# Consultar archivos utilizando un grupo SQL sin servidor

SQL es probablemente el lenguaje más utilizado para trabajar con datos en el mundo. La mayoría de los analistas de datos dominan el uso de consultas SQL para recuperar, filtrar y agregar datos, más comúnmente en bases de datos relacionales. A medida que las organizaciones aprovechan cada vez más el almacenamiento de archivos escalable para crear lagos de datos, SQL sigue siendo a menudo la opción preferida para consultar los datos. **Azure Synapse Analytics** proporciona grupos de SQL sin servidor que le permiten desacoplar el motor de consultas SQL del almacenamiento de datos y ejecutar consultas en archivos de datos en formatos de archivo comunes, como texto delimitado y Parquet.

Esta práctica de laboratorio tardará aproximadamente **40** minutos en completarse.

## Antes de que empieces

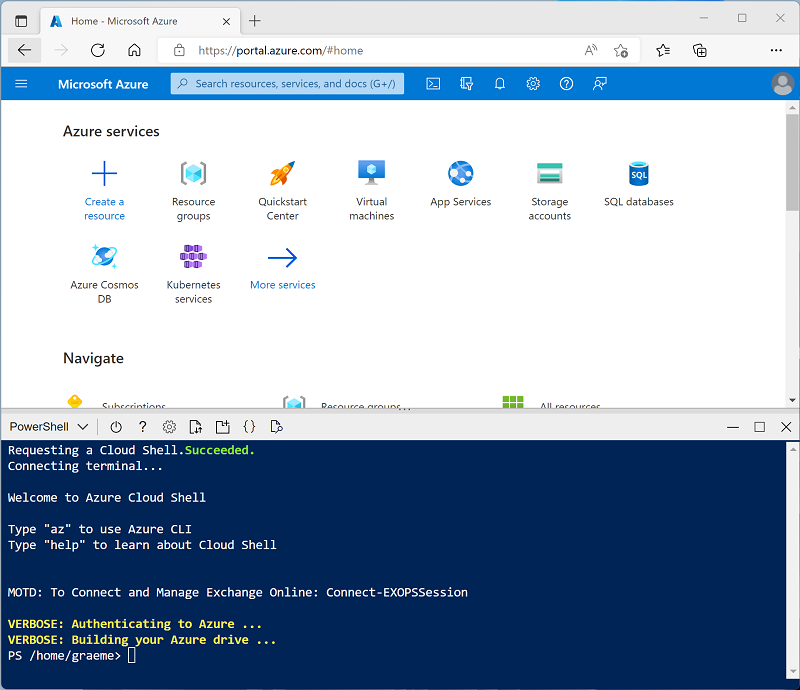
Necesitará una [suscripción de Azure](https://azure.microsoft.com/free) a la que tenga acceso de nivel administrativo.

## Aprovisionar un área de trabajo de Azure Synapse Analytics

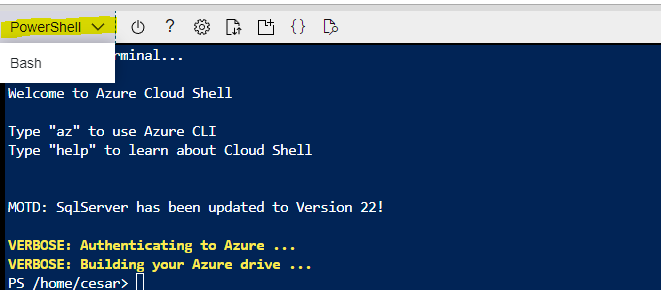
Necesitará un área de trabajo de Azure Synapse Analytics con acceso al almacenamiento del lago de datos. Puede utilizar el grupo SQL sin servidor integrado para consultar archivos en el lago de datos.

En este ejercicio, utilizará una combinación de un script de PowerShell y una plantilla de ARM para aprovisionar un área de trabajo de Azure Synapse Analytics.

1. Inicie sesión en [Azure Portal](https://portal.azure.com/) en https://portal.azure.com.
2. Utilice el botón **[>\_]** a la derecha de la barra de búsqueda en la parte superior de la página para crear un nuevo Cloud Shell en Azure Portal, seleccionando un entorno **de PowerShell** y creando almacenamiento si se le solicita. Cloud Shell proporciona una interfaz de línea de comandos en un panel en la parte inferior de Azure Portal, como se muestra aquí:

[](https://github.com/MicrosoftLearning/DP-500-Azure-Data-Analyst/blob/main/Instructions/images/cloud-shell.png)

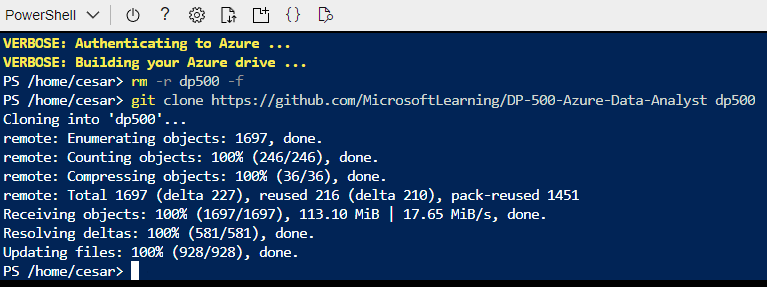
**Nota** : si anteriormente creó un shell de nube que usa un entorno Bash , use el menú desplegable en la parte superior izquierda del panel de shell de nube para cambiarlo a **PowerShell** .



1. Tenga en cuenta que puede cambiar el tamaño del shell de la nube arrastrando la barra separadora en la parte superior del panel o usando los íconos **—** , **◻** y **X** en la parte superior derecha del panel para minimizar, maximizar y cerrar el panel. Para obtener más información sobre el uso de Azure Cloud Shell, consulte la [documentación de Azure Cloud Shell](https://docs.microsoft.com/azure/cloud-shell/overview) .
2. En el panel de PowerShell, ingrese manualmente los siguientes comandos para clonar este repositorio:

rm -r dp500 -f

git clone https://github.com/MicrosoftLearning/DP-500-Azure-Data-Analyst dp500



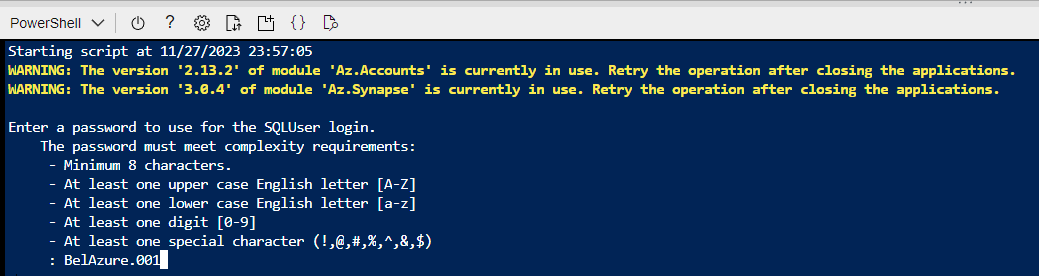
1. Una vez clonado el repositorio, ingrese los siguientes comandos para cambiar a la carpeta de esta práctica de laboratorio y ejecutar el script **setup.ps1** que contiene:

cd dp500/Allfiles/01

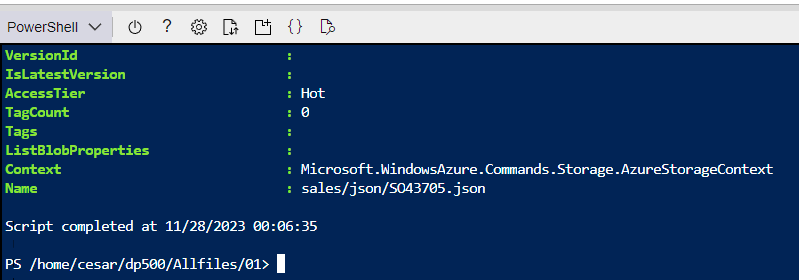
./setup.ps1

1. Si se le solicita, elija qué suscripción desea usar (esto solo sucederá si tiene acceso a varias suscripciones de Azure).
2. Cuando se le solicite, ingrese una contraseña adecuada que se establecerá para su grupo de SQL de Azure Synapse.

**Nota** : ¡Asegúrese de recordar esta contraseña!



1. Espere a que se complete el script; esto suele tardar unos 10 minutos, pero en algunos casos puede tardar más. Mientras espera, revise el artículo [Grupo de SQL sin servidor en Azure Synapse Analytics](https://docs.microsoft.com/azure/synapse-analytics/sql/on-demand-workspace-overview) en la documentación de Azure Synapse Analytics.

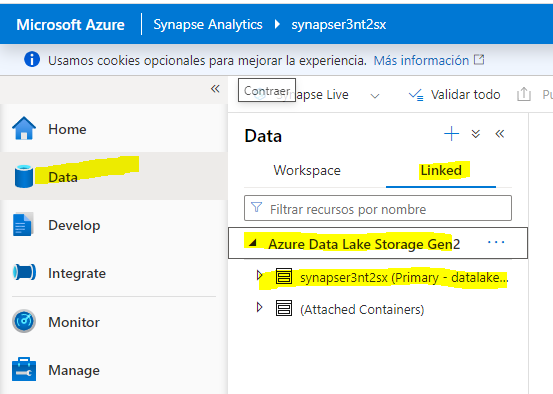


## Consultar datos en archivos.

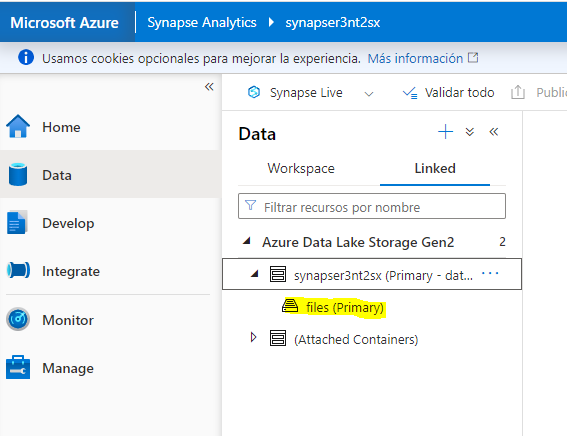
El script aprovisiona un área de trabajo de Azure Synapse Analytics y una cuenta de Azure Storage para hospedar el lago de datos y luego carga algunos archivos de datos en el lago de datos.

### Ver archivos en el lago de datos

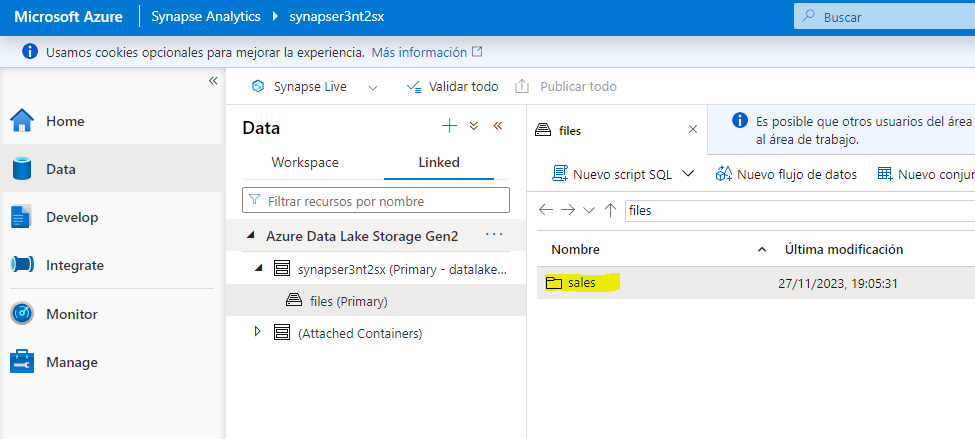
1. Una vez completado el script, en Azure Portal, vaya al grupo de recursos **dp500-**xxxxxxx que creó y seleccione su área de trabajo de Synapse.
2. En la página **Descripción general** de su espacio de trabajo de Synapse, en la tarjeta **Abrir Synapse Studio , seleccione Abrir** para abrir Synapse Studio en una nueva pestaña del navegador; iniciar sesión si se le solicita.
3. En el lado izquierdo de Synapse Studio, use el ícono **››** para expandir el menú; esto revela las diferentes páginas dentro de Synapse Studio que usará para administrar recursos y realizar tareas de análisis de datos.
4. En la página **Data** , vea la pestaña **Linked** y verifique que su espacio de trabajo incluya un vínculo a su cuenta de almacenamiento de Azure Data Lake Storage Gen2, que debe tener un nombre similar a **synapse**xxxxxxx**(Primario - datalake**xxxxxxx**)** .



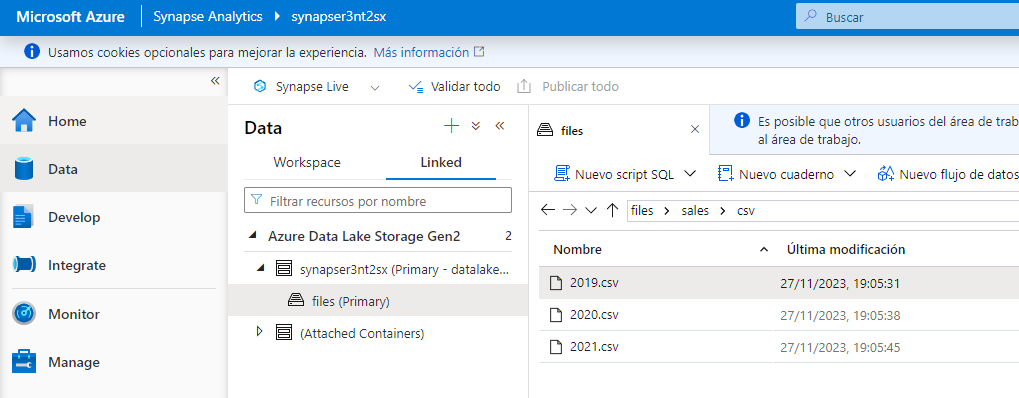
1. Expanda su cuenta de almacenamiento y verifique que contenga un contenedor de sistema de archivos llamado **files** .



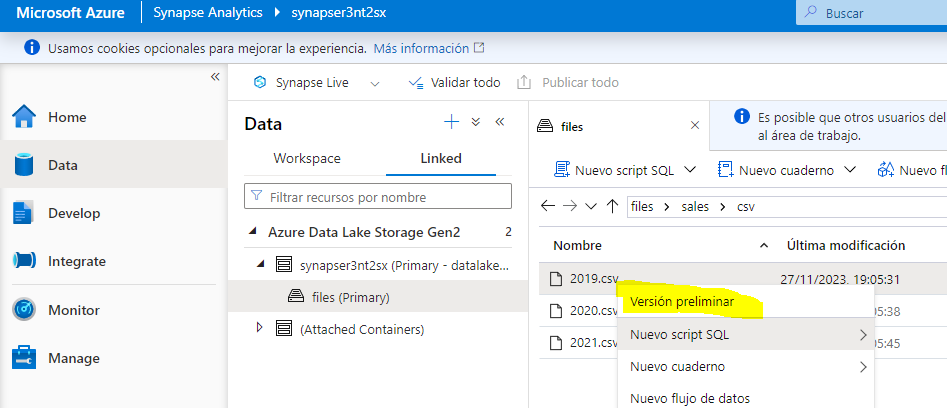
1. Seleccione el contenedor **de Files** y observe que contiene una carpeta llamada **Sales**. Esta carpeta contiene los archivos de datos que va a consultar.

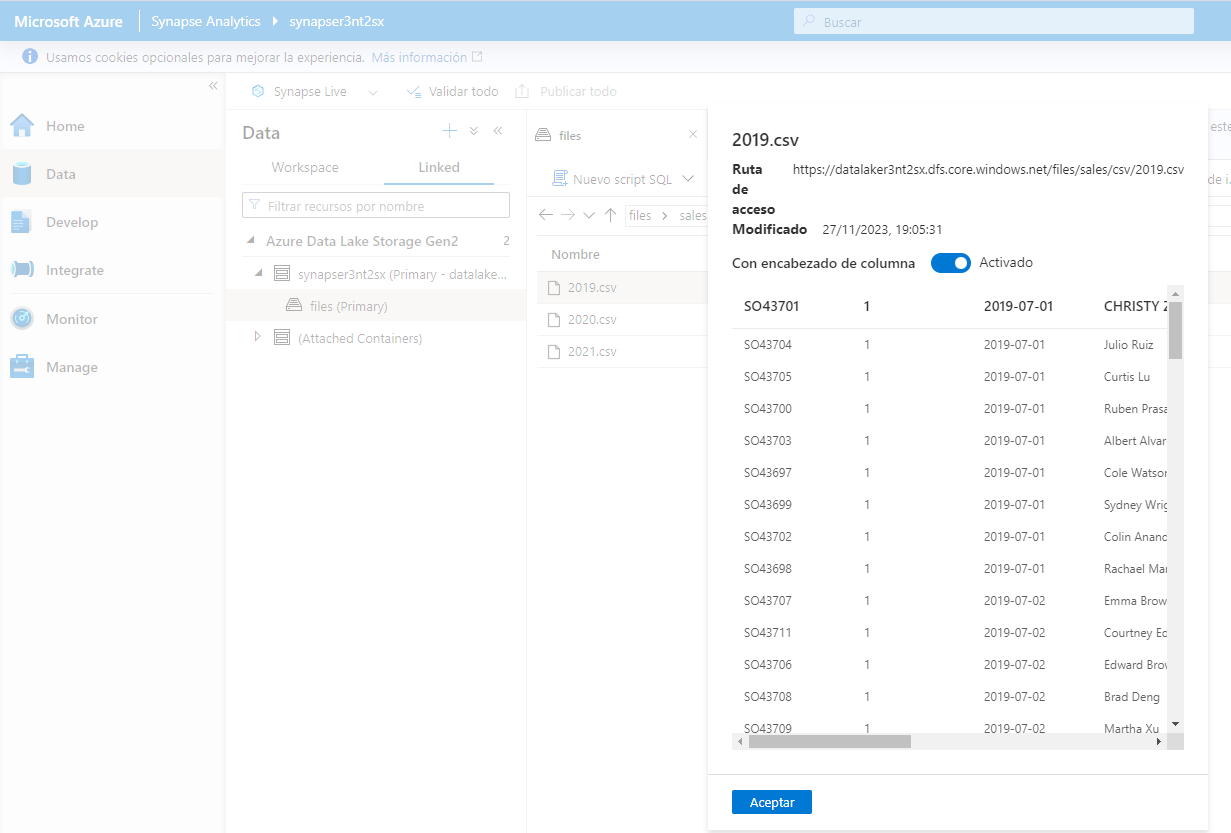


1. Abra la carpeta **sales**y la carpeta **csv** que contiene, y observe que esta carpeta contiene archivos .csv para tres años de datos de ventas.

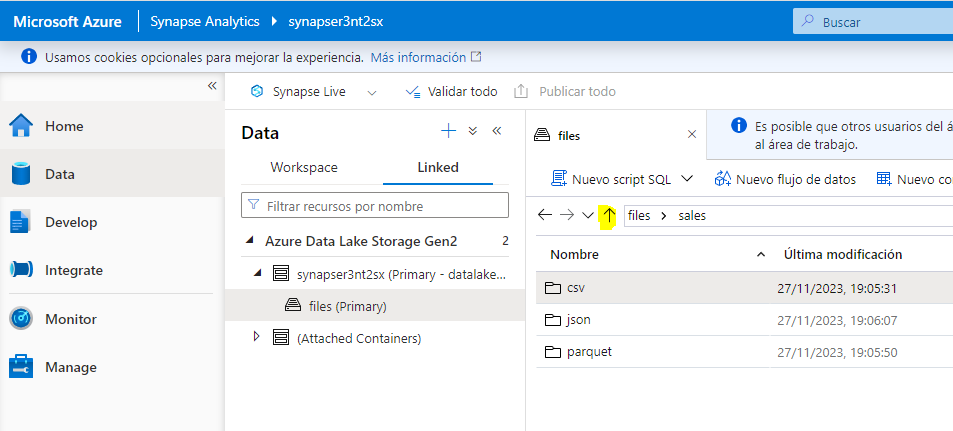


1. Haga clic derecho en cualquiera de los archivos y seleccione **Vista previa** para ver los datos que contiene. Tenga en cuenta que los archivos no contienen una fila de encabezado, por lo que puede anular la selección de la opción para mostrar encabezados de columna.

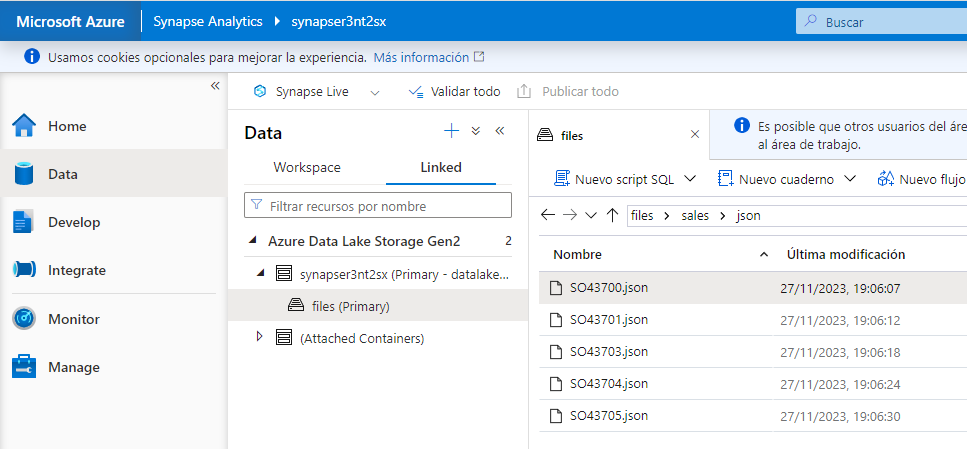


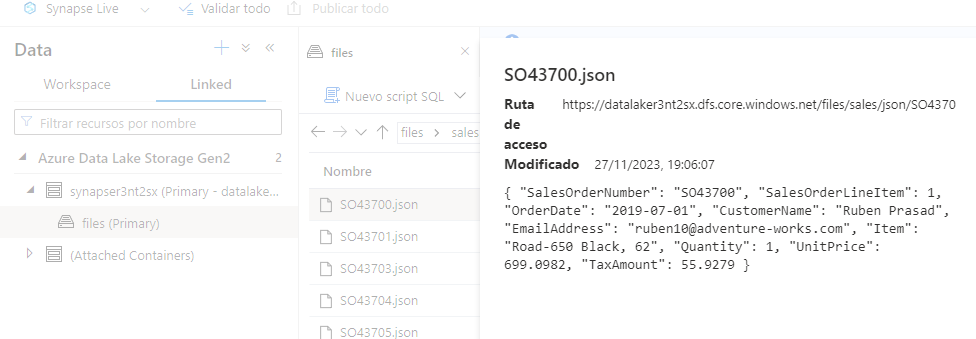


1. Cierre la vista previa y luego use el botón **↑** para regresar a la carpeta **de ventas** .

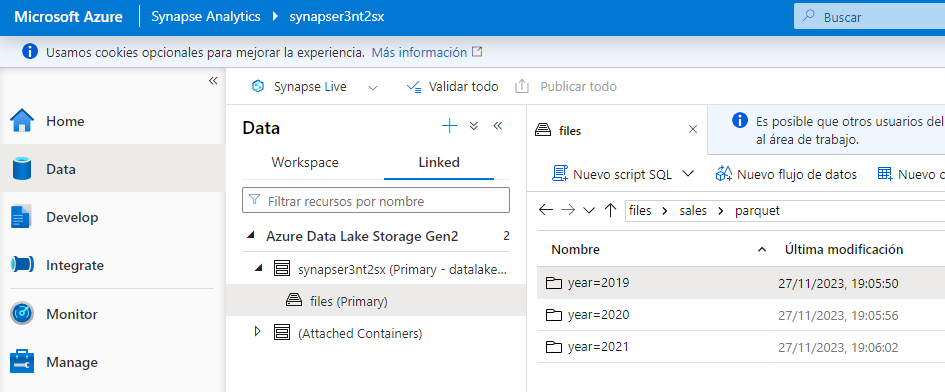


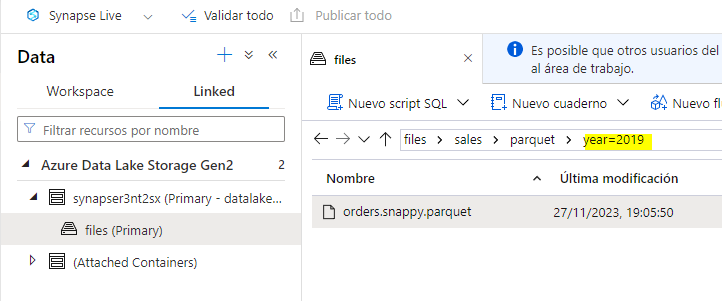
1. En la carpeta **de ventas** , abra la carpeta **json** y observe que contiene algunos pedidos de venta de muestra en archivos .json. Obtenga una vista previa de cualquiera de estos archivos para ver el formato JSON utilizado para un pedido de ventas.





1. Cierre la vista previa y luego use el botón **↑** para regresar a la carpeta **de ventas** .
2. En la carpeta **de ventas** , abra la carpeta **parquet** y observe que contiene una subcarpeta para cada año (2019-2021), en cada una de las cuales un archivo llamado **pedidos.snappy.parquet** contiene los datos de los pedidos de ese año.

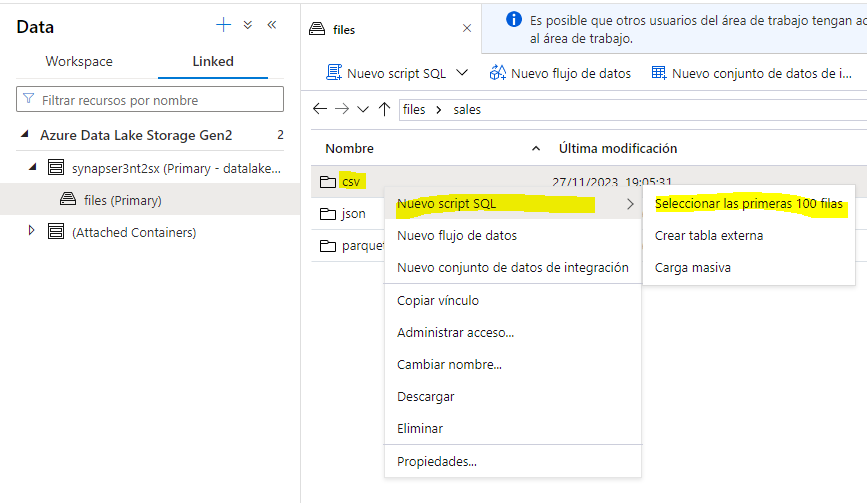




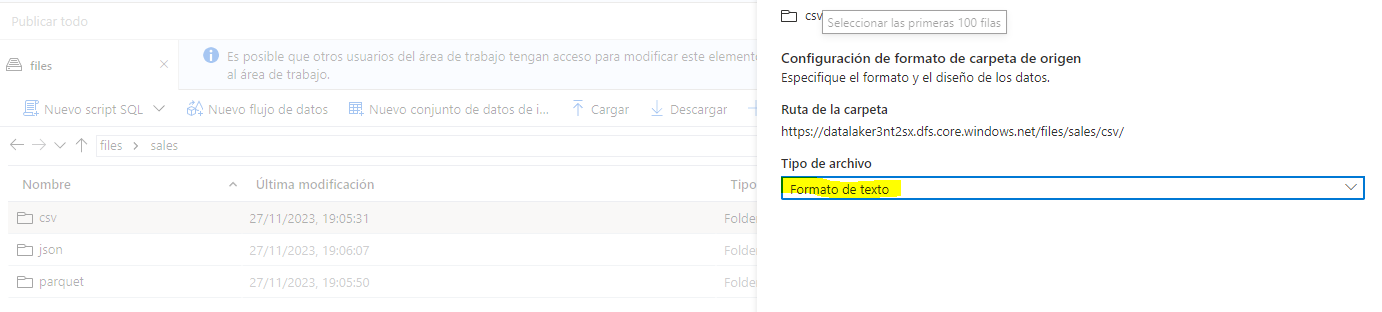
1. Return to the **sales** folder so you can see the **csv**, **json**, and **parquet** folders.

### Use SQL to query CSV files

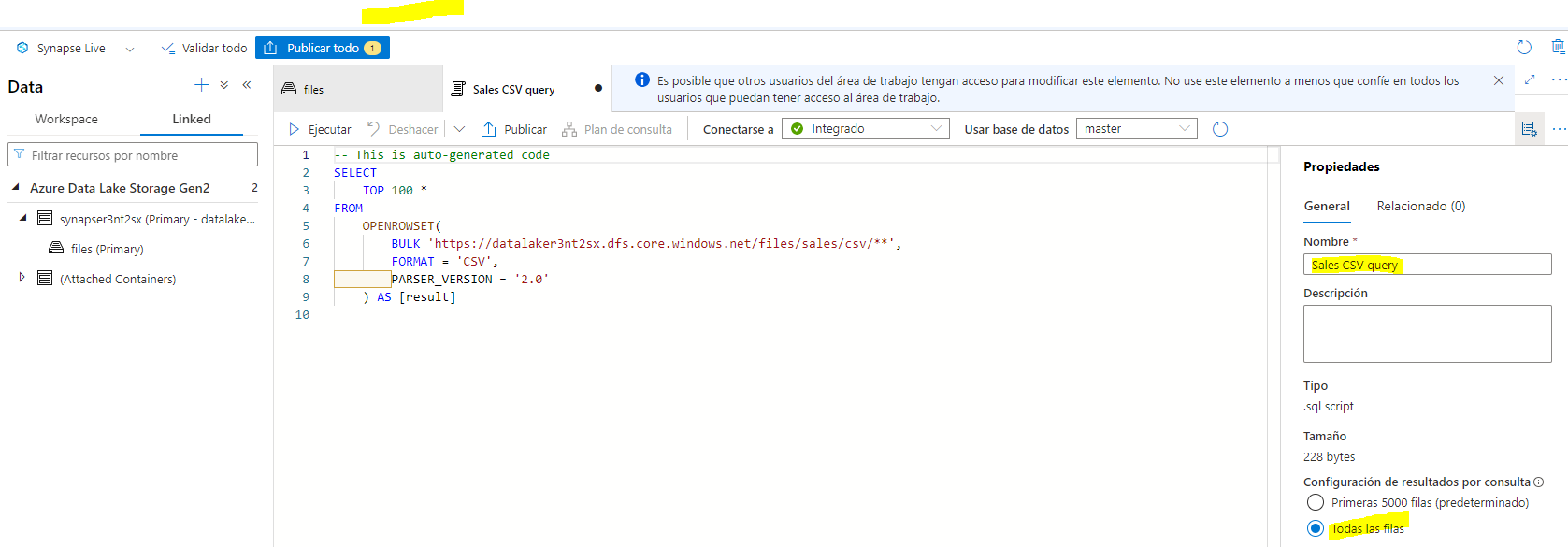
1. Select the **csv** folder, and then in the **New SQL script** list on the toolbar, select **Select TOP 100 rows**.



1. In the **File type** list, select **Text format**, and then apply the settings to open a new SQL script that queries the data in the folder.



1. In the **Properties** pane for **SQL Script 1** that is created, change the name to **Sales CSV query**, and change the result settings to show **All rows**. Then in the toolbar, select **Publish** to save the script and use the **Properties** button (which looks similar to **🗏.**) on the right end of the toolbar to hide the **Properties** pane.



1. Review the SQL code that has been generated, which should be similar to this:

-- This is auto-generated codeSELECT

    TOP 100 \*FROM

    OPENROWSET(

        BULK 'https://datalakexxxxxxx.dfs.core.windows.net/files/sales/csv/',

        FORMAT = 'CSV',

        PARSER\_VERSION='2.0'

) AS [result]

This code uses the OPENROWSET to read data from the CSV files in the sales folder and retrieves the first 100 rows of data.

Mi codigo:

-- This is auto-generated code

SELECT

    TOP 100 \*

FROM

    OPENROWSET(

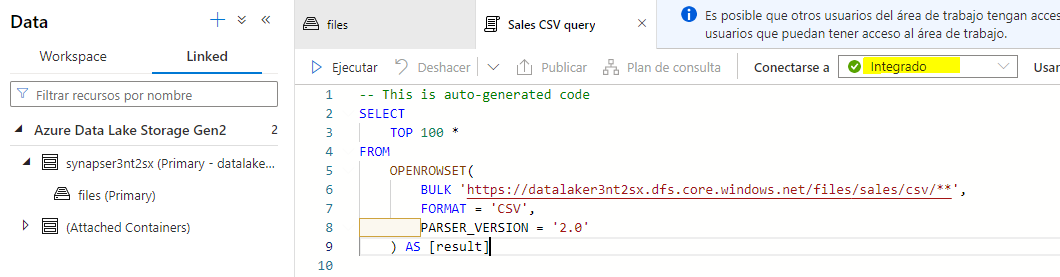
        BULK 'https://datalaker3nt2sx.dfs.core.windows.net/files/sales/csv/\*\*',

        FORMAT = 'CSV',

        PARSER\_VERSION = '2.0'

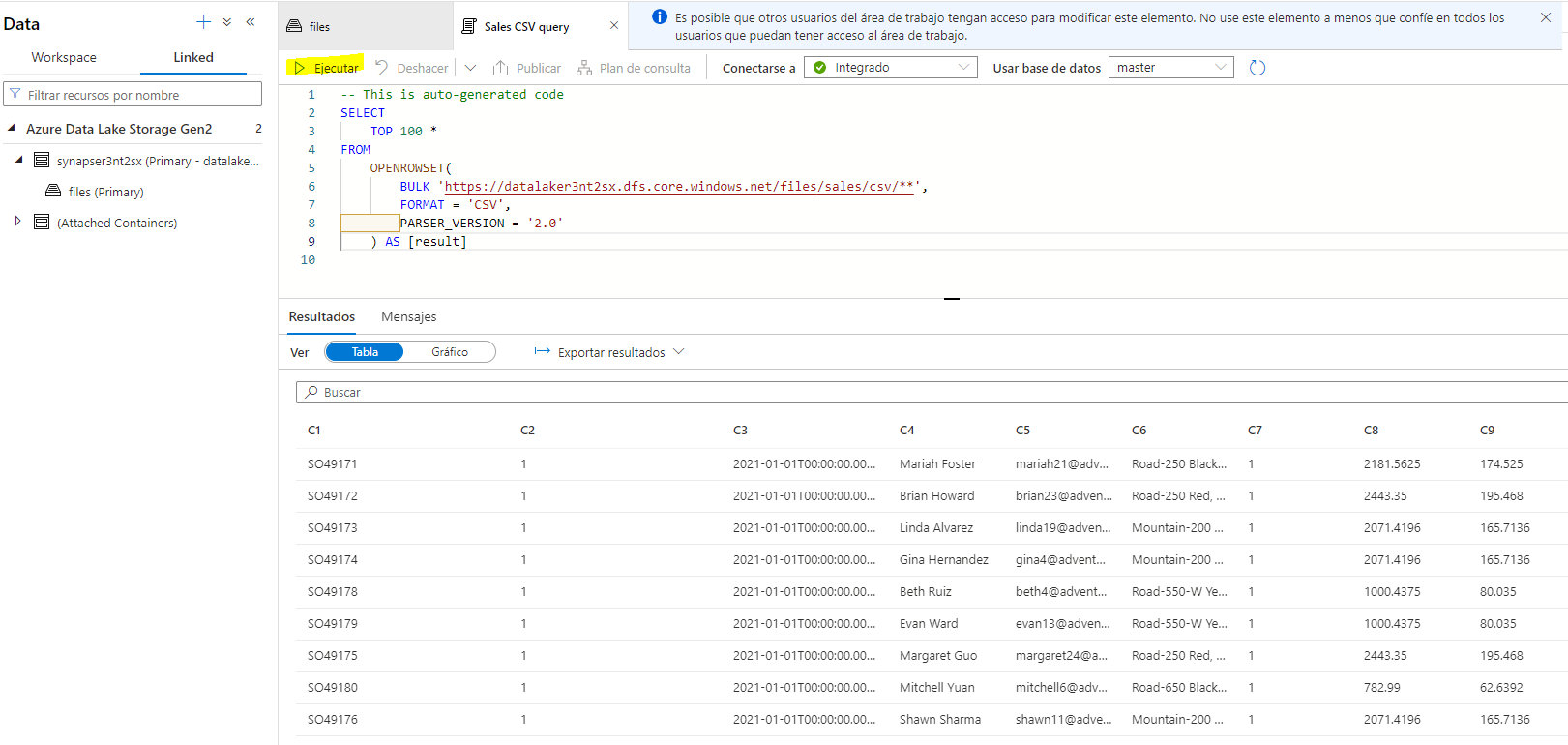
    ) AS [result]

1. In the **Connect to** list, ensure **Built-in** is selected - this represents the built-in SQL Pool that was created with your workspace.



1. On the toolbar, use the **▷ Run** button to run the SQL code, and review the results, which should look similar to this:

| **C1** | **C2** | **C3** | **C4** | **C5** | **C6** | **C7** | **C8** | **C9** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SO45347 | 1 | 2020-01-01 | Clarence Raji | [clarence35@adventure-works.com](mailto:clarence35@adventure-works.com) | Road-650 Black, 52 | 1 | 699.0982 | 55.9279 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |



1. Note the results consist of columns named C1, C2, and so on. In this example, the CSV files do not include the column headers. While it's possible to work with the data using the generic column names that have been assigned, or by ordinal position, it will be easier to understand the data if you define a tabular schema. To accomplish this, add a WITH clause to the OPENROWSET function as shown here (replacing datalakexxxxxxx with the name of your data lake storage account), and then rerun the query:

SELECT

    TOP 100 \*FROM

    OPENROWSET(

        BULK 'https://datalakexxxxxxx.dfs.core.windows.net/files/sales/csv/',

        FORMAT = 'CSV',

        PARSER\_VERSION='2.0'

)

WITH (

SalesOrderNumber VARCHAR(10) COLLATE Latin1\_General\_100\_BIN2\_UTF8,

SalesOrderLineNumber INT,

OrderDate DATE,

CustomerName VARCHAR(25) COLLATE Latin1\_General\_100\_BIN2\_UTF8,

EmailAddress VARCHAR(50) COLLATE Latin1\_General\_100\_BIN2\_UTF8,

Item VARCHAR(30) COLLATE Latin1\_General\_100\_BIN2\_UTF8,

Quantity INT,

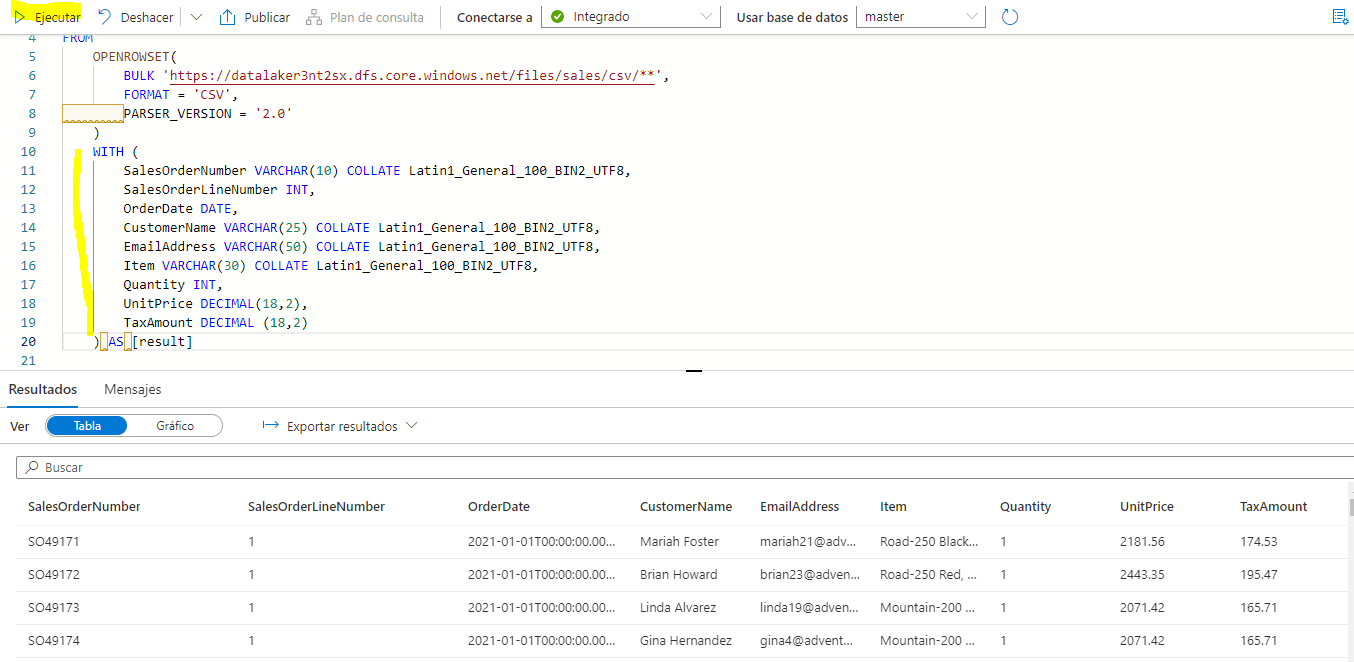
UnitPrice DECIMAL(18,2),

TaxAmount DECIMAL (18,2)

) AS [result]

Now the results look like this:

| **SalesOrderNumber** | **SalesOrderLineNumber** | **OrderDate** | **CustomerName** | **EmailAddress** | **Item** | **Quantity** | **UnitPrice** | **TaxAmount** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SO45347 | 1 | 2020-01-01 | Clarence Raji | [clarence35@adventure-works.com](mailto:clarence35@adventure-works.com) | Road-650 Black, 52 | 1 | 699.10 | 55.93 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |

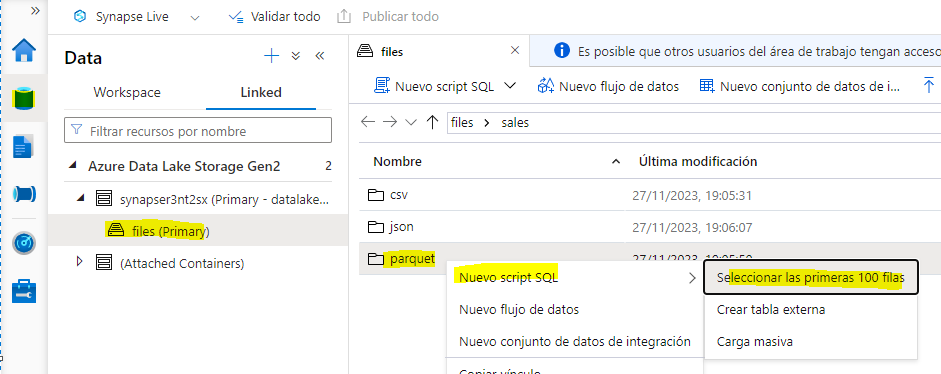


1. Publish the changes to your script, and then close the script pane.

### Use SQL to query parquet files

While CSV is an easy format to use, it's common in big data processing scenarios to use file formats that are optimized for compression, indexing, and partitioning. One of the most common of these formats is **parquet**.

1. En la pestaña **files** que contiene el sistema de archivos de su lago de datos, regrese a la carpeta **sales** para que pueda ver las carpetas **csv** , **json** y **parquet** .
2. Seleccione la carpeta **Parquet** y luego, en la lista **Nuevo script SQL de la barra de herramientas, seleccione Seleccionar las 100 filas SUPERIORES** .



1. En la lista **Tipo de archivo** , seleccione **Formato Parquet** y luego aplique la configuración para abrir un nuevo script SQL que consulte los datos de la carpeta. El script debería verse similar a este:

-- This is auto-generated codeSELECT

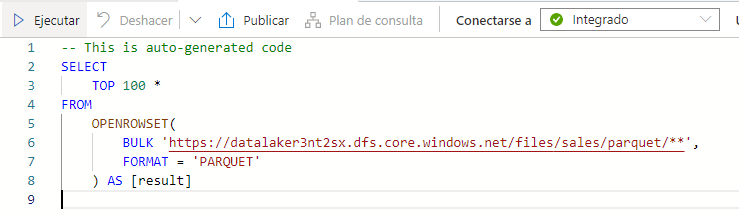
TOP 100 \*FROM

OPENROWSET(

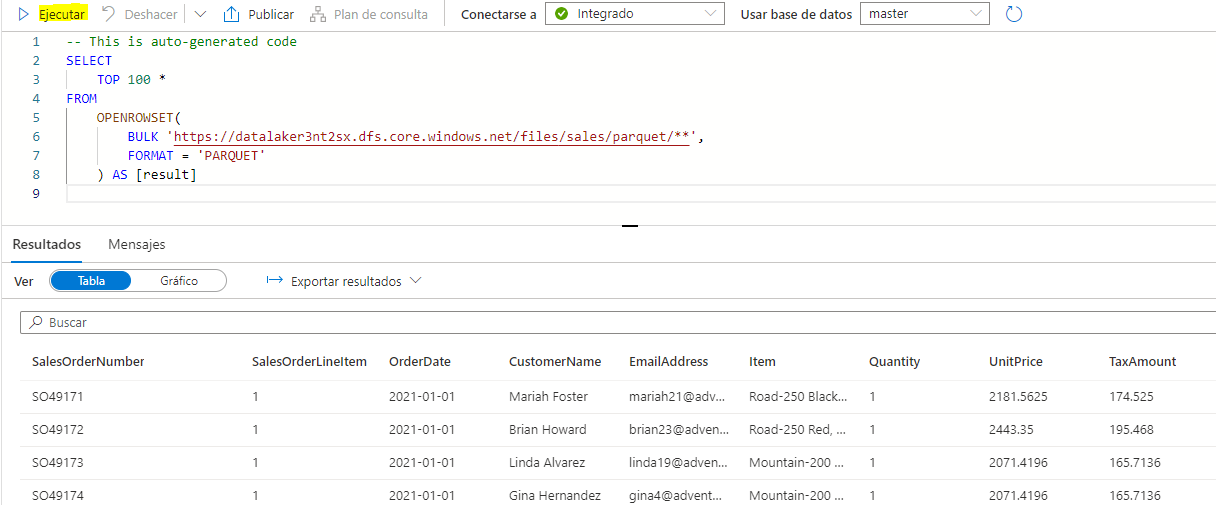
BULK 'https://datalakexxxxxxx.dfs.core.windows.net/files/sales/parquet/\*\*',

FORMAT = 'PARQUET'

) AS [result]



1. Ejecute el código y tenga en cuenta que devuelve datos de pedidos de ventas en el mismo esquema que los archivos CSV que exploró anteriormente. La información del esquema está incrustada en el archivo parquet, por lo que los nombres de las columnas apropiadas se muestran en los resultados.



1. Modifique el código de la siguiente manera (reemplazando datalakexxxxxxx con el nombre de su cuenta de almacenamiento del lago de datos) y luego ejecútelo.

SELECT YEAR(OrderDate) AS OrderYear,

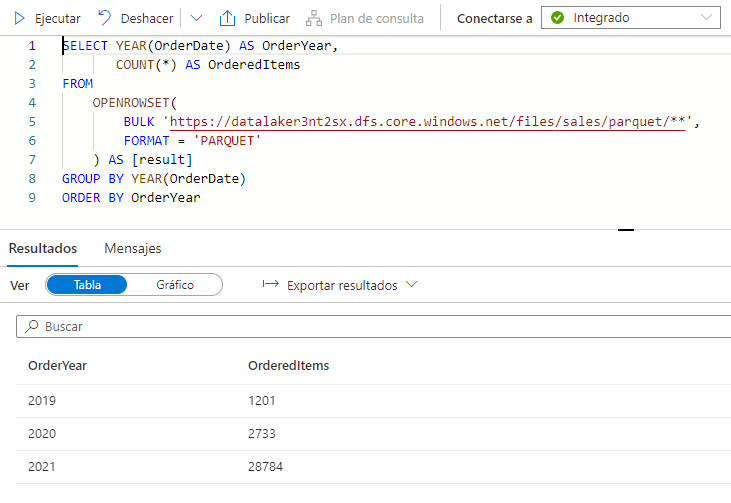
COUNT(\*) AS OrderedItemsFROM

OPENROWSET(

BULK 'https://datalakexxxxxxx.dfs.core.windows.net/files/sales/parquet/\*\*',

FORMAT = 'PARQUET'

) AS [result]GROUP BY YEAR(OrderDate)ORDER BY OrderYear



1. Tenga en cuenta que los resultados incluyen recuentos de pedidos para los tres años: el comodín utilizado en la ruta BULK hace que la consulta devuelva datos de todas las subcarpetas.

Las subcarpetas reflejan particiones en los datos de parquet, que es una técnica que se utiliza a menudo para optimizar el rendimiento de sistemas que pueden procesar múltiples particiones de datos en paralelo. También puede utilizar particiones para filtrar los datos.

1. Modifique el código de la siguiente manera (reemplazando datalakexxxxxxx con el nombre de su cuenta de almacenamiento del lago de datos) y luego ejecútelo.

SELECT YEAR(OrderDate) AS OrderYear,

COUNT(\*) AS OrderedItemsFROM

OPENROWSET(

BULK 'https://datalakexxxxxxx.dfs.core.windows.net/files/sales/parquet/year=\*/',

FORMAT = 'PARQUET'

) AS [result]WHERE [result].filepath(1) IN ('2019', '2020')GROUP BY YEAR(OrderDate)ORDER BY OrderYear

Mi código:

SELECT YEAR(OrderDate) AS OrderYear,

       COUNT(\*) AS OrderedItems

FROM

    OPENROWSET(

        BULK 'https://datalaker3nt2sx.dfs.core.windows.net/files/sales/parquet/year=\*/',

        FORMAT = 'PARQUET'

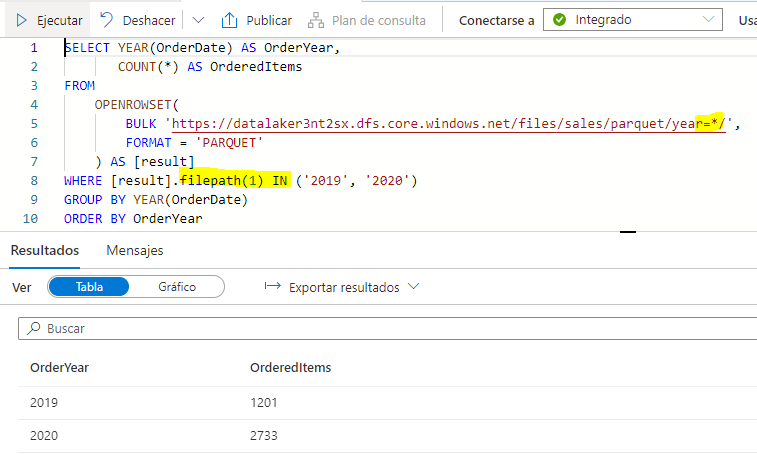
    ) AS [result]

WHERE [result].filepath(1) IN ('2019', '2020')

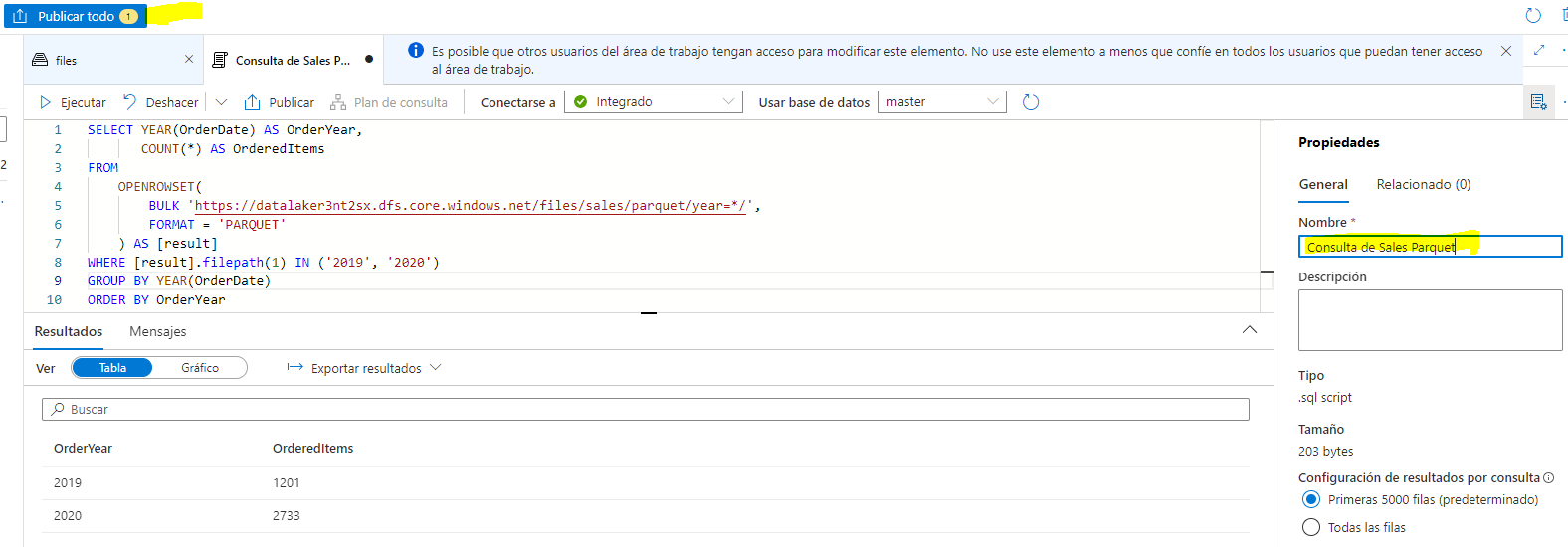
GROUP BY YEAR(OrderDate)

ORDER BY OrderYear

1. Revise los resultados y tenga en cuenta que incluyen solo los recuentos de ventas de 2019 y 2020. Este filtrado se logra incluyendo un comodín para el valor de la carpeta de partición en la ruta BULK ( año=\* ) y una cláusula WHERE basada en la propiedad de ruta de archivo del resultados devueltos por OPENROWSET (que en este caso tiene el alias [resultado] ).



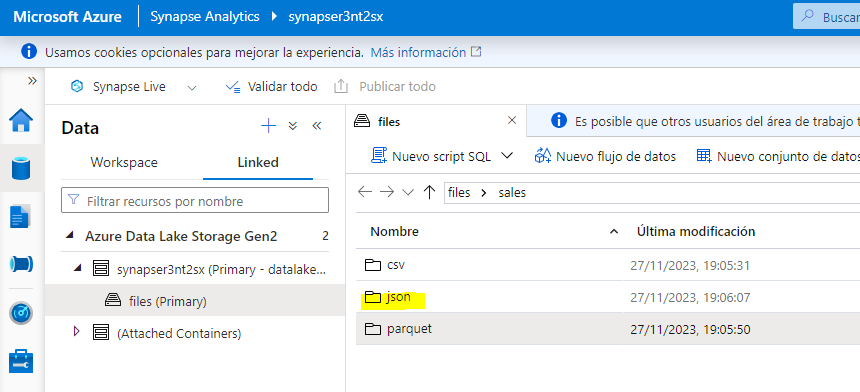
1. Asigne un nombre a su script **Consulta de Sales Parquet** y publíquelo. Luego cierre el panel de script.



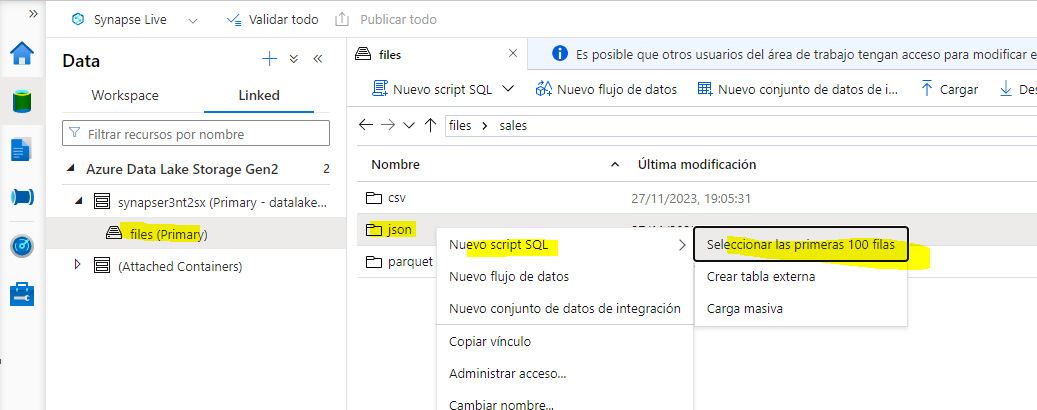
### Utilice SQL para consultar archivos JSON

JSON es otro formato de datos popular, por lo que resulta útil poder consultar archivos .json en un grupo de SQL sin servidor.

1. En la pestaña **files** que contiene el sistema de archivos de su lago de datos, regrese a la carpeta **de ventas** para poder ver las carpetas **csv** , **json** y **parquet** .



1. Seleccione la carpeta **json** y luego, en la lista **Nuevo script SQL de la barra de herramientas, seleccione Seleccionar las 100 filas SUPERIORES** .



1. En la lista **Tipo de archivo** , seleccione **Formato de texto** y luego aplique la configuración para abrir un nuevo script SQL que consulte los datos de la carpeta. El script debería verse similar a este:

-- This is auto-generated codeSELECT

TOP 100 \*FROM

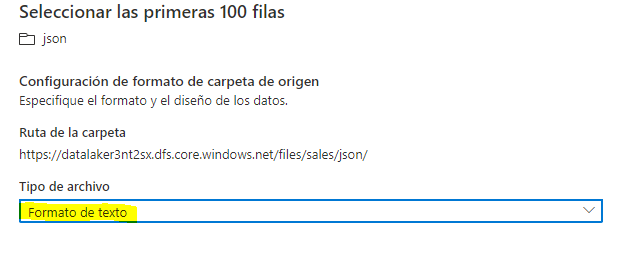
OPENROWSET(

BULK 'https://datalakexxxxxxx.dfs.core.windows.net/files/sales/json/',

FORMAT = 'CSV',

        PARSER\_VERSION = '2.0'

) AS [result]





El script está diseñado para consultar datos delimitados por comas (CSV) en lugar de JSON, por lo que debe realizar algunas modificaciones antes de que funcione correctamente.

1. Modifique el script de la siguiente manera (reemplazando datalakexxxxxxx con el nombre de su cuenta de almacenamiento del lago de datos) para:

* Elimine el parámetro de versión del analizador.
* Agregue parámetros para terminador de campo, campos entre comillas y terminadores de fila con el código de carácter 0x0b .
* Formatee los resultados como un campo único que contenga la fila de datos JSON como una cadena NVARCHAR(MAX).

SELECT

TOP 100 \*FROM

OPENROWSET(

BULK 'https://datalakexxxxxxx.dfs.core.windows.net/files/sales/json/',

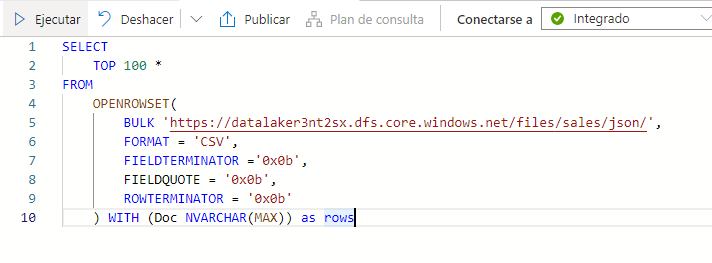
FORMAT = 'CSV',

FIELDTERMINATOR ='0x0b',

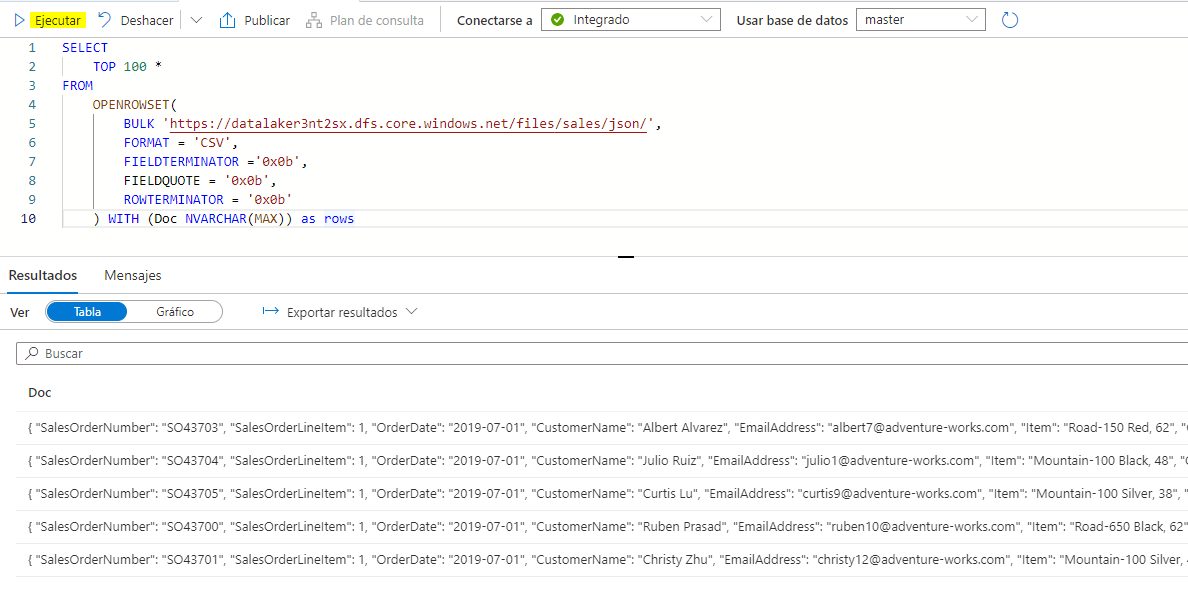
FIELDQUOTE = '0x0b',

ROWTERMINATOR = '0x0b'

) WITH (Doc NVARCHAR(MAX)) as rows



1. Ejecute el código modificado y observe que los resultados incluyen un documento JSON para cada pedido.



1. Modifique la consulta de la siguiente manera (reemplazando datalakexxxxxxx con el nombre de su cuenta de almacenamiento del lago de datos) para que use la función JSON\_VALUE para extraer valores de campos individuales de los datos JSON.

SELECT JSON\_VALUE(Doc, '$.SalesOrderNumber') AS OrderNumber,

JSON\_VALUE(Doc, '$.CustomerName') AS Customer,

DocFROM

OPENROWSET(

BULK 'https://datalakexxxxxxx.dfs.core.windows.net/files/sales/json/',

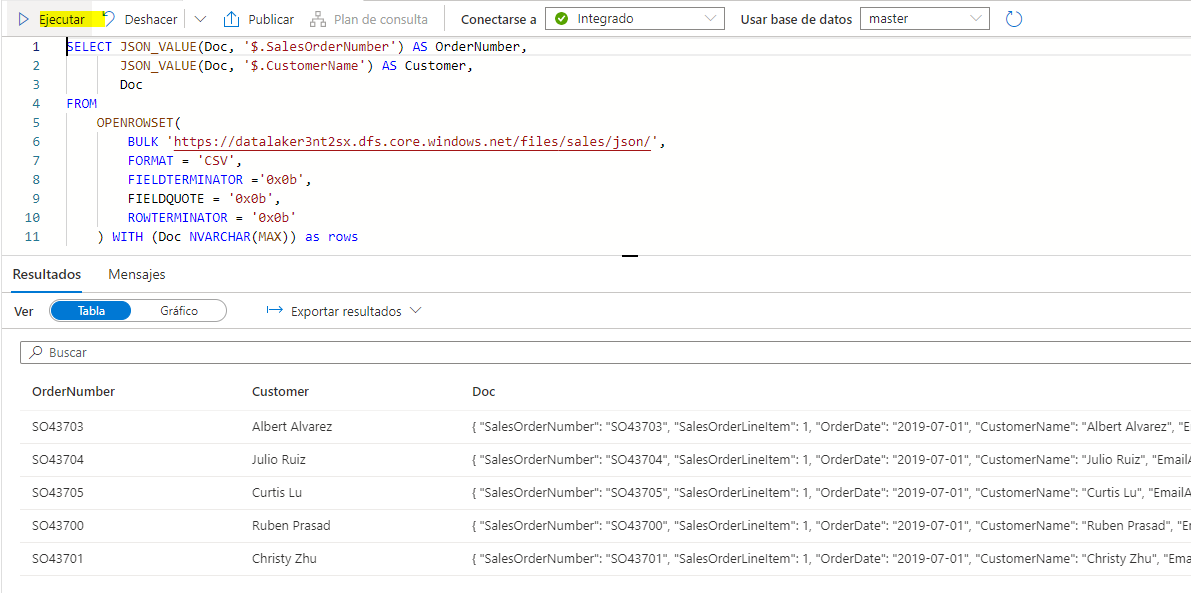
FORMAT = 'CSV',

FIELDTERMINATOR ='0x0b',

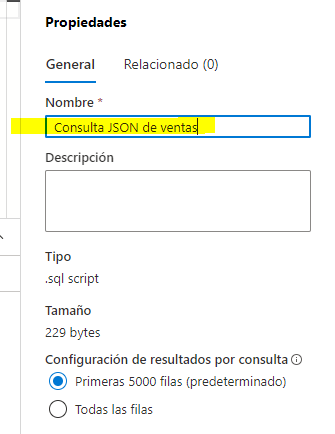
FIELDQUOTE = '0x0b',

ROWTERMINATOR = '0x0b'

) WITH (Doc NVARCHAR(MAX)) as rows



1. Asigne un nombre a su script **Consulta JSON de ventas** y publíquelo. Luego cierre el panel de script.



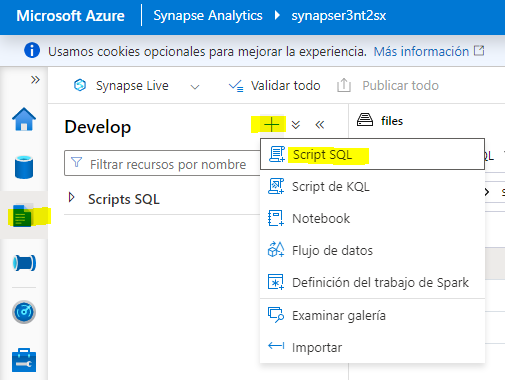
## Acceder a datos externos en una base de datos.

Hasta ahora, ha utilizado la función OPENROWSET en una consulta SELECT para recuperar datos de archivos en un lago de datos. Las consultas se ejecutaron en el contexto de la base de datos **maestra** en su grupo de SQL sin servidor. Este enfoque está bien para una exploración inicial de los datos, pero si planea crear consultas más complejas, puede ser más efectivo utilizar la capacidad **PolyBase**de **Synapse SQL** para crear objetos en una base de datos que hagan referencia a la ubicación de los datos externos.

### Crear una fuente de datos externa

Al definir una fuente de datos externa en una base de datos, puede usarla para hacer referencia a la ubicación del lago de datos donde se almacenan los archivos.

1. En Synapse Studio, en la página **Develop**, en el menú **+** , seleccione **Script SQL** .



1. En el nuevo panel de script, agregue el siguiente código (reemplazando datalakexxxxxxx con el nombre de su cuenta de almacenamiento del lago de datos) para crear una nueva base de datos y agregarle una fuente de datos externa.

CREATE DATABASE Sales

COLLATE Latin1\_General\_100\_BIN2\_UTF8;

GO;

Use Sales;

GO;

CREATE EXTERNAL DATA SOURCE sales\_data WITH (

LOCATION = 'https://datalakexxxxxxx.dfs.core.windows.net/files/sales/'

);

GO;

Mi código:

CREATE DATABASE Sales

  COLLATE Latin1\_General\_100\_BIN2\_UTF8;

GO;

Use Sales;

GO;

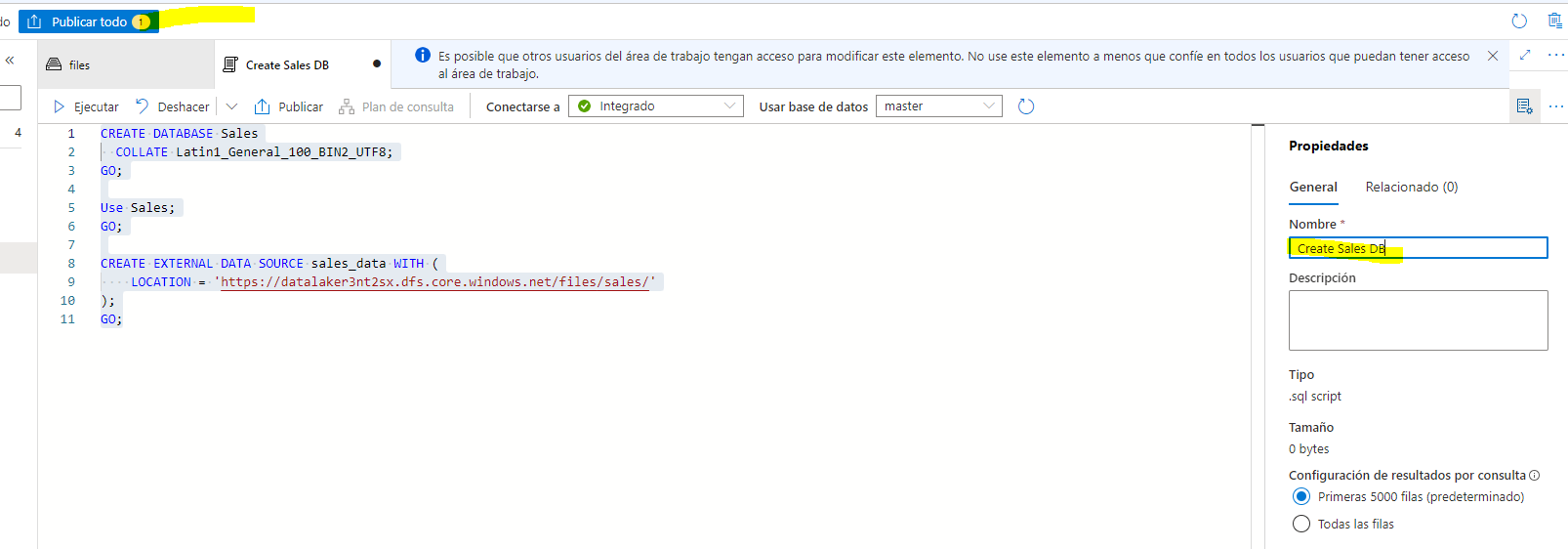
CREATE EXTERNAL DATA SOURCE sales\_data WITH (

    LOCATION = 'https://datalaker3nt2sx.dfs.core.windows.net/files/sales/'

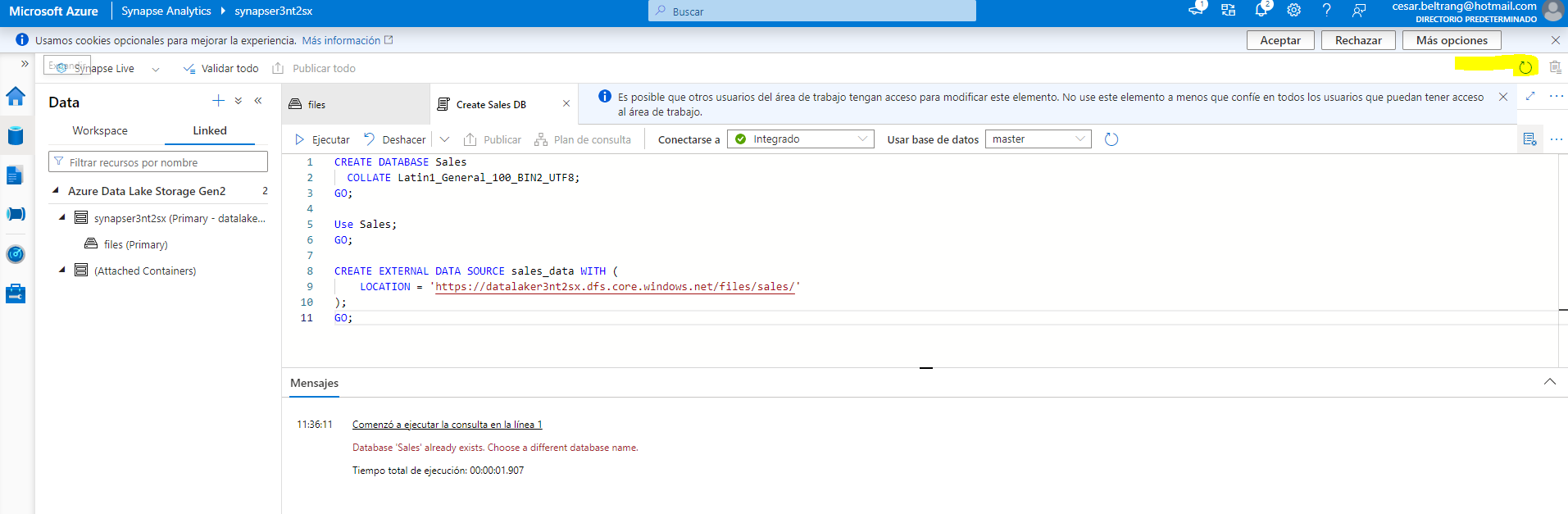
);

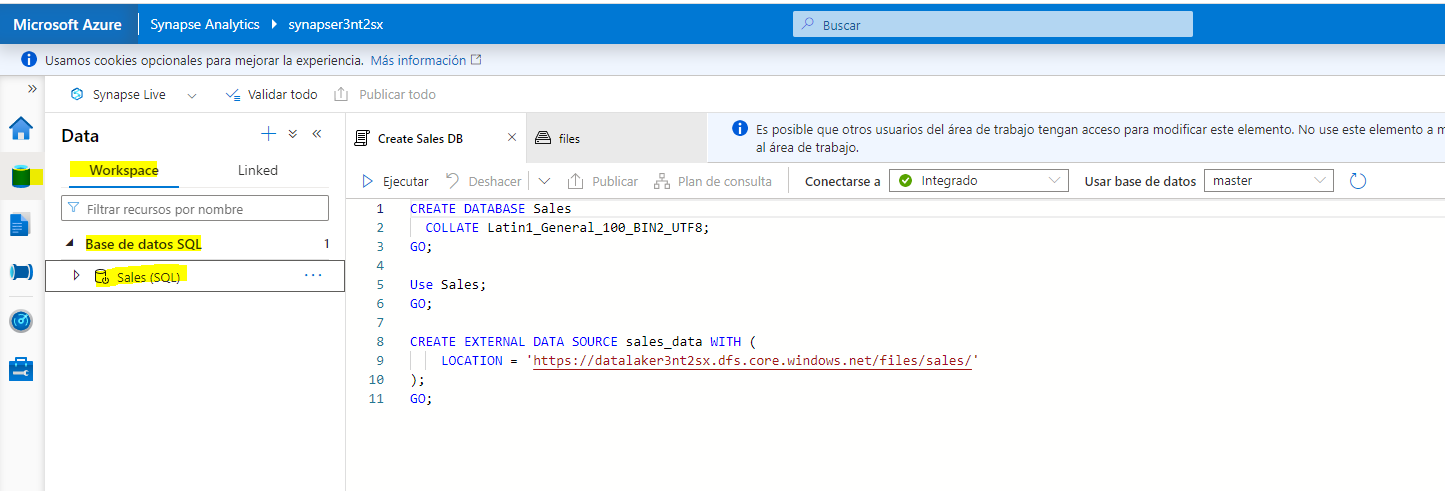
GO;

1. Modifique las propiedades del script para cambiar su nombre a **Create Sales DB** y publíquelo.

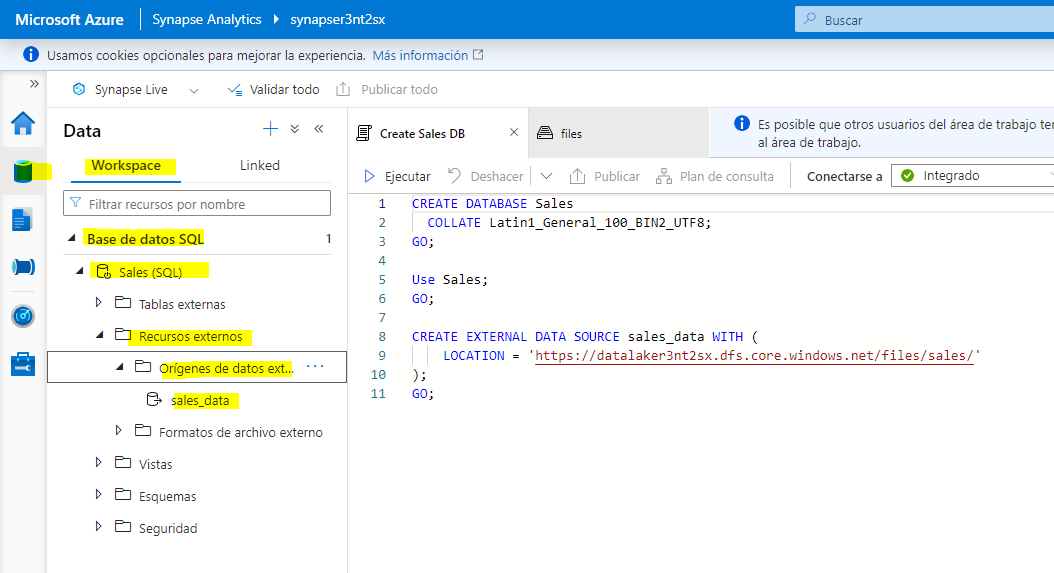


1. Asegúrese de que el script esté conectado al grupo de SQL **integrado** y a la base de datos **maestra** , y luego ejecútelo.
2. Vuelva a la página **de Data** y use el botón **↻** en la parte superior derecha de Synapse Studio para actualizar la página. Luego vea la pestaña **Workspace** en el panel **Data** , donde ahora se muestra una lista **de bases de datos SQL .**Expanda esta lista para verificar que se haya creado la base de datos **Sales.**





1. Expanda la base de datos **de Ventas** , su carpeta **de Recursos externos** y la carpeta **de Fuentes de datos externas** debajo para ver la fuente de datos externa **sales\_data** que creó.



1. En el menú **... de la base de datos Sales** , seleccione **Nuevo script SQL** > **Script vacío** . Luego, en el nuevo panel de script, ingrese y ejecute la siguiente consulta:

SELECT \*FROM

OPENROWSET(

BULK 'csv/\*.csv',

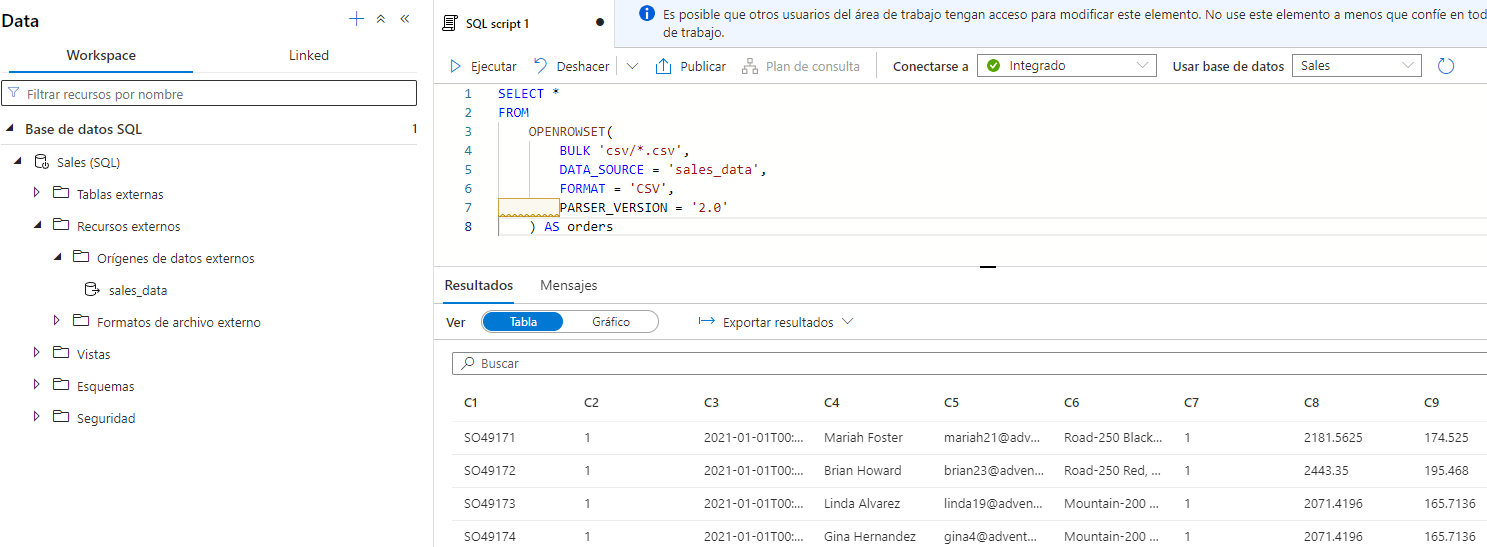
DATA\_SOURCE = 'sales\_data',

FORMAT = 'CSV',

        PARSER\_VERSION = '2.0'

) AS orders

La consulta utiliza la fuente de datos externa para conectarse al lago de datos y la función OPENROWSET ahora solo necesita hacer referencia a la ruta relativa a los archivos .csv.



1. Modifique el código de la siguiente manera para consultar los archivos de parquet utilizando la fuente de datos.

SELECT \*FROM

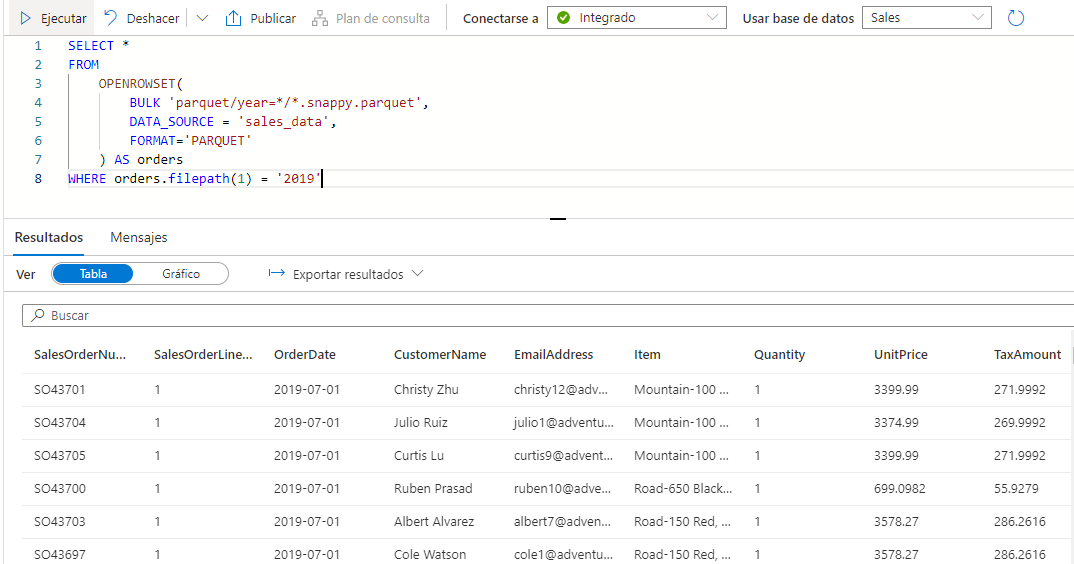
OPENROWSET(

BULK 'parquet/year=\*/\*.snappy.parquet',

DATA\_SOURCE = 'sales\_data',

FORMAT='PARQUET'

) AS ordersWHERE orders.filepath(1) = '2019'



### Crear una tabla externa

La fuente de datos externa facilita el acceso a los archivos en el lago de datos, pero la mayoría de los analistas de datos que utilizan SQL están acostumbrados a trabajar con tablas en una base de datos. Afortunadamente, también puede definir formatos de archivos externos y tablas externas que encapsulan conjuntos de filas de archivos en tablas de bases de datos.

1. Reemplace el código SQL con la siguiente declaración para definir un formato de datos externo para archivos CSV y una tabla externa que haga referencia a los archivos CSV, y ejecútelo:

CREATE EXTERNAL FILE FORMAT CsvFormat

    WITH (

        FORMAT\_TYPE = DELIMITEDTEXT,

        FORMAT\_OPTIONS(

        FIELD\_TERMINATOR = ',',

        STRING\_DELIMITER = '"'

        )

    );

GO;

CREATE EXTERNAL TABLE dbo.orders

(

    SalesOrderNumber VARCHAR(10),

    SalesOrderLineNumber INT,

    OrderDate DATE,

    CustomerName VARCHAR(25),

    EmailAddress VARCHAR(50),

    Item VARCHAR(30),

    Quantity INT,

    UnitPrice DECIMAL(18,2),

    TaxAmount DECIMAL (18,2)

)

WITH

(

    DATA\_SOURCE =sales\_data,

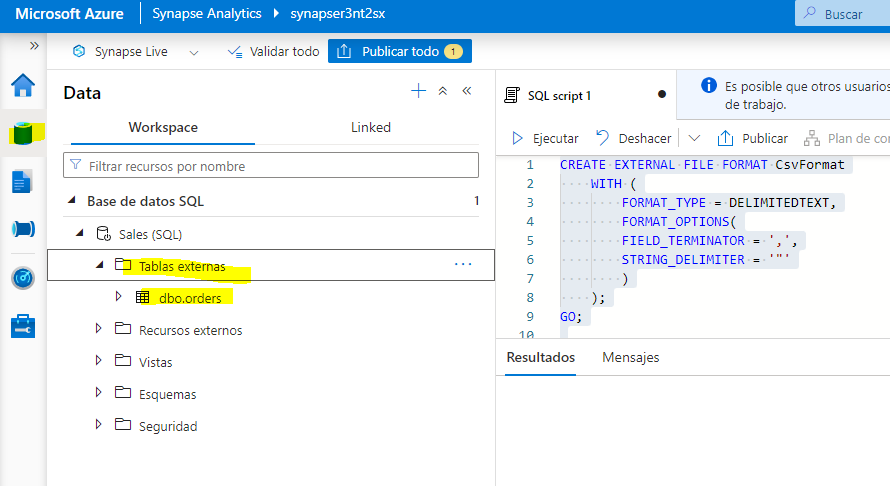
    LOCATION = 'csv/\*.csv',

    FILE\_FORMAT = CsvFormat

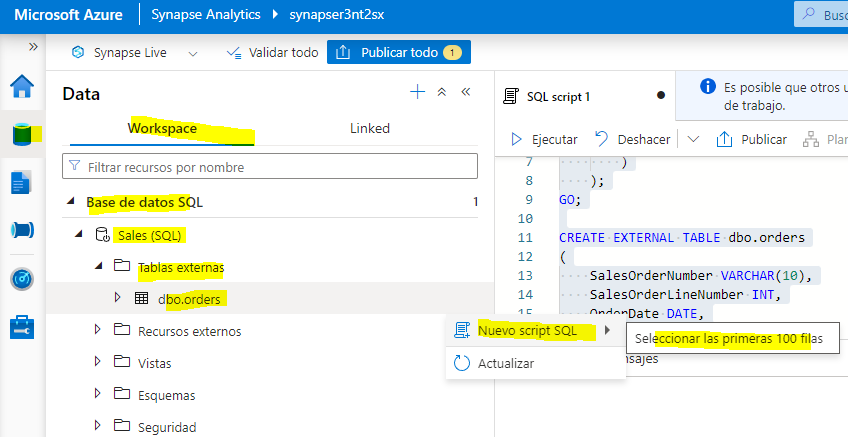
);

GO

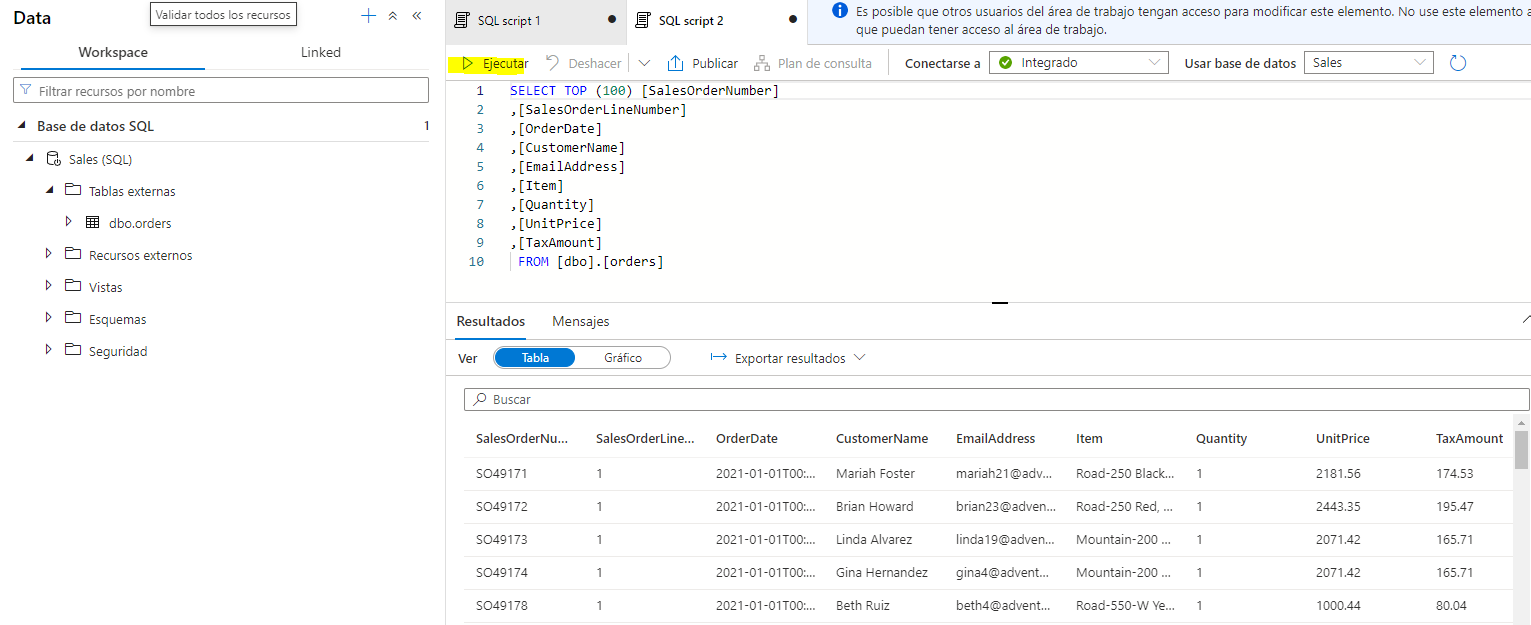
1. Actualice y expanda la carpeta **Tablas externas** en el panel **Data** y confirme que se haya creado una tabla denominada **dbo.orders en la base de datos **Sales****.



1. En el menú **... de la tabla dbo.orders** , seleccione **Nuevo script SQL** > **Seleccionar 100 filas SUPERIORES** .



1. Ejecute el script SELECT que se ha generado y verifique que recupere las primeras 100 filas de datos de la tabla, que a su vez hace referencia a los archivos en el lago de datos.

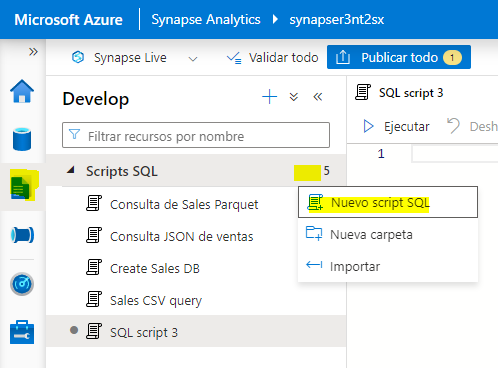


**Nota:** Siempre debe elegir el método que mejor se adapte a sus necesidades y caso de uso específicos. Para obtener información más detallada, puede consultar los artículos [Cómo usar OPENROWSET usando un grupo de SQL sin servidor en Azure Synapse Analytics](https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/develop-openrowset) y [Acceder al almacenamiento externo usando un grupo de SQL sin servidor en Azure Synapse Analytics .](https://learn.microsoft.com/en-us/azure/synapse-analytics/sql/develop-storage-files-overview?tabs=impersonation)

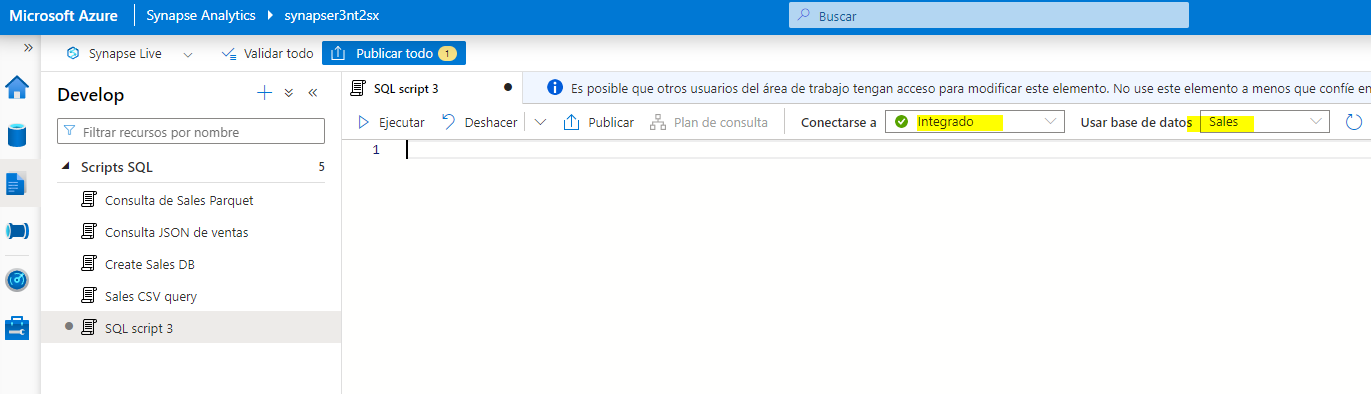
## Visualizar los resultados de la consulta

Ahora que ha explorado varias formas de consultar archivos en el lago de datos mediante consultas SQL, puede analizar los resultados de estas consultas para obtener información sobre los datos. A menudo, es más fácil descubrir conocimientos al visualizar los resultados de la consulta en un gráfico; lo cual puede hacer fácilmente utilizando la función de gráficos integrada en el editor de consultas de Synapse Studio.

1. En la página **Develop**, cree una nueva consulta SQL vacía.



1. Asegúrese de que el script esté conectado al grupo de SQL **integrado** y a la base de datos **de ventas** .



1. Ingrese y ejecute el siguiente código SQL:

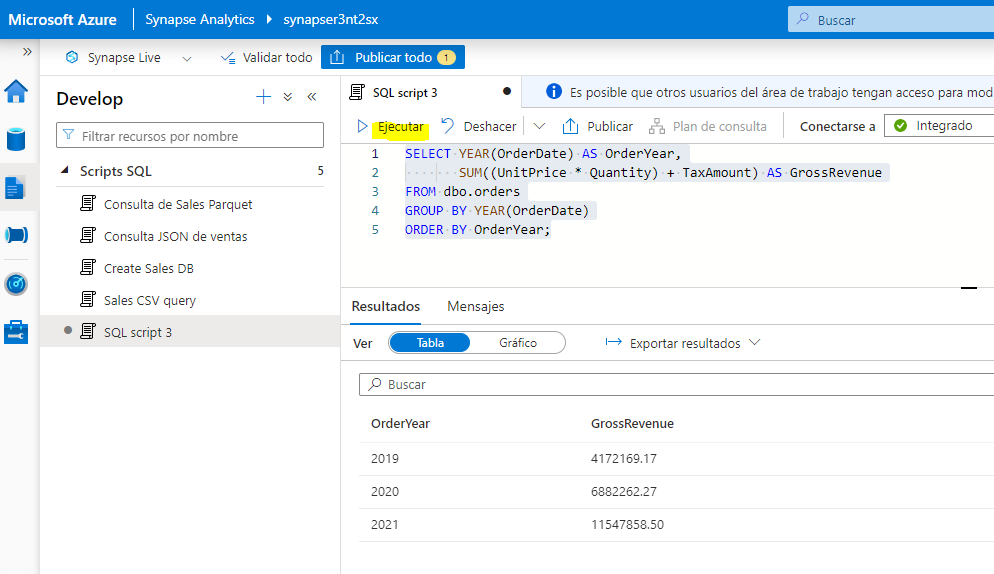
SELECT YEAR(OrderDate) AS OrderYear,

       SUM((UnitPrice \* Quantity) + TaxAmount) AS GrossRevenue

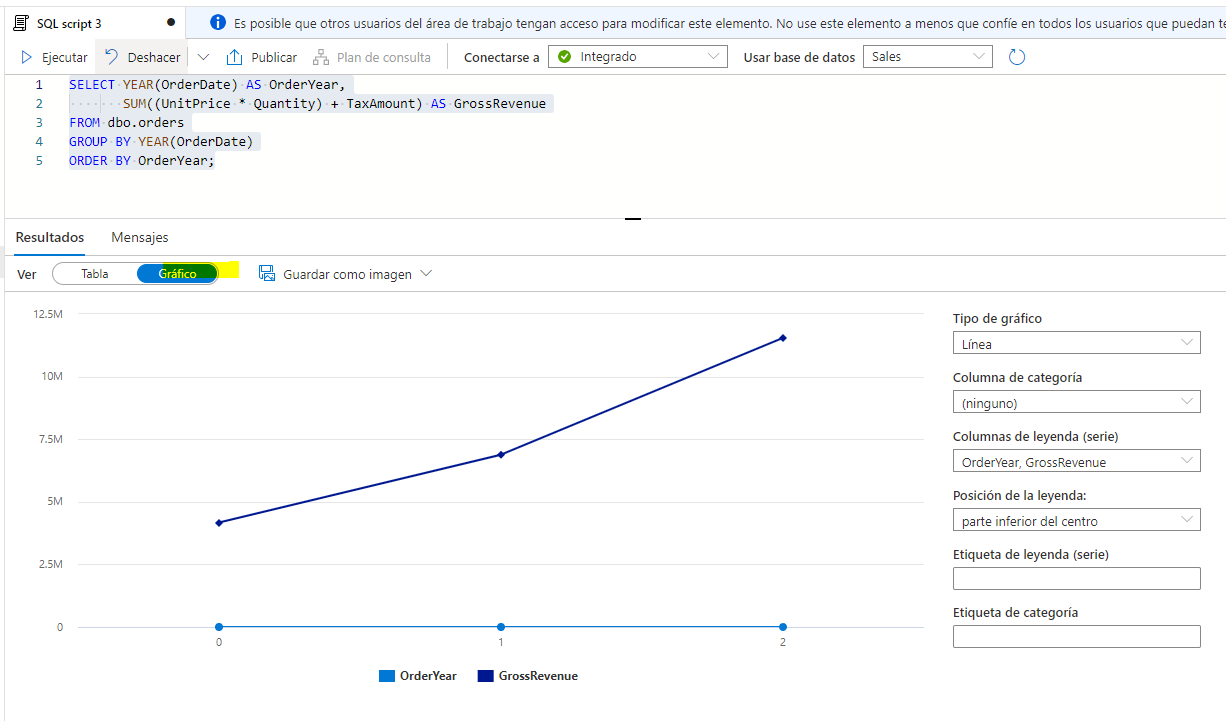
FROM dbo.orders

GROUP BY YEAR(OrderDate)

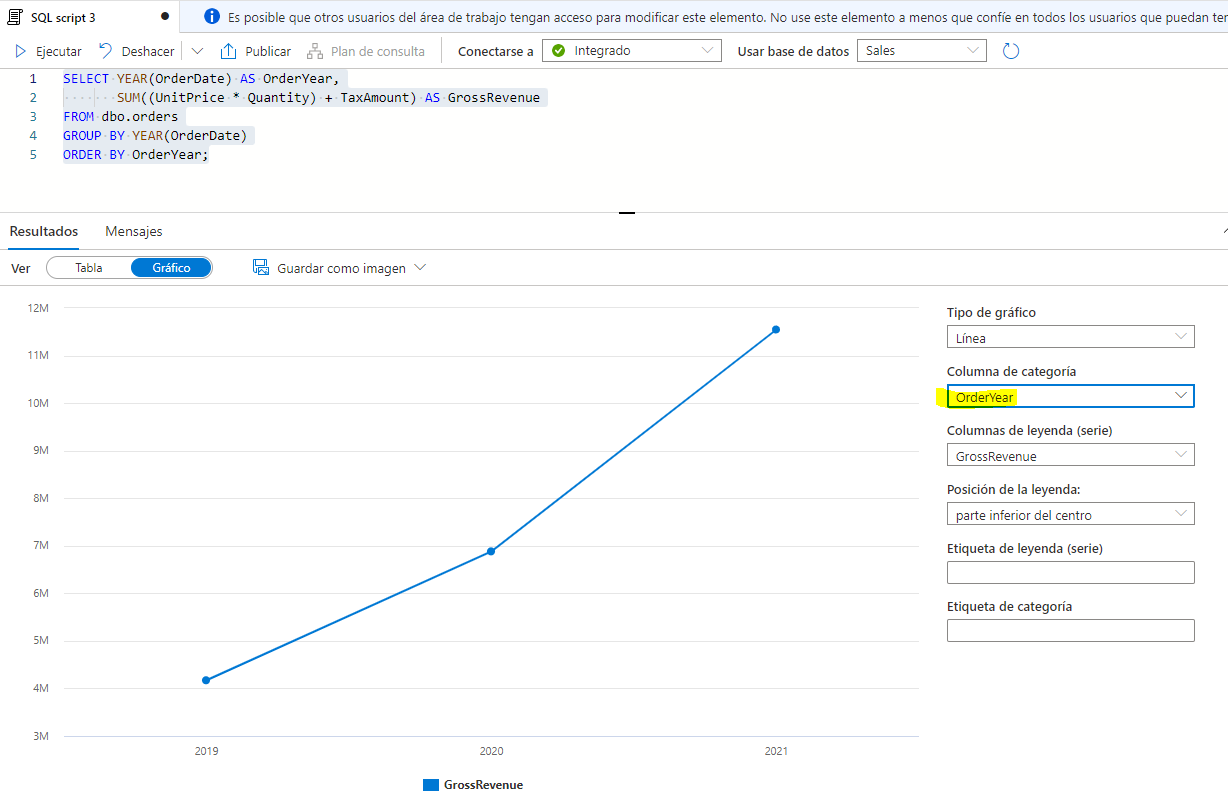
ORDER BY OrderYear;



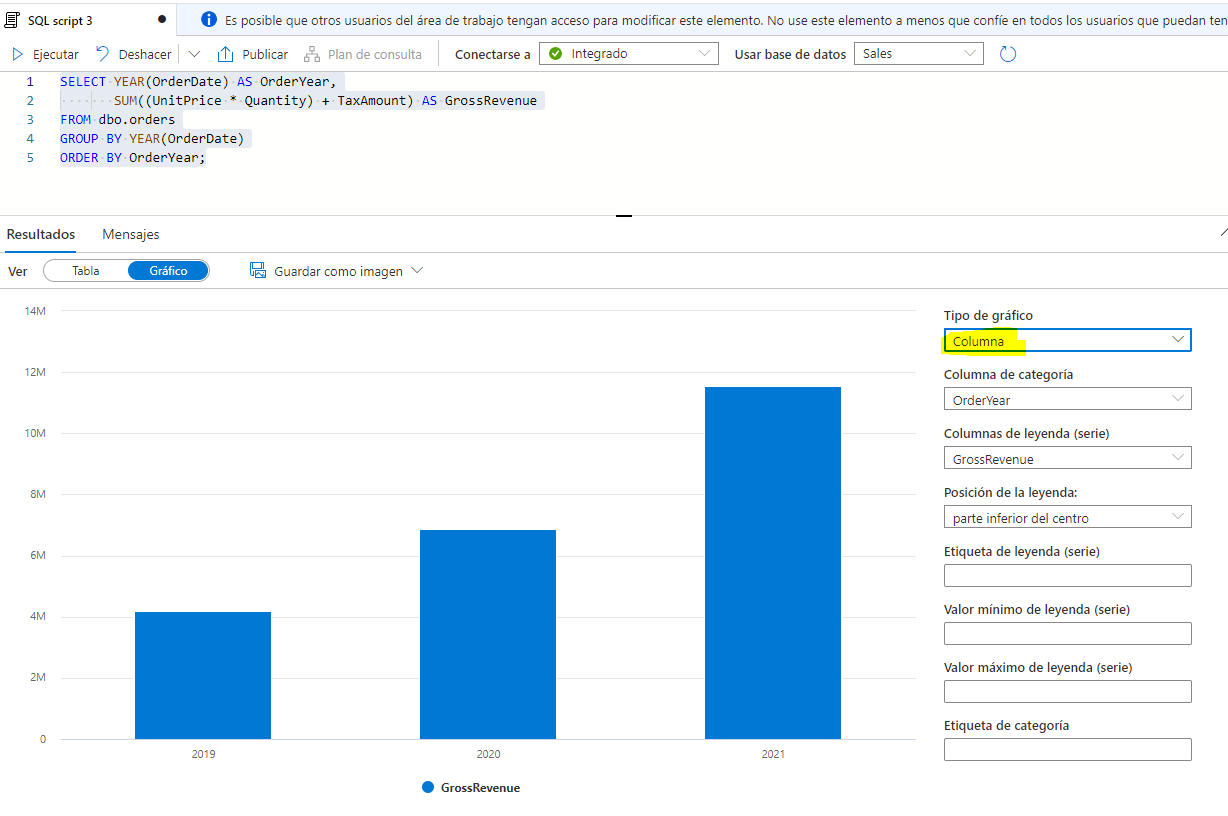
1. En el panel **Resultados** , seleccione **Gráfico** y vea el gráfico creado para usted; que debería ser un gráfico de líneas.



1. Cambie la **columna de Categoría** a **OrderYear**para que el gráfico de líneas muestre la tendencia de los ingresos durante el período de tres años de 2019 a 2021:



1. Cambie el **tipo de gráfico** a **Columna** para ver los ingresos anuales como un gráfico de columnas:

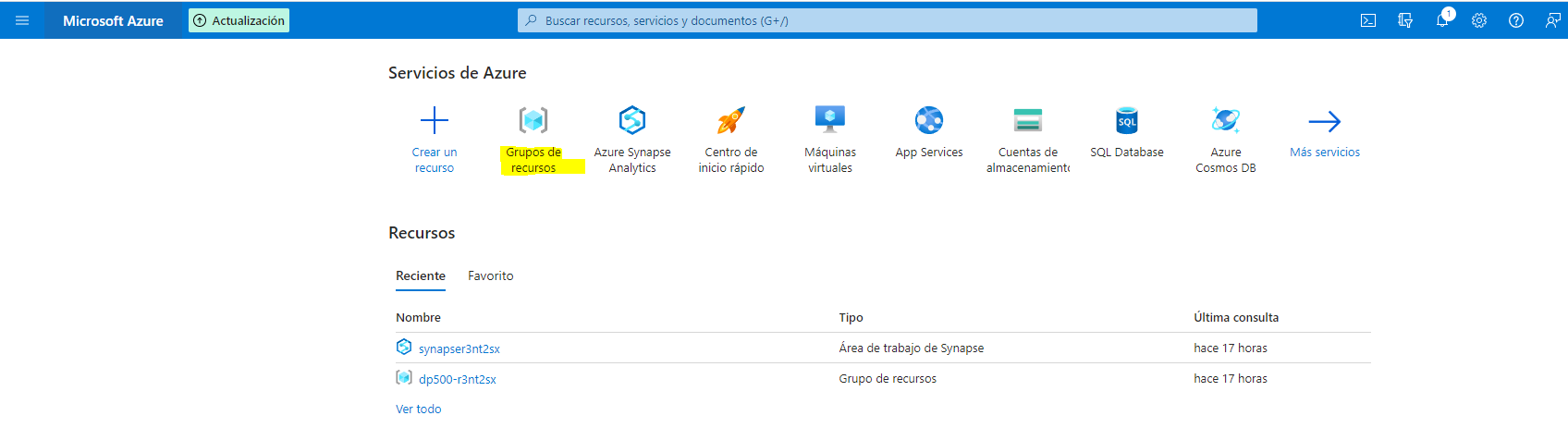


1. Experimente con la funcionalidad de gráficos en el editor de consultas. Ofrece algunas capacidades de gráficos básicas que puede utilizar mientras explora datos de forma interactiva, y puede guardar gráficos como imágenes para incluirlos en informes. Sin embargo, la funcionalidad es limitada en comparación con las herramientas de visualización de datos empresariales como Microsoft Power BI.

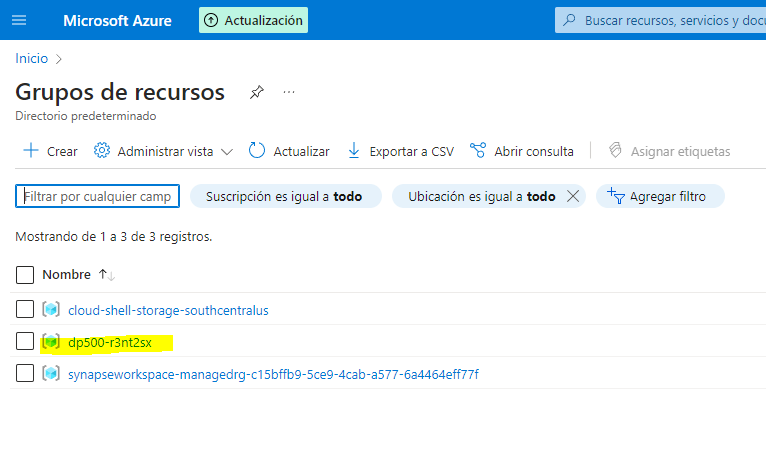
## Eliminar recursos de Azure

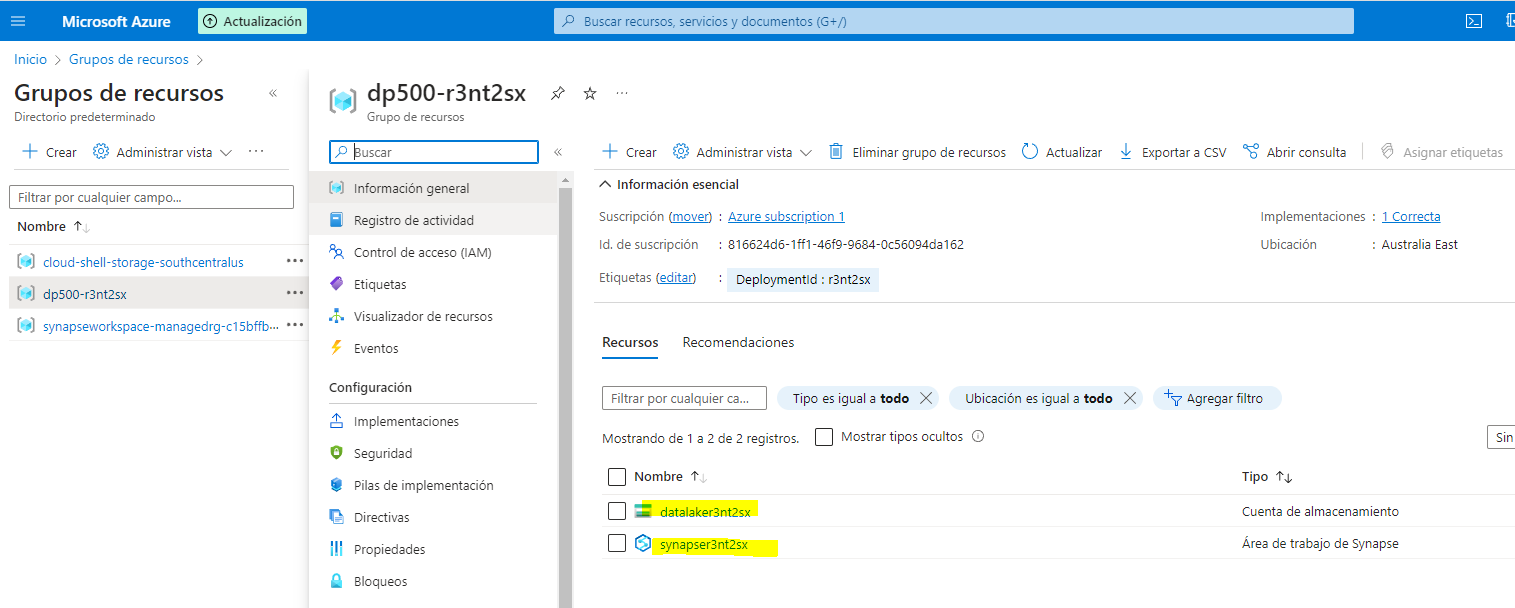
Si ha terminado de explorar Azure Synapse Analytics, debe eliminar los recursos que creó para evitar costos innecesarios de Azure.

1. Cierre la pestaña del navegador Synapse Studio y regrese a Azure Portal.
2. En Azure Portal, en la página **de inicio** , seleccione **Grupos de recursos** .

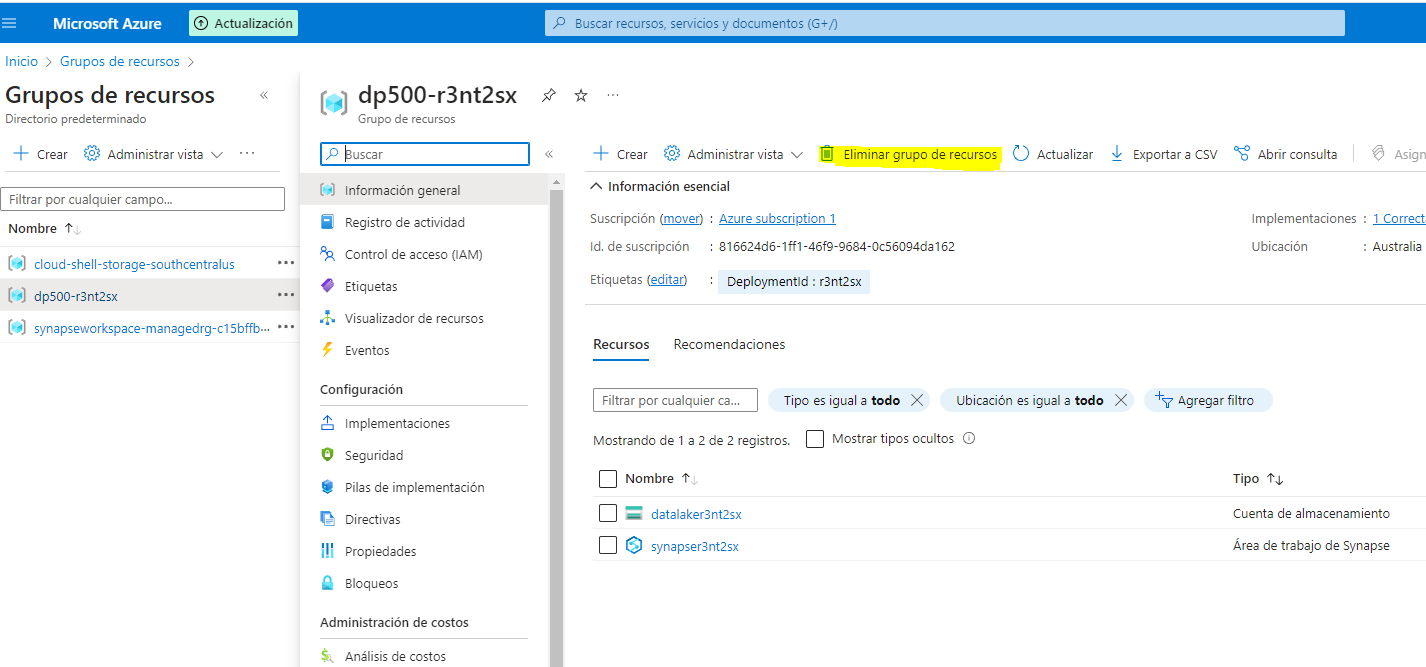


1. Seleccione el grupo de recursos **dp500-**xxxxxxx para su área de trabajo de Synapse Analytics (no el grupo de recursos administrado) y verifique que contenga el área de trabajo de Synapse y la cuenta de almacenamiento para su área de trabajo.

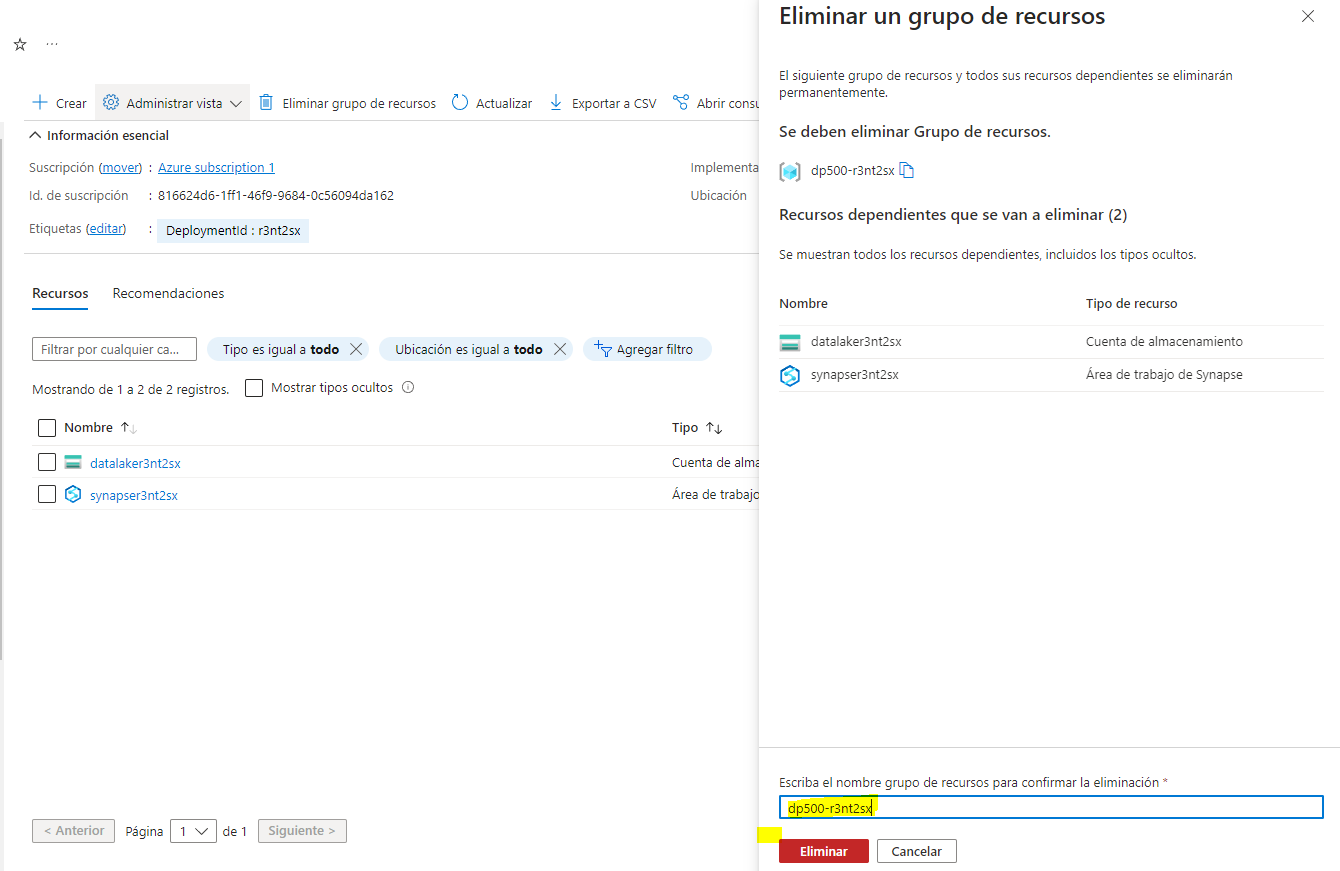


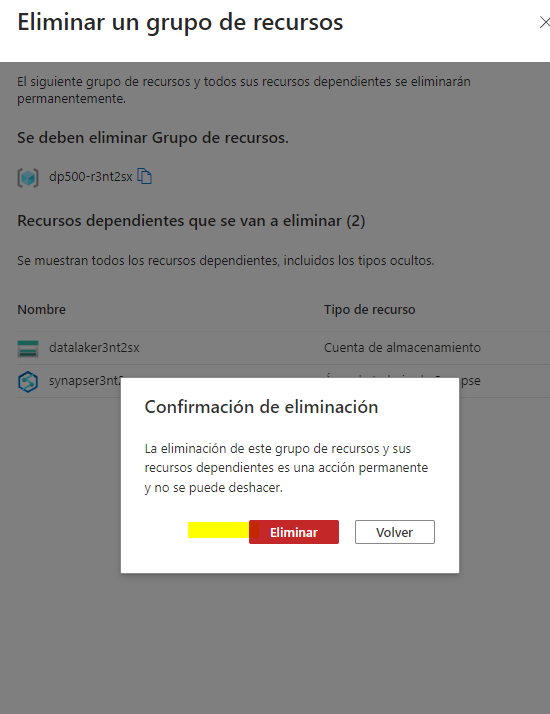


1. En la parte superior de la página **Descripción general** de su grupo de recursos, seleccione **Eliminar grupo de recursos** .



1. Ingrese el nombre del grupo de recursos **dp500-**xxxxxxx para confirmar que desea eliminarlo y seleccione **Eliminar** .





Después de unos minutos, se eliminarán el grupo de recursos del área de trabajo de Azure Synapse y el grupo de recursos del área de trabajo administrada asociado a él.

