



Welcome to Azure Synapse Technical Boot Camp

Day 1

A look into Day 1

Kick-Off	6:30-6:35	Welcome	Main Call
	6:35-7:00	Azure Synapse Analytics 101	
	7:00-7:15	Demo Walkthrough	
Ingest	7:15-7:30	Break	Table Group Call
	7:30-8:30	Data Loading & Data Lake Organization	
	8:30-9:30	Activity: Data Lake Design & Security Considerations	
	9:30-9:45	Break	
	9:45-10:30	Build Hands-on: Data Integration Part 1	
Transform	10:30-11:30	Break	Main Call
	11:30-12:00	Data Transformations	
	12:00-12:30	Activity: Data Engineering Discussion	Table Group Call
	12:30-1:30	Build Hands-on: Data Integration Part 2	
		End of Day 1	

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 workspace interface. The top navigation bar includes 'Microsoft Azure', 'Synapse Analytics', 'wsazuresynapseanalytics', and a user account 'someone@microsoft.com'. The left sidebar has a 'Home' icon selected, along with 'Data', 'Develop', 'Integrate', 'Monitor', and 'Manage' options. The main content area is titled 'Synapse workspace' and 'wsazuresynapseanalytics'. It features a 'New' button and four primary action cards: 'Ingest' (cloud icon), 'Explore and analyze' (3D bar chart icon), 'Visualize' (bar chart icon), and 'Learn' (book icon). A large, stylized 3D bar chart visualization is displayed in the background. Below these cards, the 'Recent resources' section lists six 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

A 'Show more ▾' link is at the bottom of the recent resources list.

Synapse Studio

Synapse Studio divided into **Activity hubs**.

These organize the tasks needed for building analytics solution.

The screenshot shows the Microsoft Synapse Studio interface. On the left, a vertical sidebar menu is highlighted with a red border. It contains the following items: Home, Data, Develop, Integrate, Monitor, and Manage. A red arrow points from the 'Integrate' item in this sidebar to the 'Integrate' hub icon on the main page. The main content area is titled 'Synapse workspace' and shows five activity hubs: Home, Data, Develop, Monitor, and Manage. Each hub has a brief description and an associated icon. The 'Home' hub is currently selected.

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.

Monitor
Centralized view of all resource usage and activities in the workspace.

Manage
Configure the workspace, pool, linked service, access to artifacts

Integrate
Design pipelines that move and transform data.

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 left sidebar includes links for Home, Data, Develop, Integrate, Monitor, and Manage. The main area displays a 'Synapse workspace' titled 'wsazuresynapseanalytics'. It features a 'New' button and five cards: Ingest, Explore and analyze, Visualize, and Learn. A 'Recent resources' section lists four items. The top right corner contains five icons: a speaker (Updates), a document (Switch Workspaces), a bell (Notifications), a speech bubble (Feedback), and a question mark (Knowledge Center). Red boxes and arrows highlight these five features.

Updates

Switch Workspaces

Notifications

Feedback

Knowledge Center

someone@microsoft.com MICROSOFT

wsazuresynapseanalytics

Updates

Switch Workspaces

Notifications

Feedback

Knowledge Center

Ingest

Explore and analyze

Visualize

Learn

Recent resources

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 features a 'Synapse workspace' header and a 'New' button. 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.). A red box highlights the 'Ingest', 'Explore and analyze', 'Visualize', and 'Learn' cards. The 'Recent resources' section lists five items: '05 Sentiment_Analysis_Cognitive_Services' (Last opened 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), and '05 Anomaly_Detection_Cognitive_Services' (a day ago). A 'Show more' button is at the bottom.

Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics

someone@microsoft.com MICROSOFT

Synapse workspace
wsazuresynapseanalytics

New

Ingest

Explore and analyze

Visualize

Learn

Recent resources

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

Show more ▾

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

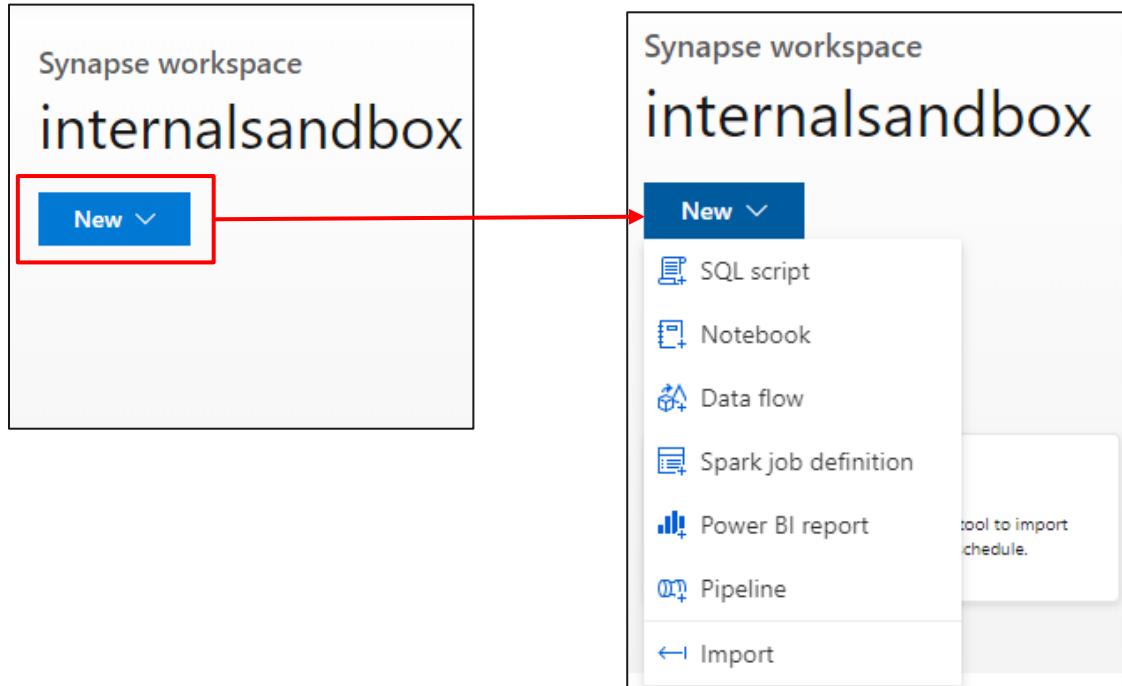
Continue [Close](#)

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 page for the workspace 'wsazuresynapseanalytics'. The left sidebar includes Home, Data (selected), Develop, Integrate, Monitor, and Manage. The main area has tabs for 'Data' (selected) and 'Linked'. Under 'Data', there is a 'Workspace' tab with a red box around it, and a 'Linked' tab. A search bar says 'Filter resources by name'. Below are sections for Databases (10 items) and Integration (24 items). The databases listed are: newpoll (SQL), NYCTaxi_Pool (SQL), Predict_Pool (SQL), Streaming_Pool (SQL), WWI_Pool (SQL), NYT2020 (SQL), SQLServerlessDB (SQL), and default (Spark).

Databases	Count
newpoll (SQL)	1
NYCTaxi_Pool (SQL)	1
Predict_Pool (SQL)	1
Streaming_Pool (SQL)	1
WWI_Pool (SQL)	1
NYT2020 (SQL)	1
SQLServerlessDB (SQL)	1
default (Spark)	1

Integration datasets	Count
wsazuresynapseanalytics (Primary...)	3
(Attached Containers)	1
Integration datasets	24

Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics

The screenshot shows the Microsoft Azure Synapse Analytics Data page for the workspace 'wsazuresynapseanalytics'. The left sidebar includes Home, Data (selected), Develop, Integrate, Monitor, and Manage. The main area has tabs for 'Data' (selected) and 'Linked'. Under 'Linked', there is a 'Workspace' tab with a red box around it, and a 'Linked' tab. A search bar says 'Filter resources by name'. Below are sections for Azure Blob Storage (3), Azure Cosmos DB (1), Azure Data Explorer (2), Azure Data Lake Storage Gen2 (2), and Integration datasets (24). The integration datasets listed are: wsazuresynapseanalytics (Primary...) and (Attached Containers).

Azure Blob Storage	Count
wsazuresynapseanalytics (Primary...)	3

Azure Cosmos DB	Count
(Attached Containers)	1

Azure Data Explorer	Count
Integration datasets	2

Azure Data Lake Storage Gen2	Count
wsazuresynapseanalytics (Primary...)	2

Integration datasets	Count
(Attached Containers)	24

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 file path navigation bar: rawdata > taxidata. Below it is a table listing files in the 'rawdata' container:

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 Storage account. On the left, the 'Data' blade shows a list of resources, including 'Linked' services like Azure Blob Storage, Azure Cosmos DB, and Azure Data Explorer. In the center, the 'rawdata' storage account is selected, displaying its contents. A red arrow points from the 'Preview' option in the context menu of the 'Products.csv' file to the preview pane on the right. The preview pane shows the first few rows of the CSV file, which contains product information.

Products.csv

Path https://azuresynapsesa.dfs.core.windows.net/rawdata/sample csv files/Products.csv
Modified 10/27/2020, 8:38:51 PM

With column header On

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

OK

Data Hub – Storage accounts

See basic file properties

The screenshot illustrates the process of viewing basic file properties in the Azure Data Hub. On the left, the 'Data' workspace shows a list of resources, including 'Azure Blob Storage', 'Azure Cosmos DB', 'Azure Data Explorer', 'Azure Data Lake Storage Gen2', and 'Integration datasets'. In the center, a detailed view of the 'rawdata' folder under 'wsazuresynapseanalytics (Primary)' is displayed. A context menu is open over the 'Products.csv' file, with the 'Properties...' option highlighted by a red box. A red arrow points from this highlighted item to the right-hand 'Properties' dialog box.

Properties

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

Content Encoding

Content Language

User Properties

Apply Cancel

Data Hub – Storage accounts

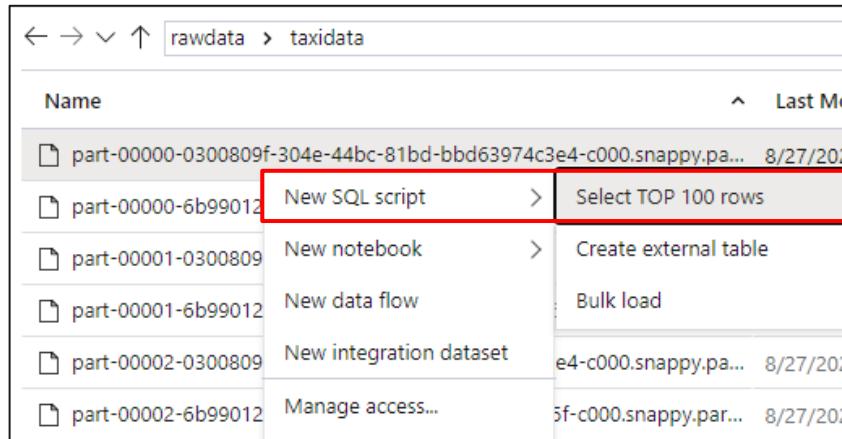
Manage Access - Configure standard POSIX ACLs on files and folders

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 resources, including 'rawdata' under 'wsazuresynapseanalytics'. In the center, the 'rawdata' folder is selected, displaying a list of files: 'Products.csv', 'Preview', 'New SQL script', 'New notebook', 'New data flow', 'New integration dataset', 'Manage access...', 'Rename...', 'Download', 'Delete', and 'Properties...'. A red arrow points from the 'Manage access...' option in the context menu to the 'Manage Access' dialog box on the right. The 'Manage Access' dialog shows the current users with permissions: '\$superuser (Owner)' and '\$superuser (Owning Group)'. It also includes sections for 'Permissions for: \$superuser' (with checkboxes for Read, Write, and Execute) and 'Add user, group, or service principal'.

Data Hub – Storage accounts

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

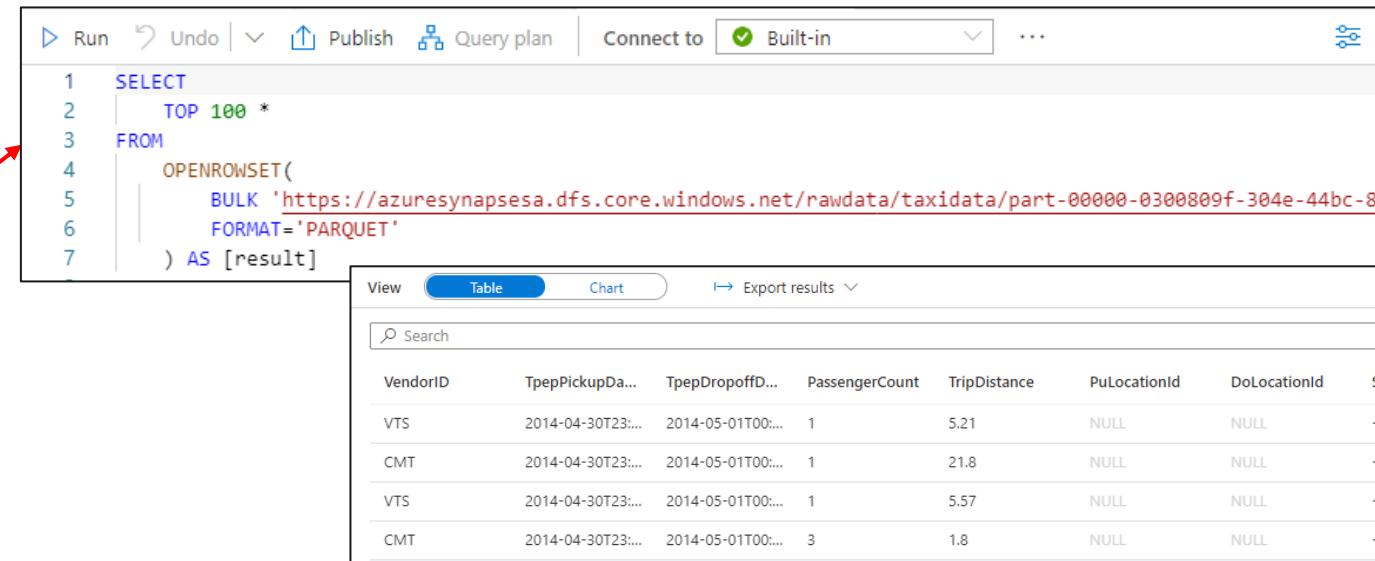
T-SQL or PySpark auto-generated.



rawdata > taxidata

Name

- part-00000-0300809f-304e-44bc-81bd-bbd63974c3e4-c000.snappy.parquet 8/27/2022
- part-00000-6b99012 New SQL script > Select TOP 100 rows
- part-00001-0300809 New notebook > Create external table
- part-00001-6b99012 New data flow Bulk load
- part-00002-0300809 New integration dataset e4-c000.snappy.parquet 8/27/2022
- part-00002-6b99012 Manage access...

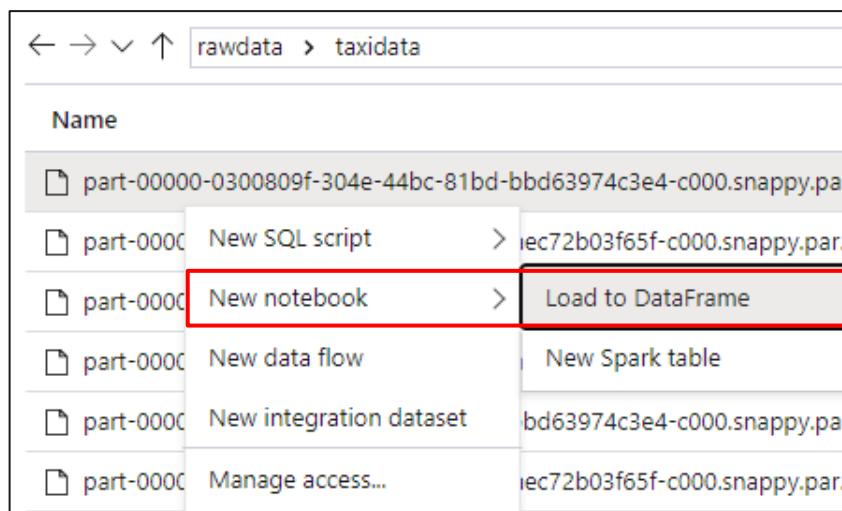


Run Undo Publish Query plan Connect to Built-in ...

```
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

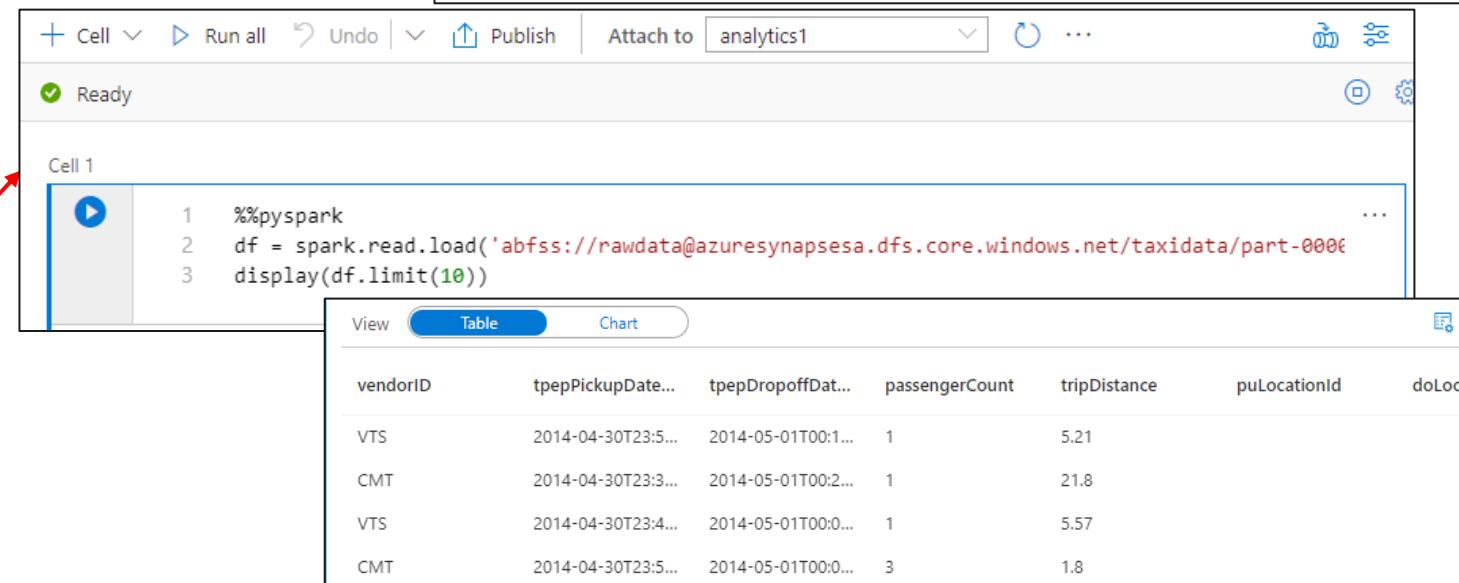
VendorID	TpepPickupDate	TpepDropoffDate	PassengerCount	TripDistance	PuLocationId	DoLocationId	...
VTS	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	1	5.21	NULL	NULL	-
CMT	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	1	21.8	NULL	NULL	-
VTS	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	1	5.57	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
- part-00000-6b99012 New SQL script > iec72b03f65f-c000.snappy.parquet
- part-00000-6b99012 New notebook > Load to DataFrame
- part-00000-6b99012 New data flow New Spark table
- part-00000-6b99012 New integration dataset bd63974c3e4-c000.snappy.parquet
- part-00000-6b99012 Manage access...



+ 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	doLocationId	...
VTS	2014-04-30T23:59:59.998Z	2014-05-01T00:00:00.000Z	1	5.21			-
CMT	2014-04-30T23:59:59.998Z	2014-05-01T00:00:23.000Z	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)' containing 'Tables' (with 'dbo.SearchLogTable' expanded) and 'Columns' (with 'dbo.NycTaxiPredict' expanded). Under 'Columns' for 'dbo.NycTaxiPredict', there's a context menu with options: 'New SQL script' (highlighted), 'Select TOP 1000 rows', 'CREATE', 'DROP', and 'DROP and CREATE'. The 'New SQL script' option has a tooltip 'Generate T-SQL script from column metadata'.

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. It shows the same tree view of databases and tables as the first screenshot. A context menu is open over the 'columns' section of the 'dbo.NycTaxiPredict' table. The 'New SQL script' option is highlighted with a red box. A red arrow points from this menu down to a 'Notebook 1' window at the bottom. The notebook window has tabs for 'Cell', 'Run all', 'Publish', 'Attach to', 'Select Spark pool', 'Language' (set to 'PySpark (Python)'), and a code cell labeled 'Cell 1' containing the PySpark command: `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 Studio interface for managing datasets. On the left, a sidebar titled 'Data' lists resources: Storage accounts (2), Databases (3), and Datasets (2). The 'NYCTaxiParquet' dataset is highlighted with a red box and has a red arrow pointing from the sidebar to its main view. The main view displays the dataset details under the 'Connection' tab. The dataset is identified as 'Parquet' type and named 'NYCTaxiParquet'. The 'Connection' tab shows the linked service as 'Lake_ArcadiaLake', and the file path is set to 'data / nyctaxi / File'. The compression type is set to 'snappy'. Other tabs available are General, Schema, and Parameters. A 'Code' button is visible in the top right corner.

Develop Hub

Overview

It provides development experience to query, analyze, model data

Benefits

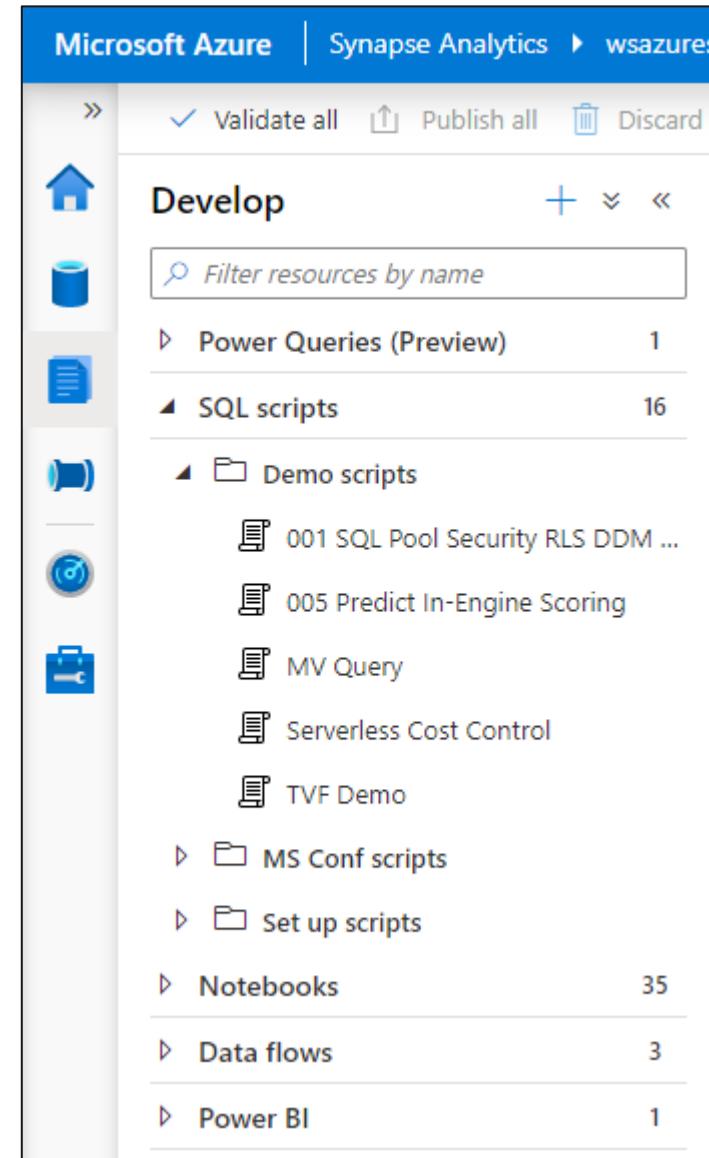
Multiple languages to analyze data under one umbrella

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

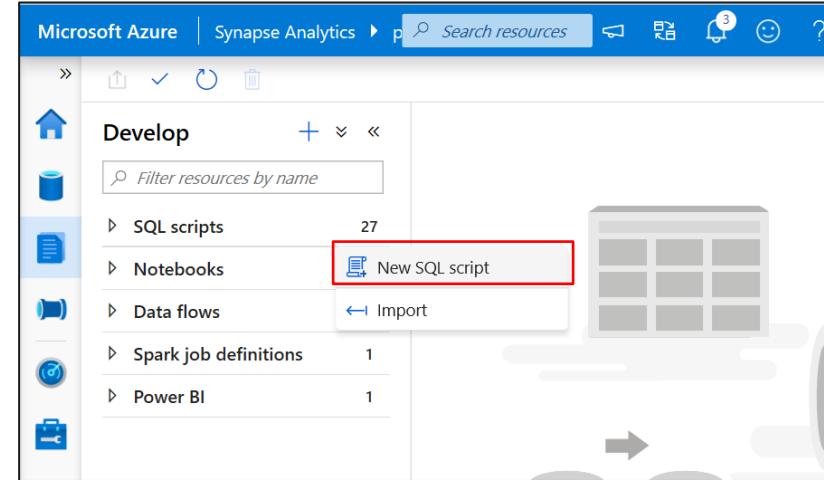
SQL Script

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

A screenshot of the Microsoft Azure Synapse Analytics Develop Hub showing an open SQL script named 'SQL script 2'. The script contains the following code:

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

A dropdown menu is open at the bottom right, showing Intellisense suggestions for the word 'CREATE':

- CREATE
- CURRENT_TIMESTAMP
- CURRENT_USER

The top right corner shows the user's email (prlangad@microsoft.com) and Microsoft logo.

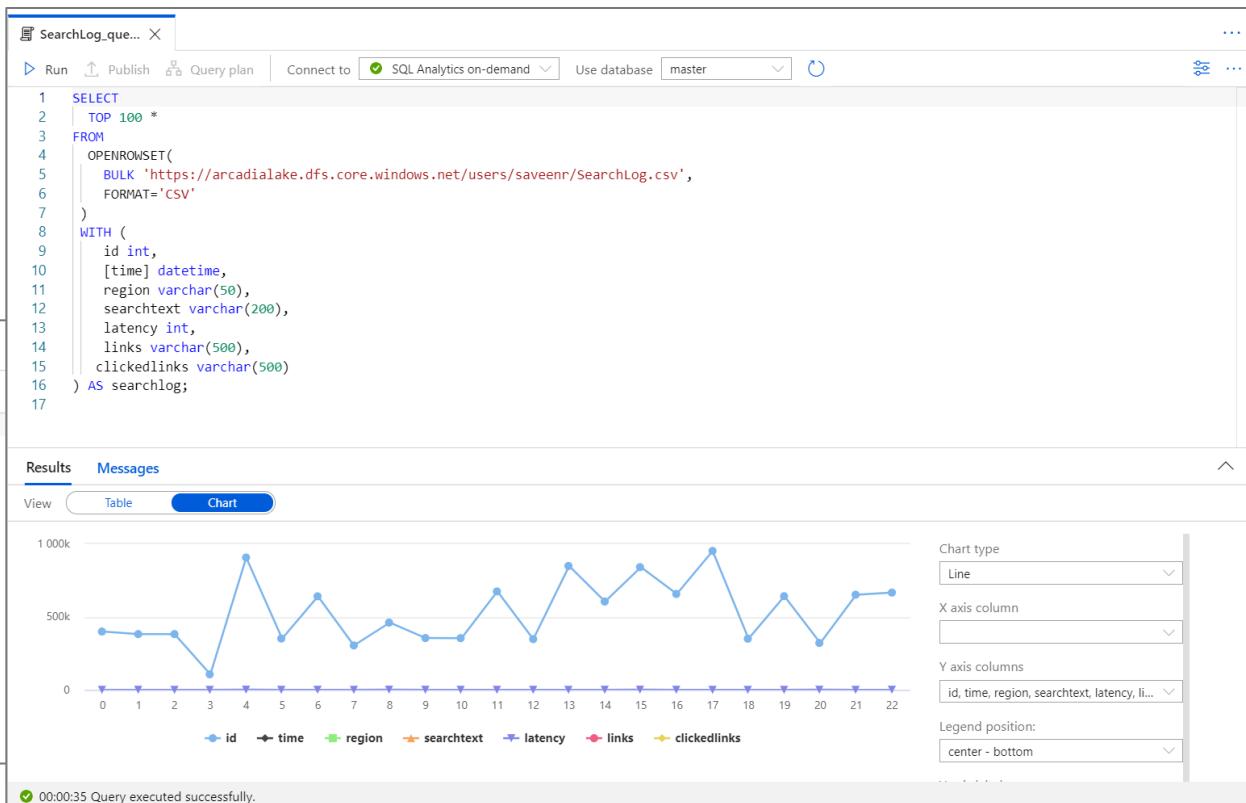
Develop Hub - SQL scripts

SQL Script

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

The screenshot shows the Azure Data Studio interface. On the left, a code editor window titled "SearchLog_que..." displays a T-SQL script for reading CSV data from a blob storage location and creating a temporary table named "searchlog". The script uses OPENROWSET and BULK options. Below the code editor is a results pane with tabs for "Results" and "Messages". The "Results" tab is selected, showing a table with columns ID, TIME, and REGION. The table contains five rows of data. At the bottom of the results pane, there is a message: "00:00:35 Query executed successfully.". To the right of the results pane is a "Chart" tab, which is currently inactive. At the bottom right of the results pane, there is a red box highlighting the "Export results" button, which has a dropdown menu showing options: CSV, Excel, JSON, and XML.

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



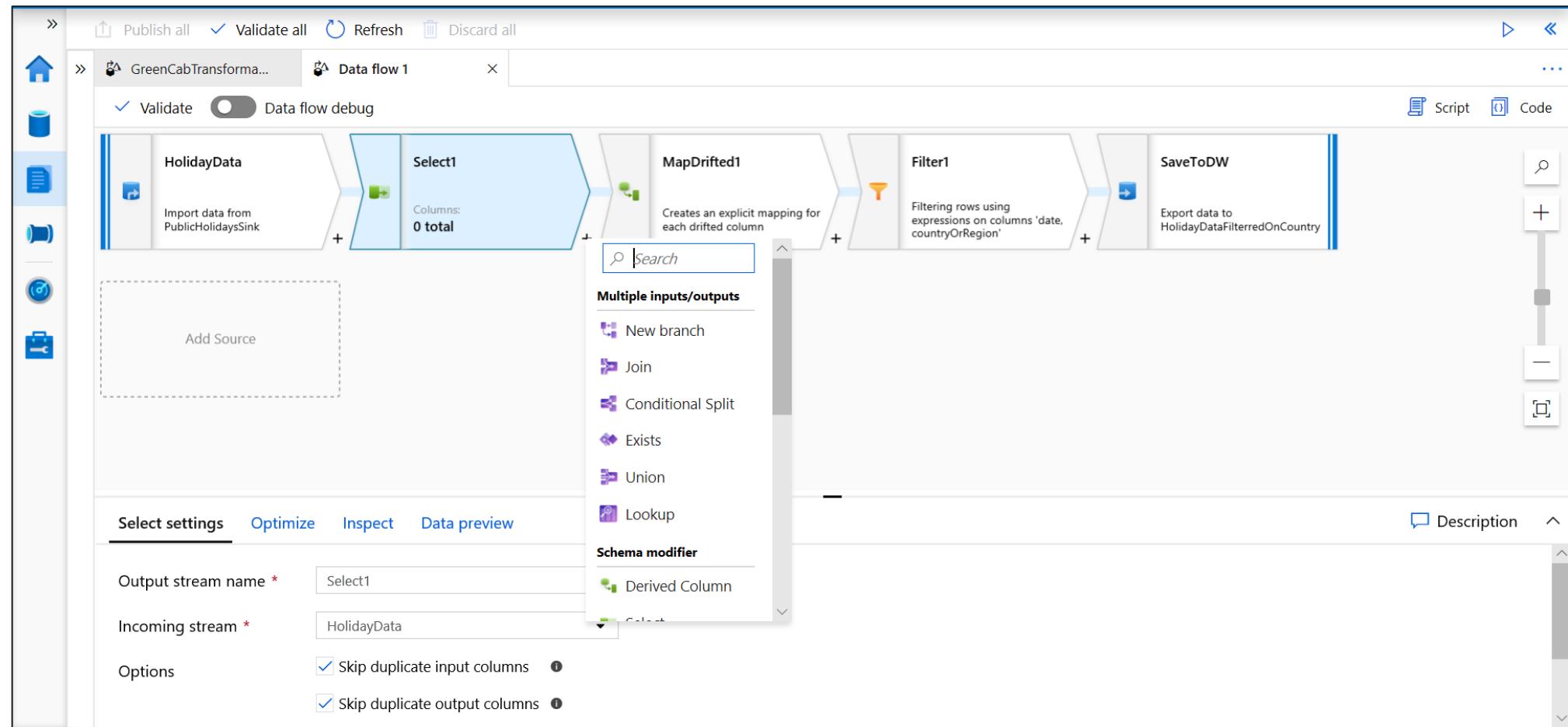
Develop Hub - Notebooks

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

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

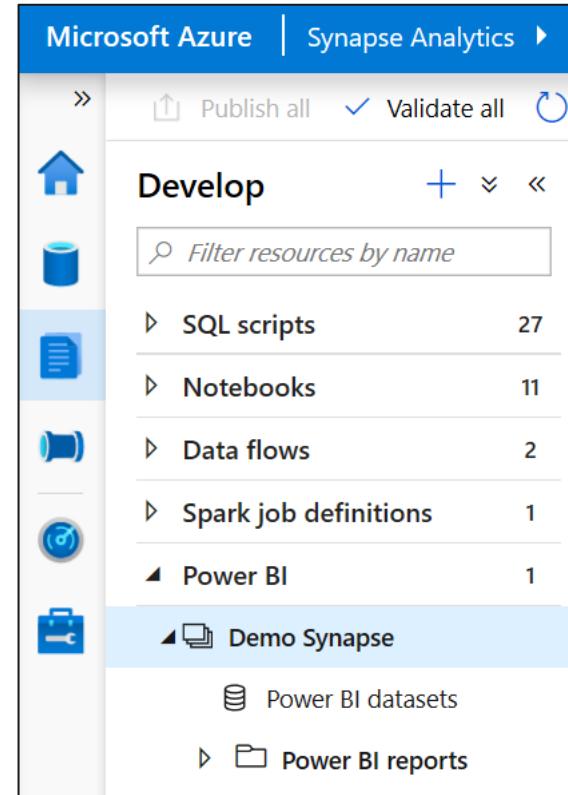
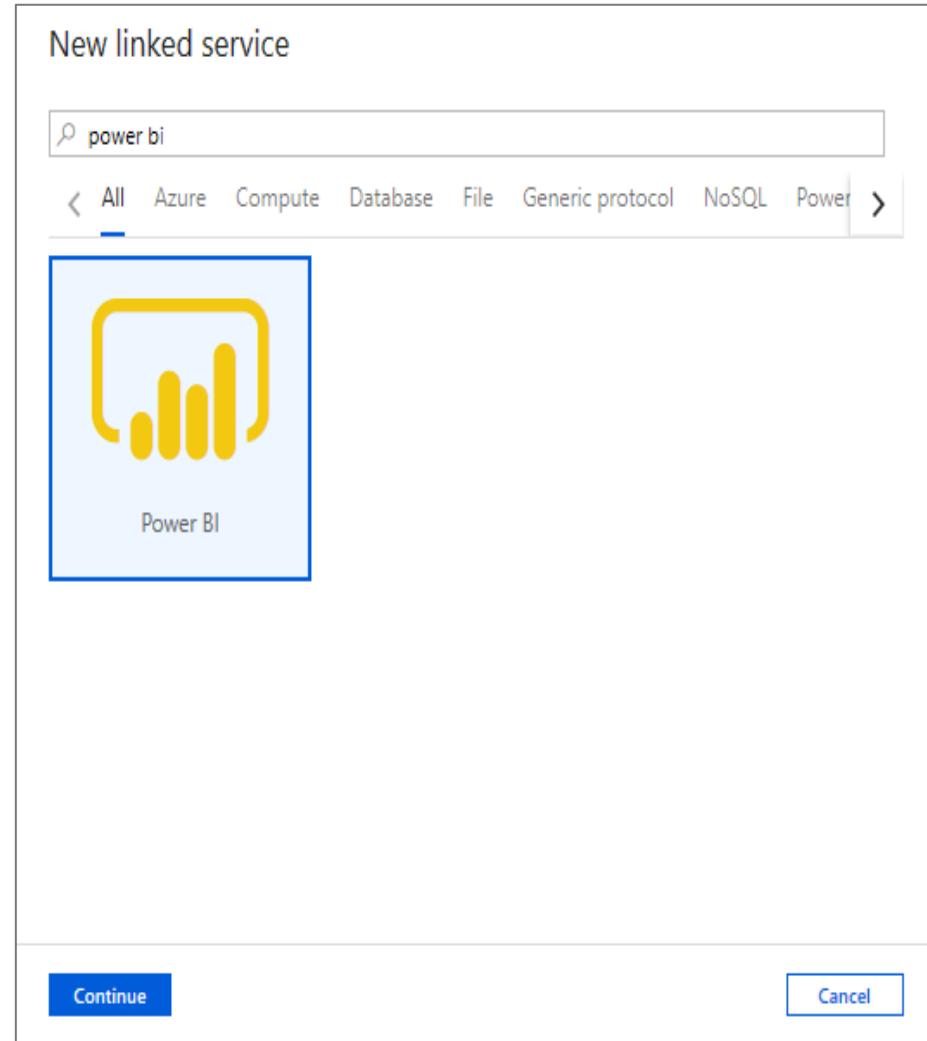
Overview

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. The left sidebar displays the 'Develop' hub with a list of resources: SQL scripts (9), Notebooks (6), Data flows (1), Power BI (1), and gaming-telemetry (Power BI datasets, Power BI reports, Report). The 'Report' item under Power BI is selected. The main content area shows a published Power BI report titled 'GAME STUDIO'. The report features a header with a game console image and a 'Total Users' summary of 24.5M. It includes a 'What If...' section with a slider for free game addons, a table comparing total users across regions (APAC and EMEA) with their forecasts and extra users, and a line chart showing user growth from August to November 2019. The right sidebar contains sections for 'VISUALIZATIONS' (with various chart icons) and 'FIELDS' (listing categories like agegroup, forecast, historical, platform, predictions, realtime, regions, scenario, and weekdays). The bottom navigation bar includes tabs for Historical, Forecast, Predictions, and a plus sign icon.

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 vs "What If" Analysis

Historical Forecast Predictions +

Visualizations Fields

Search

agegroup
forecast
historical
platform
predictions
realtime
regions
scenario
weekdays

Add data fields here

DRILL THROUGH

Cross-report Off —

Keep all filters On —

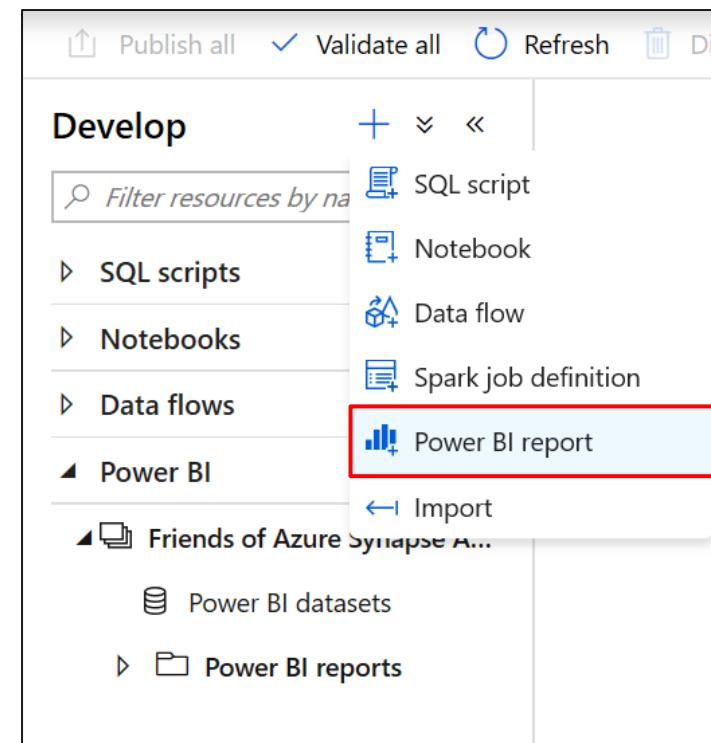
Add drill-through fields here

Develop Hub – Power BI

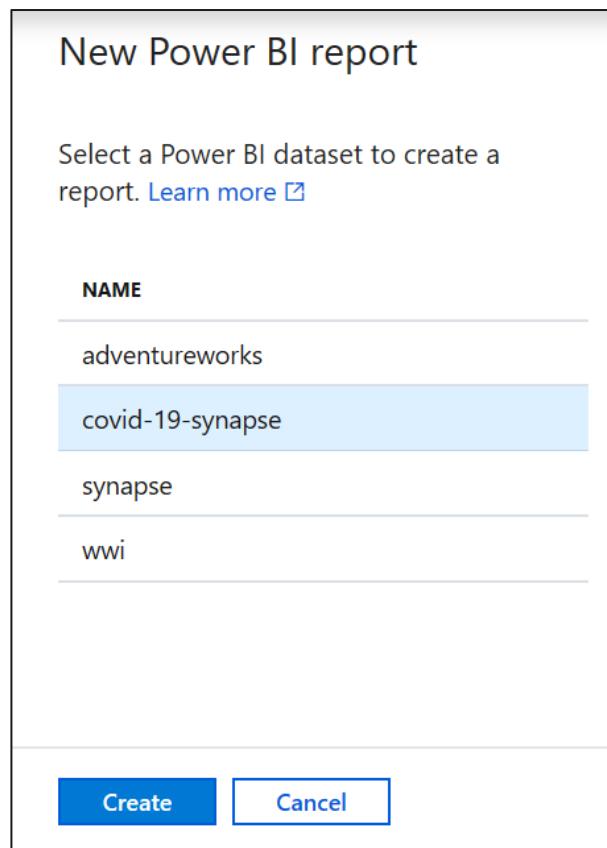
Create new reports from existing published Power BI datasets

Create new Power BI datasets

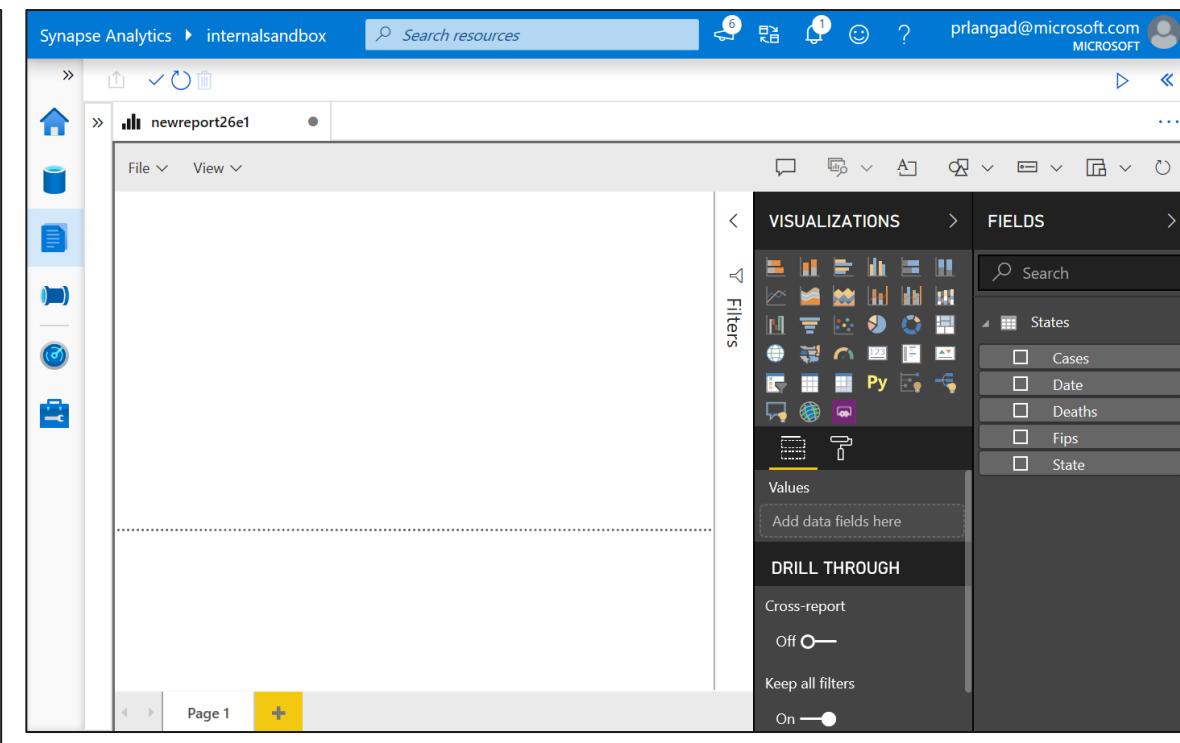
1



2



3



Develop Hub – Power BI

Edit reports in Synapse workspace

Microsoft Azure | Synapse Analytics > gamingtelemetry

Search resources

PriLangad@microsoft.com MICROSOFT

Develop Publish all Validate all Refresh Discard all

Report

Ask a question Explore Text box Shapes Buttons Visual interactions Refresh Duplicate this page Save

Filter resources by name

Develop + <>

SQL scripts 9

Notebooks 6

Data flows 1

Power BI 1

gaming-telemetry

Power BI datasets

Power BI reports

Report

Select a Platform: Console

GAME STUDIO

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

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

Extra Users 252.8K +3.4% Users Increase

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

Date Aug 2019 Sep 2019 Oct 2019 Nov 2019

Total Users 24.5M

Tabular Map Forecast Extra Users

Historical Forecast Predictions +

Visualizations Fields

Filters

agegroup forecast historical platform predictions realtime regions scenario weekdays

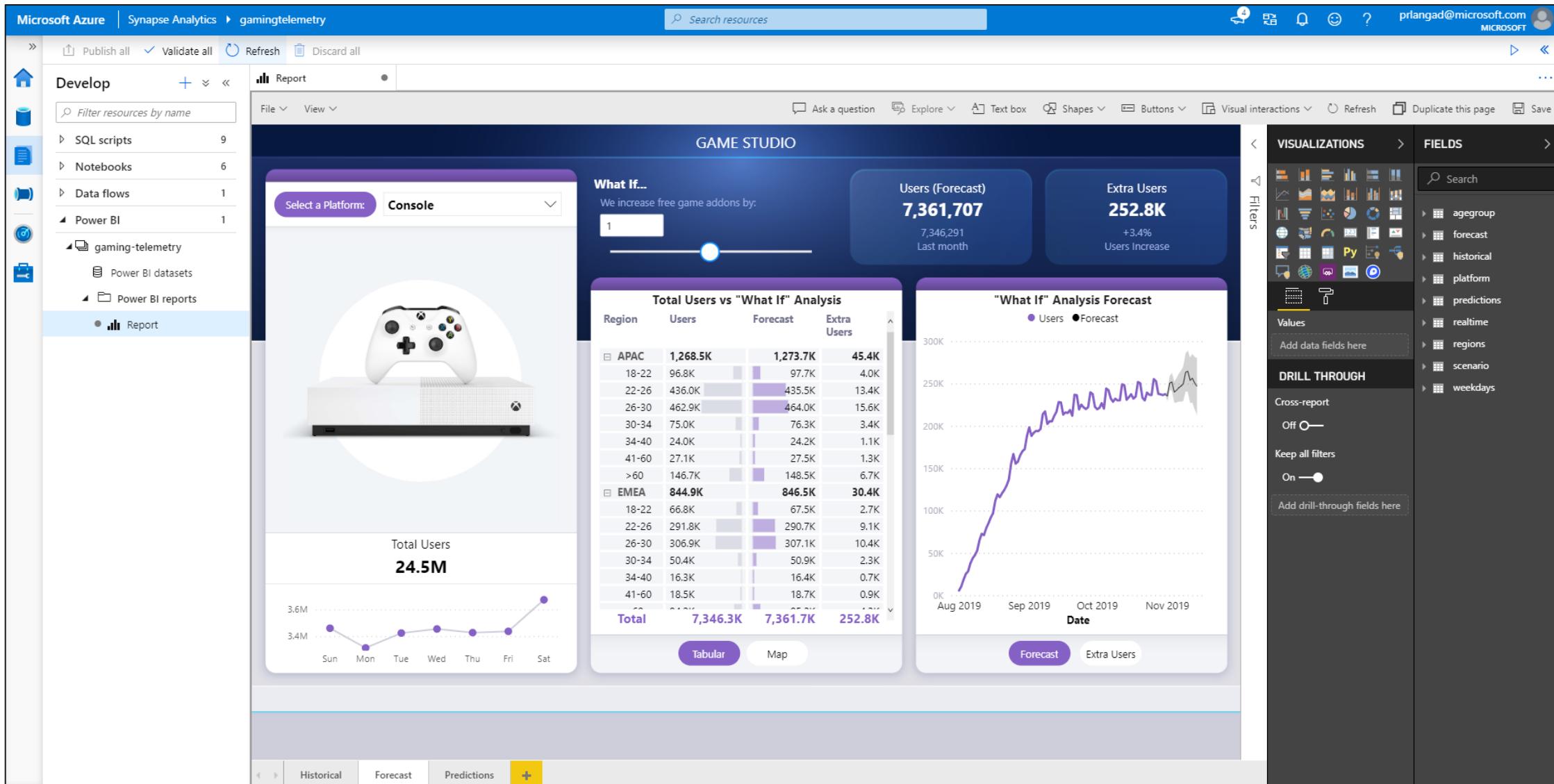
Values Add data fields here

DRILL THROUGH

Cross-report Off

Keep all filters On

Add drill-through fields here



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 center toolbar. The main area displays a Power BI report titled 'GAME STUDIO' with various visualizations, including a chart showing 'Total Users' at 24.5M and a 'What If...' analysis section. The right sidebar contains sections for 'VISUALIZATIONS' and 'FIELDS', with a search bar and a list of available fields like 'agegroup', 'forecast', and 'platform'.

Publish changes by simple save report in workspace

Save this report

Microsoft Azure | Synapse Analytics > gamingtelemetry

Develop

SQL scripts 9

Notebooks 6

Data flows 1

Power BI 1

gaming-telemetry

Power BI datasets

Power BI reports

Report

File View

Ask a question Explore Text box Shapes Buttons Visual interactions Refresh Duplicate this page Save

Search resources

prlangad@microsoft.com MICROSOFT

GAME STUDIO

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

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" Analysis Forecast

Users (Forecast) 7,613,619

Date Aug 2019 Sep 2019 Oct 2019 Nov 2019

Historical Forecast Predictions

VISUALIZATIONS FIELDS

Search

agegroup forecast historical platform predictions realtime regions scenario weekdays

Values Add data fields here

DRILL THROUGH

Cross-report Off — Keep all filters On —

Add drill-through fields here

Develop – CI/CD

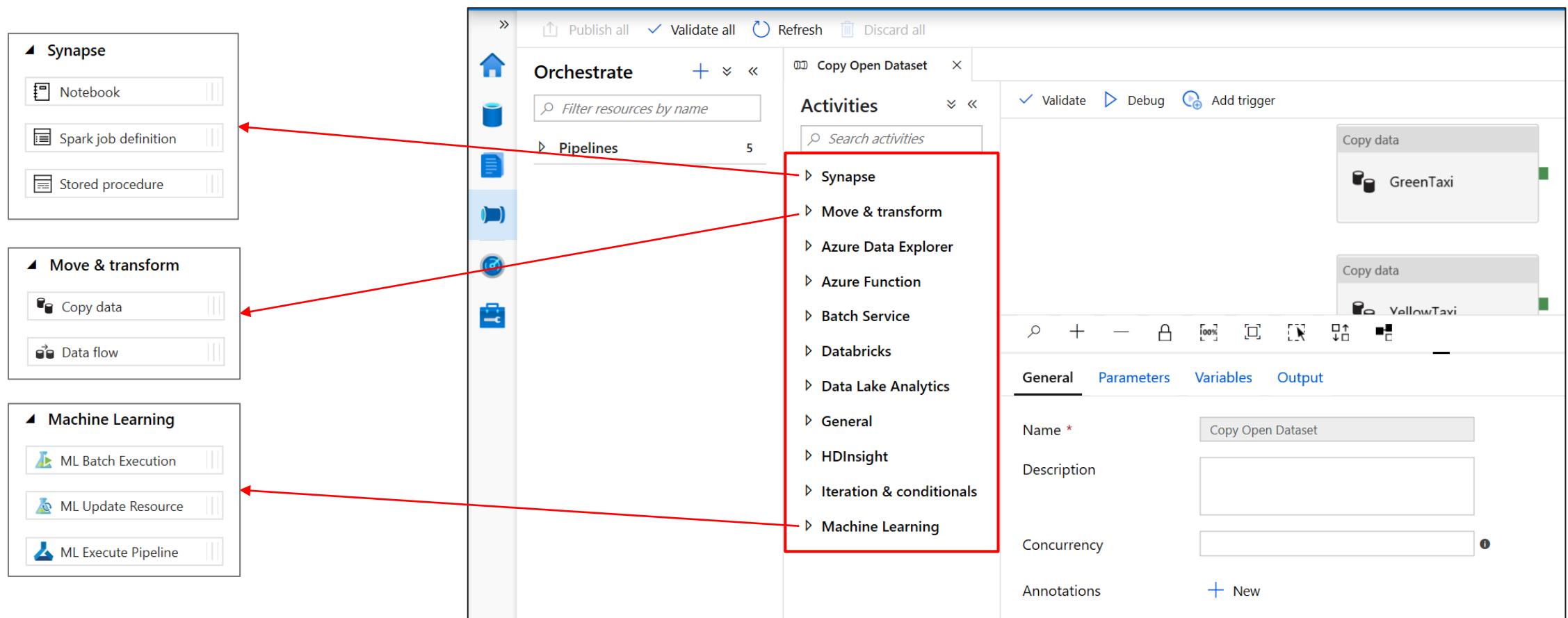
Commit artifacts to source-controlled repository and operationalize release pipelines with Synapse deployment task

The screenshot illustrates the integration of GitHub and Azure Synapse for CI/CD. On the left, a GitHub browser interface shows a repository named 'synapsetestdemo-ws-01' with a 'dev' branch. The repository contains several files and folders, including 'credential', 'dataflow', 'dataset', 'integrationRuntime', 'linkedService', 'notebook', 'pipeline', 'sparkJobDefinition', 'sqlscript', and 'README.md'. The 'README.md' file contains the text 'Initial commit'. 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, 11:02 PM'. Below this is an 'Artifacts' section listing 'azuresynapsesdev' with hash '0c8b1a872' and branch 'branch-retail-12'. To the right is a 'Stages' section showing a single stage named 'Load to Prod' with a green checkmark indicating it has 'Succeeded' on '11/17/2020, 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.



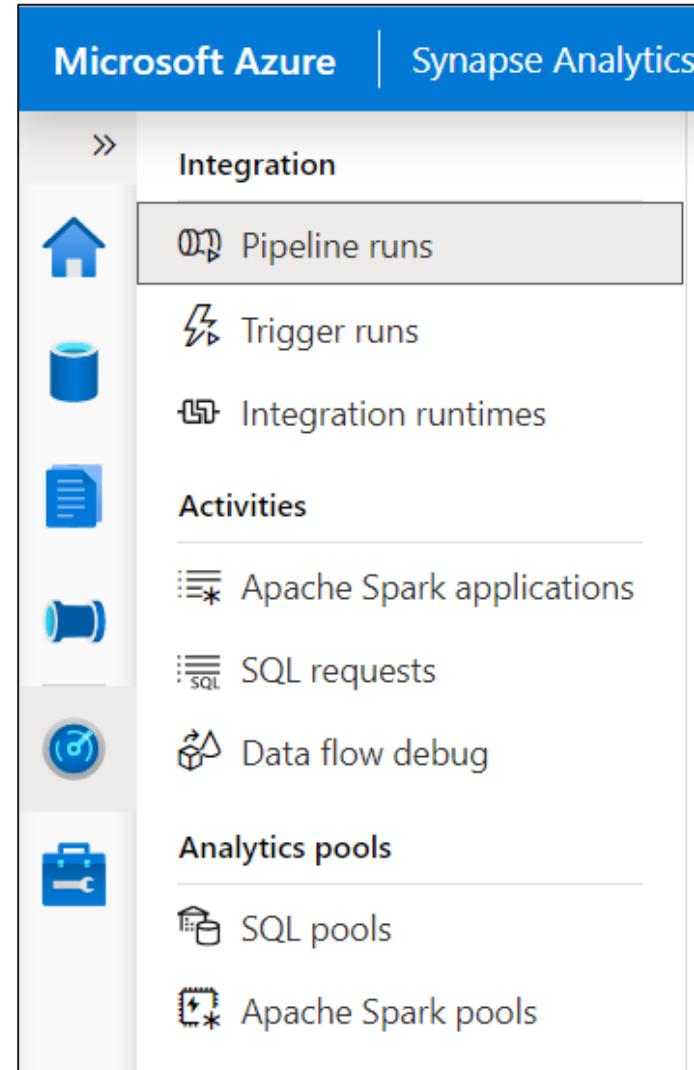
Monitor Hub

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 at the top of the list.

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. The top navigation bar includes 'Microsoft Azure', 'Synapse Analytics', and a resource name 'wsazuresynapseanalytics'. Below the navigation are buttons for 'Validate all' and 'Publish all'. On the left, a sidebar lists various management options: '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 content area is titled 'Linked services' and contains a descriptive text: 'Linked services are much like connection strings, which define the connection to external resources.' A blue link 'Learn more' is present. A red box highlights the '+ New' button, and a red arrow points from this button to a modal window titled 'New linked service'. The modal lists 15 available connectors, each with an icon and name: PayPal (Preview), Phoenix, PostgreSQL, Power BI (highlighted with a blue box), Presto (Preview), QuickBooks (Preview), REST, SAP BW, SAP BW via MDX, SAP Cloud For Customer, SAP ECC, SAP HANA, Bing-Covid-19-Dat, AzureMLService1, AzureMLServiceN, and Nellies_Keyvault. At the bottom of the modal are 'Continue' and 'Cancel' buttons.

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 Synapse Analytics portal interface. On the left, there is a navigation sidebar with various options like 'Analytics pools', 'SQL pools', 'Apache Spark pools', 'External connections', 'Linked services', 'Integration runtimes', 'Triggers', 'Access control', and 'Security'. The 'Triggers' option is highlighted with a red box and has a red arrow pointing to it from the 'New' button in the main content area. The main content area displays a 'Triggers' section with a brief description: 'To execute a pipeline set the trigger to be kicked off.' Below this is a 'New' button, also highlighted with a red box. To the right of the main content is a 'New trigger' dialog box. This dialog box contains fields for 'Name' (set to 'Trigger 2'), 'Description', 'Type' (set to 'Schedule'), 'Start Date (UTC)' (set to '10/29/2019 9:46 PM'), 'Recurrence' (set to 'Every 1 Minute(s)'), 'End' (set to 'No End'), 'Annotations', 'Activated' (set to 'No'), and 'OK' and 'Cancel' buttons at the bottom.

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 area displays the 'Access control' section, which lists three items: 'Workspace admin' (soft.com, Individual, Workspace admin). A red box highlights the '+ Add' button. Below this, a red arrow points from the 'Orchestration' section in the sidebar to the 'Add role assignment' dialog. Another red arrow points from the 'Individual' role entry in the table to the same dialog. The 'Add role assignment' dialog shows fields for 'Scope' (set to 'Workspace'), 'Role' (dropdown menu), 'Item type' (set to 'Credentials'), 'Item' (dropdown menu set to 'WorkspaceSystemIdentity'), 'Role' (dropdown menu set to 'Synapse Administrator'), and 'Select user' (text input field). The bottom of the dialog indicates '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

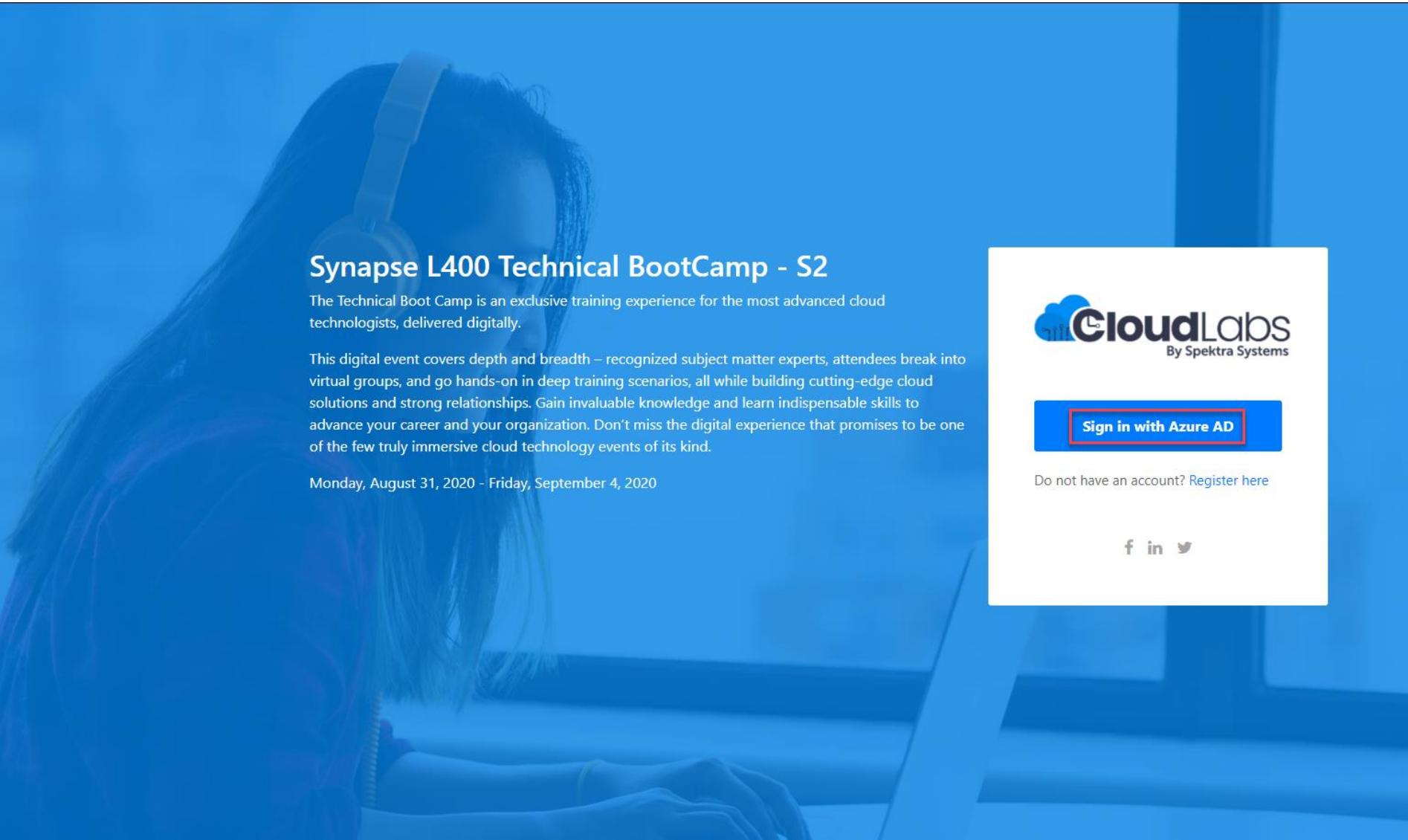
Publish branch main

Root folder /

Source control **Git configuration**

main branch
dev branch
workspace_publish branch
Create pull request [Alt+P]
New branch [Alt+N]
Switch to live mode

Login to the portal

A large, semi-transparent blue rectangular overlay covers the left side of the image, containing promotional text for the Synapse L400 Technical BootCamp - S2.

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

On the right side of the image, there is a white rectangular login box for "CloudLabs By Spektra Systems". It features a blue header bar with the text "Sign in with Azure AD" and a red rectangular border around the "Sign in with Azure AD" button. Below the header, it says "Do not have an account? Register here". At the bottom, there are social media icons for Facebook, LinkedIn, and Twitter.

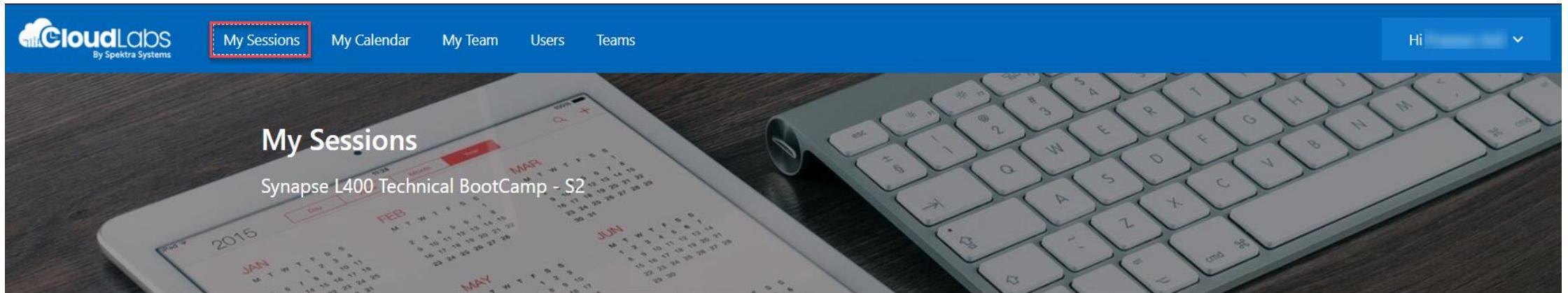
CloudLabs
By Spektra Systems

[Sign in with Azure AD](#)

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

f in tw

Sessions for Bootcamp



The screenshot shows the CloudLabs platform interface. At the top, there is a navigation bar with the CloudLabs logo, a search bar, and links for 'My Sessions', 'My Calendar', 'My Team', 'Users', and 'Teams'. A red dashed box highlights the 'My Sessions' link. On the right side of the header is a 'Hi [username]' greeting. Below the header, a large image of a keyboard and a tablet displaying a calendar is visible. The main content area is titled 'My Sessions' and shows a list for 'Synapse L400 Technical BootCamp - S2'. It includes a 'Search' bar, date filters ('All Days', 'Mon 31', 'Tue 1', 'Wed 2', 'Thu 3'), and a 'Add to Calendar' button. The total number of sessions (37) is displayed. A detailed session card for a 'Welcome' session on August 31, 2020, is shown, featuring the speaker's name (Leanne Gallagher), the date, time, duration, and a brief description of the bootcamp.

CloudLabs
By Spektra Systems

Hi [username] ▾

My Sessions

Synapse L400 Technical BootCamp - S2

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

Search 

37 Sessions

Welcome

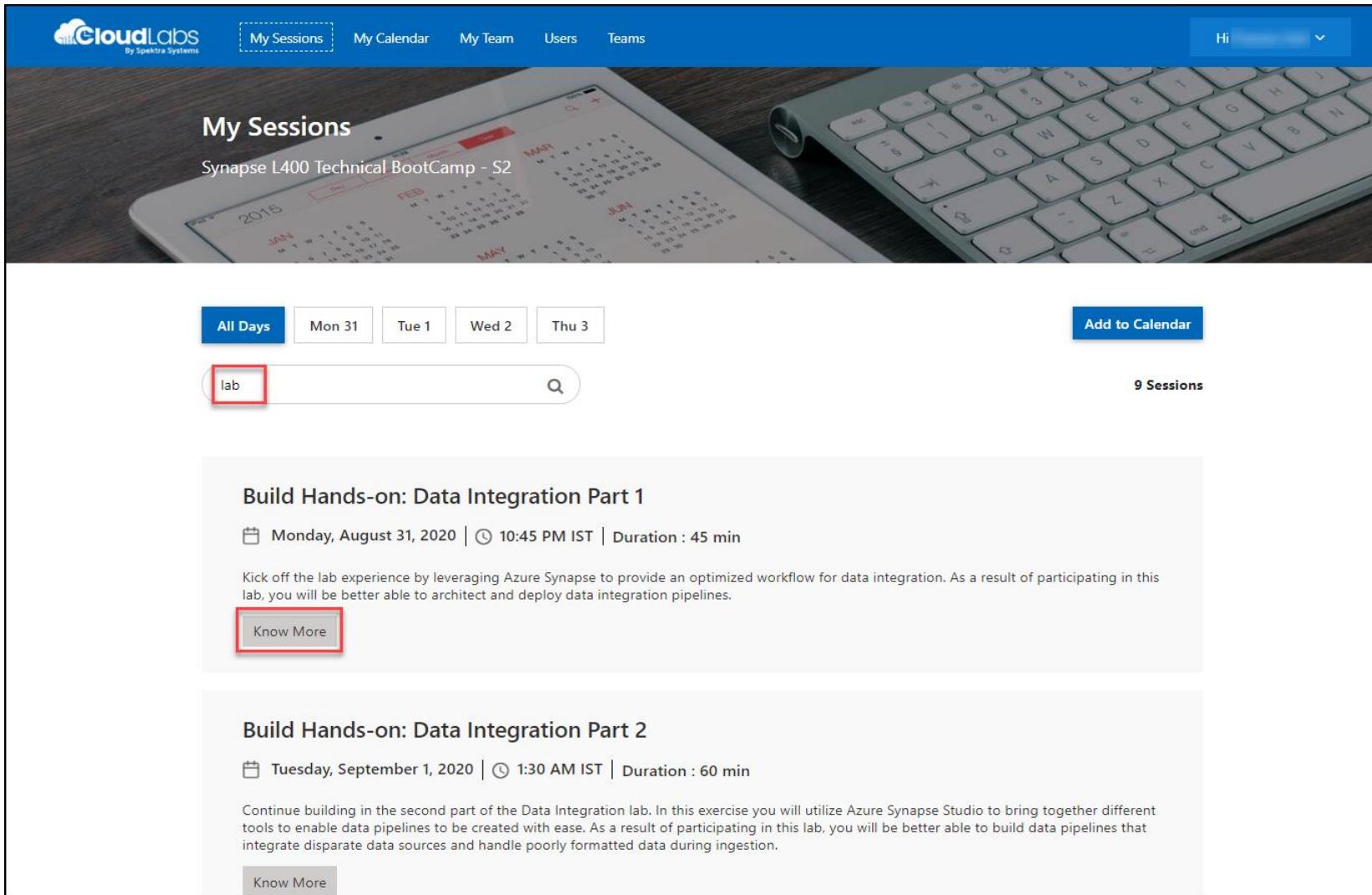
Speaker: Leanne Gallagher

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](#)

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

9 Sessions

lab

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**

Data Loading & Data Lake Organization



Agenda

1 Integration (Orchestration)

Synapse pipelines

2 Ingest files to tables

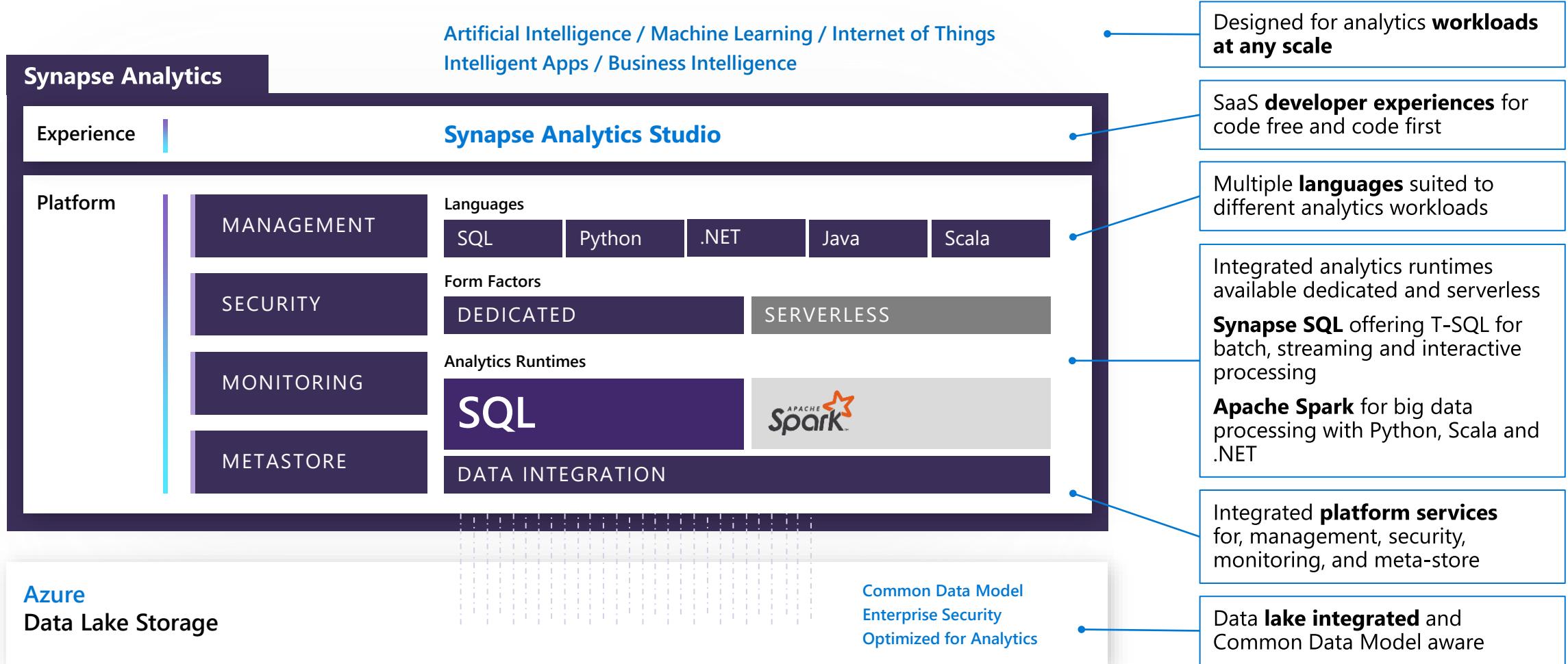
Copy versus CTAS

3 Best practices

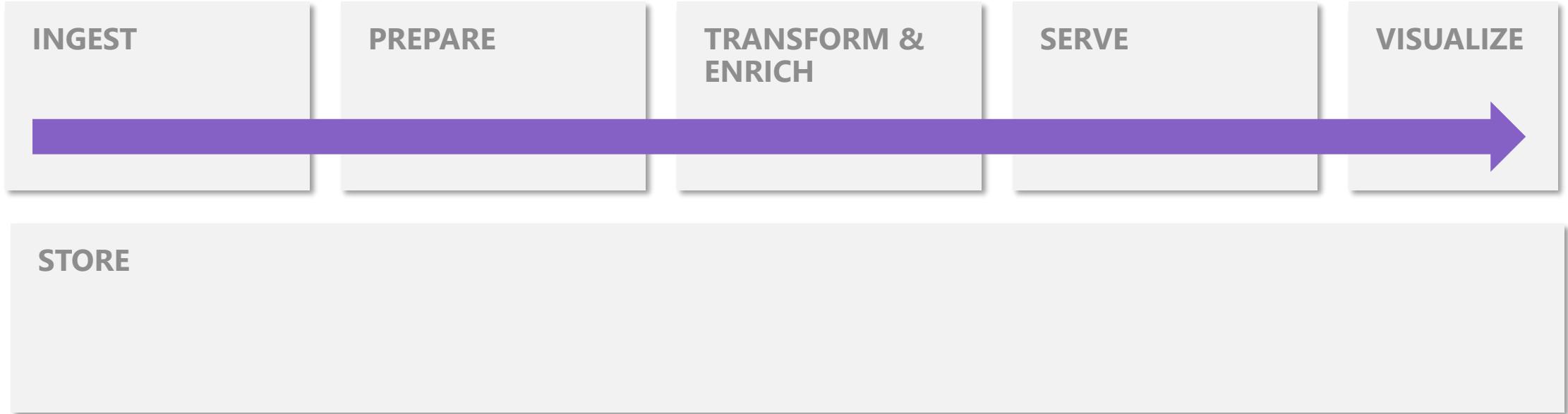
Various ingest and storage best practices

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 options like 'External connections', 'Linked services' (which is selected and highlighted with a red box), 'Orchestration', 'Triggers', 'Integration runtimes', 'Security', and 'Access control'. The main area is titled 'Linked services' and contains a table with columns 'NAME', 'TYPE', and 'ANNOTATIONS'. A red box highlights the '+ New' button. Below the table, a modal window titled 'New linked service' is open, showing a grid of connector icons and names. A red arrow points from the '+ New' button to the 'Power BI' icon in the grid. The modal also has 'Continue' and 'Cancel' buttons at the bottom.

NAME	TYPE	ANNOTATIONS
ADLSG2OpenDataSetSink	Azure Data Lake Storage Gen2	
AzureBlobStorage1	Azure Blob Storage	
AzureDataLakeStorage1	Azure Data Lake Storage Gen2	
AzureDataLakeStorage2Source		
AzureOpenDataset		
AzureOpenDataSet2		
AzureSqlDW1		

New linked service

PayPal (Preview)	Phoenix	PostgreSQL
Power BI	Presto (Preview)	QuickBooks (Preview)
REST	SAP BW Open Hub	SAP BW via MDX
SAP Cloud For Customer	SAP ECC	SAP HANA
SAP		

Continue Cancel

90+ Connectors out of the box

Datasets

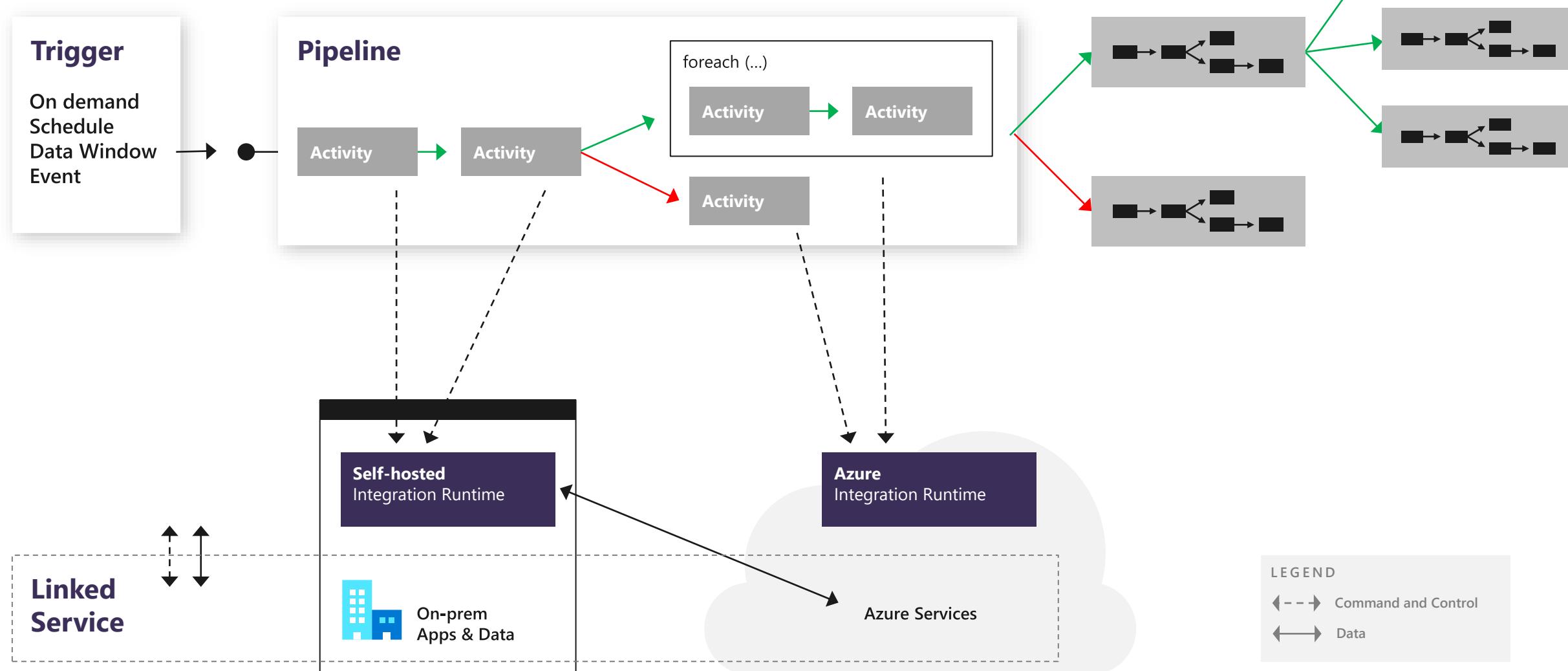
Orchestration datasets describe data that is persisted.

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
 - Dataset Type:** Parquet
 - Dataset Name:** NYCTaxiParquet
 - General Tab:** Contains fields for Linked service (Lake_ArcadiaLake), File path (data / nyctaxi / File), and Compression type (snappy). Buttons for Test connection, Open, New, Browse, and Preview data are also present.
 - Connection Tab:** Active tab, showing the linked service configuration.
 - Schema Tab:** Placeholder for dataset schema.
 - Parameters Tab:** Placeholder for dataset parameters.

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. On the left, three activity selection boxes are overlaid on the interface:

- Move & transform**: Contains "Copy data" and "Data flow". A red arrow points from this box to the "Move & transform" section in the central Activities list.
- Machine Learning**: Contains "ML Batch Execution", "ML Update Resource", and "ML Execute Pipeline". A red arrow points from this box to the "Machine Learning" section in the central Activities list.
- Synapse**: Contains "Notebook", "Spark job definition", and "Stored procedure". A red arrow points from this box to the "Synapse" section in the central Activities list.

The central area shows the "Orchestrator" interface for "Pipeline 2". The "Activities" list on the right includes:

- Move & transform
- Azure Data Explorer
- Azure Function
- Batch Service
- Data Lake Analytics
- Databricks
- General
- HDInsight
- Iteration & conditionals
- Machine Learning
- Synapse

The pipeline canvas shows a "Stored procedure" activity named "sql1_dbo_StorePredictions" connected to a "Notebook" activity named "BOOT_Basic_spark".

Bottom navigation tabs include General, Parameters, Variables, and Output. Pipeline details are shown in the General tab:

- Name: Pipeline 2
- Description: (empty)
- Concurrency: (empty)
- Annotations: (empty)

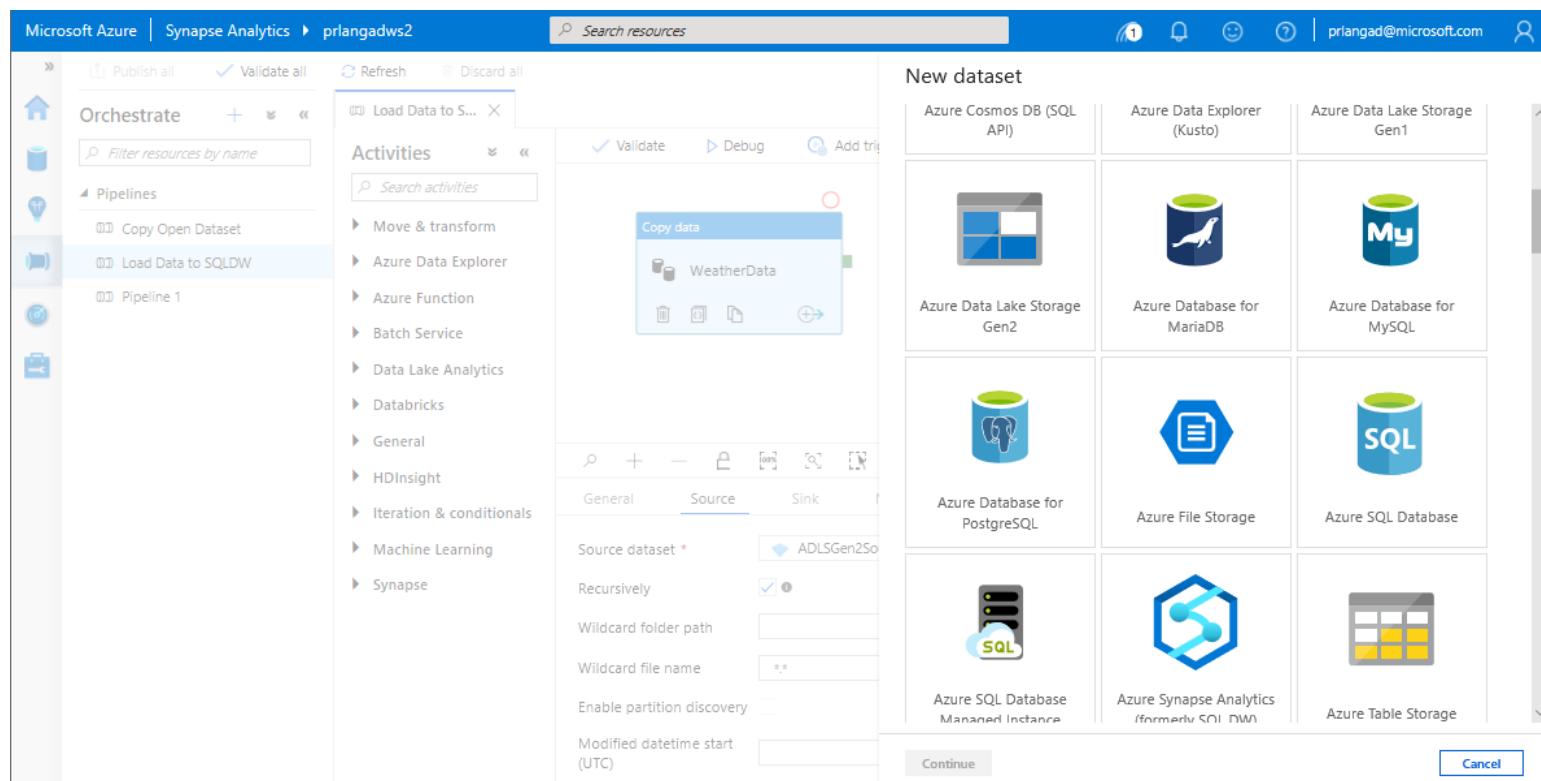
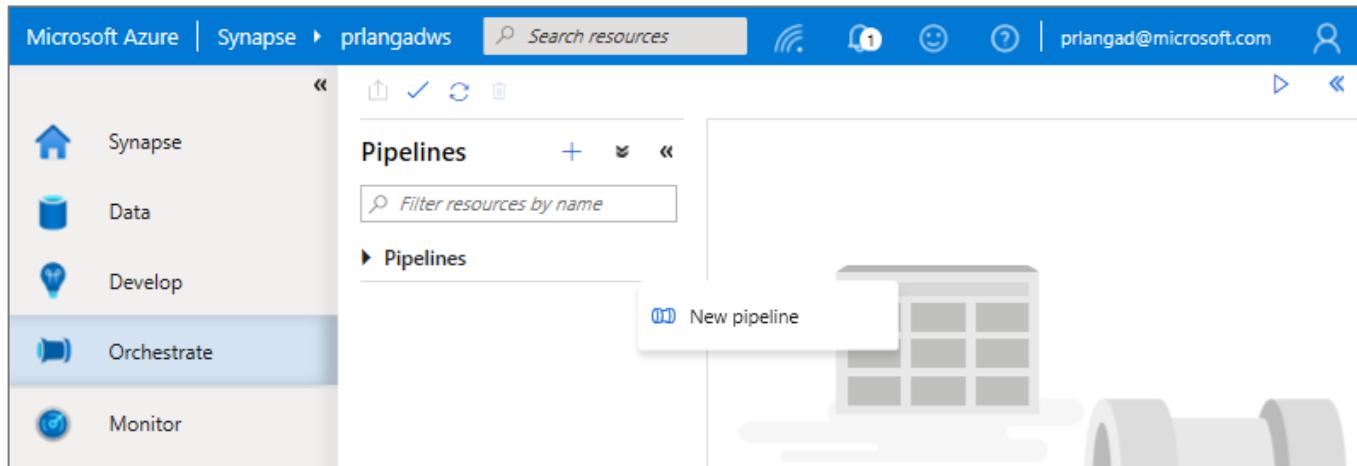
Pipelines

Overview

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

Benefits

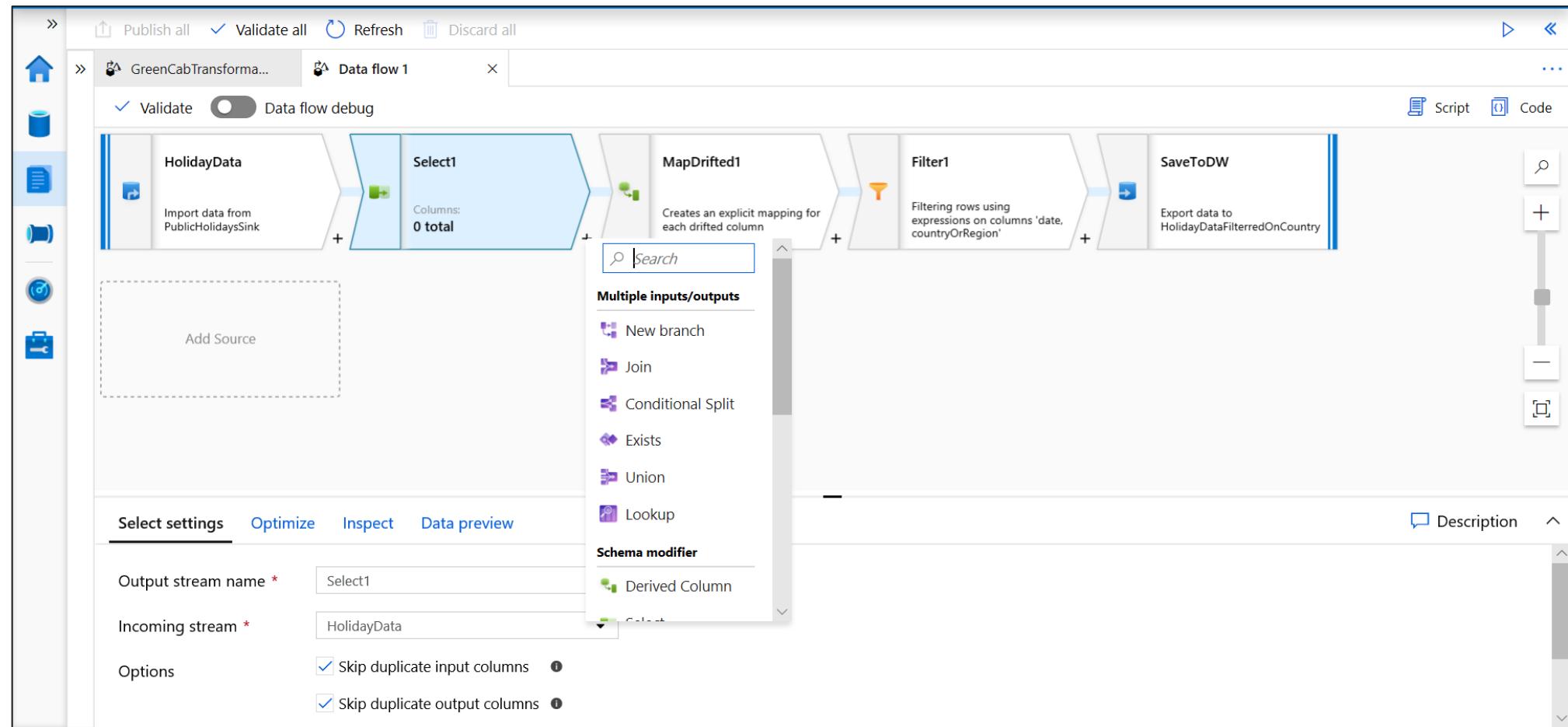
- Supports common loading patterns.
- Fully parallel loading into data lake or SQL tables.
- Graphical development experience.



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



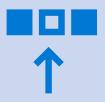
Add file handling (move files after read, write files to file names described in rows etc)



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's a sidebar with icons for 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 a sub-instruction: '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 icon 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)
- 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 lists various management options: 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 contains the following content:

- A descriptive text: "The integration runtime (IR) is the compute infrastructure to provide the following network environment." followed by a "Learn more" link.
- A button labeled "+ New" with a red box around it.
- A "Refresh" button.
- A "Filter by keyword" input field.
- A message indicating "Showing 1 - 1 of 1 items".
- Three sorting/filtering columns: "Name ↑", "Type ↑", "Sub-type ↑", and "Status ↑".
- A section titled "Integration runtime setup" with the sub-instruction: "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".
- Two options: "Azure" (represented by a cloud icon) and "Self-Hosted" (represented by a server icon).
- At the bottom are "Continue", "Back", and "Cancel" buttons.

Data Movement with Integration Runtimes

Scalable

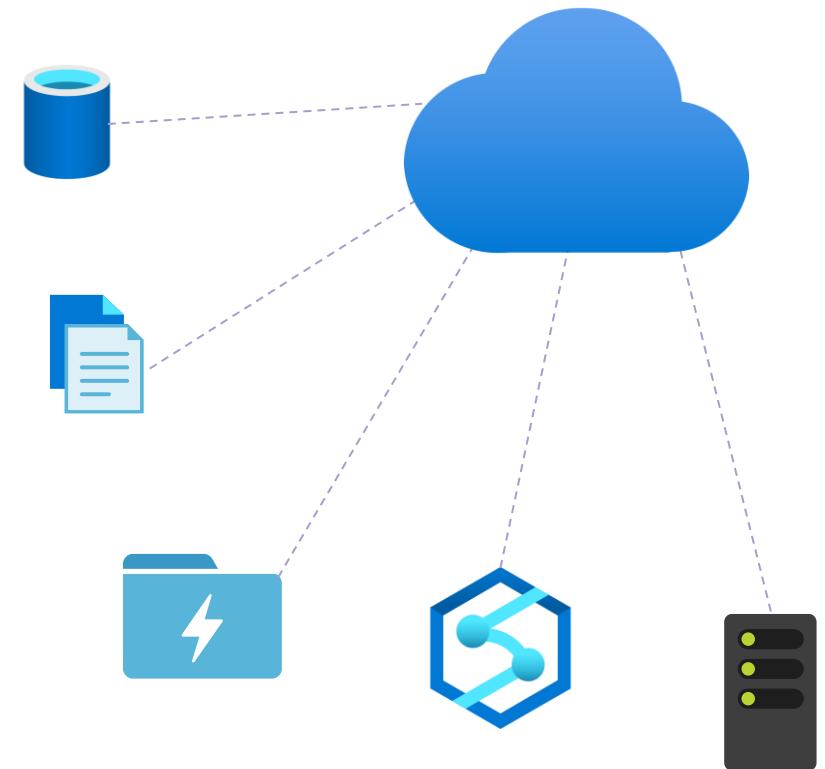
- per job elasticity
- Up to 4 GB/s

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

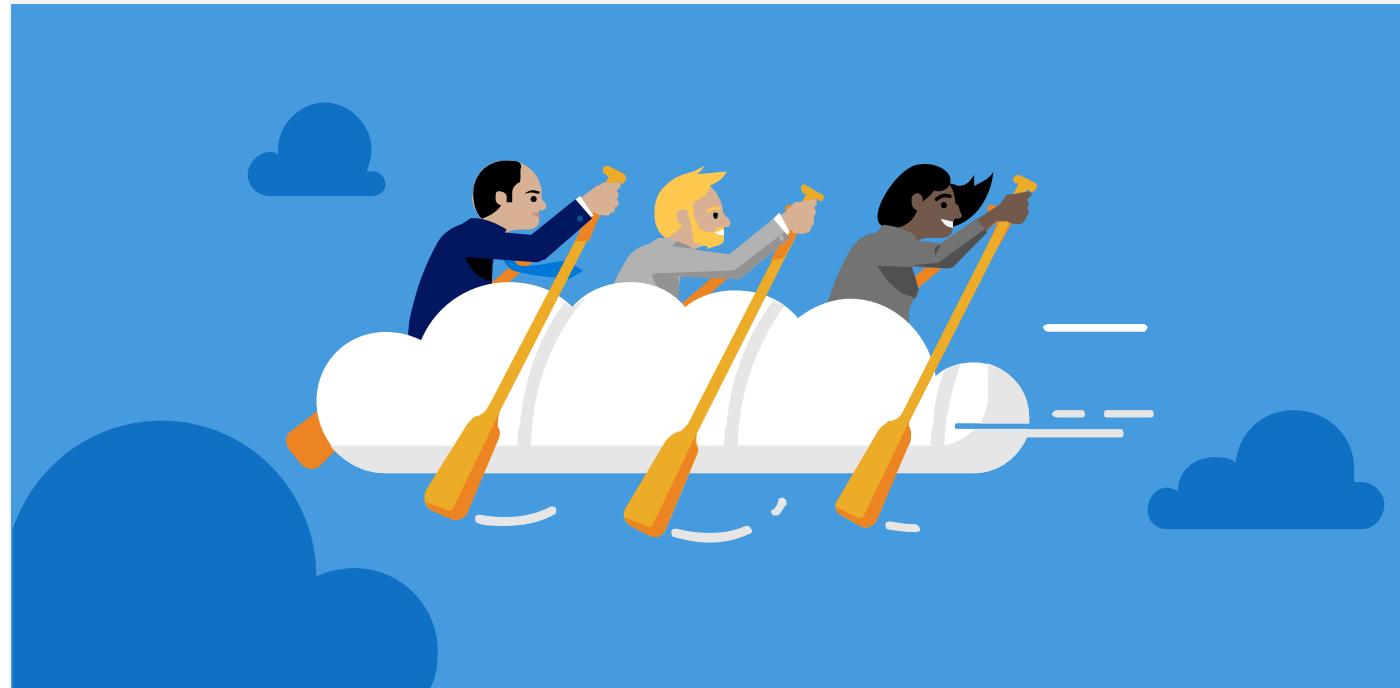
Which one of these is NOT a component of a Synapse pipeline?

A)
I.R.

B)
Linked
Service

C)
Table

D)
Activity



Pop Quiz 1

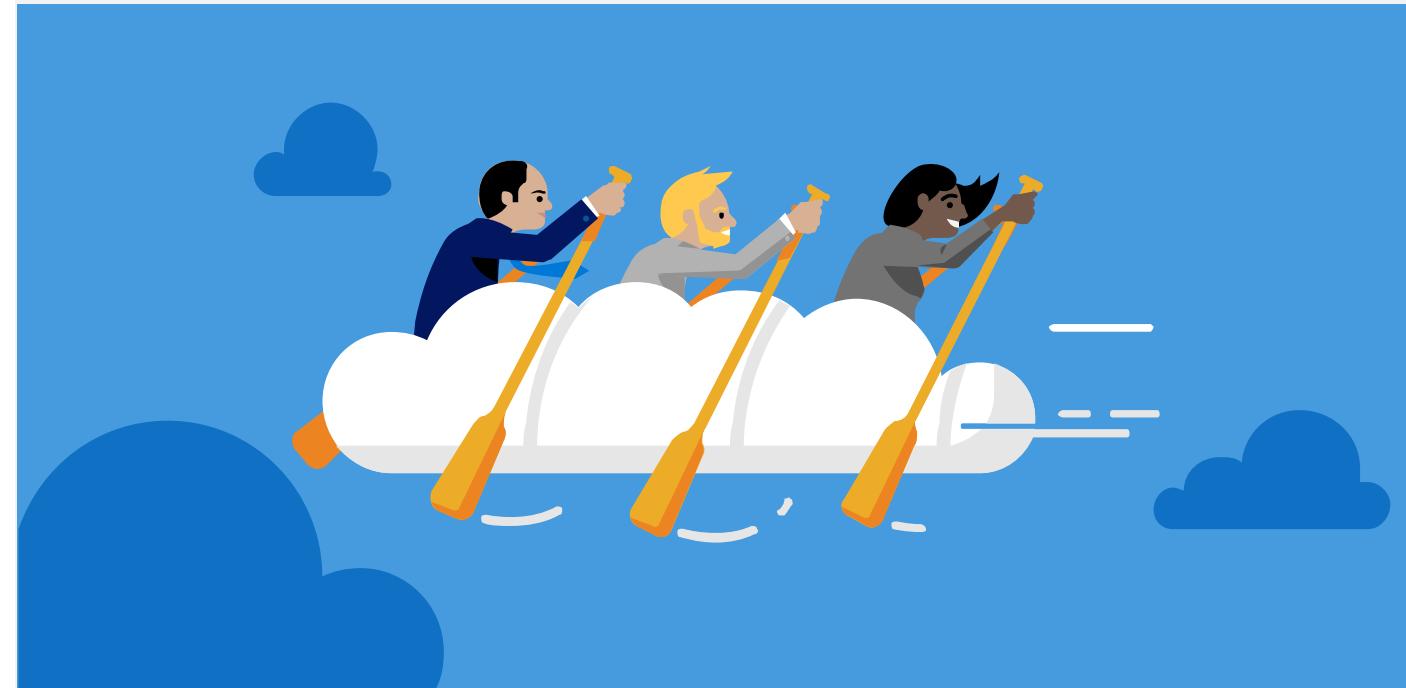
Which one of these is NOT a component of a Synapse pipeline?

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
- Enables querying via external tables
- Challenges:
 - Row width (1 MB)
 - Delimiters in text
 - Fixed line delimiter
 - Code complexity

Copy

- Relaxed permission
- 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

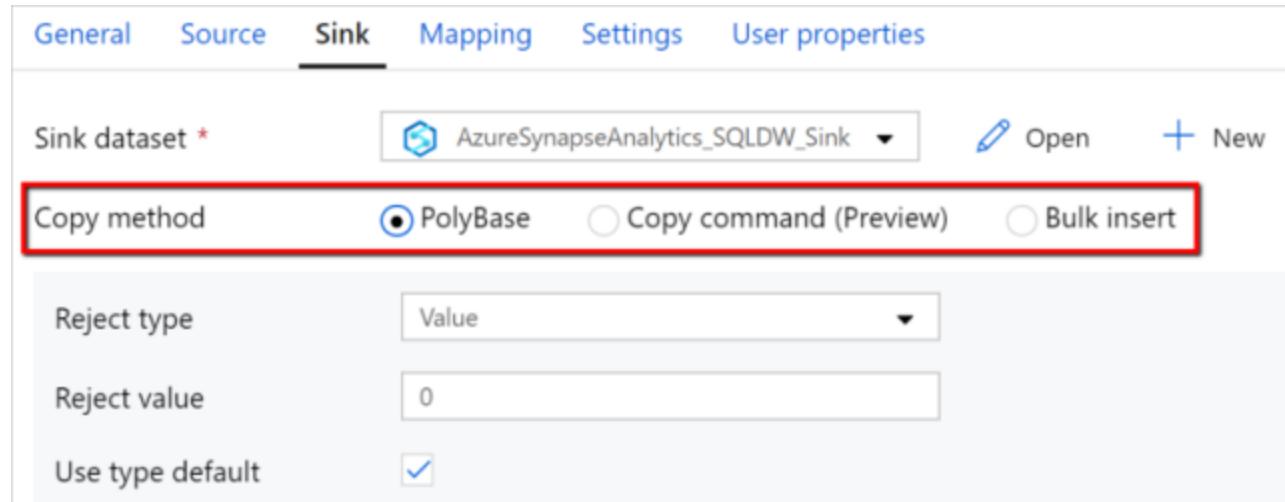
- Network unreliability, no retries
- Needs VM in cloud, performance dependent on VM configuration
- Doesn't support ADLS
- Reduces concurrency
- Control-gated performance limitation, can not scale with DWU

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.

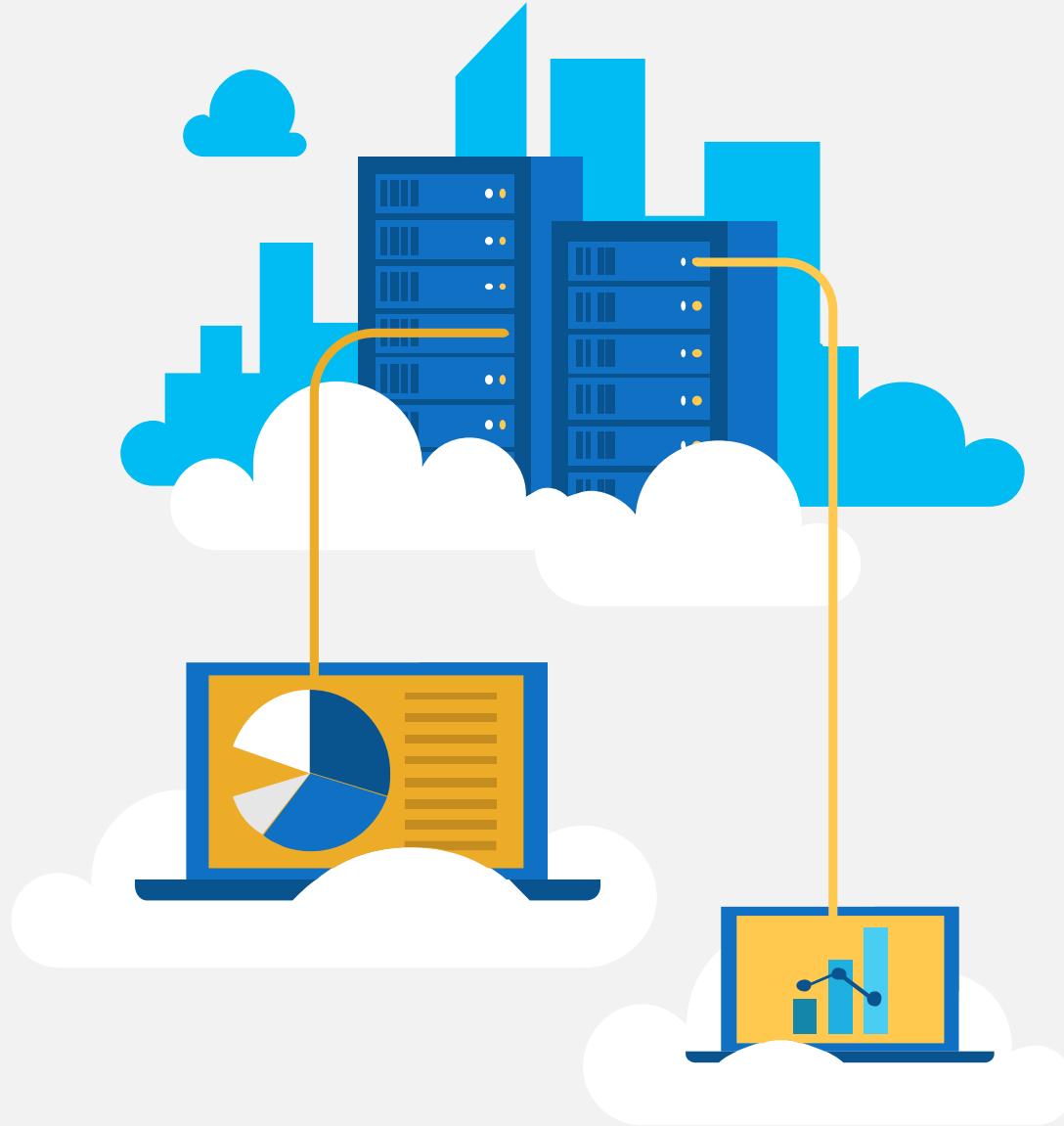
As CETAS generates Parquet files, statistics will be automatically created when the first query targets this external table, resulting in improved performance.

Pop Quiz 2

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

TRUE

FALSE

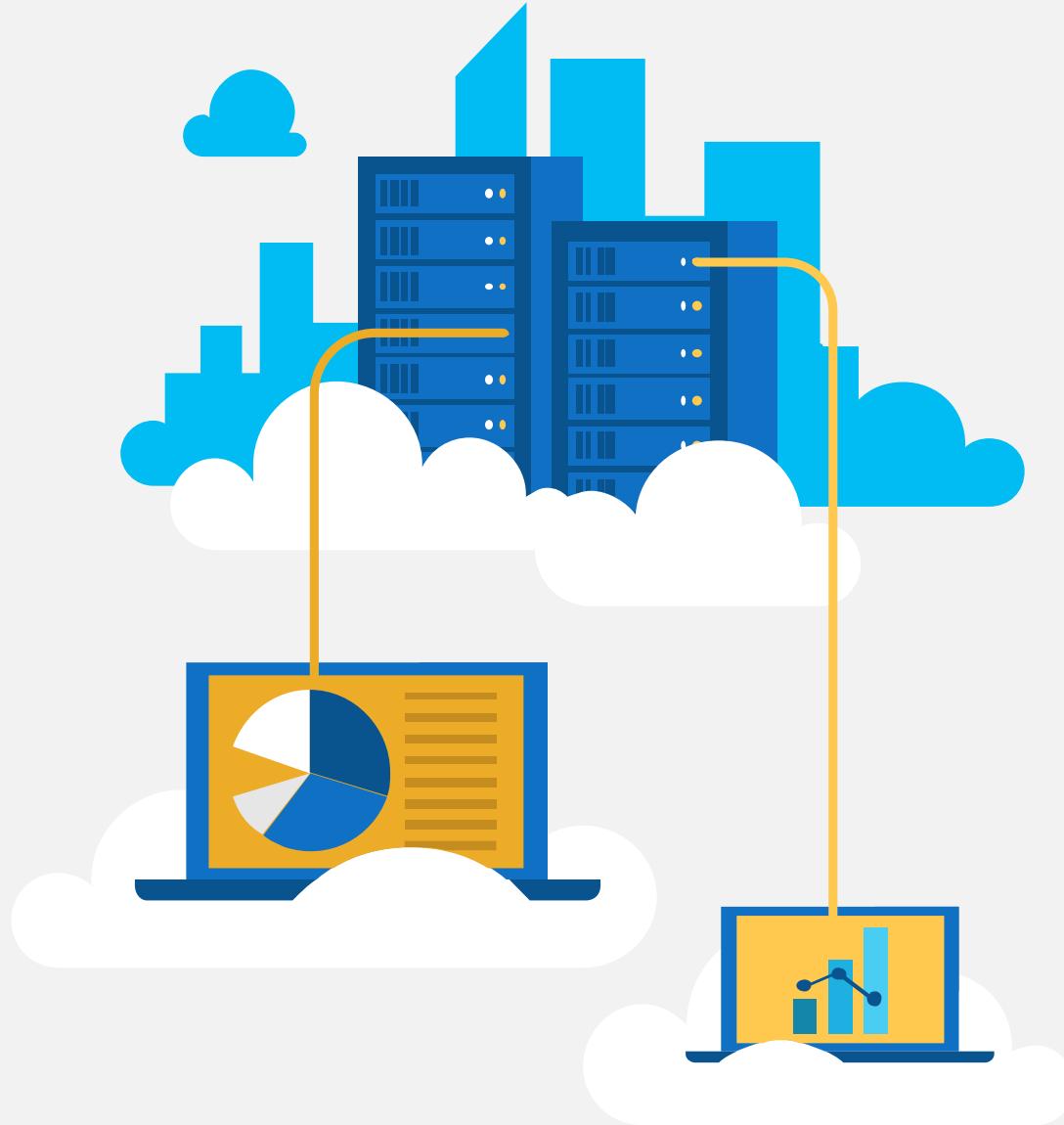


Pop Quiz 2

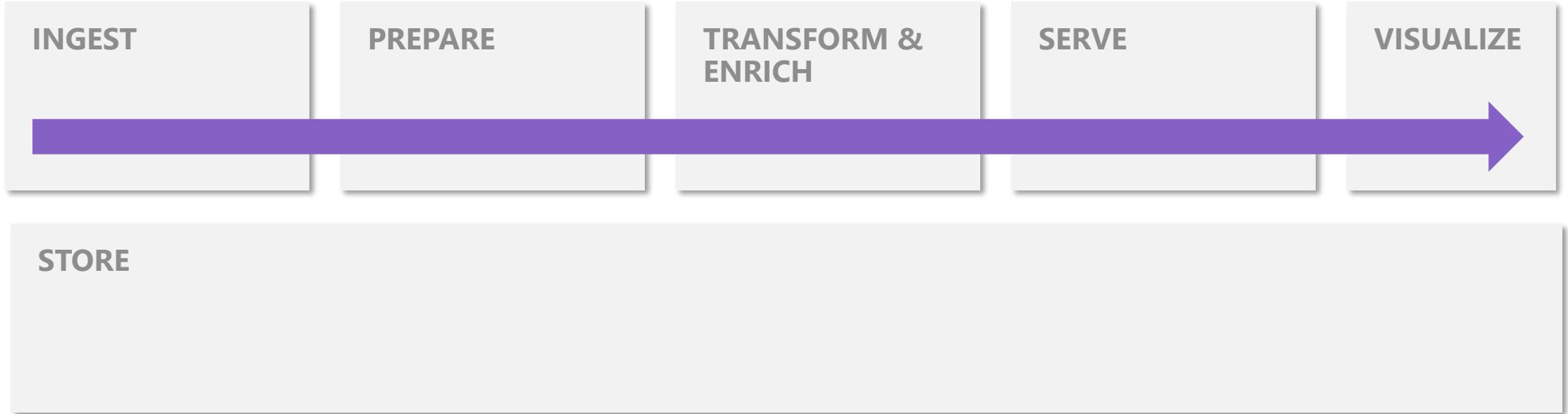
True or False: Both COPY command AND Polybase require CONTROL permissions

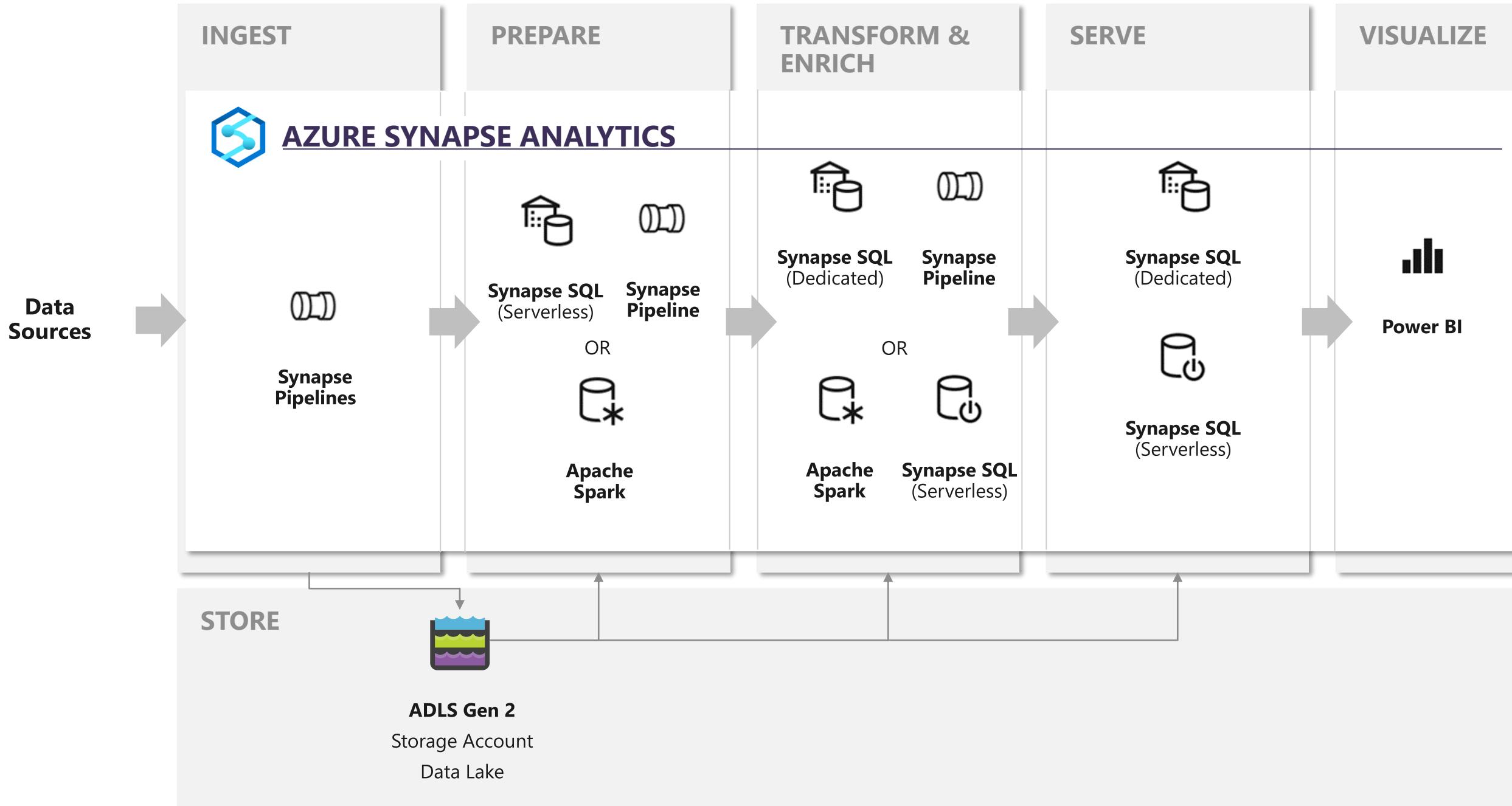
TRUE

FALSE



Modern Data Warehouse







Thank you

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.

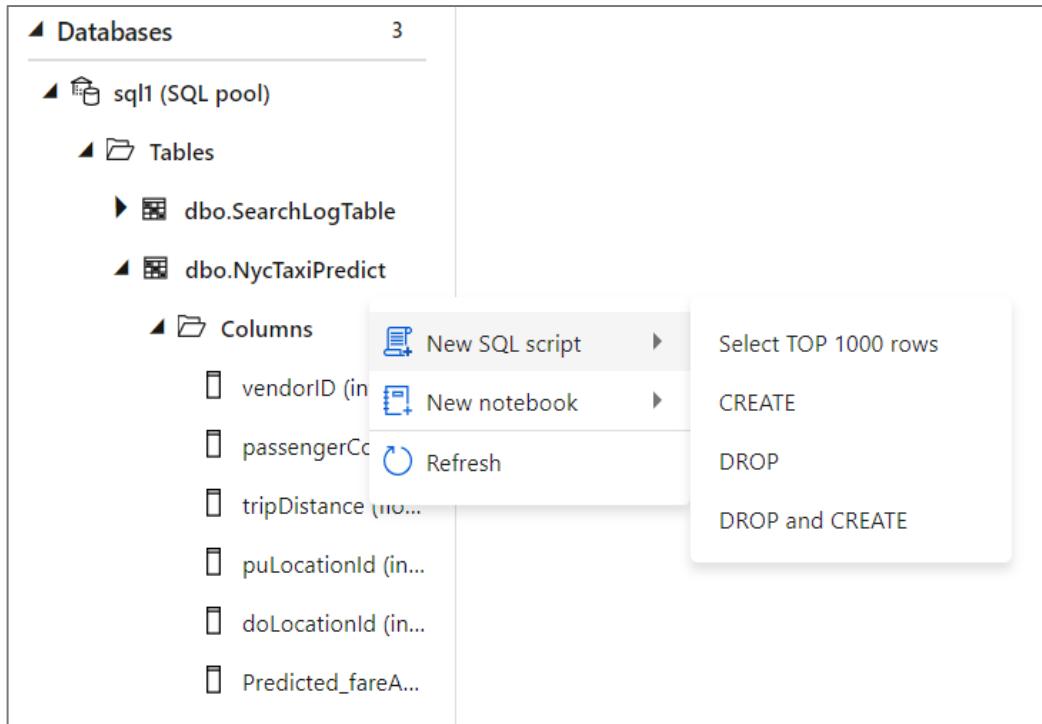
Typical Data Transformations

- 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

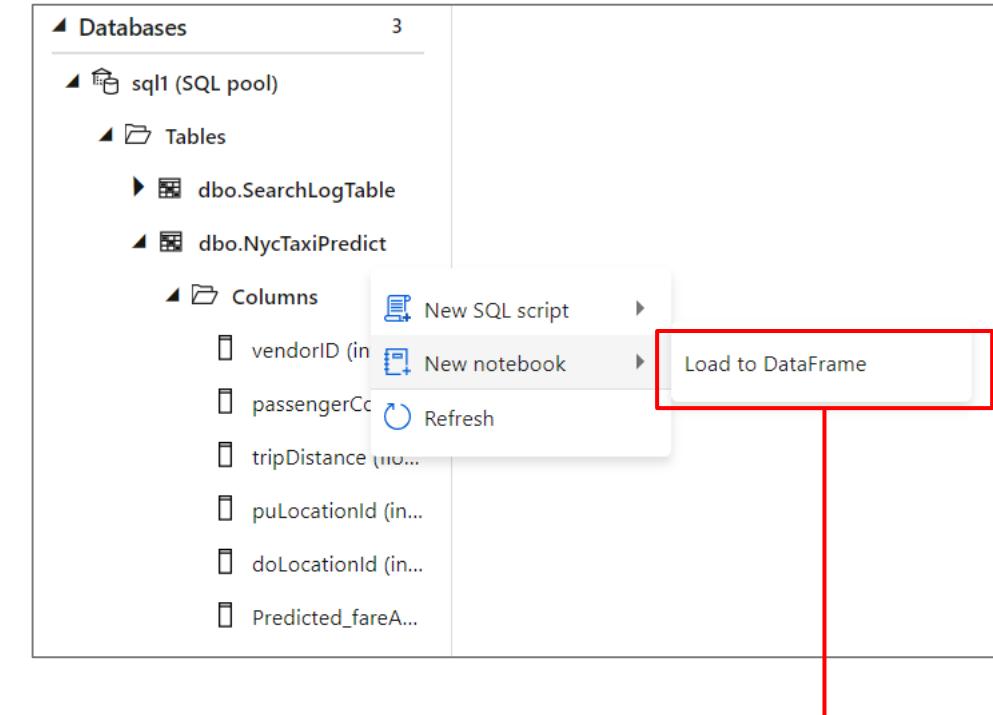
Applying transformations

Code based transformations

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



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.

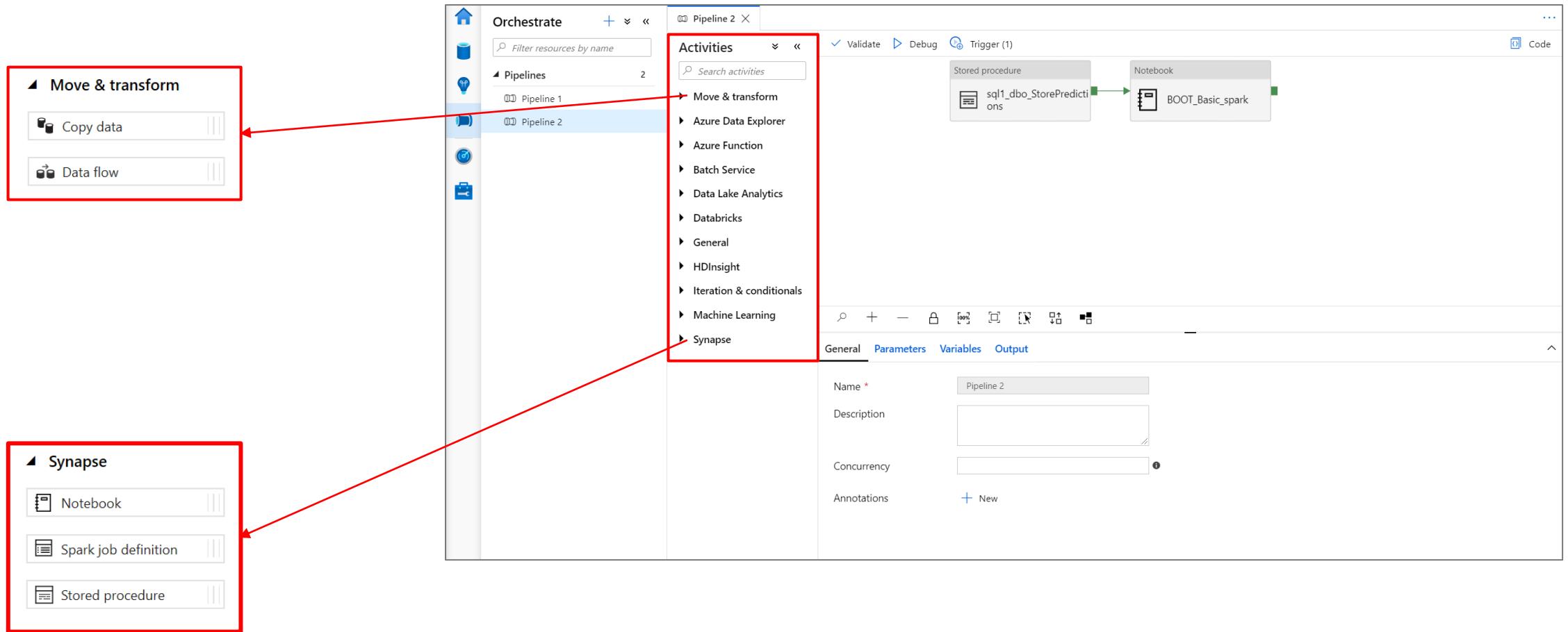


A screenshot of a Jupyter Notebook titled 'Notebook 1'. The top bar shows 'Run all', 'Undo', 'Publish', 'Outline', 'Attach to SparkPool01', 'Language Spark (Scala)', and a status message 'Ready'. The code cell contains the following PySpark code:

```
1 val df = spark.read.synapsesql("SQLPool01.wwi.Date") //used to be: spark.read.sqlanalytics  
2
```

Transform with Pipelines

Orchestrate transformations with Synapse Pipelines.



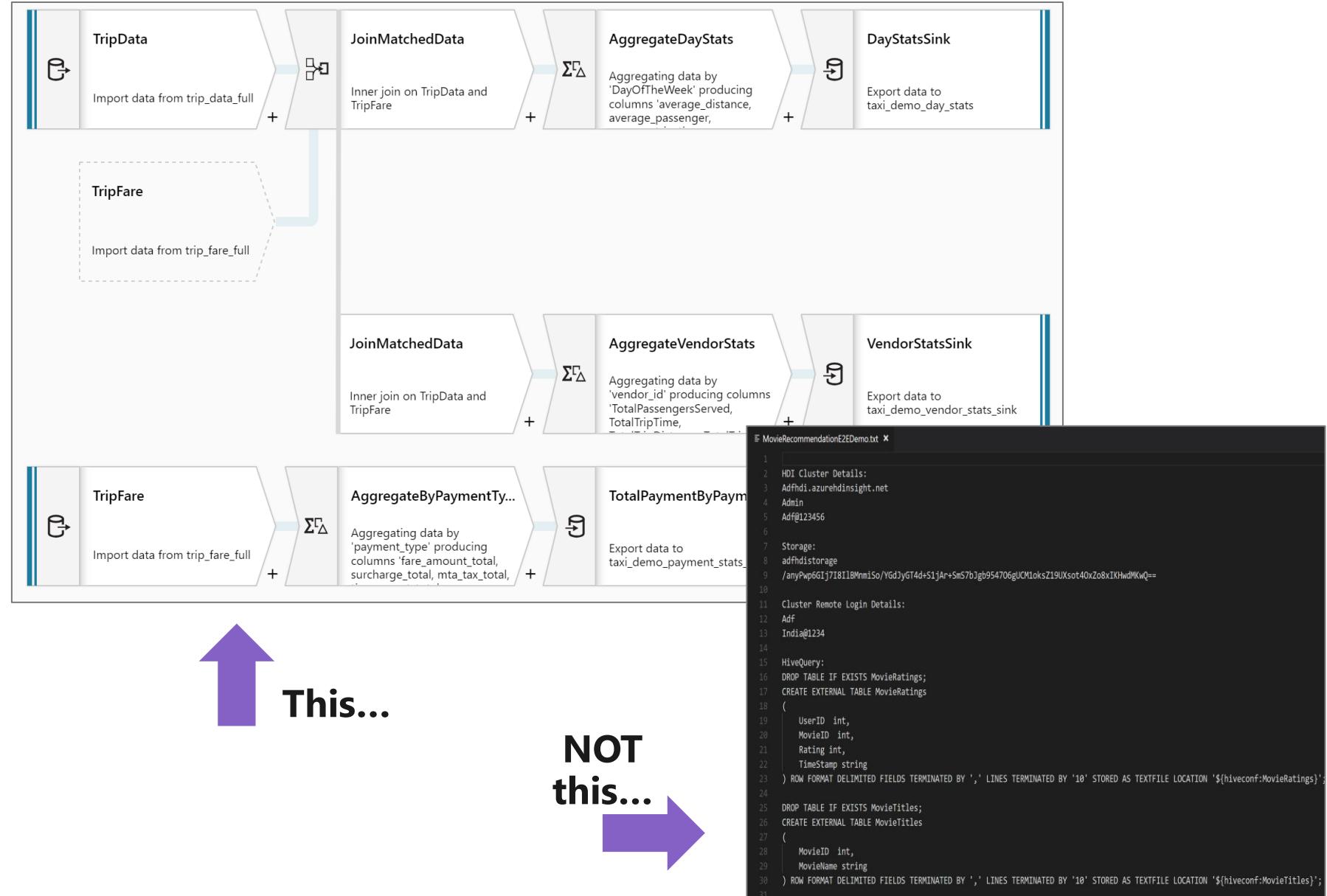
No Code Transform with Data Flows

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

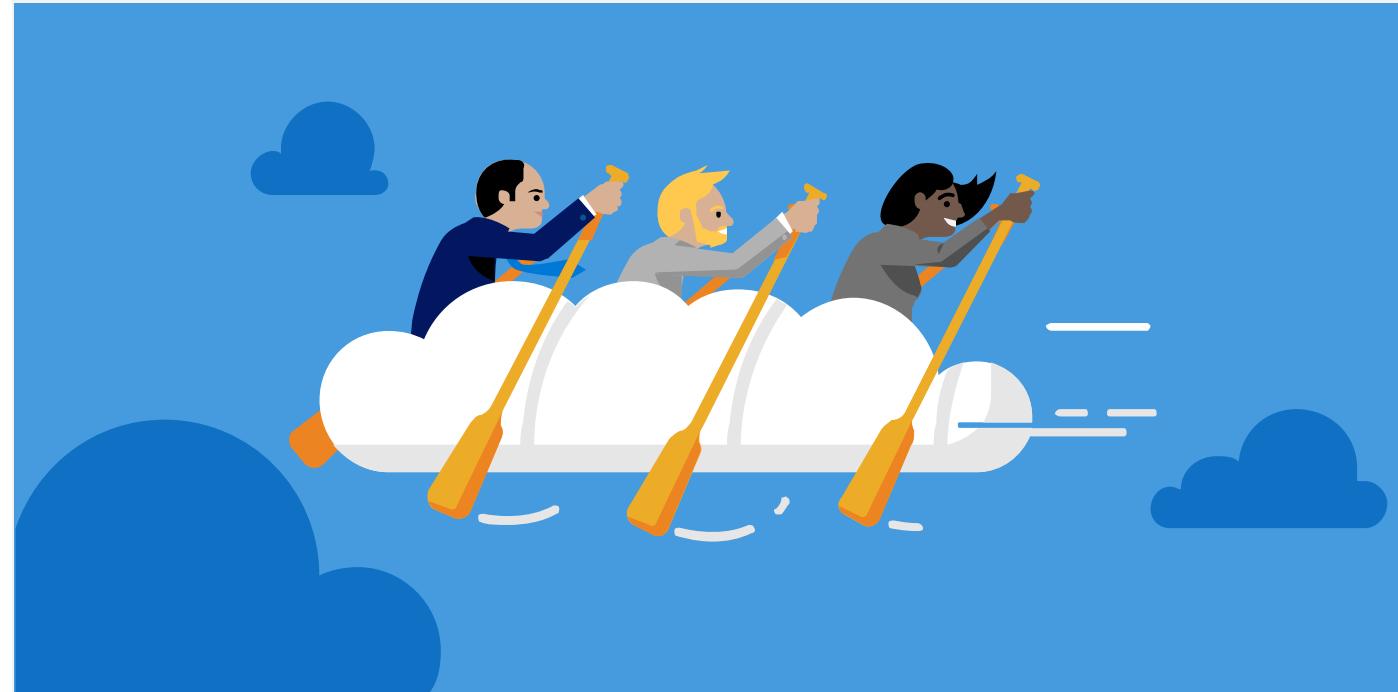
Pop Quiz 1

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

A)
100TB

B)
1PB

C)
10PB



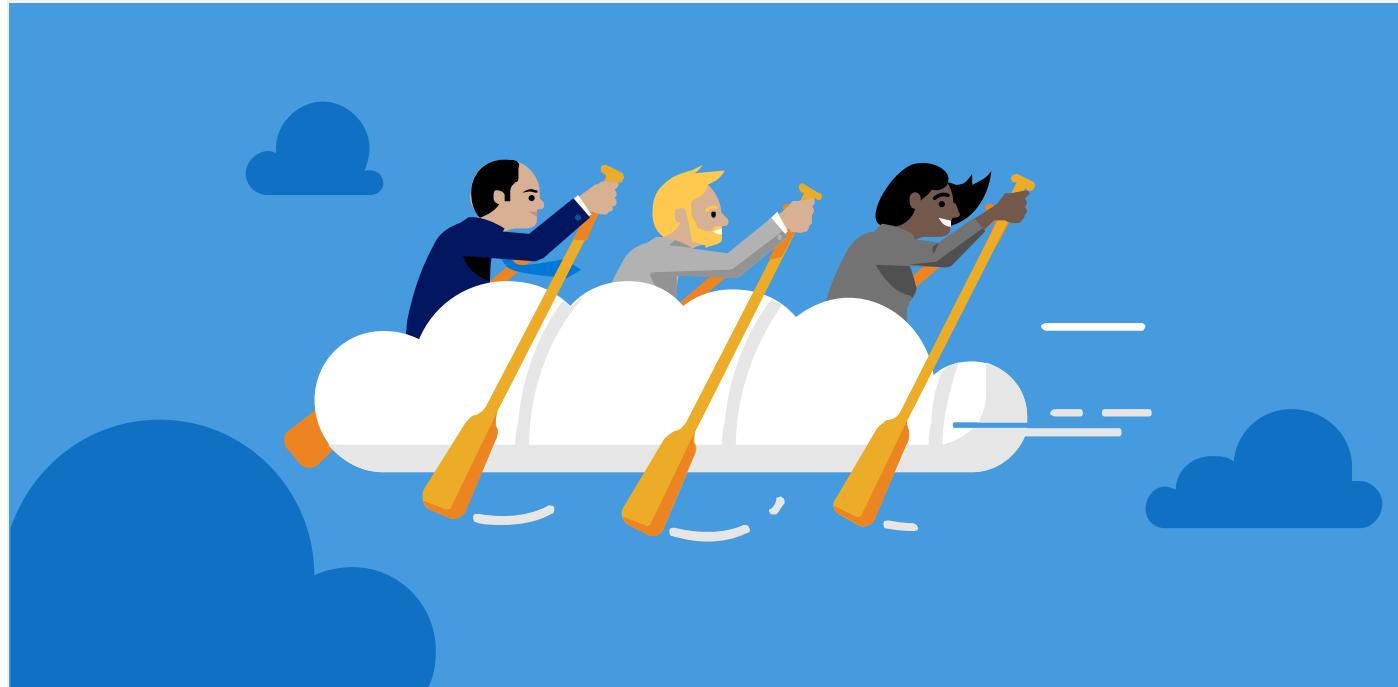
Pop Quiz 1

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

A)
100TB

**B)
1PB**

C)
10PB



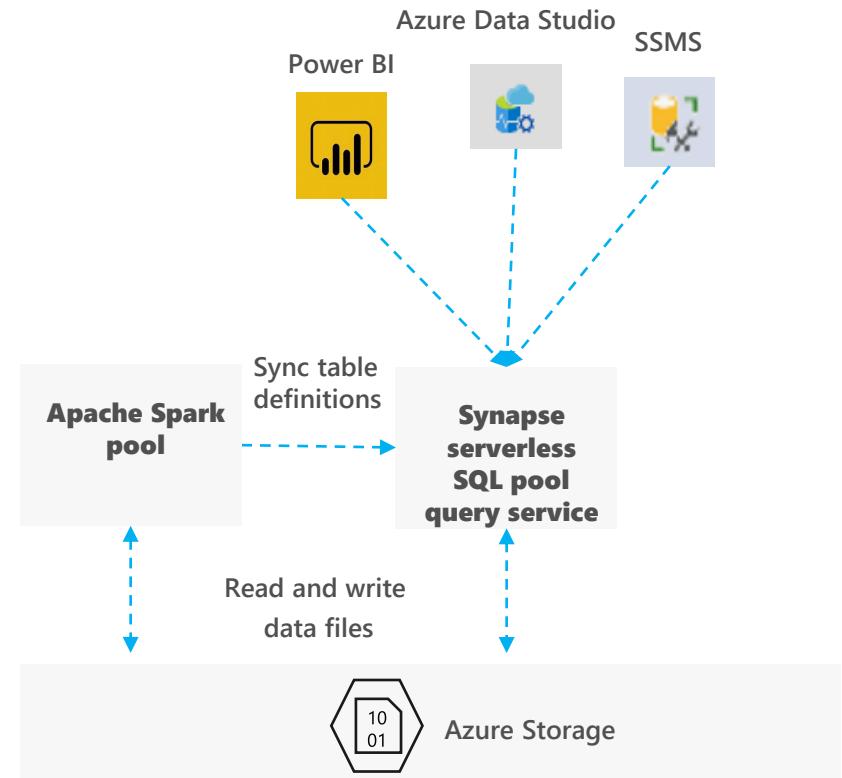
serverless SQL pool

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

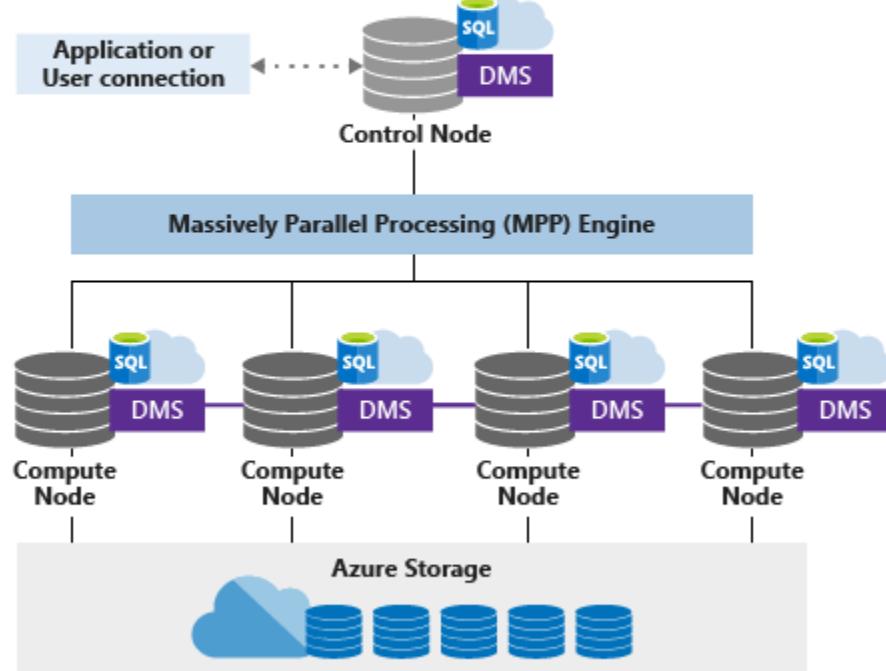
- Model raw files as virtual tables and views
- Use any tool that works with SQL to analyze files
- Use enterprise-grade security model

Easy data transformation

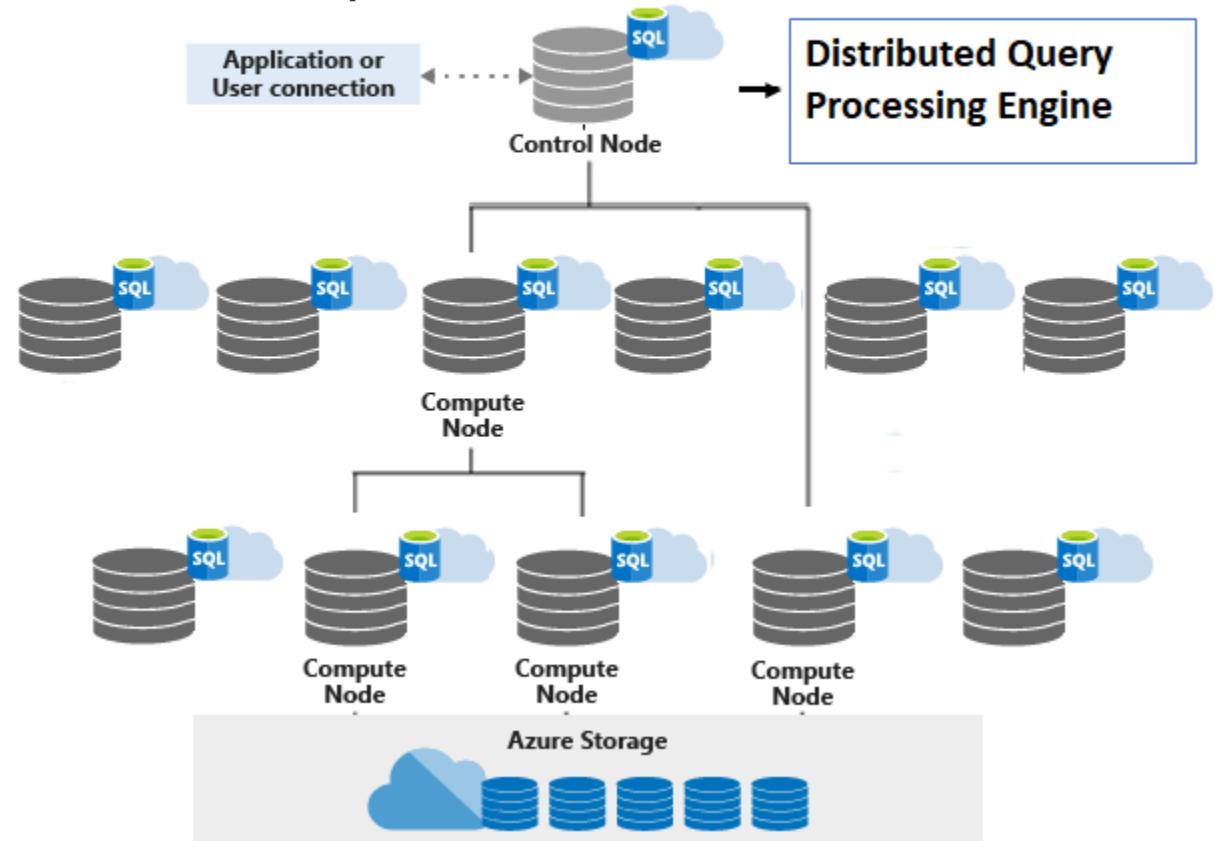
- Transform CSV to parquet format
- Move data between containers and accounts
- Save the results of queries on external storage

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 Storage account structure under the 'internalsandboxwe' dataset. A specific folder named 'opendataset' is selected, containing sub-folders like '_SUCCESS' and several parquet files. One file, 'part-00001-bd1aba93-a85a-4909-8bf4-f79afb6c946f-c000.snappy.parquet', is highlighted and has its properties displayed in a context menu.

Right Panel (SQL Editor): Displays an open SQL script titled 'SQL script 1'. The script uses a BULK OPENROWSET command to query a parquet file located at a specified URL. The 'Connect to' dropdown is set to 'SQL on-demand'.

Bottom Panel (Results): Shows the results of the executed query in a table format. The output consists of four rows of data, each representing a trip record with columns: VENDORID, TPEPICKUPDATETIME, TPEPDROPOFFDATETIME, PASSENGERCOUNT, TRIPDISTANCE, PULOCATIONID, and DOLOCATIONID.

VENDORID	TPEPICKUPDATETIME	TPEPDROPOFFDATETIME	PASSENGERCOUNT	TRIPDISTANCE	PULOCATIONID	DOLOCATIONID
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

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

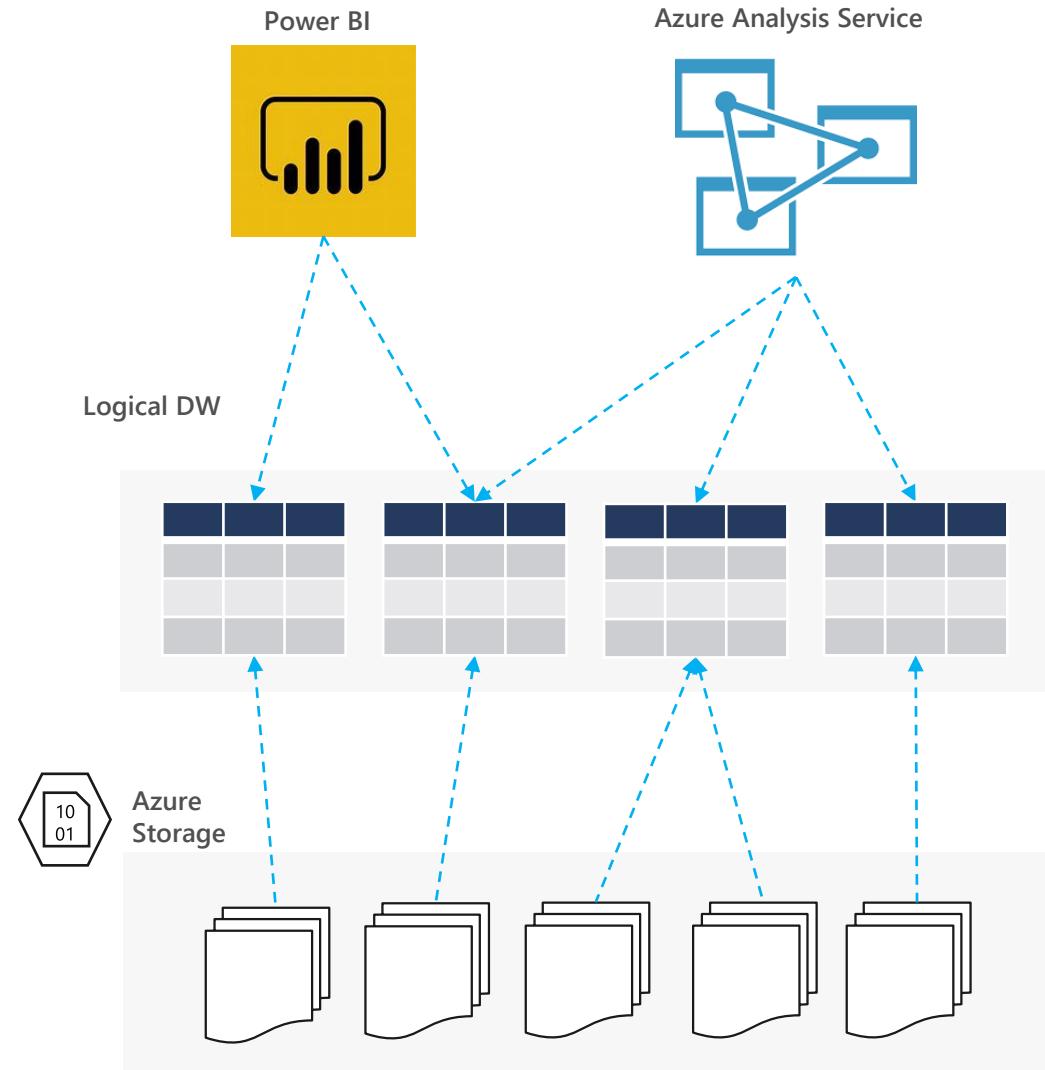
Synapse serverless SQL pool as a logical data warehouse

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

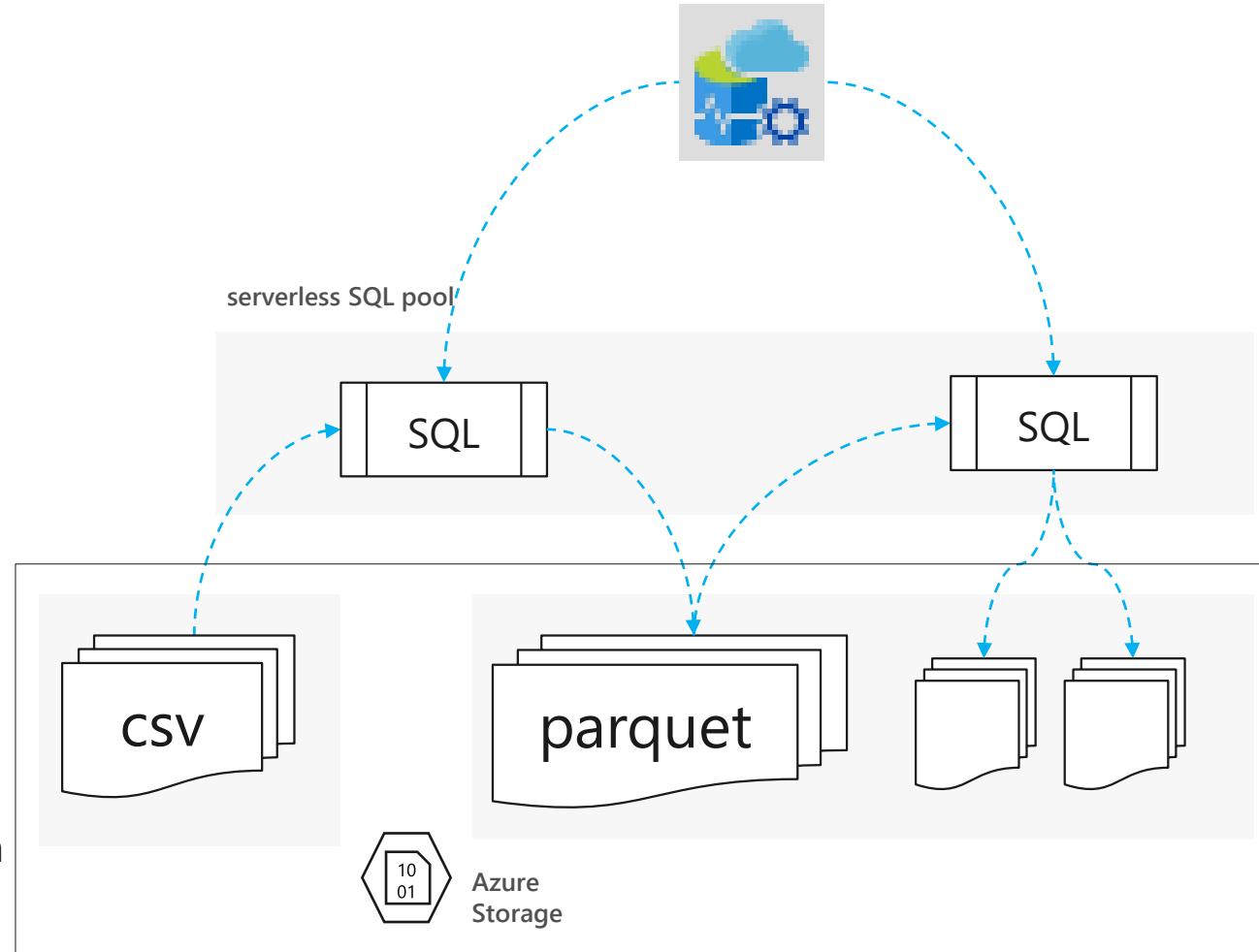
Benefits

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

The diagram illustrates the process of creating an external table from a dataset in Azure Synapse. It consists of three main components:

- Synapse UI:** Shows a dataset named "part-00000-5ae12a71-d27d-4e3a-a686-3bfb7d67c2c9-c000.snappy.parquet". A red arrow points from the "Create external table" option in the context menu to the "Create external table" dialog.
- Create external table Dialog:** This dialog is titled "Create external table" and contains the following fields:
 - Select SQL pool: Built-in (selected)
 - Select a database: SQLServerlessDB
 - External table name: adls.retailsales1 (highlighted with a red box)
 - Create external table: Using SQL script (selected)A red arrow points from the "Using SQL script" radio button to the generated SQL script on the right.
- Generated SQL Script:** The script creates an external table named "adls.retailsales1" with the following definition:

```
1 IF NOT EXISTS (SELECT * FROM sys.external_file_formats WHERE name = 'SynapseParquetFormat')
2 CREATE EXTERNAL FILE FORMAT [SynapseParquetFormat]
3 WITH ( FORMAT_TYPE = PARQUET)
4 GO
5
6 IF NOT EXISTS (SELECT * FROM sys.external_data_sources WHERE name = 'default_azureSynapseA_dfs_core_windows_net')
7 CREATE EXTERNAL DATA SOURCE [default_azureSynapseA_dfs_core_windows_net]
8 WITH (
9     LOCATION = 'https://azuresynapsesa.dfs.core.windows.net/default',
10 )
11 Go
12
13 CREATE EXTERNAL TABLE adls.retailsale (
14     [storeId] varchar(8000),
15     [productCode] varchar(8000),
16     [quantity] varchar(8000),
17     [logQuantity] varchar(8000),
18     [advertising] varchar(8000),
19     [price] varchar(8000),
20     [weekStarting] varchar(8000),
21     [id] varchar(8000)
22 )
23 WITH (
24     LOCATION = 'Parquet/part-00000-5ae12a71-d27d-4e3a-a686-3bfb7d67c2c9-c000.snappy.parquet',
25     DATA_SOURCE = [default_azureSynapseA_dfs_core_windows_net],
26     FILE_FORMAT = [SynapseParquetFormat]
27 )
28 GO
29
30 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 is a file browser titled 'CONNECTIONS' showing 'Servers' and 'Databases'. Under 'databases', there are two entries: 'dbo.data1017 (External)' and 'dbo.data1017 (External)'. The 'Columns' folder under 'dbo.data1017 (External)' is selected, displaying a list of columns: ExtractId, DayOfWeekID, DayOfWeekDescr, DayOfWeekDescrShort, ExtractDateTime, LoadTS, and DeltaActionCode. The right pane contains a 'Create external table' dialog at the top, followed by a 'Cell 1' editor window. The code in 'Cell 1' is:

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

Below this, a SQL query is run in a cell:

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

The results of the query are displayed in a table:

ExtractId	DayOfWeekID	DayOfWeekDescr	DayOfWeekDescrShort	ExtractDateTime
6b86b273ff34fce19d6b804eff5a...	1	Sunday	Sun	2020-01-22 00:00:00.000
d4735e3a265e16eee03f5a718h9b...	2	Monday	Mon	2020-01-22 00:00:00.000
4e07408562bedb8b60ce05c1aect...	3	Tuesday	Tue	2020-01-22 00:00:00.000
4b227777d4dd1fc61c6f884f4864...	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
70000000-0000-0000-0000-000000000000	7	Saturday	Sat	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)
```

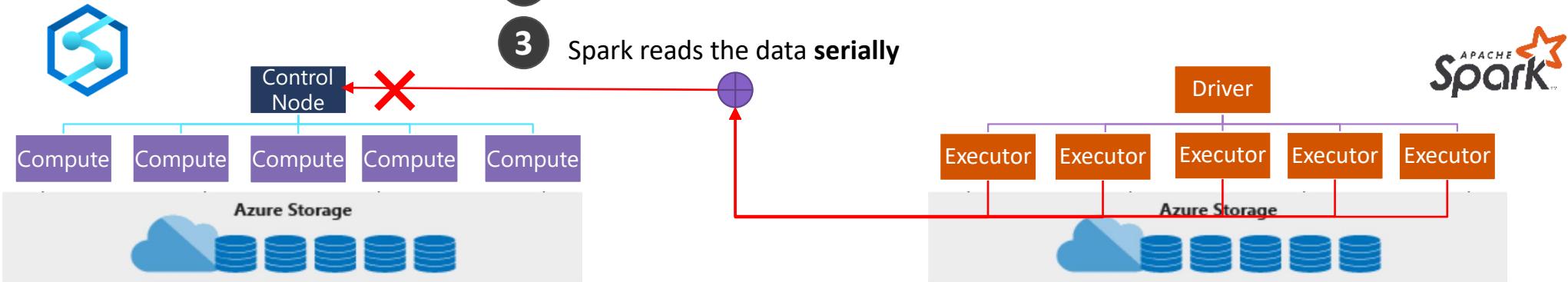
New Approach Using Scala

```
// 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")
```

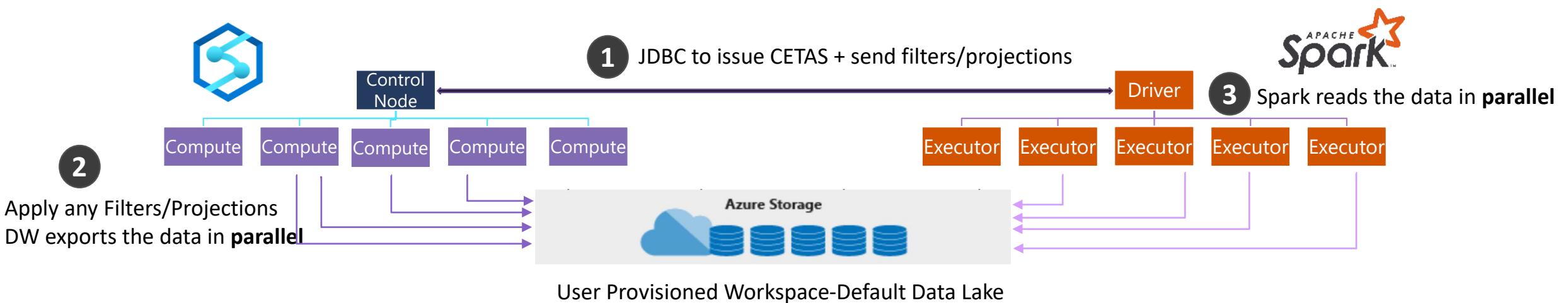
Using Python

```
%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. The left pane shows the Azure portal navigation bar and the Data section selected. The main area displays a storage account named 'nyctic' with two containers: 'green' and 'puYear=2009'. Inside 'green', there is a file named 'part-00055...c000.snappy.parquet'. A context menu is open over this file, with the 'New notebook' option highlighted by a red box and arrow.

The right pane shows the Synapse Analytics workspace. It includes a Data blade, a Pipeline blade, a Data flow blade, and a Notebook blade titled 'Notebook 4 *'. The Notebook blade contains the following PySpark code:

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

Below the code, the job execution status is shown:

ID	DESCRIPTION	STATUS	STAGES	TASKS	SUBMISSION TIME	DURATION
Job 0	load at NativeMethodAccessorImpl.java:0	Succeeded	1/1	1	11/14/2019, 9:56:49 AM	7s
Job 1	showString at NativeMethodAccessorImpl.java:0	Succeeded	1/1	1	11/14/2019, 9:56:58 AM	1s
Job 2	showString at NativeMethodAccessorImpl.java:0	Succeeded	1/1	1	11/14/2019, 9:56:59 AM	11s

Finally, the data preview section shows the schema and some sample rows of the parquet file:

vendorID	tpepPickupDateTime	tpepDropoffDateTime	passengerCount	tripDistance	puLocationId	doLocationId	startLon	startLat	endLon	endLat
1	2015-02-28 23:53:18	2015-03-01 00:00:29	6	1.63	null	null	-74.00084686279297	40.73069381713867	-73.9841537475586	40.74470520019531
1	N	1	7.5 0.5	0.5	0.3	1.76	null	0.0	10.56	
1	2015-03-01 19:21:05	2015-03-28 19:28:31	1	2.2	null	null	-73.97765350341797	40.763160705566406	-73.95502471923828	40.78600311279297
1	N	1	8.5 0.0	0.5	0.3	2.3	0.0	11.6		
1	2015-02-28 23:53:19	2015-03-01 00:12:08	5	3.23	null	null	-73.96012878417969	40.76215744018555	-73.9881591796875	40.72818896484375
1	N	1	14.5 0.5	0.5	0.3	4.74	0.0	28.54		
1	2015-03-28 19:21:05	2015-03-28 19:37:02	1	2.1	null	null	-73.98143005371094	40.7815055847168	-74.000891552734375	40.76177215576172

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The top navigation bar includes 'Microsoft Azure', 'Synapse Analytics', and the workspace name 'euang-synapse-nov-ws'. A search bar at the top right says 'Search resources'.

The left sidebar has sections for 'Develop' (Notebooks, 13), 'Data', 'Machine Learning', and 'Analytics'. A red arrow points from the text 'ts in mat' to the 'Develop' section.

The main area shows a notebook titled 'SeattleSafetyDoc' with several cells:

- Cell 1:** Contains PySpark code to set up Azure storage access and read data from a blob container. It includes a command history entry: "Command executed in 2mins 18s 412ms by euang on 11-22-2019 00:44:52.415 -08:00". Below it is a table for 'Job execution' with one job listed.
- Cell 2:** Contains a single line of PySpark code: `seasafety_df.createOrReplaceTempView('seattlesafety')`. It includes a command history entry: "Command executed in 2s 835ms by euang on 11-22-2019 00:53:37.321 -08:00".
- Cell 3:** Contains a single line of PySpark code: `display(spark.sql('SELECT * FROM seattlesafety LIMIT 10'))`. It includes a command history entry: "Command executed in 23s 901ms by euang on 11-22-2019 00:54:07.313 -08:00". Below the cell is a table view of the data.
- Cell 4:** Contains a single line of PySpark code: `seasafety_df.coalesce(1).write.csv('abfss://default@euangsynapsenovstorage.dfs.core.windows.net/demodata/seattlesafety', mode='overwrite')`.

The table data for Cell 3 is as follows:

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

View results in table format

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

Search resources

Publish all Validate all Refresh Discard all

Develop Notebooks 13

NYCTaxi_Docs_Final * SeattleSafetyDoc * Repro * PySpark (Python)

Cell 1 [3]

```

1 # Azure storage access info
2 blob_account_name = "azurereadystorage"
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.%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

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

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

Apply Cancel

Cell 4 [7]

```
1 seasafety_df.coalesce(1).write.csv('abfss://default@euangsynapsenovstorage.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 STATICRC60
 - 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

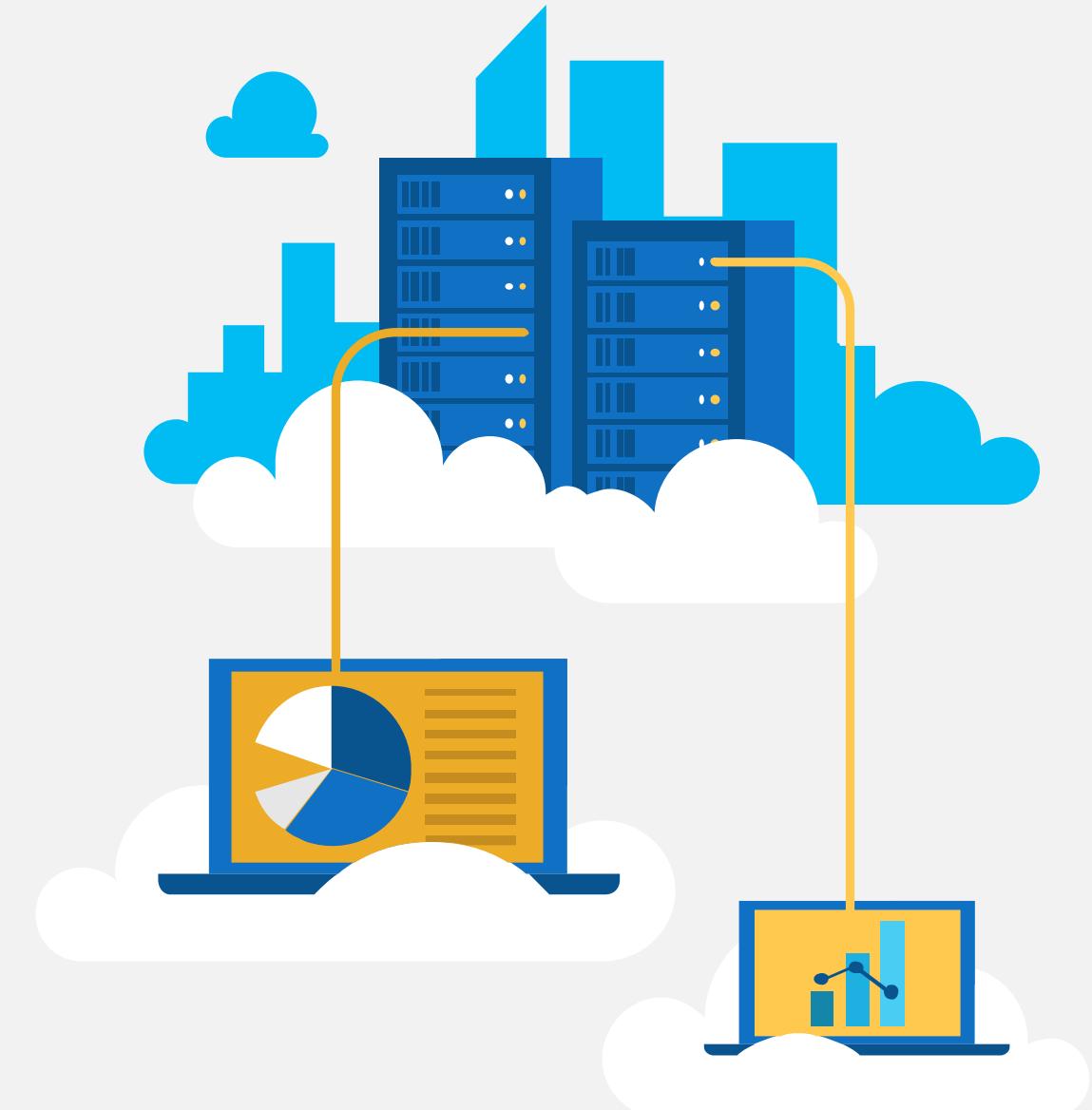
Pop Quiz 2

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



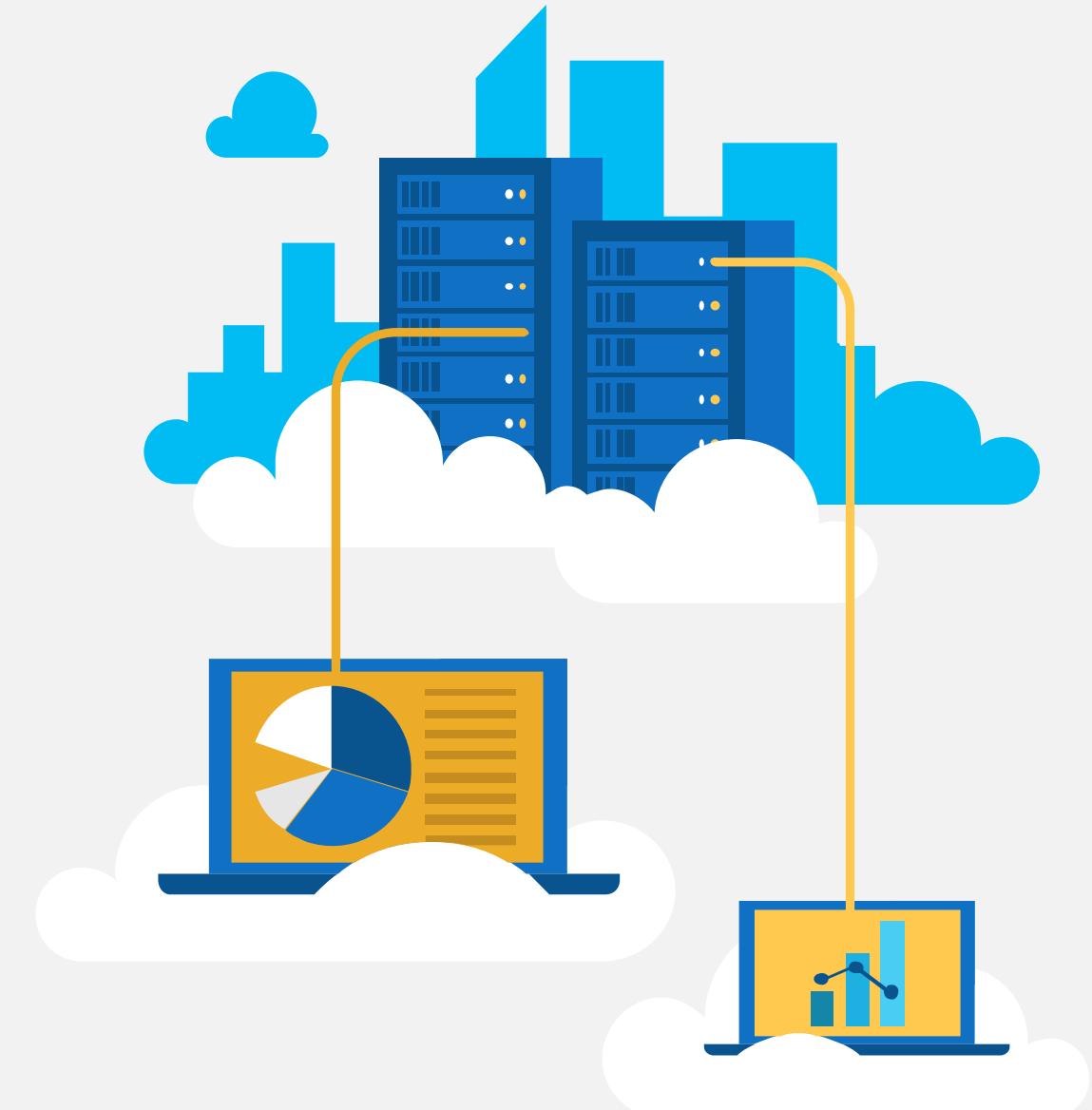
Pop Quiz 2

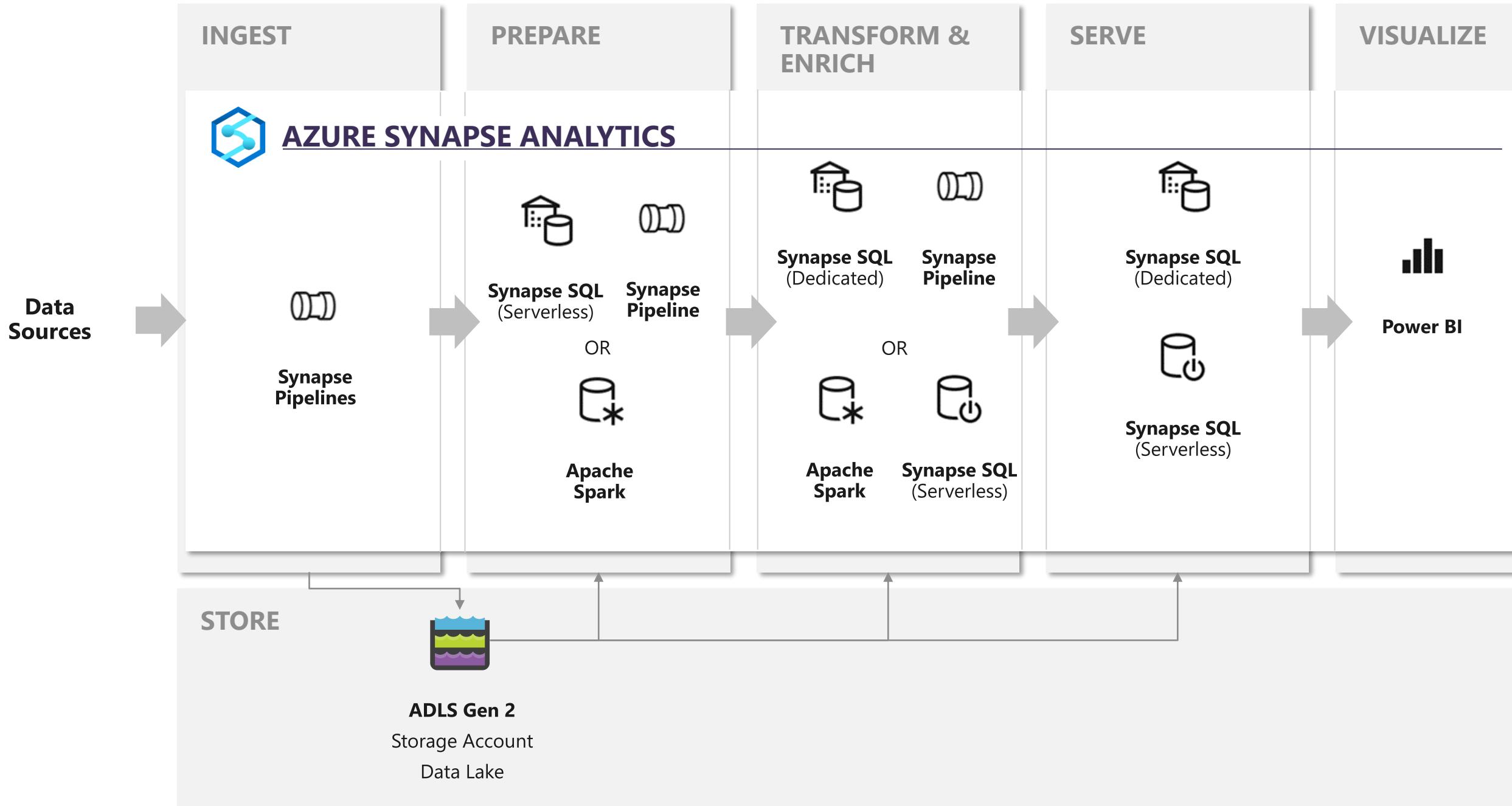
What is the optimal size for a rowgroup in columnstore format in a Synapse SQL Pool?

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

6142 6491



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

