-70c...	2022-05-14T21:084513000000			
5	e936b84-b246-4b0e-9640-26...	2022-05-14T21:025821000000	2022-05-14T21:030823000000	Done	
6	35e72852-b4c7-4797-ad7c-ea2...	2022-05-14T21:074388000000			
7	1b9e71f-d713-423b-b41d-4f1...	2022-05-14T21:434833000000	2022-05-14T21:440380000000	Done	

The screenshot shows the SQL query editor for the 'sqldb-adfbyexample' database. The query is: `SELECT * FROM [dbo].[Sales_LOAD]`. The results are displayed in a table with columns: 'id', 'RunID', 'SalesValueUSD', 'UnitSold', 'SourceFileName', and 'RunSeqNo'. A red arrow points to the 'RunSeqNo' column in the results table.

id	RunID	SalesValueUSD	UnitSold	SourceFileName	RunSeqNo
1	00544	618	Jul-23/Sales.csv	?	
2	665193	667	Jun-20/Sales.csv	?	
3	162528	494	Sep-23/Sales.csv	?	
4	307331	476	Aug-23/Sales.csv	?	

Consejo A medida que el número de actividades en sus pipelines crece, puede resultar imposible verlas todas en pantalla al mismo tiempo. Puede mover la ventana de visualización del lienzo haciendo clic y arrastrando en el espacio en blanco del lienzo. También puede cambiar el nivel de zoom de la pantalla utilizando las herramientas de visualización en el lado derecho del lienzo.

4.5.4. Array Variables

La decisión de utilizar una variable de tipo "String" en la sección anterior era apropiada porque la actividad Lookup devolvía una sola fila. Si se marca la casilla de verificación "First row only", se aplica este comportamiento (independientemente del número de filas que devuelva el conjunto de datos de origen).

Si no se marca la casilla de verificación, la actividad Lookup devuelve un array JSON llamado value. El array contiene un elemento por cada fila devuelta por el conjunto de datos, hasta un máximo de 5000 filas o 2MB de datos. Para almacenar este valor para reutilizarlo en el pipeline, asígnelo a una variable de tipo "Array".

Un requisito frecuente para los valores de los arrays es la capacidad de iterar sobre ellos, procesando cada elemento individualmente pero de la misma manera. Las actividades de ADF para hacer esto se presentan en el Capítulo 6.

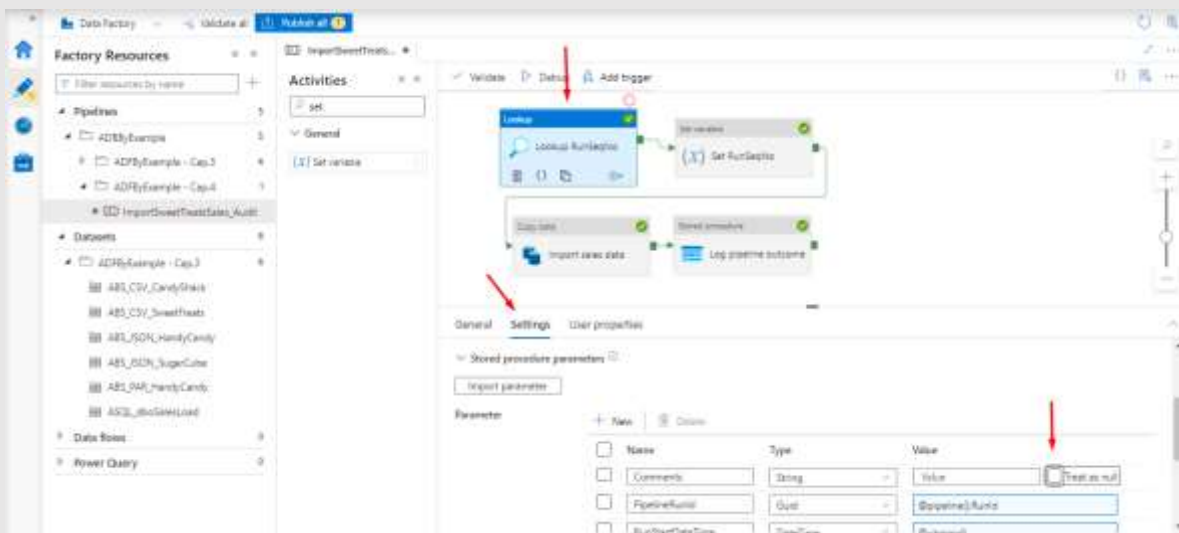
4.6. Concatenar Strings

Al utilizar el constructor de expresiones, habrá observado que existen seis grupos de funciones de expresión:

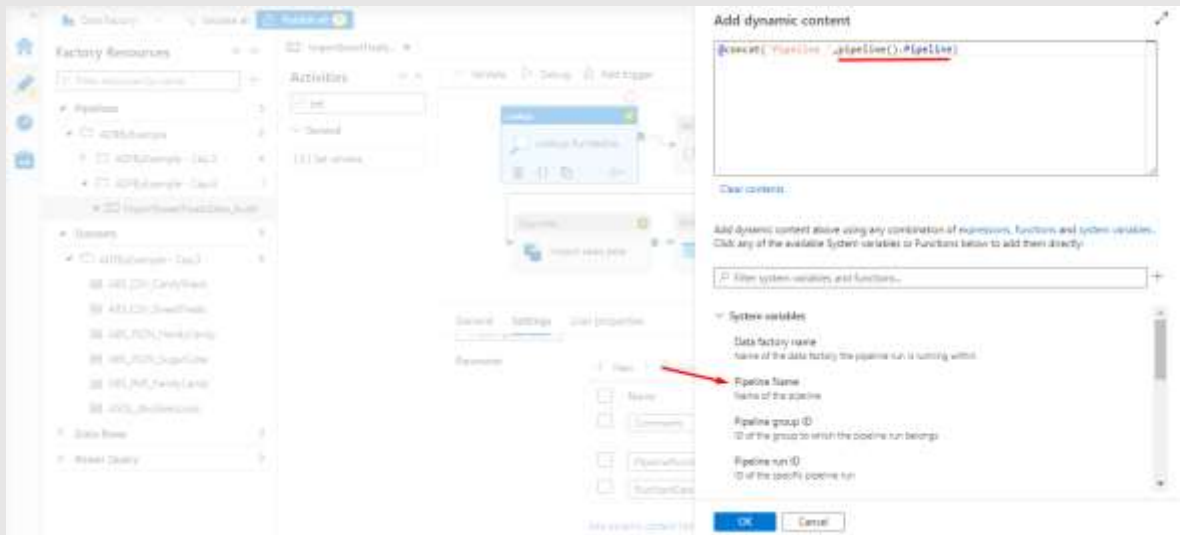
- ❖ **String functions**, que operan con strings (por ejemplo, concat, substring)
- ❖ **Math functions**, que realizan operaciones matemáticas (add, max)
- ❖ **Logical functions**, que devuelven true o false (equals, if)
- ❖ **Date functions**, que devuelven strings que representan fechas (utcnow, addhours) - recuerde que el lenguaje de expresión no tiene tipos de fecha u hora
- ❖ **Collection functions**, que actúan sobre arrays y diccionarios (length, contains)
- ❖ **Conversion functions**, que convierten los valores en diferentes tipos de datos (por ejemplo, string) o representaciones (por ejemplo, base64).

En esta sección, utilizarás la función concat para construir un mensaje concatenando varias strings juntas. El mensaje se registrará utilizando el parámetro "Comments" del procedimiento almacenado [dbo].[LogPipelineStart].

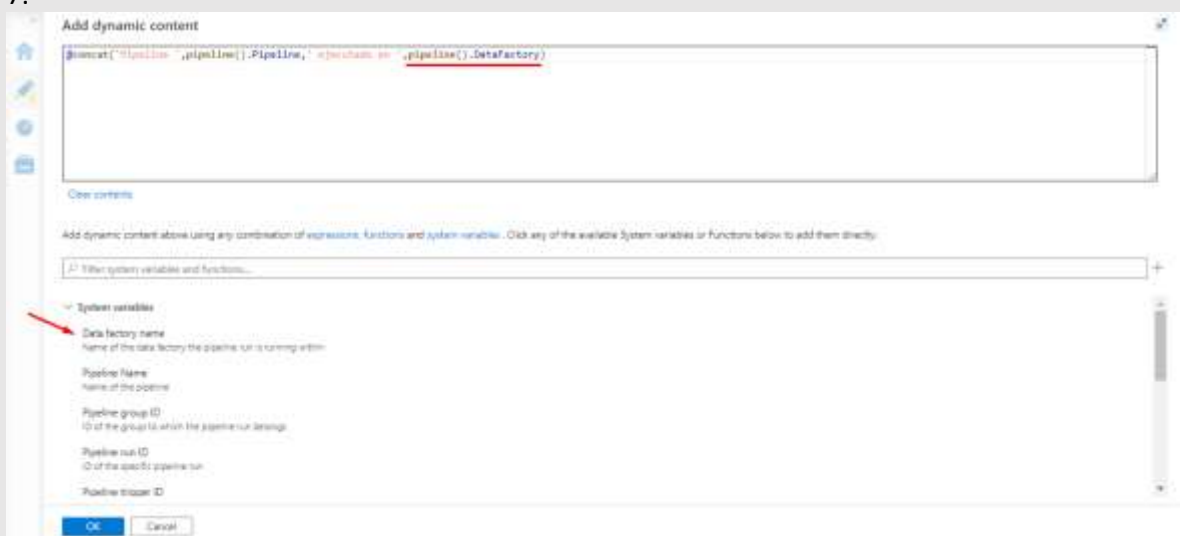
1. En el lienzo de autoría, abra la pestaña de configuración de la actividad Lookup.
2. Desmarque la casilla **Treat as null** del parámetro "Comments" y abra el constructor de expresiones para establecer su valor.

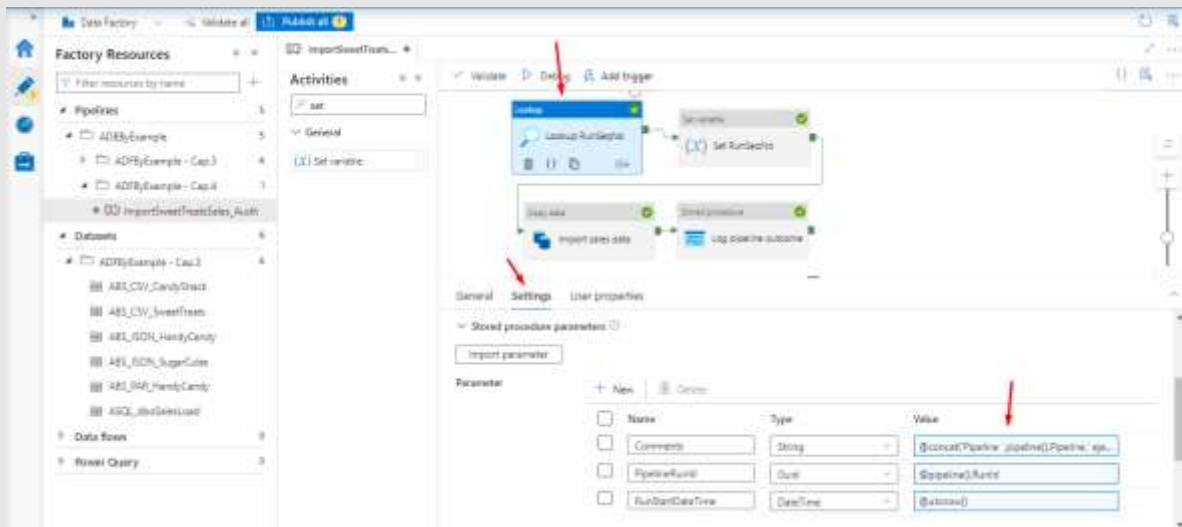


- Haga clic en la función concat -que se encuentra en la sección String functions- para añadirla al panel de expresión.
- La función toma una lista de argumentos de string separados por comas. Los valores literales de las strings están rodeados por comillas simples. Introduzca el primer argumento 'Pipeline ' (incluyendo un espacio final después de la palabra "Pipeline"). La expresión debe ser ahora @concat('Pipeline ').
- Añada un segundo argumento insertando una coma después del primero y seleccionando **Pipeline name** en la lista de variables del sistema.

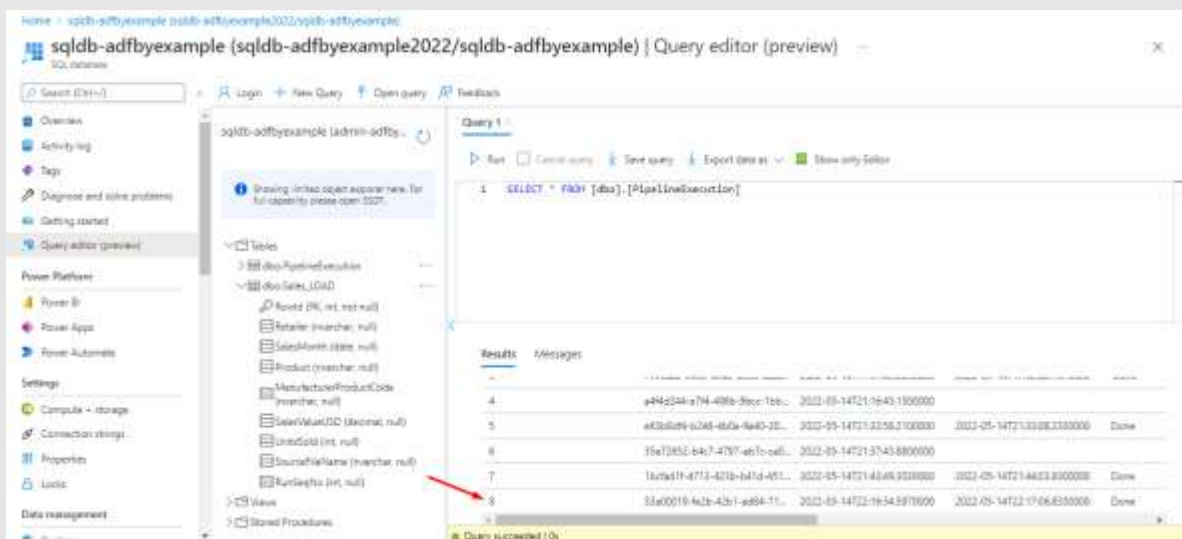
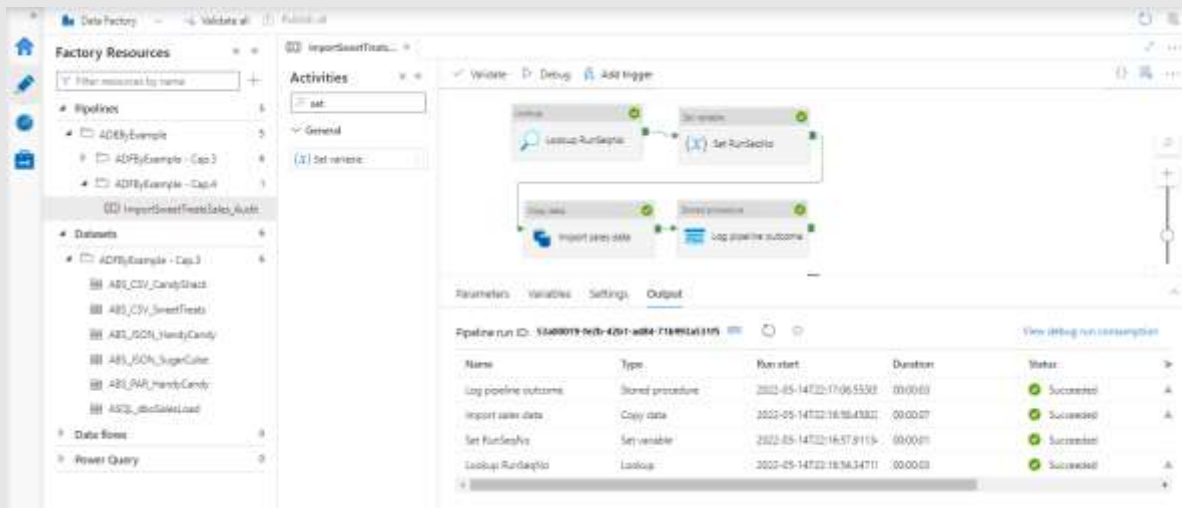


- Añade un tercer argumento string literal ' ejecutado en ' y un cuarto seleccionando la variable del sistema Data factory name. La expresión completa debe ser ahora **@concat('Pipeline ', pipeline().Pipeline, ' ejecutado en ', pipeline().DataFactory)**. Haga clic en Finalizar y ejecute el pipeline.
-





Cuando la ejecución del pipeline se haya completado, inspeccione el último registro en la tabla [dbo].[PipelineExecution] - su campo [Comments] debería contener algo como "Pipeline ImportSweetTreatsSales_Audit executed in MyDataFactory" (dependiendo del nombre de su instancia de ADF).



Query editor (preview)

Results

Pipeline	RowId	RowIdCopied	Comments
Pipeline ImportOverheadSales_Audit ejecutado en traininglakehouseprod	9	994	

Query succeeded | 0s

4.6.1. Infix Operators

El lenguaje de expresión del ADF no contiene operadores infix - todas las operaciones se implementan como funciones. De la misma manera que se utiliza `concat('a','b')` (en lugar de algo como `'a' + 'b'`), todas las operaciones matemáticas se invocan utilizando funciones que se encuentran en la sección de funciones matemáticas del constructor de expresiones, por ejemplo:

- ❖ Para calcular `1 + 2`, hay que utilizar la función `add`: `add(1,2)`
- ❖ La resta, la multiplicación y la división se implementan mediante las funciones `sub`, `mul` y `div`, respectivamente.

4.6.2. String Interpolation

Otra forma de construir expresiones de string es utilizar la interpolación de string. Un string interpolado es un valor de string que contiene expresiones de marcador de posición que se evalúan en tiempo de ejecución. Una expresión de marcador de posición está contenida entre llaves y precedida por el símbolo `@` de la siguiente manera `@{placeholderExpression}`.

Las strings interpoladas son más legibles que las construidas con la función `concat` - el equivalente de la llamada a la función que construiste en la sección anterior es

Pipeline `@{pipeline().Pipeline}` ejecutada en `@{pipeline().DataFactory}`

Las strings interpoladas no pueden anidarse dentro de otras expresiones. Si una string interpolada aparece como argumento de una función, sus expresiones de marcador de posición no se evalúan, sino que se tratan como literales de string.

4.6.3. Escapar @

El símbolo @ inicial es necesario para indicar al ADF que lo que sigue debe ser evaluado como una expresión. Si necesita especificar un string literal que comience con el carácter @, debe escaparlo usándolo dos veces:

@add(1,2) es una expresión que se evalúa como 3.

@@add(1,2) es el string literal "@add(1,2)".

4.7. Repaso del capítulo

Las expresiones permiten especificar muchas propiedades de los recursos de fábrica de forma dinámica - los valores de dichas propiedades se evalúan en tiempo de ejecución utilizando la expresión configurada. Puede utilizar una expresión siempre que aparezca el enlace Añadir contenido dinámico [Alt+Mayúsculas+D], que le permite lanzar el constructor de expresiones.

Una expresión puede ser un string literal (que puede contener expresiones de marcador de posición), una variable o una función. El anidamiento de funciones le permite construir expresiones complejas y potentes.

Las variables de usuario se crean a nivel del pipeline y son accesibles a todas las actividades mientras dure la ejecución del pipeline. Utilice la actividad Set variable para cambiar el valor predeterminado de una variable o un valor previamente establecido. La actividad Append variable le permite añadir elementos al final del valor de una variable Array.

4.7.1. Conceptos clave

- ❖ **Expresión**: Una expresión se evalúa en el momento de la ejecución del pipeline para determinar el valor de una propiedad. El tipo de datos de una expresión es string, integer, float, boolean, array, o dictionary.
- ❖ **Array**: Una colección de múltiples valores denominados elementos. Los elementos son dirigidos por un índice entero entre cero y uno menos que la longitud de la matriz.
- ❖ **Diccionario**: Una colección cuyos elementos son referidos por un nombre.
- ❖ **Constructor de expresiones**: Un editor de expresiones integrado en el ADF UX.
- ❖ **Variable de sistema**: Las variables del sistema proporcionan acceso a los valores en tiempo de ejecución de varias propiedades del sistema.
- ❖ **Variable de usuario**: Creadas para almacenar valores de string, booleanos o de array durante la ejecución de un pipeline.
- ❖ **Función de expresión**: Una de las bibliotecas de funciones disponibles para su uso en expresiones. Los tipos de funciones incluyen String, Math, Logical, Date, Collection y Type conversions.
- ❖ **Interpolación de string**: Un string literal que contiene expresiones de marcador de posición.
- ❖ **Expresión de marcador de posición (Placeholder expression)**: Una expresión incrustada en un string interpolado, evaluada en tiempo de ejecución para devolver un string.

- ❖ **Escapes:** Los literales de string que comienzan con el carácter @ deben ser escapados para evitar su interpretación como expresiones. La @ se escapa siguiéndola con un segundo carácter @.
- ❖ **Actividad de procedimiento almacenado:** Actividad del ADF que permite la ejecución de un procedimiento almacenado de la base de datos, especificando los valores de los parámetros del procedimiento almacenado según sea necesario.
- ❖ **Actividad de Lookup:** Actividad del ADF que devuelve una o varias filas de un conjunto de datos para su uso durante la ejecución de un pipeline. En el caso de los conjuntos de datos de SQL Server, las filas pueden ser devueltas desde tablas, vistas, consultas en línea o procedimientos almacenados.
- ❖ **Actividad Set variable:** Actividad del ADF utilizada para actualizar el valor de una variable de usuario.
- ❖ **Actividad Append variable:** Actividad del ADF utilizada para añadir un nuevo elemento al final del valor de una variable Array existente.
- ❖ **Actividad de dependencia:** Restricción utilizada para controlar el orden de ejecución de las actividades de un pipeline.
- ❖ **Activity output object:** Objeto JSON producido por la ejecución de una actividad del ADF. Un objeto de salida y sus propiedades están disponibles para cualquier actividad dependiente de la actividad de origen, ya sea directa o indirectamente.
- ❖ **Punto de interrupción (Breakpoint):** Un punto de interrupción de ADF UX permite ejecutar un pipeline, en modo de depuración, hasta la actividad en la que se establece el punto de interrupción (los puntos de interrupción no existen en los pipelines publicados). A diferencia de otros IDEs, no es posible reanudar la ejecución después de alcanzar un punto de interrupción.
- ❖ **\$\$FILEPATH:** Una variable de sistema reservada que permite a la actividad Copiar datos etiquetar los datos de archivo entrantes con su archivo de origen. \$\$FILEPATH está disponible únicamente para rellenar columnas adicionales en la actividad Copiar datos y no puede utilizarse en expresiones.
- ❖ **\$\$COLUMN:** Variable de sistema reservada que permite a la actividad Copiar datos duplicar una columna específica en los datos entrantes. \$\$COLUMN está disponible únicamente para rellenar columnas adicionales en la actividad de copia de datos y no puede utilizarse en expresiones.
- ❖ **Columnas adicionales:** Una fuente de actividad de datos de copia puede aumentarse con columnas adicionales, cuyos valores se especifican mediante una expresión, \$\$FILEPATH, \$\$COLUMN o un valor estático codificado.

- ❖ **Seguimiento del linaje (Lineage tracking)**: La práctica de etiquetar los datos a medida que se procesan, para permitir la identificación posterior de la información relacionada con su origen y/o procesamiento.

4.7.2. Para los desarrolladores de SSIS

Este capítulo ha introducido dos nuevas actividades que tienen funciones similares a las actividades SSIS. La correspondencia no es exacta:

- ❖ La actividad Stored procedure permite ejecutar un procedimiento almacenado. A diferencia de la tarea Execute SQL Task de SSIS, no puede utilizarla para ejecutar código SQL arbitrario o para devolver un conjunto de resultados.
- ❖ La actividad Lookup le permite cargar pequeños conjuntos de datos. Con una conexión SQL, esto es similar al uso de una Tarea de Ejecución SQL para ejecutar código SQL para devolver un conjunto de resultados, pero la actividad también puede ser usada para cargar conjuntos de datos de diferentes tipos.

El orden de ejecución de las actividades se controla mediante las dependencias de la actividad. Estas son conceptualmente similares a las restricciones de precedencia de SSIS, pero tienen algunas diferencias importantes que se discuten en el Capítulo 6.

En comparación con SSIS en Visual Studio, el soporte de ADF UX para los puntos de interrupción es limitado. Todas las funciones de expresión utilizan la sintaxis functionName(arguments) - los operadores infijos como "+" utilizados en las expresiones SSIS no están soportados.