



Welcome to Azure Synapse Technical Boot Camp

Day 1

A look into Day 1

Kick-Off	8:00-8:05	Welcome	Main Call
	8:05-8:30	Azure Synapse Analytics 101	
	8:30-8:45	Demo Walkthrough	
Ingest	8:45-9:00	Break	Table Group Call
	9:00-10:00	Data Loading & Data Lake Organization	
	10:00-11:00	Activity: Data Lake Design & Security Considerations	
	11:00-11:15	Break	
	11:15-12:00	Build Hands-on: Data Integration Part 1	
Transform	12:00-1:00	Break	Main Call
	1:00-1:30	Data Transformations	
	1:30-2:00	Activity: Data Engineering Discussion	
	2:00-3:00	Build Hands-on: Data Integration Part 2	
	3:00-3:15	Closing	

Today we will be learning and collaborating across three spaces:

- Main call (this meeting)
- Table Group channel within the event Team
- CloudLabs Learner Portal & Synapse environment

■ Presentation/
Whole Group

■ Lab

■ Activity/ Discussion/
Group Work

■ Announcements

Azure Synapse 101

Studio

A single place for Data Engineers, Data Scientists, and IT Pros to collaborate on enterprise analytics

The screenshot shows the Microsoft Azure Synapse Analytics Studio interface. The top navigation bar includes 'Microsoft Azure' and 'Synapse Analytics' with a workspace name 'wsazuresynapseanalytics'. The top right corner shows a user email 'someone@microsoft.com' and the Microsoft logo.

The left sidebar contains a navigation menu with icons and labels: Home, Data, Develop, Integrate, Monitor, and Manage. The 'Home' item is currently selected.

The main content area is titled 'Synapse workspace' and displays the workspace name 'wsazuresynapseanalytics'. It features a large circular graphic illustrating data flow and analysis, with a 3D bar chart overlaid. Below this are four cards: 'Ingest' (Perform a one-time or scheduled data load), 'Explore and analyze' (Learn how to get insights from your data), 'Visualize' (Build interactive reports with Power BI capabilities), and 'Learn' (Start with Azure Open Datasets and sample code).

The 'Recent resources' section lists five items:

Name	Last opened by you
05 Sentiment_Analysis_Cognitive_Services	4 hours ago
Predict NYCTaxi Trip Amount	4 hours ago
001 SQL Pool Security RLS DDM CLE	5 hours ago
005 Predict In-Engine Scoring	a day ago
05 Anomaly_Detection_Cognitive_Services	a day ago

At the bottom, there is a 'Show more ▾' link.

Synapse Studio

Synapse Studio divided into **Activity hubs**.

These organize the tasks needed for building analytics solution.

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. On the left, a vertical sidebar menu is highlighted with a red box, containing the following items:

- Home
- Data
- Develop
- Integrate
- Monitor
- Manage

A red arrow points from the "Integrate" menu item to the "Integrate" activity hub icon on the main dashboard. The main dashboard displays five activity hubs:

- Home**: Quick-access to common gestures, most-recently used items, and links to tutorials and documentation.
- Data**: Explore structured and unstructured data.
- Develop**: Write code and define business logic of the pipeline via notebooks, SQL scripts, Data flows, etc.
- Integrate**: Design pipelines that move and transform data.
- Monitor**: Centralized view of all resource usage and activities in the workspace.
- Manage**: Configure the workspace, pool, linked service, access to artifacts.

Home Hub

Ease of access to get updates, to switch workspace, to get notifications and to provide feedback

The screenshot shows the Microsoft Azure Synapse Analytics Home Hub. The top navigation bar includes 'Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics' and a user profile for 'someone@microsoft.com MICROSOFT'. The main content area is titled 'Synapse workspace' and 'wsazuresynapseanalytics'. It features a 'New' button and several cards: 'Ingest' (Perform a one-time or scheduled data load), 'Explore and analyze' (Learn how to get insights from your data), 'Visualize' (Build interactive reports with Power BI capabilities), and 'Learn' (Start with Azure Open Datasets and sample code). Below this is a section for 'Recent resources' with a table:

Name	Last opened by you
05 Sentiment_Analysis_Cognitive_Services	4 hours ago
Predict NYCTaxi Trip Amount	4 hours ago
001 SQL Pool Security RLS DDM CLE	5 hours ago
005 Predict In-Engine Scoring	a day ago

Home Hub

It is a starting point for the activities with key links to tasks, artifacts, documentation and sample artifacts for learning purpose

The screenshot shows the Microsoft Azure Synapse Analytics Home Hub for the workspace 'wsazuresynapseanalytics'. The left sidebar includes links for Home, Data, Develop, Integrate, Monitor, and Manage. The main area displays a 'Synapse workspace' title and four activity cards: Ingest, Explore and analyze, Visualize, and Learn. A red box highlights the first four cards. Below them is a 'Recent resources' section listing five items: 05 Sentiment_Analysis_Cognitive_Services, Predict NYCTaxi Trip Amount, 001 SQL Pool Security RLS DDM CLE, 005 Predict In-Engine Scoring, and 05 Anomaly_Detection_Cognitive_Services. The 'Learn' card contains the following text:

Learn
Start with Azure Open Datasets and sample code.

Name	Last opened by you
05 Sentiment_Analysis_Cognitive_Services	4 hours ago
Predict NYCTaxi Trip Amount	4 hours ago
001 SQL Pool Security RLS DDM CLE	5 hours ago
005 Predict In-Engine Scoring	a day ago
05 Anomaly_Detection_Cognitive_Services	a day ago

Knowledge center

Knowledge center offers open datasets, sample notebooks, SQL scripts and pipeline templates for easy start and learning

Use samples immediately

Create everything you need in just one click.

Explore sample data with Spark
Includes a sample script. If you have permissions, we'll create a new pool for you; otherwise, you can use an existing pool.
Name SampleSpark
Size Medium (8 vCores / 64 GB) - 3 nodes

Query data with SQL
Includes a sample script and serverless SQL pool - Built-in (included with your workspace).

Create external table with SQL
Includes a sample script. You can use serverless SQL pool - Built-in (included with your workspace) or a dedicated SQL pool. We will create a table for you called SampleTable.
 Create a pool Select an existing pool
Name SampleSQL
Size DW100c

[Use sample](#) [Cancel](#)

Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics

Gallery

Datasets Notebooks SQL scripts Pipelines

Filter by keyword Tags : All

 Bing COVID-19 Data Bing COVID-19 data includes confirmed, fatal, and recovered cases from all regions, updated da... ID: bing-covid-19-data Sample	 Boston Safety Data Read data about 311 calls reported to the city of Boston. This dataset is stored in Parquet format and is up... ID: city_safety_boston Sample	 COVID Tracking Project The COVID Tracking Project dataset provides the latest numbers on tests, confirmed cases, hospitalizat... ID: covid-tracking Sample	 Chicago Safety Data Read data about 311 calls reported to the city of Chicago. This dataset is stored in Parquet format and is ... ID: city_safety_chicago Sample
 European Centre for Disease Prevention and Control (ECDC) Covid-19 Cases The latest available public data on... ID: ecdc-covid-19-cases Sample	 NOAA Integrated Surface Data (ISD) NOAA Integrated Surface Data (ISD) provides Worldwide hourly weath... ID: isd Sample	 NYC Taxi & Limousine Commission - For-Hire Vehicle (FHV) trip records The For-Hire Vehicle trip records i... ID: nyc_tlc_fhv Sample	 NYC Taxi & Limousine Commission - green taxi trip records The green taxi trip records include... ID: nyc_tlc_green Sample

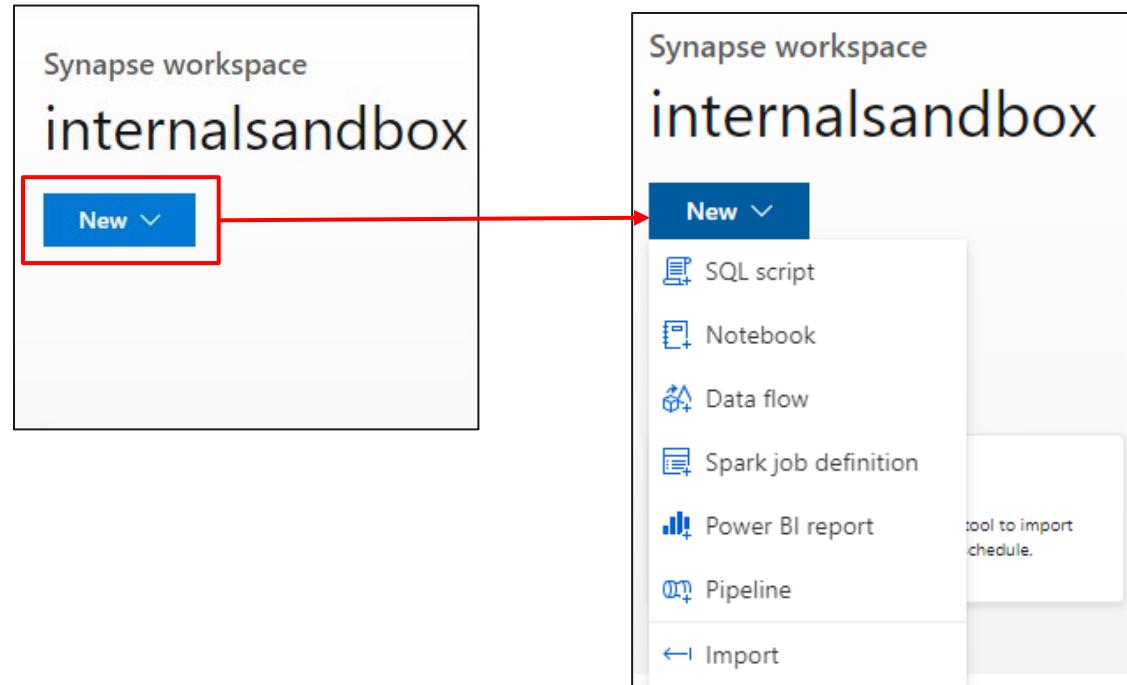
Continue [Close](#)

Home Hub

Overview

New dropdown – offers quickly start work item

Recent & Pinned – Lists recently opened code artifacts. Pin selected ones for quick access



Recent	Pinned		
		NAME	LAST OPENED BY YOU
		BOOT_AMLautoMLPredict	6 hours ago
		SQLConnector	6 hours ago
		TaxiCreateSparkTable	6 hours ago
		Notebook 1	6 hours ago
		NYCTAx1	6 hours ago
Show more ▾			

Recent	Pinned		
		NAME	LAST OPENED BY YOU
		NYCTAx1	6 hours ago

Data Hub

Explore data inside the workspace and in linked storage accounts

Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics

The screenshot shows the Microsoft Azure Synapse Analytics Data workspace page. On the left, there's a sidebar with icons for Home, Data, Develop, Integrate, Monitor, and Manage. The main area has a 'Data' tab selected, with 'Synapse live' and 'Validate all' buttons at the top. Below the tabs is a search bar and a 'Filter resources by name' input field. Under the 'Workspace' tab, there's a section for 'Databases' which lists several SQL databases: newpoll, NYCTaxi_Pool, Predict_Pool, Streaming_Pool, WWI_Pool, NYT2020, SQLServerlessDB, and default. A red box highlights the 'Workspace' tab.

Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics

The screenshot shows the Microsoft Azure Synapse Analytics Data linked storage accounts page. The sidebar and tabs are identical to the workspace page. Under the 'Linked' tab, there's a section for 'Integration datasets' which lists several linked storage accounts: Azure Blob Storage, Azure Cosmos DB, Azure Data Explorer, Azure Data Lake Storage Gen2, wsazuresynapseanalytics (Primary...), (Attached Containers), and Integration datasets. A red box highlights the 'Linked' tab.

Data Hub – Linked Storage

Browse Azure Data Lake Storage Gen2 accounts – filesystems, Azure Data Explorer – clusters, Azure Cosmos DB -containers

The screenshot shows the Microsoft Azure Synapse Analytics Data Hub interface. On the left, the 'Data' sidebar lists various linked storage resources:

- Linked Cosmos DB Analytical Store**: Points to the 'Azure Cosmos DB' item.
- Linked Azure Data Explorer**: Points to the 'Azure Data Explorer' item.
- Linked ADLS Gen2 Account**: Points to the 'wsazuresynapseanalytics (Primary...)' item.
- Container (filesystem)**: Points to the 'rawdata' item under the ADLS Gen2 account.

The main workspace area shows a dataset named 'rawdata' with a file path of 'rawdata/taxidata'. The data table lists six cached items:

Name	Last Modified	Content Type	Size
part-00000-0300809f-304e-44bc-81bd-bbd63974c3e4-c000.snappy.parq...	8/27/2020, 12:32:19 AM		121.9 MB
part-00000-6b990121-0341-456c-8723-aec72b03f65f-c000.snappy.parqu...	8/27/2020, 12:32:25 AM		535.4 MB
part-00001-0300809f-304e-44bc-81bd-bbd63974c3e4-c000.snappy.parq...	8/27/2020, 12:32:20 AM		124.5 MB
part-00001-6b990121-0341-456c-8723-aec72b03f65f-c000.snappy.parqu...	8/27/2020, 12:32:23 AM		983.7 MB
part-00002-0300809f-304e-44bc-81bd-bbd63974c3e4-c000.snappy.parq...	8/27/2020, 12:32:19 AM		123.7 MB
part-00002-6b990121-0341-456c-8723-aec72b03f65f-c000.snappy.parqu...	8/27/2020, 12:32:21 AM		966.1 MB

At the bottom, it says 'Showing 1 to 6 of 6 cached items'.

Data Hub – Storage accounts

Preview a sample of your data

The screenshot illustrates the process of previewing data from an Azure Data Lake Storage Gen2 account. On the left, the 'Data' blade shows a list of resources, including 'Azure Blob Storage', 'Azure Cosmos DB', 'Azure Data Explorer', 'Azure Data Lake Storage Gen2', and 'Integration datasets'. The 'rawdata' folder under 'wsazuresynapseanalytics (Primary...)' is selected. In the center, the 'rawdata' blade displays a file named 'Products.csv'. A red arrow points from the 'Preview' option in the context menu of 'Products.csv' to the preview pane on the right. The preview pane shows the first 10 rows of the 'Products.csv' file, which contains columns: PRODUCTID, PRODUCTNAME, PRODUCTCATEGORY, and UNITPRICE. The data includes items like Apple, Banana, Avocado, Oranges, Onion, Potato, Broccoli, Beef, and Chicken.

PRODUCTID	PRODUCTNAME	PRODUCTCATEGORY	UNITPRICE
406032	Apple	100	2.48
406064	Banana	100	1.49
406096	Avocado	100	3.49
406128	Oranges	100	2.99
406160	Onion	100	3.49
406192	Potato	100	5.49
406224	Broccoli	100	6.49
406256	Beaf	100	10.49
406288	Chicken	100	20.49

Data Hub – Storage accounts

See basic file properties

The screenshot illustrates the process of viewing basic file properties in the Azure Data Hub interface.

Left Panel (Data View):

- The "Linked" tab is selected under the "Data" section.
- The "rawdata" folder is selected in the navigation tree.
- A context menu is open over the "Products.csv" file, listing options: New SQL script, New notebook, New data flow, New integration dataset, Manage access..., Rename..., Download, Delete, and Properties... (the latter is highlighted with a red box).

Right Panel (Properties Dialog):

The "Properties" dialog shows the following details for the "Products.csv" file:

- Name:** sample csv files/Products.csv
- URL:** https://azuresynapsesa.dfs.core.windows.net/rawdata/sample csv files/Products.csv
- ABFSS Path:** abfss://rawdata@azuresynapsesa.dfs.core.windows.net/sample csv files/Products.csv
- Last modified:** 10/27/2020, 8:38:51 PM
- Cache Control:** max-age=0
- Content Type:** application/octet-stream
- Content Disposition:** (empty)
- Content Encoding:** (empty)
- Content Language:** (empty)
- User Properties:** (empty)

At the bottom of the dialog are "Apply" and "Cancel" buttons.

Data Hub – Storage accounts

Manage Access - Configure standard POSIX ACLs on files and folders

The screenshot illustrates the process of managing access to a file in Azure Data Explorer. On the left, the 'Data' workspace shows a list of storage accounts and datasets. In the center, the 'rawdata' dataset is selected, displaying a list of files including 'Products.csv'. A context menu is open over 'Products.csv', with the 'Manage access...' option highlighted by a red box and a red arrow pointing to the 'Manage Access' dialog on the right.

Data

- Workspace
- Linked

Filter resources by name

- ▷ Azure Blob Storage 3
- ▷ Azure Cosmos DB 1
- ▷ Azure Data Explorer 2
- ◁ Azure Data Lake Storage... 1
- ◀ wsazuresynapseanalytics...
 - default (Primary)
 - rawdata
 - staging
- ▷ Integration datasets 24

rawdata

New SQL script New notebook

Products.csv

Name Last Modified

- Preview
- New SQL script >
- New notebook >
- New data flow
- New integration dataset
- Manage access...**
- Rename...
- Download
- Delete
- Properties...

Manage Access

Users, groups, and service principals:

- \$superuser (Owner)
- \$superuser (Owning Group)
- Other
- Mask

Permissions for: \$superuser

Read	Write	Execute
<input checked="" type="checkbox"/> Access	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add user, group, or service principal:

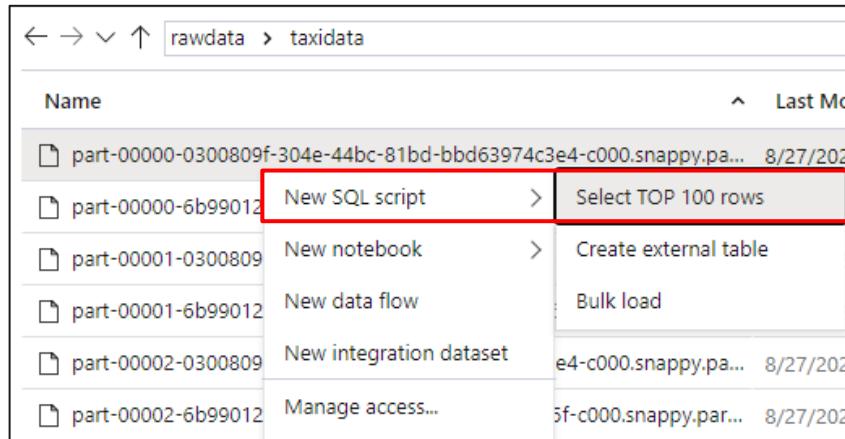
Enter a UPN or Object ID

Save Cancel

Data Hub – Storage accounts

Two simple gestures to start analyzing with SQL scripts or with notebooks.

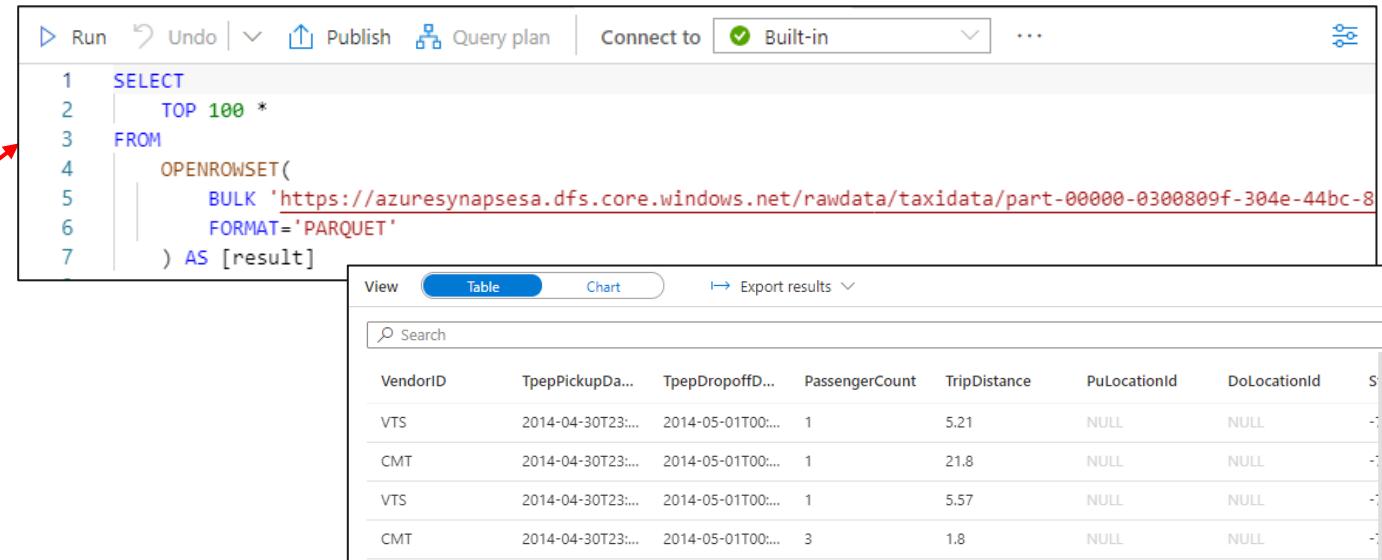
T-SQL or PySpark auto-generated.



rawdata > taxidata

Name

- part-00000-0300809f-304e-44bc-81bd-bbd63974c3e4-c000.snappy.parquet 8/27/2022
- New SQL script > Select TOP 100 rows
- New notebook > Create external table
- New data flow Bulk load
- New integration dataset e4-c000.snappy.parquet 8/27/2022
- Manage access...

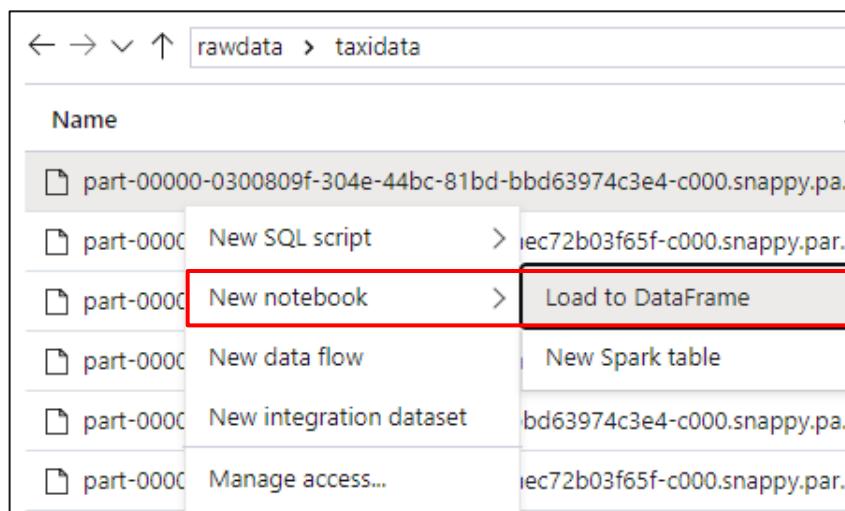


Run Undo Publish Query plan Connect to Built-in ...

```
1 SELECT
2     TOP 100 *
3     FROM
4     OPENROWSET(
5         BULK 'https://azuresynapsesa.dfs.core.windows.net/rawdata/taxidata/part-00000-0300809f-304e-44bc-81bd-bbd63974c3e4-c000.snappy.parquet',
6         FORMAT='PARQUET'
7     ) AS [result]
```

View Table Chart Export results

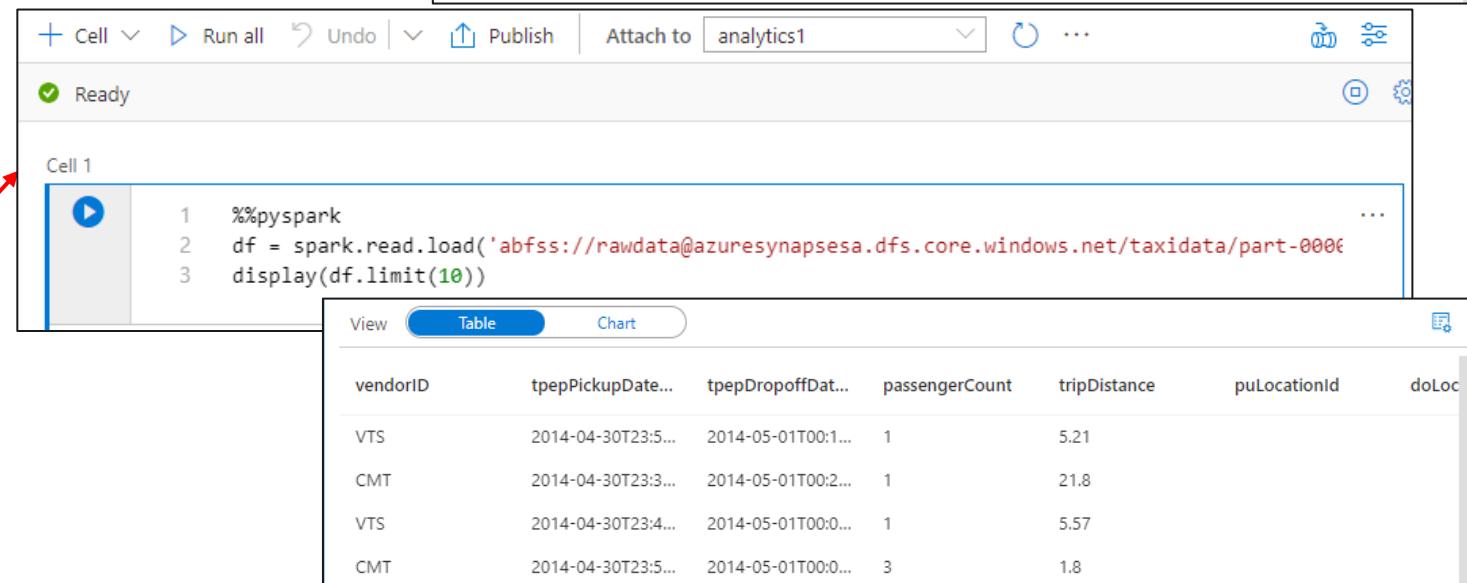
VendorID	TpepPickupDate	TpepDropoffDate	PassengerCount	TripDistance	PuLocationId	DoLocationId	...
VTS	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	1	5.21	NULL	NULL	-
CMT	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	1	21.8	NULL	NULL	-
VTS	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	1	5.57	NULL	NULL	-
CMT	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	3	1.8	NULL	NULL	-



rawdata > taxidata

Name

- part-00000-0300809f-304e-44bc-81bd-bbd63974c3e4-c000.snappy.parquet
- New SQL script > iec72b03f65f-c000.snappy.parquet
- New notebook > Load to DataFrame
- New data flow New Spark table
- New integration dataset bd63974c3e4-c000.snappy.parquet
- Manage access... iec72b03f65f-c000.snappy.parquet



+ Cell Run all Undo Publish Attach to analytics1 ...

Ready

Cell 1

```
1 %%pyspark
2 df = spark.read.load('abfss://rawdata@azuresynapsesa.dfs.core.windows.net/taxidata/part-00000-0300809f-304e-44bc-81bd-bbd63974c3e4-c000.snappy.parquet')
3 display(df.limit(10))
```

View Table Chart

vendorID	tpepPickupDate	tpepDropoffDate	passengerCount	tripDistance	puLocationId	doLoc...
VTS	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	1	5.21		-
CMT	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	1	21.8		-
VTS	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	1	5.57		-
CMT	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	3	1.8		-

Data Hub – Databases

Explore the different kinds of databases that exist in a workspace.



Data Hub – Databases

Familiar gesture to generate T-SQL scripts from SQL metadata objects such as tables.

A screenshot of the Data Hub interface. On the left, there's a tree view of databases: 'sql1 (SQL pool)' which contains 'Tables' (including 'dbo.SearchLogTable' and 'dbo.NycTaxiPredict') and 'Columns' (including 'vendorID', 'passengerC', 'tripDistance', 'puLocationId', 'doLocationId', and 'Predicted_fareA...'). A context menu is open over the 'Predicted_fareA...' column, showing options: 'New SQL script' (with a dropdown for 'Select TOP 1000 rows', 'CREATE', 'DROP', and 'DROP and CREATE'), 'New notebook' (with a dropdown for 'Refresh'), and a separator line.

Starting from a table, auto-generate a single line of PySpark code that makes it easy to load a SQL table into a Spark dataframe

A screenshot of the Data Hub interface. The tree view shows 'sql1 (SQL pool)' with 'Tables' (including 'dbo.SearchLogTable' and 'dbo.NycTaxiPredict'). Under 'dbo.NycTaxiPredict', a context menu is open over the 'Predicted_fareA...' column, with the 'Load to DataFrame' option highlighted by a red box and a red arrow pointing down to a PySpark notebook cell below. The notebook cell contains the following Python code:

```
val df = spark.read.sqlanalytics("sql1.dbo.NycTaxiPredict")
```

Data Hub – Datasets

Orchestration datasets describe data that is persisted. Once a dataset is defined, it can be used in pipelines and sources of data or as sinks of data.

The screenshot shows the Azure Data Hub - Datasets interface. On the left, there is a sidebar with a 'Data' section containing a search bar and a list of resources: Storage accounts (2), Databases (3), Datasets (2), CabDataCooked, and NYCTaxiParquet. The NYCTaxiParquet item is highlighted with a red rectangle and has a red arrow pointing from the sidebar to the main content area. The main content area displays the 'NYCTaxiParquet' dataset details. The dataset icon is a blue Parquet file symbol. The dataset name is 'NYCTaxiParquet'. Below the icon, there are tabs for General, Connection, Schema, and Parameters. The Connection tab is selected. Under the Connection tab, the 'Linked service' dropdown is set to 'Lake_ArcadiaLake'. There are buttons for 'Test connection', 'Open', and 'New'. The 'File path' field contains 'data / nyctaxi / File', with 'Browse' and 'Preview data' buttons next to it. The 'Compression type' field is set to 'snappy'.

Develop Hub

Overview

It provides development experience to query, analyze, model data

Benefits

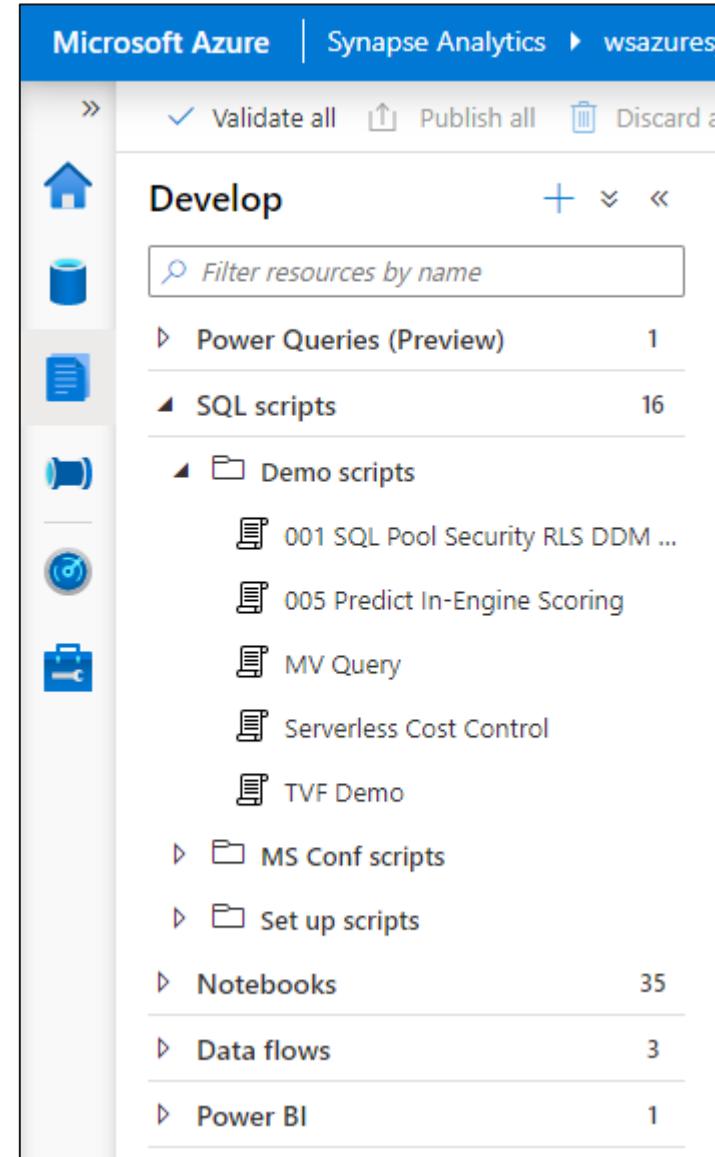
Multiple languages to analyze data under one umbrella

Switch over notebooks and scripts without loosing content

Code intellisense offers reliable code development

Create insightful visualizations

Organize artifacts in folders and sub-folders



Develop Hub - SQL scripts

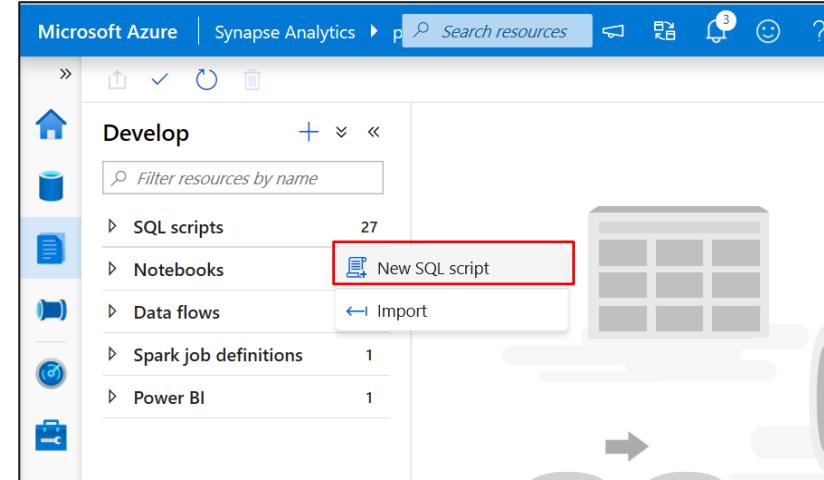
SQL Script

Authoring SQL Scripts

Execute SQL script on dedicated SQL pool or serverless SQL pool

Commit individual SQL script or multiple SQL scripts through Commit all feature

Language support and intellisense



The screenshot shows the Microsoft Azure Synapse Analytics Develop Hub with an open SQL script editor. The top navigation bar includes 'Microsoft Azure', 'Synapse Analytics', 'Search resources', and a user profile. The left sidebar is identical to the previous screenshot. The main area shows an 'SQL script 2' tab with the following code:

```
1 -- type your sql script here, we now have intellisense
2 CREATE
```

An Intellisense dropdown is open over the word 'CREATE', listing options: 'CREATE', 'CURRENT_TIMESTAMP', and 'CURRENT_USER'. The email 'prlangad@microsoft.com' and 'MICROSOFT' are visible in the top right corner.

Develop Hub - SQL scripts

SQL Script

View results in Table or Chart form and export results in several popular formats

The screenshot shows the Azure Data Studio interface with a query editor and a results pane.

Query Editor:

```
1 SELECT
2 TOP 100 *
3 FROM
4 OPENROWSET(
5      BULK 'https://arcadialake.dfs.core.windows.net/users/saveenr/SearchLog.csv',
6      FORMAT='CSV'
7 )
8 WITH (
9      id int,
10     [time] datetime,
11     region varchar(50),
12     searchtext varchar(200),
13     latency int,
14     links varchar(500),
15     clickedlinks varchar(500)
16 ) AS searchlog;
17
```

Results Pane:

The results pane displays the query results in two formats: **Table** and **Chart**.

Table View:

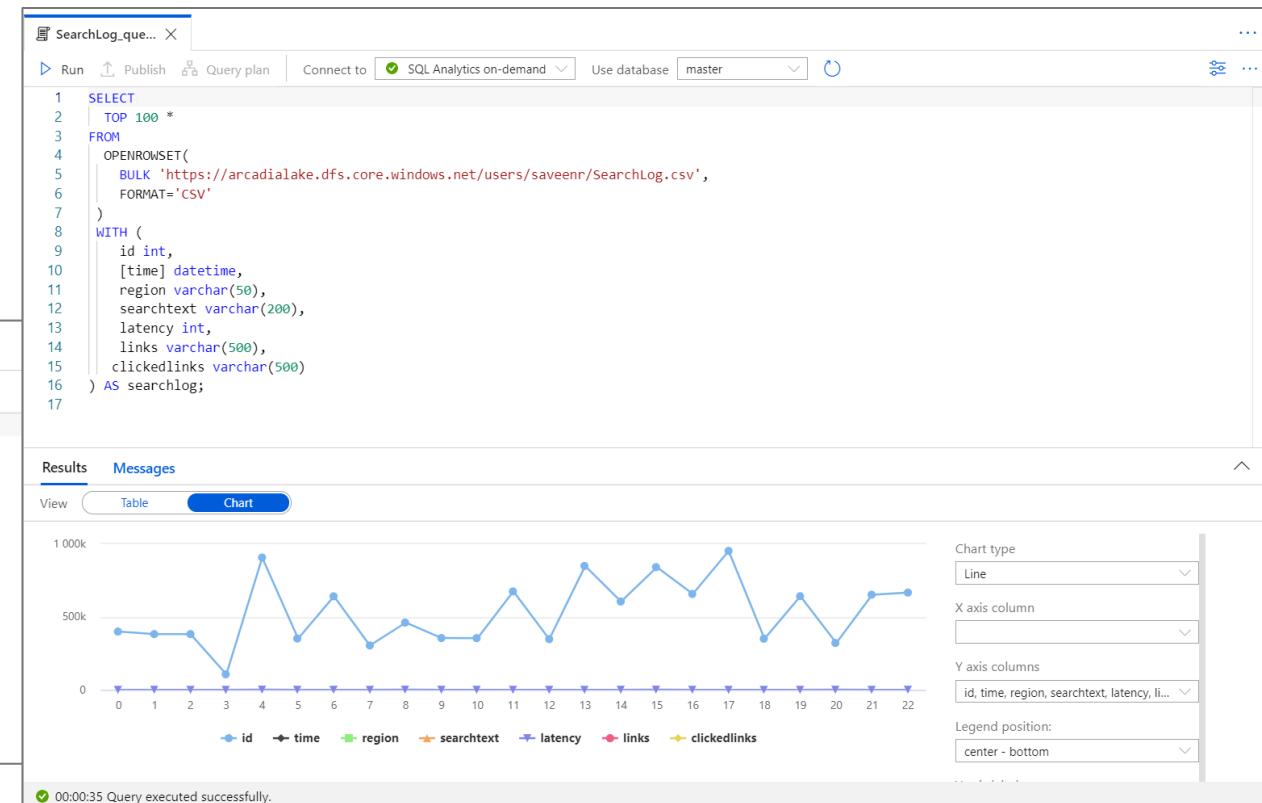
ID	TIME	REGION
399266	2019-10-15T11:53:04.0000000	en-us
382045	2019-10-15T11:53:25.0000000	en-gb
382045	2019-10-16T11:53:42.0000000	en-gb
106479	2019-10-16T11:53:10.0000000	en-ca
906441	2019-10-16T11:54:18.0000000	en-us

Chart View:

A line chart showing the distribution of search logs over time. The X-axis represents the index of the log entry, and the Y-axis represents the count of logs, ranging from 0 to 1,000k. The chart shows a fluctuating trend with peaks around index 4, 13, and 17.

Export Options:

An arrow points from the "Table" tab in the results pane to a red-bordered menu titled "Export results". This menu lists four export formats: CSV, Excel, JSON, and XML.



Develop Hub - Notebooks

As notebook cells run, the underlying Spark application status is shown. Providing immediate feedback and progress tracking.

The screenshot shows the Microsoft Azure Synapse Analytics Develop Hub - Notebooks interface. At the top, there's a navigation bar with 'Microsoft Azure' and 'Synapse Analytics' followed by a search bar 'Search resources'. On the right, there are icons for notifications, help, and a user profile with the email 'prlangad@microsoft.com' and 'MICROSOFT'.

The main area shows a notebook titled 'opendataset' with a single cell labeled 'Cell 1'. The cell contains the following PySpark code:

```
%pyspark  
data_path = spark.read.load('abfss://opendata@internalsandboxwe.dfs.core.windows.net/holidays/part-00000-bd1ab:  
data_path.show(100)
```

Below the code, a message indicates it was executed in 2mins 44s 998ms by 'prlangad' on 03-19-2020 at 11:31:56.458 -07:00. A section titled 'Job execution Succeeded' shows 'Spark 2 executors 8 cores' with three jobs listed:

ID	DESCRIPTION	STATUS	STAGES	TASKS	SUBMISSION TIME	DURATION
▶ Job 0	load at NativeMethodAccessImpl.java:0	✓ Succeeded	1/1	<div style="width: 100%; background-color: green;"></div>	3/19/2020, 11:31:35 AM	6s
▶ Job 1	showString at NativeMethodAccessImpl.java:0	✓ Succeeded	1/1	<div style="width: 100%; background-color: green;"></div>	3/19/2020, 11:31:43 AM	1s
▶ Job 2	showString at NativeMethodAccessImpl.java:0	✓ Succeeded	1/1	<div style="width: 100%; background-color: green;"></div>	3/19/2020, 11:31:45 AM	9s

Below the table, the output of the 'show(100)' command is displayed as a table schema and several rows of data. The schema includes columns like vendorID, tpepPickupDateTime, tpepDropoffDateTime, passengerCount, tripDistance, puLocationId, doLocationId, startLon, startLat, endLon, endLat, rateCodeId, storeAndFwdFlag, paymentType, fareAmount, extra, mtaTax, improvementSurcharge, tipAmount, tollsAmount, and totalAmount.

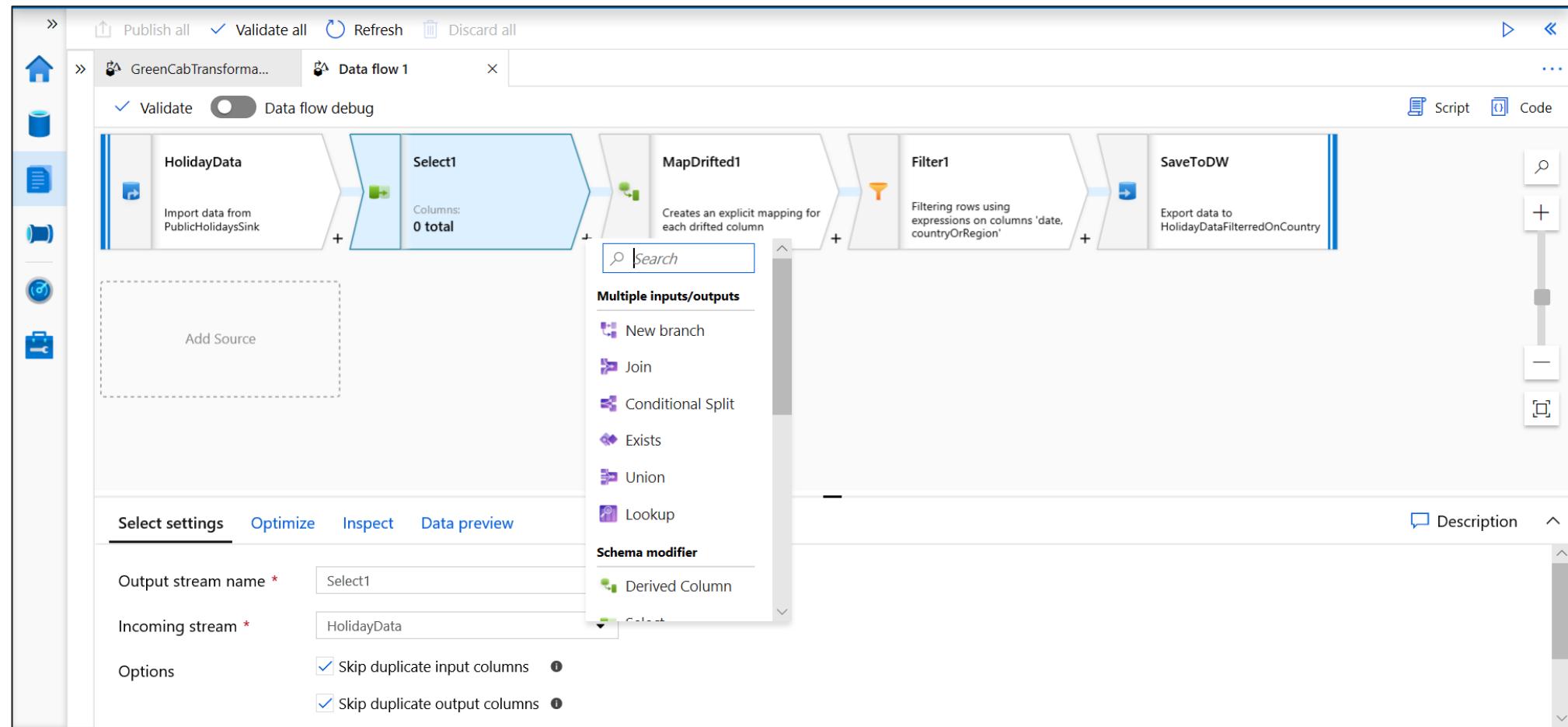
vendorID	tpepPickupDateTime	tpepDropoffDateTime	passengerCount	tripDistance	puLocationId	doLocationId	startLon	startLat	endLon	endLat	rateCodeId	storeAndFwdFlag	paymentType	fareAmount	extra	mtaTax	improvementSurcharge	tipAmount	tollsAmount	totalAmount
40.760237	2009-04-30 23:59:52	2009-05-01 00:11:14	0	1.9	null		-73.984708				1	Credit	8.5	0.0	null			1.8		
0.0	10.3										0	Credit	3.4	9.7	0.0	null		-73.956527	2.55	
40.771307	2009-05-07 01:03:26	2009-05-07 01:14:11	0	2.2	null		-74.009102				1	Credit	2.2	0.0	null					
0.0	12.25										0	Credit	0.0	0.0	null					
	CMT	2009-04-30 23:50:42	2009-05-01 00:06:43																	

At the bottom, there are buttons for 'Ready' (with a checkmark), '(Stop session)', and 'Configure session'.

Develop Hub - Data Flows

Data flows are a visual way of specifying how to transform data.

Provides a code-free experience.



Develop Hub – Power BI

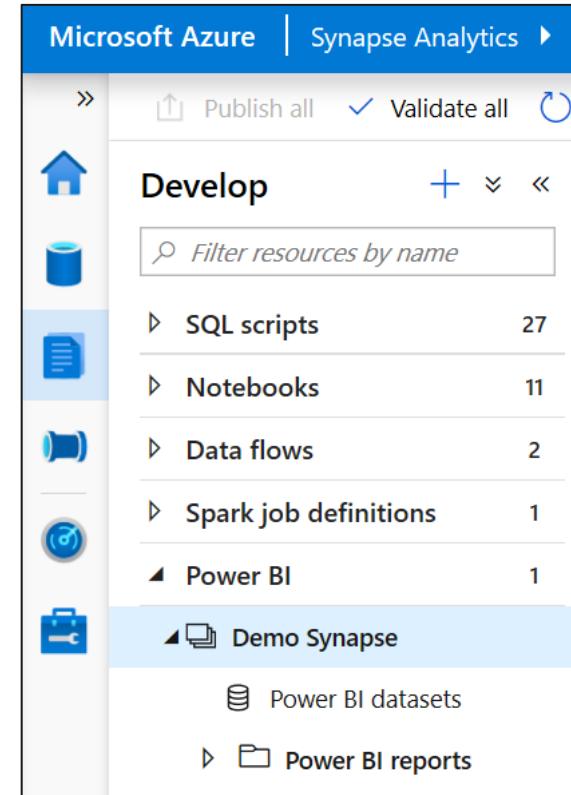
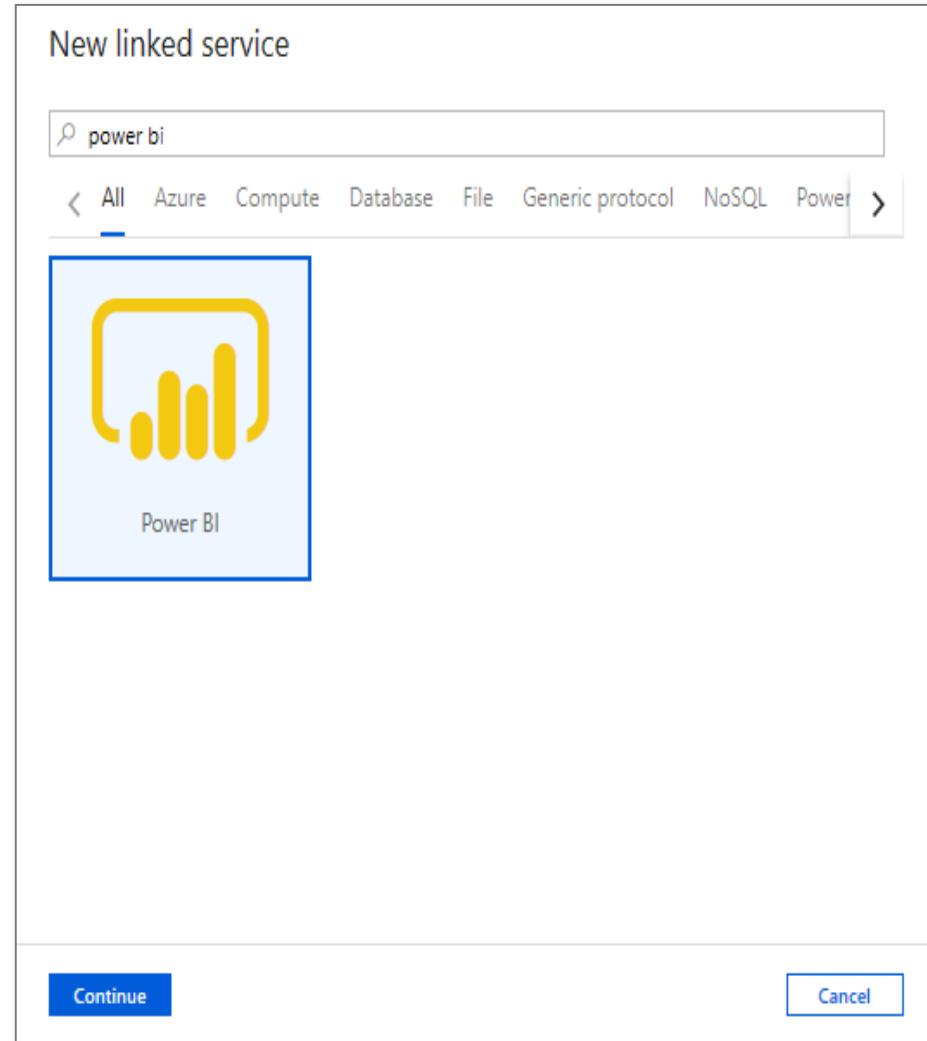
Overview

Create Power BI reports in the workspace

Provides access to published reports in the workspace

Update reports real time from Synapse workspace to get it reflected on Power BI service

Visually explore and analyze data



Develop Hub – Power BI

View published reports in Power BI workspace

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface for a "gamingtelemetry" report. The left sidebar displays navigation options like "Develop", "Power BI datasets", "Power BI reports", and "Report". The main area features a "GAME STUDIO" report with a "What If..." analysis section, a "Total Users vs 'What If' Analysis" table, and a "What If" Analysis Forecast chart. The right sidebar contains sections for "VISUALIZATIONS" and "FIELDS", showing various data fields and forecast-related options.

Report Title: GAME STUDIO

What If...
We increase free game addons by: 1

Total Users vs "What If" Analysis

Region	Users	Forecast	Extra Users
APAC	1,268.5K	1,273.7K	45.4K
18-22	96.8K	97.7K	4.0K
22-26	436.0K	435.5K	13.4K
26-30	462.9K	464.0K	15.6K
30-34	75.0K	76.3K	3.4K
34-40	24.0K	24.2K	1.1K
41-60	27.1K	27.5K	1.3K
>60	146.7K	148.5K	6.7K
EMEA	844.9K	846.5K	30.4K
18-22	66.8K	67.5K	2.7K
22-26	291.8K	290.7K	9.1K
26-30	306.9K	307.1K	10.4K
30-34	50.4K	50.9K	2.3K
34-40	16.3K	16.4K	0.7K
41-60	18.5K	18.7K	0.9K
Total	7,346.3K	7,361.7K	252.8K

"What If" Analysis Forecast

Users (Forecast) **7,361,707**
Last month
Extra Users **252.8K**
+3.4% Users Increase

Forecast Date: Aug 2019 - Nov 2019

Visualizations and Fields

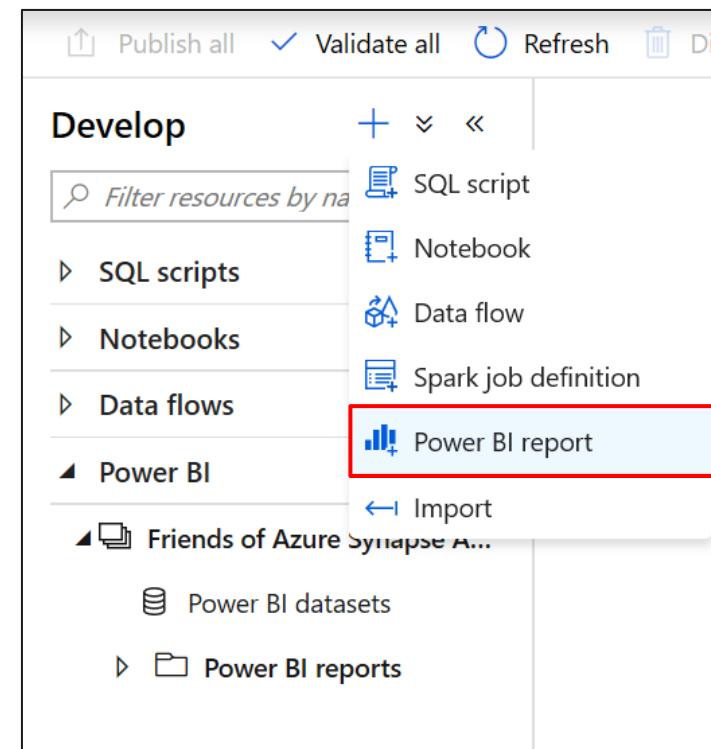
- Visualizations: agegroup, forecast, historical, platform, predictions, realtime, regions, scenario, weekdays.
- Fields: Add data fields here, Drill Through, Cross-report, Off, Keep all filters, On, Add drill-through fields here.

Develop Hub – Power BI

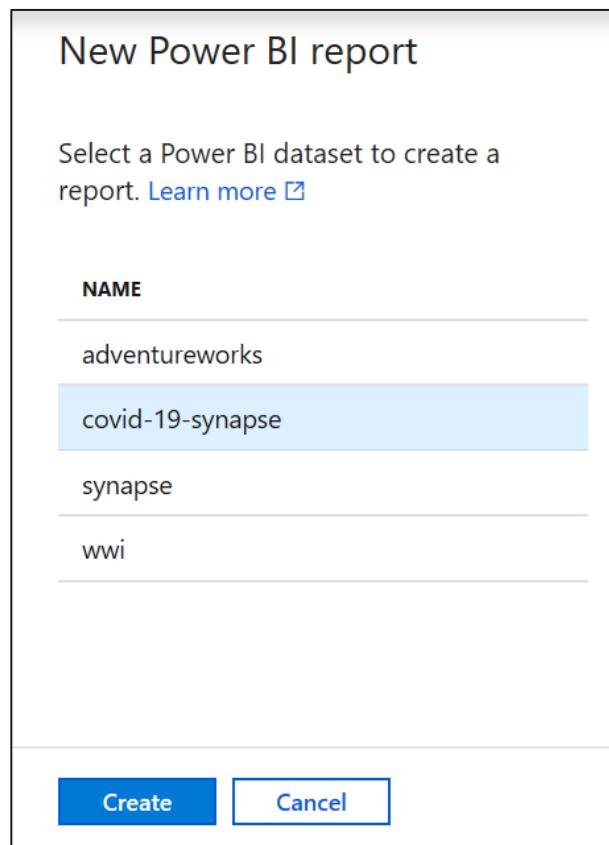
Create new reports from existing published Power BI datasets

Create new Power BI datasets

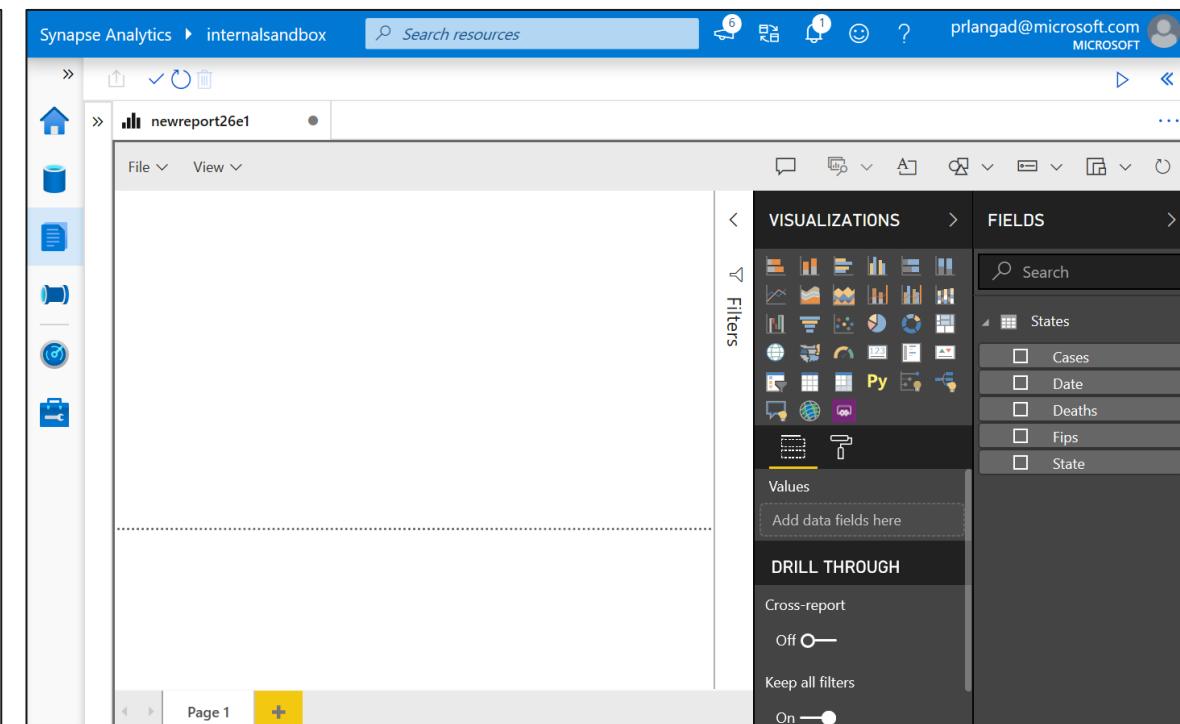
1



2



3



Develop Hub – Power BI

Edit reports in Synapse workspace

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar displays the 'Develop' hub with options for publishing, validating, and refreshing reports. The main area shows a Power BI report titled 'GAME STUDIO'. The report features a header with a game controller image and a dropdown menu set to 'Console'. It includes a 'What If...' section with a slider for adding free game addons, showing a forecast of 7,361,707 users (7,346,291 last month) and an increase of 252.8K (+3.4% Users Increase). Below this is a table titled 'Total Users vs "What If" Analysis' comparing actual users and forecasts across regions and age groups. To the right is a line chart titled '"What If" Analysis Forecast' showing user growth from August 2019 to November 2019. The bottom navigation bar includes tabs for 'Historical', 'Forecast', 'Predictions', and a plus sign. The right sidebar contains sections for 'VISUALIZATIONS' and 'FIELDS', with various icons and filters listed.

GAME STUDIO

What If...
We increase free game addons by:

Region	Users	Forecast	Extra Users
APAC	1,268.5K	1,273.7K	45.4K
18-22	96.8K	97.7K	4.0K
22-26	436.0K	435.5K	13.4K
26-30	462.9K	464.0K	15.6K
30-34	75.0K	76.3K	3.4K
34-40	24.0K	24.2K	1.1K
41-60	27.1K	27.5K	1.3K
>60	146.7K	148.5K	6.7K
EMEA	844.9K	846.5K	30.4K
18-22	66.8K	67.5K	2.7K
22-26	291.8K	290.7K	9.1K
26-30	306.9K	307.1K	10.4K
30-34	50.4K	50.9K	2.3K
34-40	16.3K	16.4K	0.7K
41-60	18.5K	18.7K	0.9K
Total	7,346.3K	7,361.7K	252.8K

"What If" Analysis Forecast

Users (Forecast) **7,361,707**
7,346,291 Last month

Extra Users **252.8K**
+3.4% Users Increase

Total Users **24.5M**

Historical **Forecast** **Predictions** **+**

Develop Hub – Power BI

Publish edited reports in Synapse workspace to Power BI workspace

A screenshot of the Microsoft Azure Synapse Analytics Develop Hub interface. The left sidebar shows a navigation tree with 'Develop' selected, under which 'Power BI' is expanded to show 'gaming-telemetry' and 'Power BI reports'. A red arrow points from the text 'Publish changes by simple save report in workspace' to the 'Save this report' button in the top right corner of the main report area. The main area displays a Power BI report titled 'GAME STUDIO' with a dashboard featuring a game controller on a console, a chart showing 'Total Users 24.5M', and a 'What If...' analysis section. The right side of the screen shows the 'VISUALIZATIONS' and 'FIELDS' panes.

Publish changes by simple save report in workspace

Save this report

Region	Users	Forecast	Extra Users
APAC	1,268.5K	1,319.0K	90.7K
18-22	96.8K	101.8K	8.1K
22-26	436.0K	448.9K	26.7K
26-30	462.9K	479.5K	31.1K
30-34	75.0K	79.7K	6.7K
34-40	24.0K	25.3K	2.2K
41-60	27.1K	28.8K	2.5K
>60	146.7K	155.1K	13.3K
EMEA	844.9K	876.7K	60.7K
18-22	66.8K	70.2K	5.5K
22-26	291.8K	299.6K	18.0K
26-30	306.9K	317.5K	20.9K
30-34	50.4K	53.2K	4.5K
34-40	16.3K	17.2K	1.5K
41-60	18.5K	19.6K	1.8K
Total	7,346.3K	7,613.6K	504.8K

What If...
We increase free game addons by:
2

Users (Forecast)
7,613,619
7,346,291
Last month

Extra Users
504.8K
+6.9%
Users Increase

Total Users vs "What If" Analysis

"What If" Analysis Forecast

Historical Forecast Predictions

Visualizations Fields

Values

Add data fields here

DRILL THROUGH

Cross-report Off —

Keep all filters On —

Add drill-through fields here

Develop – CI/CD

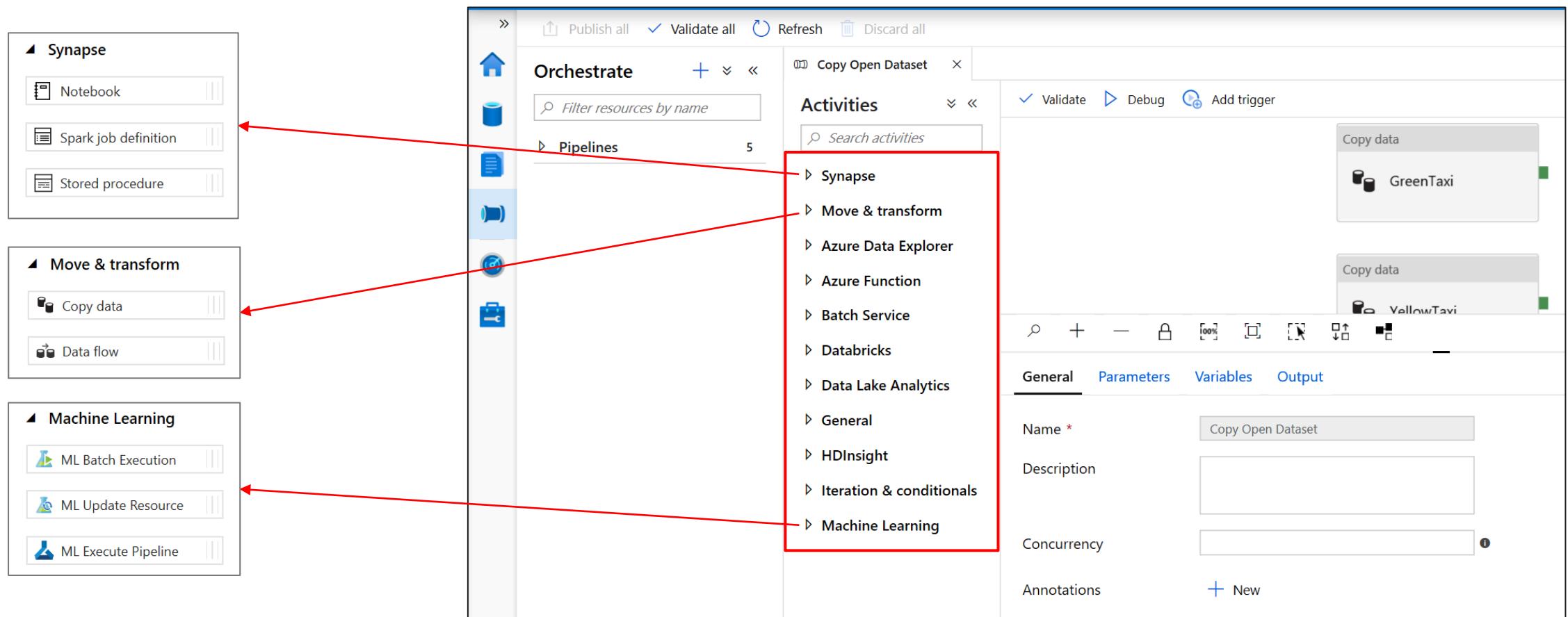
Commit artifacts to source-controlled repository and operationalize release pipelines with Synapse deployment task

The screenshot illustrates the integration of GitHub and Azure Synapse for CI/CD. On the left, a GitHub browser interface shows a list of files and commits for the repository `https://github.com/SynapseTestDemo/synapsetestdemo-ws-01/tree/dev`. The commits include various configurations for linked services and integration runtimes. On the right, an Azure Synapse pipeline interface titled "Synapse deployment v2 > Release-13" is displayed. It shows a "Release" section with a "Manually triggered" step by Priyanka Langade on 11/16/2020 at 11:02 PM, and an "Artifacts" section listing an artifact named "azuresynapse" with ID "0c8b1a872" from branch "retail-12". The "Stages" section shows a single stage named "Load to Prod" that has succeeded on 11/17/2020 at 4:54 PM.

Integrate Hub

It provides ability to create pipelines to ingest, transform and load data with 90+ inbuilt connectors.

Offers a wide range of activities that a pipeline can perform.



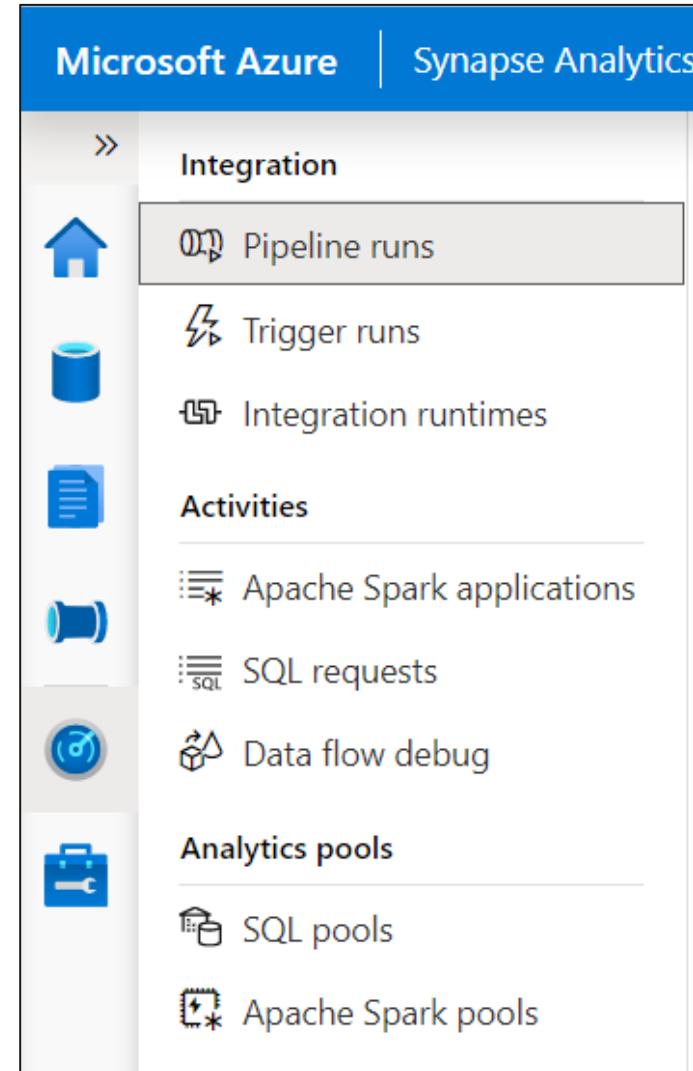
Monitor Hub

Overview

This feature provides single pane of glass to monitor orchestration, activities for Apache Spark Application and SQL requests.

Benefits

Offers additional filters to monitor specific activities or orchestration



Manage Hub

Overview

This feature provides ability to manage Analytics pools, Linked Services, Integration, Security and Source Control.

The screenshot shows the Microsoft Azure Synapse Analytics Manage Hub interface. The left sidebar has a 'Manage' icon selected. The main area shows the 'SQL pools' section, which lists six items: 'Built-in' (Serverless, Online, Auto), 'newpool' (Dedicated, Paused, DW200c), 'NYCTaxi_Pool' (Dedicated, Online, DW100c), 'Predict_Pool' (Dedicated, Online, DW1000c), 'Streaming_Pool' (Dedicated, Paused, DW2000c), and 'WWI_Pool' (Dedicated, Online, DW100c). A 'System assigned managed identity' toggle switch is visible.

Name	Type	Status	Size
Built-in	Serverless	Online	Auto
newpool	Dedicated	Paused	DW200c
NYCTaxi_Pool	Dedicated	Online	DW100c
Predict_Pool	Dedicated	Online	DW1000c
Streaming_Pool	Dedicated	Paused	DW2000c
WWI_Pool	Dedicated	Online	DW100c

Manage – Linked services

Overview

It defines the connection information needed to connect to external resources.

Benefits

Offers pre-build 90+ connectors

Easy cross platform data migration

Represents data store or compute resources

The screenshot shows the Microsoft Azure Synapse Analytics portal interface. The left sidebar contains navigation links: Analytics pools, SQL pools, Apache Spark pools, External connections, **Linked services** (which is highlighted with a red box), Integration, Triggers, Integration runtimes, Security, Access control, Credentials, Managed private endpoints, Source control, and Git configuration. The main area is titled "Linked services" and contains a brief description: "Linked services are much like connection strings, which define the connection to external resources." Below this is a "New" button, also highlighted with a red box. A modal dialog titled "New linked service" is open, displaying a grid of connector icons and names. The "Power BI" icon is selected and highlighted with a blue border. Other visible connectors include PayPal (Preview), Phoenix, PostgreSQL, Presto (Preview), QuickBooks (Preview), REST, SAP BW Open Hub, SAP BW via MDX, SAP Cloud For Customer, SAP ECC, SAP HANA, and SAP ECC.

Manage – Triggers

Overview

It defines a unit of processing that determines when a pipeline execution needs to be kicked off.

Benefits

Create and manage

- Schedule trigger
- Tumbling window trigger
- Event trigger

Control pipeline execution

The screenshot shows the Azure portal interface for managing triggers. On the left, there's a sidebar with various options like Analytics pools, SQL pools, Apache Spark pools, External connections, Linked services, Integration (with Triggers highlighted), Integration runtimes, Security, and Access control. The main area is titled 'Triggers' with the sub-instruction 'To execute a pipeline set the trigger to be kicked off.' Below this is a 'New' button, which is highlighted with a red box and a red arrow pointing from the 'Triggers' section towards it. To the right of the 'New' button is a 'Filter by keyword' input field. At the bottom, there are sorting options: Name ↑, Type ↑, Status ↑, and Pipelines ↑. A large 'New trigger' dialog box is open on the right side of the screen, containing fields for Name (Trigger 2), Description, Type (Schedule selected), Start Date (UTC) (10/29/2019 9:46 PM), Recurrence (Every 1 Minute(s)), End (No End selected), Annotations (New), and Activated (No selected). There are 'OK' and 'Cancel' buttons at the bottom of the dialog.

Manage – Access Control

Overview

It provides access control management to workspace resources and artifacts for admins

Benefits

Share workspace with the team

Increases productivity

Assign granular level permissions

Manage permissions on Spark pools,
Integration Runtimes, Linked services,
Credentials

The screenshot illustrates the Microsoft Azure Synapse Analytics workspace access control interface. The main view shows a table of role assignments:

NAME	TYPE	ROLE
soft.com	Individual	Workspace admin

A red box highlights the '+ Add' button in the top right corner of the main interface. Red arrows point from the 'Add' button to the 'Scope' field in the first 'Add role assignment' dialog and from the 'Scope' field to the 'Select user' field in the second 'Add role assignment' dialog.

Add role assignment

Grant others access to this workspace by assigning roles to users, groups, and/or service principals. [Learn more](#)

Scope * Workspace Workspace item

Role * Select a role
Filter...
Synapse Administrator
Synapse SQL Administrator
Synapse Apache Spark Administrator
Synapse Contributor (preview)
Synapse Artifact Publisher (preview)
Synapse Artifact User (preview)
Synapse Compute Operator (preview)
Synapse Credential User (preview)

Add role assignment

Grant others access to this workspace by assigning roles to users, groups, and/or service principals. [Learn more](#)

Scope * Workspace Workspace item

Item type * Credentials
WorkspaceSystemIdentity

Item * WorkspaceSystemIdentity

Role * Synapse Administrator
Select user *

Selected user(s), group(s), or service principal(s)
No users, groups, or apps selected.

Manage – Source Control

Overview

Associate Synapse workspace with a Git repository, Azure DevOps, or GitHub

Configure a repository

SynapseTestDemo

Specify the settings that you want to use when connecting to your repository.

Enter manually Use repository link

Git repository name *
synapsetestdemo-ws-01

Collaboration branch * dev

Publish branch * main

Root folder * /

Import existing resource Import existing resources to repository

Import resource into this branch

Apply **Back** **Cancel**

Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics

Validate all Commit all Publish

Configure a repository

Connect your workspace with your Git repository just within please view document here.

Setting **Disconnect**

Repository type GitHub

GitHub account SynapseTestDemo

Git repository name synapsetestdemo-ws-01

Collaboration branch dev

Publish branch main

Root folder /

Integration runtimes

Security

Access control

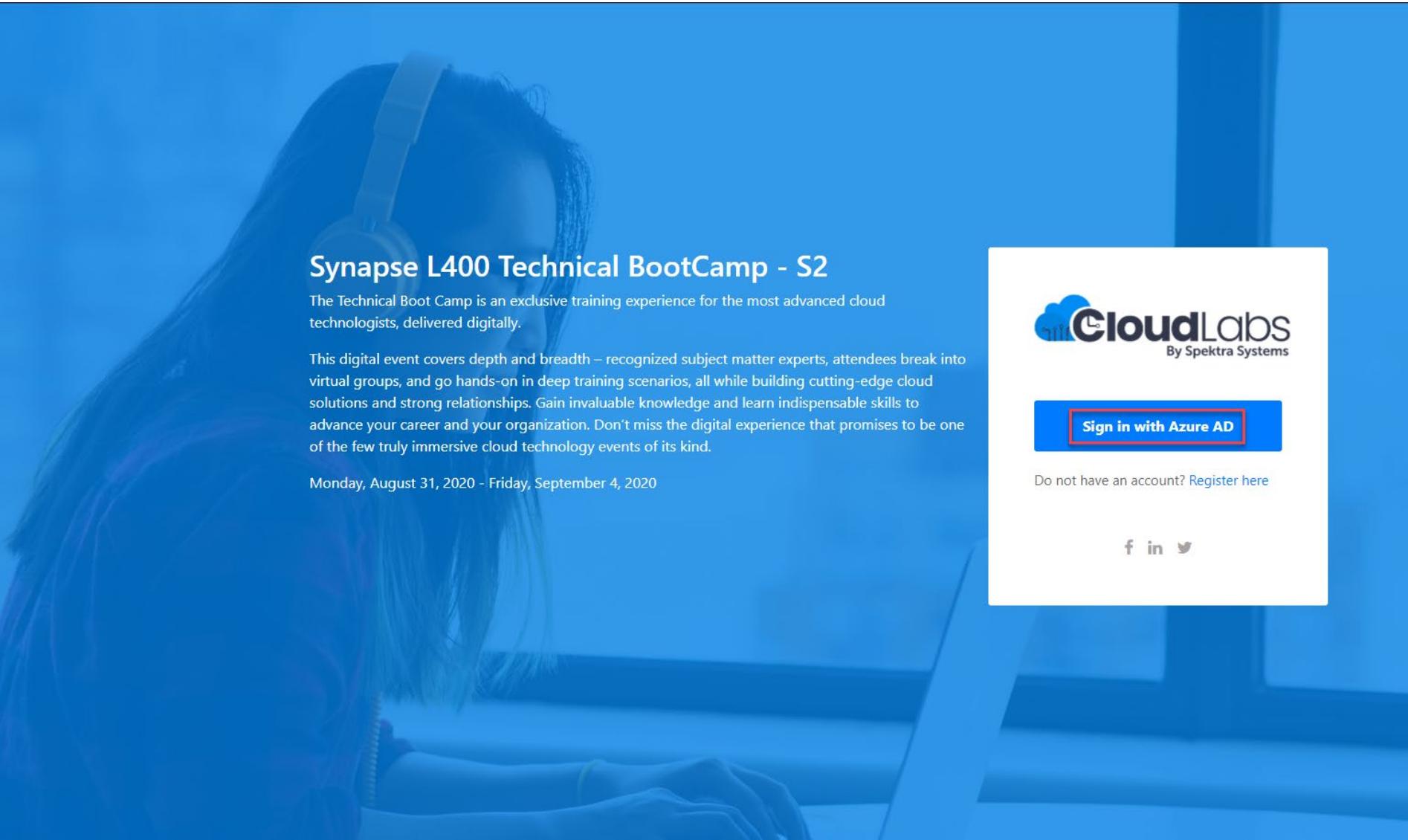
Credentials

Managed private endpoints

Source control

Git configuration

Login to the portal

A large, semi-transparent blue rectangular overlay covers the left side of the page, featuring a blurred background image of a person's head and shoulders wearing over-ear headphones. On the right side, there is a white rectangular login form.

Synapse L400 Technical BootCamp - S2

The Technical Boot Camp is an exclusive training experience for the most advanced cloud technologists, delivered digitally.

This digital event covers depth and breadth – recognized subject matter experts, attendees break into virtual groups, and go hands-on in deep training scenarios, all while building cutting-edge cloud solutions and strong relationships. Gain invaluable knowledge and learn indispensable skills to advance your career and your organization. Don't miss the digital experience that promises to be one of the few truly immersive cloud technology events of its kind.

Monday, August 31, 2020 - Friday, September 4, 2020

CloudLabs
By Spektra Systems

[Sign in with Azure AD](#)

Do not have an account? [Register here](#)

[f](#) [in](#) [t](#)

Sessions for Bootcamp

The screenshot shows the CloudLabs platform interface. At the top, there is a navigation bar with links: My Sessions (highlighted with a red dashed border), My Calendar, My Team, Users, and Teams. On the far right, it says "Hi" followed by a user icon. Below the navigation bar is a banner featuring a calendar on a tablet and a keyboard, with the text "My Sessions" and "Synapse L400 Technical BootCamp - S2".

Below the banner, there are several filter buttons: "All Days" (highlighted in blue), "Mon 31", "Tue 1", "Wed 2", "Thu 3", and a "Add to Calendar" button. There is also a search bar with the placeholder "Search" and a magnifying glass icon.

The main content area displays a single session card:

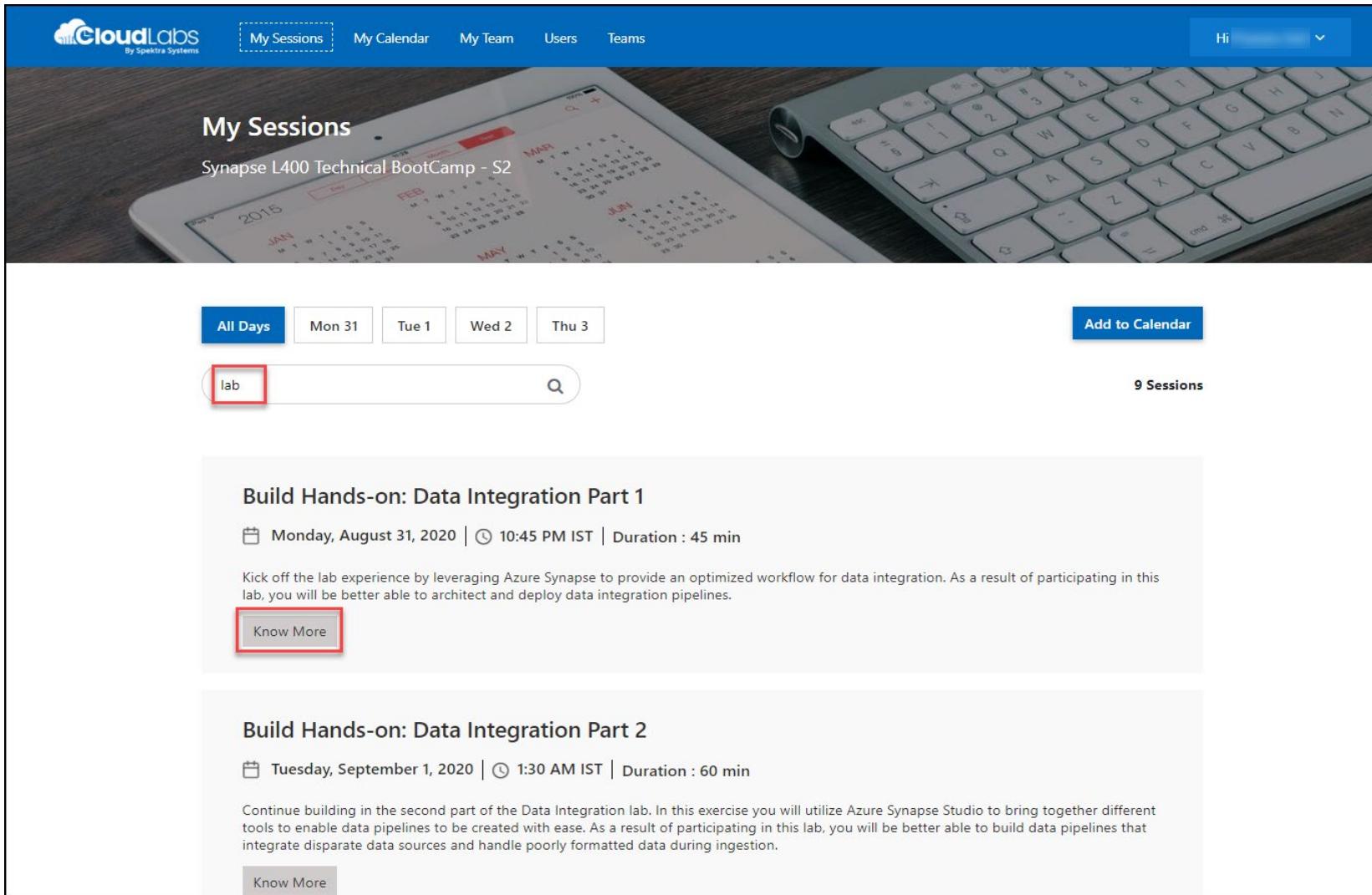
Welcome
Speaker: Leanne Gallagher
Monday, August 31, 2020 | 7:30 PM IST | Duration : 5 min

Welcome to the Azure Synapse Technical Boot Camp! Discover what we will be learning this week, learn about the resources available to you and where to find them, and connect with your peers.

[Know More](#)

At the bottom right of the main content area, it says "37 Sessions".

Search for lab



CloudLabs
By Spektra Systems

My Sessions My Calendar My Team Users Teams Hi

My Sessions

Synapse L400 Technical BootCamp - S2

All Days Mon 31 Tue 1 Wed 2 Thu 3 Add to Calendar

lab

9 Sessions

Build Hands-on: Data Integration Part 1
Monday, August 31, 2020 | 10:45 PM IST | Duration : 45 min
Kick off the lab experience by leveraging Azure Synapse to provide an optimized workflow for data integration. As a result of participating in this lab, you will be better able to architect and deploy data integration pipelines.
[Know More](#)

Build Hands-on: Data Integration Part 2
Tuesday, September 1, 2020 | 1:30 AM IST | Duration : 60 min
Continue building in the second part of the Data Integration lab. In this exercise you will utilize Azure Synapse Studio to bring together different tools to enable data pipelines to be created with ease. As a result of participating in this lab, you will be better able to build data pipelines that integrate disparate data sources and handle poorly formatted data during ingestion.
[Know More](#)

- Search for lab
- Click on Know More

Welcome Back!



7:00-7:05	Welcome	
7:05-7:15	Keynote	
7:15-7:45	Demo Walkthrough	
7:45-8:00	Break	
8:00-9:00	Data Loading & Data Lake Organization	
9:00-10:00	Activity: Data Lake Design & Security Considerations	
10:00-10:15	Break	Table Group Call
10:15-11:00	Build Hands-on: Data Integration Part 1	
11:00-12:00	Break	Main Call
12:00-12:30	Data Transformations	
12:30-13:00	Activity: Data Engineering Discussion	Table Group Call
13:00-14:00	Build Hands-on: Data Integration Part 2	
14:00-14:15	Closing	Main Call

- Presentation/
Whole Group
- Lab
- Activity/ Discussion/
Group Work
- Announcements

Data Loading & Data Lake Organization



Agenda

1 Integration (Orchestration)

Synapse pipelines

2 Ingest files to tables

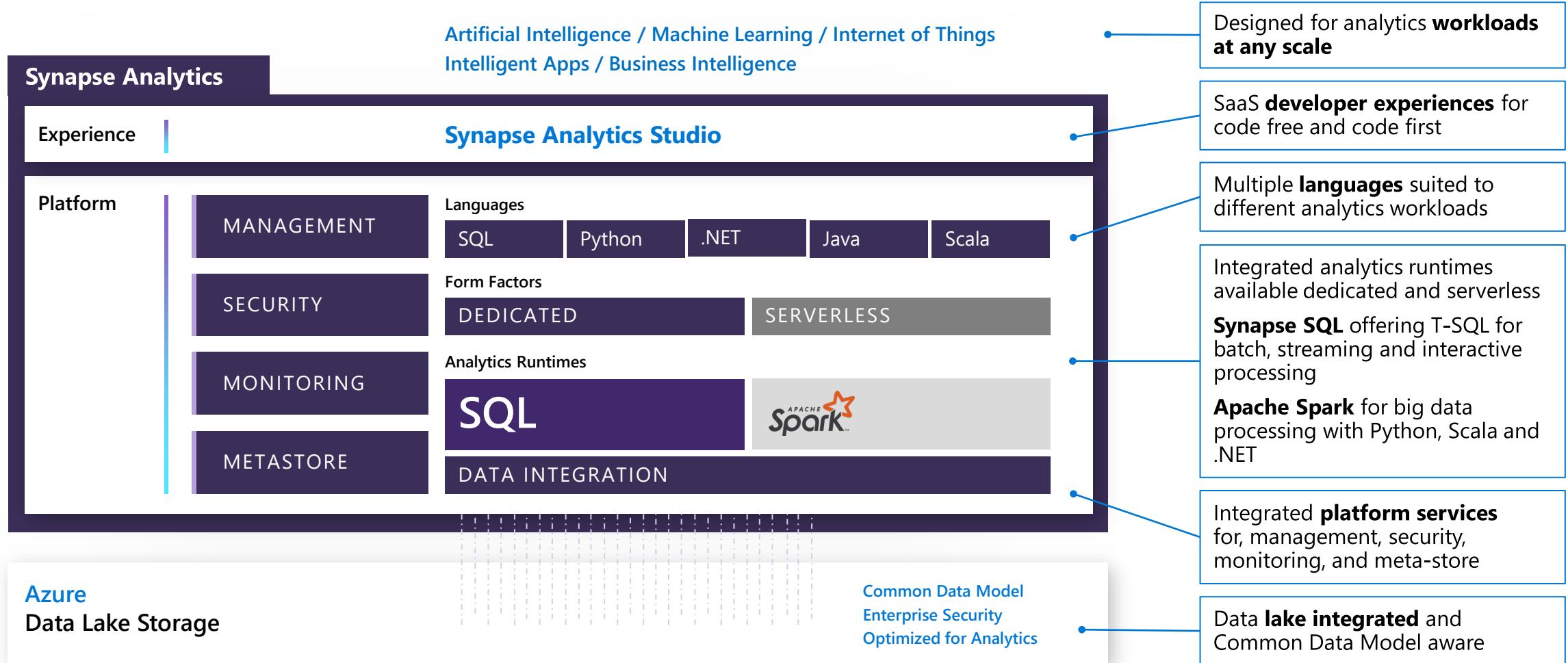
Copy versus CTAS

3 Best practices

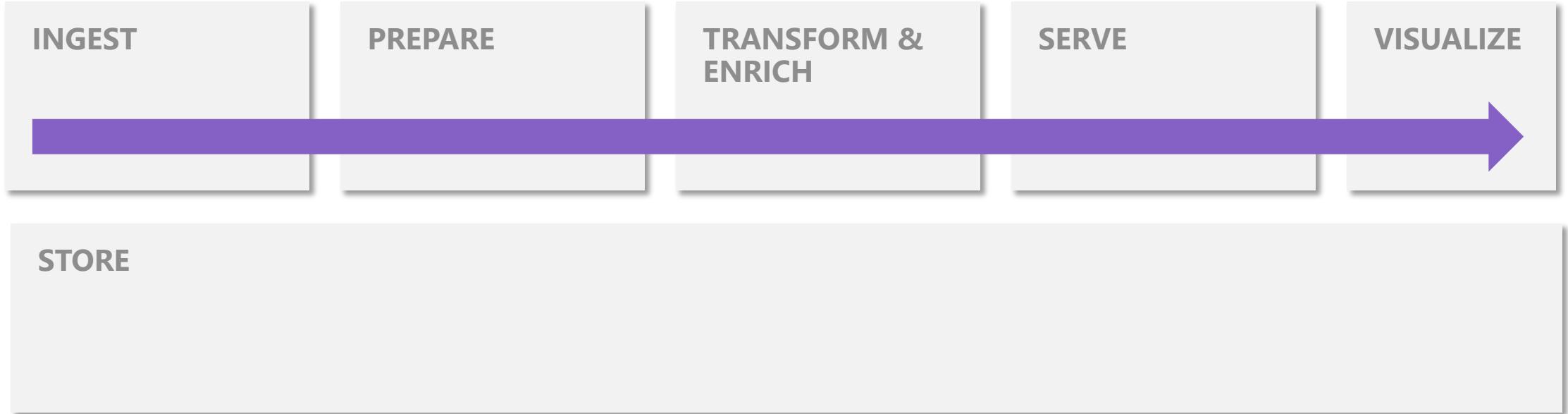
Various ingest and storage best practices

Azure Synapse Analytics

Limitless analytics service with unmatched time to insight



Modern Data Warehouse



Ingest - Integration with Pipelines

Linked services

Overview

Linked services define the connection information needed to connect to external resources.

Benefits

- Offers pre-build 90+ connectors
- Easy cross platform data migration
- Represents data store or compute resources

The screenshot shows the Microsoft Azure Synapse Analytics interface. On the left, there's a sidebar with icons for External connections, Linked services (which is selected and highlighted with a red box), Orchestration, Triggers, Integration runtimes, Security, and Access control. The main area has a title "Linked services" and a sub-instruction: "Linked services are much like connection strings, which define the connection information needed for Arcadia to connect to external resources." A red box highlights the "+ New" button. Below it is a table with columns NAME, TYPE, and ANNOTATIONS. The table lists several entries: ADLSG2OpenDataSetSink (Azure Data Lake Storage Gen2), AzureBlobStorage1 (Azure Blob Storage), AzureDataLakeStorage1 (Azure Data Lake Storage Gen2), AzureDataLakeStorage2Source (Azure Data Lake Storage Gen2), AzureOpenDataset, AzureOpenDataSet2, and AzureSqlDW1. To the right of the table is a "Search to filter items..." input field and a "New linked service" modal window. The modal contains a grid of connector icons and names: PayPal (Preview), Phoenix, PostgreSQL; Power BI (highlighted with a red box); Presto (Preview), QuickBooks (Preview); REST, SAP BW Open Hub, SAP BW via MDX; SAP Cloud For Customer, SAP ECC; SAP HANA. At the bottom of the modal are "Continue" and "Cancel" buttons. A red arrow points from the "+ New" button in the main area down to the "New linked service" modal.

90+ Connectors out of the box

Datasets

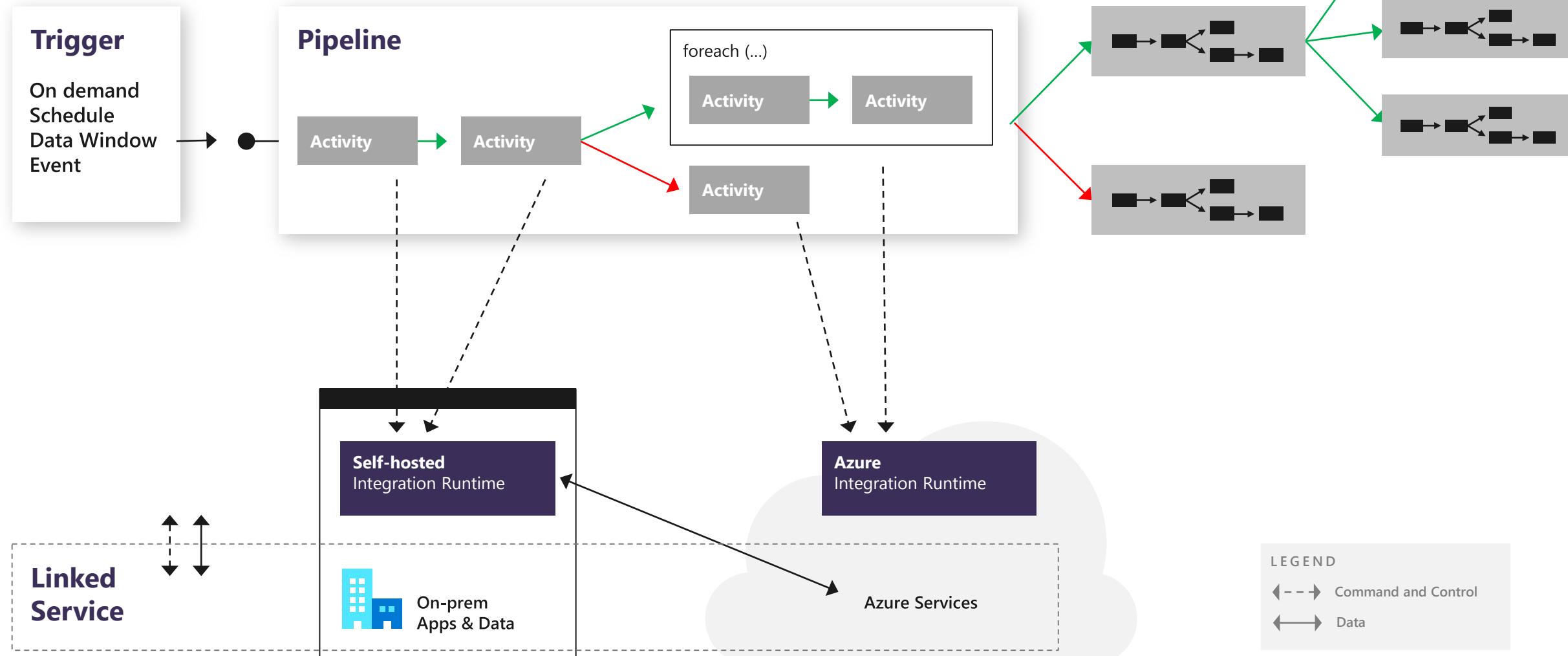
Orchestration datasets describe data that is persisted.

Once a dataset is defined, it can be used in pipelines and sources of data or as sinks of data.

The screenshot shows the Azure Data Studio interface with the following details:

- Left Panel (Data Explorer):** Shows a tree view of resources. A red arrow points from the "NYCTaxiParquet" node under the "Datasets" category to the main workspace.
- Main Workspace (NYCTaxiParquet Dataset Definition):**
 - Title Bar:** NYCTaxiParquet X
 - Icon:** Parquet icon.
 - Name:** NYCTaxiParquet
 - General Tab:** Selected tab. It includes:
 - Linked service:** Lake_ArcadiaLake
 - File path:** data / nyctaxi / File
 - Compression type:** snappy
 - Buttons:** Test connection, Open, New, Browse, Preview data.
 - Connection Tab:** Shows the current linked service selection.
 - Schema Tab:** Not visible in the screenshot.
 - Parameters Tab:** Not visible in the screenshot.

Components of Orchestration



Synapse Pipelines shares codebase with Azure Data Factory

Pipelines

Create pipelines to ingest, transform and load data with 90+ inbuilt connectors.

Offers a wide range of activities that a pipeline can perform.

The screenshot shows the Azure Data Factory Orchestrate interface for creating a pipeline. A red box highlights the 'Activities' pane on the right, which lists various types of activities including Move & transform, Machine Learning, and Synapse. Three smaller callout boxes point from the left side of the slide to specific activity categories in this pane:

- Move & transform**: Points to the 'Move & transform' section in the Activities pane, which includes 'Copy data' and 'Data flow' options.
- Machine Learning**: Points to the 'Machine Learning' section in the Activities pane, which includes 'ML Batch Execution', 'ML Update Resource', and 'ML Execute Pipeline' options.
- Synapse**: Points to the 'Synapse' section in the Activities pane, which includes 'Notebook', 'Spark job definition', and 'Stored procedure' options.

The main workspace shows Pipeline 2 configuration. It includes a 'Stored procedure' activity named 'sql1_dbo_StorePredictions' connected to a 'Notebook' activity named 'BOOT_Basic_spark'. The pipeline has one trigger and is set to validate.

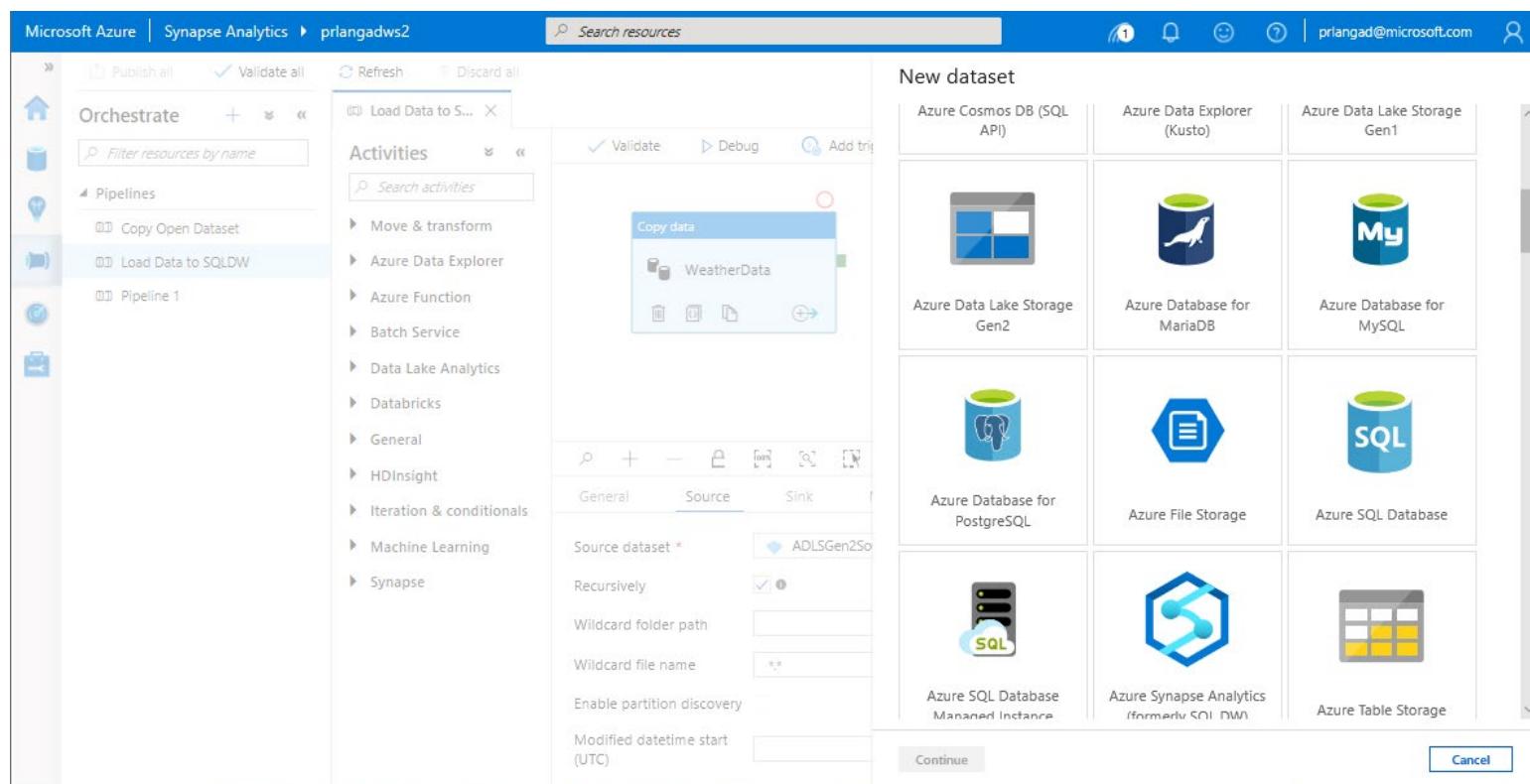
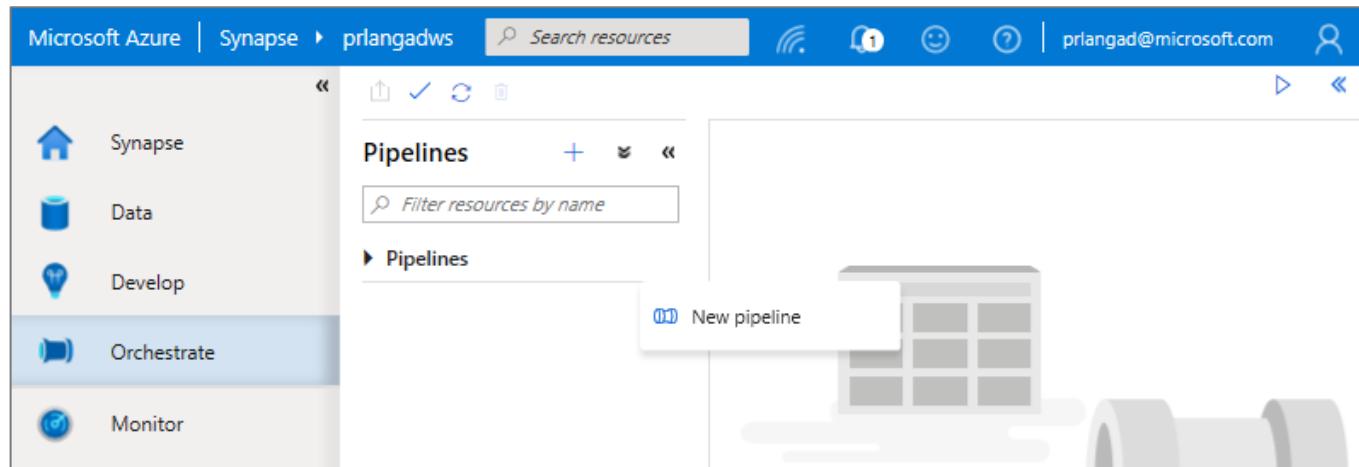
Pipelines

Overview

- Provide ability to load data from storage account to desired linked service.
- Load data by manual execution of pipeline or by orchestration.

Benefits

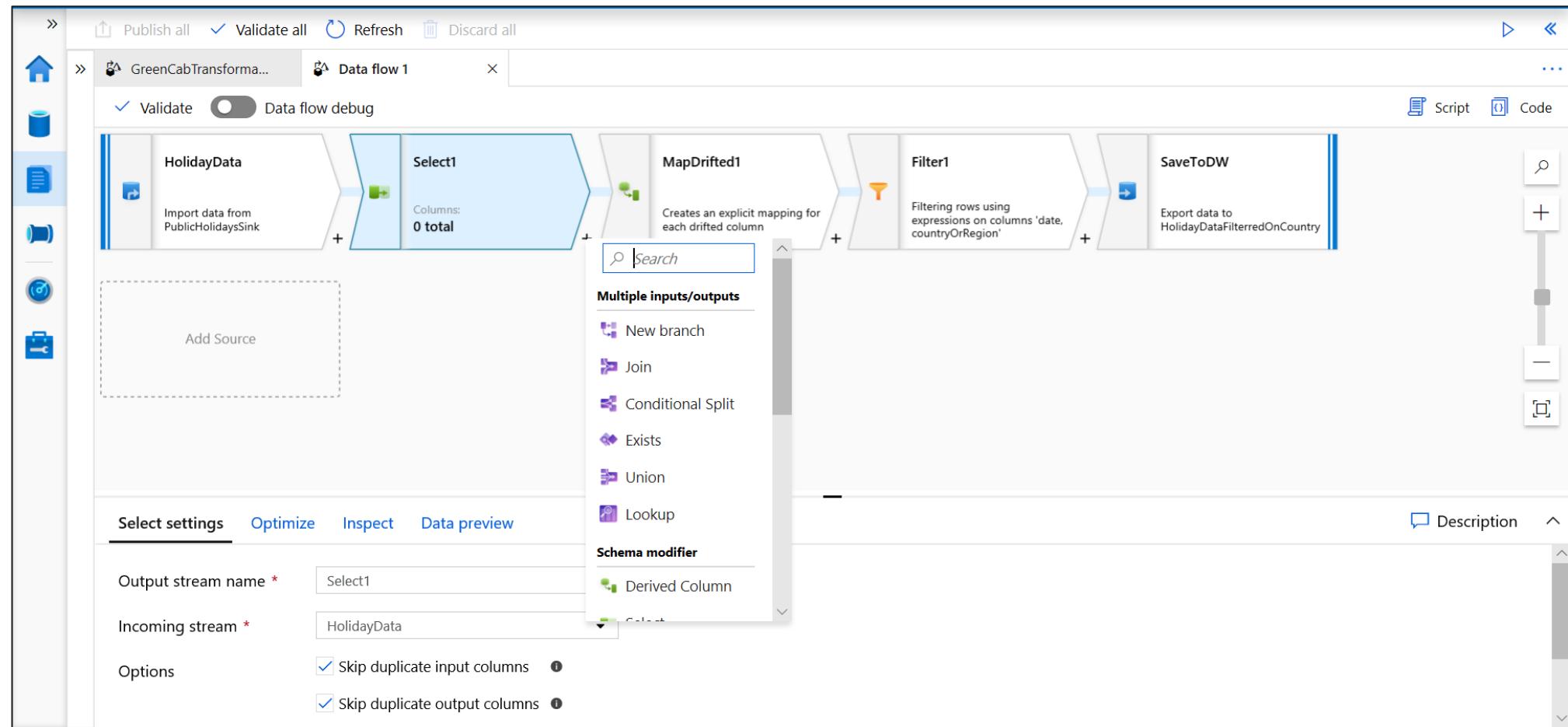
- Supports common loading patterns.
- Fully parallel loading into data lake or SQL tables.
- Graphical development experience.



Develop Hub - Data Flows

Data flows are a visual way of specifying how to transform data.

Provides a code-free experience.



Dataflow Capabilities



Handle upserts, updates, deletes on sql sinks



Add new partition methods



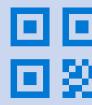
Add schema drift support



Add file handling (move files after read, write files to file names described in rows etc)



New inventory of functions (for e.g Hash functions for row comparison)



Commonly used ETL patterns(Sequence generator/Lookup transformation/SCD...)



Data lineage – Capturing sink column lineage & impact analysis(invaluable if this is for enterprise deployment)



Implement commonly used ETL patterns as templates(SCD Type1, Type2, Data Vault)

Triggers

Overview

Triggers represent a unit of processing that determines when a pipeline execution needs to be kicked off.

Data Integration offers 3 trigger types as –

1. Schedule – gets fired at a schedule with information of start date, recurrence, end date
2. Event – gets fired on specified Storage event
3. Tumbling window – gets fired at a periodic time interval from a specified start date, while retaining state

The screenshot shows the Microsoft Data Integration interface. On the left, there is a sidebar with various options: Analytics pools, SQL pools, Apache Spark pools, External connections, Linked services, Integration, Triggers (which is highlighted with a red box), Integration runtimes, Security, and Access control. The main area is titled 'Triggers' and contains the following text: 'To execute a pipeline set the trigger to be kicked off.' Below this is a 'New' button, a 'Filter by keyword' input field, and an 'Annotations : Any' button. At the bottom, there are sorting options: Name ↑↓, Type ↑↓, Status ↑↓, and Pipelines ↑↓. A red arrow points from the 'Triggers' sidebar to a 'New trigger' dialog box on the right. The dialog box has the following fields:

- Name *: Trigger 2
- Description: (empty)
- Type *:
 - Schedule (radio button selected)
 - Tumbling window
 - Event
- Start Date (UTC) *: 10/29/2019 9:46 PM
- Recurrence *:
 - Every 1 Minute(s)
- End *:
 - No End (radio button selected)
 - On Date
- Annotations: + New
- Activated *:
 - Yes
 - No (radio button selected)

At the bottom right of the dialog box are 'OK' and 'Cancel' buttons.

It also provides ability to monitor pipeline runs and control trigger execution.

Manage – Integration runtimes

Overview

Integration runtimes are the compute infrastructure used by Pipelines to provide the data integration capabilities across different network environments. An integration runtime provides the bridge between the activity and linked services.

Benefits

Offers Azure Integration Runtime or Self-Hosted Integration Runtime

Azure Integration Runtime – provides fully managed, serverless compute in Azure

Self-Hosted Integration Runtime – use compute resources in on-premises machine or a VM inside private network

The screenshot shows the Azure Synapse studio interface with the 'Synapse live' workspace selected. The left sidebar contains navigation links: Analytics pools, SQL pools, Apache Spark pools, External connections, Linked services, Integration, Triggers, Integration runtimes (which is highlighted with a red box), Security, Access control, and Credentials. The main area is titled 'Integration runtimes' and displays a message: 'The integration runtime (IR) is the compute infrastructure to provide the following network environment.' A 'Learn more' link is present. Below this is a table header with columns: Name ↑, Type ↑, Sub-type ↑, and Status ↑. A red box highlights the '+ New' button. To the right, a sub-dialog titled 'Integration runtime setup' is open, asking 'Choose the network environment of the data source/destination or external compute to which the integration runtime will connect to for data movement or dispatch activities'. It shows two options: 'Azure' (with a cloud icon) and 'Self-Hosted' (with a server icon). At the bottom of the sub-dialog are 'Continue', 'Back', and 'Cancel' buttons.

Data Movement with Integration Runtimes

Scalable

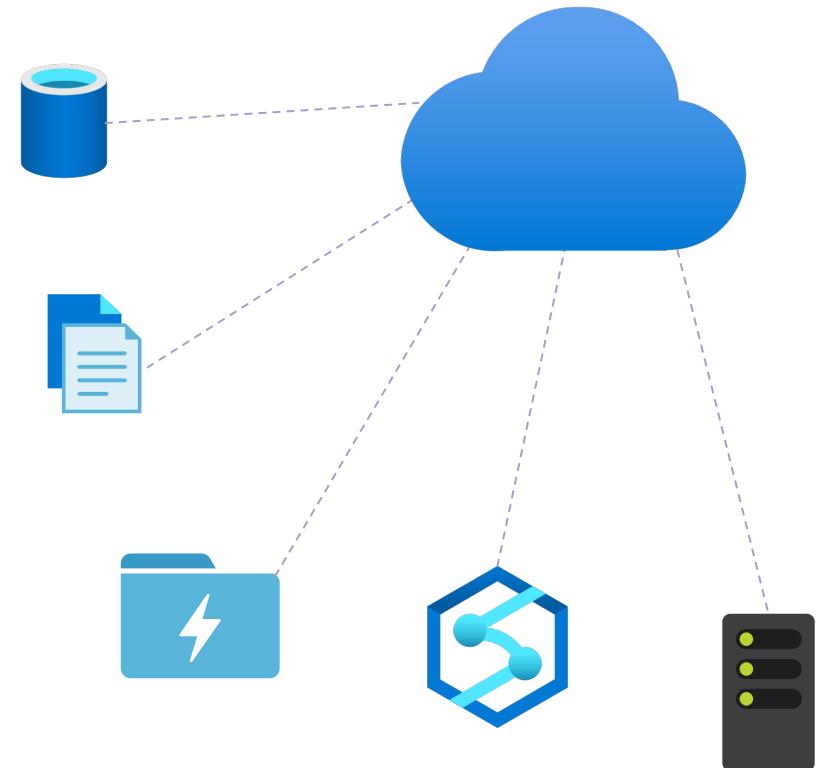
- per job elasticity
- Up to 4 GB/s

Simple

- Visually author or via code (Python, .Net, etc.)
- Serverless, no infrastructure to manage

Access all your data

- 90+ connectors provided and growing (cloud, on premises, SaaS)
- Data Movement as a Service: 25 points of presence worldwide
- Self-hostable Integration Runtime for hybrid movement



Pop Quiz 1

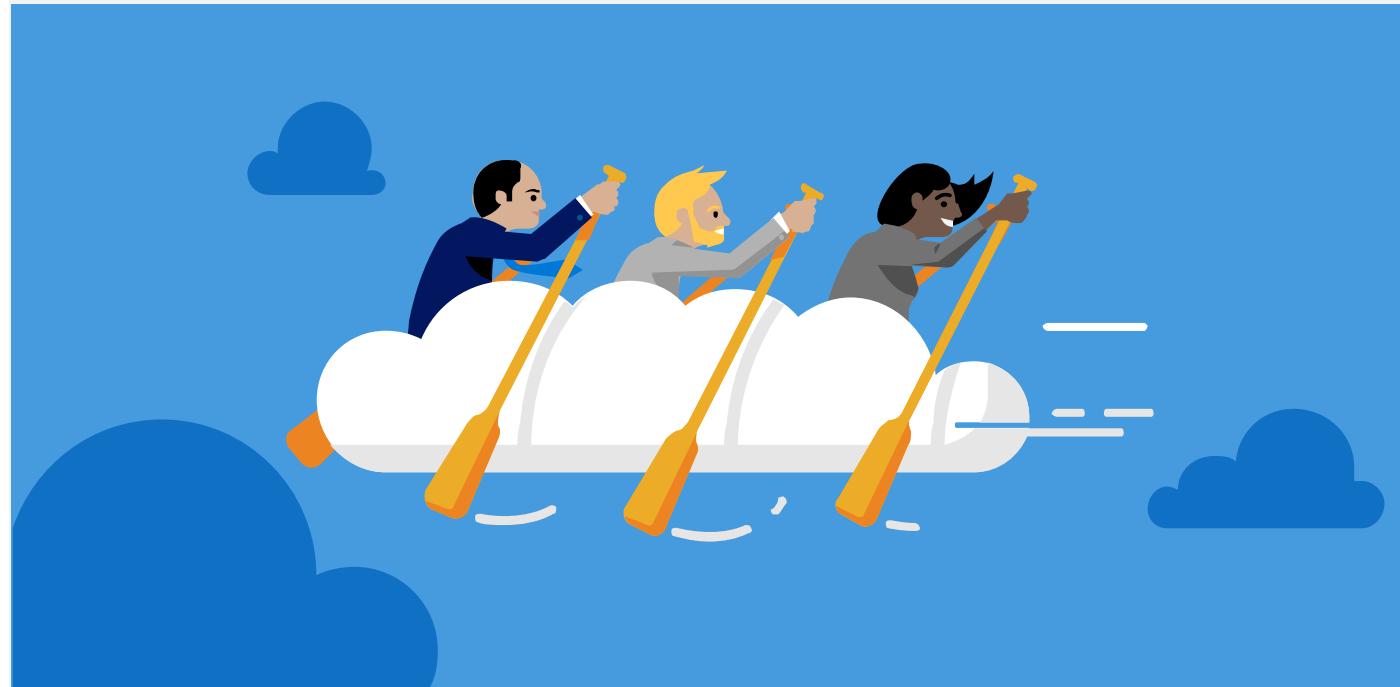
Which one of these is NOT a component of a Synapse pipeline?

A)
I.R.

B)
Linked
Service

C)
Table

D)
Activity



Pop Quiz 1

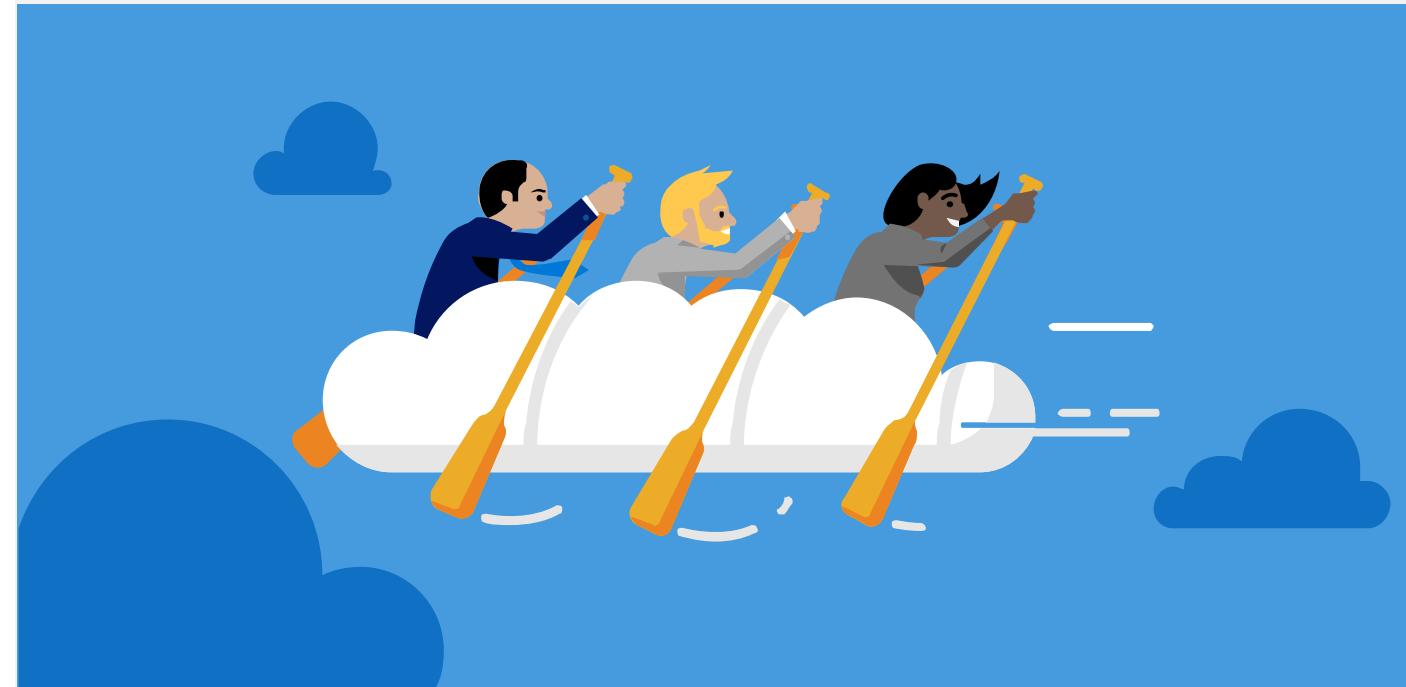
Which one of these is NOT a component of a Synapse pipeline?

A)
I.R.

B)
Linked
Service

C)
Table

D)
Activity



Files and Tables

COPY command

Overview

Copies data from source to destination

Benefits

- Retrieves data from all files from the folder and all its subfolders.
- Supports multiple locations from the same storage account, separated by comma
- Supports Azure Data Lake Storage (ADLS) Gen 2 and Azure Blob Storage.
- Supports CSV, PARQUET, ORC file formats

```
COPY INTO test_1
FROM 'https://XYZ.blob.core.windows.net/customerdatasets/test_1.txt'
WITH (
    FILE_TYPE = 'CSV',
    CREDENTIAL=(IDENTITY= 'Shared Access Signature',
    SECRET='<Your_SAS_Token>'),
    FIELDQUOTE = """",
    FIELDTERMINATOR=';',
    ROWTERMINATOR='0X0A',
    ENCODING = 'UTF8',
    DATEFORMAT = 'ymd',
    MAXERRORS = 10,
    ERRORFILE = '/errorsfolder/'--path starting from the storage container,
    IDENTITY_INSERT
)
```

```
COPY INTO test_parquet
FROM 'https://XYZ.blob.core.windows.net/customerdatasets/test.parquet'
WITH (
    FILE_FORMAT = myFileFormat
    CREDENTIAL=(IDENTITY= 'Shared Access Signature',
    SECRET='<Your_SAS_Token>')
)
```

Create External Table As Select (Polybase)

Overview

- Creates an external table and then exports results of the SELECT statement. These operations will import data into the database for the duration of the query

Steps:

- Create Master Key
- Create Credentials
- Create External Data Source
- Create External Data Format
- Create External Table

```
-- Create a database master key if one does not already exist
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'S0me!nfo'
;

-- Create a database scoped credential with Azure storage account key as the secret.
CREATE DATABASE SCOPED CREDENTIAL AzureStorageCredential
WITH
    IDENTITY = '<my_account>',
    SECRET   = '<azure_storage_account_key>'
;
;

-- Create an external data source with CREDENTIAL option.
CREATE EXTERNAL DATA SOURCE MyAzureStorage
WITH
(
    LOCATION  = 'wasbs://daily@logs.blob.core.windows.net/',
    CREDENTIAL = AzureStorageCredential
    , TYPE     = HADOOP
)
;

-- Create an external file format
CREATE EXTERNAL FILE FORMAT MyAzureCSVFormat
WITH (FORMAT_TYPE = DELIMITEDTEXT,
      FORMAT_OPTIONS(
          FIELD_TERMINATOR = ',',
          FIRST_ROW = 2)
)
;

--Create an external table
CREATE EXTERNAL TABLE dbo.FactInternetSalesNew
WITH(
    LOCATION = '/files/Customer',
    DATA_SOURCE = MyAzureStorage,
    FILE_FORMAT = MyAzureCSVFormat
)
;

AS SELECT T1.* FROM dbo.FactInternetSales T1 JOIN dbo.DimCustomer T2
ON ( T1.CustomerKey = T2.CustomerKey )
OPTION ( HASH JOIN);
```

Polybase vs Copy

Polybase

- GA, stable
- Needs CONTROL permission
- Fastest (at present)
- Enables querying via external tables
- Challenges:
 - Row width
 - Delimiters in text
 - Fixed line delimiter
 - Code complexity

Copy

- Relaxed permission
- Slightly slower, but improving
- No row width limit
- Supports delimiters in text
- Supports custom column and row delimiters

Best Practices for Files and Tables

Question...

How many different methods of loading ADLS can you think of?

What about a Synapse SQL Pool?



Ingest Flat files to tables

Ingest flat file data into Azure Storage (Azure Data Lake Store Gen2)

- When your data sources are on-premises, you need to move the data to Azure Storage before ingestion.
- Data in other cloud platforms needs to be moved to Azure Storage before ingestion.

Load from flat files as relational tables within the data warehouse

Ingest - Structuring ADLS Gen2

- Separate storage accounts for each environment: dev, test, & production.
- Use a common folder structure to organize data by degree of refinement.

ADLS Gen 2 Filesystem

Raw Data
/bronze

Query Ready
/silver

Report Ready
/gold

Ingest from on-premises data sources

Fastest is done by batch:

- Extract from data source to multiple CSV/Parquet files
- Use AzCopy to upload to ADLS

Alternative is query-insert:

- Set up SSIS self-hosted integration runtime on-premises
- Use Synapse Pipeline to extract/copy
- Use Synapse Pipeline to execute load procedure

Large Migrations:

- Use Azure Data Box where available

Ingest from Cloud Data Sources

Options:

- Extract using Synapse Pipelines
- Write to ADLS as Parquet files
- AzCopy is a fast move for files from S3 to ADLS

Ingest File Data Sources

Look out for these file format challenges...

Invalid file format

- Multiple row types
- Ragged columns

Row size > 1Mb

Datetime format/s (e.g., use of nanosecond date time)

NULL value literal/s

Free form text

Parquet partitions

XML data

Use of non-standard line delimiters (e.g., CR)

...and try these Solutions

- Use Spark to pre-process and fix data errors
- Flatten and parse XML in Spark
- Use COPY to ingest complex CSV instead of Polybase

Ingest and Store – Formats

For batch flat files, Azure Synapse Analytics supports CSV, Parquet, ORC, and JSON formats.

Ingest streaming data messages/events via Event Hub or IoT Hub.

Parquet format recommended for storing ingested data at various levels of refinement.

Ingest - When to BCP / Bulk Copy

Green fields: Never

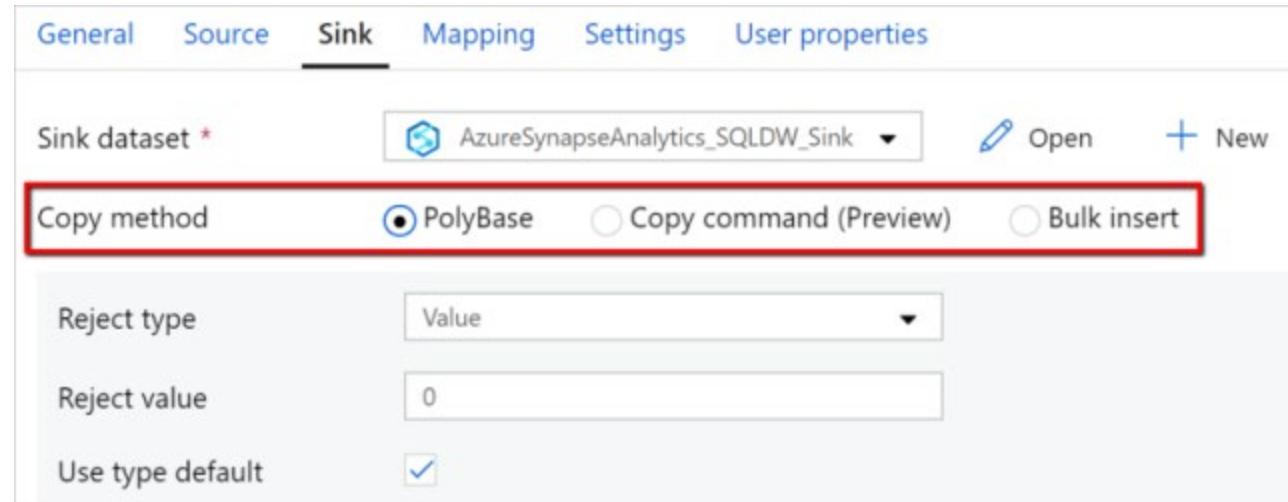
- Network unreliability, no retries
- Needs VM in cloud, performance dependent on VM configuration
- Doesn't support ADLS
- Reduces concurrency
- Control-gated performance limitation, can not scale with DWU

Migrations:

- Use Synapse Pipeline or AzCopy
- Bulk Copy will work, but it will be slower than other methods

Ingest – Synapse Pipelines

- Un-check USE TYPE DEFAULT, it is not a best practice.
- Land data in ADLS Gen2, then ingest using Polybase / COPY.
 - This means you can re-ingest the same data set without having to repeat extracts, and better demonstrate ingestion performance.



Ingest and Store – Loading staging tables

Indexing

Use Heap tables

Speed load performance by staging data in heap tables and temporary tables prior to running transformations.

Only load to a CCI table if the test requires a load to a single table, then complex end-user queries against that table.

Ingest and Store – Loading staging tables

Distribution

Use Round Robin Distribution for:

- Potentially useful tables created from raw input.
- Temporary staging tables used in data preparation.

Other distribution considerations:

- Never load to a REPLICATED table
- Load to a ROUND_ROBIN table if the test is ONLY raw ingestion performance, or if the table is very small
- Load to a HASH table if the task is a pipeline with subsequent transformations using the loaded table

Ingest – Scaling to shorten duration

Ingestion duration is correlated with the number of DWU's allocated to the SQL Pool.

For every *doubling* of the DWU's you *halve* the ingestion time.

$$2d = t/2$$

d: DWU

T: ingestion time

Only applies from DWU500c – DWU30000c

Export to files with CETAS

CETAS = parallel operation that creates external table metadata and exports the SELECT query results to a set of files in your storage account.

Store frequently used parts of queries, like joined reference tables, to a new set of files. You can then join to this single external table instead of repeating common joins in multiple queries.

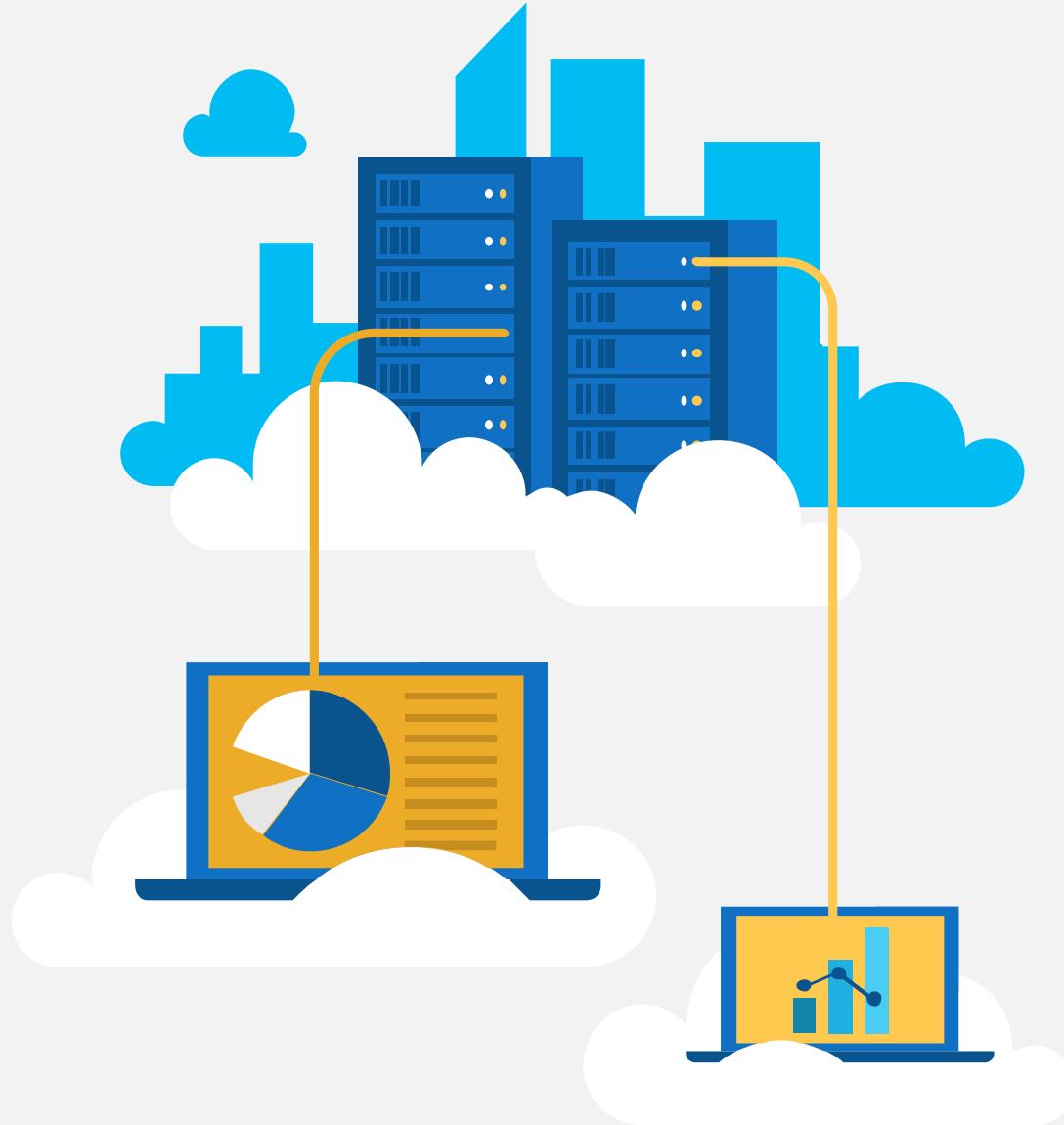
As CETAS generates Parquet files, statistics will be automatically created when the first query targets this external table, resulting in improved performance.

Pop Quiz 2

True or False: Both COPY command AND Polybase require CONTROL permission

TRUE

FALSE

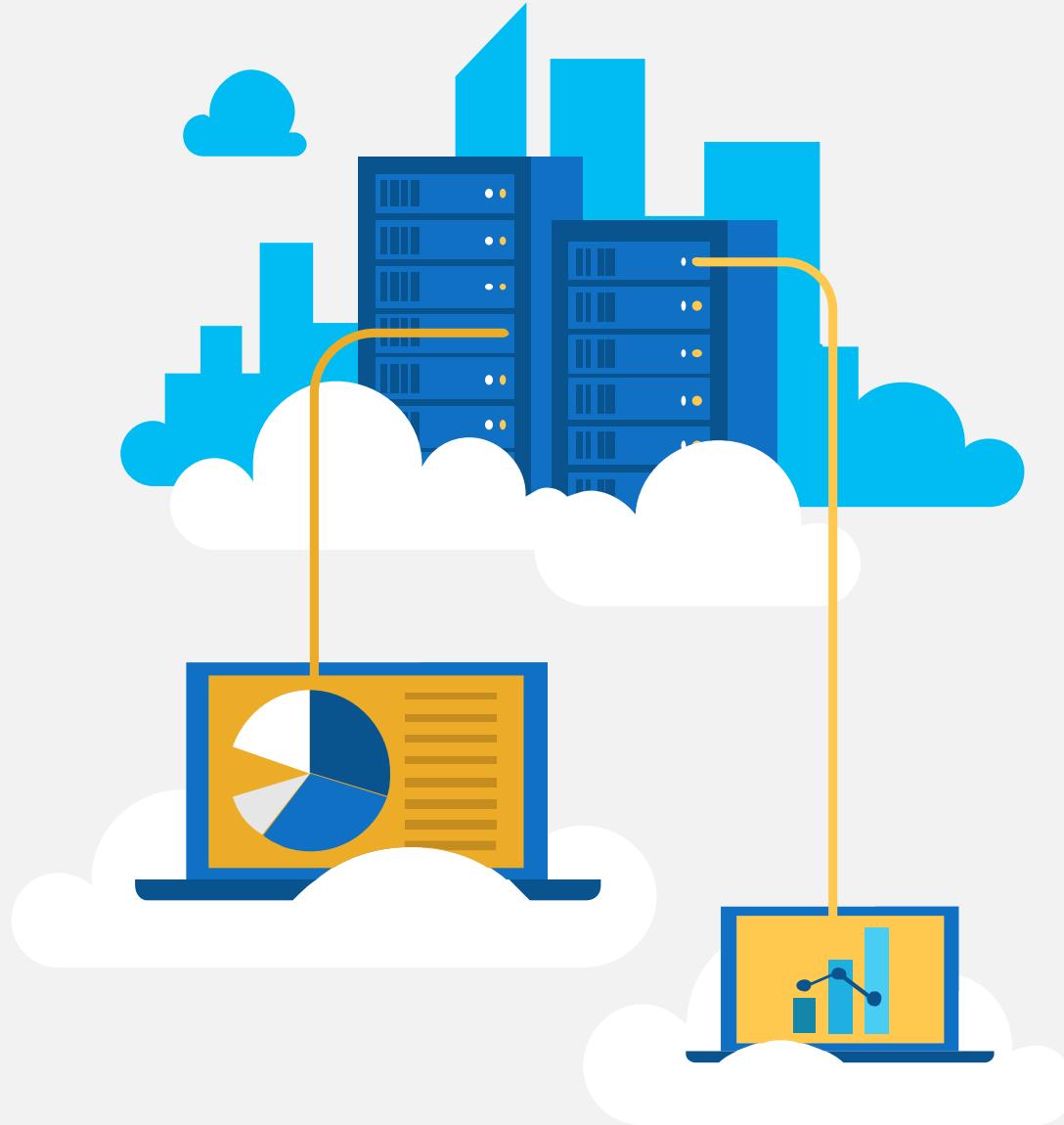


Pop Quiz 2

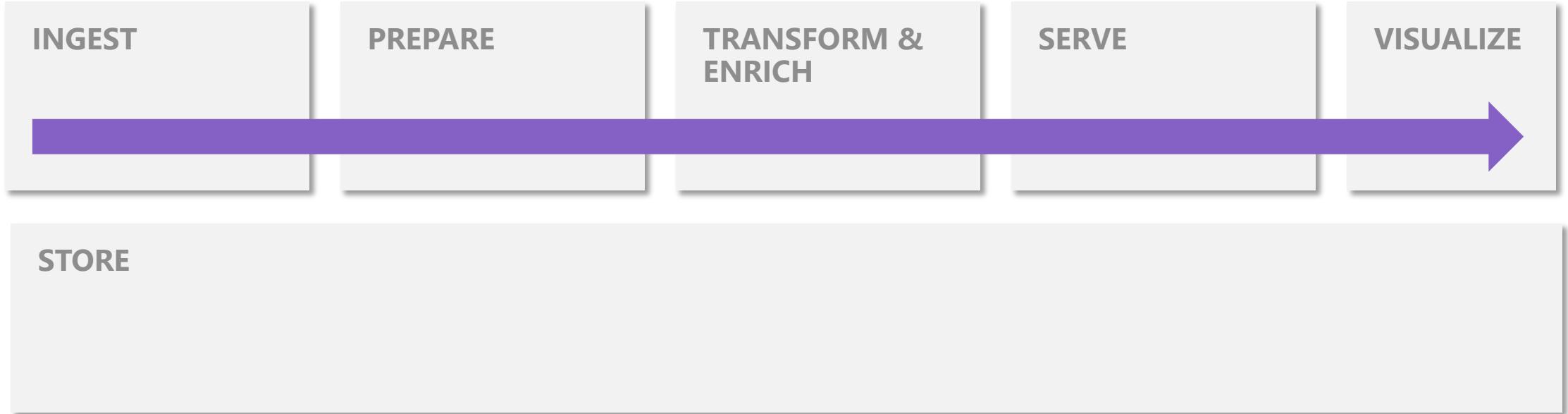
True or False: Both COPY command AND Polybase require CONTROL permissions

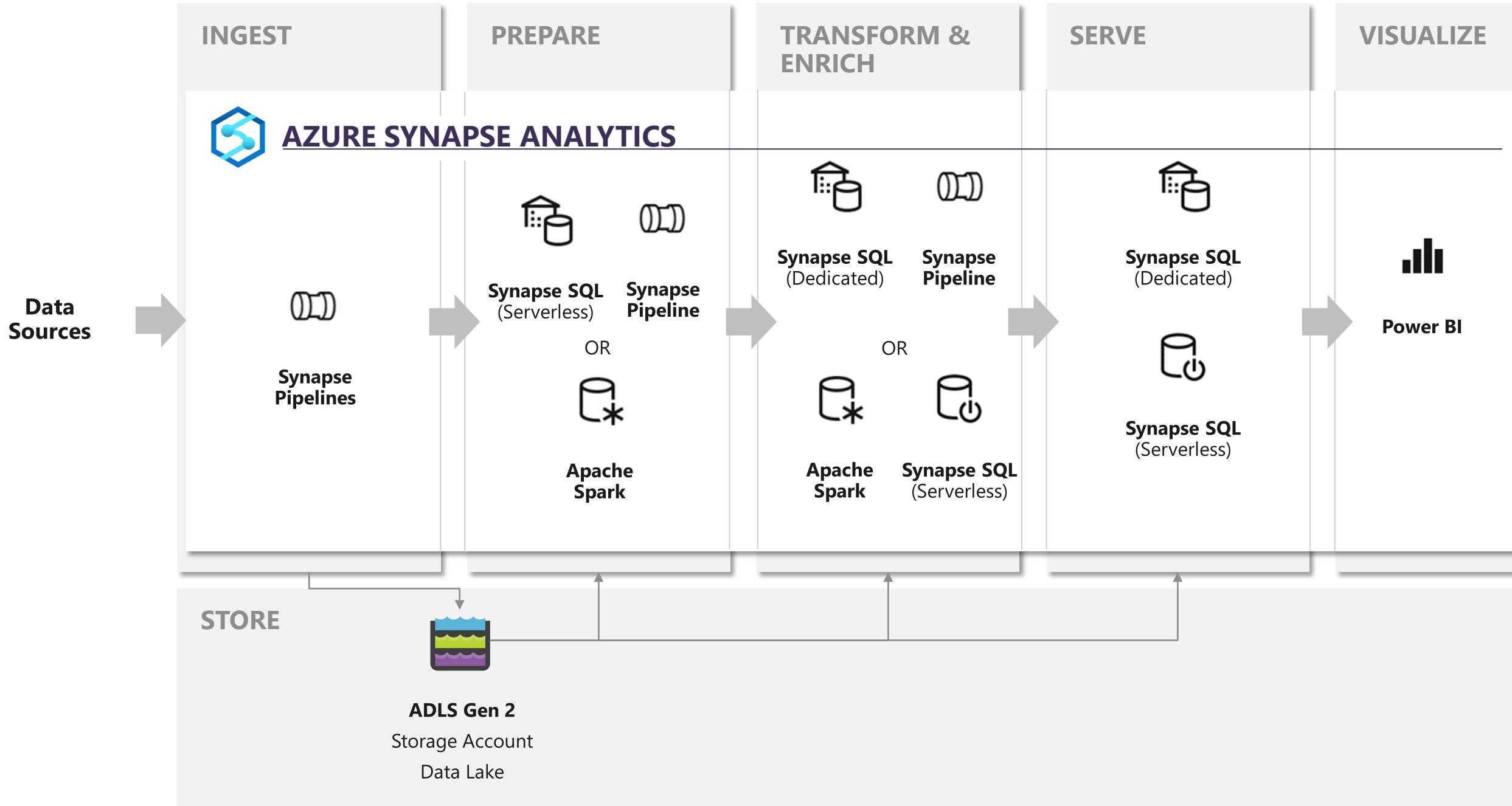
TRUE

FALSE



Modern Data Warehouse







Thank you

DEMO: How to whiteboard designs



Breakout Activity: Data Lake Design & Security Considerations

OUTCOMES

As a result of participating in this activity, you will better be able to **design, optimize and secure a file system within ADLS Gen2.**

ACTIVITY



60 minutes



Learning Adviser



Table Group Channel



As a team review WWI requirements provided in student guide



Open the **whiteboard** and as a team [answer the provided challenge questions.](#)

The questions are pre-loaded into the whiteboard.



Welcome

Keynote

Demo Walkthrough

Break

Data Loading &
Data Lake
Organization

**Activity: Data
Lake Design &
Security
Considerations**

Break

Lab: Data
Integration Part
1

Break

Data
Transformations

Activity: Data
Engineering
Discussion

Lab: Data
Integration Part
2

Closing

DEMO: How to access your lab environment



Build Hands-on Lab: Data Integration Part 1

OUTCOMES

As a result of participating in this lab, you will be better able to **architect and deploy data integration pipelines**.

ACTIVITY



45 minutes



Independent



Table Group Channel & CloudLabs Environment



Independently review and complete Exercises 1-4 in the Lab 1 Guide.



Login to your **CloudLabs environment** and **work through a set of tasks you would typically follow** in work associated with ingesting and integrating your customer's data.

The Lab Guide is available in the 'Files' tab of the General channel. Engage your Table Group call for support and troubleshooting.



Welcome

Keynote

Demo Walkthrough

Break

Data Loading &
Data Lake
Organization

Activity: Data
Lake Design &
Security
Considerations

Break

**Lab: Data
Integration
Part 1**

Break

Data
Transformations

Activity: Data
Engineering
Discussion

Lab: Data
Integration Part
2

Closing

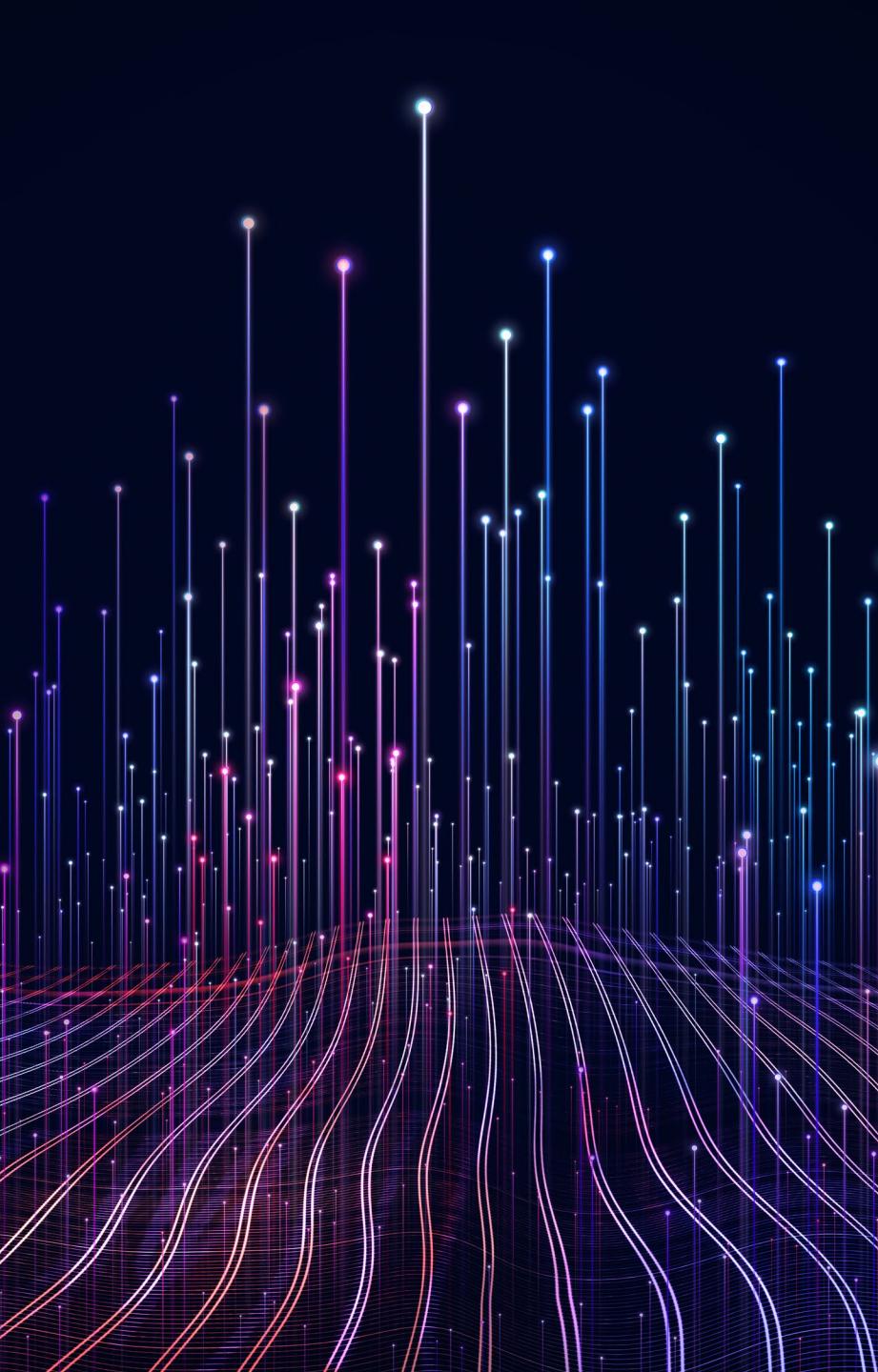
Welcome Back!

8:00-8:05	Welcome	
8:05-8:15	Keynote	
8:15-8:45	Demo Walkthrough	Main Call
8:45-9:00	Break	
9:00-10:00	Data Loading & Data Lake Organization	
10:00-11:00	Activity: Data Lake Design & Security Considerations	
11:00-11:15	Break	Table Group Call
11:15-12:00	Build Hands-on: Data Integration Part 1	
12:00-1:00	Break	Main Call
1:00-1:30	Data Transformations	Main Call
1:30-2:00	Activity: Data Engineering Discussion	Table Group Call
2:00-3:00	Build Hands-on: Data Integration Part 2	
3:00-3:15	Closing	Main Call



- Presentation/
Whole Group
- Lab
- Activity/ Discussion/
Group Work
- Announcements

Data Transformations



Agenda

1 Preparing to transform

Understanding and exploring the data.

2 Apply transformations

Apply coded and code-free transformations.

3 Serverless transforms

Use Azure Synapse serverless SQL to transform data with SQL scripts.

4 Transform with Spark

Here we have an example of what the agenda item would look like.

5 Best practices

Best practices for data transformation.

Typical Data Transformations

- Create persistent staging area / data vault
- Standardize data from different sources
- Remove duplicate rows
- Impute missing values
- Calculate derived values
- Prepare data for facts and dimensions

Applying transformations

Code based transformations

Familiar gesture to generate T-SQL scripts from SQL metadata objects such as tables.

A screenshot of the Azure Data Studio interface. On the left, there's a tree view of databases, tables, and columns. In the main area, a context menu is open over a column named 'Predicted_fareA...'. The menu items include 'New SQL script' (with a 'New notebook' option), 'Select TOP 1000 rows', 'CREATE', 'DROP', and 'DROP and CREATE'.

Starting from a table, auto-generate a single line of PySpark code that makes it easy to load a SQL table into a Spark dataframe and author transforms in a notebook.

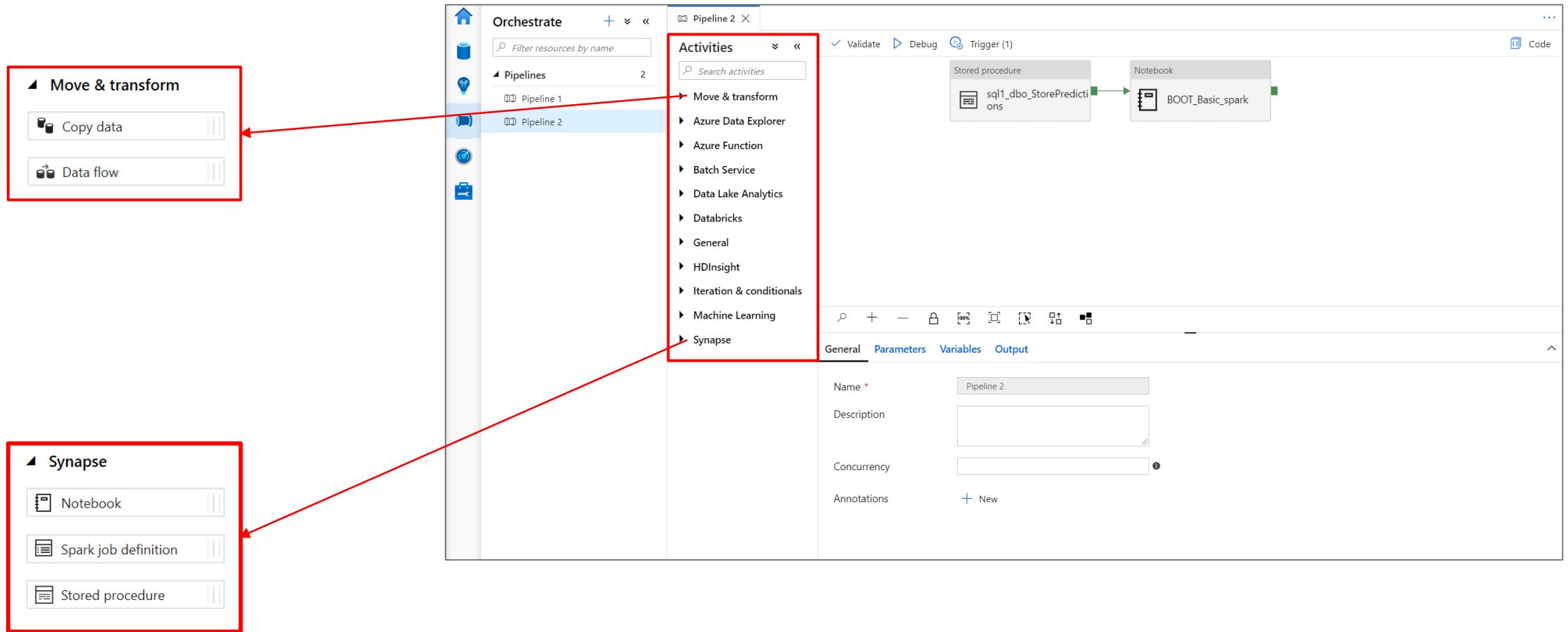
A screenshot of the Azure Data Studio interface. It shows a tree view of databases, tables, and columns. A context menu is open over a column named 'Predicted_fareA...'. The menu includes options like 'New SQL script' (with 'New notebook'), 'Refresh', and 'Load to DataFrame'. A red box highlights the 'Load to DataFrame' option. A red arrow points from this option down to a 'Notebook 1' window at the bottom.

Notebook 1

```
Cell 1
[ ] 1 val df = spark.read.sqlanalytics("sql1.dbo.NycTaxiPredict")
```

Transform with Pipelines

Orchestrate transformations with Synapse Pipelines.



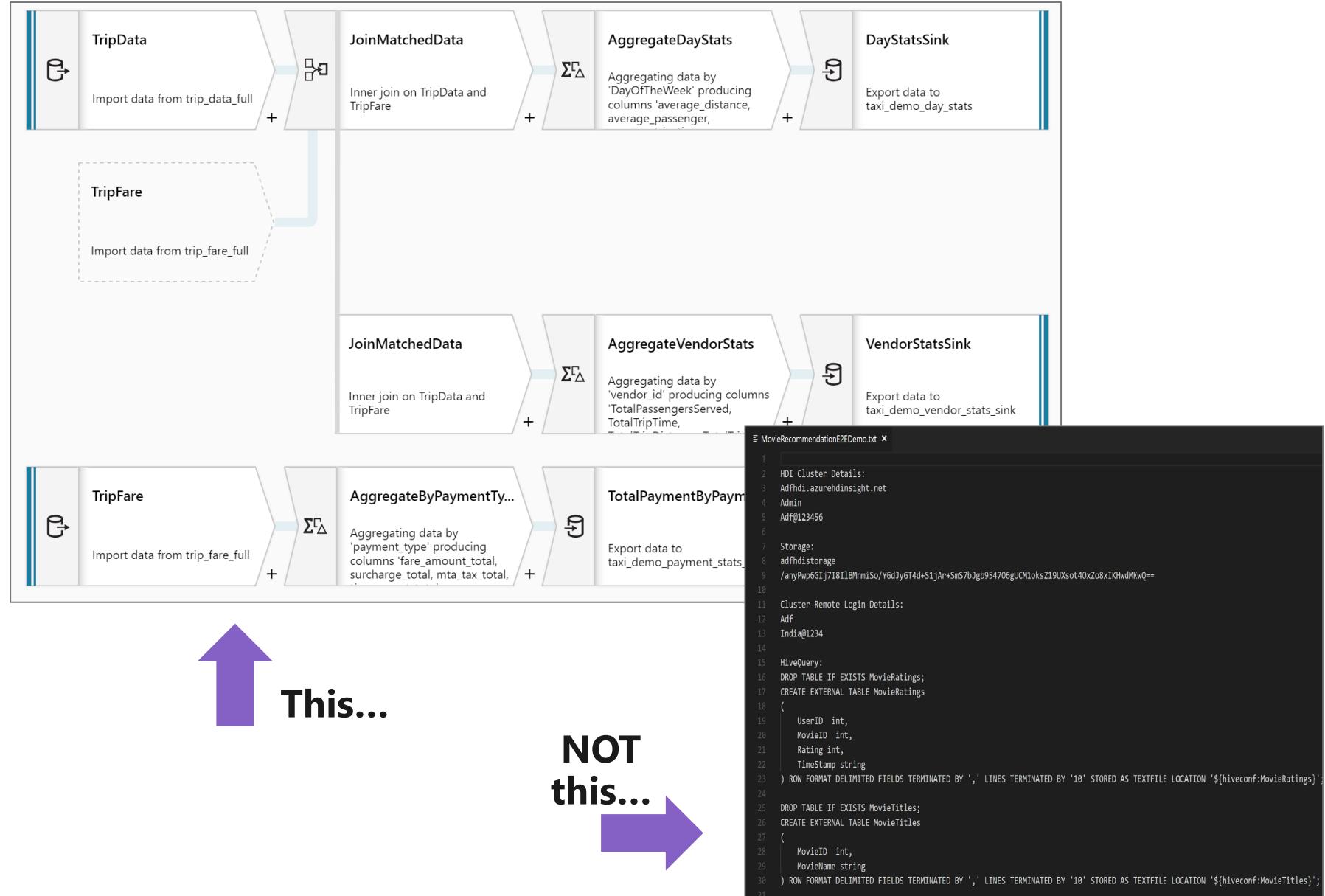
No Code Transform with Data Flows

Overview

It offers data cleansing, transformation, aggregation, conversion, etc

Benefits

- Cloud scale via Spark execution
- Guided experience to easily build resilient data flows
- Flexibility to transform data per user's comfort
- Monitor and manage dataflows from a single pane of glass



Transform with serverless SQL

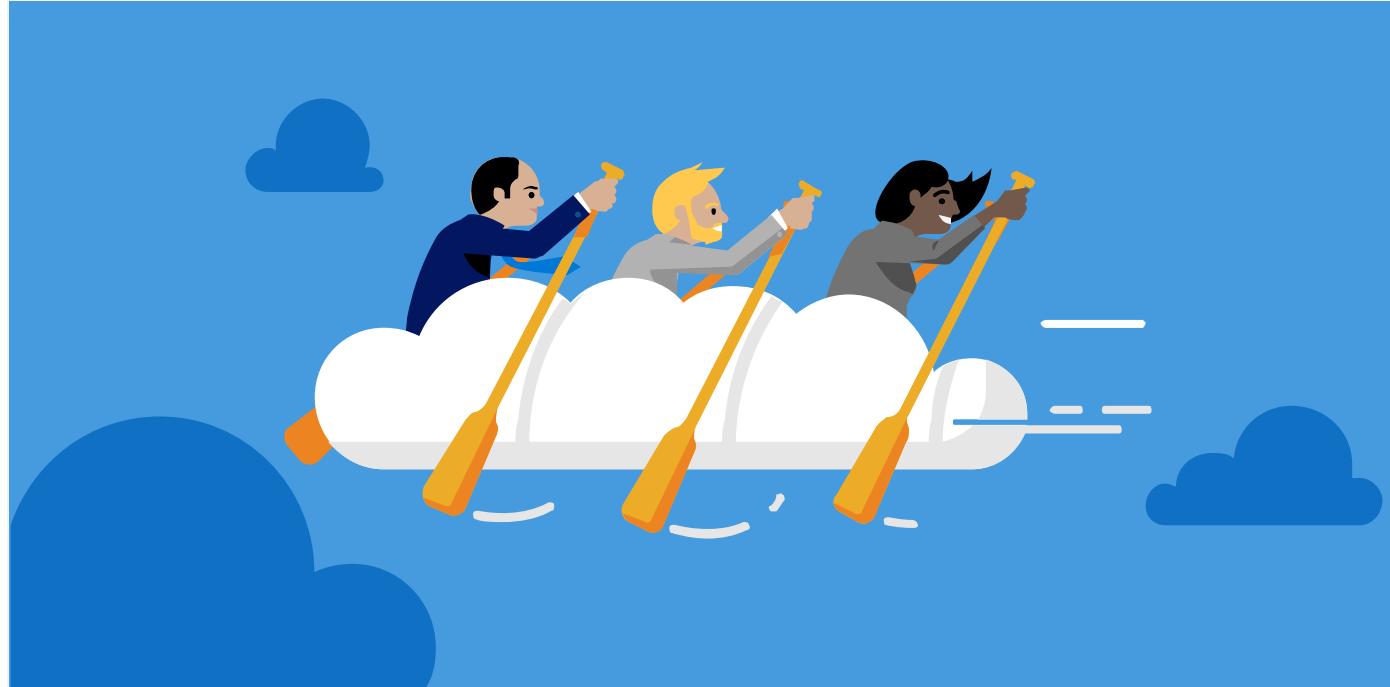
Pop Quiz 1

What's the largest scale TPC-H workload serverless SQL has successfully run?

A)
100TB

B)
1PB

C)
10PB



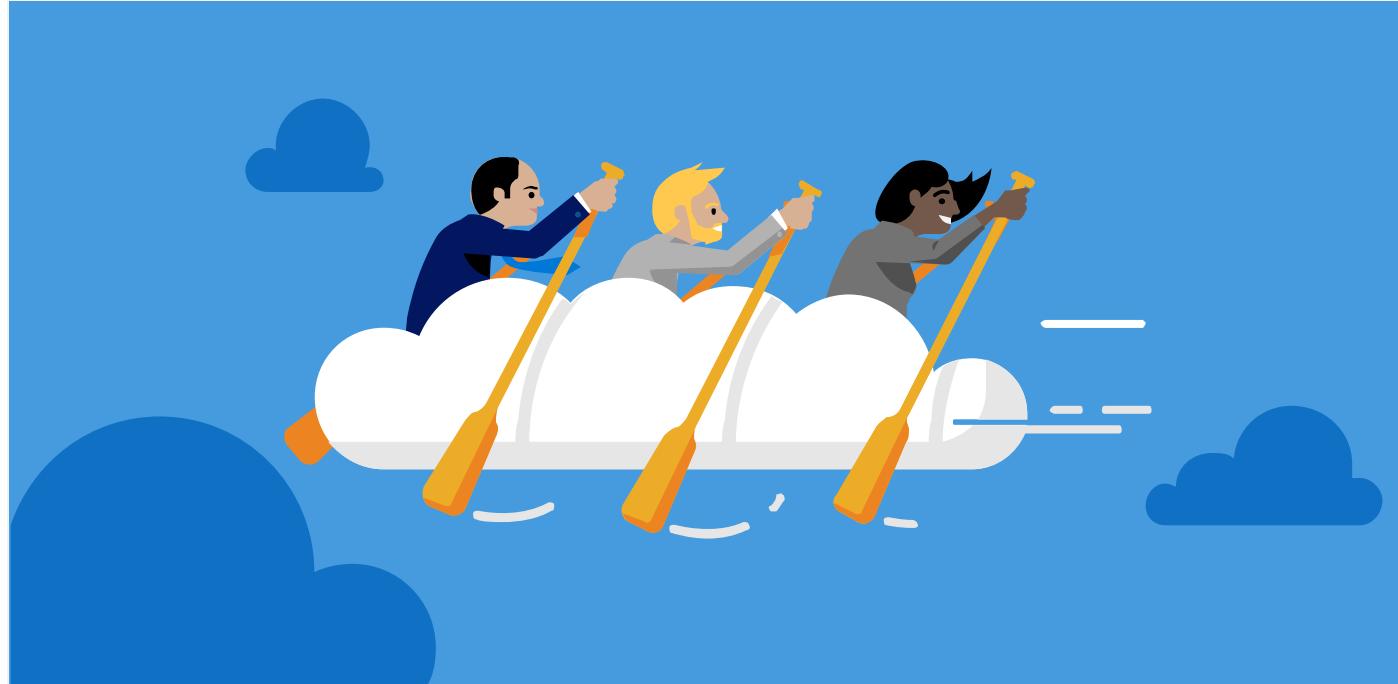
Pop Quiz 1

What's the largest scale TPC-H workload a serverless SQL pool has successfully run?

A)
100TB

**B)
1PB**

C)
10PB



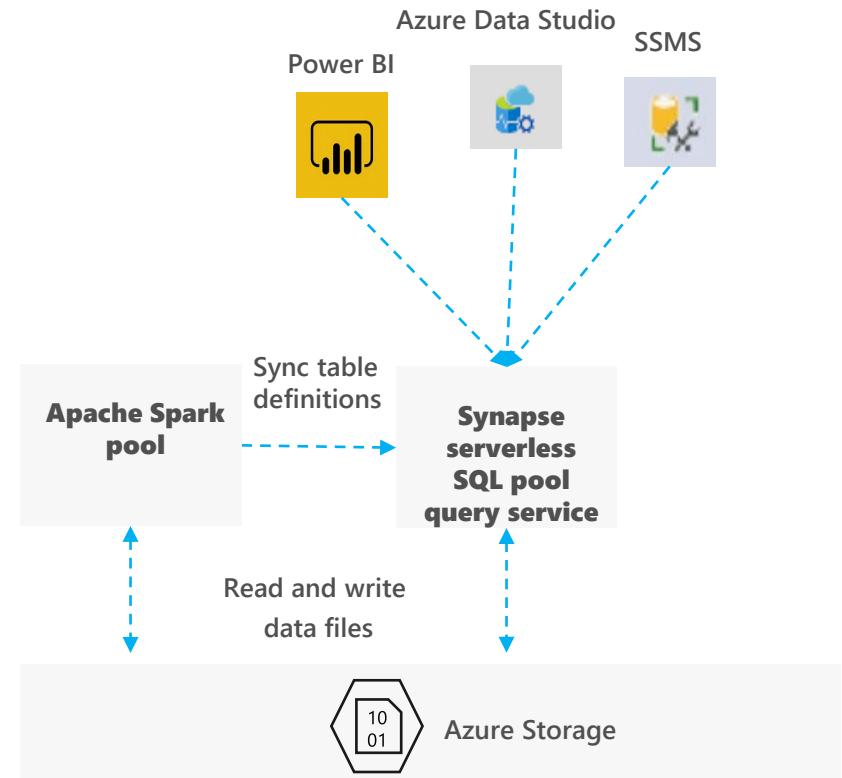
serverless SQL pool

Overview

An interactive query service that enables you to use standard T-SQL queries over files in Azure storage.

Benefits

- Use SQL to work with files on Azure storage
 - Directly query files on Azure storage using T-SQL
 - Logical Data Warehouse on top of Azure storage
 - Easy data transformation of Azure storage files
- Supports any tool or library that uses T-SQL to query data
- Automatically synchronize tables from Spark
- Serverless
 - No infrastructure, no upfront cost, no resource reservation
 - Pay only for query execution (per data processed)



Recommended usage scenarios

Quick data exploration

- Easily explore schema and data in files on Azure storage
- Supports various file formats (Parquet, CSV, JSON)
- Direct connector to Azure storage for large BI ecosystem

Logical Data Warehouse

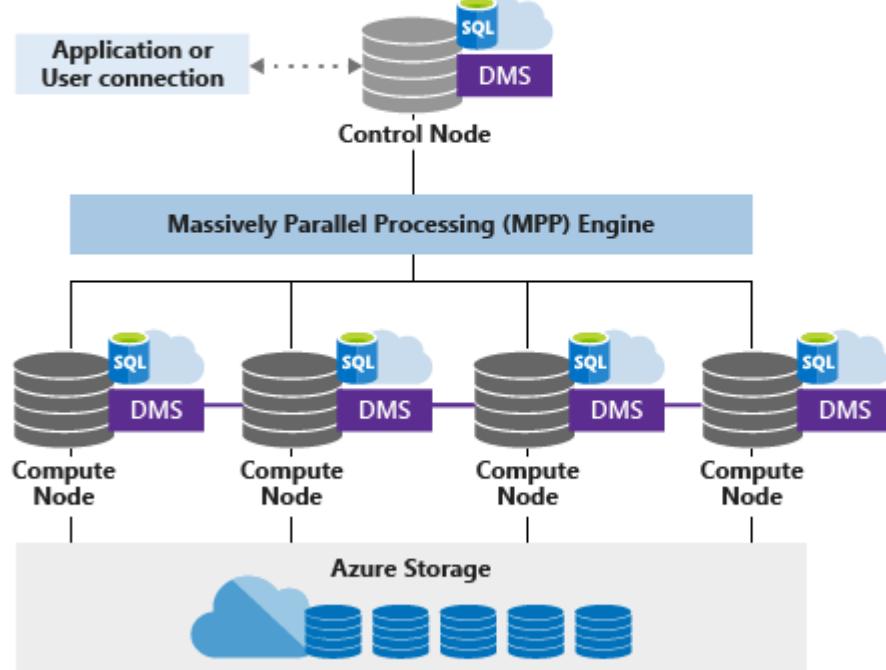
- Model raw files as virtual tables and views
- Use any tool that works with SQL to analyze files
- Use enterprise-grade security model

Easy data transformation

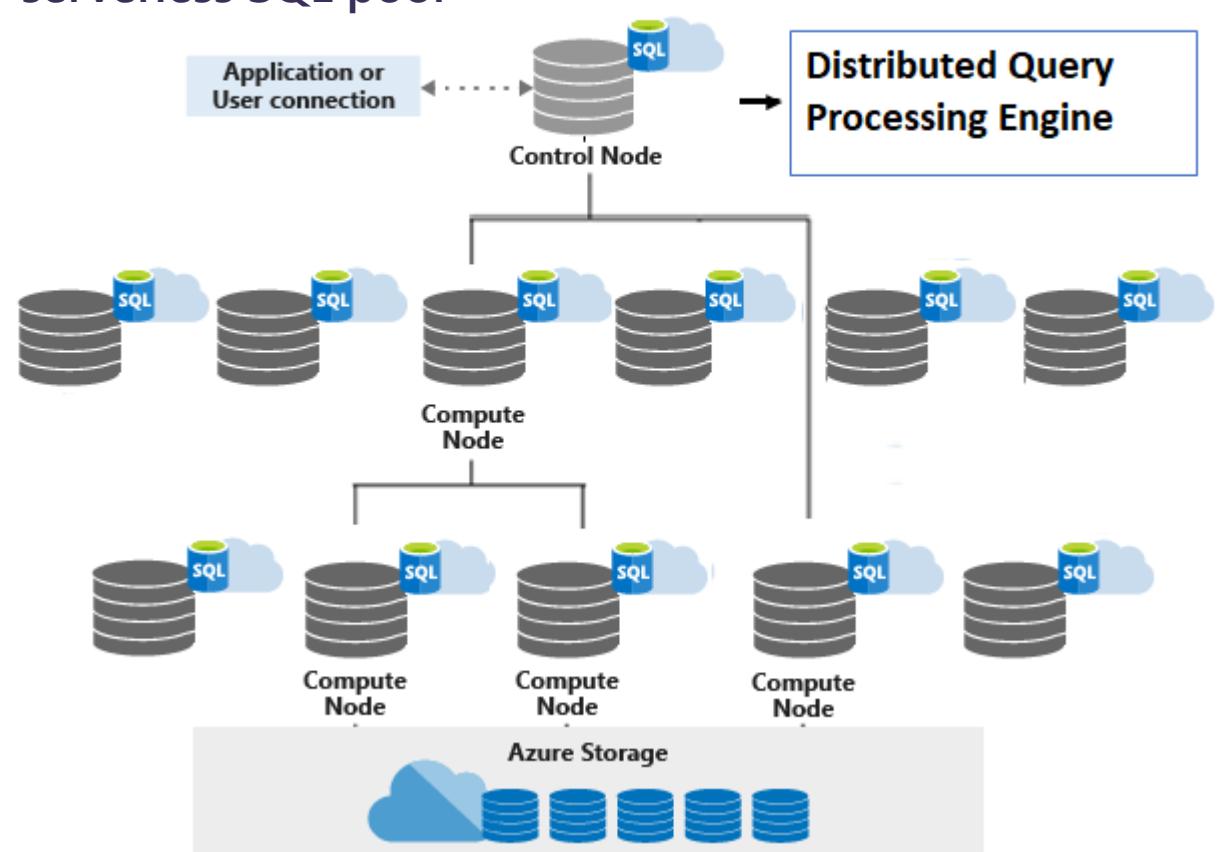
- Transform CSV to parquet format
- Move data between containers and accounts
- Save the results of queries on external storage

serverless SQL pool

dedicated SQL pool



serverless SQL pool



Easily explore files on storage

The screenshot illustrates the process of exploring files on storage and executing SQL queries against them.

Left Panel (File Explorer): Shows the Azure portal navigation bar and a sidebar with categories: Storage accounts (1), Databases (3), and Datasets (5). The main area displays a list of files under 'opendataset' in the 'internalsandboxwe' storage account. A specific file, 'New SQL script - Select TOP 100 rows', is highlighted with a red box.

Right Panel (Query Editor): Shows the 'opendataset' details page. The 'SQL script 1' tab is selected, displaying the following T-SQL script:

```
1 SELECT
2     TOP 100 *
3     FROM
4     OPENROWSET(
5         BULK 'https://internalsandboxwe.dfs.core.windows.net/opendataset/holidays/part-00001-bd1aba93-a85a-4909-8bf4-f79afb6c946f-c000.snappy.parquet'
6         FORMAT='PARQUET'
7     ) AS [r];
```

The 'Connect to' dropdown is set to 'SQL on-demand' (highlighted with a red box).

Bottom Panel (Results): Displays the results of the executed query in a table format. The results show four rows of data from the 'holidays' dataset.

VENDORID	TPEPICKUPDATETIME	TPEPDROPOFFDATETIME	PASSENGERCOUNT	TRIPDISTANCE	PULOCATIONID	DLOCATIONID
VTS	2009-05-07T23:1...	2009-05-07T23:2...	1	2.94	NULL	NULL
VTS	2009-05-07T16:3...	2009-05-07T16:3...	5	0.73	NULL	NULL
VTS	2009-05-08T14:5...	2009-05-08T15:0...	3	0.55	NULL	NULL
VTS	2009-05-07T15:5...	2009-05-07T16:1...	1	2.5	NULL	NULL

A message at the bottom indicates: '00:00:31 Query executed successfully.'

Easily query files in various formats

Overview

Use OPENROWSET function to access data stored in various file formats

Benefits

Enables you to read CSV, parquet, and JSON files

Provides unified T-SQL interface for all file types

Use standard SQL language to transform and analyze returned data

- Use JSON functions to get the data from underlying files.
- Use JSON functions to get data from PARQUET nested types

```
SELECT TOP 10 *
FROM OPENROWSET(
    BULK 'https://XYZ.blob.core.windows.net/csv/taxi/*.csv',
    FORMAT = 'CSV')
WITH (
    country_code VARCHAR(4),
    country_name VARCHAR(50),
    year INT,
    population INT
) AS nyc
```

```
SELECT TOP 10 *
FROM OPENROWSET(
    BULK 'https://XYZ.blob.core.windows.net/parquet/taxi/*.parquet',
    FORMAT = 'PARQUET') AS nyc
```

```
SELECT TOP 10 *
    JSON_VALUE(jsonContent, '$.countryCode') AS country_code,
    JSON_VALUE(jsonContent, '$.countryName') AS country_name,
    JSON_VALUE(jsonContent, '$.year') AS year
    JSON_VALUE(jsonContent, '$.population') AS population
FROM OPENROWSET(
    BULK 'https://XYZ.blob.core.windows.net/json/taxi/*.json',
    FORMAT='CSV',
    FIELDTERMINATOR = '0x0b',
    FIELDQUOTE = '0x0b',
    ROWTERMINATOR = '0x0b'
)
WITH ( jsonContent varchar(MAX) ) AS json_line
```

	country_code	country_name	year	population
1	LU	Luxembourg	2017	594130

Automatic schema inference

Overview

OPENROWSET will automatically determine columns and types of data stored in external file.

Benefits

No need to up-front analyze file structure to query the file
OPENROWSET identifies columns and their types based on underlying file metadata.

Perfect solution for data exploration where schema is unknown.

The functionality is available for both parquet & CSV files.

```
SELECT TOP 10 *
FROM OPENROWSET(
    BULK 'https://XYZ.blob.core.windows.net/csv/taxi/*.parquet',
    FORMAT = 'PARQUET') AS nyc
```

	country_code	country_name	year	population
1	LU	Luxembourg	2017	594130

```
SELECT
    TOP 100 *
FROM
    OPENROWSET(
        BULK 'https://azuresynapsesa.dfs.core.windows.net/default/RetailData/StoreDemoGraphics.csv',
        FORMAT = 'CSV',
        PARSER_VERSION='2.0',
        HEADER_ROW = TRUE) AS [result]
```

StoreId	RatioAge60	CollegeRatio	Income	HighIncome15...	LargeHH	MinoritiesRatio	More1FullTime...	DistanceNeare...	SalesN
2	0.232864734	0.248934934	10.55320518	0.463887065	0.103953406	0.114279949	0.303585347	2.110122129	1.1428
5	0.117368032	0.32122573	10.92237097	0.535883355	0.103091585	0.053875277	0.410568032	3.801997814	0.6818

Defined the query result schema inline

Overview

Specify columns and types at query time.

Benefits

Define result schema at query time in WITH clause.

No need for external format files.

Explicitly define exact return types, their sizes, and collations.

Improve performance by column elimination in parquet files.

```
SELECT TOP 10 *
FROM OPENROWSET(
    BULK 'https://XYZ.blob.core.windows.net/csv/taxi/*.csv',
    FORMAT = 'CSV')
WITH (
    country_code VARCHAR(4),
    country_name VARCHAR(50),
    year INT,
    population INT
) AS nyc
```

	country_code	country_name	year	population
1	LU	Luxembourg	2017	594130

Customize the content parsing to fit your case

Overview

Uses OPENROWSET function to access data from various types of CSV files.

Benefits

Ability to read CSV files with custom format

- With or without header row
- Handle any new-line terminator (Windows or Unix style)
- Use custom field terminator and quote character
- Read UTF-8 and UTF-16 encoded files
- Use only a subset of columns by specifying column position after column types

```
SELECT *
FROM OPENROWSET(
    BULK 'https://XYZ.blob.core.windows.net/csv/population/population.csv',
    FORMAT = 'CSV',
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n'
)
WITH (
    [country_code] VARCHAR (5) 2,
    [country_name] VARCHAR (100) 4,
    [year] smallint 7,
    [population] bigint 9
) AS [r]
WHERE
    country_name = 'Luxembourg'
    AND year = 2017
```

Second, fourth, seventh and ninth columns are returned

	country_code	country_name	year	population
1	LU	Luxembourg	2017	594130

Easily query multiple files, with wildcards

Overview

Uses OPENROWSET function to access data from multiple files or folders using wildcards in path

Benefits

Offers reading multiple files/folders through usage of wildcards

Offers reading specific file/folder

Supports use of multiple wildcards

```
SELECT YEAR(pickup_datetime) AS [year],  
       SUM(passenger_count) AS passengers_total,  
       COUNT(*) AS [rides_total]  
FROM OPENROWSET(  
    BULK 'https://XYZ.blob.core.windows.net/csv/taxi/year=*/month=1/*.parquet',  
    FORMAT = 'PARQUET') AS nyc  
GROUP BY YEAR(pickup_datetime)  
ORDER BY YEAR(pickup_datetime)
```

	year	passengers_total	rides_total
1	2001	14	10
2	2002	29	16
3	2003	22	16
4	2008	378	188
5	2009	594	353
6	2016	102093687	61758523
7	2017	184464988	113496932
8	2018	86272771	53925040
9	2019	37	29
...	2020	6	6

Query partitioned data, using the folder structure

Overview

Uses OPENROWSET function to access data partitioned in sub-folders

Benefits

Use filepath() function to access actual values from file paths.

Eliminate sub-folders/partitions before the query starts execution

Query Spark/Hive partitioned data sets

```
SELECT  
    r.filepath(1) AS [year]  
    ,r.filepath(2) AS [month]  
    ,COUNT_BIG(*) AS [rows]  
FROM OPENROWSET(  
    BULK 'https://XYZ.blob.core.windows.net/year=*/month=/*/*.parquet',  
    FORMAT = 'PARQUET') AS [r]  
WHERE r.filepath(1) IN ('2017')  
    AND r.filepath(2) IN ('10', '11', '12')  
  
GROUP BY r.filepath(),r.filepath(1),r.filepath(2)  
ORDER BY filepath
```

year	month	rows
2017	10	9768815
2017	11	9284803
2017	12	9508276

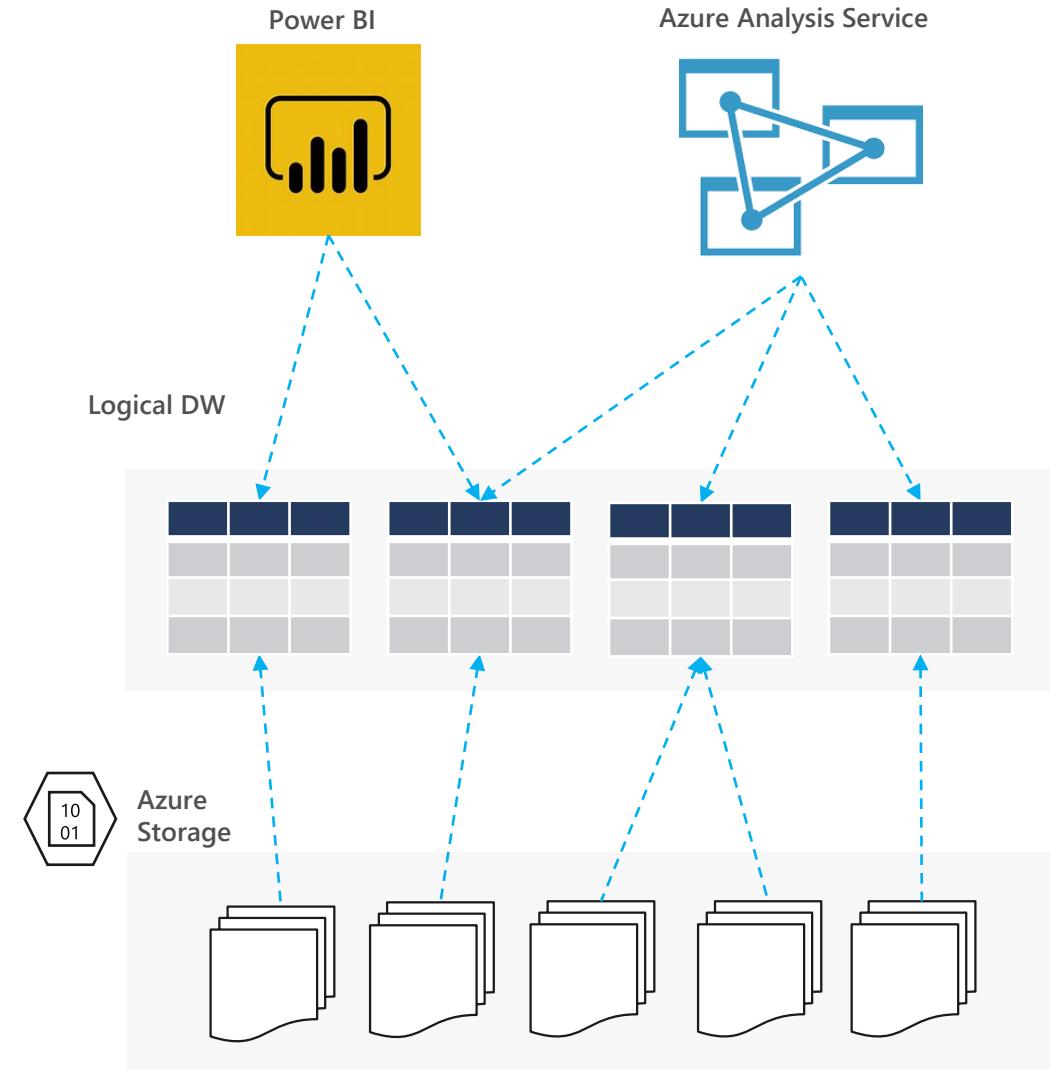
Synapse serverless SQL pool as a logical data warehouse

Overview

Logical relational layer on top of physical files in Azure Storage.

Benefits

- Abstract physical storage and file formats using well understandable relational concepts such as tables and views.
- Direct connector to Azure storage for large ecosystem of BI tools
- BI tools that use SQL can work with files on storage
 - Analytic tools use external tables that represent proxy to actual files.
 - No need for custom connectors in BI tools.
- Provides complex data processing (joining and aggregation) on top of raw files.
- Apply enterprise-ready security model and access control using battle-tested SQL Server permission model on top of Azure storage files



Logical Data Warehouse views

Overview

serverless SQL pool logical data warehouse views are created on external files placed in customer Azure storage

Benefits

Create SQL views on externally stored data

Access files using the view from various tools and language

Leverage rich T-SQL language to process and analyze data in external files exposed via views

Create PowerBI reports on the views created on external data

```
USE [mydbname]
GO

DROP VIEW IF EXISTS populationView
GO

CREATE VIEW populationView AS
SELECT *
FROM OPENROWSET(
    BULK 'https://XYZ.blob.core.windows.net/csv/population/*.csv',
    FORMAT = 'CSV',
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n'
)
WITH (
    [country_code] VARCHAR (5) ,
    [country_name] VARCHAR (100),
    [year] smallint,
    [population] bigint
) AS [r]
```

```
SELECT
    country_name, population
FROM populationView
WHERE
    [year] = 2019
ORDER BY
    [population] DESC
```

	country_name	population
1	China	1389618778
2	India	1311559204
3	United States	331883986
4	Indonesia	264935824
5	Pakistan	210797836
6	Brazil	210301591
7	Nigeria	208679114
8	Bangladesh	161062905
9	Russia	141944641
10	Mexico	127318112

Logical Data Warehouse - tables

Overview

Create external tables that reference external files in your serverless SQL pool logical data warehouse

Benefits

Create external tables that reference set of files on Azure storage.

Join and transform multiple tables in the same query.

Enables you to analyze external files with the same experience that you have in classic databases.

Manage column statistics in external tables.

Manage access rights per table.

Create PowerBI reports on the views created on external data

```
USE [mydbname]
```

```
GO
```

```
DROP TABLE IF EXISTS dbo.Population
```

```
GO
```

```
CREATE EXTERNAL TABLE dbo.Population (
```

```
country_code VARCHAR (5) COLLATE Latin1_General_BIN2,  
country_name VARCHAR (100) COLLATE Latin1_General_BIN2,  
year smallint,  
population bigint
```

```
)
```

```
WITH(
```

```
LOCATION = '/csv/population/population-* .csv',  
DATA_SOURCE = MyAzureStorage,  
FILE_FORMAT = MyAzureCSVFormat
```

```
)
```

```
CREATE STATISTICS stat_country_name  
ON dbo.Population(country_name);
```

```
SELECT
```

```
country_name, population
```

```
FROM population
```

```
WHERE year = 2019
```

```
ORDER BY population DESC
```

	country_name	population
1	China	1389618778
2	India	1311559204
3	United States	331883986
4	Indonesia	264935824
5	Pakistan	210797836
6	Brazil	210301591
7	Nigeria	208679114
8	Bangladesh	161062905
9	Russia	141944641
10	Mexico	127318112

Easy data transformation

Overview

Easily perform data transformations of Azure Storage files using SQL queries

Optimize data pipeline - achieve more using serverless SQL pool

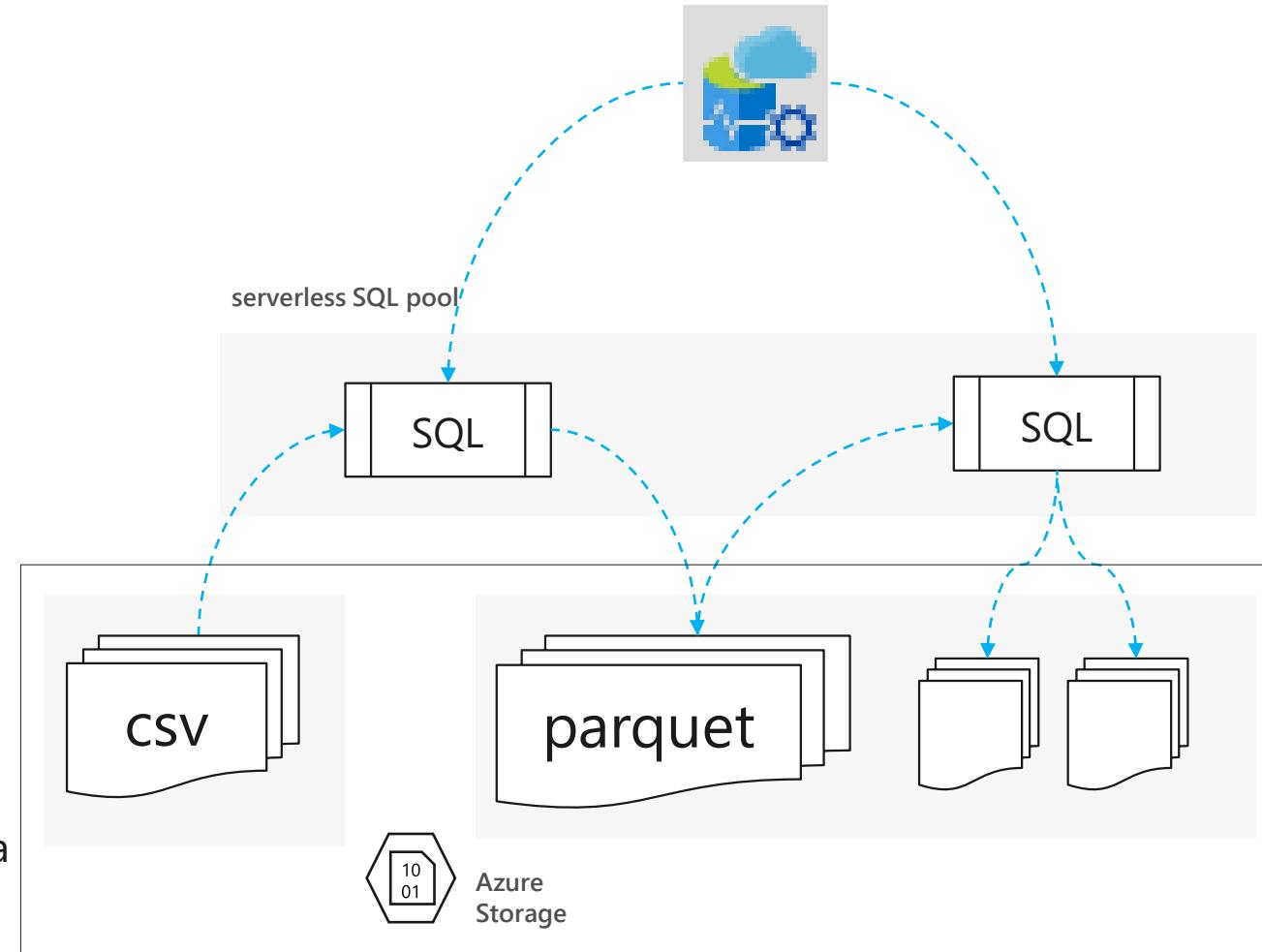
Benefits

Single statement transformations:

- convert CSV or JSON files to Parquet
- copy files from one storage account to another
- re-partition data to new location(s)
- store results of your query on Azure Storage

SQL ETL pipelines

- Use SQL commands to transform data
- Chain SQL statement for build ETL process
- Materialize reports created on the current snapshot of data



Easy data transformation with CETAS

Overview

Create external tables as select (CETAS) enables you to easily transform data and store the results of query on Azure storage

Benefits

Select any data set and store it in parquet format.

Pre-calculate and store results of query and store them permanently on Azure storage.

Use saved data using external table.

Improve performance of your reports by permanently storing the result based on current snapshot of data as parquet files.

```
-- copy CSV dataset into parquet data set
CREATE EXTERNAL TABLE parquet.Population
WITH(
    LOCATION = '/parquet/population',
    DATA_SOURCE = MyAzureStorage,
    FILE_FORMAT = MyAzureParquetFormat )
AS
SELECT *
FROM csv.Population

-- pre-create report using new parquet data-set
CREATE EXTERNAL TABLE parquet.PopulationByMonth2017
WITH(
    LOCATION = '/parquet/population/bymonth/2017',
    DATA_SOURCE = MyAzureStorage,
    FILE_FORMAT = MyAzureParquetFormat )
AS
SELECT month = p.month, population = COUNT ( p.population )
FROM parquet.Population p
WHERE p.year = 2017
GROUP BY p.month

-- Reporting tools can now directly read data from pre-created report
SELECT *
FROM parquet.PopulationByMonth2017
```

UI based data transformation

Synapse live Validate all Publish all 1

SQL script 10 default

New SQL script New notebook New data flow New integration dataset Upload Download More

default > Parquet

Name	Last Modified	Content Type
_SUCCESS	11/16/2020, 4:49:15 PM	
part-00000-5ae12a71-d27d-4e3a-a686-3bfb7d67c2c9-c000.snappy.parquet	11/16/2020, 4:49:14 PM	

New SQL script > Select TOP 100 rows
New notebook > Create external table
New data flow
New integration dataset
Manage access...
Rename...
Download
Delete
Properties...

Create external table

part-00000-5ae12a71-d27d-4e3a-a686-3bfb7d67c2c9-c000.snappy.parquet

External tables provide a convenient way to persist the schema of data residing in your data lake which can be reused for future adhoc analytics. [Learn more](#)

Select SQL pool * ⓘ

Built-in

Select a database * ⓘ

SQLServerlessDB

External table name * ⓘ

adls.retailsales1

Create external table *

Automatically

Using SQL script

This will include the create external table definition and the SELECT Top 100 in your SQL script. You will be required to run the SQL script to create the external table

Create **Cancel** [join meetup](#)

```
1  SELECT TOP 100 * FROM adls.retailsale
2  GO
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```

```
IF NOT EXISTS (SELECT * FROM sys.external_file_formats WHERE name = 'SynapseParquetFormat')
CREATE EXTERNAL FILE FORMAT [SynapseParquetFormat]
WITH ( FORMAT_TYPE = PARQUET)
GO

IF NOT EXISTS (SELECT * FROM sys.external_data_sources WHERE name = 'default_azureSynapseA_DFS_Core_Windows_Net')
CREATE EXTERNAL DATA SOURCE [default_azureSynapseA_DFS_Core_Windows_Net]
WITH (
    LOCATION = 'https://azuresynapsesa.dfs.core.windows.net/default',
)
GO

CREATE EXTERNAL TABLE adls.retailsale (
    [storeId] varchar(8000),
    [productCode] varchar(8000),
    [quantity] varchar(8000),
    [logQuantity] varchar(8000),
    [advertising] varchar(8000),
    [price] varchar(8000),
    [weekStarting] varchar(8000),
    [id] varchar(8000)
)
WITH (
    LOCATION = 'Parquet/part-00000-5ae12a71-d27d-4e3a-a686-3bfb7d67c2c9-c000.snappy.parquet',
    DATA_SOURCE = [default_azureSynapseA_DFS_Core_Windows_Net],
    FILE_FORMAT = [SynapseParquetFormat]
)
GO

SELECT TOP 100 * FROM adls.retailsale
```

Automatic syncing of Spark tables

Overview

Tables created in Spark pool are automatically created as external tables that reference external files in your serverless SQL pool logical data warehouse

Benefits

Tables designed using Spark languages are immediately available in serverless SQL pool.

Schema definition matches original

Spark table updates are applied in serverless SQL pool

No need to manually create SQL tables that match Spark tables

Spark and serverless SQL pool tables reference the same external files.

The screenshot shows the Azure Data Studio interface with two main panes. The left pane displays the 'Connections' sidebar with 'Servers' expanded, showing 'Sql on-demand', 'Databases', 'System Databases', 'default', 'Tables', and two external tables: 'dbo.data1013' and 'dbo.data1017'. The 'Columns' section under 'Tables' is selected. The right pane shows a 'Create external table' dialog with the following SQL code:

```
1 %%sql
2 create table data1017 using parquet
3 location 'abfss://container@demostorage.dfs.core.windows.net/data/'
```

Below this, a 'SQLQuery_1' query window is open with the following SQL:

```
1 SELECT TOP (10) [ExtractId]
2 , [DayOfWeekID]
3 , [DayOfWeekDescr]
4 , [DayOfWeekDescrShort]
5 , [ExtractDateTime]
6 , [LoadTS]
7 , [DeltaActionCode]
8 FROM [default]..[data1017]
```

The results pane shows the following data:

ExtractId	DayOfWeekID	DayOfWeekDescr	DayOfWeekDescrShort	ExtractDateT
6b86b273ff34fce19d6b804eff5a...	1	Sunday	Sun	2020-01-22 00:00:00.000
d4735e3a265e16eee03f50718b9b...	2	Monday	Mon	2020-01-22 00:00:00.000
4e07408562bedb8b60ce05c1aect...	3	Tuesday	Tue	2020-01-22 00:00:00.000
4b22777d4dd1fc61c6f884f4864...	4	Wednesday	Wed	2020-01-22 00:00:00.000
ef2d127de37b942baad06145e54b...	5	Thursday	Thu	2020-01-22 00:00:00.000
e7f6c011776e8db7cd330b54174f...	6	Friday	Fri	2020-01-22 00:00:00.000

Metastore

Overview

It offers the different computational engines of a workspace to share databases and Parquet-backed tables between its Apache Spark pools, serverless SQL pool, and dedicated SQL pool.

Benefits

- The shared metadata model supports the modern data warehouse pattern.
- The Spark created databases and all their tables become visible in any of the Azure Synapse workspace Spark pool instances and can be used from any of the Spark jobs provided necessary permissions are provided.
- Databases are created automatically in the serverless SQL pool metadata.
- The external and managed tables created by Spark job are made accessible as external tables in the serverless SQL pool metadata in the dbo schema of the corresponding database.
- Spark created databases and their Parquet-backed tables will be mapped into the SQL pools for which metadata synchronization enabled.

Transform with Spark

Transforming with Spark – Querying SQL Pools

Existing Approach

```
val jdbcUsername = "<SQL DB ADMIN USER>"  
val jdbcPwd = "<SQL DB ADMIN PWD>"  
val jdbcHostname = "servername.database.windows.net"  
val jdbcPort = 1433  
val jdbcDatabase = "<AZURE SQL DB NAME>"
```

```
val jdbc_url =  
  s"jdbc:sqlserver://${jdbcHostname}:${jdbcPort};database=${jdbcDatabase};  
  encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.databas  
e.windows.net;loginTimeout=60;"
```

```
val connectionProperties = new Properties()  
  
connectionProperties.put("user", s"${jdbcUsername}")  
connectionProperties.put("password", s"${jdbcPwd}")  
  
val sqlTableDf = spark.read.jdbc(jdbc_url, "dbo.Tbl1", connectionProperties)
```

New Approach Using Scala

```
// Construct a Spark DataFrame from SQL Pool table  
var df = spark.read.sqlanalytics("sql1.dbo.Tbl1")  
  
// Write the Spark DataFrame into SQL Pool table  
df.write.sqlanalytics("sql1.dbo.Tbl2")
```

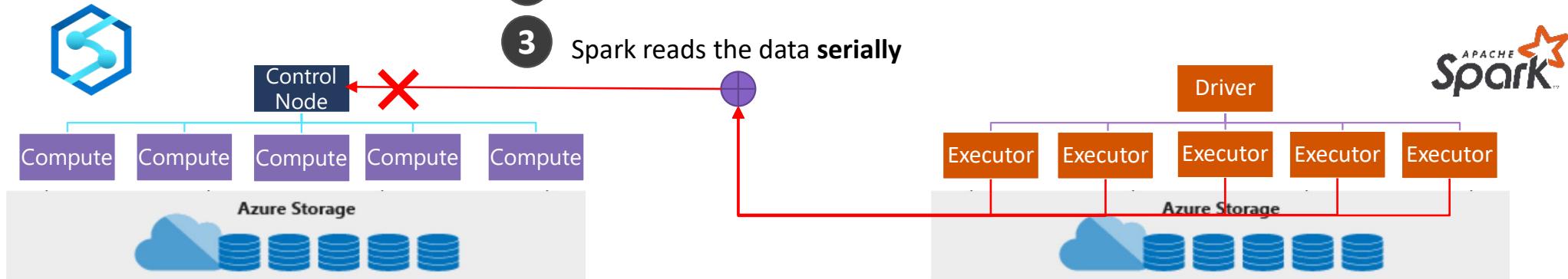
Using Python

```
%spark  
var df = spark.read.sqlanalytics("sql1.dbo.Tbl1")  
df.createOrReplaceTempView("tbl1")
```

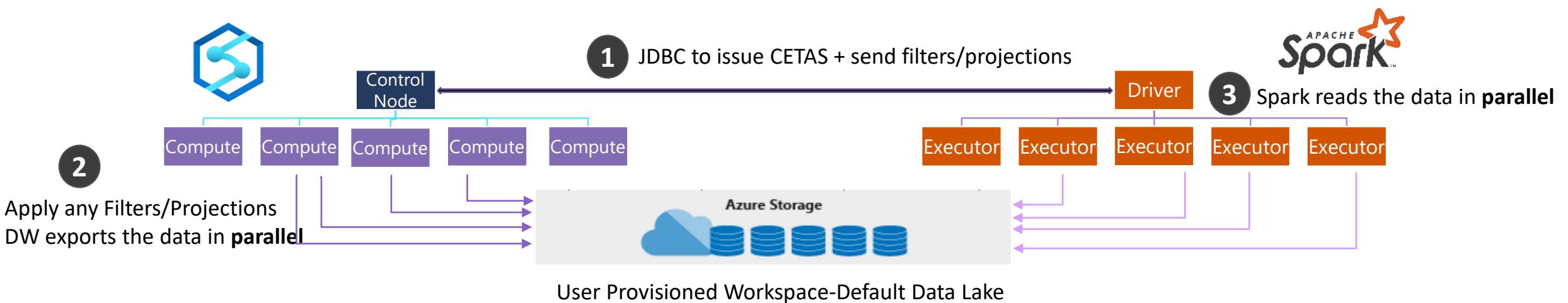
```
%pyspark  
sample = spark.sql("SELECT * FROM tbl1")  
sample.createOrReplaceTempView("tblnew")
```

```
%spark  
var df = spark.sql("SELECT * FROM tblnew")  
df.write.sqlanalytics("sql1.dbo.tbl2",  
  Constants.INTERNAL)
```

Existing Approach: JDBC



New Approach: JDBC and Polybase



Create Notebook on files in storage

The screenshot illustrates the process of creating a Notebook on files stored in Azure Storage. It consists of two main windows:

- Left Window (Storage Explorer):** Shows the Azure portal interface with the path: nyctic > green > puYear=2009 > puMonth=1. A context menu is open over a file named "part-00055...snappy.parquet". The "New notebook" option is highlighted with a red box and arrow.
- Right Window (Synapse Analytics):** Shows a Notebook titled "Notebook 4" running a PySpark job. The code in Cell 1 is:

```
[3] 1 %pyspark
2 data_path = spark.read.load('abfss://nyctic@prlangaddemosa.dfs.core.windows.net/yellow/puYear=2015/puMonth=3/part-00133-tid-210938564719836543-aea5b543-5e83-'
3 data_path.show(10)
```

The command was executed in 3mins 59s 249ms by prlangad on 11-14-2019 09:57:11.863 -08:00. The job execution status shows three tasks completed successfully.

Microsoft Azure Synapse Analytics > euang-synapse-nov-ws

Search resources

Publish all Validate all Refresh Discard all

Develop

Notebooks 13

- 00_DataPrep
- 01_TrainingUseMllib_cleanup
- automl_arclad_validate
- Data Download_GreenCab
- Data Download_HolidayData
- Data Download_Weather
- Data Download_YellowCab
- Explore_Join_Aggregate
- * NYCTaxi_Docs_Final
- NYCTaxi_Docs_Final_PySpark
- * Repro
- * SeattleSafetyDoc
- SparkPerf

NYCTaxi_Docs_Final * SeattleSafetyDoc * Repro

Cell 1

```
1 # Azure storage access info
2 blob_account_name = "azureopendatastorage"
3 blob_container_name = "citydatacontainer"
4 blob_relative_path = "Safety/Release/city=Seattle"
5 blob_sas_token = r""
6
7 # Allow SPARK to read from Blob remotely
8 wasbs_path = 'wasbs://{}@{}.blob.core.windows.net/{}'.format(blob_container_name, blob_account_name, blob_relative_path)
9 spark.conf.set('fs.azure.sas.{}.blob.core.windows.net'.format(blob_container_name), blob_sas_token)
10
11 # SPARK read parquet, note that it won't load any data yet
12 seasafety_df = spark.read.parquet(wasbs_path)
```

Command executed in 2mins 18s 412ms by euang on 11-22-2019 00:44:52.415 -08:00

Job execution In progress Spark 1 executors 4 cores

ID	DESCRIPTION	STATUS	STAGES	TASKS	SUBMISSION TIME	DURATION
Job 0	parquet at NativeMethodAccessImpl.java:0	In progress	0/1 (1 active)	100%	11/22/2019, 12:44:46 AM	9m54s

View in monitoring Spark history server

Cell 2

```
1 seasafety_df.createOrReplaceTempView('seattlesafety')
```

Command executed in 2s 835ms by euang on 11-22-2019 00:53:37.321 -08:00

Cell 3

```
[6] 1 display(spark.sql('SELECT * FROM seattlesafety LIMIT 10'))
```

Command executed in 23s 901ms by euang on 11-22-2019 00:54:07.313 -08:00

View Table Chart

dataType	dataSubtype	dateTime	category	address	latitude	longitude
Safety	911_Fire	2011-03-04T10:00:26.000Z	Aid Response	517 3rd Av	47.602172	-122.330863
Safety	911_Fire	2015-06-08T02:59:35.000Z	Trans to AMR	10044 65th Av S	47.511314	-122.252346
Safety	911_Fire	2015-06-08T21:10:52.000Z	Aid Response	Aurora Av N / N 125th St	47.719572	-122.344937
Safety	911_Fire	2007-09-17T13:03:34.000Z	Medic Response	1st Av N / Republican St	47.623272	-122.355415
Safety	911_Fire	2007-11-19T17:46:57.000Z	Aid Response	7724 Ridge Dr Ne	47.684393	-122.275254
Safety	911_Fire	2008-06-15T14:32:33.000Z	Medic Response	6940 62nd Av Ne	47.678789	-122.262227
Safety	911_Fire	2007-06-18T23:05:58.000Z	Medic Response	5107 S Myrtle St	47.538902	-122.268825
Safety	911_Fire	2005-06-06T19:23:10.000Z	Aid Response	532 Belmont Av E	47.623505	-122.324033
Safety	911_Fire	2017-03-06T19:45:36.000Z	Trans to AMR	610 1st Av N	47.624659	-122.355403
Safety	911_Fire	2017-06-23T18:21:21.000Z	Automatic Fire Alarm Resd	7711 8th Av Nw	47.685137	-122.366006

Cell 4

```
[7] 1 seasafety_df.coalesce(1).write.csv('abfss://default@euangsynapsenovstorage.dfs.core.windows.net/demodata/seattlesafety', mode='overwrite')
```

View results in table format



Microsoft Azure Synapse Analytics > euang-synapse-nov-ws

Search resources

Publish all Validate all Refresh Discard all

Develop Notebooks

NYCTaxi_Docs_Final * SeattleSafetyDoc * Repo * PySpark (Python)

Cell 1

```
[3] 1 # Azure storage access info  
2 blob_account_name = "azuresyndatastorage"  
3 blob_container_name = "citydatacontainer"  
4 blob_relative_path = "Safety/Release/city=Seattle"  
5 blob_sas_token = r""  
6  
7 # Allow SPARK to read from Blob remotely  
8 wasbs_path = 'wasbs://%' % (blob_container_name, blob_account_name, blob_relative_path)  
9 spark.conf.set( 'fs.azure.sas.%s.blob.core.windows.net' % (blob_container_name, blob_account_name), blob_sas_token)  
10  
11 # SPARK read parquet, note that it won't load any data yet  
12 seasafety_df = spark.read.parquet(wasbs_path)
```

Command executed in 2mins 18s 412ms by euang on 11-22-2019 00:44:52.415 -08:00

Job execution In progress Spark 1 executors 4 cores

ID	DESCRIPTION	STATUS	STAGES	TASKS	SUBMISSION TIME	DURATION
Job 0	parquet at NativeMethodAccessImpl.java:0	In progress	0/1 (1 active)		11/22/2019, 12:44:46 AM	13m43s

View in monitoring Spark history server

Cell 2

```
[5] 1 seasafety_df.createOrReplaceTempView('seattlesafety')
```

Command executed in 2s 835ms by euang on 11-22-2019 00:53:37.321 -08:00

Cell 3

```
[6] 1 display(spark.sql('SELECT * FROM seattlesafety'))
```

Command executed in 11s 526ms by euang on 11-22-2019 00:58:21.241 -08:00

SQL support

View Table Chart

Chart type pie chart X axis column category Y axis columns longitude Aggregation COUNT Y axis label Total X axis label category

Apply Cancel

Aid Response

Medic Response

Automatic Fire Alarm False

Medic Response, 7 per Rule

Aid Response Yellow

MVI - Motor Vehicle Incident

Medic Response, 6 per Rule

Motor Vehicle Accident

Automatic Medical Alarm

IRED 1 Unit

Auto Fire Alarm

Automatic Fire Alarm Resd

Trans to AMR

longitude

Cell 4

```
[7] 1 seasafety_df.coalesce(1).write.csv('abfss://default@euangsypnsestorage.dfs.core.windows.net/demodata/seattlesafety', mode='overwrite')
```

Microsoft Azure | Synapse Analytics > euang-synapse-nov-ws

Search resources

Publish all Validate all Refresh Discard all

Develop + <

Data Download... * NYCTaxi_Docs_...

Cell Run all Publish Attach to Select Spark pool Language PySpark (Python)

10
11 # Creating a temp table allows easier manipulation during the session, they are not persisted between sessions,
12 # for that write the data to storage like above.
13 sampled_taxi_df.createOrReplaceTempView("nytaxi")

Exploratory Data Analysis

Look at the data and evaluate its suitability for use in a model, do this via some basic charts focussed on tip values and relationships.

Cell 9

```
1 #The charting package needs a Pandas dataframe or numpy array do the conversion
2 sampled_taxi_pd_df = sampled_taxi_df.toPandas()
3
4 # Look at tips by amount count histogram
5 ax1 = sampled_taxi_pd_df['tipAmount'].plot(kind='hist', bins=25, facecolor='lightblue')
6 ax1.set_title('Tip amount distribution')
7 ax1.set_xlabel('Tip Amount ($)')
8 ax1.set_ylabel('Counts')
9 plt.suptitle('')
10 plt.show()
11
12 # How many passengers tip'd by various amounts
13 ax2 = sampled_taxi_pd_df.boxplot(column=['tipAmount'], by=['passengerCount'])
14 ax2.set_title('Tip amount by Passenger count')
15 ax2.set_xlabel('Passenger count')
16 ax2.set_ylabel('Tip Amount ($)')
17 plt.suptitle('')
18 plt.show()
19
20 # Look at the relationship between fare and tip amounts
21 ax = sampled_taxi_pd_df.plot(kind='scatter', x= 'fareAmount', y = 'tipAmount', c='blue', alpha = 0.10, s=2.5*(sampled_taxi_pd_df['passengerCount']))
22 ax.set_xlabel('Fare Amount ($)')
23 ax.set_ylabel('Tip Amount ($)')
24 plt.axis([-2, 80, -2, 20])
25 plt.suptitle('')
26 plt.show()
27
```

Tip amount distribution

Tip amount by Passenger count

Exploratory data analysis with graphs – histogram, boxplot etc

Best practices

Serverless SQL Pools

- Co-locate storage and serverless SQL pools
- Consider Azure Storage throttling
- Prepare files for querying (CSV, JSON -> Parquet)
- Push wildcards to lower levels in the path
- Use appropriate data types and check inferred data types
- Use filename and filepath functions to target specific partitions
- Use PARSE VERSION 2.0 to query CSV files
- Use CETAS to enhance query performance and joins
- Choose SAS credentials over Azure AD pass-through (for now)

CCI vs Heap

- Transformations using Heap tables are generally faster than CCI. This is because rows need to be assembled from column stores on read tables, and columnar compression is needed on targets.
- The wider the table, and the more text fields it contains, the faster Heap is over CCI.
- Use Heap tables at transformation layer, use CCI tables where appropriate at presentation layer

CCI Best Practice

- MAX data types not supported
- At least 1 million rows * 60 distributions * number of partitions
- At least 100k rows per batch, up to 1million
- Load using at least LARGERC or STATICCRC60
 - Create a loading user
- Minimal UPDATE and DELETE (or REBUILD frequently)

Automatic statistics management – Dedicated SQL

Overview

Statistics are automatically created and maintained for dedicated SQL pool. Incoming queries are analyzed, and individual column statistics are generated on the columns that improve cardinality estimates to enhance query performance.

Statistics are automatically updated as data modifications occur in underlying tables. By default, these updates are synchronous but can be configured to be asynchronous.

Statistics are considered out of date when:

- There was a data change on an empty table
- The number of rows in the table at time of statistics creation was 500 or less, and more than 500 rows have been updated
- The number of rows in the table at time of statistics creation was more than 500, and more than $500 + 20\%$ of rows have been updated

-- Turn on/off auto-create statistics settings

```
ALTER DATABASE {database_name}
```

```
SET AUTO_CREATE_STATISTICS { ON | OFF }
```

-- Turn on/off auto-update statistics settings

```
ALTER DATABASE {database_name}
```

```
SET AUTO_UPDATE_STATISTICS { ON | OFF }
```

-- Configure synchronous/asynchronous update

```
ALTER DATABASE {database_name}
```

```
SET AUTO_UPDATE_STATISTICS_ASYNC { ON | OFF }
```

-- Check statistics settings for a database

```
SELECT      is_auto_create_stats_on,  
            is_auto_update_stats_on,  
            is_auto_update_stats_async_on  
FROM        sys.databases
```

Statistics (serverless SQL)

- Automatic creation available only for Parquet and CSV support
- Same goes for recreation of statistics
- Only single-column statistics are currently supported
- CSV sampling not supported yet (only FULLSCAN)

CTAS vs Insert / Update / Delete / Merge

- Prefer CTAS when you update or delete more than 10% of rows
- Prefer CTAS when you are updating or deleting a clustered Columnstore index, and do not have time for an offline rebuild

UPDATE FROM and DELETE FROM

- Azure Synapse Analytics does not currently support (*) joins in UPDATE FROM and DELETE FROM queries.
- Implement the join as a temporary / transient table, then UPDATE / DELETE from that table
- (*) Coming soon

Simple is better than clever

- Persist standard columns early, to avoid calculations and functions in WHERE clause
- Unroll CTEs and JOIN sub-selects to transient / temporary tables to manage distribution
- Simple queries are easier to tune and debug

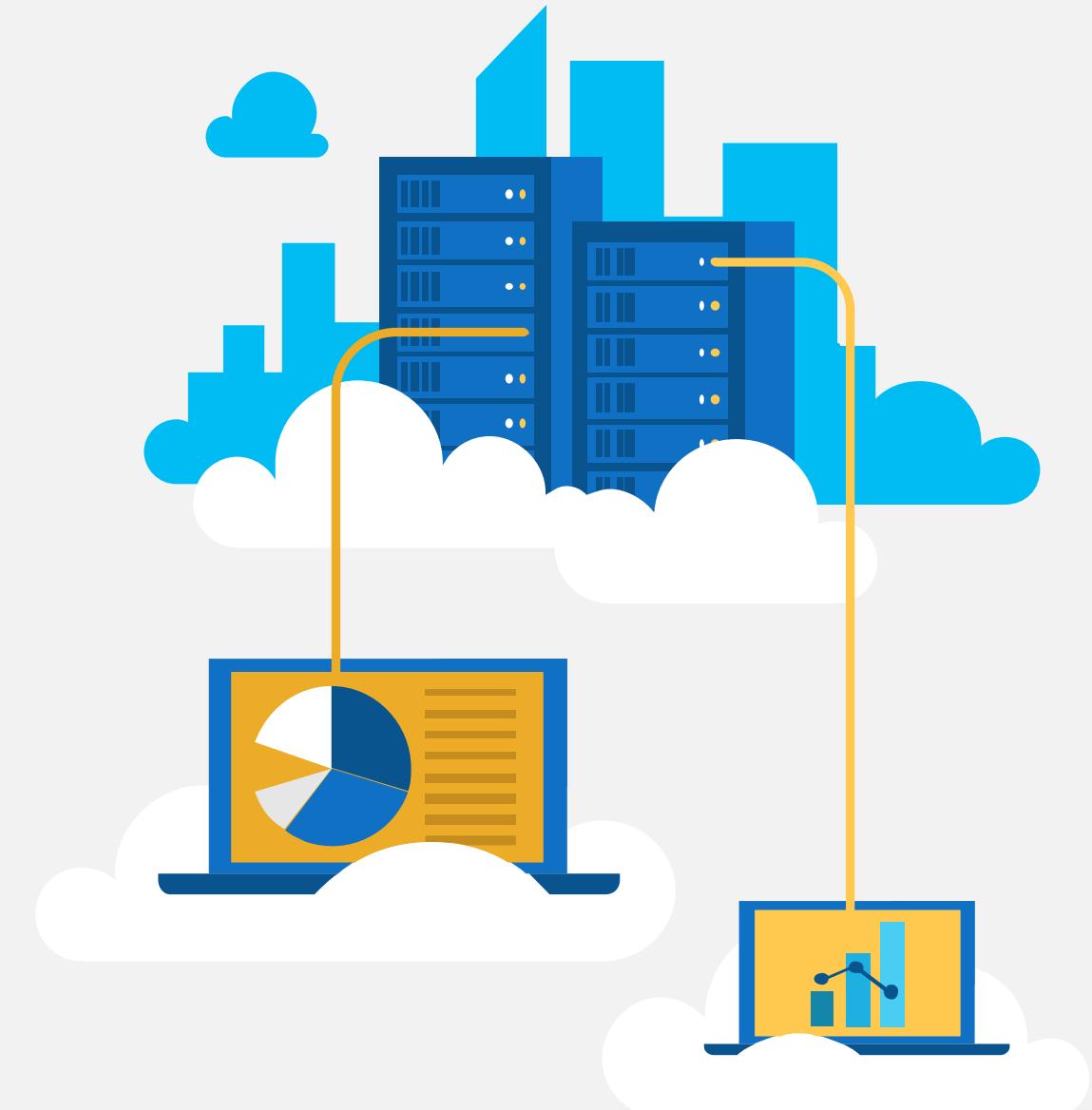
Pop Quiz 2

What is the optimal size for a rowgroup in columnstore format in a Synapse SQL Pool?

A)
99,999

B)
60,000,000

C)
1,048,576



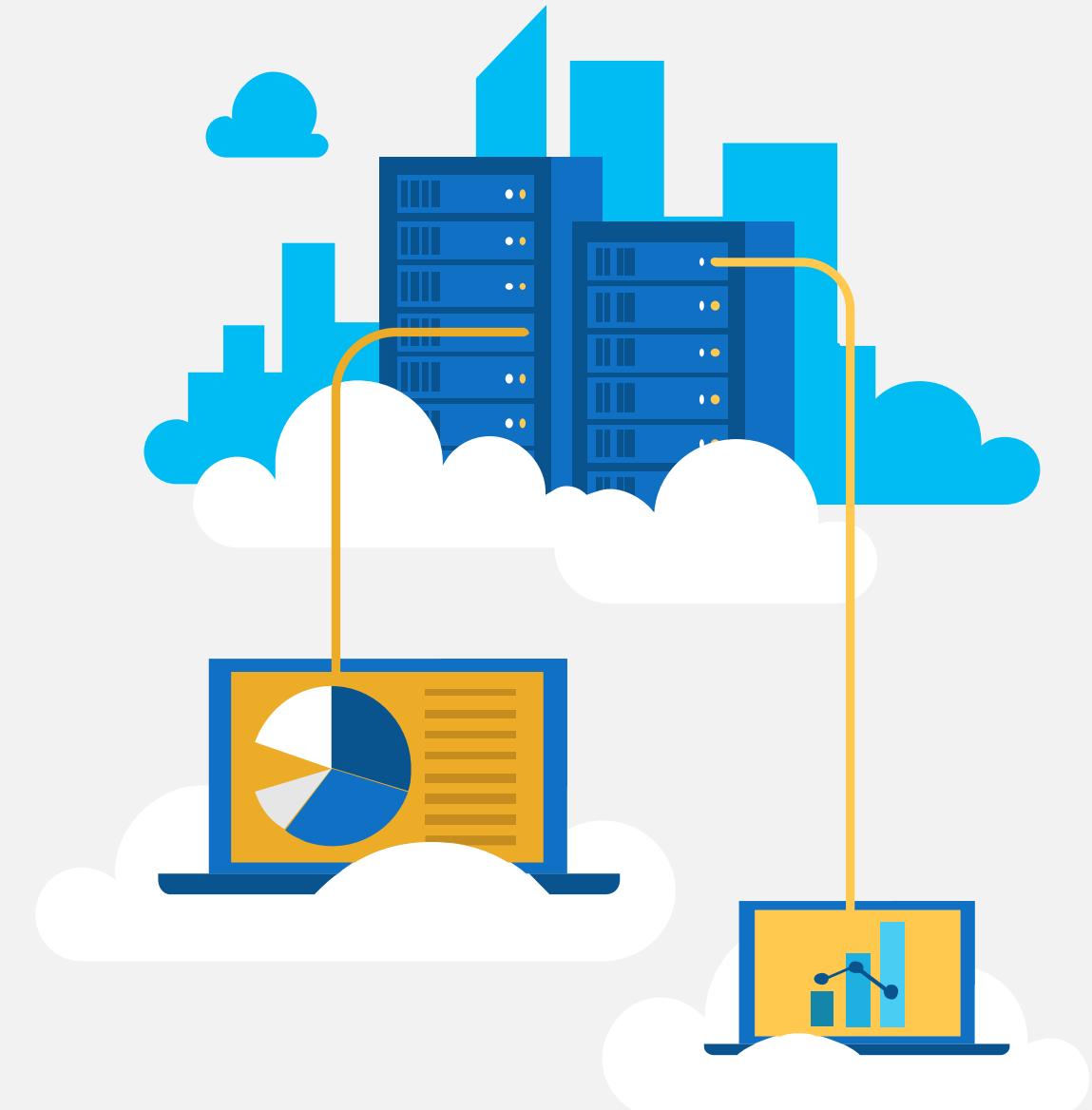
Pop Quiz 2

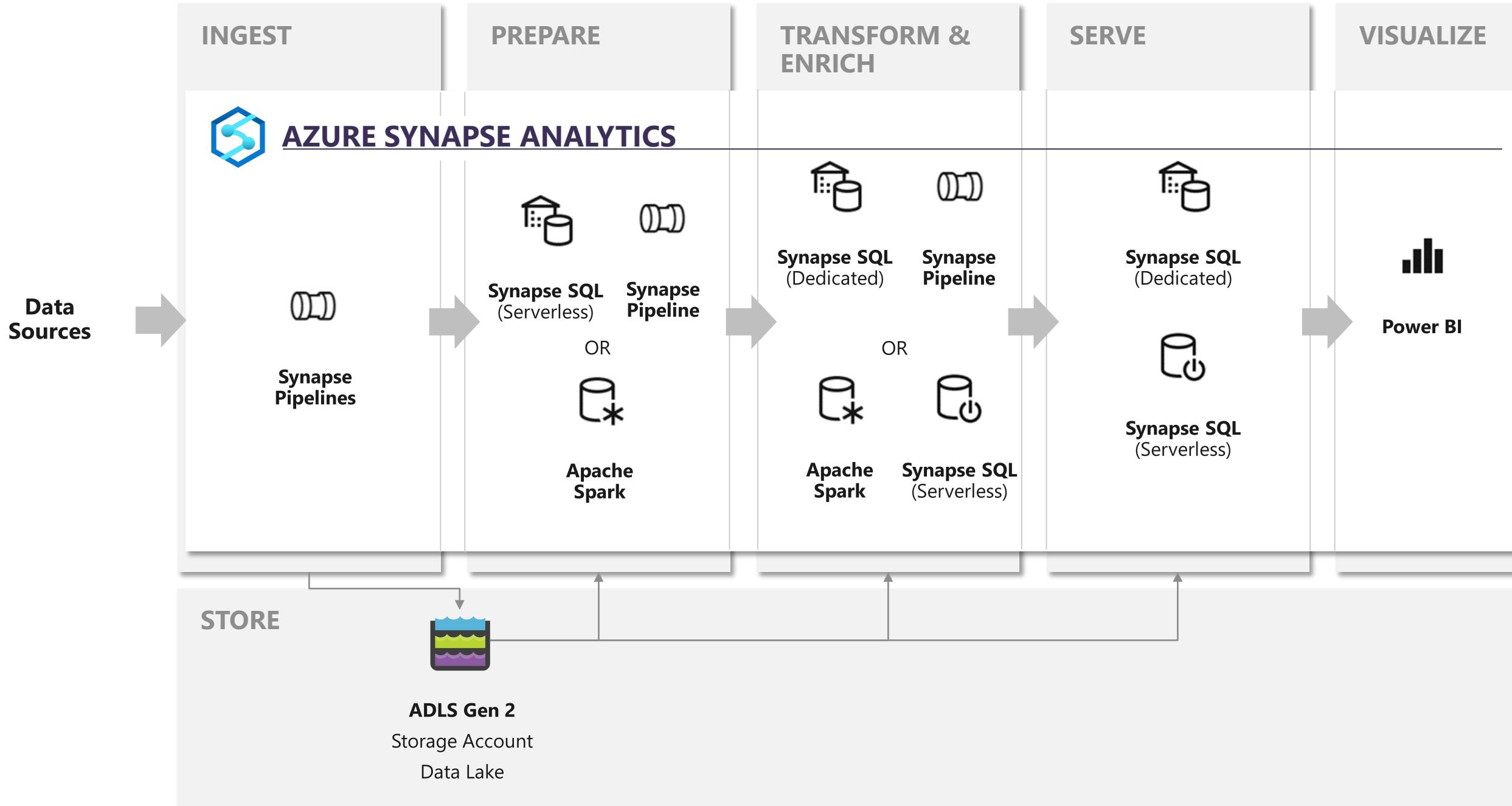
What is the optimal size for a rowgroup in columnstore format in a Synapse SQL Pool?

A)
99,999

B)
60,000,000

C)
1,048,576







Thank you



Full Group Activity: Data Engineering Discussion

Objective: As a result of participating in this activity, you will better be able to decide on which Azure Synapse Analytics component to use for specific data engineering scenarios.

What you will be doing: The facilitator will be posting questions to the entire group using an interactive tool called Mentimeter. The answers you post using Mentimeter will then drive the Data Engineering Discussion.

Total Activity Time: 30 minutes



Mentimeter Poll

Scan QR Code

or

Go to www.menti.com and use code

75 71 00



Hands-On Build Activity: Data Integration Part 2

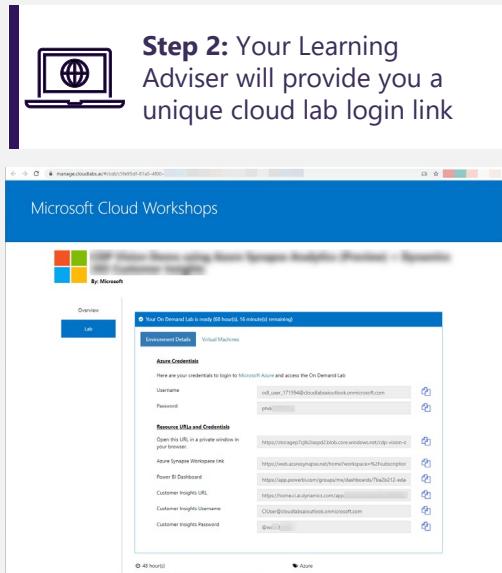
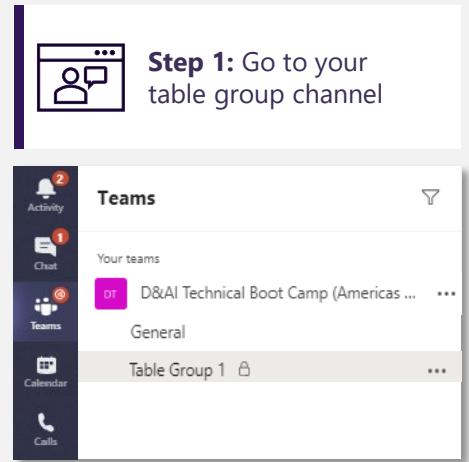
Objective: As a result of participating in this lab, you will be better able to build data pipelines that integrate disparate data sources and handle poorly formatted data during ingestion.

What you will be doing: Like the Part 1 exercise, you **work individually in your lab environment to complete a set of tasks** you would typically follow in work associated with ingesting and integrating your customer's data. A set of tasks are outlined for you to follow and complete.

Total Activity Time: 60 minutes



Lab: individual lab exercise



The Microsoft Synapse workspace interface shows the 'Data Integration Part 1' lab environment. It includes sections for 'Import', 'Explore', 'Analyze', and 'Visualize'. The 'Resources' section shows recent and printed resources. The 'Useful links' section provides links to documentation and feedback forms. The 'Data Integration Part 1' section lists various tasks and exercises related to data integration.

Step 1: Go to your table group channel

Step 2: Your Learning Adviser will provide you a unique cloud lab login link

Step 3: Login and begin the lab. Engage your table group via a Microsoft Teams "meet now" call to support each other (meet now button in upper right corner of application)

Closing

Thank you for your participation in today's Azure Synapse Technical Boot Camp!

TODAY

We learned:

- ✓ Best practices for rapid and reliable data ingestion into a Data Warehouse
- ✓ A well architected data lake is built upon scalable and secure partitioning structure
- ✓ Established best practices for data transformation within the data engineering pipeline in order to efficiently go from raw to structured data for analysis

TOMORROW

We will learn to:

- Implement optimization strategies for data warehouse using SQL
- Apply security concepts to a customer scenario

