

# Exercise - Identify modern data warehouse architecture components

10 minutes

There is more to a data warehouse than simply storing business data. Data grows at an exponential rate, year over year. Not just the volume of data, but the variety of data, from structured, to semi-structured, and to a greater degree, unstructured that must be managed. The velocity and variety of data leads to data engineering challenges when it comes to ingesting, transforming, and preparing the data for machine learning, reporting, and other purposes.

The modern data warehouse serves to address these challenges. A good data warehouse adds value, such as acting as a central location for all your data, scale with the data as it grows over time, and providing familiar tools and ecosystem for your data engineers, data analysts, data scientists, and developers.

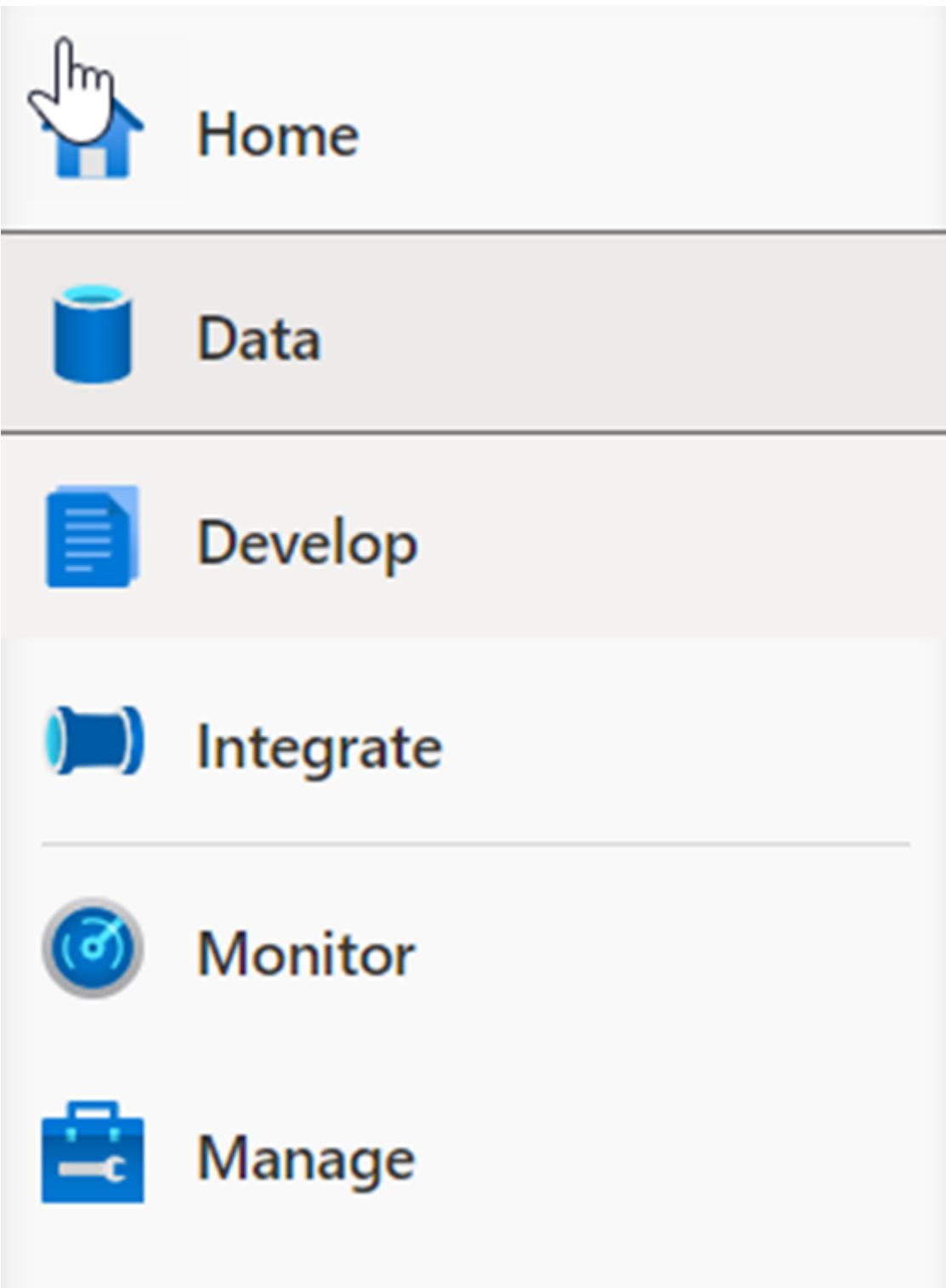
Let's look at each of these elements in detail.

## One place for all your data

With a modern data warehouse, we have one hub for all data when using Synapse Analytics.

Synapse Analytics enables you to ingest data from multiple data sources through its orchestration pipelines.

1. Select the **Integrate** hub.



Manage integration pipelines within the Integrate hub. If you are familiar with Azure Data Factory (ADF), then you will feel at home in this hub. The pipeline creation experience is the same as in ADF, which gives you another powerful integration built into Azure Synapse Analytics, removing the need to use Azure Data Factory for data movement and transformation pipelines.

2. Expand Pipelines and select **Customize EMail Analytics** (1). Select the **Copy data** activity on the canvas (2), select the **Source** tab (3), then select **Preview data** (4).

The screenshot shows the Azure Data Factory pipeline editor. On the left, under 'Integrate' > 'Pipelines', the 'Customize EMail Analytics' pipeline is selected (marked with a red box 1). In the center, the 'Activities' pane shows the 'Copy data' activity (marked with a red box 2). Below it, the 'Source' tab is selected (marked with a red box 3) and the 'Preview data' button is highlighted (marked with a red box 4).

Here we see the source CSV data that the pipeline ingests.

The screenshot shows the 'Preview data' window for the 'EmailAnalytics.csv' object. It displays a sample of the data with the following columns: Recency, History\_Segment\_ID, History\_Segment, History, Men, Women, Zip\_Code, Newbie, Channel, and Segment. The data rows are as follows:

Recency	History_Segment_ID	History_Segment	History	Men	Women	Zip_Code	Newbie	Channel	Segment
8	2	\$100 - \$200	78.24	1	0	Suburban	0	Web	Web Only
2	3	\$200 - \$350	39.78	0	1	Suburban	0	Web	Mens Mail
11	1	\$0 - \$100	199.95	1	0	Suburban	1	Phone	Web Only
3	2	\$100 - \$200	514.52	0	1	Suburban	1	Web	Mens Mail
1	2	\$100 - \$200	229.53	1	0	Suburban	0	Web	Mens Mail
4	2	\$200 - \$350	107.64	0	1	Urban	1	Web	Women

3. Close the preview, then select Open next to the **CustomEmailAnalytics** source dataset.

General   Source   Sink   Mapping   Settings   User properties

Source dataset \* CustomEmailAnalytics

File path type  File path in dataset  Wildcard file path  List of files

4. Show the **Linked service** associated with the dataset's connection, as well as the CSV file path (1). Close (2) the dataset to return to the pipeline.

Custom Email Analytics X 2

DelimitedText  
CustomEmailAnalytics

CSV

Connection Schema Parameters

Linked service \* asaexpdalakeinaday42    1

File path \* customcsv / Directory / EmailAnalytics.csv

Compression type none

Column delimiter Comma (,)

Row delimiter Auto detect (\r\n, or \r\r\n)

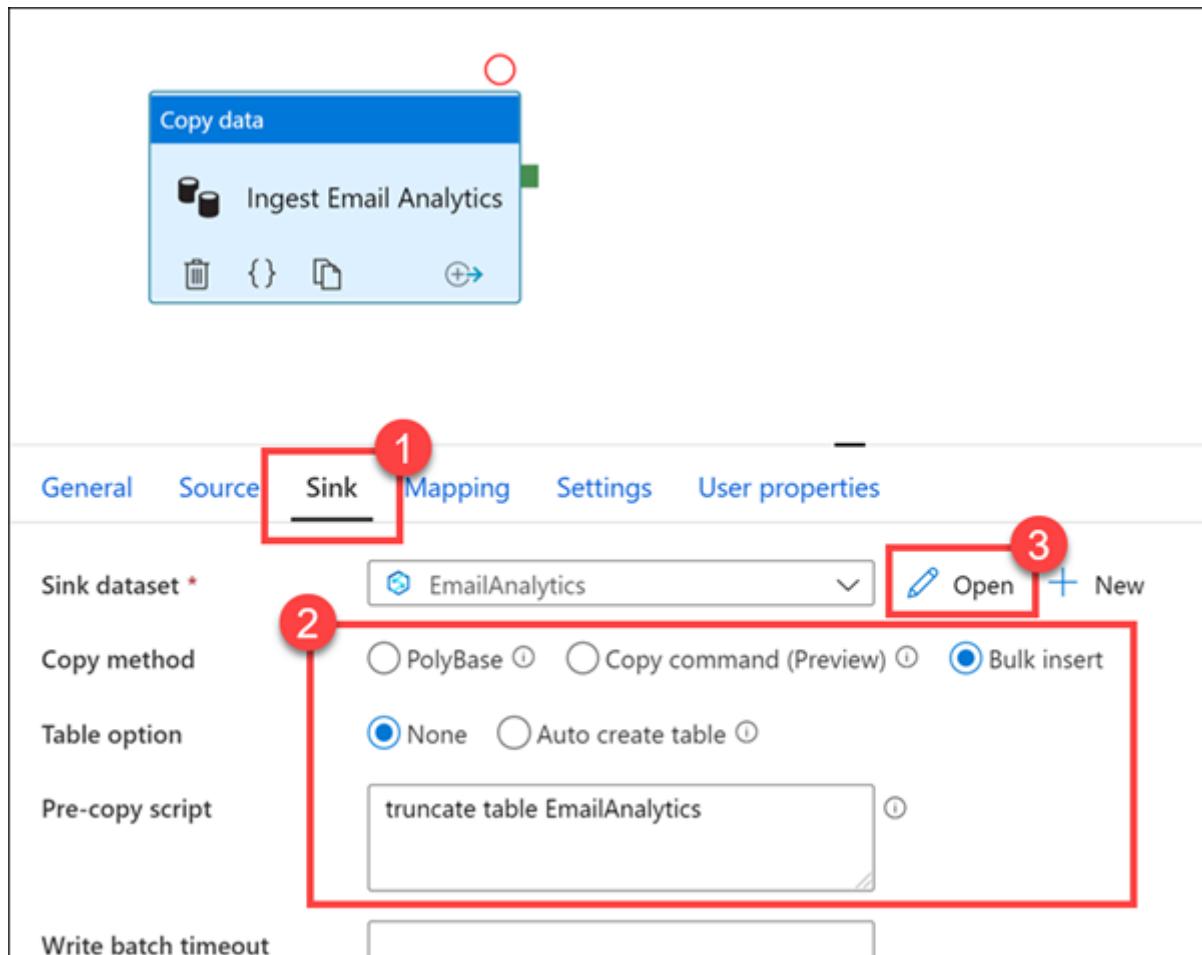
Encoding Default(UTF-8)

Escape character Backslash (\)

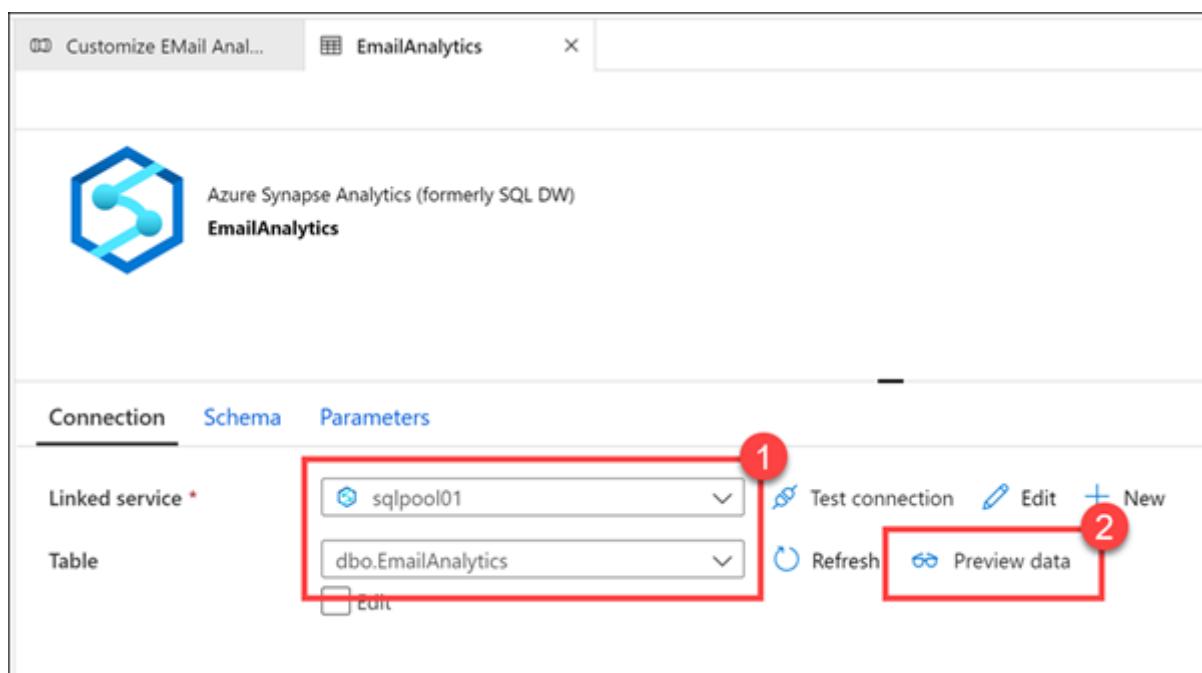
Quote character Double quote (")

First row as header

5. On the pipeline, select the **Sink** tab (1). The bulk insert copy method is selected and there is a pre-copy script that truncates the **EmailAnalytics** table, which runs prior to copying the data from the CSV source (2). Select **Open** next to the **EmailAnalytics** sink dataset (3).



6. The **Linked service** is the Azure Synapse Analytics SQL pool, and the **Table** is **EmailAnalytics** (1). The Copy data activity in the pipeline uses the connection details in this dataset to copy data from the CSV data source into the SQL pool. Select **Preview data** (2).



We can see that the table already contains data, which means that we have successfully run the pipeline in the past.

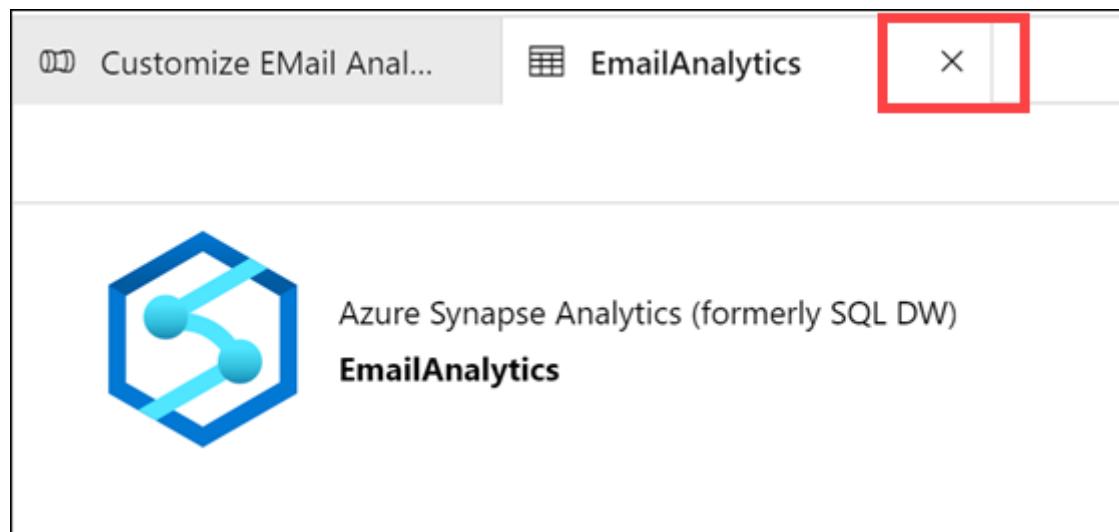
Preview data

Linked service: sqlpool01

Object: dbo.EmailAnalytics

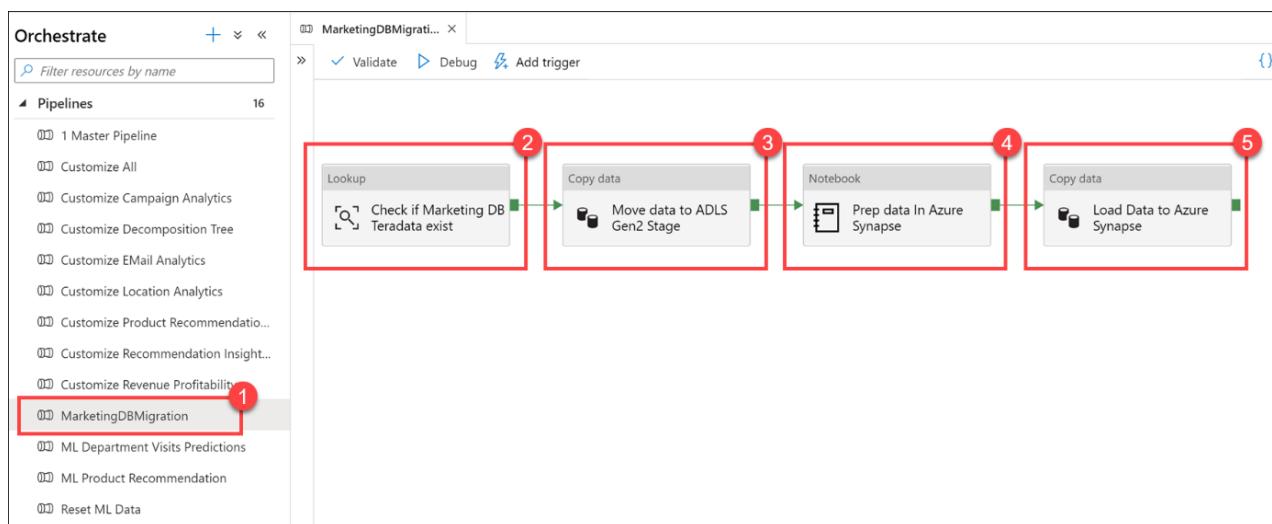
Recency	History_Segment_ID	History_Segment	History	Men	Women	Zip_Code	Newbie	Channel	S
2	3	\$200 - \$350	39.78	0	1	Suburban	0	Web	M M
3	2	\$100 - \$200	78.69	0	1	Suburban	0	Phone	M M
11	3	\$200 - \$350	318.52	0	1	Urban	0	Web	W E
9	2	\$100 - \$200	98.56	1	0	Suburban	1	Web	W E
9	3	\$200 - \$350	103.07	1	0	Suburban	0	Web	M M

## 7. Close the EmailAnalytics dataset.



8. Select the **Mapping** tab. This is where you configure the mapping between the source and sink datasets. The **Import schemas** button attempts to infer the schema for your datasets if they are based on unstructured or semi-structured data sources, like CSV or JSON files. It also reads the schema from structured data sources, like Synapse Analytics SQL pools. You also have the option to manually create your schema mapping by clicking on **+ New mapping** or by modifying the data types.

9. Select the **MarketingDBMigration** (1) pipeline. Direct your attention to the pipeline's canvas (2).

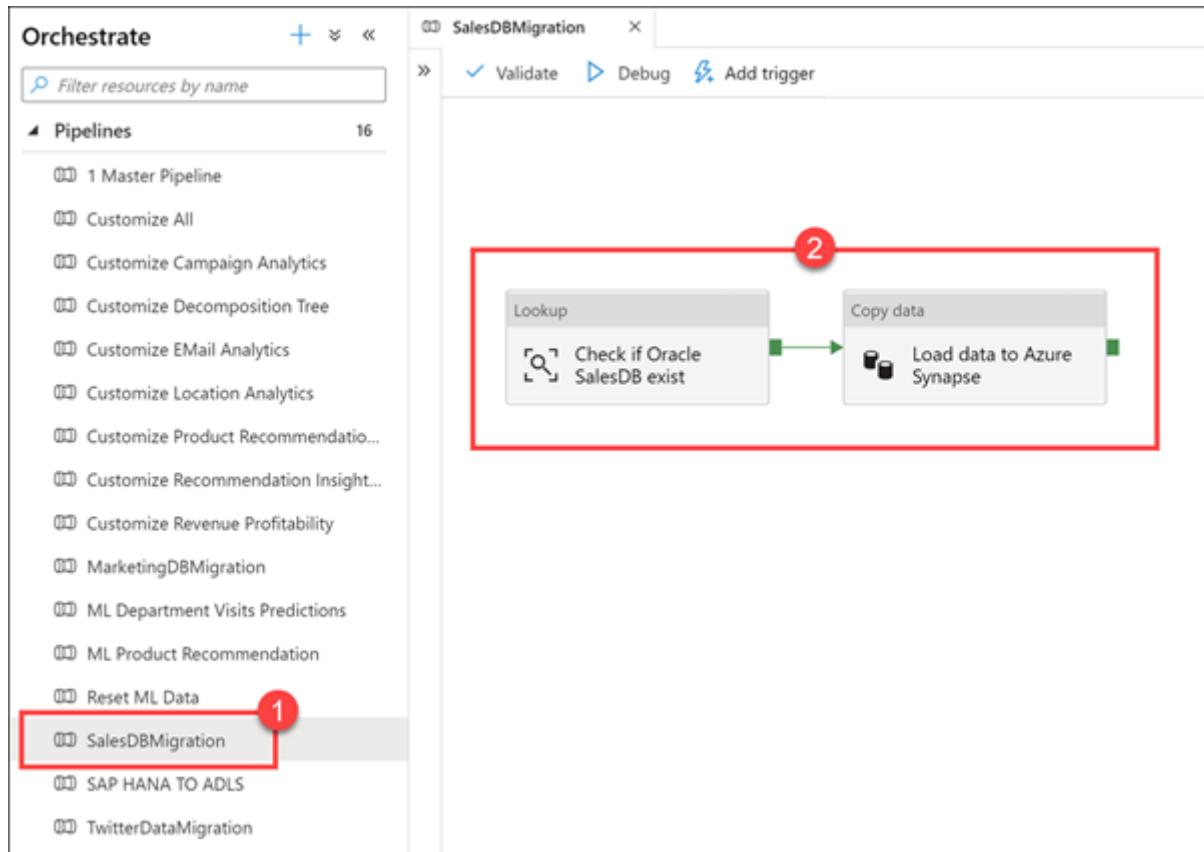


This pipeline is responsible for copying data from a Teradata database. The first activity is a **lookup** (2) to make sure that the source data exists. If data exists, it flows to the **copy data activity** (3) to move the source data into the data lake (ADLS Gen2 primary data source). The next step is a **Notebook activity** (4), which uses Apache Spark within a Synapse Notebook to perform data engineering tasks. The last step is another **copy data activity** (5) that loads the prepared data and stores it into an Azure Synapse SQL pool table.

This workflow is common when conducting data movement orchestration. Synapse Analytics pipelines makes it easy to define data movement and transformation steps, and

encapsulates these steps into a repeatable process that you can maintain and monitor within your modern data warehouse.

10. Select the **SalesDBMigration** (1) pipeline. Direct your attention to the pipeline's canvas (2).



Here is another example of a data movement orchestration pipeline that helps us combine external data sources into our warehouse. In this case, we load data from an Oracle sales database into an Azure Synapse SQL pool table.

11. Select the **SAP HANA TO ADLS** pipeline. This pipeline copies data from a financial SAP HANA data source into the SQL pool.
12. Select the + button at the top of the **Orchestrate** blade, then select **Pipeline** to create a new pipeline.

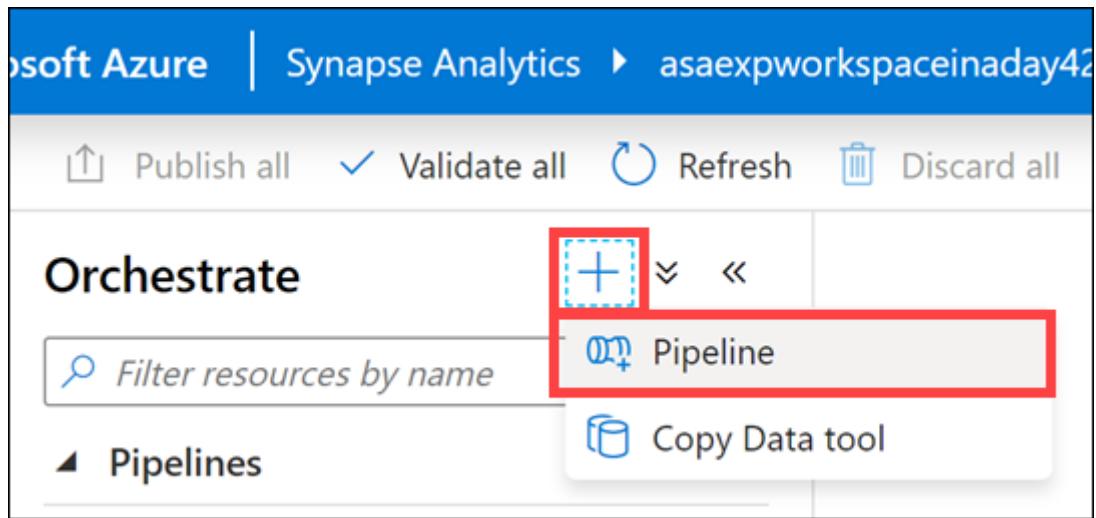
Microsoft Azure | Synapse Analytics ▶ asaexpworkspaceinaday42

Publish all   Validate all   Refresh   Discard all

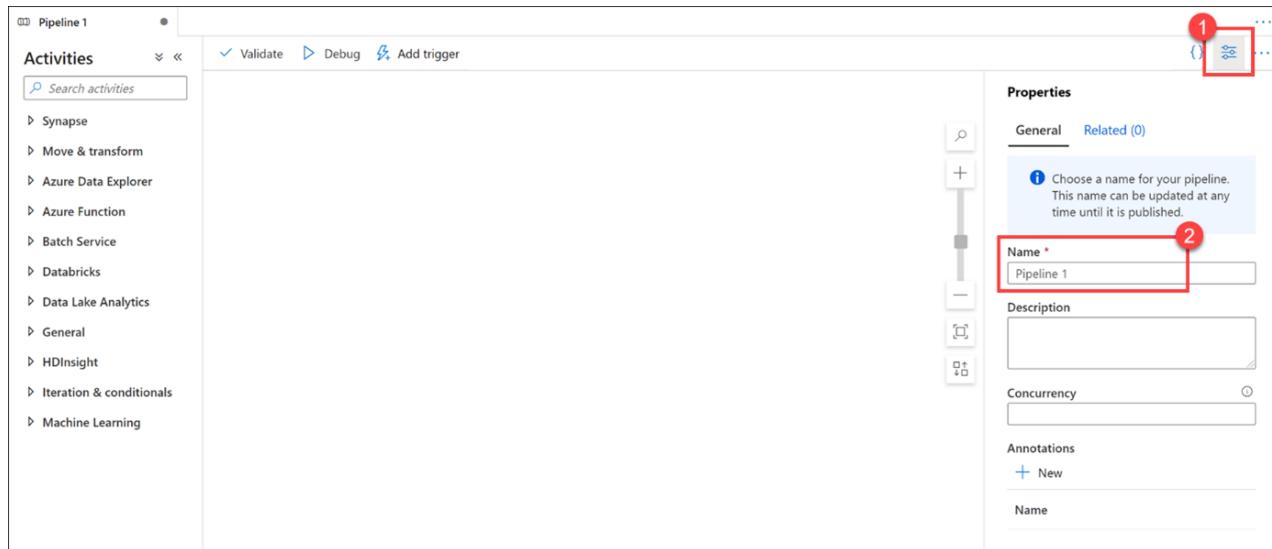
## Orchestrate

Filter resources by name

Pipelines   Pipeline   Copy Data tool



When the new pipeline opens, the **Properties** blade appears (1), allowing you to name the pipeline (2).



1

2

3

Pipeline 1

Activities

Synapse

Move & transform

Azure Data Explorer

Azure Function

Batch Service

Databricks

Data Lake Analytics

General

HDIInsight

Iteration & conditionals

Machine Learning

Validate   Debug   Add trigger

Properties

General   Related (0)

Choose a name for your pipeline. This name can be updated at any time until it is published.

Name \* Pipeline 1

Description

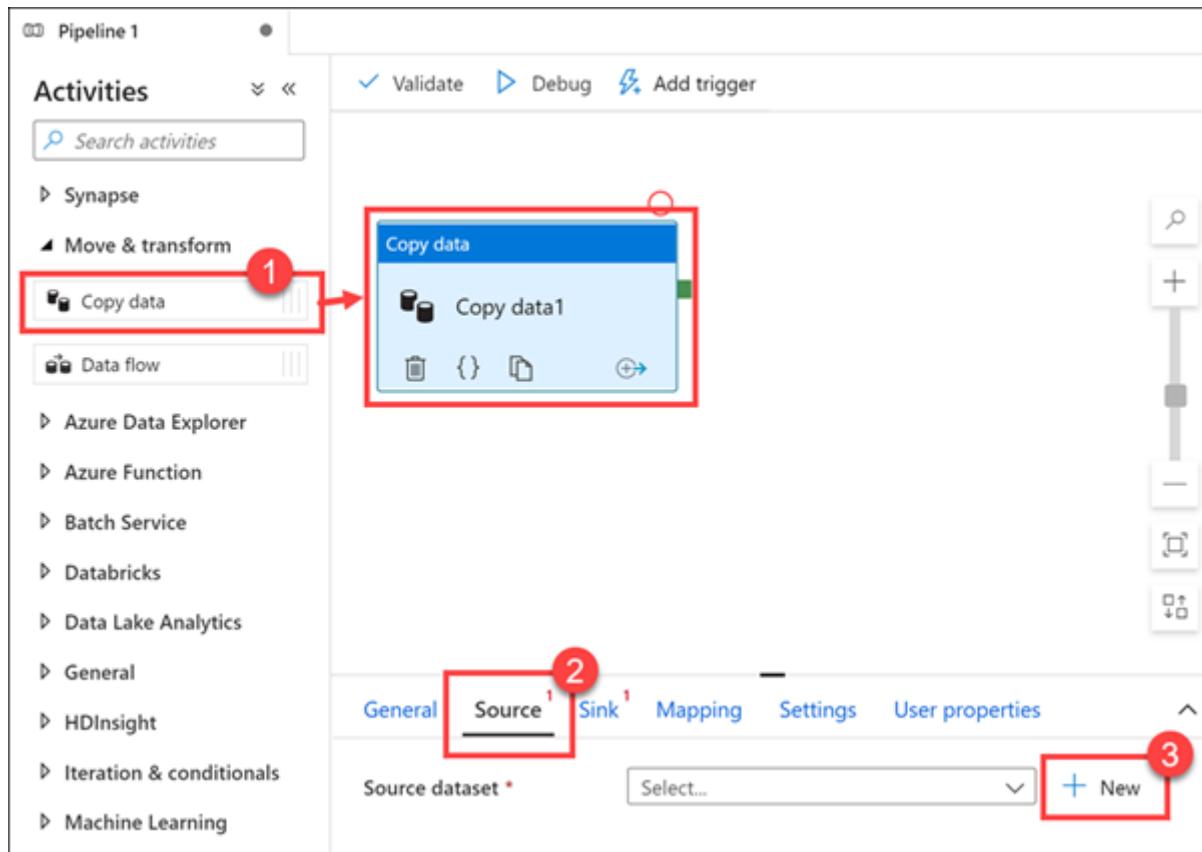
Concurrency

Annotations

+ New

Name

13. Expand the Move & transform activity group, then drag the **Copy data** activity onto the design canvas (1). With the Copy data activity selected, select the **Source** tab (2), then select **+ New** (3) next to the source dataset.



14. Scroll through the list of dataset sources to show the large number of data connections at your disposal, and then click cancel.

## New dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. [Learn more](#)

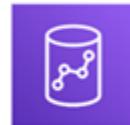
Select a data store

 Search

All    Azure    Database    File    Generic protocol    NoSQL    Services and apps



Amazon Marketplace Web Service



Amazon Redshift



Amazon S3



Apache Impala



Azure Blob Storage



Azure Cosmos DB  
(MongoDB API)



Azure Cosmos DB (SQL)



Azure Data Explorer



Azure Data Lake Storage

Continue

Cancel

## Unlimited data scale

1. Select the Manage hub.



Home



Data



Develop



Integrate



Monitor



Manage

2. Select SQL pools (1). Hover over SQLPool01 and select the Scale button (2).

**Analytics pools**

- SQL pools** (1)
- Apache Spark pools
- External connections
- Linked services
- Orchestration
- Triggers
- Integration runtimes
- Security

**SQL pools**

SQL on-demand is immediately available for your workspace. SQL pools can be configured

+ New Refresh Allow pipelines (Coming soon)

Showing 1-2 of 2 items (1 On-demand, 1 SQL pool)

Name	Type
SQL on-demand	SQL on-demand
SQLPool01	SQL pool (2)

3. Drag the **Performance level** slider right and left.

## Scale

SQLPool01

Configure the settings that best align to the workload needs on the SQL pool. [Learn more about performance levels](#)

Performance level

DW100c ← → DW5000c

Estimated price ⓘ

Est. cost per hour  
60.00 USD

You can scale out or back compute by adjusting the number of Data Warehouse Units (DWUs) assigned to your SQL pool. This adjusts the loading and query performance linearly as you add more units.

To perform a scale operation, SQL pool first kills all incoming queries and then rolls back transactions to ensure a consistent state. Scaling only occurs once the transaction rollback is complete.

You can scale SQL compute at any time by using this slider. You can also programmatically adjust the Data Warehouse Units, enabling scenarios where you automatically scale your pool based on a schedule or other factors.

4. Cancel the Scale dialog, then select **Apache Spark pools** (1) in the Manage hub left-hand menu. Hover over **SparkPool01** and select the **auto-scale settings** button (2).

Analytics pools

- [SQL pools](#)
- [Apache Spark pools](#) 1

External connections

- [Linked services](#)
- [Orchestration](#)
- [Triggers](#)
- [Integration runtimes](#)
- [Security](#)
- [Access control](#)
- [Managed private endpoints](#)

Apache Spark pools

Apache Spark pools can be finely tuned to run different kinds of Apache Spark workloads using specific configuration libraries, permissions, etc. [Learn more](#)

+ New    ⌂ Refresh    ⌂ Search

Showing 1 of 1 item

Name	Size
SparkPool01	Small (4 vCPU / 32 GB) - 3 to 4 nodes

Open auto-scale settings for this pool 2

5. Drag the **Number of nodes** slider right and left.

### Auto-scale settings

[SparkPool01](#)

Configure the settings that best align with the workload on the Apache Spark pool.

**Node size family**  
MemoryOptimized

**Node size \***  
Small (4 vCPU / 32 GB)

**Autoscale \*** ⓘ  
[Enabled](#)   [Disabled](#)

**Number of nodes \***  
3  200  
Estimated price ⓘ  
Est. cost per hour  
2.04 to 136.00 USD

You can configure the Apache Spark pool to have a fixed size by disabling the autoscale setting. Here we have enabled autoscale and set the minimum and maximum number of nodes to control the amount of scale applied. When you enable autoscale, Synapse Analytics monitors the resource requirements of the load and scales the number of nodes up or down. It does this by continuously monitoring pending CPU, pending memory, free CPU, free memory, and used memory per node metrics. It checks these metrics every 30 seconds and makes scaling decisions based on the values.

It can take 1-5 minutes for a scaling operation to complete.

6. Cancel the auto-scale dialog, then select **Linked services** (1) in the Manage hub left-hand menu. Make note of the **WorkspaceDefaultStorage** ADLS Gen2 storage account (2).

The screenshot shows the Azure Synapse Analytics Manage hub. On the left, there's a sidebar with various options like Analytics pools, SQL pools, Apache Spark pools, External connections (with 'Linked services' highlighted by a red box and circled with a red number 1), Orchestration, Triggers, Integration runtimes, Security, Access control, and Managed private endpoints. The main area is titled 'Linked services' and contains a list of items. The list includes:

Name	Type
asaexpdatalakeinaday42	Azure Data Lake Storage Gen2
asaexpkeyvaultinaday42	Azure Key Vault
asaexppowerbiinaday42	Power BI
asaexpworkspaceinaday42-WorkspaceDefaultSqlServer	Azure Synapse Analytics (formerly SQL DW)
asaexpworkspaceinaday42-WorkspaceDefaultStorage	Azure Data Lake Storage Gen2
sqlpool01	Azure Synapse Analytics (formerly SQL DW)

The row for 'asaexpworkspaceinaday42-WorkspaceDefaultStorage' is also highlighted with a red box and circled with a red number 2.

When you provision a new Azure Synapse Analytics workspace, you define the default storage Azure Data Lake Storage Gen2 account. Data Lake Storage Gen2 makes Azure Storage the foundation for building enterprise data lakes on Azure. Designed from the start to service multiple petabytes of information while sustaining hundreds of gigabits of throughput, Data Lake Storage Gen2 allows you to easily manage massive amounts of data.

Its hierarchical namespace organizes files into a hierarchy of directories for efficient access and more granular security, down to the file-level.

ADLS Gen2 provides virtually limitless scale for your data lake. You can attach additional ADLS Gen2 accounts for greater scale and flexibility as needed.

## Familiar tools and ecosystem

1. Select the **Develop** hub.



Home



Data



Develop



Integrate



Monitor



Manage

2. Expand **SQL scripts** and select **1 SQL Query With Synapse (1)**. Make sure you are connected to **SQLPool01 (2)**. **Highlight (3)** the first line of the script and execute. Observe that the number of records in the Sales table is **3,443,486 (4)**.

The screenshot shows the Azure Data Studio interface. On the left, the sidebar has sections for Develop, SQL scripts, Notebooks, Data flows, and Power BI. A red box labeled '1' highlights the 'SQL scripts' section. In the main area, a query titled '1 SQL Query With Sy...' is open. The top bar shows 'Run', 'Undo', 'Publish', 'Query plan', 'Connect to', and a dropdown set to 'SQLPool01'. A red box labeled '2' is over the 'Connect to' dropdown. The query itself starts with 'SELECT COUNT\_BIG(1) as TotalCount FROM dbo.Sales(nolock)'. A red box labeled '3' highlights this line. Below the query, the 'Results' tab is selected, showing a table with one row. The first column is 'TotalCount' and the value is '3443486'. A red box labeled '4' highlights this row.

```

1 SELECT COUNT_BIG(1) as TotalCount FROM dbo.Sales(nolock)
2
3
4 -- 3,443,487
5 --let's execute the below query
6 -- We have Data from SALES,Products,MillennialCustomers and Twitter.
7
8 select CustKey, UserName, Emailstatus, Department, [Twitter Sentiment], cast(round(TotalSale/10000,0) as int) as Revenue
9 from (SELECT P.Department, TA.Sentiment AS [Twitter Sentiment],
10 sum(S.TotalAmount) as TotalSale,
11 M.UserName, M.Emailstatus, M.CustKey
12 FROM dbo.Sales as S
13 inner join dbo.Products as P on P.Products_ID= S.ProductId inner join [dbo].[Dim_Customer] DC
14 left outer join dbo.[TwitterAnalytics] TA on TA.username=DC.userName on DC.id=S.Customerid
15 inner join dbo.[MillennialCustomers] as M on M.CustKey = S.Customerid
16 where DC.FullName!="N/A"
17 group by DC.FullName,P.Department,TA.Sentiment,M.UserName,M.CustKey,M.Emailstatus)
18 as result
19
20

```

CustKey	UserName	Emailstatus	Department	Twitter Sentim...	Revenue
154	Jack82	NULL	Accessories	NULL	0
295	Mabel46	Opened	Entertainment	NULL	0
200	Maureen.Zboncak53	Unopened	Accessories	NULL	0
251	Ryan_Ullrich	Opened	Entertainment	NULL	0
184	Beth.Heathcote	NULL	Accessories	NULL	0
42	Danielle.Blanda	Unopened	Entertainment	NULL	0

If we execute the first line in this SQL script, we can see that we have almost 3.5 million rows contained within.

### 3. Highlight the rest of the script (lines 8 - 18) and execute.

The screenshot shows the Azure Data Studio interface with the same query as before, but now lines 8 through 18 are highlighted with a blue selection. The results table now contains multiple rows of data, showing the joined information from Sales, Products, Dim\_Customer, and MillennialCustomers tables, along with calculated Revenue values.

CustKey	UserName	Emailstatus	Department	Twitter Sentim...	Revenue
154	Jack82	NULL	Accessories	NULL	0
295	Mabel46	Opened	Entertainment	NULL	0
200	Maureen.Zboncak53	Unopened	Accessories	NULL	0
251	Ryan_Ullrich	Opened	Entertainment	NULL	0
184	Beth.Heathcote	NULL	Accessories	NULL	0
42	Danielle.Blanda	Unopened	Entertainment	NULL	0

00:00:02 Query executed successfully.

One of the benefits of using a modern data warehouse like Synapse Analytics is that you can combine all your data in one place. The script we just executed joins data from a sales database, product catalog, millennial customers extracted from demographics data, and twitter.

4. Select the 2 JSON Extractor (1) script and make sure you're still connected to SQLPool01.

Highlight the **first select statement** (2) (line 3). Observe that the data stored in the **TwitterData** column (3) is in JSON format.

The screenshot shows the Azure Synapse Studio interface. On the left, the 'Develop' sidebar lists 'SQL scripts' (1), 'Notebooks' (3), 'Data flows' (1), and 'Power BI' (1). The main area displays the '2 JSON Extractor' script. Line 3 contains the first select statement: 'SELECT top (100) \* from dbo.[TwitterRawData]'. A red box highlights this line, and a red circle with the number '2' is placed above it. Another red box highlights the 'TwitterData' column in the results table, and a red circle with the number '3' is placed above it. The results table shows several rows of data, each containing a JSON object in the 'TwitterData' column.

ID	TwitterData
444	{"Time": "2019-10-25T08:49:51.939000Z", "Hashtag": "#shopping", "Tweet": "RT @ViecBuonBan: Coffee Highl...
924	{"Time": "2019-10-25T08:50:38.245000Z", "Hashtag": "#clothing", "Tweet": "#lookbook #clothing IPEGA PG-90...
1404	{"Time": "2019-10-25T08:51:33.903000Z", "Hashtag": "#fashion", "Tweet": "Under Armour names Patrik Frisk as...
1884	{"Time": "2019-10-25T01:29:21.593000Z", "Hashtag": "#sunglasses", "Tweet": "Sale \$88.35 https://t.co/ce3O3T...
144	{"Time": "2019-10-25T08:39:48.427000Z", "Hashtag": "#chinlaccet", "Tweet": "Sale \$3.95 https://t.co/X9uRQO...

Azure Synapse enables you to store JSON in standard textual format. Use standard SQL language for querying JSON data.

5. Highlight the next SQL statement (**lines 8 - 15**) and execute.

2 JSON Extractor

Run Undo Publish Query plan Connect to SQLPool01 ...

```

1 --JSON Extractor
2   --First, Azure Synapse enables you to store JSON in standard textual format, use standard SQL language fo
3   |   SELECT top (100) * from dbo.[TwitterRawData]
4
5
6   -- Second, let's take JSON data and extract specific structured columns.
7
8   | SELECT
9     JSON_VALUE( TwitterData,'$.Time') AS Time,
10    JSON_VALUE( TwitterData,'$.Hashtag') AS Hashtag,
11    JSON_VALUE( TwitterData,'$.Tweet') AS Tweet,
12    JSON_VALUE( TwitterData,'$.City') AS City ,
13    JSON_VALUE( TwitterData,'$.Sentiment') AS Sentiment ,
14    JSON_VALUE( TwitterData,'$.Language') AS Language
15   FROM dbo.[TwitterRawData] WHERE ISJSON(TwitterData) > 0
16
17   --## Third, let's filter for #sunglasses.
18   --The query below fetches JSON data and filters it by hashtag.<br>

```

Results Messages

View Table Chart Export results ▾

Search

Time	Hashtag	Tweet	City	Sentiment	Language
2019-10-25T08:49:51.9390000Z	#shopping	RT @ViecBuon...	London, England	Positive	English
2019-10-25T08:50:38.2450000Z	#clothing	#lookbook #clo...	Lima, Peru	Neutral	English
2019-10-25T08:51:33.9030000Z	#fashion	Under Armour ...	NULL	Neutral	English
2019-10-25T01:29:21.5930000Z	#sunglasses	Sale \$88.35 htt...	West Hartford, ...	Neutral	English
2019-10-25T08:39:48.4220000Z	#sunglasses	Sale €3.95 http...	West Hartford	Neutral	English

00:00:00 Query executed successfully.

We can use JSON functions, such as JSON\_VALUE and ISJSON to extract the JSON data and extract it to specific structured columns.

6. Highlight the next SQL statement (lines 21 - 29) and execute.

2 JSON Extractor

Run Undo Publish Query plan Connect to SQLPool01 ...

```

16
17 --## Third, let's filter for #sunglasses.
18 --The query below fetches JSON data and filters it by hashtag.<br>
19 --Please note, this extracts specific columns in a structured format
20
21 SELECT
22     JSON_VALUE( TwitterData,'$.Time') AS Time,
23     JSON_VALUE( TwitterData,'$.Hashtag') AS Hashtag,
24     JSON_VALUE( TwitterData,'$.Tweet') AS Tweet,
25     JSON_VALUE( TwitterData,'$.City') AS City ,
26     JSON_VALUE( TwitterData,'$.Sentiment') AS Sentiment ,
27     JSON_VALUE( TwitterData,'$.Language') AS Language
28 FROM dbo.[TwitterRawData]
29 WHERE ISJSON(TwitterData) > 0 And JSON_VALUE( TwitterData,'$.Hashtag')='#sunglasses'
30
31
32 -- petsa
33 select * from (

```

Results Messages

View Table Chart Export results

Search

Time	Hashtag	Tweet	City	Sentiment	Language
2019-10-25T01:29:21.5930000Z	#sunglasses	Sale \$88.35 htt...	West Hartford, ...	Neutral	English
2019-10-25T08:39:48.4220000Z	#sunglasses	Sale \$3.95 http...	West Hartford, ...	Neutral	English
2019-10-25T01:43:03.5990000Z	#sunglasses	Just a simple gi...	New Jersey	Neutral	English
2019-10-25T01:24:06.5320000Z	#sunglasses	RT @Simplious...	NULL	Positive	English

00:00:00 Query executed successfully.

We want to filter for the #sunglasses hashtag. This query fetches and extracts the JSON data into structured columns, then filters on the derived Hashtag column.

The last script does the same thing, but just using a subquery format.

7. Select the 8 External Data To Synapse Via Copy Into (1) script. DO NOT EXECUTE. Scroll through the script file, using the commentary below to explain what it does.

```

1 --Step 1 Let's create table
2 IF OBJECT_ID(N'dbo.Twitter', N'U') IS NOT NULL
3 BEGIN
4     DROP TABLE [dbo].[Twitter]
5 END
6 GO
7 SET ANSI_NULLS ON
8 GO
9 SET QUOTED_IDENTIFIER ON
10 GO
11 CREATE TABLE [dbo].[Twitter]
12 (
13     [Time] [nvarchar](4000) NULL,
14     [Hashtag] [nvarchar](4000) NULL,
15     [Tweet] [nvarchar](4000) NULL,
16     [City] [nvarchar](4000) NULL,
17     [UserName] [nvarchar](4000) NULL,
18     [RetweetCount] [int] NULL,
19     [FavouriteCount] [int] NULL,
20     [Sentiment] [nvarchar](4000) NULL,
21     [SentimentScore] [int] NULL,
22     [IsRetweet] [int] NULL,
23     [HourOfDay] [nvarchar](4000) NULL,
24     [Language] [nvarchar](4000) NULL
25 )
26 WITH
27 (
28     DISTRIBUTION = ROUND_ROBIN,
29     CLUSTERED COLUMNSTORE INDEX
30 );
31 GO
32
33 -- Step 2 Copy data from all PARQUET files in to the table
34 COPY INTO [dbo].[Twitter]

```

In this script, we create a table to store Twitter data stored in Parquet files. We use the **COPY** command to quickly and efficiently load all data stored in Parquet files into the new table.

Finally, we select the first 10 rows to verify the data load.

The **COPY** command and PolyBase can be used to import data from various formats into the SQL pool, either through T-SQL scripts like we see here, or from orchestration pipelines.

## 8. Select the Data hub.



Home



Data



Develop



Integrate

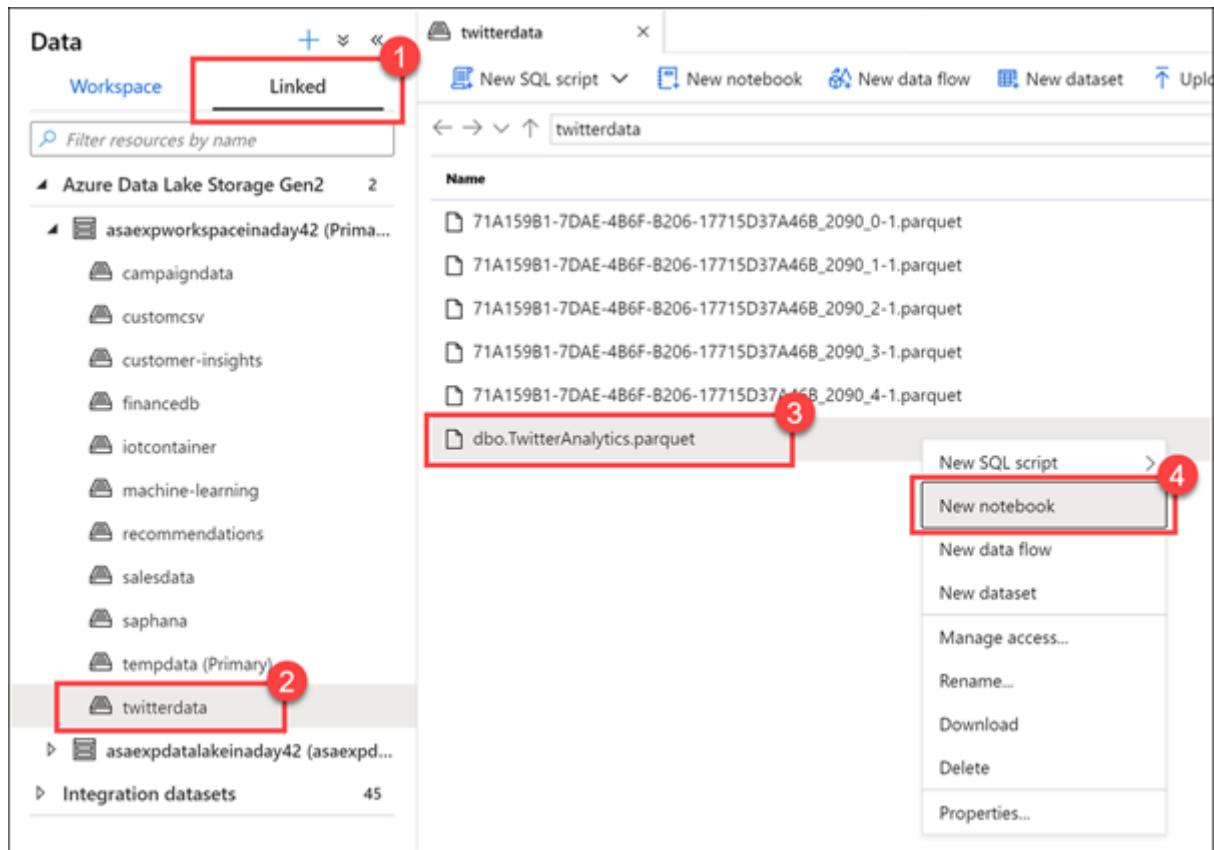


Monitor



Manage

9. Select the **Linked tab** (1), expand the Azure Data Lake Storage Gen2 group, expand the Primary storage account, then select the **twitterdata** container (2). Right-click on the **dbo.TwitterAnalytics.parquet** file (3), then select **New notebook** (4).



Synapse Studio provides several options to work with files stored in attached storage accounts, such as creating a new SQL script, a notebook, data flow, or new dataset.

Synapse Notebooks enable you to harness the power of Apache Spark to explore and analyze data, conduct data engineering tasks, and do data science. Authentication and authorization with linked services, such as the primary data lake storage account, are fully integrated, allowing you to immediately start working with files without dealing with account credentials.

Here we see a new notebook that loads a Spark DataFrame (1) with the Parquet file that we right-clicked on in the Data hub. We can immediately start exploring the file contents in just a couple simple steps. At the top of the notebook, we see that it is attached to **SparkPool01**, our Spark pool, and the notebook language is set to **Python** (2).

Do not execute the notebook unless the Spark pool is ready (3). It can take up to 5 minutes to start the pool if it is idle. Alternatively, you can execute the notebook, then come back to it later to view the results.

1

2

3

Notebook 1

Cell 1

1 %pyspark  
2 data\_path = spark.read.load("abfss://twitterdata@asaexpdatalakeinaday42.dfs.core.windows.net/db0.TwitterAnalytics.p  
3 display(data\_path.limit(10))

Command executed in 4mins 16s 4ms by joel on 09-07-2020 17:23:40.984 -04:00

> Job execution Succeeded Spark 2 executors 8 cores

View Table Chart

Time	Hashtag	Tweet	City	UserName
2019-10-25T08:49:19.9490000Z	#shopping	RT @ViecBuonBan: Coffee Highla...	Potsdam, Germany	Johnny_Medhurst3
2019-10-25T08:49:36.6220000Z	#fashion	Sports Bra Club 🎈 Top by GymB...	Zagreb, Croatia	Sergio91
2019-10-25T08:51:29.6250000Z	#fashion	Into her Eyes flickr https://t.co/...	Zürich	Angelina_Witting
2019-10-25T08:51:42.1490000Z	#fashion	#fashion #beautiful #DMTBeauty...	Springfield Gardens, Queens	Jay.Hegmann
2019-10-25T08:39:48.4220000Z	#sunglasses	Sale \$3.95 https://t.co/X9wBB0ei...	West Hartford, CT	Edmond_Beatty
2019-10-25T08:49:21.3940000Z	#fashion	If u want to buy this dress text m...		Marjorie.Heaney50
2019-10-25T08:51:18.3110000Z	#fashion	#makeup #cosmetic #fashion #e...		Emilio_Mohr
2019-10-25T08:51:40.7000000Z	#fashion	Spitfire Petit expanding in NE, S ...		Rhonda_Simonis55
2019-10-25T08:49:51.9390000Z	#shopping	RT @ViecBuonBan: Coffee Highla...	London, England	Erica.Gorczany
2019-10-25T08:49:45.1750000Z	#fashion	RT @Glamaroni: So good I had t...	Castleford, Yorkshire, UK	Lyle.Ratke47

Ready | Stop session | Configure session