



Cloud Innovator  
**MEGAZONE**

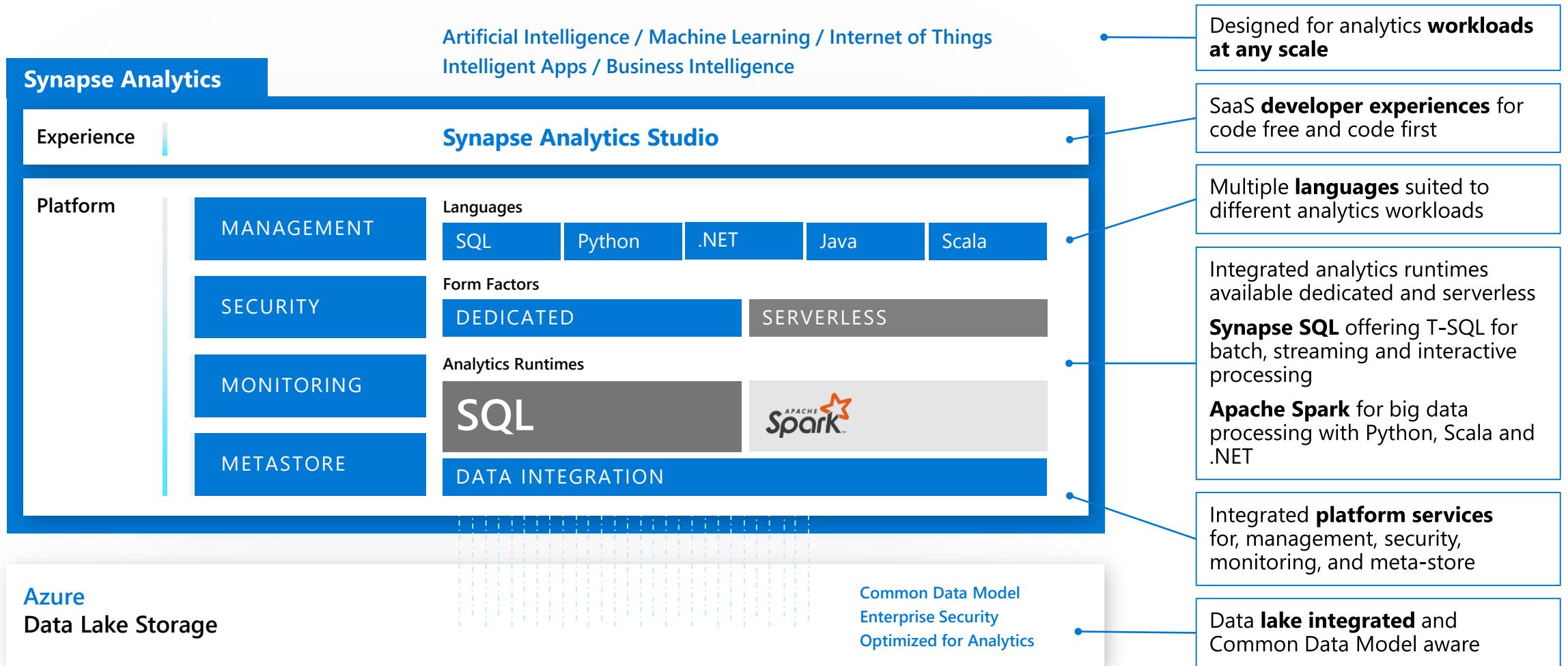


# Azure Synapse Analytics



# Azure Synapse Analytics

통찰력을 갖춘 분석 서비스





# Sections

- Studio
- Integrate
- Synapse SQL
  - Dedicated SQL Pool
  - Serverless SQL Pool
- Spark Analytics



# Azure Synapse Analytics Studio

# Studio

Studio는 작업자의 분석과 협업을 위한 작업 공간입니다.

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 ▾

# Synapse Studio

Synapse Studio는 작업을 위해 각 Activity hubs 로 나뉘고, 각 Activity hub는 분석을 위한 하위 Activity 들로 구성되어 있습니다.

The screenshot shows the Microsoft Azure Synapse Studio interface. The top navigation bar includes 'Microsoft Azure', 'Synapse Analytics', and a workspace dropdown ('wsazuresynapseanalytics'). The main area displays a 'Synapse workspace' titled 'wsazuresy' with a 'New' button. On the left, a red box highlights the 'Home', 'Data', 'Develop', 'Integrate', 'Monitor', and 'Manage' buttons in the sidebar. A red arrow points from the 'Integrate' button to the 'Integrate' section in the center. The center area features six activity hubs: 'Home', 'Data', 'Develop', 'Integrate', 'Monitor', and 'Manage'. Each hub has a brief description and a corresponding icon. The 'Recent resources' sidebar on the left lists several items, including '05 Sentiment\_Anal', 'Predict NYCTaxi Tri...', '001 SQL Pool Secu...', '005 Predict In-Eng...', and '05 Anomaly\_Detect...'. At the bottom, a footer states 'Copyright 2021 © MEGAZONE & MEGAZONECLOUD CORP. ALL RIGHT RESERVED.'

Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics

Synapse workspace  
wsazuresy

New ▾

Ingest  
Perform a scheduled

Recent resources

Name

05 Sentiment\_Anal

Predict NYCTaxi Tri...

001 SQL Pool Secu...

005 Predict In-Eng...

05 Anomaly\_Detect...

Show more ▾

Home

Quick-access to common gestures, most-recently used items, and links to tutorials and documentation.

Data

Explore structured and unstructured data

Develop

Write code and define business logic of the pipeline via Notebooks, SQL scripts, Data flows, etc.

Integrate

Design pipelines that move and transform data.

Monitor

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

Manage

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

Copyright 2021 © MEGAZONE & MEGAZONECLOUD CORP. ALL RIGHT RESERVED.

# Home Hub

Synapse를 구성하는 Resource 업데이트 릴리즈, 작업공간 전환, 알림, 피드백 제공에 쉽게 접근 가능합니다.

Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics

someone@microsoft.com MICROSOFT

Home Data Develop Integrate Monitor Manage

Synapse workspace

wsazuresynapseanalytics

Updates Switch Workspaces Notifications Feedback

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



# Synapse Studio Home hub

# Home Hub

사용자들의 학습을 위해 데이터 수집 작업, 학습 문서로의 링크, 샘플 데이터를 제공합니다.

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 the 'wsazuresynapseanalytics' workspace. A red box highlights the 'Ingest', 'Explore and analyze', 'Visualize', and 'Learn' sections. Below this, the 'Recent resources' section lists several items, each with a file icon, name, and last opened time.

**Synapse workspace**  
**wsazuresynapseanalytics**

**New**

**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.

**Learn**  
Start with Azure Open Datasets and sample code.

**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 ▾

# Home Hub - Learn

Knowledge Center는 사용자에게 학습 할 수 있는 샘플 데이터를 제공합니다.

The screenshot shows the Microsoft Azure Synapse Analytics Knowledge Center page. At the top, there is a navigation bar with icons for notifications, search, and help, followed by the email address 'someone@microsoft.com' and the Microsoft logo. The main content area has a blue header 'Knowledge center'. Below it, a message says 'Get started with Azure Open Datasets and sample code. Return to the Knowledge center periodically as we provide updated content.' To the left, a sidebar lists several icons: a house (Home), a cylinder (Data), a document (Samples), a folder (Pools), a gear (Studio), and a briefcase (Samples). A callout box labeled 'Learn' with the sub-instruction 'Start with Azure Open Datasets and sample code.' is connected by an arrow to the 'Samples' icon in the sidebar. The central part of the page features a large image of a 3D bar chart with a line graph overlaid, representing data analysis. To the right of this image is a section titled 'Experience limitless scale' with the sub-instruction 'Deliver insights from all your data, across data warehouses and big data analytics systems, with blazing speed.' Below this are three smaller sections: 'Use samples immediately' (with the sub-instruction 'Click once and we'll create everything you need, from scripts and notebooks to pools and data.'), 'Browse gallery' (with the sub-instruction 'Select from sample code and Azure Open Datasets to quickly get started in your workspace.'), and 'Tour Synapse studio' (with the sub-instruction 'Familiarize yourself with key features of Synapse Studio. Start by taking a tour of the homepage.'). At the bottom of the page, there is a copyright notice: 'Copyright 2021 © MEGAZONE & MEGAZONECLOUD CORP. ALL RIGHT RESERVED.'

# Knowledge center

Knowledge Center는 간편한 시작과 학습을 위한 open datasets, sample Notebooks, SQL scripts and pipeline templates을 제공합니다.

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

[Continue](#) [Close](#)

# Home Hub

사용자가 Synapse Analytics의 기능을 학습할 수 있도록 기술 문서, 커뮤니티 센터로 이동하는 링크를 제공합니다.

**Feature showcase**

Get started with Azure Synapse Analytics

Take a step-by-step tour of Synapse exploring the breadth of capabilities offered.

[Learn more](#)

**Community**

**Azure HDInsight Spark Language Runtime Comparison - Lower is better**

Microsoft and the .NET Foundation announce the release of version 1....  
.NET for Apache Spark is now available in Synapse Analytics.

Oct 26

**Synapse SQL: SQL permissions**

**ADLS: ACL permissions**

**Storage files**

Securing access to ADLS files using Synapse SQL permission model

Define serverless SQL permissions using SQL runtime permission model.

Oct 19

**Transactional Work** (Transactional work and writes)

**Analytical Work** (Analytics work for analytical queries)

**Azure Synapse Link** (Cloud Native Hadoop)

**Azure Cosmos DB** (Container)

**Azure Synapse Analytics** (Machine learning, Big data analysis, BI Dashboards)

Analyze CosmosDB data using Synapse Link and Transact-SQL Ia...

Analyze data and create reports from Cosmos DB.

Oct 16

**Quickly Get Started with Samples in Azure Synapse Analytics**

Be even more productive with the knowledge center in Synapse Studio.

Sep 24

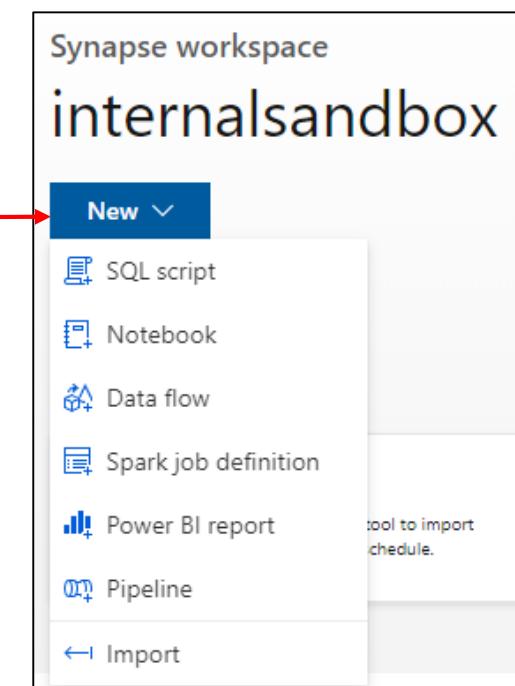
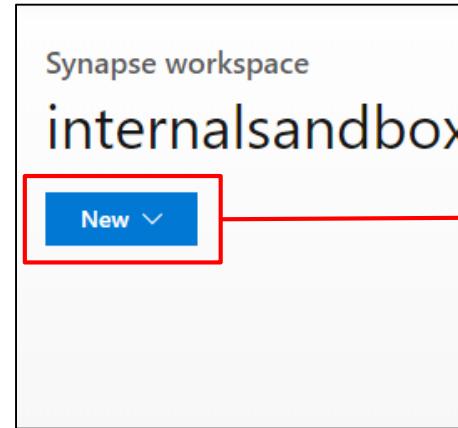
[Visit the community center](#)

# Home Hub

## Overview

**New** dropdown – 빠른 작업을 시작할 수 있도록 단축 키를 제공 합니다.

**Recent & Pinned** – 최근에 사용했던 SQL Script, Spark Code 등 가공물 내역을 나열합니다.



Recent	Pinned	
NAME		LAST OPENED BY YOU
<a href="#">BOOT_AMLautoMLPredict</a>		6 hours ago
<a href="#">SQLConnector</a>		6 hours ago
<a href="#">TaxiCreateSparkTable</a>		6 hours ago
<a href="#">Notebook 1</a>		6 hours ago
<a href="#">NYCTAx</a>		6 hours ago
<a href="#">Show more ▾</a>		

Recent	Pinned	
NAME		LAST OPENED BY YOU
<a href="#">NYCTAx</a>		6 hours ago



# Synapse Studio

## Data hub

# Data Hub

Workspace의 내부 데이터와 Workspace에 Link 된 Storage Account의 데이터를 탐색할 수 있습니다.

The screenshot shows the Microsoft Azure Synapse Analytics Data Hub interface. On the left, there is a sidebar with icons for Home, Data, Develop, Integrate, Monitor, and Manage. The main area has a breadcrumb navigation bar: Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics. Below the breadcrumb is a search bar with filters: Synapse live, Validate all, and a plus sign icon. The main content area is titled 'Data' and has two tabs: 'Workspace' (which is selected and highlighted with a red box) and 'Linked'. A search bar labeled 'Filter resources by name' is present. Under the 'Workspace' tab, there is a section for 'Databases' with 10 items: newpoll (SQL), NYCTaxi\_Pool (SQL), Predict\_Pool (SQL), Streaming\_Pool (SQL), WWI\_Pool (SQL), NYT2020 (SQL), SQLServerlessDB (SQL), and default (Spark). Each item has a small icon and a downward arrow indicating it's a folder.

The screenshot shows the Microsoft Azure Synapse Analytics Data Hub interface with the 'Linked' tab selected (highlighted with a red box). The main content area is titled 'Data' and has two tabs: 'Workspace' and 'Linked' (which is selected and highlighted with a red box). A search bar labeled 'Filter resources by name' is present. Under the 'Linked' tab, there is a section for 'Integration datasets' with 24 items: Azure Blob Storage (3), Azure Cosmos DB (1), Azure Data Explorer (2), Azure Data Lake Storage Gen2 (2), wsazuresynapseanalytics (Primary...) (2), (Attached Containers) (2), and Integration datasets (24). Each item has a small icon and a downward arrow indicating it's a folder.

# Data Hub – Linked Storage

Azure Data Lake Storage Gen2 Accounts의 Filesystems, Azure Data Explorer 의 Clusters, Azure Cosmos DB의 Containers 데이터를 찾아볼 수 있습니다.

The screenshot shows the Microsoft Azure Synapse Analytics Studio interface. On the left, a sidebar titled "Data" lists various linked storage resources:

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

The main workspace shows a file browser for the "rawdata" folder under "taxidata". The file path is displayed as "rawdata > taxidata". The file list shows several cached items:

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

At the bottom, a message indicates "Showing 1 to 6 of 6 cached items".

# Data Hub – Storage accounts

데이터 미리보기 기능을 지원합니다.

The screenshot illustrates the Azure Synapse Studio interface, specifically the 'Data' hub. On the left, the 'Data' hub shows a list of storage accounts and datasets. In the center, a detailed view of the 'rawdata' dataset within the 'wsazuresynapseanalytics' linked service is displayed. A context menu is open over the 'Products.csv' file, with the 'Preview' option highlighted and a red box and arrow pointing to it. To the right, a preview pane shows the contents of the 'Products.csv' file, which contains the following data:

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

A blue 'OK' button is visible at the bottom of the preview pane.

# Data Hub – Storage accounts

파일의 속성 확인이 가능합니다.

The screenshot illustrates the Azure Synapse Studio interface for managing storage accounts. On the left, the 'Data' sidebar shows a 'Linked' workspace with resources like Azure Blob Storage, Azure Cosmos DB, Azure Data Explorer, and Azure Data Lake Storage Gen2. The 'rawdata' dataset is selected in the center workspace area. A context menu is open over the 'Products.csv' file, with the 'Properties...' option highlighted by a red box and a red arrow pointing to the detailed properties dialog on the right.

**Properties Dialog Content:**

- 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: (empty)

Buttons at the bottom: Apply and Cancel.

# Data Hub – Storage accounts

접근 관리 기능 - 폴더와 파일에 관한 standard POSIX ACLs 설정이 가능합니다.

The screenshot illustrates the process of managing access to a file in Azure Synapse Studio. On the left, the 'Data' workspace shows a list of resources, including 'rawdata' under 'wsazuresynapseanalytics'. In the center, the 'rawdata' folder is selected, displaying files like '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 '\$superuser (Owner)' and '\$superuser (Owning Group)', along with an 'Other' section and a 'Mask' section. It also lists 'Permissions for: \$superuser' with checkboxes for 'Read', 'Write', and 'Execute', where 'Read' and 'Write' are checked. Below this, there's a section to 'Add user, group, or service principal' with a text input field 'Enter a UPN or Object ID' and a 'Save' button at the bottom.

# Data Hub – Storage accounts

Storage Account 내부 파일의 데이터를 조회하기 위해 Studio 에서는 SQL scripts 과 Notebook code 를 자동 생성하는 기능이 있습니다.

```

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]
  
```

VendorID	TpepPickupDate	TpepDropoffDate	PassengerCount	TripDistance	PuLocationId	DoLocationId
VTS	2014-04-30T23:5...	2014-05-01T00:1...	1	5.21	NULL	NULL
CMT	2014-04-30T23:5...	2014-05-01T00:2...	1	21.8	NULL	NULL
VTS	2014-04-30T23:5...	2014-05-01T00:3...	1	5.57	NULL	NULL
CMT	2014-04-30T23:5...	2014-05-01T00:3...	3	1.8	NULL	NULL

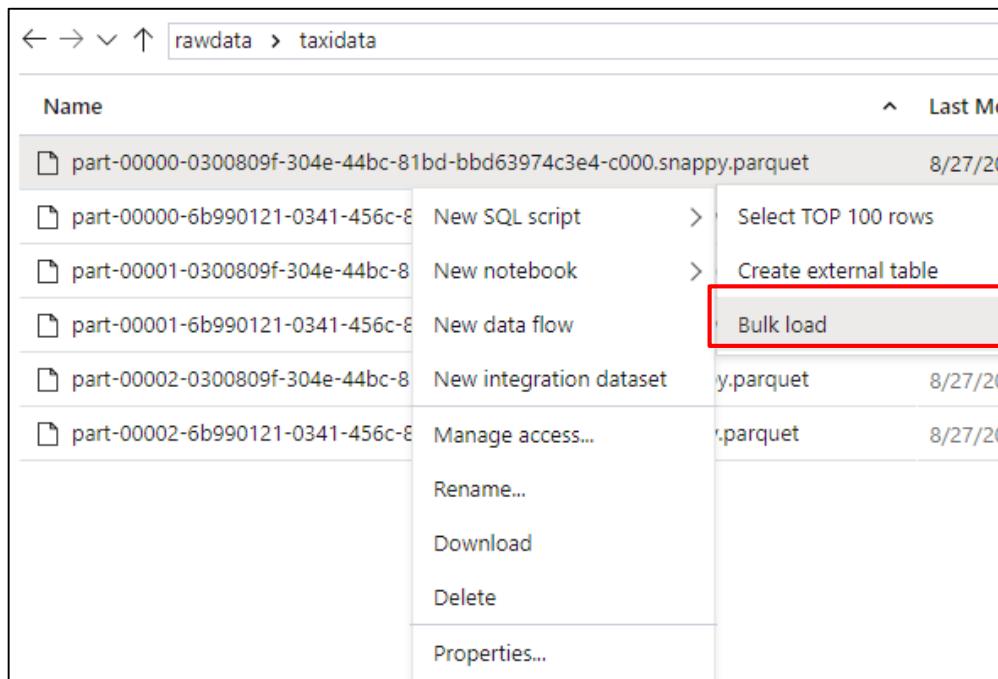
```

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))
  
```

vendorID	tpepPickupDate...	tpepDropoffDat...	passengerCount	tripDistance	puLocationId	doLoc...
VTS	2014-04-30T23:5...	2014-05-01T00:1...	1	5.21		
CMT	2014-04-30T23:5...	2014-05-01T00:2...	1	21.8		
VTS	2014-04-30T23:5...	2014-05-01T00:3...	1	5.57		
CMT	2014-04-30T23:5...	2014-05-01T00:3...	3	1.8		

# Data Hub – Bulk load wizard

- Bulk Load Wizard 기능을 사용하여 Storage Container 또는 Storage File 를 활용한 Data의 Bulk Load가 가능합니다.
- Data load를 위해 자동 T-SQL script 생성을 지원합니다.
- 파이프라인과 통합을 가능하게 하는 Stored Procedure 생성을 제공합니다.



### Bulk load

**Source storage location**  
Select the storage location where the files containing the data is staged. Azure Data Lake Storage (ADLS) Gen2 and Azure Blob Storage are supported. [Learn more](#)

**Storage account**  
wsazuresynapseanalytics-WorkspaceDefaultStorage

**Connect via integration runtime \***  
AutoResolveIntegrationRuntime

**Input file or folder** rawdata/taxidata/part-00000-0300809f-304e

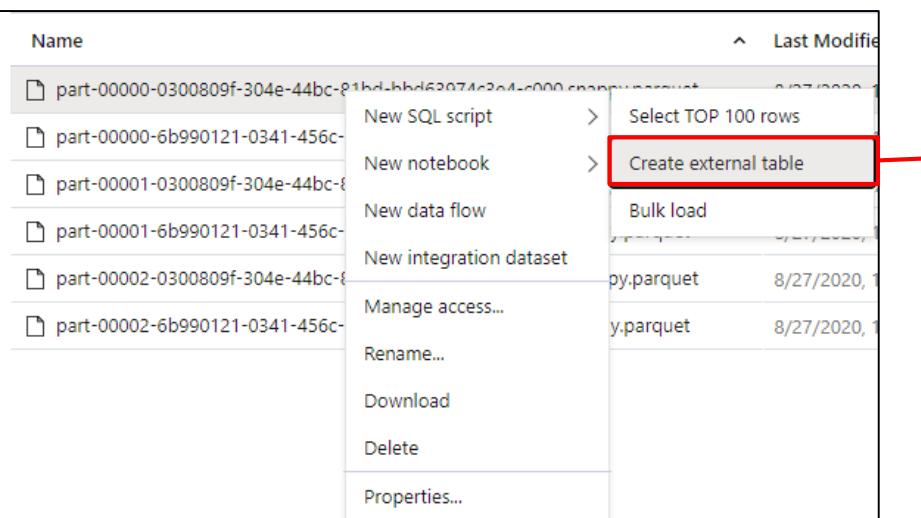
```

1 --Uncomment the 4 lines below to create a stored procedure for da
2 --CREATE PROC bulk_load_test
3 --AS
4 --BEGIN
5 COPY INTO dbo.test
6 (vendorID 1, tpepPickupDateTime 2, tpepDropoffDateTime 3, passeng
7 FROM 'https://azuresynapsesa.dfs.core.windows.net/rawdata/taxidat
8 WITH
9 (
10     FILE_TYPE = 'PARQUET'
11     ,MAXERRORS = 0
12     ,IDENTITY_INSERT = 'OFF'
13 )
14 --END
15 GO
16
17 SELECT TOP 100 * FROM test
18 GO

```

# Data Hub – Storage accounts

- External File Format, External File 원본 위치 지정 및 External Table을 자동으로 만들어 생산성을 향상 시킬 수 있습니다.
- External File Format, External File 원본 위치 및 External Table을 생성하는 스크립트를 제공하여 수동으로 실행 가능합니다.



**Create external table**

part-00000-0300809f-304e-44bc-81b...

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

Select SQL pool \* ①

NYCTaxi\_Pool

Select a database \* ①

NYCTaxi\_Pool

External table name \* ①

dbo.ast\_nyc

Create external table \*

Automatically

Using SQL script

This will automatically create the external table in your database where you can quickly SELECT Top 100 in your SQL script

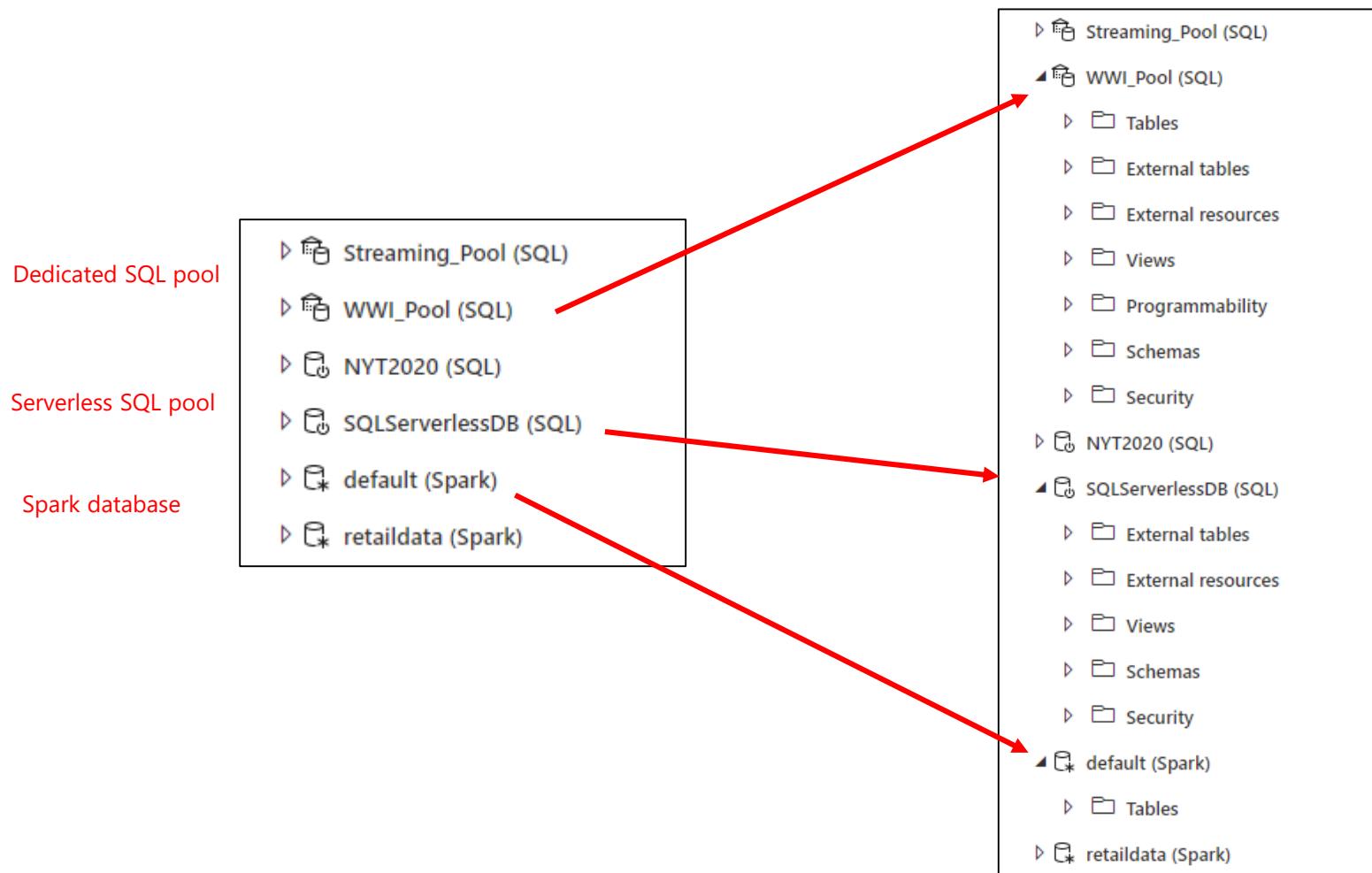
**Create**    **Cancel**

```
Run Undo Publish Query plan Connect to NYCTaxi_Pool
1 SELECT TOP 100 * FROM dbo.ast_nyc
2 GO
```

```
Run Undo Publish Query plan Connect to NYCTaxi_Pool Use database NYCTaxi_Pool
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 = 'rawdata_azureSynapseA_dfs_core_windows_net')
7 CREATE EXTERNAL DATA SOURCE [rawdata_azureSynapseA_dfs_core_windows_net]
8 WITH (
9     LOCATION = 'abfss://rawdata@azuresynapsesa.dfs.core.windows.net',
10    TYPE = HADOOP
11 )
12 GO
13
14 CREATE EXTERNAL TABLE dbo.ast_nyc (
15     [vendorID] varchar(8000),
16     [tpepPickupDateTime] datetime2(7),
17     [tpepDropoffDateTime] datetime2(7),
18     [passengerCount] int,
19     [tripDistance] float,
20     [puLocationId] varchar(8000),
21     [doLocationId] varchar(8000),
22     [startLon] float,
23     [startLat] float,
24     [endLon] float,
25     [endLat] float,
26     [rateCodeId] int,
27     [storeAndFwdFlag] varchar(8000),
28     [paymentType] varchar(8000),
29     [fareAmount] float,
30     [extra] float,
31     [mtaTax] float,
32     [improvementSurcharge] varchar(8000),
33     [tipAmount] float,
34     [tollsAmount] float,
35     [totalAmount] float,
36     [puYear] int,
37     [puMonth] int
38 )
39 WITH (
40     LOCATION = 'taxidata/part-00000-0300809f-304e-44bc-81bd-bbd63974c3e4-c000.snappy.parquet',
41     DATA_SOURCE = [rawdata_azureSynapseA_dfs_core_windows_net],
42     FILE_FORMAT = [SynapseParquetFormat],
43     REJECT_TYPE = VALUE,
```

# Data Hub – Databases

Workspace 내부에 존재하는 데이터베이스를 탐색합니다.



# Data Hub – Databases

- 테이블과 같은 Object에서 간단히 데이터를 조회할 수 있는 T-SQL Scripts를 생성할 수 있습니다.

Databases 3

- sql1 (SQL pool)
  - Tables
    - dbo.SearchLogTable
    - dbo.NycTaxiPredict
      - Columns
        - New SQL script
        - New notebook
        - Select TOP 1000 rows
        - CREATE
        - Drop
        - Drop and Create

- SQL Pool의 Table에서 Spark DataFrame으로 데이터를 조회할 수 있는 PySpark Code를 자동 생성할 수 있습니다.

Databases 3

- sql1 (SQL pool)
  - Tables
    - dbo.SearchLogTable
    - dbo.NycTaxiPredict
      - Columns
        - New SQL script
        - New notebook
        - Refresh
        - Load to DataFrame

# Data Hub – Datasets

데이터 세트가 정의되면 파이프라인에서 데이터 소스 또는 데이터 싱크로 사용할 수 있습니다.

The screenshot shows the Azure Synapse Analytics Studio interface. On the left, there is a sidebar titled "Data" with a search bar and a list of 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 configuration page. The main area displays the "NYCTaxiParquet" dataset details. At the top, there is a title bar with the dataset name and a "Code" button. Below the title bar, there is a Parquet file icon and the dataset name. The configuration page includes tabs for General, Connection, Schema, and Parameters. Under the Connection tab, the "Linked service" dropdown is set to "Lake\_ArcadiaLake". There are buttons for "Test connection", "Open", and "New". The "File path" field is set to "data / nyctaxi / File", with "Browse" and "Preview data" buttons. The "Compression type" is set to "snappy".



# Synapse Studio

## Develop hub

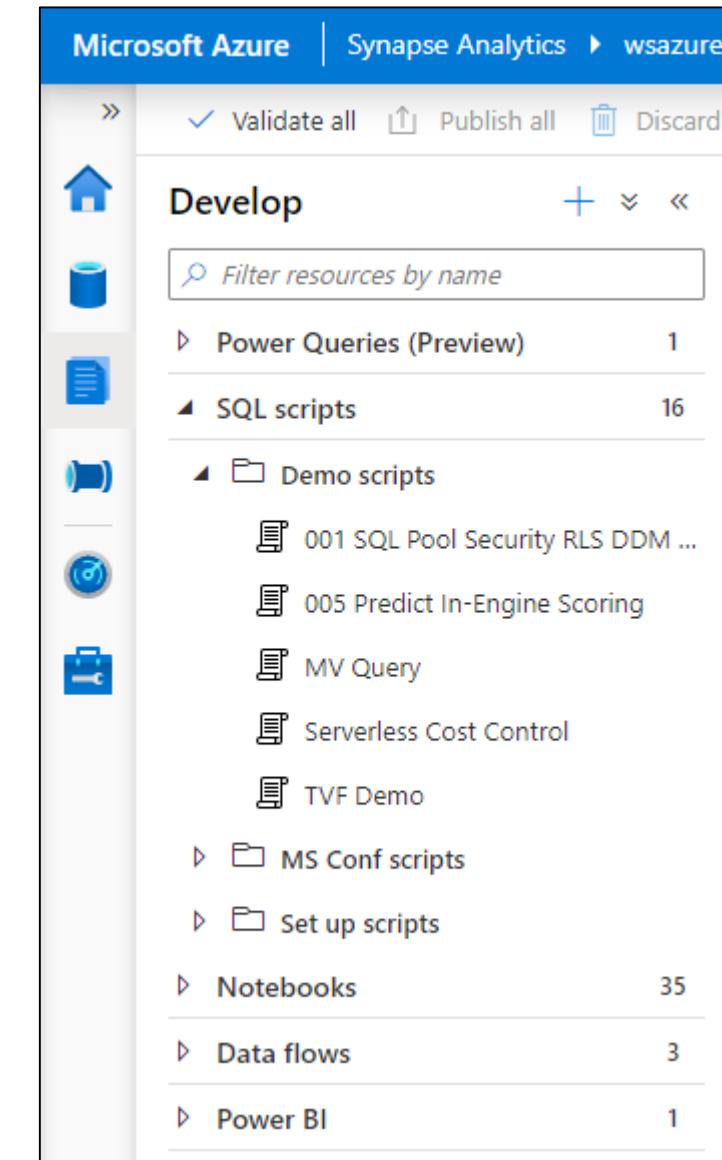
# Develop Hub

## Overview

Data Hub는 쿼리, 분석, 모델 데이터에 관한 개발 환경을 제공합니다.

## Benefits

- 개발 Hub에서 다양한 언어를 사용하여 데이터 분석이 가능합니다.
- SQL Script와 Notebook Code를 손쉽게 전환하며 분석할 수 있습니다.
- SQL Script와 Notebook Code 작성에 최적화되어 있습니다.
- 데이터를 시각화하여 간단하게 볼 수 있습니다.

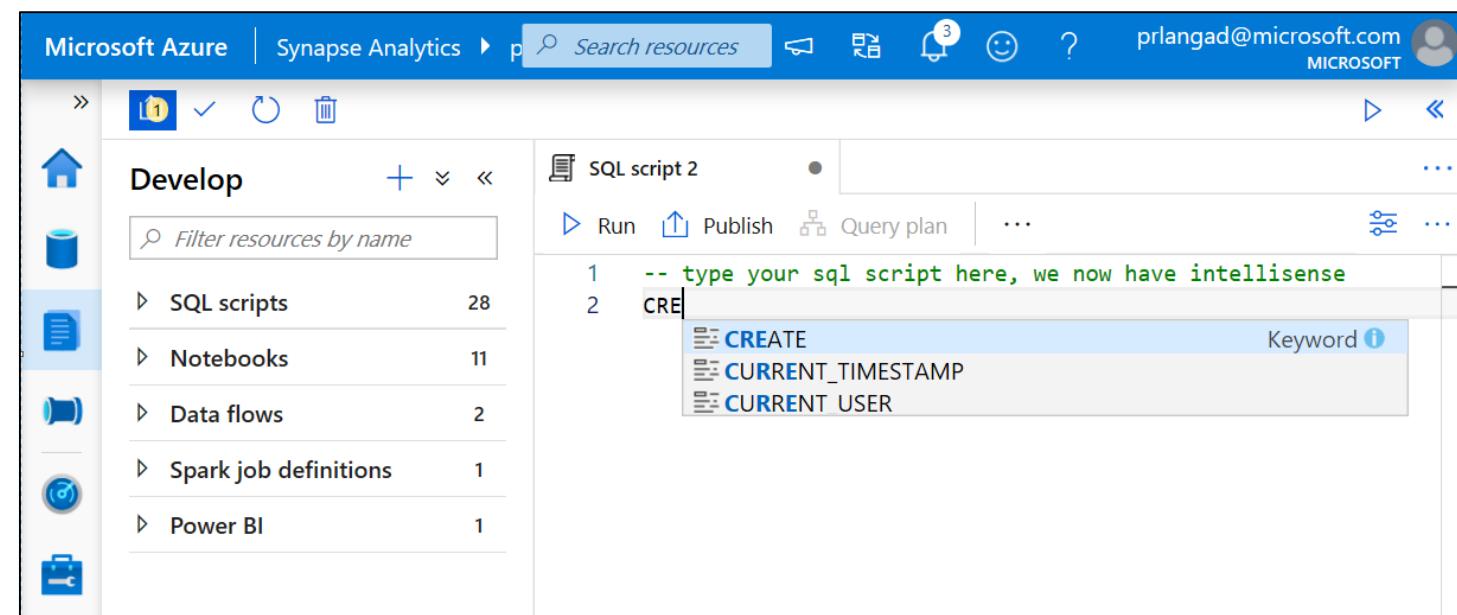
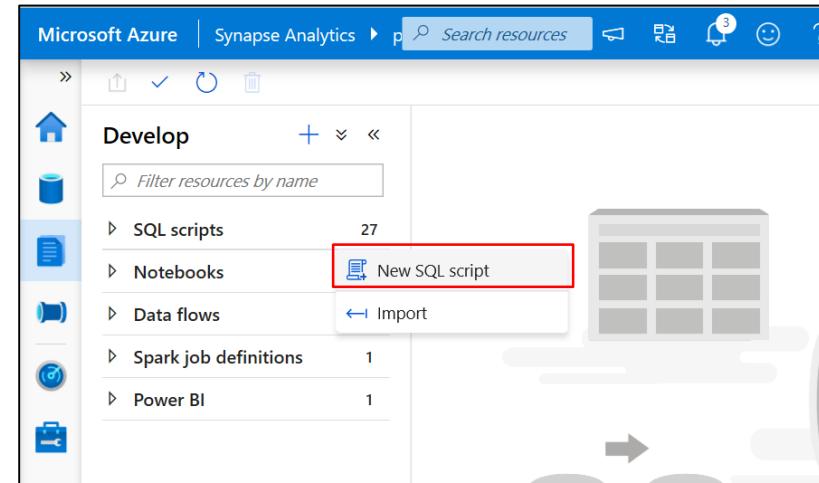


# Develop Hub - SQL scripts

## SQL Script

전용 SQL 풀 서버와 Serverless SQL 풀에서 실행 가능합니다.

단일 SQL 스크립트나 여러 개의 SQL 스크립트를 동시에 사용 가능합니다.



# Develop Hub - SQL scripts

## SQL Script

테이블 또는 차트 형식 안의 결과값을 볼 수 있으며 CSV나 JSON 파일로 추출할 수 있습니다.

The screenshot shows the Azure Synapse Analytics Develop Hub interface. At the top, there is a code editor window titled "SearchLog\_que... X" containing the following T-SQL script:

```

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

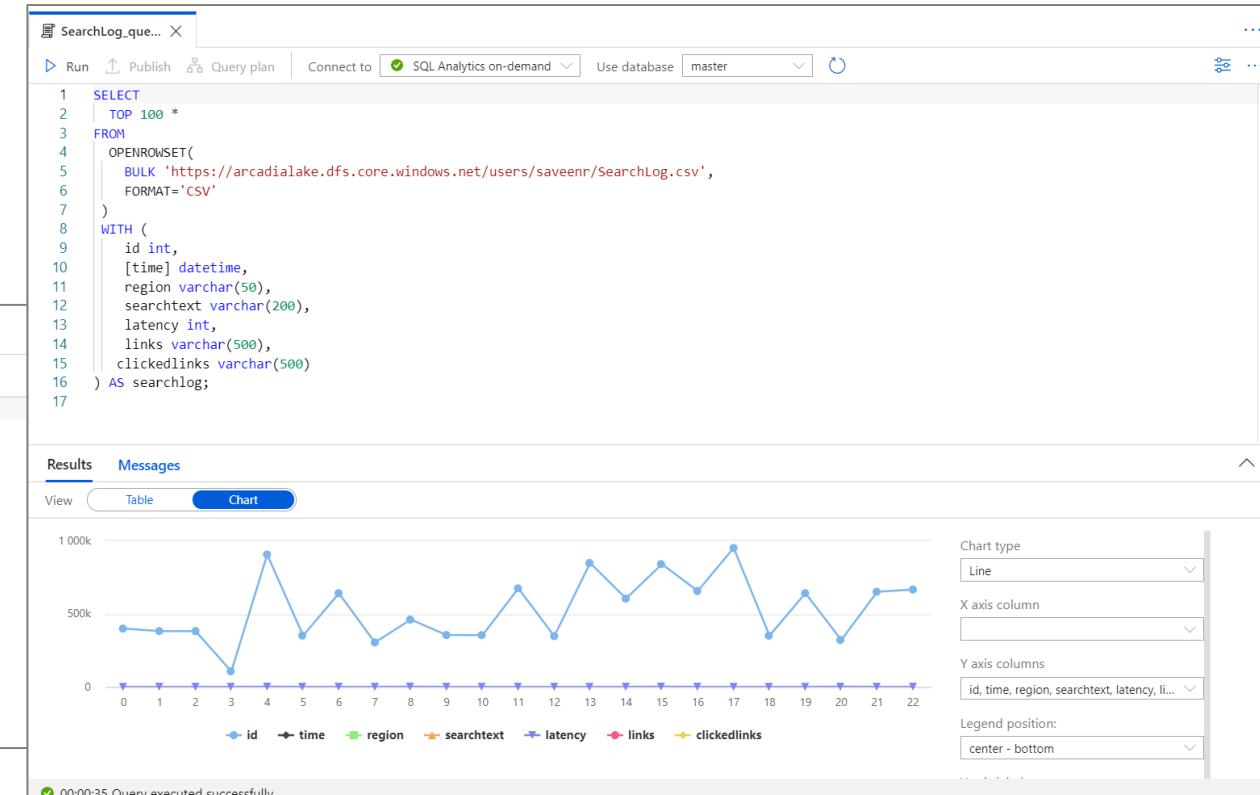
```

Below the code editor is a results table with columns: ID, TIME, and REGION. The table contains the following data:

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

At the bottom of the results table, a message says "00:00:35 Query executed successfully."

To the right of the results table is a "Results" card with tabs for "View", "Table" (which is selected), and "Chart". Below the chart tab is a "Export results" dropdown menu with options: CSV, Excel, JSON, and XML. The "JSON" option is highlighted with a red box.



# Develop Hub - Notebooks

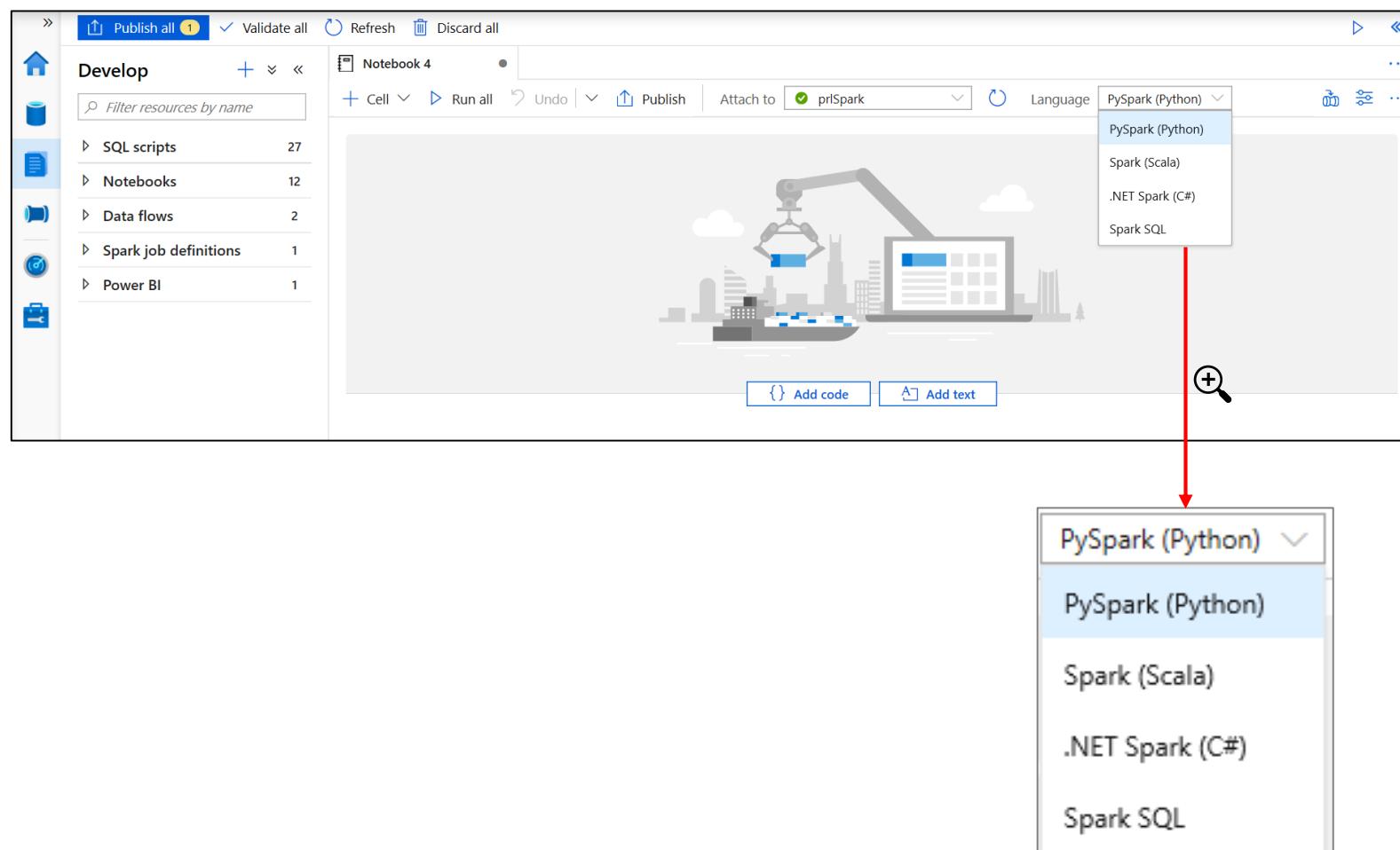
## Notebooks

Notebook Code로 Python, Scala, C#, Spark SQL과 같은 언어 사용이 가능합니다.

해당 언어들을 통해 임시 테이블 생성을 제공합니다.

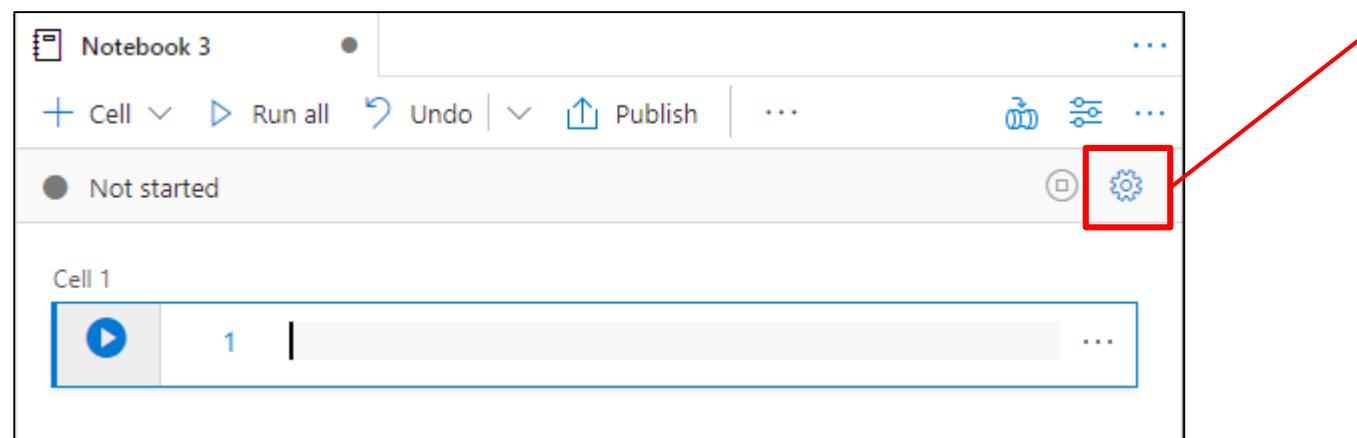
Syntax 문법 에러, 들여쓰기 등의 코드 작성시 편의성을 제공합니다.

결과값을 추출할 수 있습니다.



# Develop Hub - Notebooks

Configure Session은 개발자가 Notebook을 사용하는데 인프라 영역의 리소스 사용량을 늘리거나 줄여 관리하는 기능입니다.



**Configure session**

Livy session ID  
-

Status  
Not started

Attach to \* ⓘ  
analytics1

**analytics1**  
Refresh at 12:04:28 AM

Medium (8 vCores / 56 GB) 3 - 10 nodes 0.00% utilized

Available session sizes ⓘ

Small	19 executors	<a href="#">Use</a>
Medium	9 executors	<a href="#">Use</a>

Executor size \* ⓘ  
Small (4 vCores, 28GB memory)

Executors \* ⓘ  
 2

Driver size \* ⓘ  
Small (4 vCores, 28GB memory)

Session timeout (minutes) \* ⓘ  
30

[Apply](#) [Cancel](#)

# Develop Hub - Notebooks

Notebook에 있는 Cell을 실행하면,  
Spark 애플리케이션 상태가 아래로  
보여지게 됩니다.

애플리케이션 상태를 확인하여  
즉각적인 피드백과 진행 상황을  
확인할 수 있습니다.

The screenshot shows the Microsoft Azure Synapse Analytics Develop Hub - Notebooks interface. At the top, there's a navigation bar with 'Microsoft Azure' and 'Synapse Analytics'. Below it, a search bar says 'Search resources' and a user profile 'prlangad@microsoft.com MICROSOFT'. The main area shows a notebook titled 'opendataset' with a cell labeled 'Cell 1' containing PySpark code to load data from a file path. Below the code, a message indicates it was executed in 2 minutes and 44 seconds by 'prlangad' on March 19, 2020, at 11:31:56.458 -07:00. A section titled 'Job execution Succeeded' shows three tasks: 'Job 0' (load at NativeMethodAccessImpl.java:0), 'Job 1' (showString at NativeMethodAccessImpl.java:0), and 'Job 2' (showString at NativeMethodAccessImpl.java:0), all of which succeeded. At the bottom, a preview of the data shows columns like vendorID, tpepPickupDateTime, tpepDropoffDateTime, passengerCount, tripDistance, puLocationId, doLocationId, startLon, startLat, endLon, endLat, rateCodeId, storeAndFwdFlag, paymentType, fareAmount, extra, mtaTax, improvementSurcharge, tipAmount, tollsAmount, and totalAmount. The data preview includes several rows of taxi trip information.

ID	DESCRIPTION	STATUS	STAGES	TASKS	SUBMISSION TIME	DURATION
▶ Job 0	load at NativeMethodAccessImpl.java:0	<span style="color: green;">✓ Succeeded</span>	1/1	<div style="width: 100%; background-color: #2e8b57;"></div>	3/19/2020, 11:31:35 AM	6s
▶ Job 1	showString at NativeMethodAccessImpl.java:0	<span style="color: green;">✓ Succeeded</span>	1/1	<div style="width: 100%; background-color: #2e8b57;"></div>	3/19/2020, 11:31:43 AM	1s
▶ Job 2	showString at NativeMethodAccessImpl.java:0	<span style="color: green;">✓ Succeeded</span>	1/1	<div style="width: 100%; background-color: #2e8b57;"></div>	3/19/2020, 11:31:45 AM	9s

```

+---+-----+-----+-----+-----+-----+-----+-----+
| vendorID | tpepPickupDateTime | tpepDropoffDateTime | passengerCount | tripDistance | puLocationId | doLocationId | startLon | startLat |
| endLon | endLat | rateCodeId | storeAndFwdFlag | paymentType | fareAmount | extra | mtaTax | improvementSurcharge | tipAmount | tollsAmount |
| totalAmount |
+---+-----+-----+-----+-----+-----+-----+-----+
| CMT | 2009-04-30 23:59:52 | 2009-05-01 00:11:14 | 1 | Credit | 1.9 | 8.5 | 0.0 | null | null | -73.984708 | null | 1.8 |
| 40.760237 | -73.960426 | 40.761527 | null | 0 | 10.3 | null | null | null | null | null | null |
| CMT | 2009-05-07 01:03:26 | 2009-05-07 01:14:11 | 1 | Credit | 3.4 | 9.7 | 0.0 | null | null | -73.956527 | null | 2.55 |
| 40.771307 | -73.941002 | 40.80763 | null | 0.0 | 12.25 | null | null | null | null | null | null |
| CMT | 2009-04-30 23:50:42 | 2009-05-01 00:06:43 | 1 | 2.2 | null | null | null | null | -74.009102 | null |
+---+

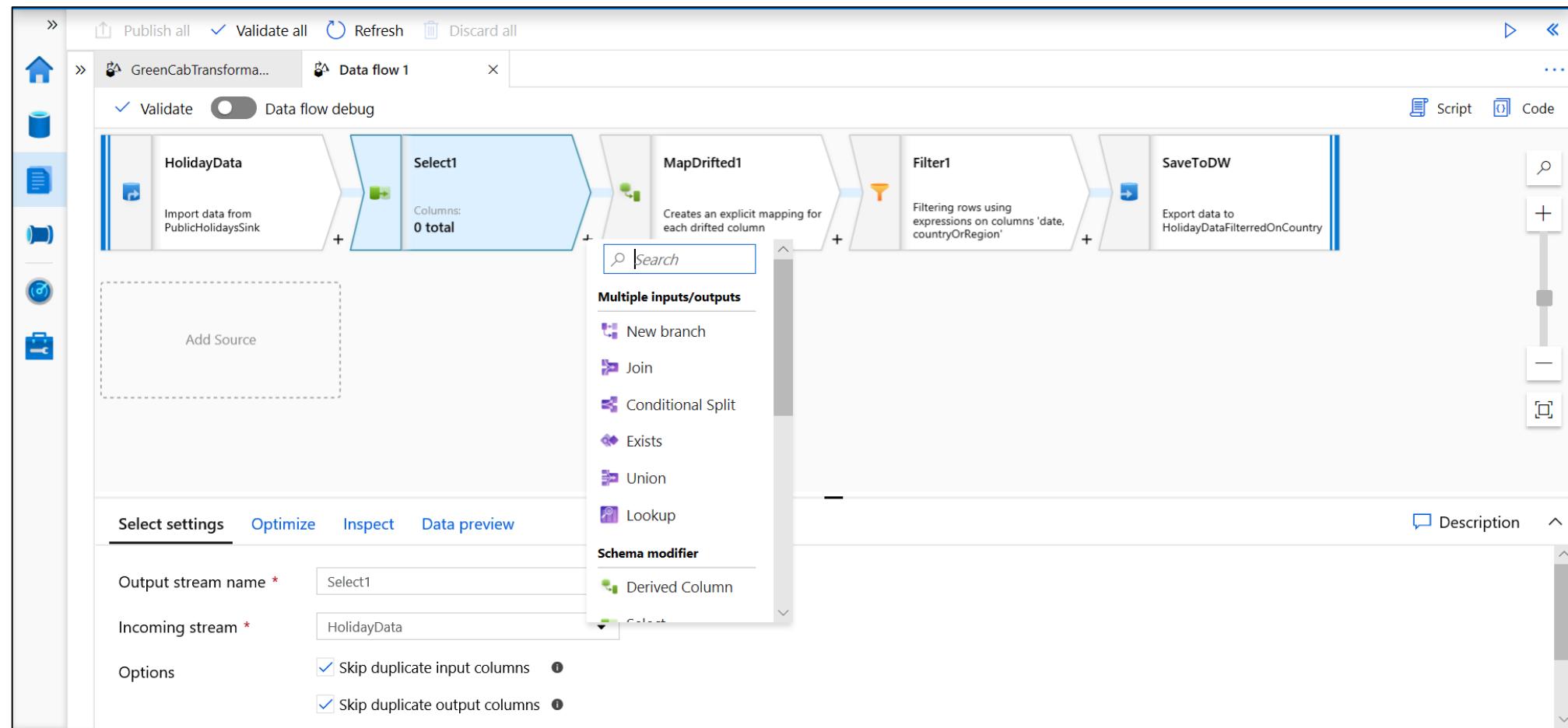
```

Ready (Stop session) | Configure session

# Develop Hub - Data Flows

데이터 흐름 매핑은 데이터 변환을 시각적으로 지정하는 방법을 제공하는 파이프라인 활동입니다.

코드의 작성 없이 데이터 흐름을 정의할 수 있습니다.



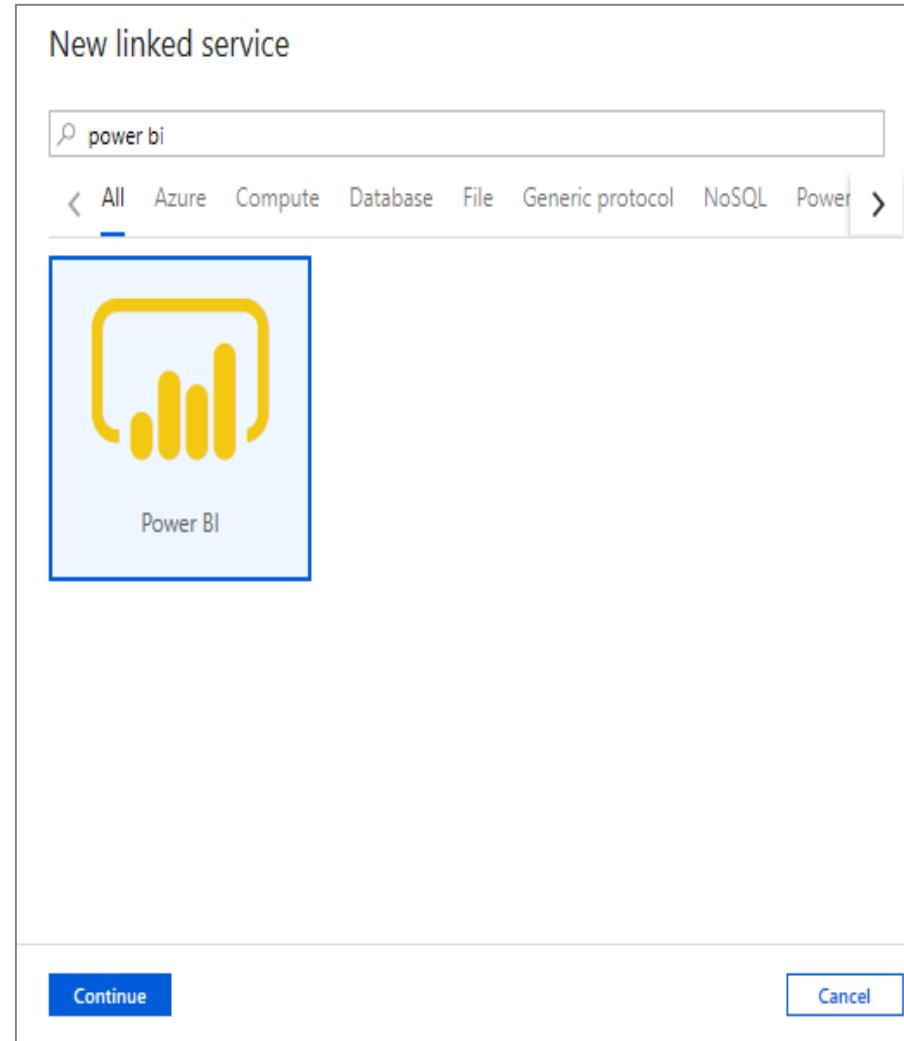
# Develop Hub – Power BI

## Overview

Power BI 리포트는 워크스페이스 안에서 생성할 수 있습니다.

워크스페이스 안에서 완성된 리포트에 접근이 가능하고 실시간으로 Synapse 워크스페이스로부터 업데이트가 가능합니다.

분석된 데이터를 시각화하여 볼 수 있습니다.



- SQL scripts 27
- Notebooks 11
- Data flows 2
- Spark job definitions 1
- Power BI 1
  - Demo Synapse
    - Power BI datasets
    - Power BI reports

# Develop – CI/CD

Synapse에 Repository를 생성하여 Artifacts들의 CI/CD 구현이 가능합니다.

The screenshot displays two side-by-side interfaces. On the left is a GitHub repository page for 'SynapseTestDemo/synapsetestdemo-ws-01'. The 'dev' branch is selected, showing a list of commits. The most recent commit is 'Initial commit' by 'Priyanka Langade' on 11/16/2020 at 11:02 PM. On the right is an Azure Synapse deployment pipeline interface titled 'Synapse deployment v2 > Release-13'. The pipeline has two stages: 'Release' and 'Stages'. The 'Release' stage shows a 'Manually triggered' run by Priyanka Langade on 11/16/2020 at 11:02 PM. The 'Stages' stage shows a 'Load to Prod' step that succeeded on 11/17/2020 at 4:54 PM. Below the stages, under 'Artifacts', there is a reference to 'azuresynapseprod' with ID '0c8b1a872' and branch 'retail-12'.

credential Adding linkedService: synapsedemosws-WorkspaceDefaultStorage 61 min ago

dataflow Updating integrationRuntime: AutoResolveIntegrationRuntime 39 min ago

dataset Updating integrationRuntime: AutoResolveIntegrationRuntime 39 min ago

integrationRuntime Adding linkedService: synapsedemosws-WorkspaceDefaultStorage 61 min ago

linkedService Updating integrationRuntime: AutoResolveIntegrationRuntime 39 min ago

notebook Updating integrationRuntime: AutoResolveIntegrationRuntime 39 min ago

pipeline Updating integrationRuntime: AutoResolveIntegrationRuntime 39 min ago

sparkJobDefinition Adding linkedService: synapsedemosws-WorkspaceDefaultStorage 61 min ago

sqlscript Updating integrationRuntime: AutoResolveIntegrationRuntime 39 min ago

README.md Initial commit 14 days ago

README.md

**synapsetestdemo-ws-01**

↑ Synapse deployment v2 > Release-13

Pipeline Variables History + Deploy Cancel Refresh Edit ...

Release

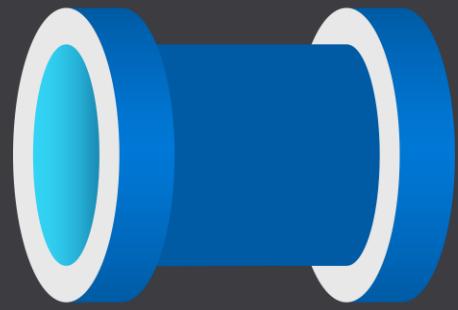
Manually triggered by Priyanka Langade 11/16/2020, 11:02 PM

Artifacts

azuresynapseprod  
0c8b1a872  
branch-retail-12

Stages

Load to Prod Succeeded on 11/17/2020, 4:54 PM



# Synapse Studio

## Integrate hub

# Integrate Hub

Integrate Hub에서는 시냅스 내부/외부로 연결할 수 있는 90개 이상의 Connector들을 이용하여 데이터의 수집, 변형, 적재를 위한 파이프라인을 생성합니다. 파이프라인에 의해 수행되는 Activity는 지원 범위가 넓습니다.

The screenshot shows the Azure Synapse Analytics Integrate Hub interface. On the left, there are three collapsed sections: 'Synapse' (Notebook, Spark job definition, Stored procedure), 'Move & transform' (Copy data, Data flow), and 'Machine Learning' (ML Batch Execution, ML Update Resource, ML Execute Pipeline). Red arrows point from each of these sections to the corresponding connector categories in the central Activities catalog. The Activities catalog is a tree view with a red border, listing connectors under various categories: Synapse, Move & transform, Azure Data Explorer, Azure Function, Batch Service, Databricks, Data Lake Analytics, General, HDInsight, Iteration & conditionals, and Machine Learning. To the right of the catalog, a pipeline editor window is open, showing two 'Copy data' activities: one for 'GreenTaxi' and one for 'YellowTaxi'. The pipeline editor has tabs for General, Parameters, Variables, and Output, with the General tab selected. The General tab contains fields for Name (Copy Open Dataset), Description, Concurrency, and Annotations.



# Synapse Studio Monitor hub

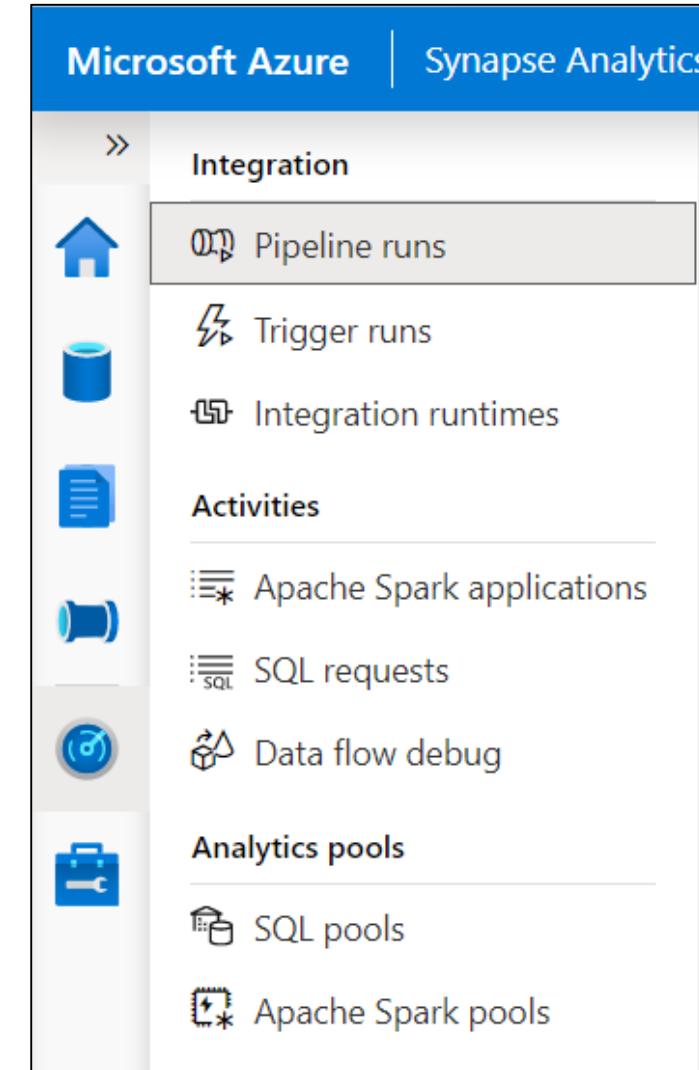
# Monitor Hub

## Overview

모니터 허브에서는 Apache Spark Application과 SQL Activity, 오케스트레이션의 각각에 대한 상황을 제공합니다.

## Benefits

오케스트레이션이나 구체적인 Activity의 상황을 필터링을 통해 모니터링을 할 수 있습니다.



# Monitor Hub - Integration

## Overview

Pipeline의 진행 상황을 위한 Synapse 워크스페이스 안에서 모니터 오케스트레이션을 합니다.

## Benefits

모든 Pipeline 또는 특정 Pipeline을 추적합니다.

Activity의 구체적인 실행과 Pipeline을 모니터링합니다.

Activity의 실패 또는 Pipeline의 실패 원인을 찾아냅니다.

Pipeline runs				
All status		Rerun	Cancel	Refresh
Pipeline Name	Run Start	DURATION	Triggered By	Status
Load Data to SQLDW	10/25/2019, 3:49:42 PM	00:10:55	Manual trigger	<span style="color: green;">✓ Succeeded</span>
Copy Open Dataset	10/25/2019, 2:17:54 PM	00:14:12	Manual trigger	<span style="color: green;">✓ Succeeded</span>
Pipeline 1	10/24/2019, 1:23:43 PM	00:00:08	Manual trigger	<span style="color: green;">✓ Succeeded</span>

Trigger runs			
All status		Refresh	Edit columns
Showing 1 - 1 items			
Trigger Name	Trigger Type	Trigger Time	Status
TriggerCopy_csvdata100	ScheduleTrigger	4/10/20, 12:14:00 AM	<span style="color: green;">✓ Succeeded</span>

# Monitor Hub - Spark applications

## Overview

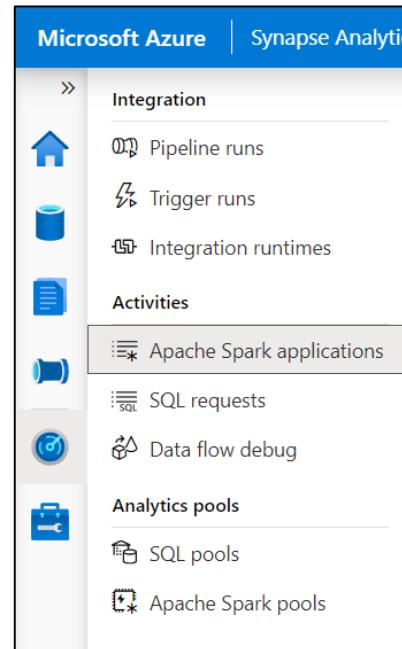
Activity의 상태를 위한 Spark Pool, Spark Application 상태를 모니터링합니다.

## Benefits

Apache Spark Application의 Pool에 대한 필터링 적용

이용 가능한 필터링 제공

1. 애플리케이션 이름
2. Live ID
3. 상태
4. 완성 시간



Apache Spark applications				
Application name	Submitter	Submit time	Status	Pool
Synapse_automlpool_1...	negust@microsoft.com	11/30/20, 7:23:26 PM	Stopped	automlpool
Synapse_automlpool_1...	negust@microsoft.com	11/30/20, 7:14:00 PM	Stopped	automlpool
Synapse_automlpool_1...	negust@microsoft.com	11/30/20, 5:09:00 PM	Stopped (session timed ou...	automlpool
Notebook 3_analyticsp...	negust@microsoft.com	11/30/20, 12:52:51 PM	Stopped	analyticspool
Notebook 4_analytics1...	prlangad@microsoft.com	11/24/20, 6:29:41 PM	Stopped	analytics1
Synapse_hbpool_16059...	charlesf@microsoft.com	11/20/20, 2:30:44 PM	Stopped	hbpool
Synapse_analyticspool...	negust@microsoft.com	11/20/20, 11:20:30 AM	Stopped	analyticspool

# Monitor Hub – SQL requests

## Overview

Activity의 진행 상태를 위한 SQL 요청을 모니터링합니다.

## Benefits

Pool이 SQL 요청을 얻기 위해 필터링 적용합니다.

쿼리 구문을 허용합니다.  
이용 가능한 추가 필터링 제공

1. 시작시간
2. 종료시간
3. 요청 ID
4. 세션 ID
5. 제공자
6. 워크로드 그룹

SQL requests							
Request ID ↑↓		Request content ↑↓		Submit time ↑↓		Duration	
Pacific Time (US & C... : Last 30 days	Status : All	Pool : Predict_Pool	Add filter				
Showing 1 ~ 100 of 248 items							
QID125878	USE [DWShellDb]	12/1/20, 12:27:53 AM	0s	System	Completed	0s	
QID125879	--Backing up Logical Azure Data	12/1/20, 12:27:53 AM	20s	System	Completed	0s	
QID125637	USE [DWShellDb]	11/30/20, 8:27:53 PM	0s	System	Completed	0s	
QID125638	--Backing up Logical Azure Data	11/30/20, 8:27:53 PM	15s	System	Completed	0s	
QID125529	USE [Predict_Pool]	11/30/20, 6:41:15 PM	0s	anrampal@microsoft.com	Completed	0s	
QID125530	SELECT s.NAME AS SchemaNam	11/30/20, 6:41:15 PM	0s	anrampal@microsoft.com	Completed	0s	
QID125421	USE [Predict_Pool]	11/30/20, 4:54:56 PM	0s	negust@microsoft.com	Completed	0s	

Microsoft Azure   Synapse Analytics > wsazuresynapseanalytics				
Integration		SQL requests		
Activities		Request ID ↑↓ Request content ↑↓ Submit time ↑↓ Duration Data processed		
Pacific Time (US & C... : Last 30 days	Status : All	Pool : Built-in	Add filter	
Showing 1 ~ 100 of 279 items				
12162198	SELECT TOP 100* FROM OPEI	11/30/20, 8:57:12 PM	1s	1 MiB
11573006	SELECT TOP 100* FROM OPEI	11/30/20, 6:23:32 PM	6s	1 MiB
9079654	SELECT product = ISNULL(p.pro	11/30/20, 7:28:38 AM	6s	1 MiB
9066730	SELECT product = ISNULL(p.pro	11/30/20, 7:26:17 AM	5s	1 MiB
9065769	SELECT * FROM OPENROWSET(	11/30/20, 7:25:47 AM	2s	1 MiB
9062482	SELECT * FROM OPENROWSET(	11/30/20, 7:25:17 AM	18s	18 MiB

# Monitor Hub – Analytics pools

**SQL pools**

Refresh Edit columns

Pool : All

Showing 1 - 6 of 6 items

Pool name ↑↓	Type ↑↓	Status ↑↓	Size ↑↓	CPU utilizati... ⓘ ↑↓	Memory utili... ⓘ ↑↓	Created on ↑↓
Built-in	Serverless	✓ Online	Auto	N/A	N/A	N/A
newpoll	Dedicated	● Paused	DW200c	-	-	11/5/2020 5:42:52 AM
NYCTaxi_Pool	Dedicated	✓ Online	DW100c	1.1	23	8/27/2020 6:03:32 AM
Predict_Pool	Dedicated	✓ Online	DW1000c	-	5.33	9/9/2020 1:13:26 AM
Streaming_Pool	Dedicated	● Paused	DW2000c	-	-	8/27/2020 4:04:18 AM
WWI_Pool	Dedicated	✓ Online	DW100c	0.12	20.22	8/26/2020 7:18:23 PM

**Apache Spark pools**

Refresh Edit columns

Pool : analytics

Showing 1 - 4 of 4 items

Pool name ↑↓	Size	Active users	Allocated vCores	Allocated memory (GB)	Created on
analytics2	Medium (8 vCores / 64 GB) - 10 nodes	0	0	0	10/30/20, 2:47:16 PM
analytics1	Medium (8 vCores / 64 GB) - 3 to 10 nodes	0	0	0	10/28/20, 3:36:18 AM
AnalyticsPool99	Medium (8 vCores / 64 GB) - 3 to 10 nodes	0	0	0	9/10/20, 7:17:53 AM
analyticspool	Medium (8 vCores / 64 GB) - 3 to 15 nodes	0	0	0	8/26/20, 7:13:44 PM



# Synapse Studio Manage hub

# Manage Hub

## Overview

Manage Hub은 분석 풀, 연결 서비스, 통합, 보안 그리고 Source Control을 제공합니다.

The screenshot shows the Microsoft Azure Synapse Analytics Manage Hub interface. The left sidebar has a 'Manage' section selected. The main area displays 'SQL pools' with a table listing six items. The table columns are Name, Type, Status, and Size. The items are:

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

At the top, there are buttons for 'Synapse live', 'Validate all', 'Publish all' (with a count of 1), and a 'System assigned managed identity' toggle. The left sidebar also lists Home, Data, Develop, Integrate, Monitor, and Manage sections.

# Manage – dedicated SQL pools

## Overview

전용 SQL 풀은 정지, 재시작, 스케일 변경이 가능하며, 스튜디오에서 Tag를 지정할 수 있습니다.

Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics

Synapse live ▾ Validate all Publish all 1

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

Analytics pools
 

- SQL pools**
- Apache Spark pools
- External connections
- Linked services

Integration
 

- Triggers
- Integration runtimes

Security
 

- Access control
- Credentials
- Managed private endpoints

Source control
 

- Git configuration

**SQL pools**

Serverless SQL pool is immediately available for your workspace. Dedicated SQL pools can be scaled up or down based on your needs.

+ New Refresh System assigned managed identity

Showing 1-6 of 6 items (1 Serverless, 5 Dedicated)

Name	Type
Built-in	Serverless
newpool	Dedicated
NYCTaxi_Pool	Dedicated
Predict_Pool	Pause
Streaming_Pool	Scale
WWI_Pool	Assign tags

**Scale**

NYCTaxi\_Pool

Scaling can impact workload management settings. Consider using the [workload management scale experience](#) in the Azure portal to configure the settings that best align to your workload needs. [Learn more about performance levels](#)

Performance level DW500c

Estimated price ⓘ  
Est. cost per hour 6.00 USD

Apply Cancel

# Manage – serverless SQL pools

## Overview

Serverless SQL 풀은 T-SQL 또는 시냅스 스튜디오 안에서 진행되고 있는 데이터의 양에 대한 리소스를 측정하고 관리합니다.

The screenshot shows the Azure Synapse Analytics portal interface. On the left, there's a navigation sidebar with Home, Data, Develop, Integrate, Monitor, and Manage sections. Under the Manage section, 'Analytics pools' is selected, which further branches into 'SQL pools' and 'Apache Spark pools'. The main content area is titled 'SQL pools' and contains a message about serverless pools being immediately available. It shows 1-6 of 6 items (1 Serverless, 5 Dedicated). A table lists the pool types: 'Built-in' (Serverless) and '... More' (Dedicated). A red box highlights the '... More' button. Below the table, a code editor window shows the following T-SQL script:

```

1 sp_set_data_processed_limit
2     @type = N'daily',
3     @limit_tb = 10
4
5 sp_set_data_processed_limit
6     @type= N'weekly',
7     @limit_tb = 100
8
9 sp_set_data_processed_limit
10    @type= N'monthly',
11    @limit_tb = 1000

```

### Cost Control

Workspace Budget limit for a period. [Learn more](#)

**Daily limit** ⓘ

Enable  Disable

**Data used today**  
2 MB

10 TB

**Weekly limit** ⓘ

Enable  Disable

**Data used this week**  
23 MB

100 TB

**Monthly limit** ⓘ

Enable  Disable

**Data used this month**  
2 MB

0 TB

**Apply** **Cancel**

# Manage – Apache Spark pools

## Overview

Apache Spark 풀은 스튜디오에서 패키지, 구성 관리를 업로드하고, 태그를 지정해 주며 스케일 관리를 합니다. 또한 스파크 풀의 역할 지정을 허용합니다.

The screenshot shows the Microsoft Azure Synapse Analytics portal. On the left, the navigation menu includes Home, Data, Develop, Integrate, Monitor, and Manage. Under the Manage section, Analytics pools is selected. The main content area is titled "Apache Spark pool" and displays a list of existing pools: analyticspool, AnalyticsPool99, analytics1, and analytics2. A context menu is open over the first pool, listing options: Auto-pause settings, Autoscale settings, Packages, Spark configuration, Assign tags, View role assignments, and Delete. The "View role assignments" option is highlighted with a red box.

The screenshot shows the "View role assignment" dialog box. It displays four role assignments for the pool "analyticspool":

Name	Type	Scope
Synapse Administrator	Individual	Workspace (Inherited)
Priyanka Langade	Individual	Workspace (Inherited)
SynapseWsAdmin	Group	Workspace (Inherited)
SynapseWsAdmin@serv	Group	Workspace (Inherited)
Charles Feddersen	Individual	Workspace (Inherited)
wsazuresynapseanalytics	Service Principal	Workspace (Inherited)

Below this, there is a "Manage packages" section where a file named "requirements.txt" is listed. The "requirements.txt" file is highlighted with a red box.

# Manage – Linked services

## Overview

Linked Service는 외부 리소스에 연결하려는 연결 정보를 정의합니다.

## Benefits

90개 이상의 Connector를 제공합니다.  
컴퓨팅 리소스 또는 데이터 스토어를 나타냅니다.

Microsoft Azure | Synapse Analytics > wsazuresynapseanalytics

Synapse live | Validate all | Publish all

**Linked services**

Linked services are much like connection strings, which define the connection to external resources. [Learn more](#)

**New**

**New linked service**

PayPal (Preview)	Phoenix	PostgreSQL
Power BI	Presto (Preview)	QuickBooks (Preview)
AzureDataExplorer	AzureMLService1	AzureMLServiceN
bing-covid-19-d	SAP BW Open Hub	SAP BW via MDX
Nellies_Keyvault	SAP Cloud For Customer	SAP ECC
C4C	SAP HANA	SAP HANA

Show 1 - 15 of 15

Name ↑

Continue Cancel

# Manage – Triggers

## Overview

Trigger는 Pipeline을 수행하기 위해 Schedule 설정을 통하여 자동 실행 or 사용자에 의한 수동 실행이 가능한 기능입니다.

## Benefits

Trigger를 생성하고 관리할 수 있습니다.

- 트리거의 3가지 종류

Schedule 트리거

Dumbling Window 트리거

Event 트리거

**New trigger**

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

Name **\***  
Trigger 2

Description

Type **\***  
 Schedule    Tumbling window    Event

Start Date (UTC) **\***  
10/29/2019 9:46 PM

Recurrence **\***  
Every 1 Minute(s)

End **\***  
 No End    On Date

Annotations  
+ New

Activated **\***  
 Yes    No

**OK**   **Cancel**

# Manage – Integration runtimes

## Overview

## Benefits

Integration runtime은 서로 다른 네트워크 환경에서 통합 환경을 제공하기 위한 Pipeline에 의해 사용된 컴퓨팅 Infra Structure 입니다.

Integration runtime은 Activity와 Linked Service 사이의 다리 역할을 제공합니다.

Integration runtime은 Azure Integration runtime 또는 Self-Hosted된 Integration runtime을 제공합니다.

Azure Integration runtime은 완전히 관리되는 Azure 내의 서버리스 컴퓨팅을 제공합니다.

Self-Hosted 통합 런타임은 오프레미스 서버 또는 사설 네트워크 안의 가상 머신 안의 컴퓨팅 리소스를 사용합니다.

The integration runtime (IR) is the compute infrastructure to provide the following network environment. [Learn more](#)

**+ New** **Refresh**

**Filter by keyword**

Showing 1 - 1 of 1 items

Name	Type	Sub-type	Status
AutoResolv...	Azure	Self-Hosted	Normal

**Integration runtime setup**

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:

- Azure
- Self-Hosted

[Continue](#) [Back](#) [Cancel](#)

# Manage – Access Control

## Overview

Access Control은 워크스페이스 리소스부터 관리자의 Artifact까지 접근 제어 관리를 제공합니다.

## Benefits

- 팀 내에서 워크스페이스를 공유
- 생산성의 증가
- 단계적인 허가 가능 수준 지정
- 스파크 풀, 통합 런타임, 연결 서비스, 비밀번호의 권한 관리

The screenshot shows the Microsoft Azure Access Control interface for a workspace named 'internalsandbox'. The main pane displays a table of access roles:

NAME	TYPE	ROLE
soft.com	Individual	Workspace admin
core	Group	Workspace admin

A red box highlights the '+ Add' button in the top right, and a red arrow points from it to the 'Add role assignment' dialog box below. Another red arrow points from the 'Add' button to the 'Add role assignment' dialog box on the right.

**Add role assignment**

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

**Scope \***  Workspace  Workspace item

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

**Add role assignment**

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

**Scope \***  Workspace  Workspace item

**Item type \***  Credentials  
 WorkspaceSystemIdentity

**Item \***  WorkspaceSystemIdentity

**Role \***  Synapse Administrator  
 Select user \*

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

# Manage – Source Control

## Overview

소스 제어는 Git configuration에서 Azure devops hub, Github을 연동할 수 있습니다.

**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**

**Configure a repository**

main branch

Filter...

dev branch

main branch

workspace\_publish branch

Create pull request [Alt+P]

New branch [Alt+N]

Switch to live mode

**Repository type** GitHub

**GitHub account** SynapseTestDemo

**Git repository name** synapsetestdemo-ws-01

**Collaboration branch** dev

**Publish branch** main

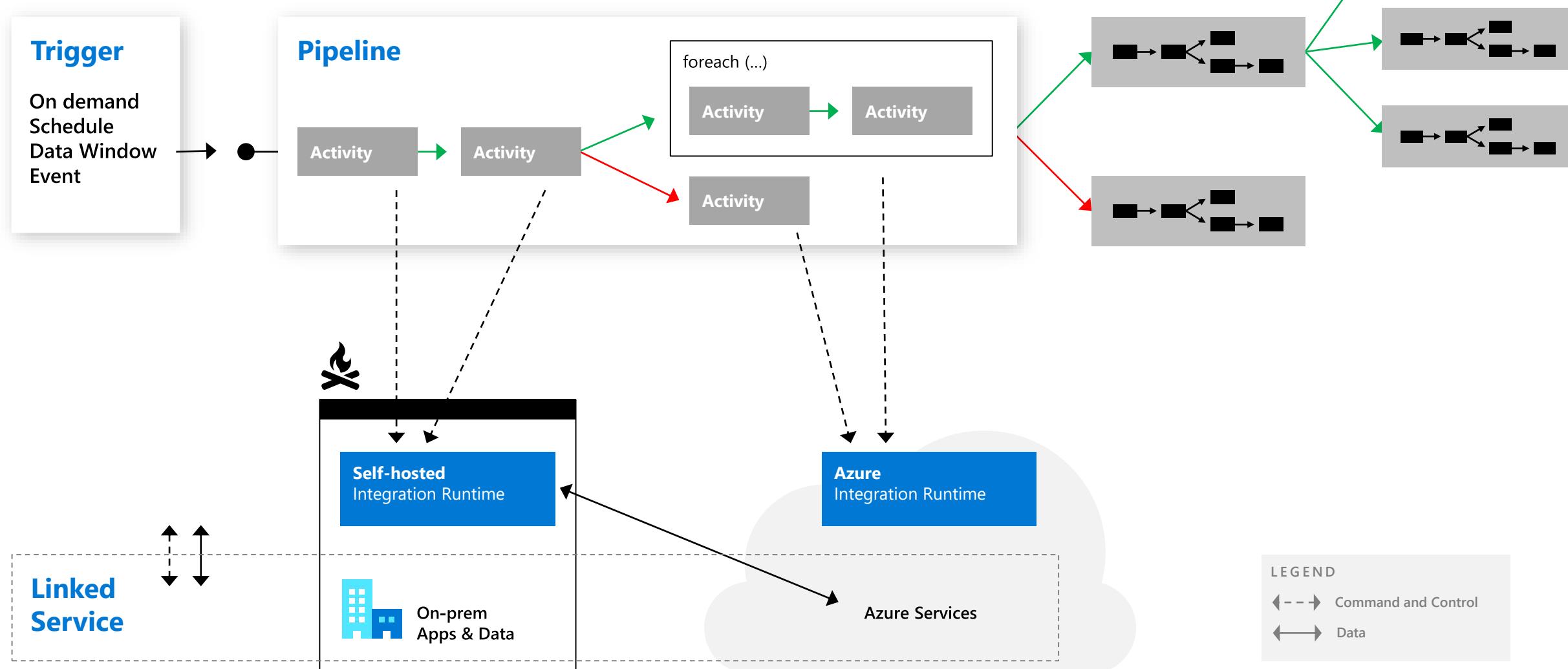
**Root folder** /



# Azure Synapse Analytics

## Integrate

# Orchestration @ Scale



# 90+ Connectors out of the box

Azure (15)	Database & DW (26)		File Storage (6)	File Formats(6)	NoSQL (3)	Services and App (28)		Generic (4)
Blob storage	Amazon Redshift	Oracle	Amazon S3	AVRO	Cassandra	Amazon MWS	Oracle Service Cloud	Generic HTTP
Cosmos DB - SQL API	DB2	Phoenix	File system	Binary	Couchbase	CDS for Apps	PayPal	Generic OData
Cosmos DB - MongoDB API	Drill	PostgreSQL	FTP	Delimited Text	MongoDB	Concur	QuickBooks	Generic ODBC
Data Explorer	Google BigQuery	Presto	Google Cloud Storage	JSON		Dynamics 365	Salesforce	Generic REST
Data Lake Storage Gen1	Greenplum	SAP BW Open Hub	HDFS	ORC		Dynamics AX	SF Service Cloud	
Data Lake Storage Gen2	HBase	SAP BW via MDX	SFTP	Parquet		Dynamics CRM	SF Marketing Cloud	
Database for MariaDB	Hive	SAP HANA				Google AdWords	SAP C4C	
Database for MySQL	Apache Impala	SAP table				HubSpot	SAP ECC	
Database for PostgreSQL	Informix	Spark				Jira	ServiceNow	
File Storage	MariaDB	SQL Server				Magento	Shopify	
SQL Database	Microsoft Access	Sybase				Marketo	Square	
SQL Database MI	MySQL	Teradata				Office 365	Web table	
SQL Data Warehouse	Netezza	Vertica				Oracle Eloqua	Xero	
Search index						Oracle Responsys	Zoho	
Table storage								

## Data Movement - Scalable

Per job elasticity - Up to 4 GB/s

# Pipelines

## Overview

Pipeline은 Storage Account 와 Linked Service로 부터 데이터를 적재할 수 있는 기능을 지원합니다. Pipeline은 자동 혹은 수동으로 수행하여 Data를 적재 할 수 있습니다.

## Benefits

Supports common loading patterns

Fully parallel loading into data lake or SQL tables

Graphical development experience

The screenshot displays two views of the Azure Synapse Analytics Pipelines interface:

- Top View:** Shows the 'Orchestrate' blade with the 'Pipelines' section selected. A 'New pipeline' button is highlighted. The URL is 'Microsoft Azure | Synapse Analytics > internalsandboxwe5'.
- Bottom View:** Shows the detailed configuration of a 'CopyPipeline\_0313' activity. The 'Activities' list includes 'Synapse', 'Move & transform', 'Azure Data Explorer', 'Azure Function', 'Batch Service', 'Databricks', 'Data Lake Analytics', 'General', 'HDInsight', 'Iteration & conditionals', and 'Machine Learning'. The 'Source' tab is selected, showing 'Source dataset' as 'SourceDataset\_mrk'. Other settings include 'Recursively' checked, 'Wildcard folder path', 'Wildcard file name', 'Enable partition discovery', and date/time filters for 'Modified datetime start (UTC)' and 'Modified datetime end (UTC)'. On the right, a 'New dataset' pane lists various data stores: Amazon Marketplace Web Service, Amazon Redshift, Amazon S3, Apache Impala, Azure Blob Storage, Azure Cosmos DB (MongoDB API), Azure Cosmos DB (SQL API), Azure Data Explorer (Kusto), Azure Data Lake Storage Gen1, Azure Data Lake Storage, and Azure Synapse Analytics.

# Prep & Transform Data

## Overview

데이터 정제(cleaning), 변형, 집계, 전환 등의 기능을 지원합니다.

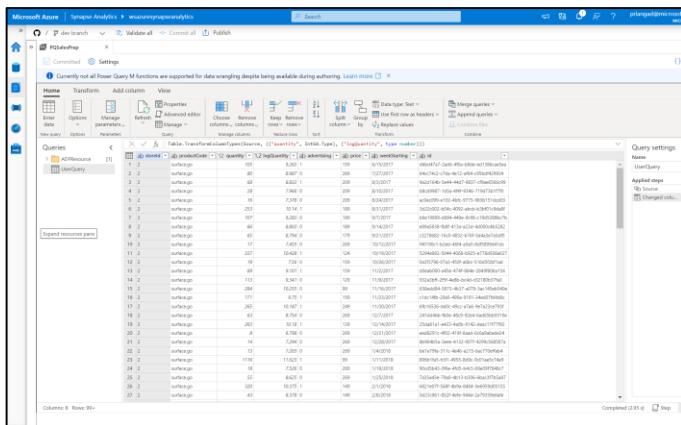
## Benefits

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



```

# MoveRecommendationE2EDemo.txt
1
2 HDFS Cluster Details:
3   Adfhdi.azurehdinsight.net
4 Admin
5 Adf@123456
6
7 Storage:
8 /anywp6G1j7tE1l0Mm1So/Vg2)gT4d+51Ar+vSn7bJg95476gjCHloksZI9Uxsc40xZob1KwdWkQ==
9
10 Cluster Remote Login Details:
11 Adf
12 Indv@1234
13
14 HiveQuery:
15 DROP TABLE IF EXISTS MovieRatings;
16 CREATE EXTERNAL TABLE MovieRatings
17 (
18   UserID int,
19   MovieID int,
20   Rating int,
21  TimeStamp string
22 ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE LOCATION '${hiveconf:MovieRatings}';
23
24
25 DROP TABLE IF EXISTS MovieTitles;
26 CREATE EXTERNAL TABLE MovieTitles
27 (
28   MovieID int,
29   MovieName string
30 ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE LOCATION '${hiveconf:MovieTitles}';

```

...  
not

# Triggers

## Overview

트리거는 파이프라인을 실행하는 기능입니다.

3가지 종류의 트리거를 제공합니다.

1. Schedule – 시작일, 반복, 종료일의 예약에 따라 수행됩니다.
2. Event – 지정한 특정 이벤트가 발생했을 때 수행됩니다.
3. Tumbling Window – 지정된 시작일로부터 일정한 시간 간격으로 파이프라인을 수행합니다.

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

New trigger

Name \* Trigger 1

Description

Type \*  Schedule  Tumbling window  Event

Start Date (UTC) \* 10/30/2019 11:20 PM

Recurrence \* Every 1 Minute(s)

End \*  No End  On Date

Annotations + New

Activated \*  Yes  No

OK

Microsoft Azure | Synapse Analytics > prlangadws2

Analytics pools  
SQL pools  
Apache Spark pools  
External connections  
Linked services  
Orchestration  
Triggers  
Integration runtimes  
Security  
Access control  
Managed Virtual Networks

Triggers

To execute a pipeline set the trigger. Triggers represent a unit of processing that determines when a pipeline execution needs to be kicked off.

+ New

Showing 1 - 1 of 1 items

NAME ↑↓	TYPE ↑↓	STATUS ↑↓	NUMBER OF PIPELINES ↑↓	ANNOTATIONS ↑↓
HolidayUpdateTrigger	Schedule	Started	1	



# Azure Synapse Analytics

## Synapse SQL

# Key features

## Rich Surface Area

- T-SQL을 사용하여 데이터 분석
- 엔터프라이즈급 보안

## Dedicated SQL Pool

- 엔터프라이즈급 DW
- 인덱싱 & 캐싱 지원
- 외부 데이터 조회 및 LOAD
- 워크로드 관리

## Serverless SQL Pool

- 외부 데이터 조회
- 원시 데이터를 테이블 및 뷰로 모델화
- 쉬운 데이터 변환

# Synapse SQL - clients and tools

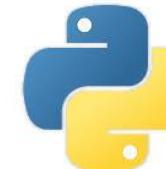
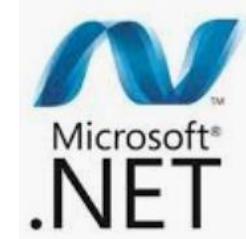
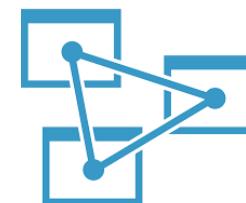
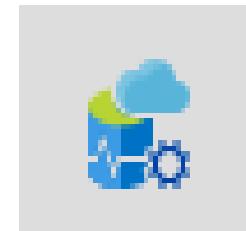
## Tools

Web IDE - Synapse Studio

Client tools - Azure Data Studio, SSMS,

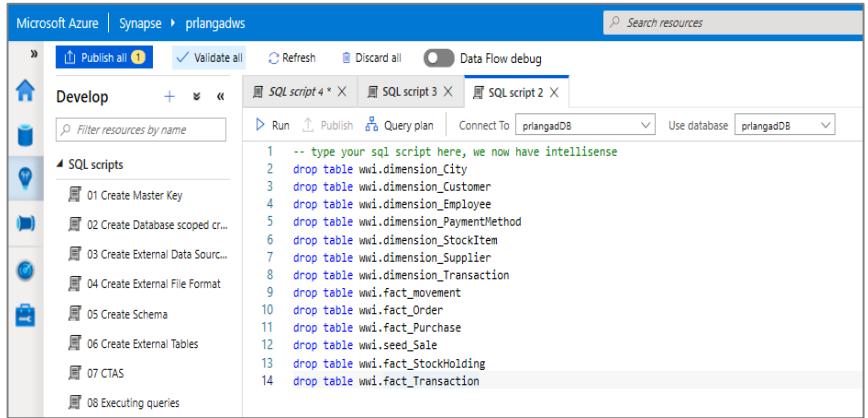
Any tool/library that uses standard SQL can access serverless SQL pool

- PowerBI
- Azure Analytic Services
- Client languages and drivers that works with Azure SQL can be used to access serverless SQL pool

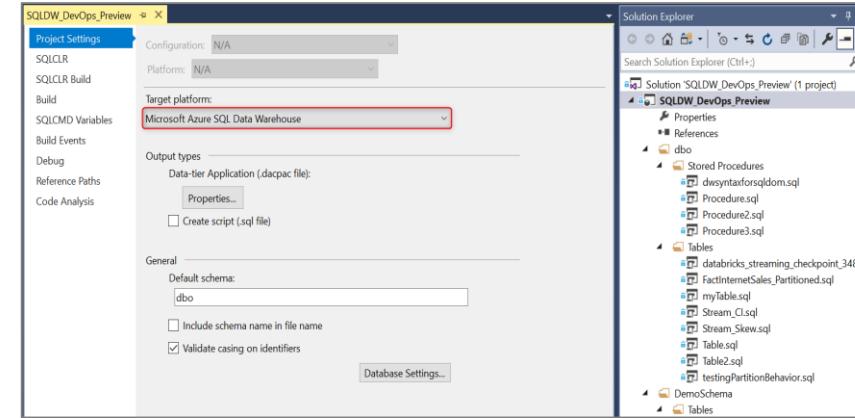


# Developer Tools

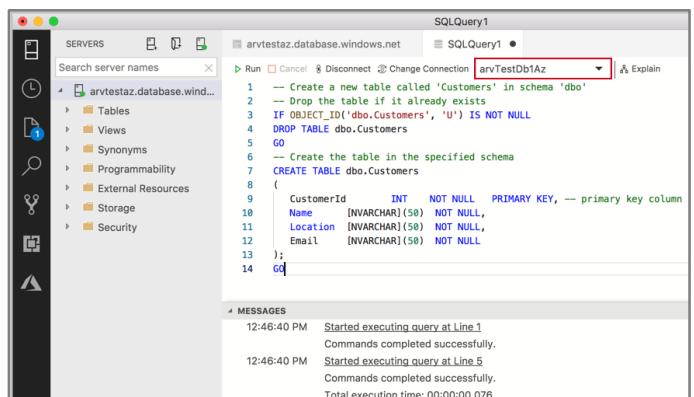
## Azure Synapse Analytics



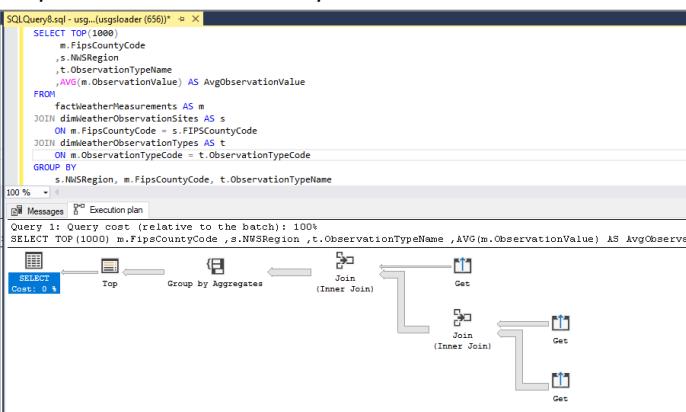
## Visual Studio - SSDT database projects



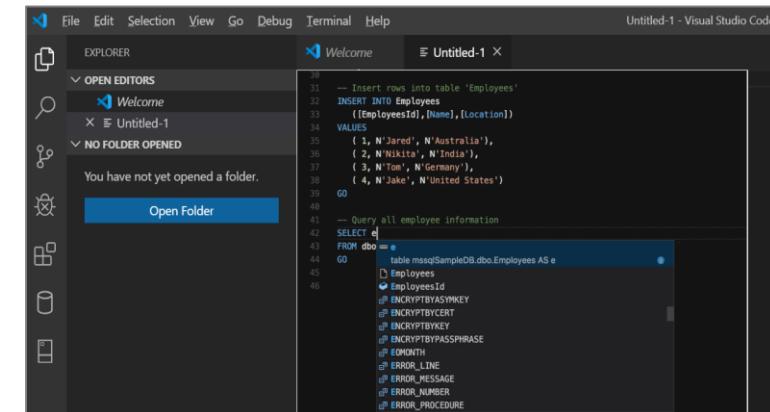
## Azure Data Studio (queries, extensions etc.)



## SQL Server Management Studio (queries, execution plans etc.)



## Visual Studio Code

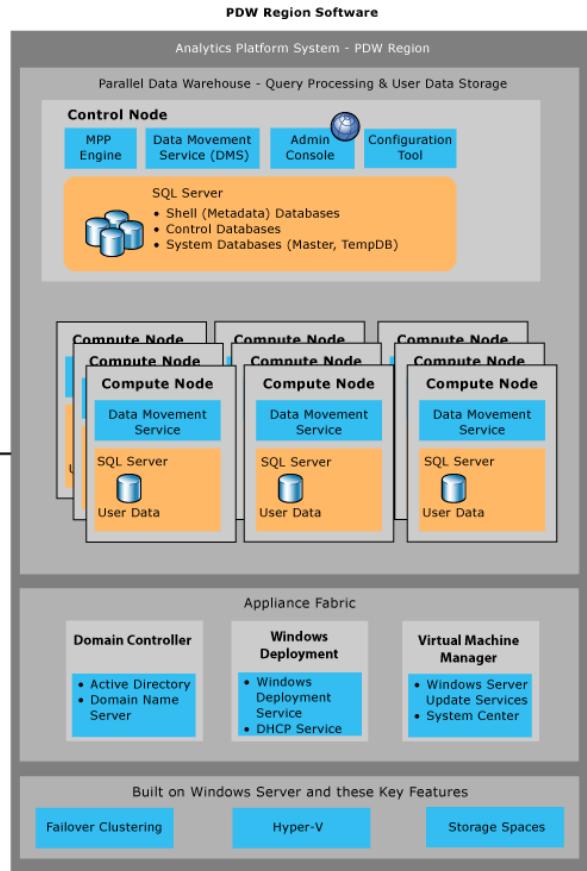
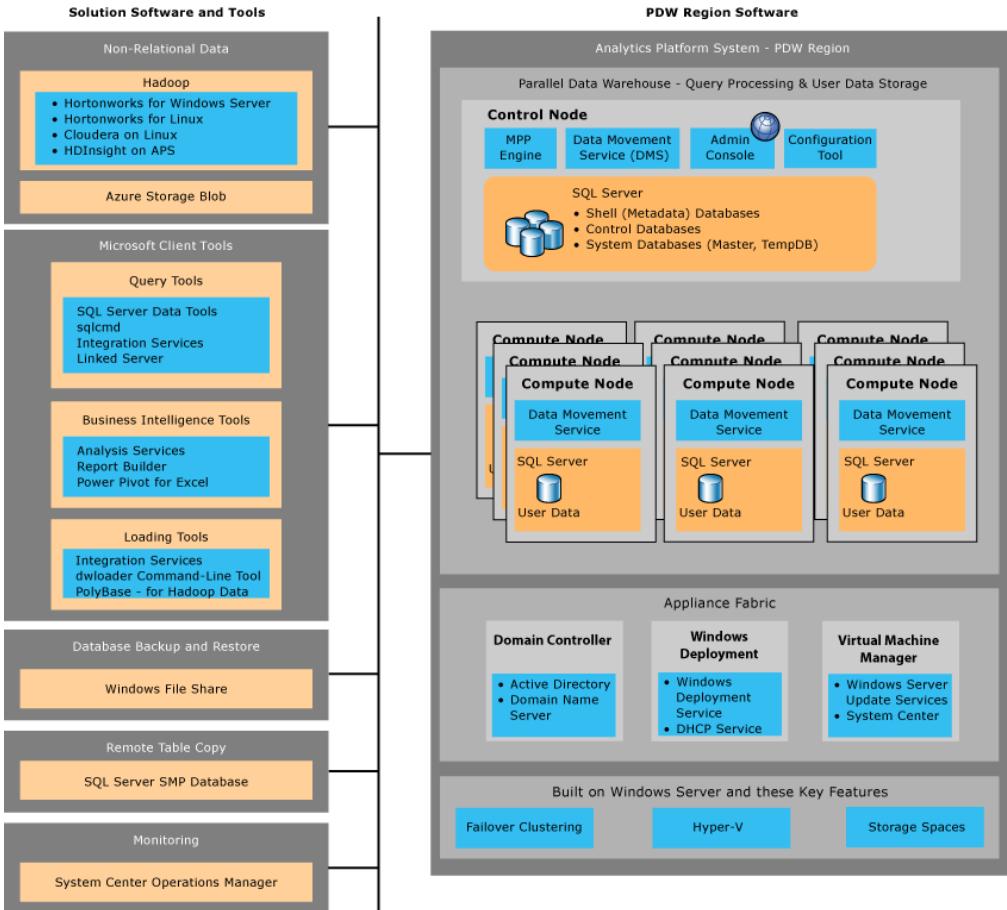




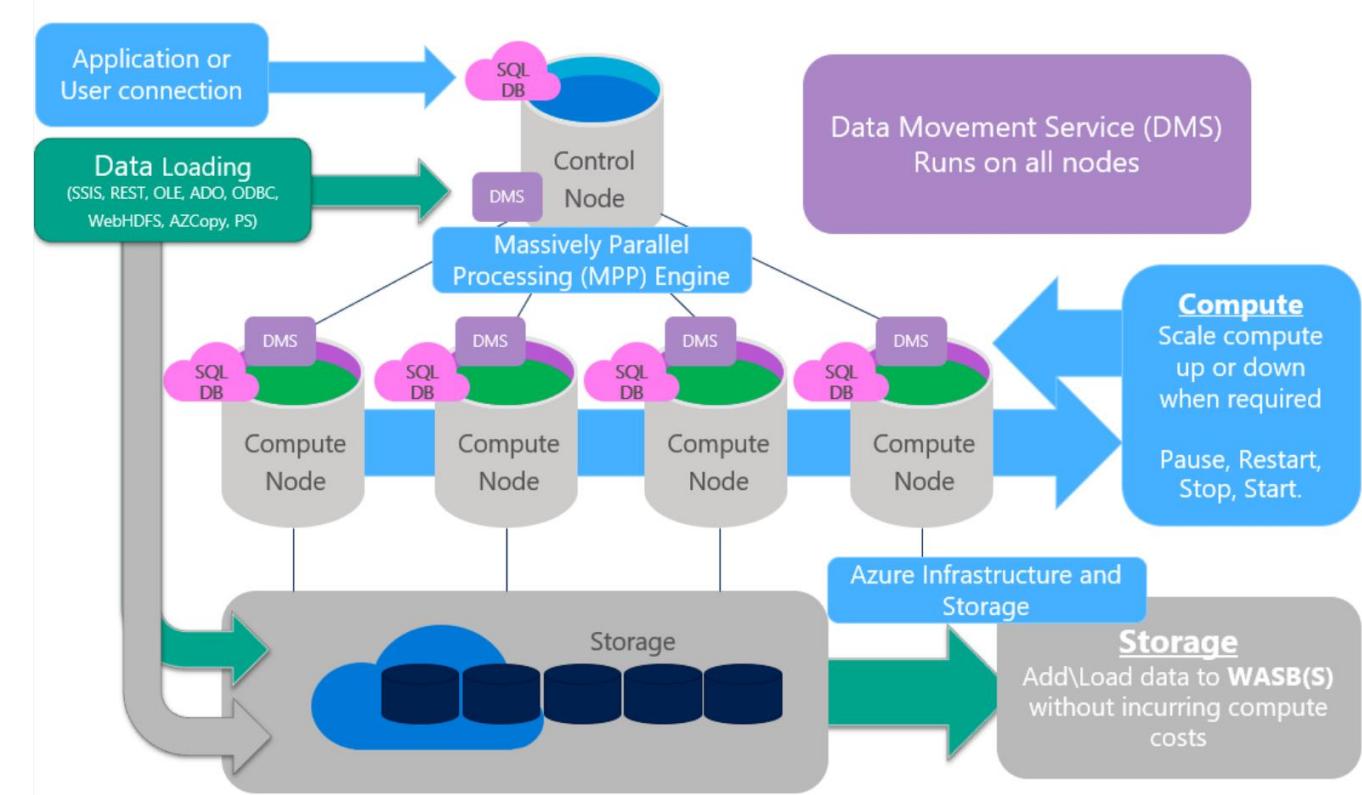
# Azure Synapse Analytics

## Synapse dedicated SQL pool

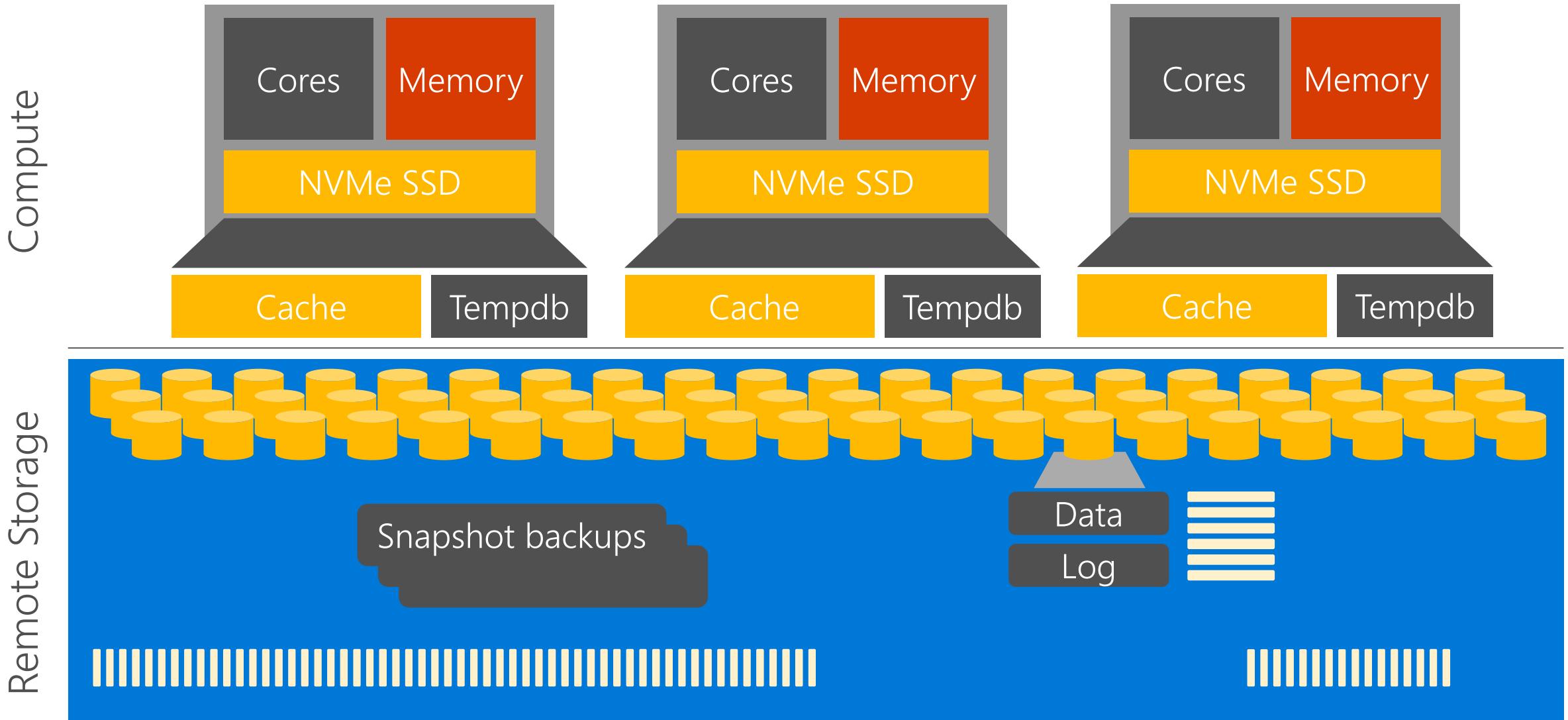
# PDW Architecture (Parallel Data Warehouse)



# Synapse SQL Pool Architecture

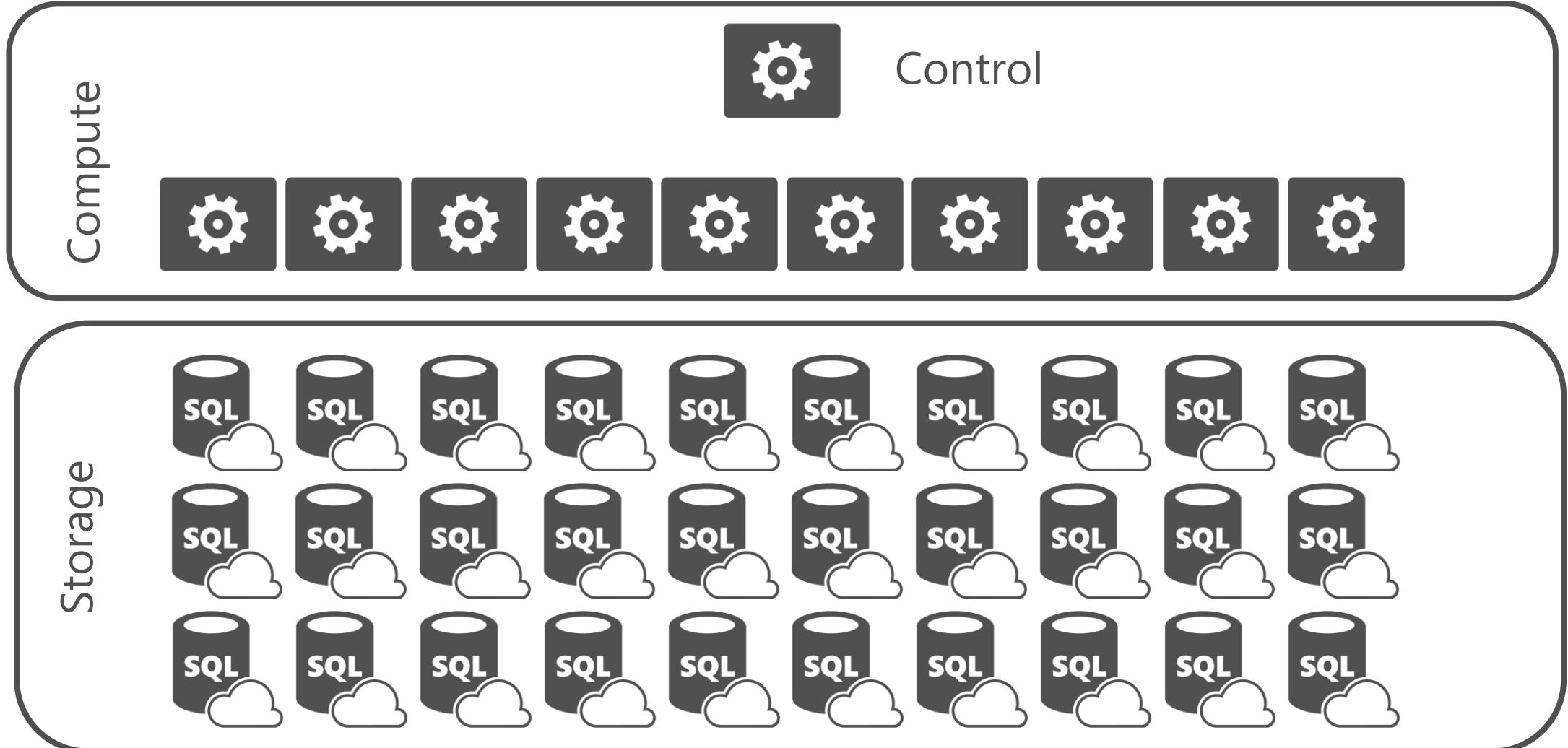


# TIERED STORAGE MODEL AUTOMATES DATA TEMPERATURE

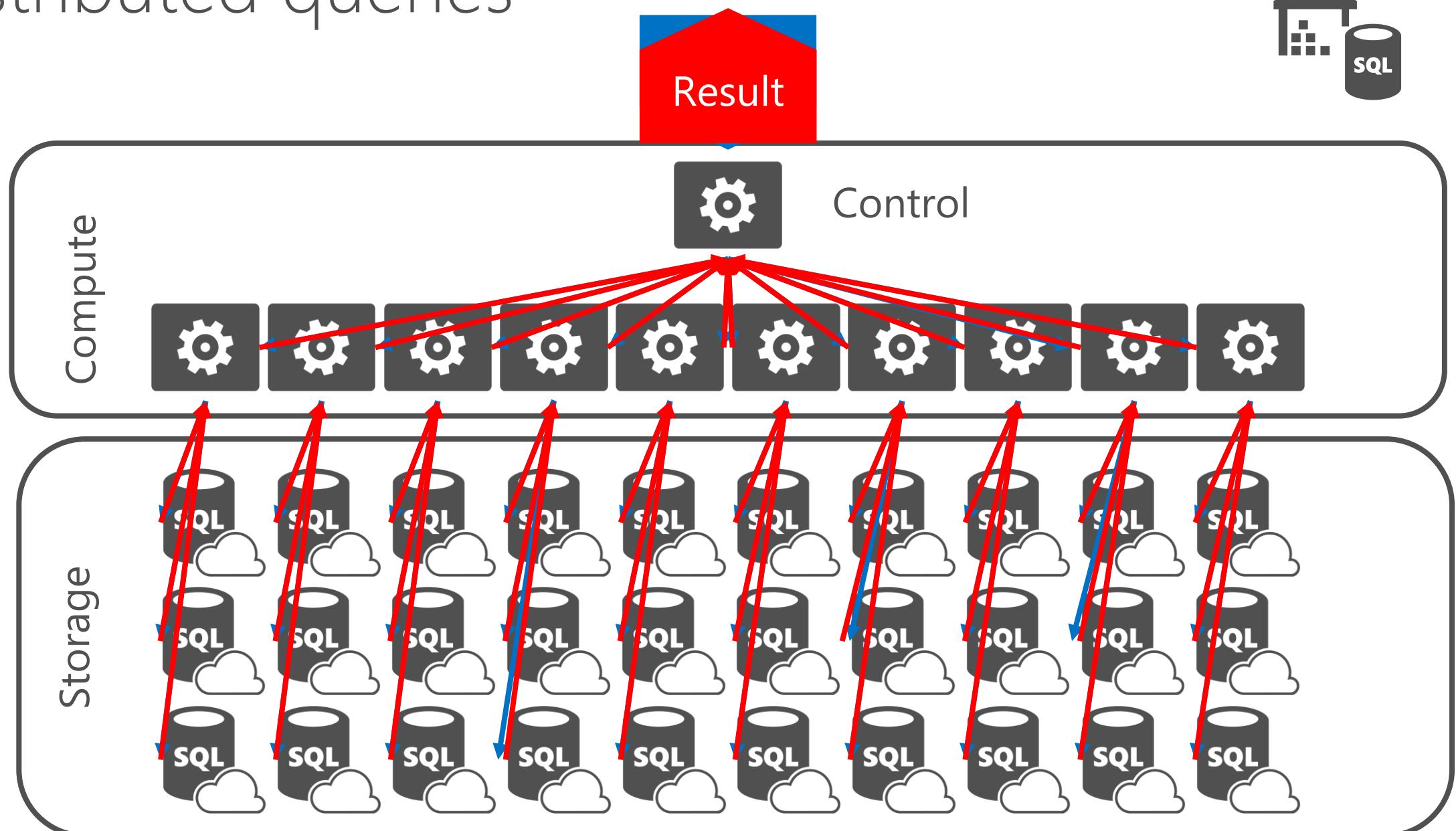


# Synapse Analytics Architectural overview

# Logical overview



# Distributed queries

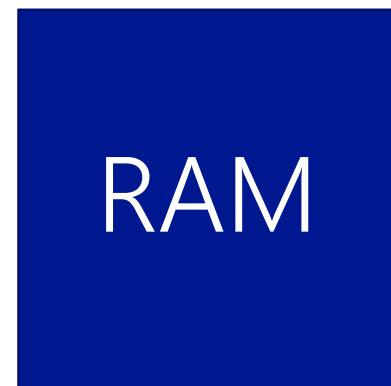
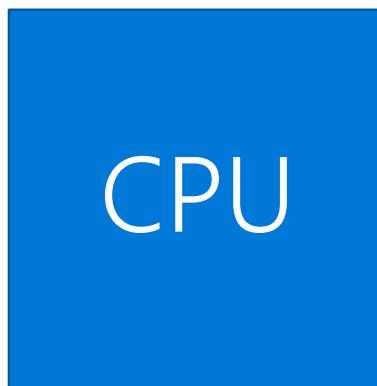


# Scale operations

# Data Warehouse Units

Normalized amount of compute

Converts to billing units i.e. what you pay



DWUc
100
200
300
400
500
600
1000
1200
1500
2000
3000
6000
...
30000

# Management Interfaces

	Portal	PowerShell	T-SQL
Create	✓	✓	✗
Pause and resume	✓	✓	✗
Scale	✓	✓	✓

# Scaling via T-SQL and PowerShell

T-SQL

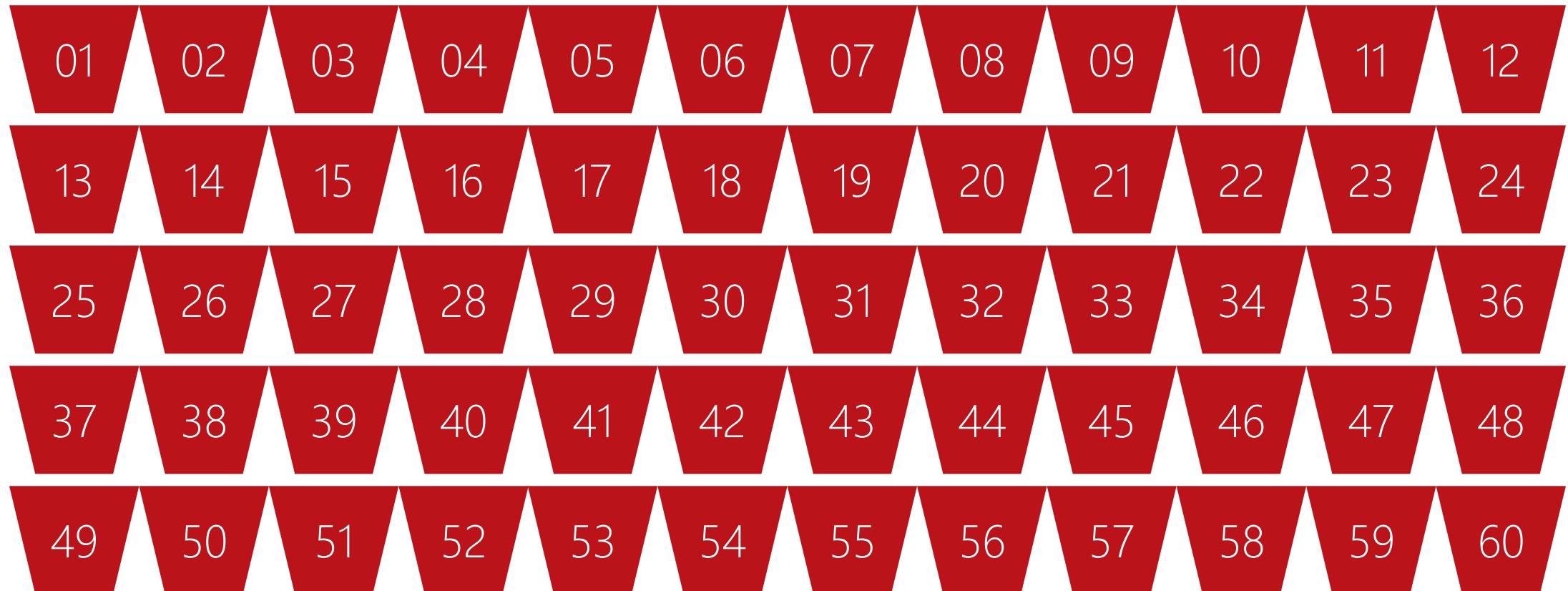
```
ALTER DATABASE ContosoRetailDW  
MODIFY (service_objective = 'DW100');
```

PowerShell

```
Set-AzureRmSqlDatabase  
-ResourceGroupName "RG_name"  
-ServerName "SRV_name"  
-DatabaseName "DB_name"  
-RequestedServiceObjectiveName "Dw1000"
```

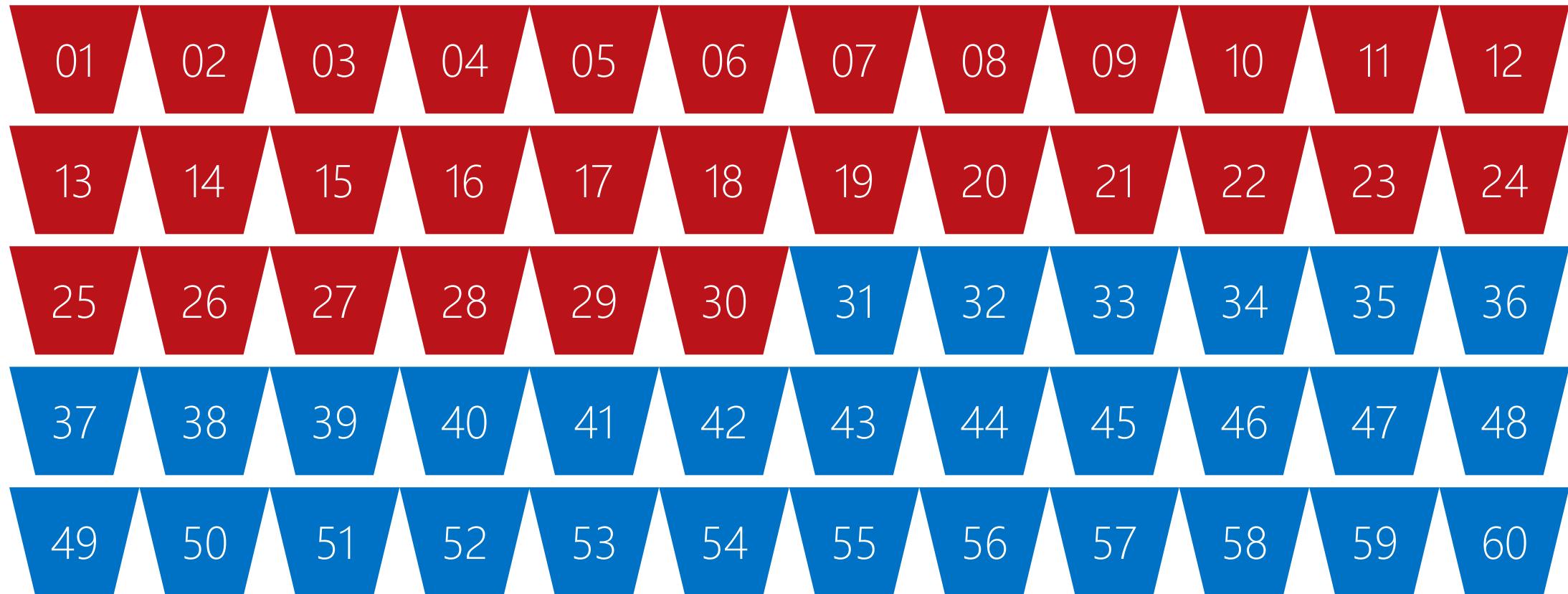
# Mapping Compute in SQLDW

DW100



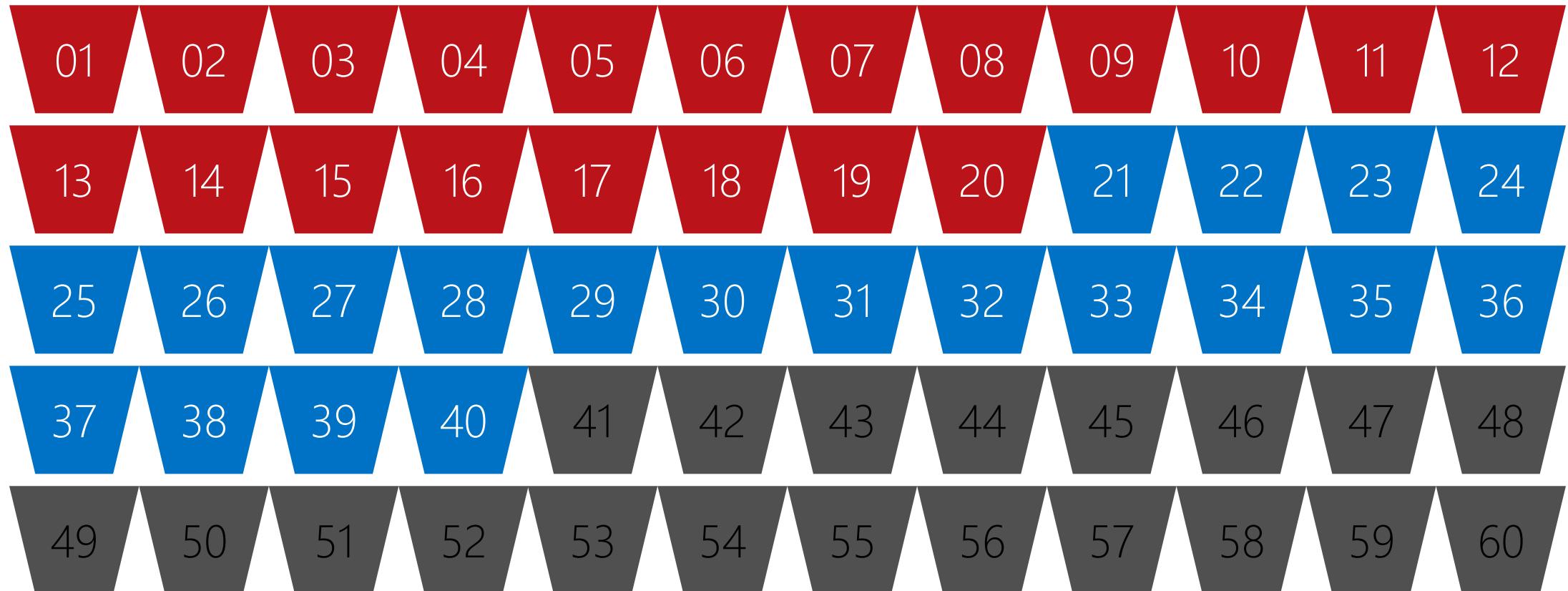
# Mapping Compute in SQLDW

DW1000



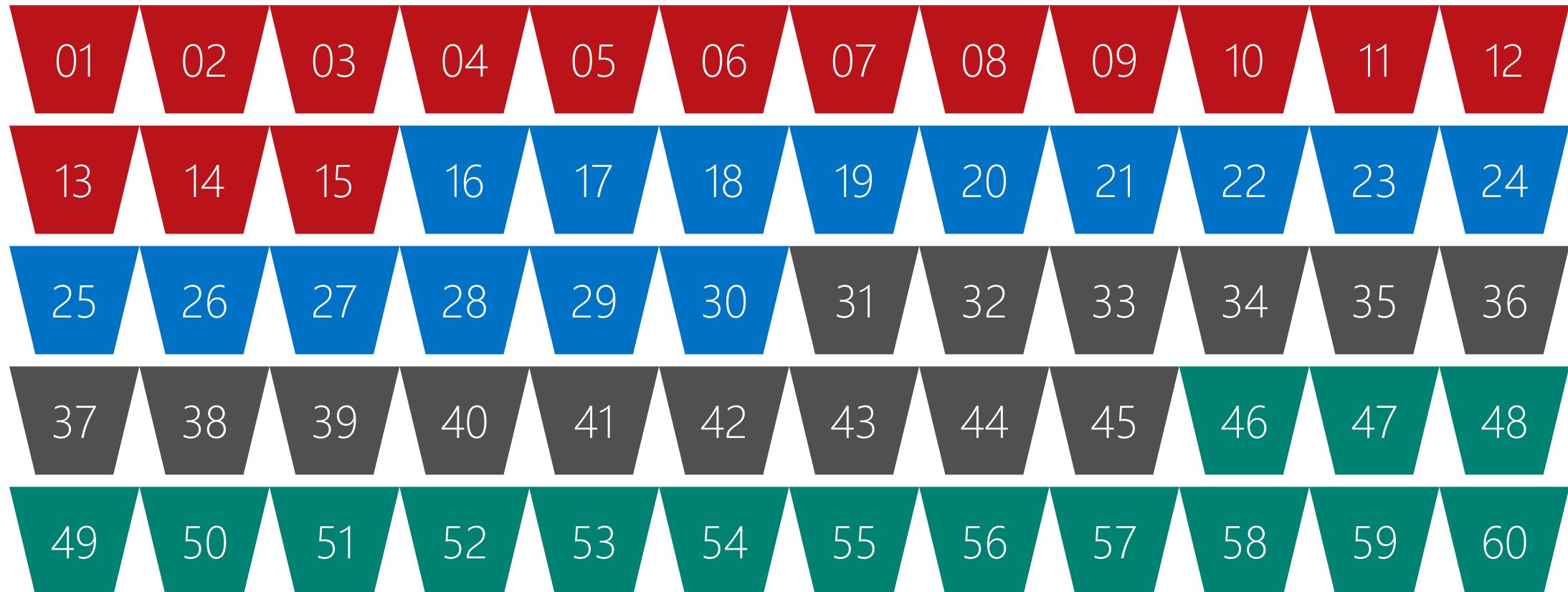
# Mapping Compute in S QLDW

DW1500



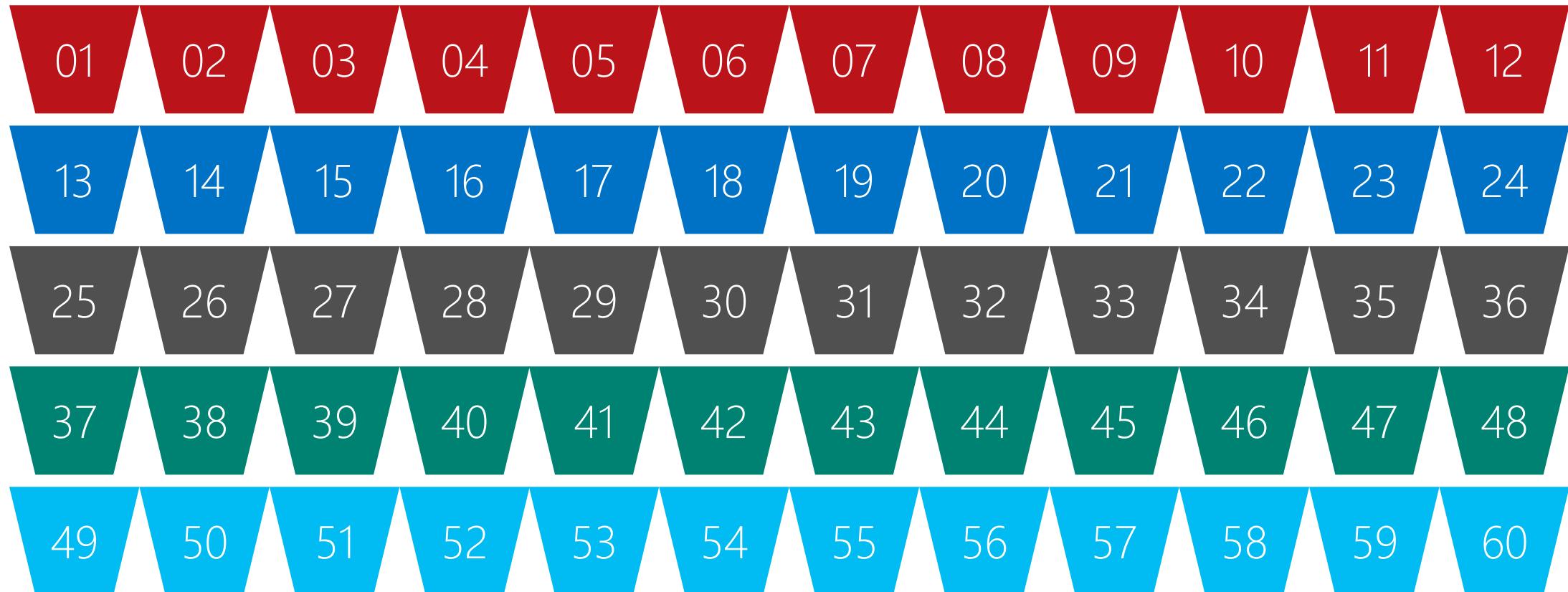
# Mapping Compute in SQLDW

DW2000



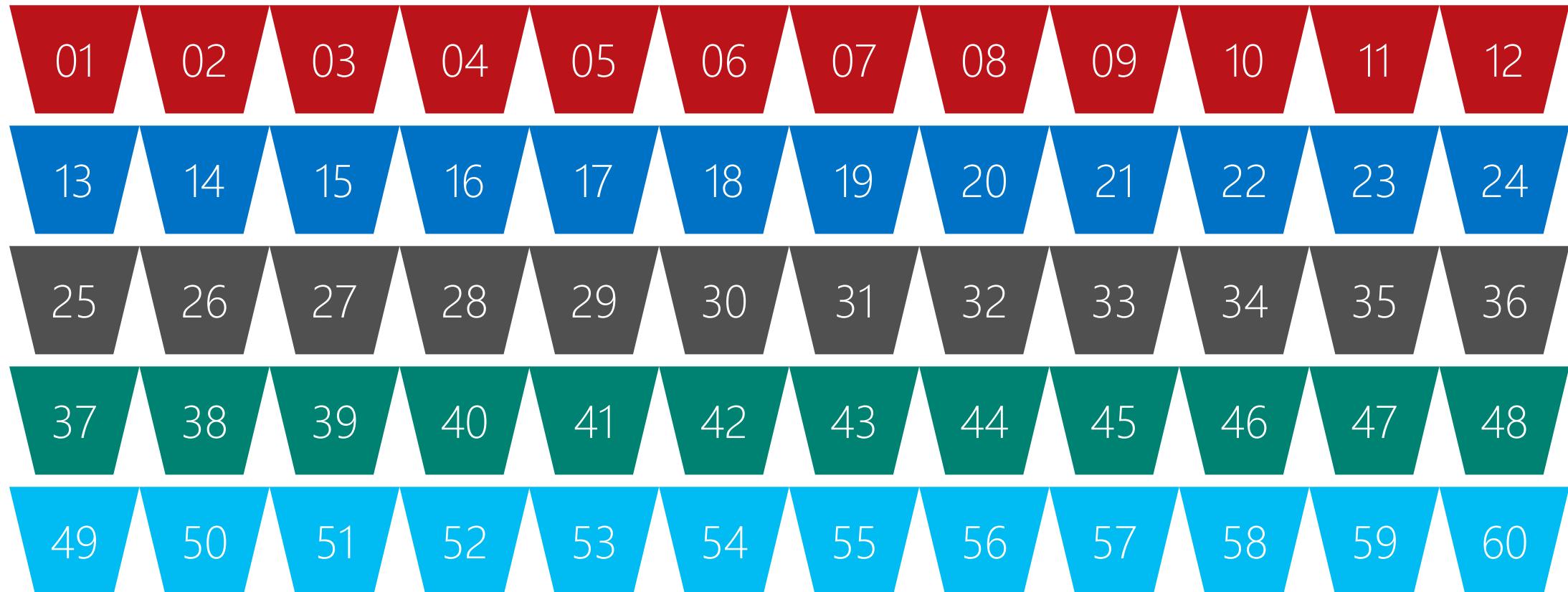
# Mapping Compute in SQLDW

DW2500



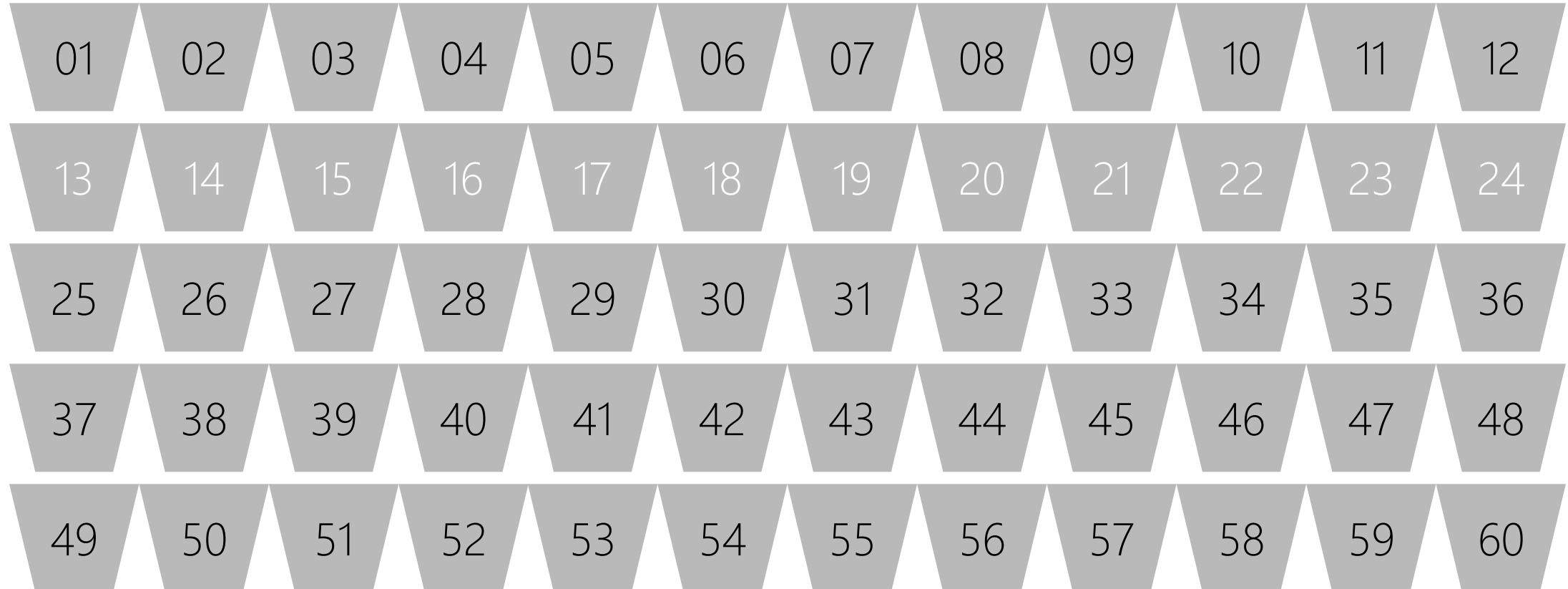
# Pausing compute in SQLDW

DW2500



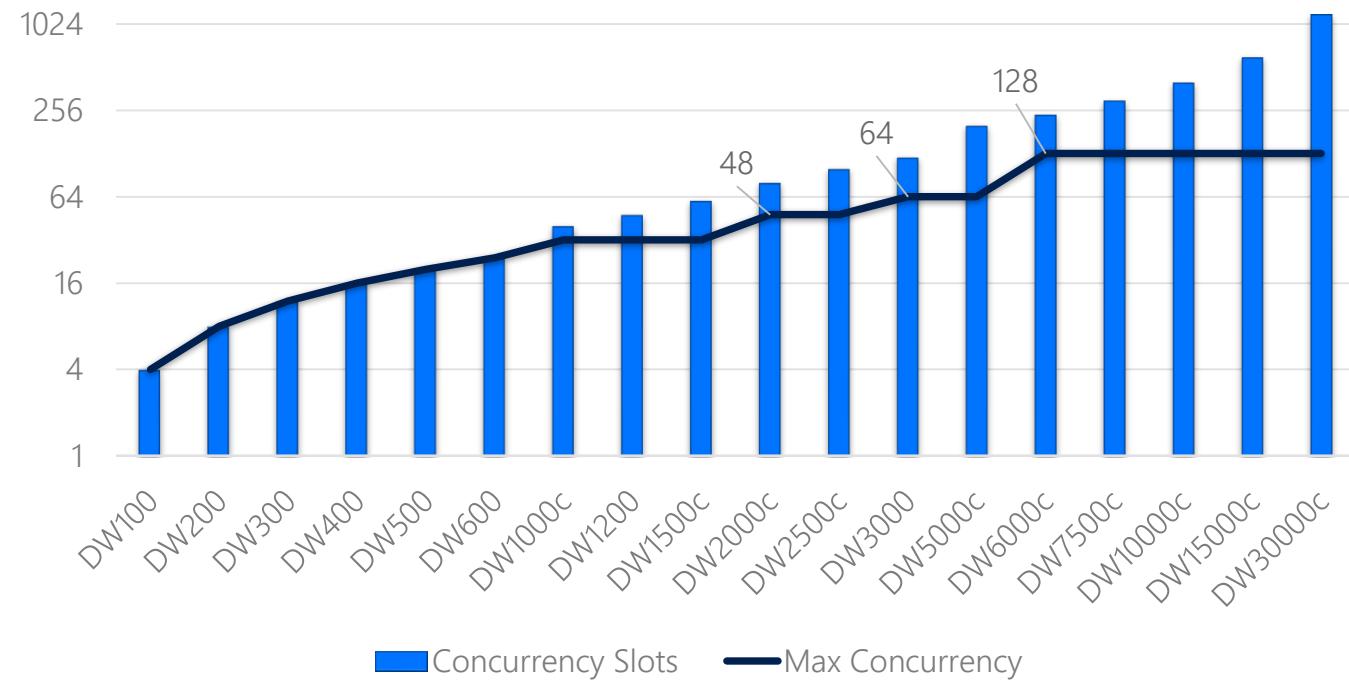
# Resuming compute in SQLDW

DW2500



# Concurrency

# Concurrent Query and Slots



서비스 수준	최대 동시 쿼리 수	사용 가능한 동시성 슬롯 수	쿼리당 필요한 최소의 리소스% (작업 그룹 사용시)
DW100c	4	4	25%
DW200c	8	8	12.5%
DW300c	12	12	8%
DW400c	16	16	6.25%
DW500c	20	20	5%
DW1000c	32	40	3%
DW1500c	32	60	3%
DW2000c	48	80	2%
DW2500c	48	100	2%
DW3000c	64	120	1.5%
DW5000c	64	200	1.5%
DW6000c	128	240	0.75%
DW7500c	128	300	0.75%
DW10000c	128	400	0.75%
DW15000c	128	600	0.75%
DW30000c	128	1200	0.75%

# Resource Class

## 정적 리소스 클래스

- 데이터의 집합 또는 크기가 고정적일 때 사용
- 현재 성능 수준에 관계없이 동일한 양의 메모리를 할당
- 서비스 성능 수준을 높이면 더 많은 쿼리를 수행 가능
- 정적 리소스 클래스 종류
  - Staticrc10, Staticrc20, Staticrc30, Staticrc40, Staticrc50, Staticrc60, Staticrc70, Staticrc80

## 동적 리소스 클래스

- 데이터 양이 증가하거나 변화하는 경우에 적합
- 서비스 수준을 높이면 쿼리에 더 많은 메모리 할당
- 동적 리소스 클래스 종류
  - smallrc, mediumrc, largerc, xlargerc
- 리소스 클래스 메모리 할당

서비스 수준	smallrc	mediumrc	largerc	xlargerc
DW100c	25%	25%	25%	70%
DW200c	12.5%	12.5%	22%	70%
DW300c	8%	10%	22%	70%
DW400c	6.25%	10%	22%	70%
DW500c	5%	10%	22%	70%
DW1000c – DW3000c	3%	10%	22%	70%

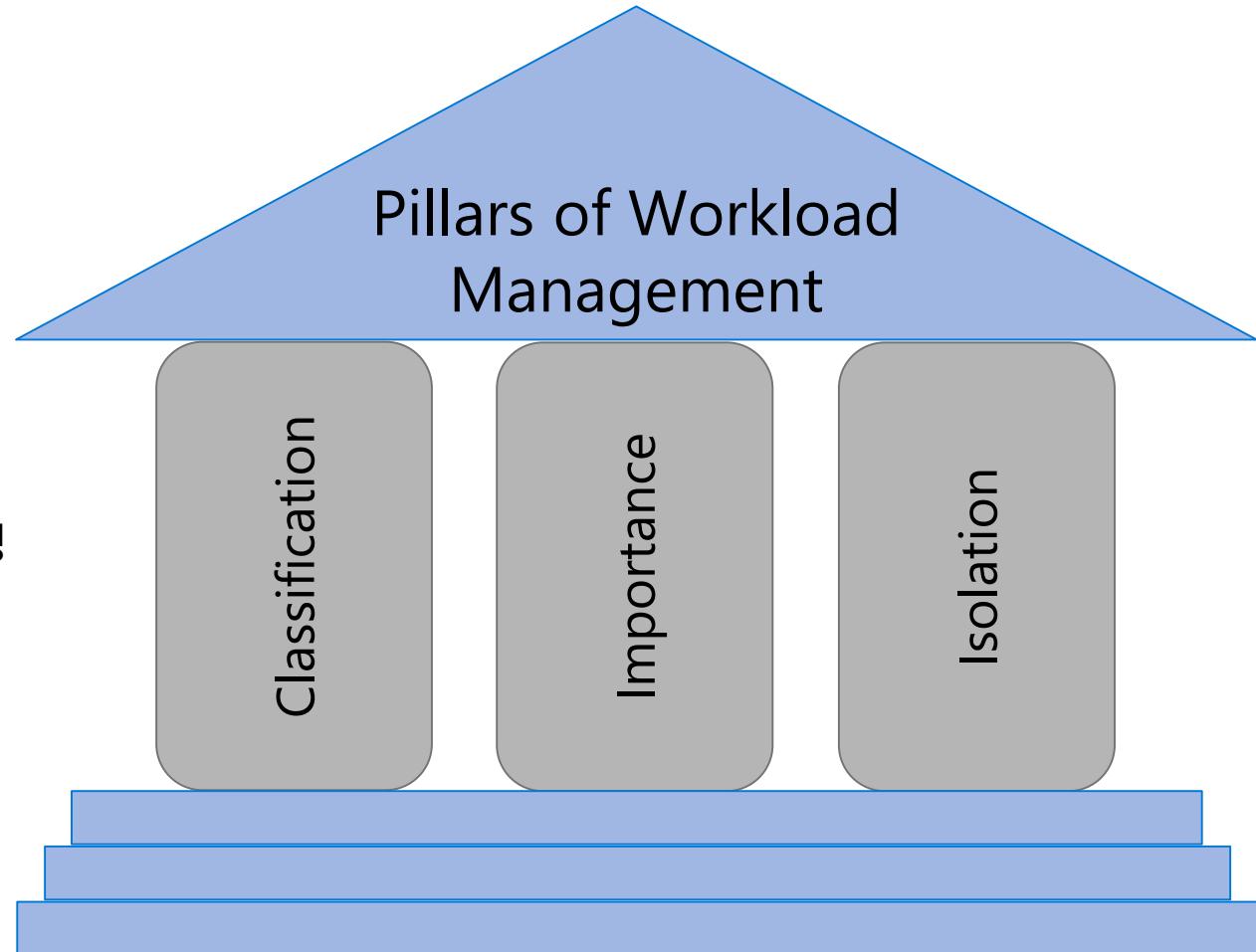
# Workload Management

## Overview

리소스의 효율적인 관리를 위해 사용

## 워크로드 관리 3대 요소

- 워크로드 분류 - 작업 요청을 그룹에 할당하고 중요도 설정
- 워크로드 중요도 - 그룹에 설정된 중요도에 따라 리소스의 액세스에 영향 받음
- 워크로드 독립성 - 워크로드 그룹에 리소스 예약



# Workload Isolation

## Overview

- 워크로드 그룹에 고정 리소스를 할당
- 로드 중인 다양한 리소스에 대한 최대 및 최소 사용량을 할당하며 Synapse SQL을 Offline하지 않고 리소스 조정 가능

## Benefits

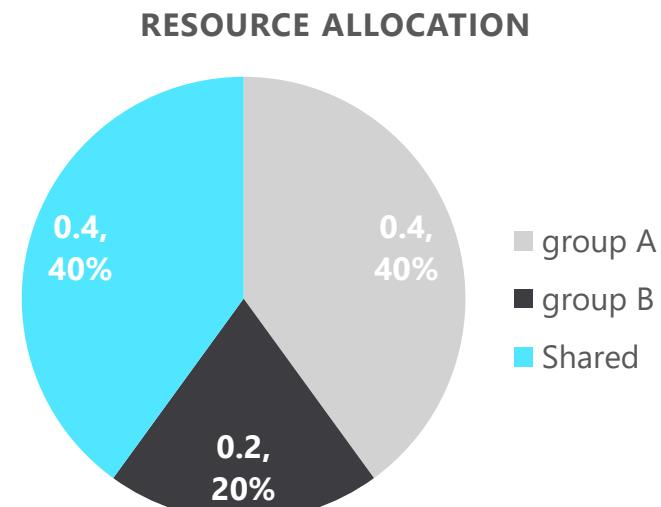
- 요청 그룹에 대한 리소스 예약
- 요청 그룹이 사용할 수 있는 리소스 양 제한
- 중요도에 따라 액세스되는 공유 리소스
- 쿼리 타임아웃 설정에 의한 DBA 업무 감소

## Monitoring DMVs

[sys.workload\\_management\\_workload\\_groups](#)

Query to view configured workload group.

```
CREATE WORKLOAD GROUP group_name
WITH
(
    MIN_PERCENTAGE_RESOURCE = value
    , CAP_PERCENTAGE_RESOURCE = value
    , REQUEST_MIN_RESOURCE_GRANT_PERCENT = value
    [[,] REQUEST_MAX_RESOURCE_GRANT_PERCENT = value ]
    [[,] IMPORTANCE = {LOW | BELOW_NORMAL | NORMAL | ABOVE_NORMAL | HIGH} ]
    [[,] QUERY_EXECUTION_TIMEOUT_SEC = value ]
)[;]
```



# Workload Classification & Importance

## 워크로드 분류

- 미리 생성된 워크로드의 분류에 따라 리소스를 매핑
- 워크로드 중요도와 함께 사용하여 리소스를 효과적으로 공유 가능

## 워크로드 중요도

- 워크로드 중요도에 따라 대기열에 관계없이 우선 순위가 높은 쿼리가 리소스를 즉시 액세스 할 수 있음
- 워크로드 중요도를 생략했을 경우 중요도는 NORMAL로 설정됨

## Monitoring DMVs

[sys.workload\\_management\\_workload\\_classifiers](#)

[sys.workload\\_management\\_workload\\_classifier\\_details](#)

Query DMVs to view details about all active workload classifiers.

```
CREATE WORKLOAD CLASSIFIER classifier_name
WITH
(
    WORKLOAD_GROUP = 'name'
    , MEMBERNAME = 'security_account'
    [,] IMPORTANCE = {LOW|BELOW_NORMAL|NORMAL|ABOVE_NORMAL|HIGH} []
    [,] WLM_LABEL = 'label'
    [,] WLM_CONTEXT = 'name'
    [,] START_TIME = 'start_time'
    [,] END_TIME = 'end_time'
)[;]
```

**WORKLOAD\_GROUP:** maps to an existing resource class

**IMPORTANCE:** specifies relative importance of request

**MEMBERNAME:** database user, role, AAD login or AAD group

# User & Authentication

-- Master Database

```
CREATE LOGIN MedRCLLogin WITH PASSWORD = 'rladydtjs$1971';
```

```
CREATE USER LoadingUser FOR LOGIN MedRCLLogin;
```

-- Sqlpool Database

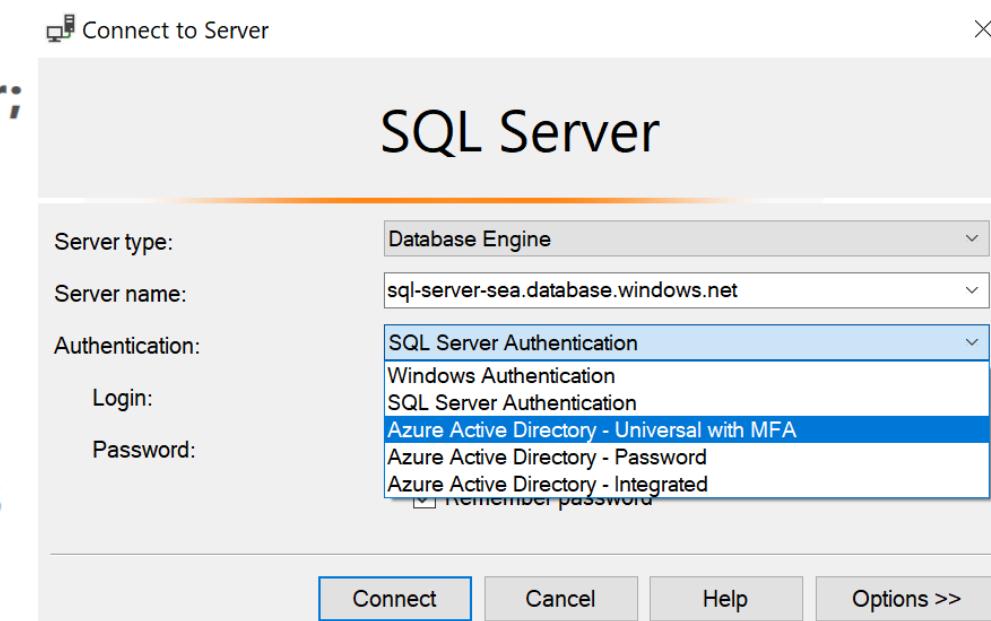
```
CREATE USER LoadingUser FOR LOGIN MedRCLLogin;
```

```
GRANT CONTROL ON DATABASE:[yongdw] to LoadingUser;
```

```
EXEC sp_addrolemember 'mediumrc', 'LoadingUser';
```

```
Create Login Mary@domainname.net From EXTERNAL PROVIDER;  

Create User Mary From Login Mary@domainname.net;
```



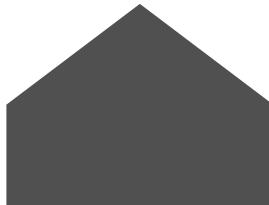
# Tables & Distribution

# Table Distribution Options

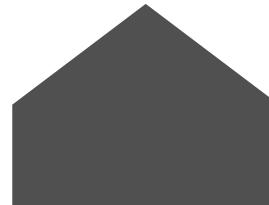
	Hash Distributed	Round Robin (Default)	Replicated
분산 방법	<ul style="list-style-type: none"><li>해시 알고리즘 리턴 값에 의해 노드로 데이터 분산</li><li>해시 알고리즘에 의한 결과값이 같으면 같은 노드에 데이터 분산</li><li>해쉬 키 값은 Single Column Only</li></ul>	<ul style="list-style-type: none"><li>모든 노드에 데이터가 분산</li><li>비용 대비 단순한 저장 방식</li></ul>	<ul style="list-style-type: none"><li>모든 노드에 데이터 복제</li></ul>
장점	<ul style="list-style-type: none"><li>아래 작업 수행 시 속도가 빠름</li></ul> <p>COUNT ( DISTINCT &lt;hashed_key&gt; ) OVER PARTITION BY &lt;hashed_key&gt; JOIN &lt;table_name&gt; ON &lt;hashed_key&gt; GROUP BY &lt;hashed_key&gt;</p>	<ul style="list-style-type: none"><li>초기 적재 속도가 빠름</li></ul>	<ul style="list-style-type: none"><li>쿼리 계획을 단순화하고 데이터 이동 감소</li><li>Hash 테이블과 조인 시 최적</li></ul>
사용 예	<ul style="list-style-type: none"><li>Fact Table</li></ul>	<ul style="list-style-type: none"><li>초기 데이터 이관 시 Data Load</li><li>스테이징 테이블에 사용</li></ul>	<ul style="list-style-type: none"><li>Dimension Table (Fact Table 을 설명해주는 Table)</li><li>Small Table</li></ul>
유의 사항	<ul style="list-style-type: none"><li>특정 노드로 데이터 쓸림 발생</li><li>NULS, Default 값 사용시 부적합</li><li>Hash key 값은 update 되면 안됨</li></ul>	<ul style="list-style-type: none"><li>쿼리 수행 시 노드간 데이터 이동 발생</li></ul>	<ul style="list-style-type: none"><li>한 노드에서 두개의 복제 테이블이 조인 시 많은 공간이 소비됨</li></ul>

# Creating tables : Round Robin, Hash

```
CREATE TABLE [build].[FactOnlineSales]
(
    [OnlineSalesKey]           int          NOT NULL
    , [DateKey]                datetime     NOT NULL
    , [StoreKey]               int          NOT NULL
    , [ProductKey]              int         NOT NULL
    , [PromotionKey]            int         NOT NULL
    , [CurrencyKey]             int         NOT NULL
    , [CustomerKey]             int         NOT NULL
    , [SalesOrderNumber]        nvarchar(20) NOT NULL
    , [SalesOrderLineNumber]   int          NULL
    , [SalesQuantity]            int         NOT NULL
    , [SalesAmount]              money        NOT NULL
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX
    , DISTRIBUTION = ROUND_ROBIN
)
;
```



```
CREATE TABLE [build].[FactOnlineSales]
(
    [OnlineSalesKey]           int          NOT NULL
    , [DateKey]                datetime     NOT NULL
    , [StoreKey]               int          NOT NULL
    , [ProductKey]              int         NOT NULL
    , [PromotionKey]            int         NOT NULL
    , [CurrencyKey]             int         NOT NULL
    , [CustomerKey]             int         NOT NULL
    , [SalesOrderNumber]        nvarchar(20) NOT NULL
    , [SalesOrderLineNumber]   int          NULL
    , [SalesQuantity]            int         NOT NULL
    , [SalesAmount]              money        NOT NULL
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX
    , DISTRIBUTION = HASH([ProductKey])
)
;
```



# Creating tables : Replicated

```
CREATE TABLE dbo.DimCustomer
(
    CustomerKey          int           NOT NULL
    , GeographyKey        int           NULL
    , CustomerAlternateKey nvarchar(15) NOT NULL
    , Title                nvarchar(8)  NULL
    , FirstName             nvarchar(50) NULL
    , LastName              nvarchar(50) NULL
    , BirthDate             date         NULL
    , Gender                nvarchar(1)  NULL
    , EmailAddress          nvarchar(50) NULL
    , YearlyIncome          money        NULL
    , DateFirstPurchase     date         NULL
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX
    , DISTRIBUTION = REPLICATED
)
```



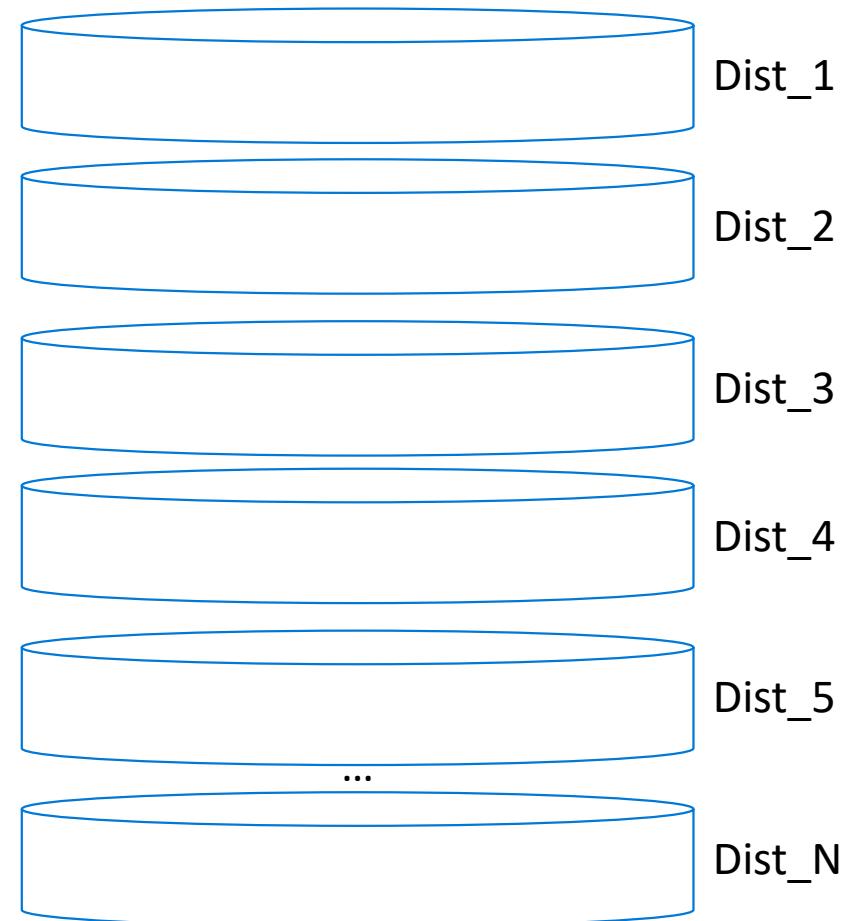
# Hash Distributed

```
CREATE TABLE ProductSales  
WITH (DISTRIBUTION=HASH(AccountID))  
AS ...
```

ProductSales – Raw Data

Account	SalesAmt	...
47	\$1,234.36	...
36	\$2,345.47	...
14	\$3,456.58	...
25	\$4,567.69	...
48	\$5,678.70	...
37	\$6,789.81	...
51	\$6,789.81	...
...	...	...

Hash(40)

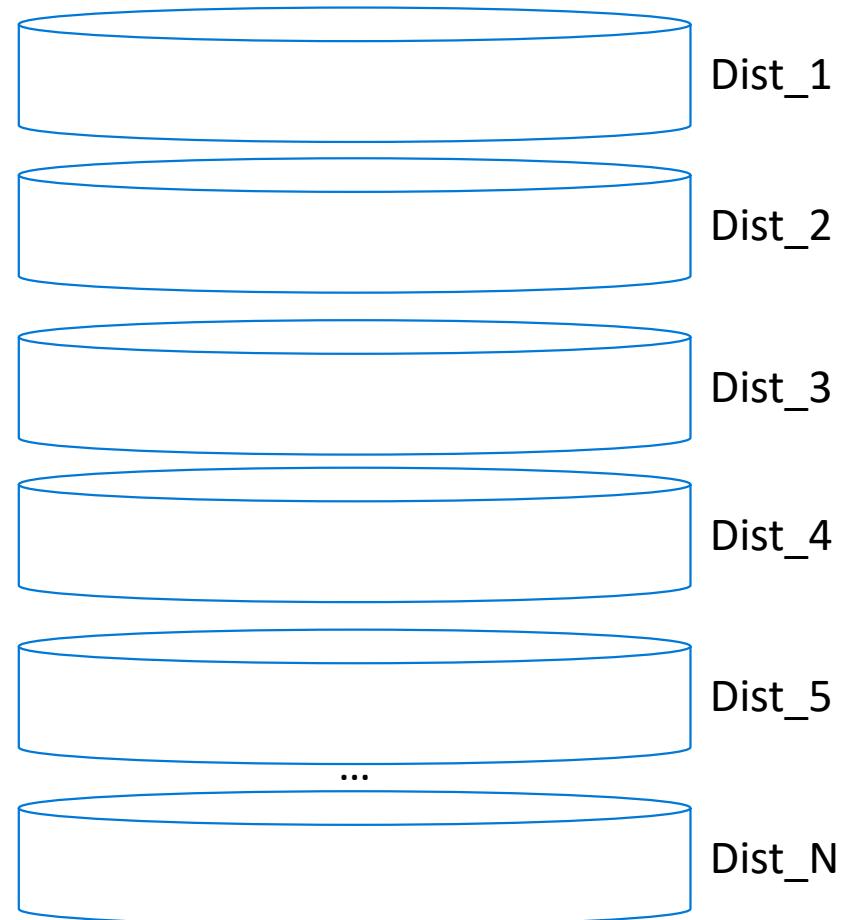


# Round Robin Distributed

```
CREATE TABLE ProductSales  
WITH (DISTRIBUTION = ROUND_ROBIN)  
AS ...
```

ProductSales – Raw Data

AccountID	SalesAmt	...
47	\$1,234.36	...
36	\$2,345.47	...
14	\$3,456.58	...
25	\$4,567.69	...
47	\$5,678.70	...
37	\$6,789.81	...
42	\$1,632.25	...
88	\$4,453.21	...
52	\$7,892.81	...
91	\$9,549.64	...
88	\$2,498.14	...
42	\$3,145.99	...
23	\$5,145.99	...



# 데이터의 이동은 왜 발생하는가?

Data has to be co-located to operated on...

일반 원인:

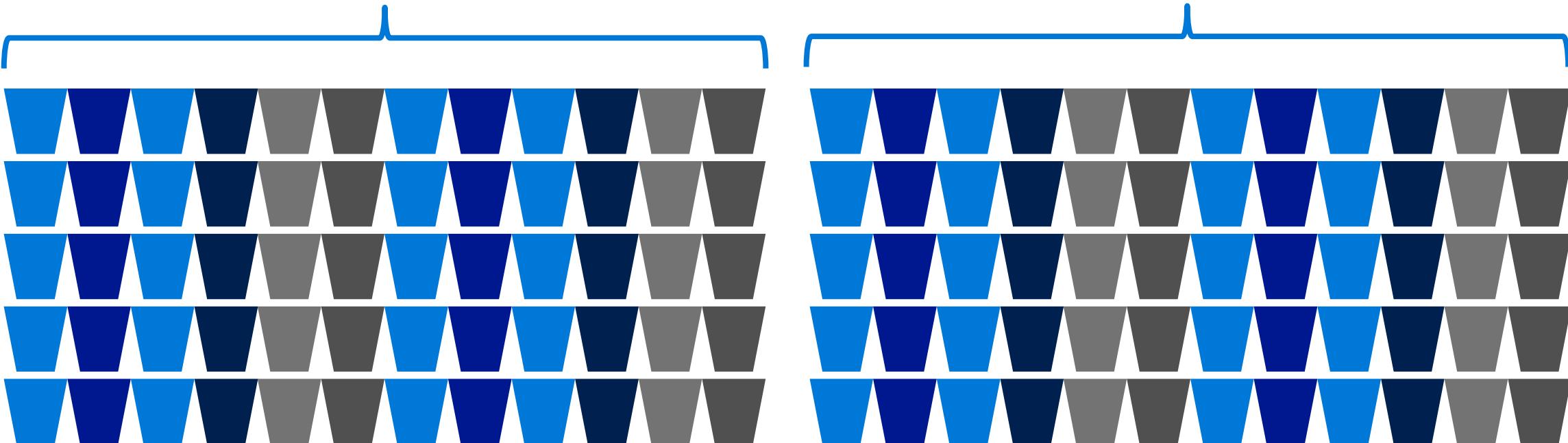
호환 불가능한 조인

호환 불가능한 집계

# HASH table Join – 데이터 이동 미발생

Store\_Sales HASH([ProductKey])  
[ProductKey] INT NULL

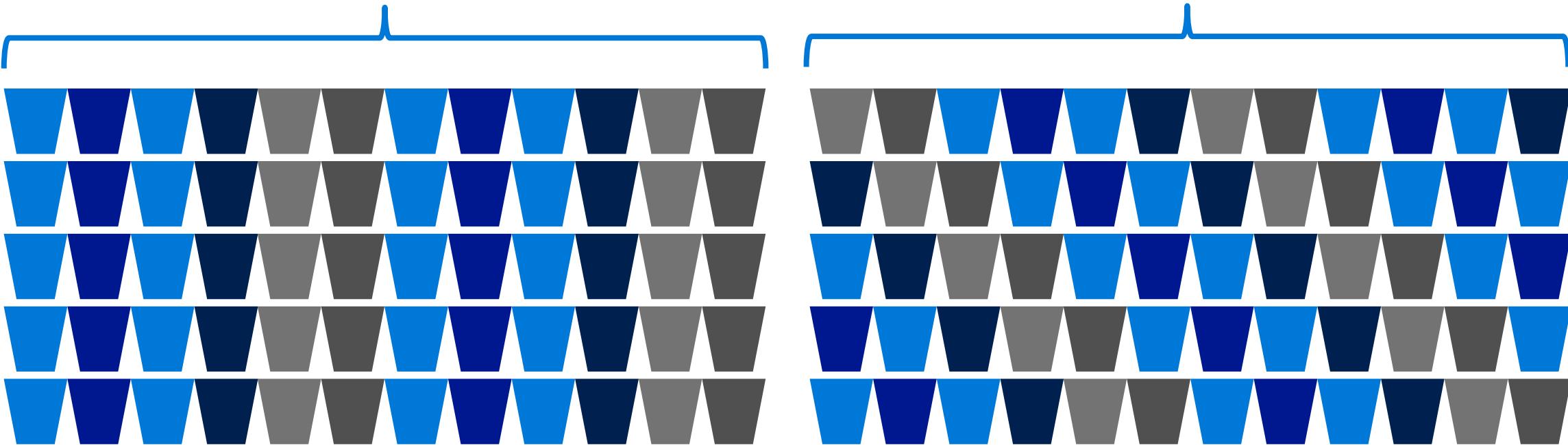
Web\_Sales HASH([ProductKey])  
[ProductKey] INT NULL



# HASH table Join – 데이터 이동 발생

Store\_Sales HASH([ProductKey])  
[ProductKey] INT NULL

Web\_Sales HASH([ProductKey])  
[ProductKey] **BIGINT** NULL



# 데이터 이동 Case Study

## Aggregation - Compatible

Resolved completely on each compute node  
No Data Movement

1. Hash Distribution Key is contained in the group by keys
2. Count Distinct on distribution key

-- FactOnlineSales distributed on ProductKey

```
SELECT COUNT_BIG(*)  
FROM [cso].[FactOnlineSales]  
GROUP BY [ProductKey]
```

```
SELECT COUNT_BIG(DISTINCT ([ProductKey]))  
FROM [cso].[FactOnlineSales]
```

## Aggregation - Incompatible

Partially aggregated on each node  
Shuffle move co-locates rows with same group by key

1. Table is round robin distributed
2. Hash Distribution key is not contained in group by keys
3. Count Distinct on non-distribution key or on round robin table

-- FactOnlineSales distributed on ProductKey

```
SELECT COUNT_BIG(*)  
FROM [cso].[FactOnlineSales]  
GROUP BY [StoreKey]
```

```
SELECT COUNT_BIG(DISTINCT [DateKey])  
FROM [cso].[FactOnlineSales]
```

# Data Movement Types for a Query

DMS Operation	Description
ShuffleMoveOperation	Distribution → Hash algorithm → New distribution Changing the distribution column in preparation for join.
PartitionMoveOperation	Distribution → Control Node Aggregations - count(*) is count on nodes, sum of count
BroadcastMoveOperation	Distribution → Copy to all distributions Changes distributed table to replicated table for join.
TrimMoveOperation	Replicated table → Hash algorithm → Distribution When a replicated table needs to become distributed. Needed for outer joins.
MoveOperation	Control Node → Copy to all distributions Data moved from Control Node back to Compute Nodes resulting in a replicated table for further processing.

# Tables – Partitions

## Overview

- 테이블 데이터를 작은 그룹으로 분리하는 작업
- 대부분 날짜 컬럼을 파티션 키를 사용
- 모든 테이블에 파티션 가능 지원

RANGE RIGHT – Used for time partitions

RANGE LEFT – Used for number partitions

## Benefits

- 데이터 로딩, 조회에 효율성 및 속도 증가
- 데이터 조회 시, 불필요한 스캔의 감소와 이로 인한 I/O 의 감소로 상당한 쿼리 속도 증가

```

CREATE TABLE partitionedOrderTable
(
    OrderId INT NOT NULL,
    Date DATE NOT NULL,
    Name VARCHAR(2),
    Country VARCHAR(2)
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX,
    DISTRIBUTION = HASH([OrderId]),
    PARTITION (
        [Date] RANGE RIGHT FOR VALUES (
            '2000-01-01', '2001-01-01', '2002-01-01',
            '2003-01-01', '2004-01-01', '2005-01-01'
        )
    )
);

```

# Tables – DW Distributions & Partitions Illustrated

## 논리적 Table 구조

OrderId	Date	Name	Country
85016	11-2-2018	V	UK
85018	11-2-2018	Q	SP
85216	11-2-2018	Q	DE
85395	11-2-2018	V	NL
82147	11-2-2018	Q	FR
86881	11-2-2018	D	UK
93080	11-3-2018	R	UK
94156	11-3-2018	S	FR
96250	11-3-2018	Q	NL
98799	11-3-2018	R	NL
98015	11-3-2018	T	UK
98310	11-3-2018	D	DE
98979	11-3-2018	Z	DE
98137	11-3-2018	T	FR
...	...	...	...

## 물리적 Data distribution

( Hash distribution (OrderId), Date partitions )

### Distribution1 (OrderId 80,000 – 100,000)

#### 11-2-2018 partition

OrderId	Date	Name	Country
85016	11-2-2018	V	UK
85018	11-2-2018	Q	SP
85216	11-2-2018	Q	DE
85395	11-2-2018	V	NL
82147	11-2-2018	Q	FR
86881	11-2-2018	D	UK
...	...	...	...

#### 11-3-2018 partition

OrderId	Date	Name	Country
93080	11-3-2018	R	UK
94156	11-3-2018	S	FR
96250	11-3-2018	Q	NL
98799	11-3-2018	R	NL
98015	11-3-2018	T	UK
98310	11-3-2018	D	DE
98979	11-3-2018	Z	DE
98137	11-3-2018	T	FR
...	...	...	...

...

x 60 distributions (shards)

- 각 Shard 는 같은 Date로 파티션 되어있음
- 최적의 압축과 Clustered Columnstore table의 퍼포먼스 향상을 위해서는 각 파티션에 최소 100만 건의 row가 필요

# Tables – Partitions

	logical_table_name	row_group_id	state	state_desc	total_rows	trim_reason_desc	physical_name
1	FactSalesQuotaCCI	3	3	COMPRESSED	514925	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_1
2	FactSalesQuotaCCI	3	3	COMPRESSED	512875	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_2
3	FactSalesQuotaCCI	3	3	COMPRESSED	511350	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_3
4	FactSalesQuotaCCI	3	3	COMPRESSED	509161	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_4
5	FactSalesQuotaCCI	3	3	COMPRESSED	512640	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_5
6	FactSalesQuotaCCI	3	3	COMPRESSED	512640	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_6
7	FactSalesQuotaCCI	3	3	COMPRESSED	512638	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_7
8	FactSalesQuotaCCI	3	3	COMPRESSED	511216	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_8
9	FactSalesQuotaCCI	3	3	COMPRESSED	511216	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_9
10	FactSalesQuotaCCI	3	3	COMPRESSED	512360	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_10
11	FactSalesQuotaCCI	3	3	COMPRESSED	511216	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_11
12	FactSalesQuotaCCI	3	3	COMPRESSED	510102	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_12
13	FactSalesQuotaCCI	3	3	COMPRESSED	509792	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_13
14	FactSalesQuotaCCI	3	3	COMPRESSED	512640	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_14
15	FactSalesQuotaCCI	3	3	COMPRESSED	512388	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_15
16	FactSalesQuotaCCI	3	3	COMPRESSED	513046	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_16
17	FactSalesQuotaCCI	3	3	COMPRESSED	513105	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_17
18	FactSalesQuotaCCI	3	3	COMPRESSED	512640	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_18
19	FactSalesQuotaCCI	3	3	COMPRESSED	512607	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_19
20	FactSalesQuotaCCI	3	3	COMPRESSED	511583	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_20
21	FactSalesQuotaCCI	3	3	COMPRESSED	512156	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_21
22	FactSalesQuotaCCI	3	3	COMPRESSED	512479	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_22
23	FactSalesQuotaCCI	3	3	COMPRESSED	511480	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_23
24	FactSalesQuotaCCI	3	3	COMPRESSED	512171	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_24
25	FactSalesQuotaCCI	3	3	COMPRESSED	512753	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_25
26	FactSalesQuotaCCI	3	3	COMPRESSED	511216	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_26
27	FactSalesQuotaCCI	3	3	COMPRESSED	509843	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_27
28	FactSalesQuotaCCI	3	3	COMPRESSED	511065	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_28
29	FactSalesQuotaCCI	3	3	COMPRESSED	510639	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_29
30	FactSalesQuotaCCI	3	3	COMPRESSED	510366	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_30
31	FactSalesQuotaCCI	3	3	COMPRESSED	511247	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_31
32	FactSalesQuotaCCI	3	3	COMPRESSED	511917	BULKLOAD	Table_0916c48a65e14db285dfed9c57386023_32

# Tables – Indexes

## Clustered Columnstore index (Default Primary)

- 최적의 데이터 압축 지원
- 최고의 쿼리 퍼포먼스 지원

## Clustered index (Primary)

- 단일 행, 극소수 행을 조회할 때 최적  
(Multi-Column Index 가능, But 1개 이상의 Index 생성 불가)

## Heap (Primary)

- 임시 데이터를 Loading 시 최적
- 사이즈가 작은 Look up Table 조회 시 사용

## Nonclustered indexes (Secondary)

- Table에 Multiple Columns Index 를 허용
- Single Table에 여러 개의 Multiple Non-Clustered 허용
- Primary Indexes 대안으로 Non-Clustered Indexes 생성 가능
- Lookup 테이블 조회에 성능 기대

-- Create table with index

```
CREATE TABLE orderTable
(
    OrderId INT NOT NULL,
    Date DATE NOT NULL,
    Name VARCHAR(2),
    Country VARCHAR(2)
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX |
    HEAP |
    CLUSTERED INDEX (OrderId)
);
```

-- Add non-clustered index to table

```
CREATE INDEX NameIndex ON orderTable (Name);
```

# Creating tables

```
CREATE TABLE [dbo].[DimStore]
```

```
(  
    [StoreKey]           int            NOT NULL  
    , [GeographyKey]     int            NOT NULL  
    , [StoreName]         nvarchar(100)  NOT NULL  
    , [StoreType]         nvarchar(15)   NULL  
    , [StoreDescription] nvarchar(300)  NOT NULL  
    , [Status]            nvarchar(20)   NOT NULL  
    , [OpenDate]          datetime       NOT NULL  
    , [CloseDate]         datetime       NULL  
    , [ETLLoadID]         int            NULL  
    , [LoadDate]          datetime       NULL  
    , [UpdateDate]        datetime       NULL  
)
```

```
WITH
```

```
( CLUSTERED INDEX([StoreKey])  
    , DISTRIBUTION = ROUND_ROBIN  
)  
;
```

Row

```
CREATE TABLE [dbo].[FactOnlineSales]
```

```
(  
    [OnlineSalesKey]      int            NOT NULL  
    , [DateKey]            datetime       NOT NULL  
    , [StoreKey]            int            NOT NULL  
    , [ProductKey]          int            NOT NULL  
    , [PromotionKey]        int            NOT NULL  
    , [CurrencyKey]         int            NOT NULL  
    , [CustomerKey]         int            NOT NULL  
    , [SalesOrderNumber]    nvarchar(20)  NOT NULL  
    , [SalesOrderLineNumber] int            NULL  
    , [SalesQuantity]        int            NOT NULL  
    , [SalesAmount]          money          NOT NULL  
)
```

```
WITH
```

```
( CLUSTERED COLUMNSTORE INDEX  
    , DISTRIBUTION = HASH([ProductKey])  
)  
;
```

Column

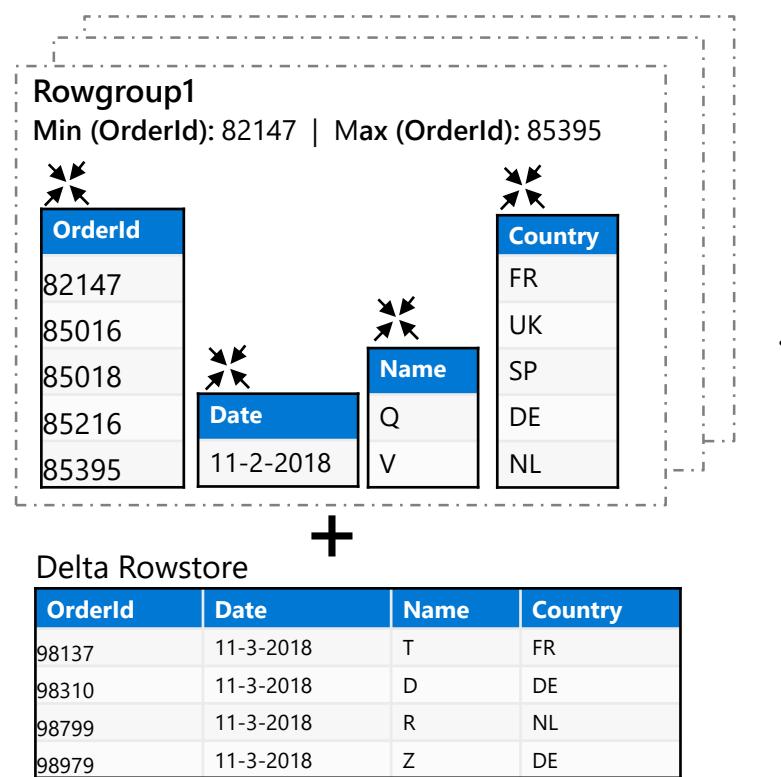
Distribution

# Synapse SQL (provisioned) Columnstore Tables

## Logical table structure

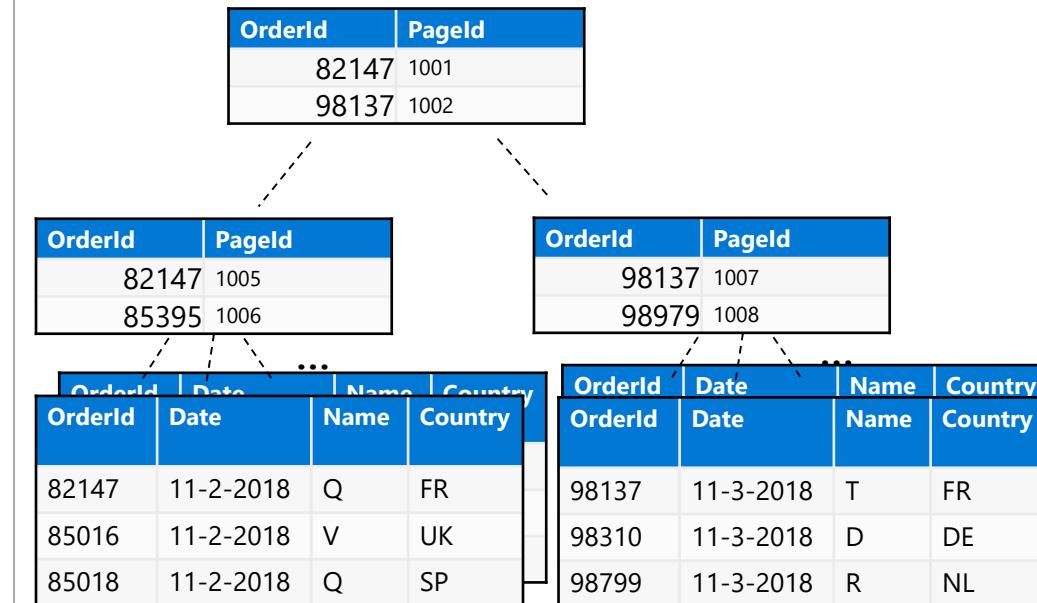
OrderId	Date	Name	Country
85016	11-2-2018	V	UK
85018	11-2-2018	Q	SP
85216	11-2-2018	Q	DE
85395	11-2-2018	V	NL
82147	11-2-2018	Q	FR
86881	11-2-2018	D	UK
93080	11-3-2018	R	UK
94156	11-3-2018	S	FR
96250	11-3-2018	Q	NL
98799	11-3-2018	R	NL
98015	11-3-2018	T	UK
98310	11-3-2018	D	DE
98979	11-3-2018	Z	DE
98137	11-3-2018	T	FR
...	...	...	...

## Clustered columnstore index (OrderId)



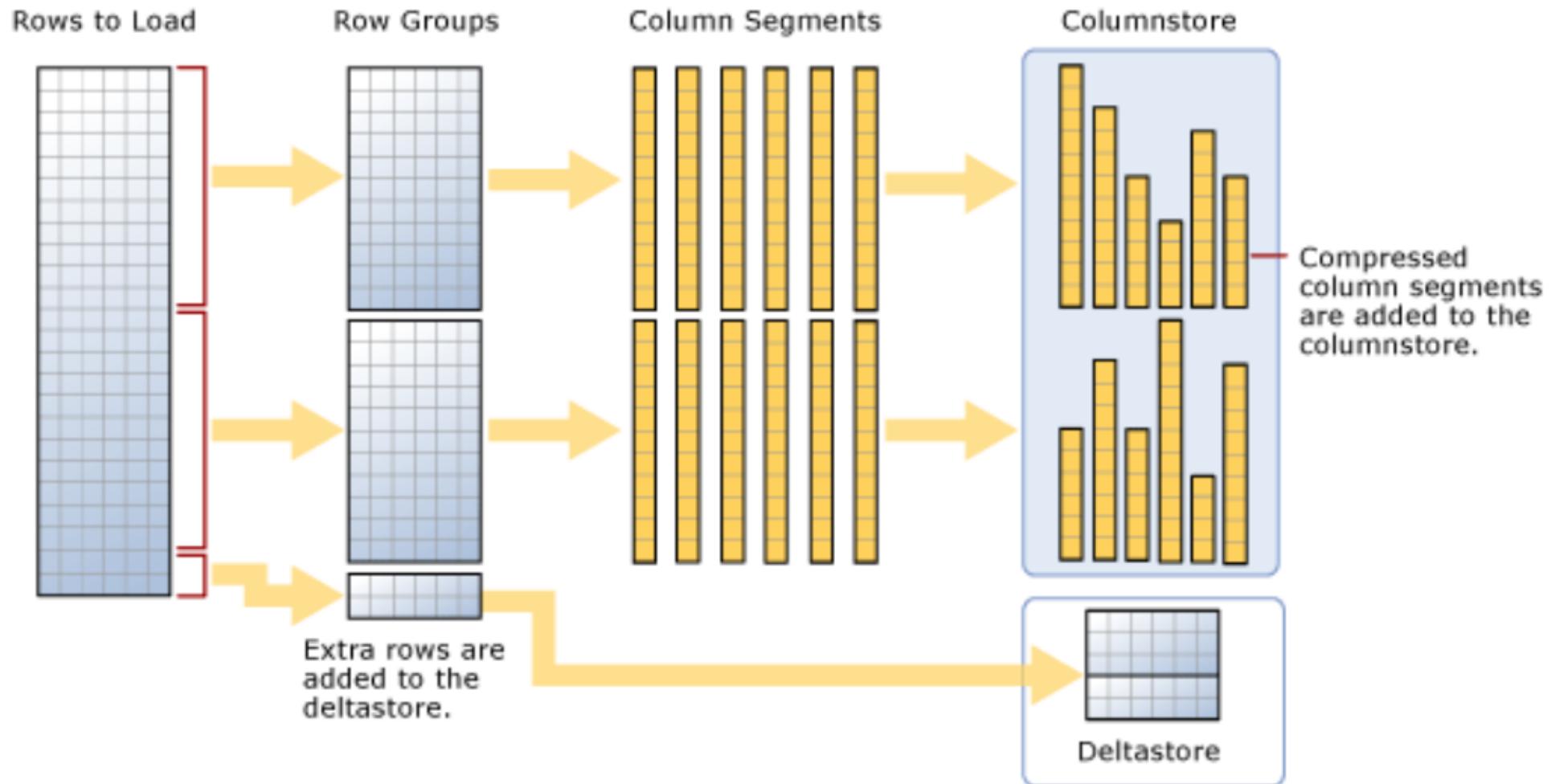
- Data stored in compressed columnstore segments after being sliced into groups of rows (rowgroups/micro-partitions) for maximum compression
- Rows are stored in the delta rowstore until the number of rows is large enough to be compressed into a columnstore

## Clustered/Non-clustered rowstore index (OrderId)



- Data is stored in a B-tree index structure for performant lookup queries for particular rows.
- Clustered rowstore index: The leaf nodes in the structure store the data values in a row (as pictured above)
- Non-clustered (secondary) rowstore index: The leaf nodes store pointers to the data values, not the values themselves

# Columnstore Storage Model



# Ordered Clustered Columnstore Indexes

## Overview

Ordered Clustered Columnstore Index를 사용하면 테이블 조회 시 필요한 세그먼트만 읽음으로써 쿼리 시간을 크게 단축할 수 있습니다.

### -- Create Table with Ordered Columnstore Index

```
CREATE TABLE sortedOrderTable
```

```
(  
    OrderId INT NOT NULL,  
    Date DATE NOT NULL,  
    Name VARCHAR(2),  
    Country VARCHAR(2)  
)
```

```
WITH
```

```
(  
    CLUSTERED COLUMNSTORE INDEX ORDER (OrderId)
```

### -- Create Clustered Columnstore Index on existing table

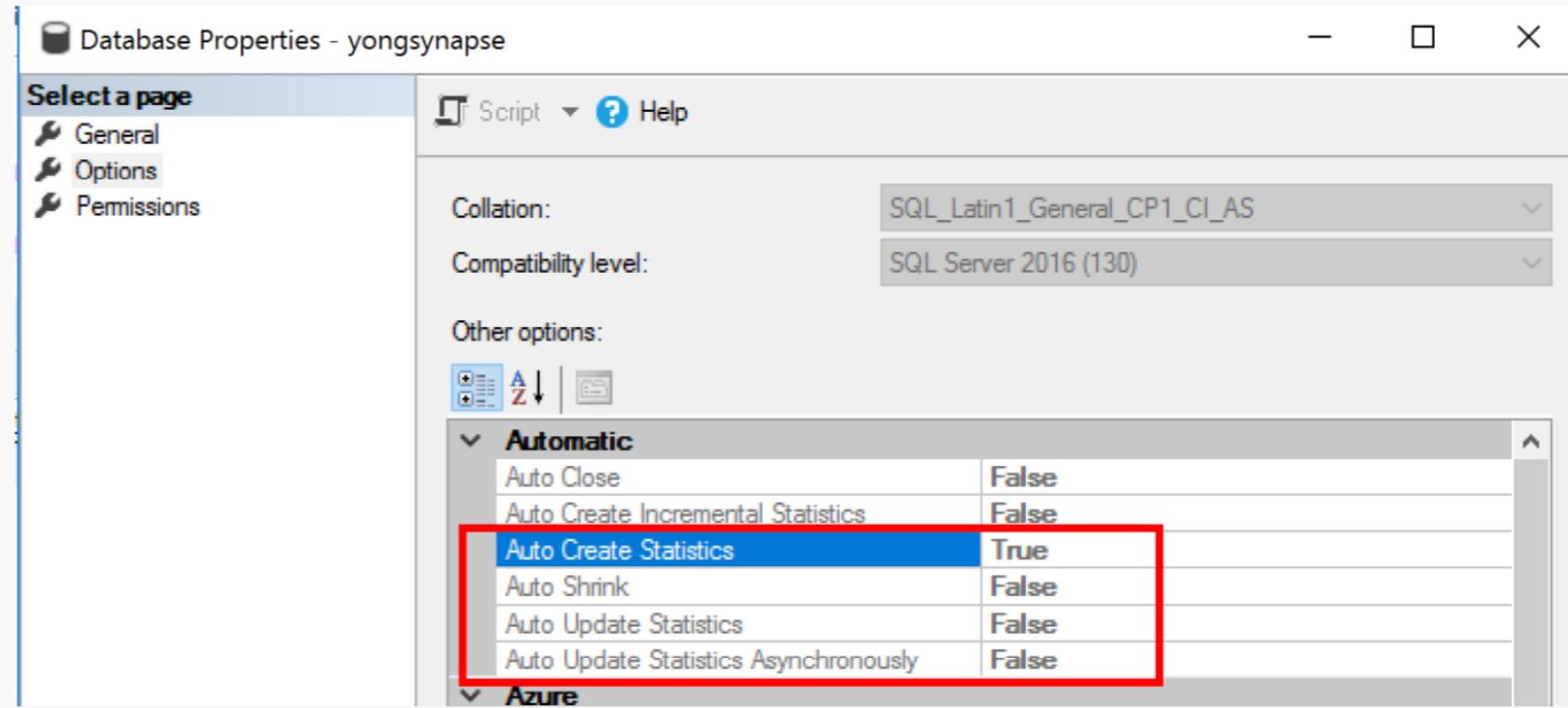
```
CREATE CLUSTERED COLUMNSTORE INDEX cciOrderId  
ON dbo.OrderTable ORDER (OrderId)
```

### -- Insert data into table with ordered columnstore index

```
INSERT INTO sortedOrderTable
```

```
VALUES (1, '01-01-2019','Dave', 'UK')
```

# Statistics



The screenshot shows the 'Database Properties - yongsynapse' window in SQL Server Management Studio. The 'Select a page' pane on the left has 'General' selected. The main pane shows database settings: Collation is set to 'SQL\_Latin1\_General\_CI\_AS', and Compatibility level is set to 'SQL Server 2016 (130)'. Under 'Other options', the 'Automatic' group is expanded, showing the following configuration:

Auto Close	False
Auto Create Incremental Statistics	False
Auto Create Statistics	True
Auto Shrink	False
Auto Update Statistics	False
Auto Update Statistics Asynchronously	False

Below the 'Automatic' group, the 'Azure' group is partially visible. The 'Auto Create Statistics' row is highlighted with a red box.

```
CREATE STATISTICS [statistics_name] ON [schema_name].[table_name]([column_name]) WITH FULLSCAN;
CREATE STATISTICS col1_stats ON dbo.table1 (col1) WITH SAMPLE = 50 PERCENT;
UPDATE STATISTICS [dbo].[table1] ([stats_col1]);
```

'Optimized for Elasticity'  
Sizing & Storage tiers

# Sizing factors

Database capacity

Tempdb

Concurrency & Memory

Load

Transaction size

Memory management



DWU

# Sizing Node Count / CPU & Memory

## Node Count

Performance level	Compute nodes
DW100c	1
DW200c	1
DW300c	1
DW400c	1
DW500c	1
DW1000c	2
DW1500c	3
DW2000c	4
DW2500c	5
DW3000c	6
DW5000c	10
DW6000c	12
DW7500c	15
DW10000c	20
DW15000c	30
DW30000c	60

## CPU (Hyperthreaded vCore)

Convert to DTU = DWU \* 9

$$\begin{aligned} \text{vCore} &= \text{DTU} / 100 \\ &= \text{DWU} * 9 / 100 \\ &= (6000 * 9) / 100 \\ &= 54000 / 100 \\ &= 540 \text{ vCore} \end{aligned}$$

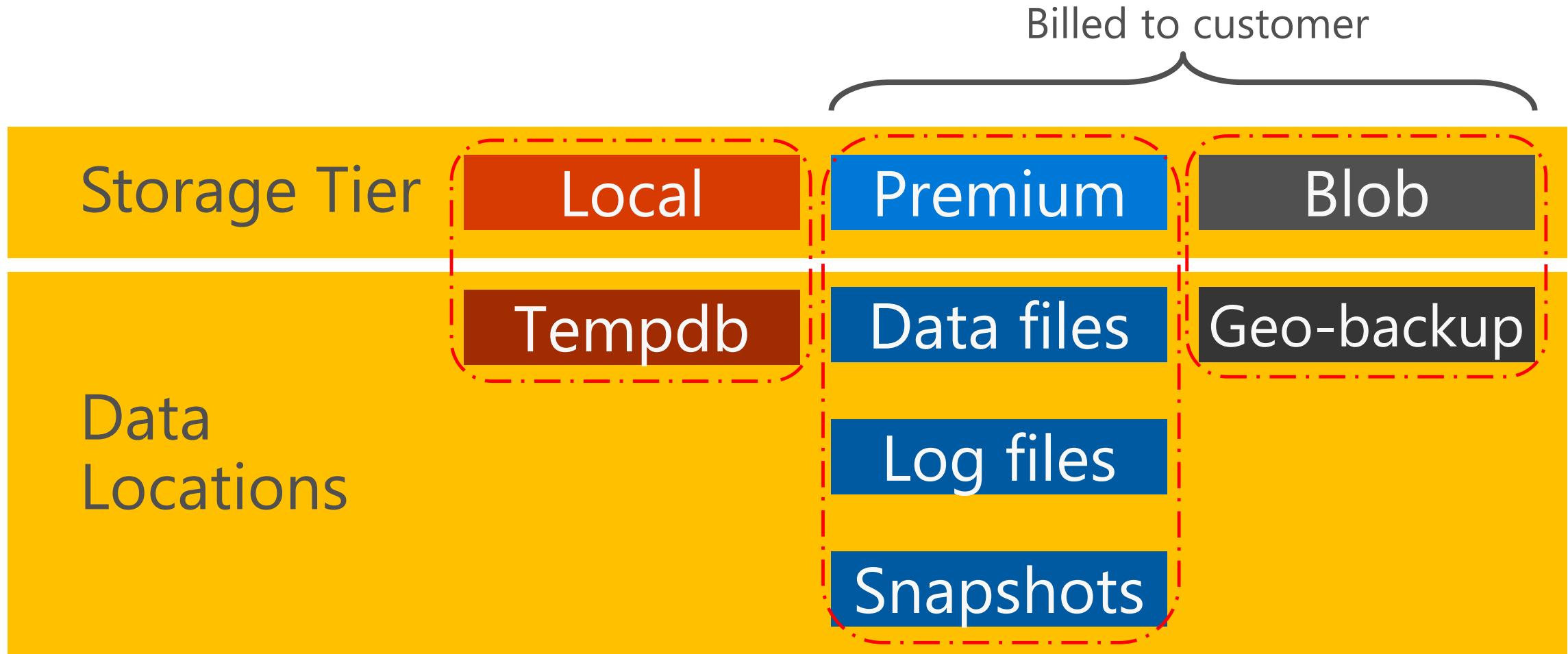
## Memory

DWU \* 60

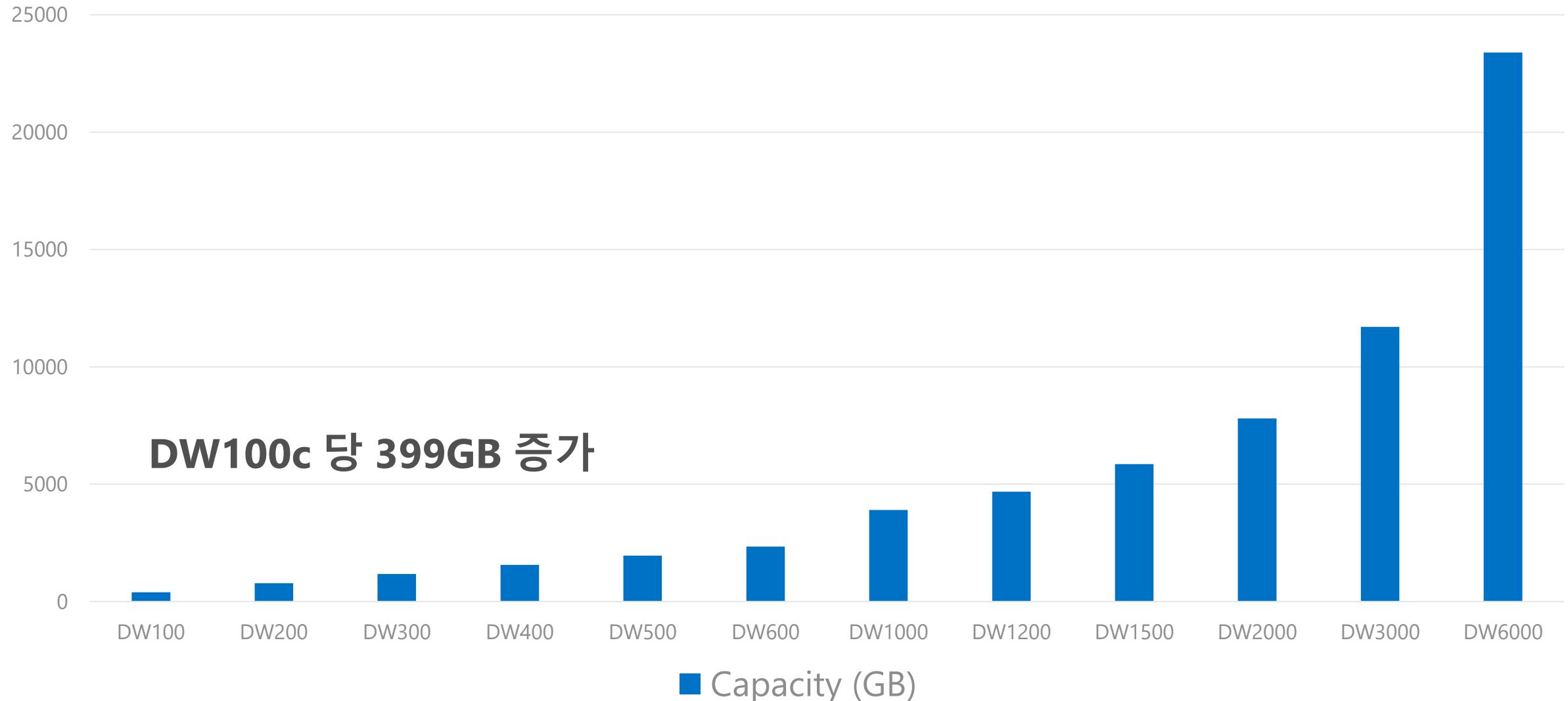
(DW6000c \* 60 = 3.6 Tb)

Performance level	Memory per data warehouse (GB)
DW100c	60
DW200c	120
DW300c	180
DW400c	240
DW500c	300
DW1000c	600
DW1500c	900
DW2000c	1,200
DW2500c	1,500
DW3000c	1,800
DW5000c	3,000
DW6000c	3,600
DW7500c	4,500
DW10000c	6,000
DW15000c	9,000
DW30000c	18,000

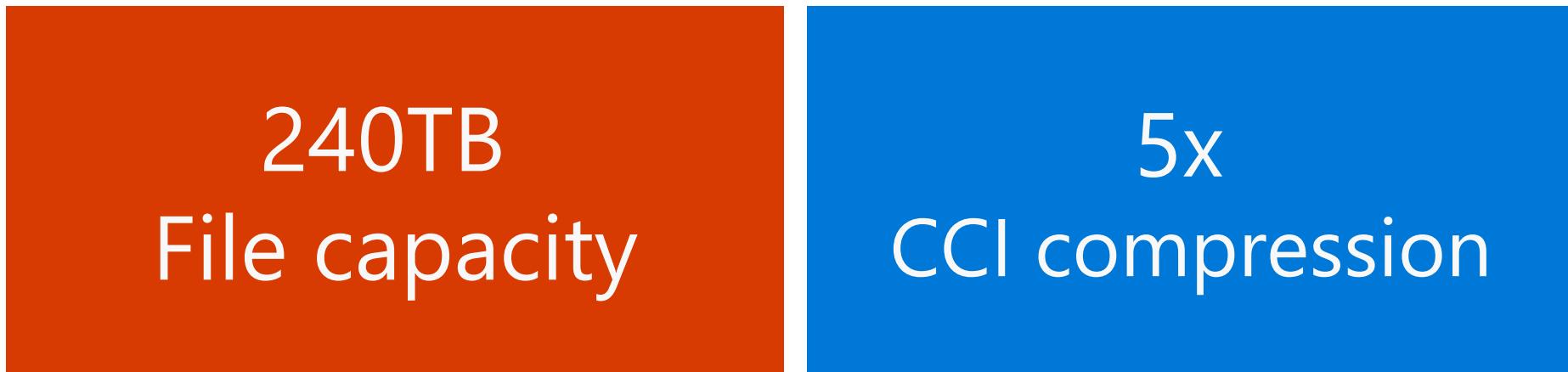
# Storage tiers / Data locations



# Local Storage: Tempdb sizing



# Premium Storage : DB FILE Capacity Limits



>1PB  
Db capacity

# Storage : Snapshots & backups

Premium Storage:  
S snapshots

Frequency

8  
hours

Retention

7  
days

RPO : 8 Hours

Blob Storage:  
Geo-redundant backups

Frequency and Retention

1  
Geo-backup

24hr  
RPO

Essentials ^



Resource group

jrijpartnerrg

Status

Online

Location

West US

Subscription name

ElasticScaleDev\_657854

Subscription ID

Server name

jrijpartner.database.windows.net

Connection strings

[Show database connection strings](#)

Performance tier

DataWarehouse (300 DWUs)

Geo-backup policy

Enabled

# Loading

# Sizing for the data load

## Delimited text 가이드

- 파일의 SIZE 가 클 경우, 데이터를 나누어 여러 개의 파일로 분할하여 사용합니다.
- 하나의 파일에 하나의 Reader를 매칭합니다.
- Delimited text는 가장 빠른 속도를 제공합니다.

압축된 텍스트 파일은 병렬 액세스에 제한이 발생할 수 있습니다.

파일의 데이터를 분할하거나 다른 파일 포맷을 사용합니다.

DWU	Readers	Writers
DW100	8	60
DW200	16	60
DW300	24	60
DW400	32	60
DW500	40	60
DW600	48	60
DW1000	60	60

# Loading Option

PolyBase	BCP	SSIS	ADF
<ul style="list-style-type: none"><li>• T-SQL 을 사용한 적재방식</li><li>• 적재 속도가 빠르고 가장 선호되는 적재 옵션</li><li>• CTAS 를 사용한 초기 데이터 적재</li><li>• 증분 데이터는 INSERT/INTO 를 사용함. OR CTAS를 사용하여 별도 테이블을 만들고 나중에 최종 테이블과 파티션 SWITCH 사용</li></ul>	<ul style="list-style-type: none"><li>• Bulk Copy Program UTILITY</li><li>• 명령어를 사용한 데이터 적재 방식</li><li>• 사이즈가 작은 파일에 사용 (&lt; 10 GB )</li><li>• DWU 수준을 올린다고 적재 성능이 높아지지 않음</li><li>• Thread 를 병렬로 지정하여 퍼포먼스를 향상시킬 수 있음</li></ul>	<ul style="list-style-type: none"><li>• MS 에서 제공하는 데이터 이관, 통합 기능이 있는 Tool 을 이용한 데이터 적재 방식</li><li>• 퍼포먼스 향상을 위해 병렬 Threads 를 증가시켜야 함</li></ul>	<ul style="list-style-type: none"><li>• 파이프라인을 이용한 데이터 적재 방식 (Polybase 방식)</li><li>• T-SQL 스크립트를 작성하지 않고 단순히 화면에서 설정을 세팅하므로 작업 방식이 매우 단순</li></ul>

# Polybase Sample

```
CREATE MASTER KEY;

CREATE DATABASE SCOPED CREDENTIAL AzureStorageCredential
WITH
    IDENTITY = 'user',
    SECRET =
'dIzoQupXULTk/Dy09gXWbCfP2hD6TJ8H9Lf0yp0T+ubQ3uy//PB9tNxbt04WN47dT0strIHbw83zAkB1bm+Xiw=='
;

CREATE EXTERNAL DATA SOURCE AzureStorage
WITH (
    TYPE = HADOOP,
    LOCATION = 'wasbs://datacontainer@yongstorage.blob.core.windows.net',
    CREDENTIAL = AzureStorageCredential
);
CREATE EXTERNAL FILE FORMAT TextFile
WITH (
    FORMAT_TYPE = DelimitedText,
    FORMAT_OPTIONS (FIELD_TERMINATOR = ',')
);
```

# Polybase Sample

```
CREATE EXTERNAL TABLE dbo.DimDate2External (
    DateId INT NOT NULL,
    CalendarQuarter TINYINT NOT NULL,
    FiscalQuarter TINYINT NOT NULL
)
WITH (
    LOCATION='/datedimension/',
    DATA_SOURCE=AzureStorage,
    FILE_FORMAT=TextFile
);
```

```
CREATE TABLE dbo.DimDate2
WITH
(
    CLUSTERED COLUMNSTORE INDEX,
    DISTRIBUTION = ROUND_ROBIN
)
AS
SELECT * FROM [dbo].[DimDate2External];
```

# Polybase Sample

```
CREATE EXTERNAL TABLE dbo.DimDate2External (
    DateId INT NOT NULL,
    CalendarQuarter TINYINT NOT NULL,
    FiscalQuarter TINYINT NOT NULL
)
WITH (
    LOCATION='/datedimension/',
    DATA_SOURCE=AzureStorage,
    FILE_FORMAT=TextFile
);
```

```
CREATE TABLE dbo.DimDate2
WITH
(
    CLUSTERED COLUMNSTORE INDEX,
    DISTRIBUTION = ROUND_ROBIN
)
AS
SELECT * FROM [dbo].[DimDate2External];
```



# Azure Synapse Analytics

## Synapse serverless SQL pool

# Recommended usage scenarios

## Quick data exploration

- Azure storage 내부 파일의 스키마와 데이터 조회가 가능합니다.
- Parquet, CSV, JSON 파일 포맷을 지원합니다.
- BI Tool을 위해 Azure storage 에 액세스 할 수 있는 Direct connector를 지원합니다.

## Logical Data Warehouse

- Azure storage 내부 raw files을 virtual tables 와 views 로 생성이 가능합니다.
- SQL을 사용하는 Tool을 이용하여 file 분석이 가능합니다.
- enterprise-grade security model

## Easy data transformation

- CSV을 parquet 포맷으로 변경을 지원합니다.
- 컨테이너와 Storage 계정들을 오가며 데이터 이동이 가능합니다.
- 외부 스토리지에 대한 쿼리 결과를 저장 가능합니다.

# Easily explore files on storage

The screenshot illustrates the process of querying data stored in Azure Storage using a Serverless SQL pool.

**Left Panel (Storage Explorer):**

- Shows the Azure portal navigation bar: Microsoft Azure | Synapse Analytics > internalsandboxwe5.
- The left sidebar shows categories: Data, Storage accounts (1), Databases (3), and Datasets (5).
- The main area displays a list of datasets under "opendataset". One item, "New SQL script - Select TOP 100 rows", is highlighted with a red box.

**Right Panel (SQL Editor):**

- The title bar shows Microsoft Azure | Synapse Analytics > internalsandboxwe5.
- The top ribbon includes Publish all, Validate all, Refresh, Discard all, and a search bar.
- The main area contains a SQL script titled "SQL script 1" with the following code:

```

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

```

- The "Connect to" dropdown is set to "SQL on-demand", which is also highlighted with a red box.
- The "Results" tab is selected, showing a table with four rows of data:

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

- A message at the bottom right indicates: "00:00:31 Query executed successfully."

# Easily query files in various formats

## Overview

다양한 파일 포맷에 저장된 데이터에 접속하기 위해 OPENROWSET 함수를 사용합니다

## Benefits

- CSV, parquet, and JSON 파일 조회가 가능합니다.
- 모든 파일에 대해 통합된 T-SQL 사용이 가능합니다.
- 데이터의 변환과 조회 결과 데이터 분석을 위해 standard SQL을 사용합니다.
- 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/csv/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

OPENROWSET은 외부 파일의 컬럼과 컬럼의 데이터 타입을 자동으로 정의합니다.

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

## 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.

	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

사용자가 쿼리의 WITH 절에 컬럼명과 컬럼의 데이터 타입을 명시할 수 있습니다.

## 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

OPENROWSET 함수를 이용하여 사용자가 파일의 구분자를 명시할 수 있고, 원하는 컬럼만 선택적으로 명시하여 결과 값을 불러올 수 있습니다.

## 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-18 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

OPENROWSET 함수에서 WildCards를 사용하여 여러 폴더와 여러 파일을 동시에 질의하여 결과를 얻을 수 있습니다.

```
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/tax/year=*/month=1/*.parquet',
    FORMAT = 'PARQUET') AS nyc
 GROUP BY YEAR(pickup_datetime)
 ORDER BY YEAR(pickup_datetime)
```

## Benefits

Offers reading multiple files/folders through usage of wildcards

Offers reading specific file/folder

Supports use of multiple wildcards

	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

OPENROWSET 함수를 사용하여 여러 개의 파일로부터 원하는 데이터만을 조회할 수 있습니다.

## 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

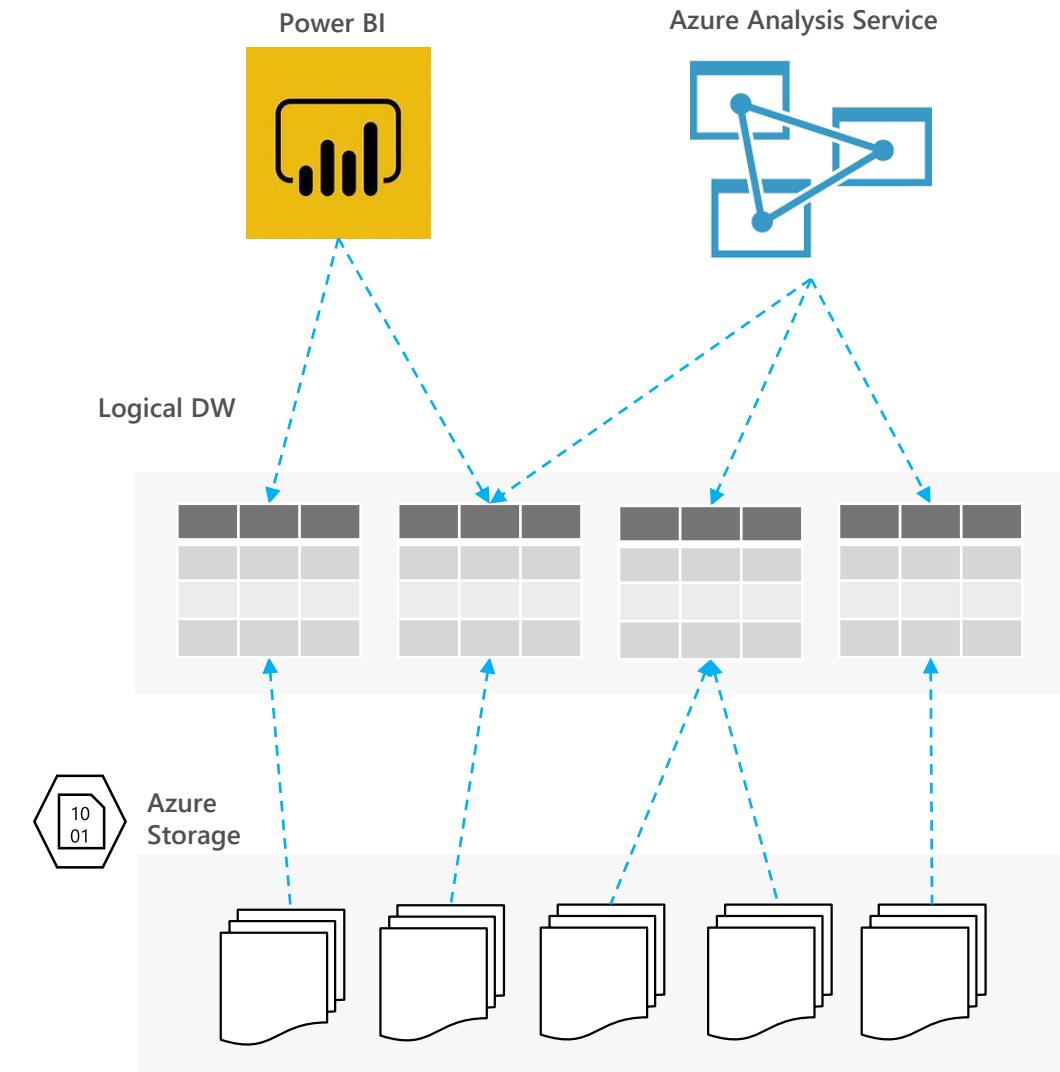
# Synapse serverless SQL pool as a logical data warehouse

## 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 pool은 논리적 Data Warehouse로서 Azure Storage에 있는 파일들을 사용하여 View 를 생성할 수 있습니다.

## 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

# Creating views

**Microsoft Azure | Synapse Analytics > internalsandbox...**

**Data**

**Develop**

```

CREATE VIEW yellow_2017 AS
Select *
FROM
OPENROWSET(
    BULK 'https://internalsandboxwe.dfs.core.windows.net/opendataset/nyctlc/yellow/puYear=2017/\*/\*',
    FORMAT='PARQUET'
) AS [r];

```

**Results**

```

-- type your sql script here, we now have intellisense
SELECT
YEAR(tpepPickupDateTime),
passengerCount,
COUNT(*) AS cnt
FROM
yellow_2017
GROUP BY
passengerCount,
YEAR(tpepPickupDateTime)
ORDER BY
YEAR(tpepPickupDateTime),
passengerCount

```

**Chart**

(NO COLUMN NAME)	PASSENGERCOUNT	CNT
2017	0	166086
2017	1	81034075
2017	2	16545571
2017	3	4748869
2017	4	2257813
2017	5	5407319

00:01:00 Query executed successfully.

**Microsoft Azure | Synapse Analytics > internalsandbox...**

**Develop**

```

opendataset SQL script 1 SQL script 2 SQL script 3
Run Publish Query plan Connect to SQL on-demand Use database DefSQLOnDemand
1 -- type your sql script here, we now have intellisense
2 SELECT
3     YEAR(tpepPickupDateTime),
4     passengerCount,
5     COUNT(*) AS cnt
6 FROM
7     yellow_2017
8 GROUP BY
9     passengerCount,
10    YEAR(tpepPickupDateTime)
11 ORDER BY
12    YEAR(tpepPickupDateTime),
13    passengerCount

```

**Results**

**Table**

(NO COLUMN NAME)	PASSENGERCOUNT	CNT
2017	0	166086
2017	1	81034075
2017	2	16545571
2017	3	4748869
2017	4	2257813
2017	5	5407319

00:01:00 Query executed successfully.

**Microsoft Azure | Synapse Analytics > internalsandbox...**

**Develop**

```

opendataset SQL script 1 SQL script 2 SQL script 3
Run Publish Query plan Connect to SQL on-demand Use database DefSQLOnDemand
1 -- type your sql script here, we now have intellisense
2 SELECT
3     YEAR(tpepPickupDateTime),
4     passengerCount,
5     COUNT(*) AS cnt
6 FROM
7     yellow_2017
8 GROUP BY
9     passengerCount,
10    YEAR(tpepPickupDateTime)
11 ORDER BY
12    YEAR(tpepPickupDateTime),
13    passengerCount

```

**Results**

**Chart**

00:01:00 Query executed successfully.

Chart type: Line  
Category column: (none)  
Legend (series) columns: Column 0, passengerCount, cnt  
Legend position: center - bottom  
Legend (series) label:

# Logical Data Warehouse - tables

## Overview

Serverless SQL Pool은 논리적 Data Warehouse로서 Azure Storage에 있는 파일들을 사용하여 External Files을 생성할 수 있습니다.

## 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

SQL queries을 사용하여 Azure Storage Files의 데이터 포맷을 쉽게 변환할 수 있습니다.

Optimize data pipeline - achieve more using serverless SQL pool

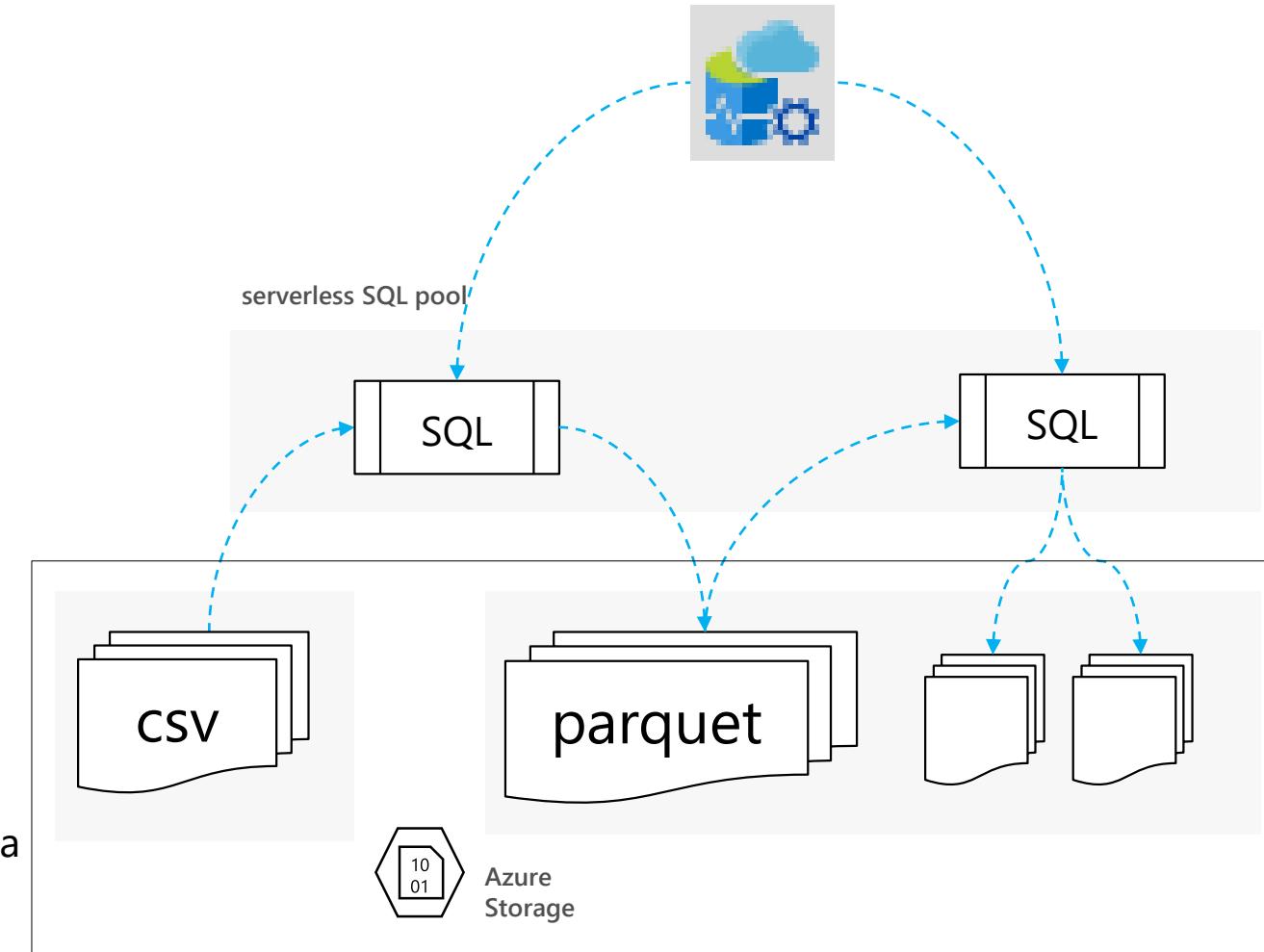
## Benefits

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)를 이용하여 쉬운 데이터 포맷 변환이 가능하며 Azure storage에 쿼리 결과를 저장할 수 있습니다.

## Benefits

Select any data set and store it in parquet format.

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

Use saved data using external table.

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

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

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

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

# UI based data transformation

Synapse live Validate all Publish all 1

SQL script 10 default

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

default > Parquet

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

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

Create external table

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

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

Select SQL pool \*  Built-in

Select a database \*  SQLServerlessDB

External table name \*  adls.retailsales1

Create external table \*  Automatically  Using SQL script

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

Create  Cancel [join meetup](#)

```

1  SELECT TOP 100 * FROM adls.retailsale
2  GO
3
4

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

Spark Pool에서 테이블을 생성하면 Serverless SQL Pool에서 해당 테이블을 참조할 수 있는 External Table이 자동 생성된다.

## 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 references the same external files.

The screenshot shows the Azure Synapse Analytics studio interface. On the left, there's a sidebar with icons for Connections, Servers, Databases, Tables, Columns, Keys, Constraints, and another Tables icon. The 'Tables' icon has a blue circle with the number '1' on it, indicating new or pending changes. The main area shows a 'Create external table' dialog at the top, containing the following SQL code:

```

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

```

Below this is a 'Cell 1' section with a play button and the following SQL code:

```

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

```

To the right, there's a 'SQLQuery\_1 - sq1kon...oud!SA' query editor window. It shows a SELECT statement:

```

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

```

At the bottom, there are 'Results' and 'Messages' tabs. The 'Results' tab displays the following data:

	ExtractId	DayOfWeekID	DayOfWeekDescr	DayOfWeekDescrShort	ExtractDateT
1	6b86b273ff34fce19d6b804eff5a...	1	Sunday	Sun	2020-01-22 0
2	d4735e3a265e16eee03f5a718h9b...	2	Monday	Mon	2020-01-22 0
3	4e07408562bedb8b60c0531uct...	3	Tuesday	Tue	2020-01-22 0
4	4b22777d4dd1fc61c6f884f4864...	4	Wednesday	Wed	2020-01-22 0
5	ef2d127de37b942baad06145e54b...	5	Thursday	Thu	2020-01-22 0
6	e7f6c011776e8db7cd330b54174f...	6	Friday	Fri	2020-01-22 0
7	70000000-0000-0000-0000-000000000000	7	Saturday	Sat	2020-01-22 0



# Azure Synapse Analytics

## Apache Spark



# Azure Synapse Apache Spark - Summary

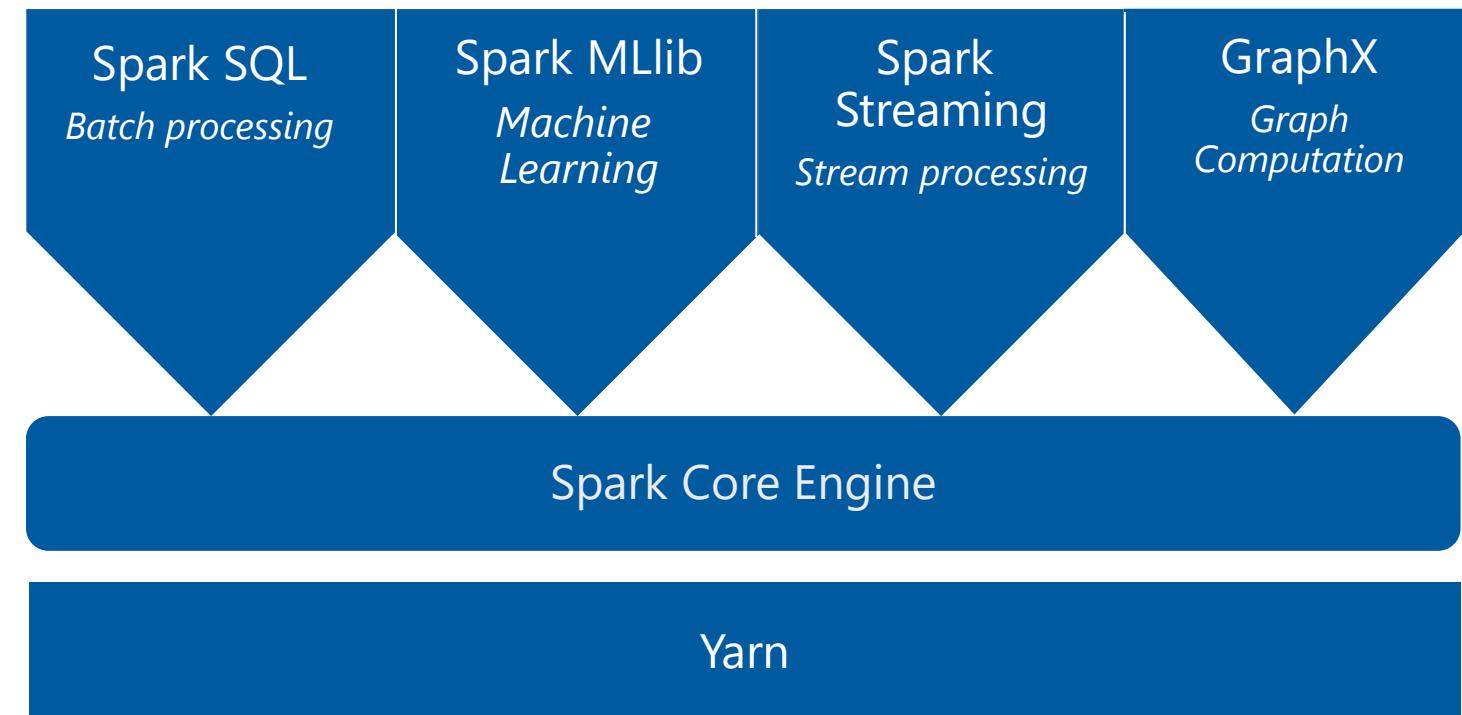
- Apache Spark 2.4 derivation
  - Linux Foundation Delta Lake 0.6 지원
  - .Net Core 3.0 지원
  - Python 3.6 + Anacondas 지원
- 다른 Azure Synapse services 와의 연결성
  - 통합된 보안과 Sign on
  - 통합된 Metadata
  - 통합 및 간소화된 provisioning
  - interact 기반 노트북이 포함된 통합된 UX
  - Synapse SQL (provisioned) pools에 대한 Fast load 제공
- Core scenarios
  - Data Prep/Data Engineering/ETL
  - Spark ML과 Azure ML 통합을 통한 Machine Learning
  - Library management를 통한 확장성
- 효율적인 자원 활용
  - 빠른 시작
  - Auto scale (up and down)
  - Auto pause
  - 최소 3개 노드 필요
- 다양한 언어 지원
  - .Net (C#), PySpark, Scala, Spark SQL, Java

# Apache Spark

빅데이터 분석을 위한 **unified open source**인, 병렬 데이터 처리 프레임워크

## Spark Unifies:

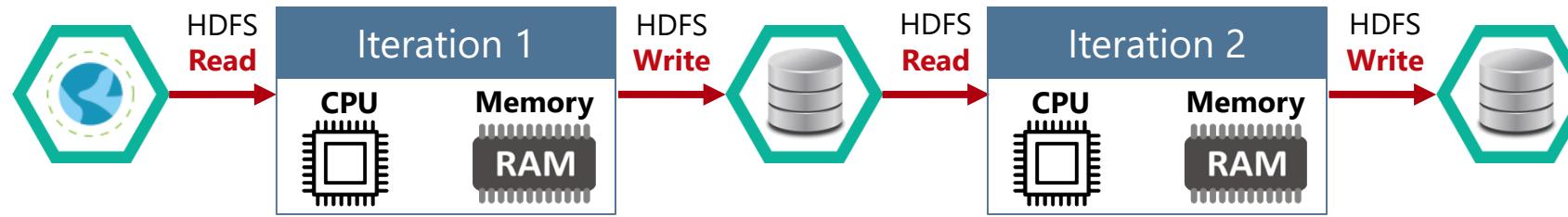
- 배치 처리
- SQL 상호작용
- 실시간 처리
- Machine Learning
- Deep Learning
- Graph 처리



<http://spark.apache.org>

# Motivation for Apache Spark

기존 방법: 복잡하거나, 온라인 event-hub 처리 등과 같은 MapReduce job에는 많은 disk I/O가 발생됨.

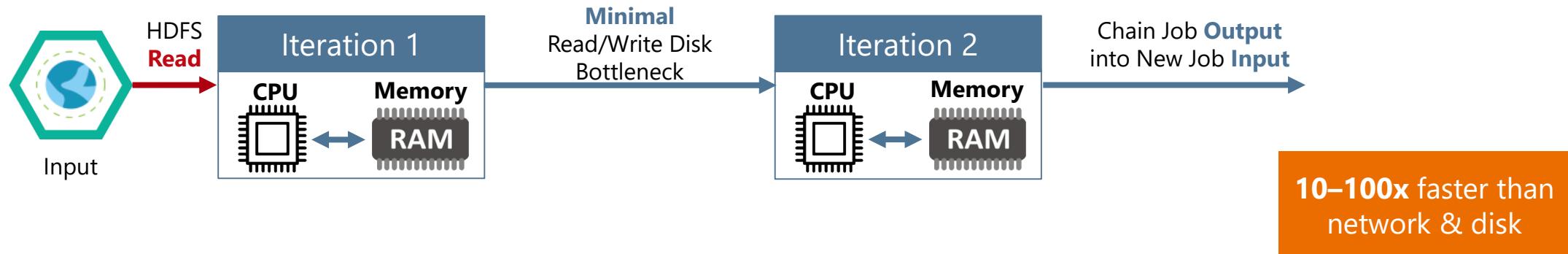


# Motivation for Apache Spark

기존 문제점: 복잡하거나, 온라인 event-hub 처리 등과 같은 MapReduce job에는 많은 disk I/O가 발생됨.

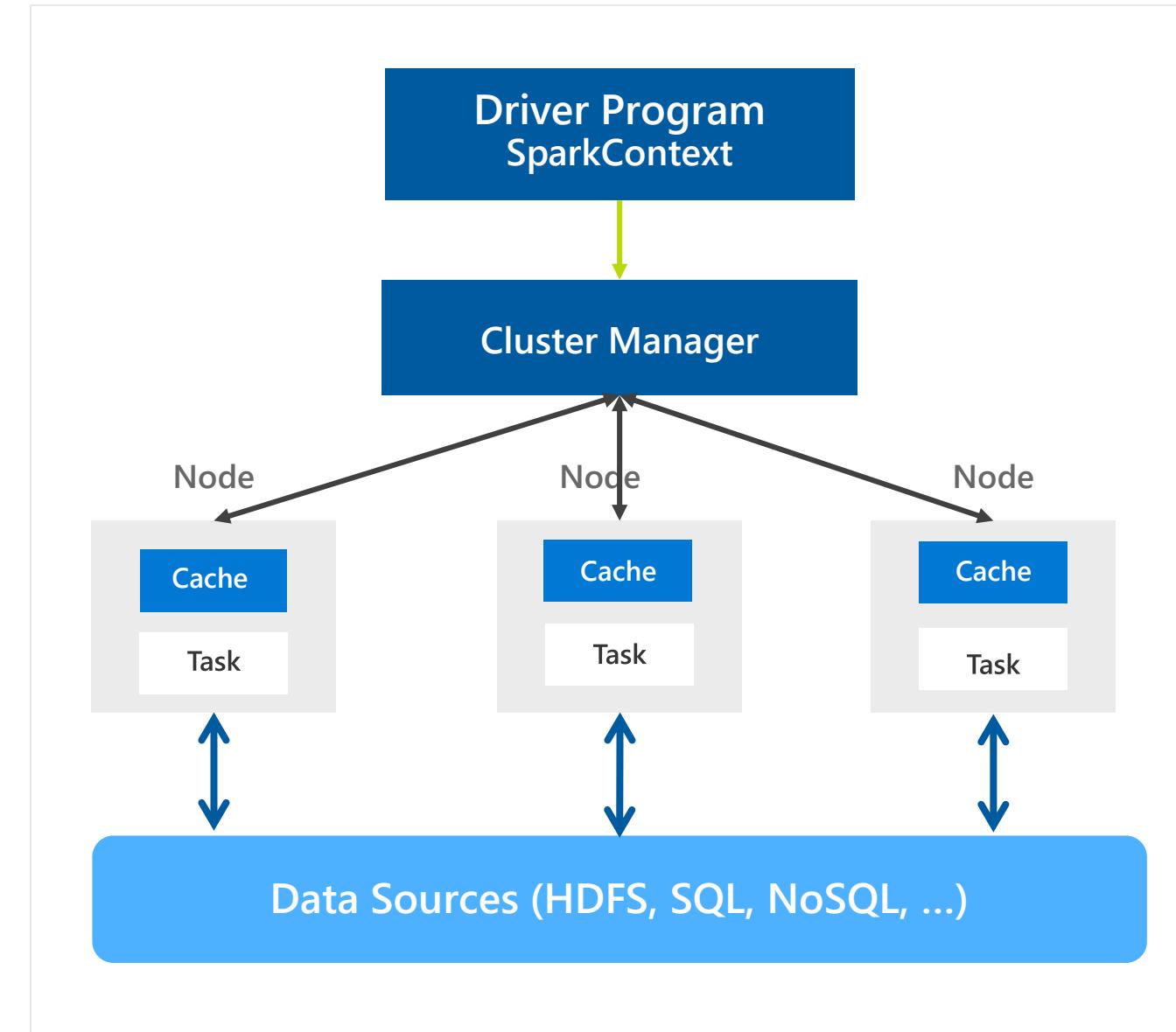


해결 방법: 새로운 분산 처리 엔진을 이용하여, data를 in-memory 형식으로 사용.



# General Spark Cluster Architecture

- 'Driver'는 사용자의 main() 함수를 구동하고, 다양한 병렬 작업들을 worker node에서 구동하게 합니다.
- 작업들의 결과는 'Driver'에 의해 취합되어 집니다.
- Worker node들은 HDFS, SQL과 같은 Data Source들로부터 데이터를 읽고 쓰게 됩니다.
- Worker node는 변환된 데이터를 메모리에 RDD로 cache합니다.
- Worker node들과 Driver Node는 public cloud의 VM들과 같이 실행됩니다.

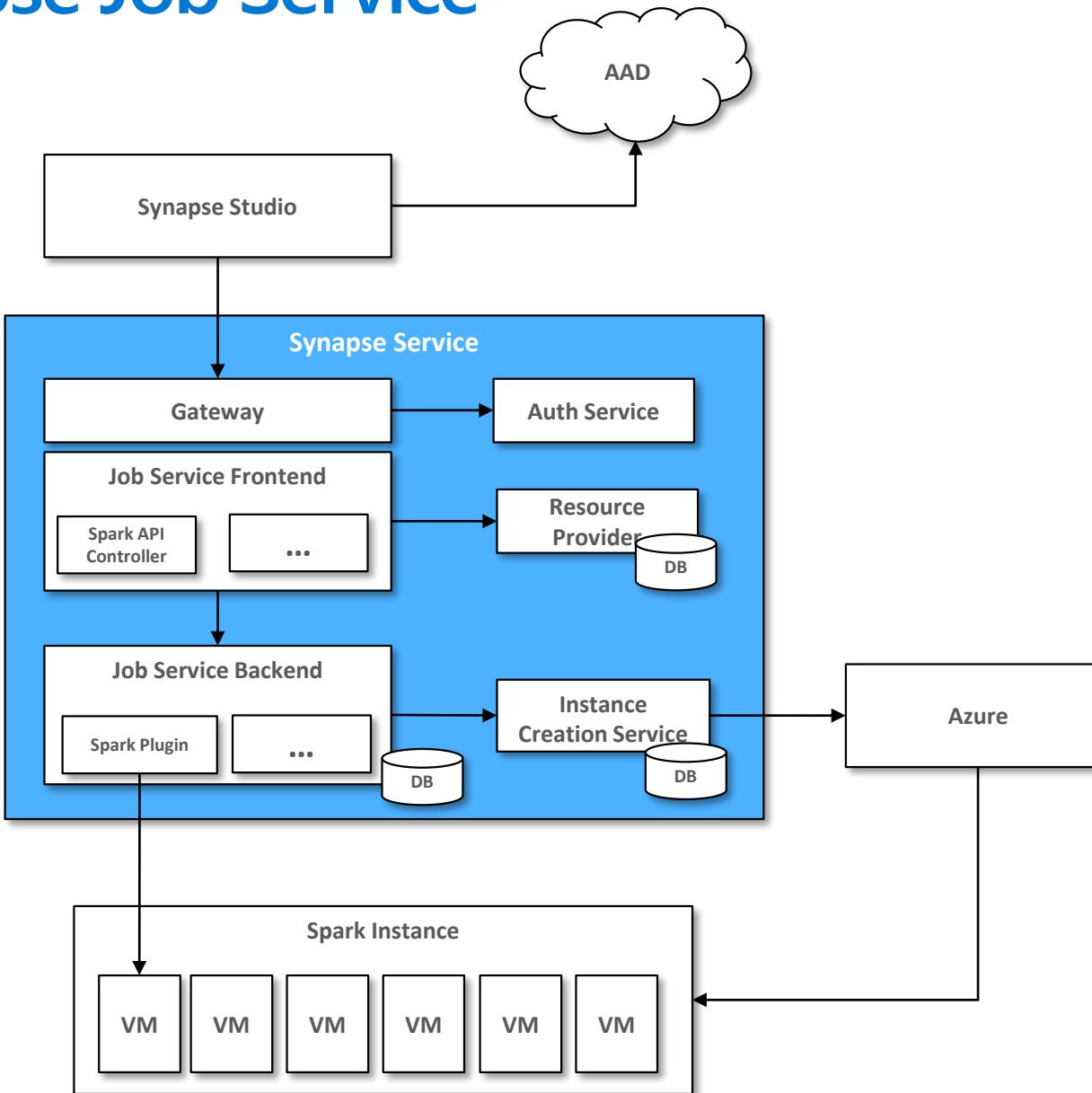




# Azure Synapse Apache Spark

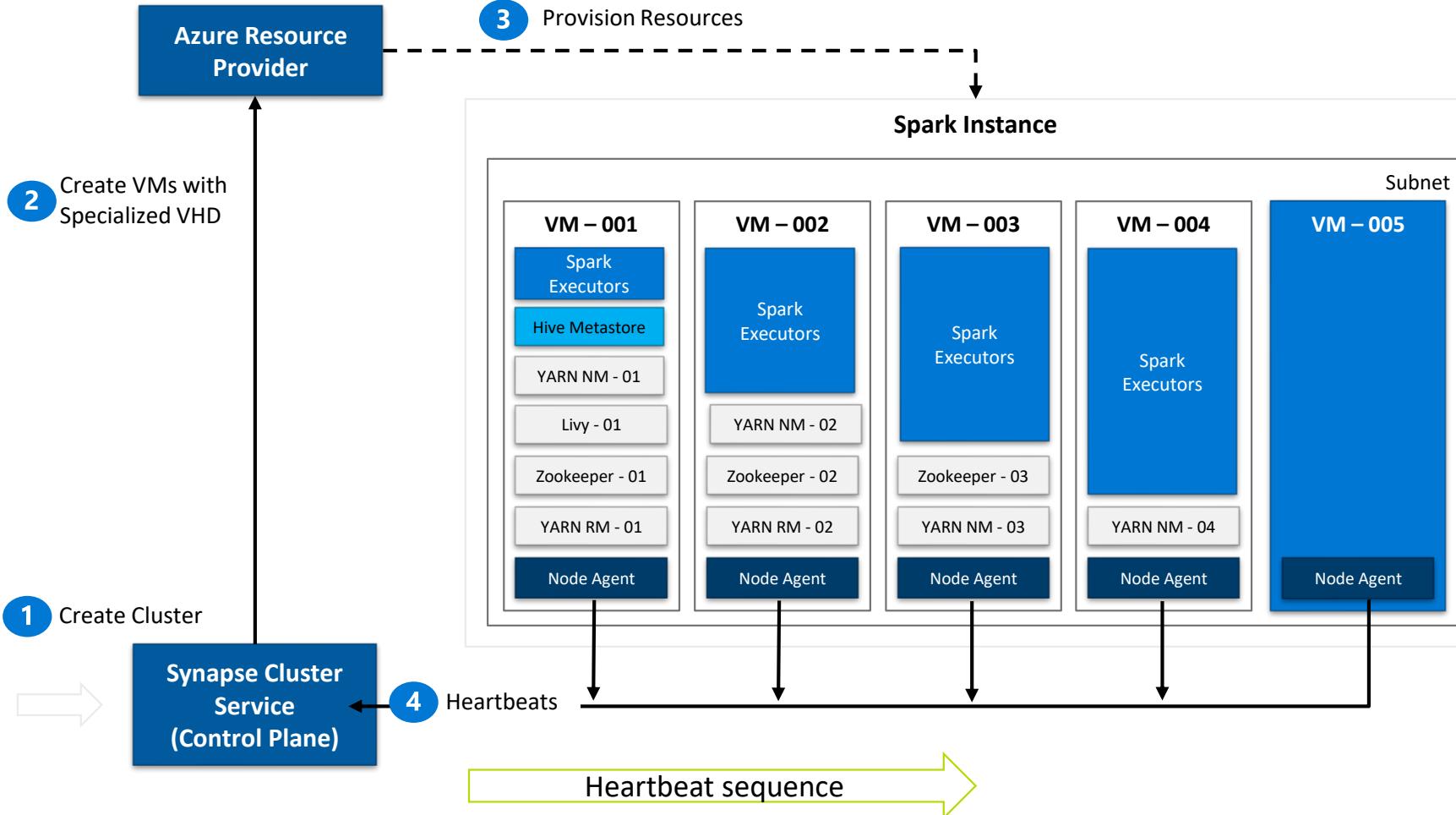
## Architecture Overview

# Synapse Job Service



1. Synapse Workspace와 spark pool을 생성
2. Notebook을 spark pool에 연결하고 spark 코드를 입력
3. Notebook client는 AAD로 부터 사용자 토큰을 받게 되며, Synapse Gateway에 Spark session 생성 요청을 보냄
4. 이후, Synapse Gateway는 workspace와 spark pool에 대한 인증과 권한을 확인한 후, Job Service frontend에서 호스팅하는 Spark (Livy) controller로 해당 요청을 포워딩 함
5. Job Service frontend는 Job Service backend로, 2개의 job을 생성하는 요청 하나는 cluster 생성 요청이고, 다른 하나는 spark session 생성 요청
6. Job Service backend는 Workspace와 Spark pool의 자세한 정보를 얻기 위해, Synapse Resource Provider와 연락하게 되고, Synapse Instance Service에게 cluster 생성을 위임
7. Instance가 만들어지게 되면, Job Service backend는 spark session 생성 요청을, cluster 안에 있는 Livy endpoint로 요청하게 됨
8. Spark session이 만들어지게 되면, Notebook client는 Spark statement들을 Job Service frontend로 보내게 됨
9. Job Service frontend는 backend로부터 만들어진 해당 cluster를 생성한 특정 사용자를 위해 만들어진 실제 Livy endpoint를 가지고 있으며, statement 실행을 위해 직접 statement를 보냄

# Synapse Spark Instances



1. Synapse Job Service는 Spark pool에 설명에 따라, Cluster Service에게 cluster 생성 요청을 보냅니다.
2. Cluster Service는 Azure SDK를 사용하여 specialized VHD를 이용하여 VM을 만들도록 Azure에 요청합니다.
3. Specialized VHD는 cluster 타입(e.g. Spark)에 의해 필요한 모든 서비스들이 포함되어져 있습니다.
4. VM이 부팅 되면, Node Agent는 heartbeat을 Cluster Service로 보내고 node configuration을 받습니다.
5. 첫번째 heartbeat을 기반으로 하여, node들은 초기화되고, role들이 assign 됩니다.
6. extra 노드들은 첫번째 heartbeat에서 삭제됩니다.
7. Cluster Service가 cluster가 준비되었다고 판단될 때, livy endpoint를 Job Service에게 반환합니다.

# Synapse Spark Instances

## Scale settings

sparkpool

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

Node size family \*

Memory Optimized

Node size \*

Medium (8 vCores / 64 GB)

Autoscale \* ⓘ

Enabled  Disabled

Number of nodes \*

3 3

Executor size \* ⓘ  
Medium (8 vCores, 56GB memory)

Executors \* ⓘ  
 2

Driver size \* ⓘ  
Medium (8 vCores, 56GB memory)

Executor size \* ⓘ  
Small (4 vCores, 28GB memory)

Executors \* ⓘ  
 5

Driver size \* ⓘ  
Small (4 vCores, 28GB memory)

- 기존의 Spark와 같이, Executor의 크기를 지정할 수 있습니다.
- 노드의 최대 크기까지 Executor를 정할 수 있습니다.

# Synapse Spark Instances

sparkpool3  
Refresh at 10:20:29 AM

XXLarge (64 vCores / 400 GB) 3 - 3 nodes  
0.00% utilized

Available session sizes ⓘ

Small	41 executors	<a href="#">Use</a>
Medium	20 executors	<a href="#">Use</a>
Large	8 executors	<a href="#">Use</a>
XLarge	2 executors	<a href="#">Use</a>
XXLarge	2 executors	<a href="#">Use</a>

Executor size \* ⓘ

Executors \* ⓘ

Driver size \* ⓘ

sparkpool3  
Refresh at 10:20:29 AM

XXLarge (64 vCores / 400 GB) 3 - 3 nodes  
0.00% utilized

Available session sizes ⓘ

Small	41 executors	<a href="#">Use</a>
Medium	20 executors	<a href="#">Use</a>
Large	8 executors	<a href="#">Use</a>
XLarge	2 executors	<a href="#">Use</a>
XXLarge	2 executors	<a href="#">Use</a>

Executor size \* ⓘ

Executors \* ⓘ

Driver size \* ⓘ

# Creating a Spark pool (1 of 2)

Azure Portal을 이용한 Spark Pool 생성 할 때, 기본설정 되어진 부분을 통해 쉽게 생성이 가능합니다.

Basics 탭에서 입력해야 하는 부분은 pool 이름만 설정하면 다른 부분은 기본 설정값이 들어 있어 생성이 쉽습니다.

Home > Synapse workspaces > euang-synapse-nov-ws - Apache Spark pools > Create Apache Spark pool

## Create Apache Spark pool

**Basics \***   **Additional settings \***   **Tags**   **Summary**

Create a Synapse Analytics Apache Spark pool with your preferred configurations. Complete the Basics tab then go to Review + create to provision with smart defaults, or visit each tab to customize.

**Apache Spark pool details**

Name your Apache Spark pool and choose its initial settings.

Apache Spark pool name \*

Node size family

Node size \*

Autoscale \*  Enabled  Disabled

Number of nodes \*   40

사용자 입력이 필수인 부분

기본 설정 값이 입력되어져 있음

# Creating a Spark pool (2 of 2) - optional

Additional settings 탭을 통해, 세부적인 설정 또한 가능합니다.

Spark의 버전, auto-pause의 시간 설정 가능

Home > prlangadws2 > Create Apache Spark pool

Create Apache Spark pool

Basics \* Additional settings \* Tags Summary

Customize additional configuration parameters including autoscale and component versions.

Auto-pause

Enter required settings for this Apache Spark pool, including setting auto-pause and picking versions.

Auto-pause \* ⓘ Enabled Disabled

Number of minutes idle \* 15

Component versions

Select the Apache Spark version for your Apache Spark pool.

Apache Spark *	2.4
Python	3.6.1
Scala	2.11.12
Java	1.8.0_222
.NET Core	3.1
.NET for Apache Spark	0.10.0
Delta Lake	0.5.0

Packages

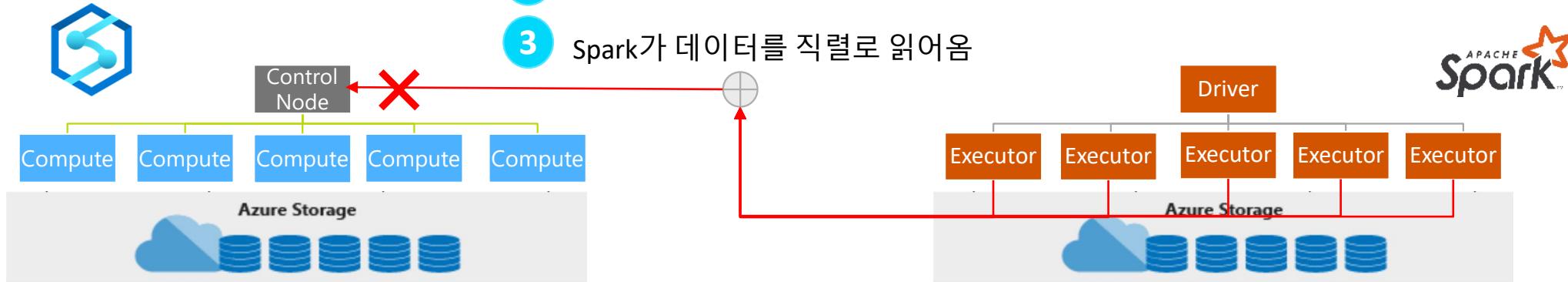
Upload environment configuration file ("PIP freeze" output).

File upload Select a file Upload

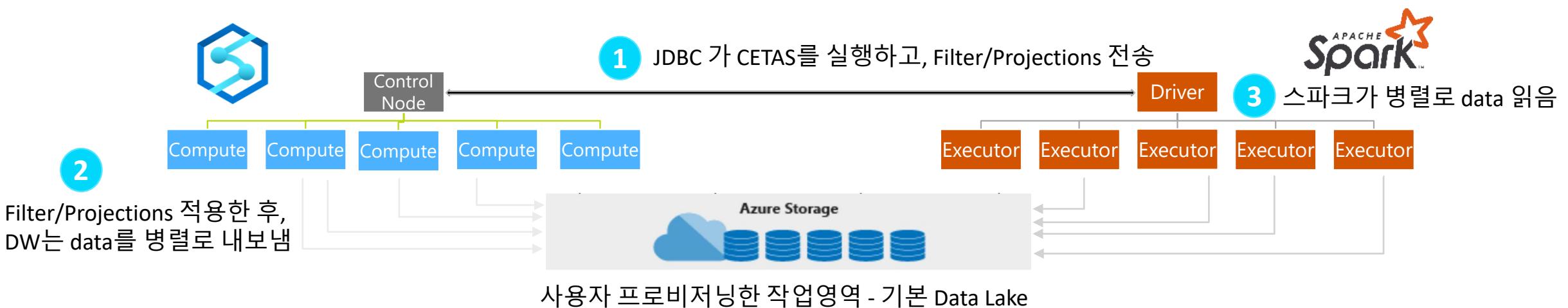
Review + create < Previous Next: Tags >

사용하는 package 이름과 버전  
업로드하여, 라이브러리 import 가능

## Existing Approach: JDBC



## New Approach: JDBC and Polybase



# Code-Behind Experience

## 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

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

# 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. In the main workspace, a storage account named 'nyctic' is selected, and a specific file path 'nyctic/green/puYear=2009/puMonth=1/part-00055' is highlighted. 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 where a new notebook has been created and is running a PySpark job. The job log shows the command used to read the data from the specified path.

**Left Pane (Storage Account View):**

- Microsoft Azure | Synapse Analytics > prlangadws2
- Data (selected)
- Storage accounts: prlangaddemosa (Primary) - filesystem, holidaydatacontainer, isdweatherdatacontainer, nyctic
- File path: nyctic/green/puYear=2009/puMonth=1/part-00055

**Right Pane (Synapse Analytics Workspace):**

- Job status: Job execution Succeeded (Spark 2 executors 8 cores)
- Job details:
 

ID	Description	Status	Stages	Tasks	Submission Time	Duration
Job 0	load at NativeMethodAccessorImpl.java:0	Succeeded	1/1	-	11/14/2019, 9:56:49 AM	7s
Job 1	showString at NativeMethodAccessorImpl.java:0	Succeeded	1/1	-	11/14/2019, 9:56:58 AM	1s
Job 2	showString at NativeMethodAccessorImpl.java:0	Succeeded	1/1	-	11/14/2019, 9:56:59 AM	11s
- Job logs (Cell 1):
 

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

Command executed in 3mins 59s 249ms by prlangad on 11-14-2019 09:57:11.863 -08:00
- Job execution summary:
 

ID	Description	Status	Stages	Tasks	Submission Time	Duration
Job 0	load at NativeMethodAccessorImpl.java:0	Succeeded	1/1	-	11/14/2019, 9:56:49 AM	7s
Job 1	showString at NativeMethodAccessorImpl.java:0	Succeeded	1/1	-	11/14/2019, 9:56:58 AM	1s
Job 2	showString at NativeMethodAccessorImpl.java:0	Succeeded	1/1	-	11/14/2019, 9:56:59 AM	11s
- Data preview (Job 0 output):
 

Vendor ID	Pickup Date Time	Proposed Date Time	Passenger Count	Trip Distance	Pickup Location ID	Dropoff Location ID	Start Lon	Start Lat	End Lon	End Lat
2	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	0.5	10.56	0.0		
1	2015-03-28 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	0.5	11.6	0.0		
2	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	0.5	20.54	0.0		
1	2015-03-28 19:21:05	2015-03-28 19:37:02	1	2.1	null	null	-73.98143005371094	40.7815055847168	-74.00081552734375	40.76177215576172

테이블  
포맷으로  
결과  
출력

Cell 1

```

1 # Azure storage access info
2 blob_account_name = "azureopendatastorage"
3 blob_container_name = "citydatacontainer"
4 blob_relative_path = "Safety/Release/city=Seattle"
5 blob_sas_token = r"""
6
7 # Allow SPARK to read from Blob remotely
8 wasbs_path = 'wasbs://%' % (blob_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 seassafety_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 NativeMethodAccessorImpl.java:0	In progress	0/1 (1 active)		11/22/2019, 12:44:46 AM	9m54s

View Table Chart

Cell 2

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

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

Cell 3

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

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

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

Cell 4

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

차트 포맷으로 결과 출력

SQL support

The screenshot shows the Azure Synapse Analytics workspace interface. The top navigation bar includes 'Publish all', 'Validate all', 'Refresh', 'Discard all', 'Data Download...', 'NYCTaxi\_Docs...', 'SeattleSafetyD...', 'Repro...', 'Cell', 'Run all', 'Publish', 'Attach to', 'euangnosyntaxcheck', 'Language: PySpark (Python)', and a search bar 'Search resources'.

**Cell 1:** Contains Python code for reading data from Azure Blob Storage using PySpark. A red box highlights the 'Language' dropdown.

```
[3]
1 # Azure storage access info
2 blob_account_name = "azureopendatastorage"
3 blob_container_name = "citydatacontainer"
4 blob_relative_path = "Safety/Release/city=Seattle"
5 blob_sas_token = r""
6
7 # Allow SPARK to read from Blob remotely
8 wasbs_path = 'wasbs://'+blob_account_name+'.'+blob_container_name+'.blob.core.windows.net/'+blob_relative_path
9 spark.conf.set('fs.azure.sas.%s.%s.blob.core.windows.net'.format(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)
```

**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

**Cell 2:** Contains Python code to create a temporary view named 'seattlesafety'.

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

**Cell 3:** Shows a pie chart output generated by the SQL command: `display(spark.sql('SELECT * FROM seattlesafety'))`. A red box highlights the SQL command, and a red arrow points to the chart.

**Chart Type:** pie chart  
**X axis column:** category  
**Y axis columns:** longitude  
**Aggregation:** COUNT  
**Y axis label:** Total  
**X axis label:** category

**Cell 4:** Contains Python code to write the DataFrame to a CSV file in ABFS.

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

Develop    +    Refresh    Discard all

Data Download...    NYCTaxi\_Docs... \*    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_title('Tip amount by Fare amount')
23 ax.set_xlabel('Fare Amount ($)')
24 ax.set_ylabel('Tip Amount ($)')
25 plt.axis([-2, 80, -2, 20])
26 plt.suptitle('')
27 plt.show()

```

상자 도표, 히스토그램 등과 같은 그래프들을 사용하여 데이터 탐색 가능

# Library Management - Python

## Overview

Spark pool 수준에서 python library 추가 가능

## Benefits

Input requirements.txt in simple pip freeze format

Add new libraries to your cluster

Update versions of existing libraries on your cluster

Ability to specify different requirements file for different pools  
within the same workspace

## Constraints

The library version must exist on PyPI repository

Version downgrade of an existing library not allowed

The screenshot shows the Azure Synapse Analytics Library Management interface. On the left, there's a sidebar with options like Analytics pools, SQL pools, Apache Spark pools (which is selected), External connections, Linked services, Orchestration, Triggers, Integration runtimes, Security, Access control, and Managed private endpoints. The main area is titled 'Apache Spark pools' and shows three items: 'priLangadSpark2', 'priLang-syntaxcheck', and 'priSpark'. To the right is a 'Properties' panel for 'priSpark'. It includes fields for Name (priSpark), URL (a long subscription-specific URL), Creation date (10/30/2019, 12:50:37 PM), and sections for Configuration and Workspace. At the bottom, there's a 'Packages' section with a button to 'Upload environment config file'. This 'Packages' section is highlighted with a red box.

# Library Management - Python

설치된 library 리스트 및 버전 정보 확인 가능

The screenshot shows the Azure Synapse Analytics interface with the 'Develop' workspace selected. In the left sidebar, under 'Notebooks', 'Notebook 4' is selected. The main area displays a notebook titled 'Notebook 4 \*'. A code cell in the notebook contains the following Python script:

```
1 import pprint
2 import pip
3 installed_packages = pip.get_installed_distributions()
4 installed_packages_list = sorted([ "%s==%s" % (i.key, i.version)
5         for i in installed_packages])
6 pprint.pprint(installed_packages_list)
```

The output of the cell shows a long list of installed Python packages and their versions:

```
['absl-py==0.8.1',
 'adal==1.2.2',
 'alabaster==0.7.10',
 'altair==3.2.0',
 'applicationinsights==0.11.9',
 'asn1crypto==1.0.1',
 'astor==0.8.0',
 'astroid==1.4.9',
 'astropy==1.3.2',
 'attrs==19.2.0',
 'azure-common==1.1.23',
 'azure-graphrbac==0.61.1',
 'azure-mgmt-authorization==0.60.0',
 'azure-mgmt-containerregistry==2.8.0',
 'azure-mgmt-keyvault==2.0.0',
 'azure-mgmt-resource==5.1.0',
 'azure-mgmt-storage==4.2.0',
 'azure-storage-blob==2.1.0',
 'azure-storage-common==2.1.0']
```

At the bottom of the notebook interface, there are session controls: 'Ready' (green), 'Stop session' (grey), 'Spark history server' (grey), and 'Configure session' (grey).

# Spark ML Algorithms

## Spark ML Algorithms

Classification and Regression	<ul style="list-style-type: none"><li>• Linear Models (SVMs, logistic regression, linear regression)</li><li>• Naïve Bayes</li><li>• Decision Trees</li><li>• Ensembles of trees (Random Forest, Gradient-Boosted Trees)</li><li>• Isotonic regression</li></ul>
Clustering	<ul style="list-style-type: none"><li>• k-means and streaming k-means</li><li>• Gaussian mixture</li><li>• Power iteration clustering (PIC)</li><li>• Latent Dirichlet allocation (LDA)</li></ul>
Collaborative Filtering	<ul style="list-style-type: none"><li>• Alternating least squares (ALS)</li></ul>
Dimensionality Reduction	<ul style="list-style-type: none"><li>• SVD</li><li>• PCA</li></ul>
Frequent Pattern Mining	<ul style="list-style-type: none"><li>• FP-growth</li><li>• Association rules</li></ul>
Basic Statistics	<ul style="list-style-type: none"><li>• Summary statistics</li><li>• Correlations</li><li>• Stratified sampling</li><li>• Hypothesis testing</li><li>• Random data generation</li></ul>

# Microsoft Spark Utilities

## Overview

ADLS Gen2와 Azure Blob Storage를 포함한 다양한 file system들과 작업 가능한 유ти리티 제공

## Benefits

It supports multiple methods for file systems such as List, Copy, Move, Write, Append, Delete file or directory, View file properties, Create new directory, Preview file content.

It supports environment utilities to get username, user id, job id, workspace name, pool name, cluster id.

It supports to get the access tokens of linked services and manage secrets in Azure Key Vault.

The screenshot shows a Jupyter Notebook interface with three code cells:

- Cell 1:** Displays code: `[2] 1 from notebookutils import mssparkutils  
2 mssparkutils.fs.help()`. It includes a note: "Command executed in 437ms by prlangad on 11-24-2020 18:32:02.018 -08:00". Below it, a description states: "mssparkutils.fs provides utilities for working with various FileSystems." and "Below is overview about the available methods:" followed by a list of methods like cp, mv, ls, mkdirs, putfile, headfile, append, and rm.
- Cell 2:** Displays code: `[3] 1 mssparkutils.credentials.help()`. It includes a note: "Command executed in 355ms by prlangad on 11-24-2020 18:32:47.633 -08:00". Below it, a list of methods for managing Azure Key Vault secrets.
- Cell 3:** Displays code: `[4] 1 mssparkutils.env.help()`. It includes a note: "Command executed in 472ms by prlangad on 11-24-2020 18:33:14.526 -08:00". Below it, a list of methods for getting environment variables like user name, user id, job id, workspace name, pool name, and cluster id.

# Hypespace

## Overview

Hypespace를 이용하여 Apache Spark 사용자가 인덱스를 생성할 수 있음

## Benefits

It helps accelerate your workloads or queries containing filters on predicates with high selectivity or a join that requires heavy shuffles.

Maintain the indexes through a multi-user concurrency model.

Leverage these indexes automatically, within your Spark workloads, without any changes to your application code for query/workload acceleration.

It supports index operations as create index, list index, restore index, delete index, vacuum index

Languages supported: Scala, Python, .NET

The screenshot shows a PySpark notebook interface with the following sections and code examples:

- Create indexes:**

```

Cell 4
[ ] 1 # Create indexes from configurations
2
3
4 hyperspace.createIndex(emp_DF, emp_IndexConfig)
5 hyperspace.createIndex(dept_DF, dept_IndexConfig1)
6 hyperspace.createIndex(dept DF, dept IndexConfig2)
    
```
- List indexes:**

```

Cell 6
[ ] 1 hyperspace.indexes().show()
    
```
- Index usage:**

```

Cell 8
[ ] 1 # Enable Hypespace
2 Hyperspace.enable(spark)
3
4 emp_DF = spark.read.parquet(emp_Location)
5 dept_DF = spark.read.parquet(dept_Location)
6
7 emp_DF.show(5)
8 dept_DF.show(5)
9
10 # Filter with equality predicate
11
12 eqFilter = dept_DF.filter("deptId = 20").select("deptName")
13 eqFilter.show()
14
15 hyperspace.explain(eqFilter, True, displayHTML)
    
```

# IntelliJ IDE



Apache Spark application과 spark pool에 submit 할 수 있는 IntelliJ 플러그인 지원

```

object DataAnalysis {
  def main(args: Array[String]): Unit = {
    val conf = new SparkConf().setAppName("DataAnalysis_v3")
    val sc = new SparkContext(conf)
    val spark:SparkSession = SparkSession.builder()
      .appName( name= "RunBatchJob")
      .config(conf)
      .getOrCreate()

    import spark.implicits._

    //Read file
    //val df = sc.textFile("abfss://rawdata@synapse.dfs.core.windows.net/employees.csv")
    val df: DataFrame = (spark.read
      .option("header", "true")
      .option("inferSchema", "true")
      .format( source = "csv")
      .load( path = "abfss://rawdata@synapsedemos.dfs.core.windows.net/employees.csv"))

    //Check number of records
    println("Number of records: " + df.count())

    //check schema
    df.printSchema()

    //display sample dataframe
    df.show( numRows = 5)

    //add new column YearsOfService based on hire
    val currentDate = DateTimeFormatter.ofPattern("yyyy-MM-dd")
    val derivedExpression = "currentDate - df.col(hireDate).cast("java.util.Date")"
    val df_yearsOfService = df.withColumn( colName = "YearsOfService", expr = derivedExpression)
    val df_yearsOfService = spark.sqlContext.sql(df_yearsOfService)
    df_yearsOfService.show( numRows = 5)
    //get employees who are married
    val df2 = df.filter( condition = $"MarriedID" === 1)

    // save married employees data in separate file
    df2.write.format( source = "parquet").mode( saveMode = "overwrite").save("abfss://rawdata@synapsedemos.dfs.core.windows.net/employees_married.parquet")
  }
}

```

# Synapse Notebook: Connect to AML workspace

The screenshot shows the Azure Synapse Notebook interface. The left sidebar shows 'Develop' mode with a list of resources: SQL scripts, Notebooks, Data flows, Spark job definitions, and Power BI. A red arrow points from the text '간단한 코드로 AML workspace와 연결 가능' to the code in Cell 5.

**간단한 코드로 AML workspace와 연결 가능**

**Check the Azure ML Core SDK Version to Validate Your Installation**

Cell 3

```
[5] 1 import azureml.core
2 print("SDK Version:", azureml.core.VERSION)
```

Command executed in 1s 258ms by balapv on 11-12-2019 14:41:52.805 -08:00

SDK Version: 1.0.69

**Connect to Azure Workspace**

Cell 5

```
[6] 1 ## Import the Workspace class and check the Azure ML SDK version.
2 from azureml.core import Workspace
3
4 ws = ws = Workspace(subscription_id = "6560575d-fa06-4e7d-95fb-f962e74efd7a",
5 | | | | | resource_group = "balapv-synapse-rg", workspace_name = "AML-WS-synapse")
6
7 print(ws.name, ws.location, ws.resource_group, sep='\t')
```

Command executed in 3s 909ms by balapv on 11-12-2019 14:41:55.491 -08:00

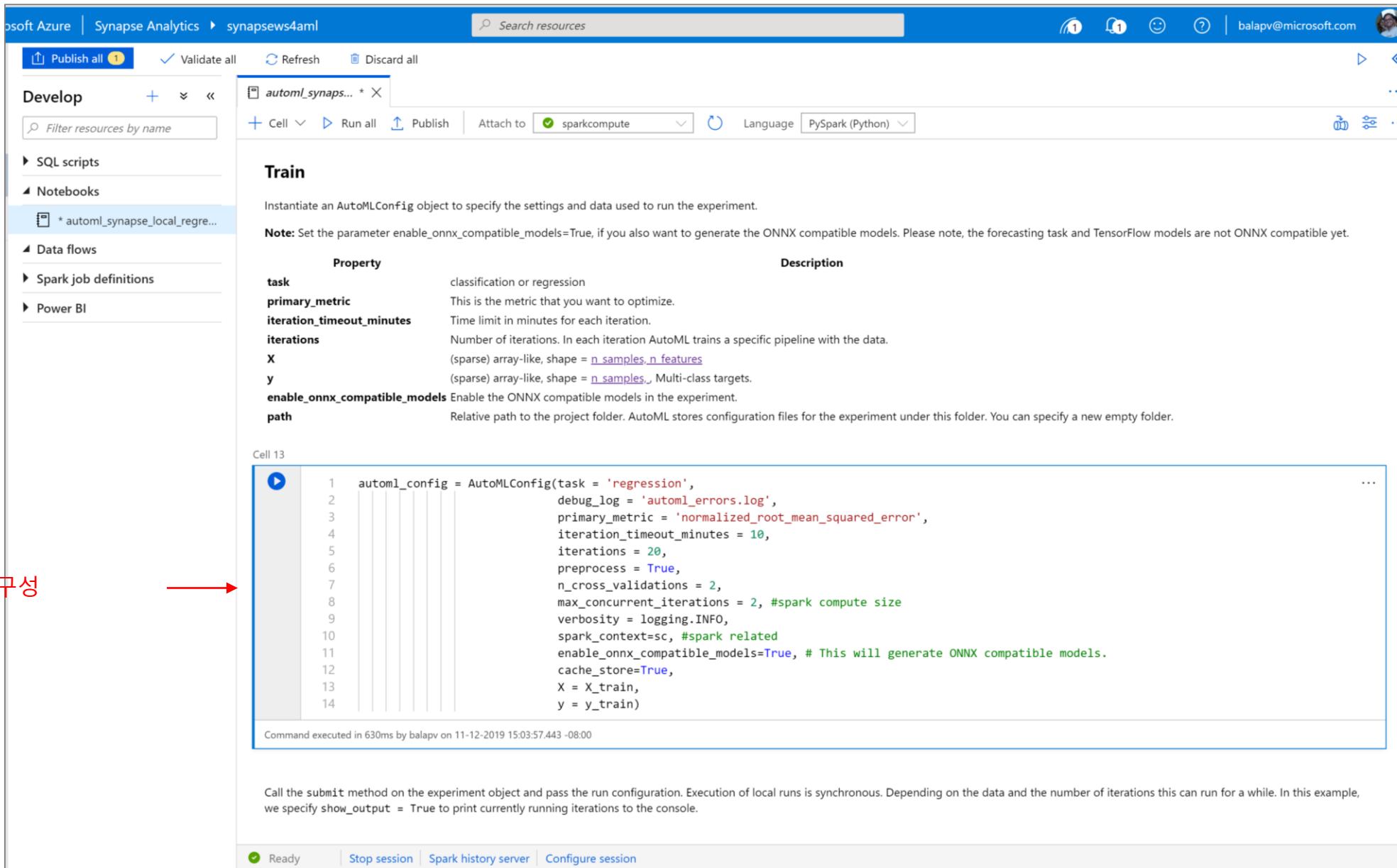
AML-WS-synapse westus2 balapv-synapse-rg

Cell 6

```
[7] 1 # import modules
2 import azureml.core
3 import pandas as pd
4 from azureml.core.authentication import ServicePrincipalAuthentication
5 from azureml.core.workspace import Workspace
6 from azureml.core.experiment import Experiment
```

Running | Stop session | Spark history server | Configure session

# Synapse Notebook: Configure AML job to run on Synapse



The screenshot shows the Azure Synapse Analytics notebook interface. On the left, the sidebar lists notebooks, data flows, and spark job definitions. The main area is titled "Train" and contains documentation for the AutoMLConfig object. It includes a note about enabling ONNX compatible models. Below the documentation is a code cell labeled "Cell 13" containing Python code for setting up an AutoMLConfig object. A red arrow points from the text "Parameter 구성" (Parameter Configuration) to the code cell.

**Parameter 구성** →

```

1 automl_config = AutoMLConfig(task = 'regression',
2                               debug_log = 'automl_errors.log',
3                               primary_metric = 'normalized_root_mean_squared_error',
4                               iteration_timeout_minutes = 10,
5                               iterations = 20,
6                               preprocess = True,
7                               n_cross_validations = 2,
8                               max_concurrent_iterations = 2, #spark compute size
9                               verbosity = logging.INFO,
10                              spark_context=sc, #spark related
11                              enable_onnx_compatible_models=True, # This will generate ONNX compatible models.
12                              cache_store=True,
13                              X = X_train,
14                              y = y_train)

```

Call the `submit` method on the experiment object and pass the run configuration. Execution of local runs is synchronous. Depending on the data and the number of iterations this can run for a while. In this example, we specify `show_output = True` to print currently running iterations to the console.

Ready | Stop session | Spark history server | Configure session

# Synapse Notebook: Run AML job

Run AutoML job

```
Cell 15
1 local_run = experiment.submit(automl_config, show_output = True)
```

Command executed in 12mins 34s 972ms by balapv on 11-12-2019 15:17:53.089 -08:00

Running an experiment on spark cluster: automl-local-regression-Synapse.  
Parent Run ID: AutoML\_ad8600ab-a1ab-4b6b-b233-059d969e0a0e

\*\*\*\*\*  
ITERATION: The iteration being evaluated.  
PIPELINE: A summary description of the pipeline being evaluated.  
DURATION: Time taken for the current iteration.  
METRIC: The result of computing score on the fitted pipeline.  
BEST: The best observed score thus far.  
\*\*\*\*\*

ITERATION	PIPELINE	DURATION	METRIC	BEST
1	StandardScalerWrapper ElasticNet	0:00:38	0.0021	0.0021
2	StandardScalerWrapper ElasticNet	0:00:32	0.0054	0.0021
0	StandardScalerWrapper ElasticNet	0:01:20	0.0004	0.0004
4	StandardScalerWrapper RandomForest	0:00:33	0.0179	0.0004
3	StandardScalerWrapper ElasticNet	0:00:36	0.0036	0.0004
5	StandardScalerWrapper LightGBM	0:00:28	0.0109	0.0004
6	MaxAbsScaler DecisionTree	0:00:34	0.0168	0.0004
7	MaxAbsScaler RandomForest	0:00:41	0.0104	0.0004
8	MaxAbsScaler DecisionTree	0:01:05	0.0077	0.0004
9	MaxAbsScaler DecisionTree	0:00:48	0.0086	0.0004
10	StandardScalerWrapper DecisionTree	0:00:39	0.0058	0.0004
11	MaxAbsScaler DecisionTree	0:00:45	0.0096	0.0004
13	MaxAbsScaler ExtremeRandomTrees	0:00:47	0.0147	0.0004
12	MaxAbsScaler ExtremeRandomTrees	0:01:54	0.0096	0.0004
14	StandardScalerWrapper ElasticNet	0:00:39	0.0027	0.0004
15	StandardScalerWrapper ElasticNet	0:00:54	0.0010	0.0004
16	StandardScalerWrapper ElasticNet	0:00:48	0.0023	0.0004
17	MaxAbsScaler ElasticNet	0:00:31	0.0239	0.0004
18	StandardScalerWrapper ElasticNet	0:00:53	0.0014	0.0004
19	VotingEnsemble	0:01:59	0.0004	0.0004

Get Azure Portal URL for Monitoring Runs

Running | Stop session | Spark history server | Configure session

ML 잡 결과

# End

