



# BIG DATA CONFERENCE EUROPE 2020



## Azure Synapse Analytics

*A data lakehouse*

James Serra

Data & AI Architect

Microsoft, NYC MTC

[jamesserra3@gmail.com](mailto:jamesserra3@gmail.com)

Blog: [JamesSerra.com](https://JamesSerra.com)

*Modified 11/20/20*

# About Me

- Microsoft, Big Data Evangelist
- In IT for 30 years, worked on many BI and DW projects
- Worked as desktop/web/database developer, DBA, BI and DW architect and developer, MDM architect, PDW/APS developer
- Been perm employee, contractor, consultant, business owner
- Presenter at PASS Business Analytics Conference, PASS Summit, Enterprise Data World conference
- Certifications: MCSE: Data Platform, Business Intelligence; MS: Architecting Microsoft Azure Solutions, Design and Implement Big Data Analytics Solutions, Design and Implement Cloud Data Platform Solutions
- Blog at [JamesSerra.com](http://JamesSerra.com)
- Former SQL Server MVP
- Author of book "Reporting with Microsoft SQL Server 2012"



# Agenda

---

- Introduction to Synapse
  - Massively Parallel Processing (MPP) Intro
  - Synapse Demo
- 
-

# Multiple analytics platforms

Big Data



Data  
Lake

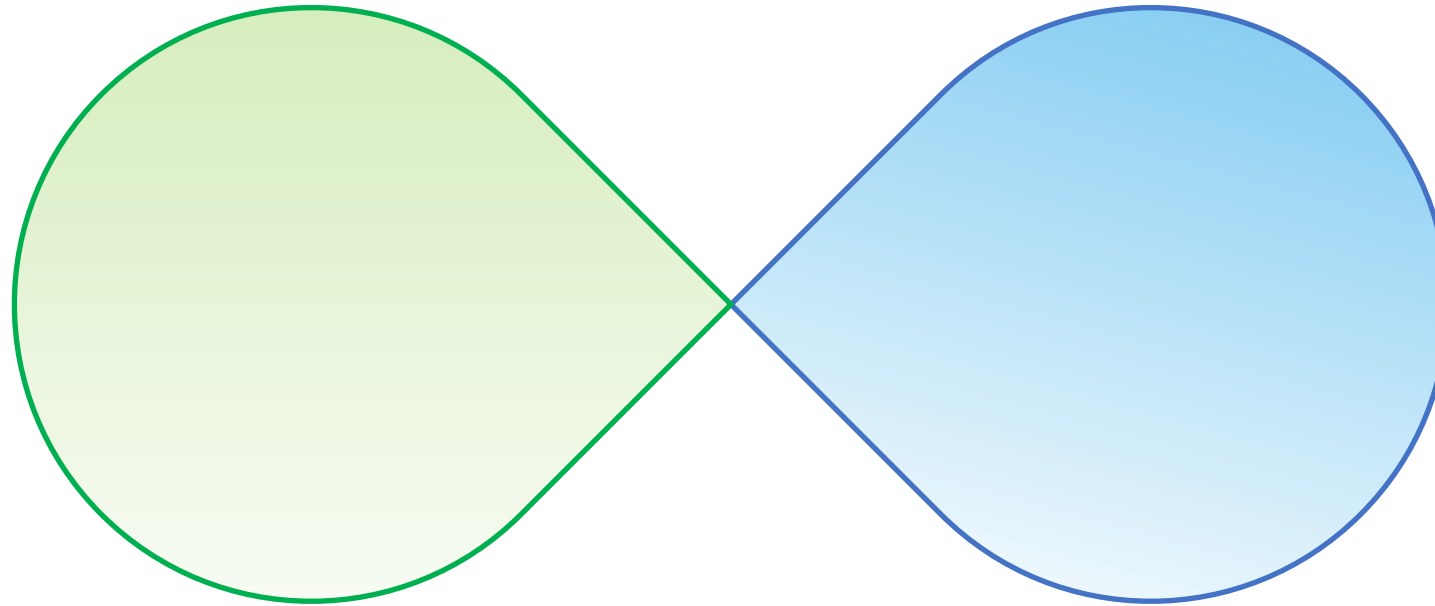
Relational Data



Data  
Warehouse

**OR**

**Azure brings** these two worlds together



**Welcome to Azure Synapse Analytics**

Data warehousing & big data analytics—*Data Lakehouse*

# Modern Data Warehouse



## On-premises data

Oracle, SQL, Teradata, fileshares, SAP



## Cloud data

Azure, AWS, GCP



## SaaS data

Salesforce, Dynamics

### INGEST



Azure Data Factory

### PREPARE



Azure Data Factory



Azure Databricks

### TRANSFORM & ENRICH



Azure Data Factory



Azure Databricks

### SERVE



Azure SQL Data Warehouse

### VISUALIZE



Power BI

### STORE

Azure Data Lake Storage

# Azure Synapse Analytics - *Data Lakehouse*



## On-premises data

Oracle, SQL, Teradata, fileshares, SAP



## Cloud data

Azure, AWS, GCP



## SaaS data

Salesforce, Dynamics



# Azure Synapse Analytics

## STORE

Azure Data Lake Storage

## VISUALIZE



Power BI



### Synapse Analytics (GA)



### Synapse Analytics (GA) (formerly SQL DW) "GA"



### Synapse Analytics (PREVIEW) "PP"

Add new capabilities to the GA service

- New GA features**
- Resultset caching
  - Materialized Views
  - Ordered columnstore
  - JSON support
  - Dynamic Data Masking
  - SSDT support
  - Read committed snapshot isolation
  - Private LINK support
  - Workload Isolation
  - Updatable hash key
- Public preview features**
- Simple ingestion with COPY
  - Share DW data with Azure Data Share
  - Native Prediction/Scoring
- Private preview features**
- Streaming ingestion & analytics in DW
  - Fast query over Parquet files
  - FROM clause with joins
  - SQL MERGE support
  - Column encryption
  - Multi-column hash distribution

- Public preview features**
- Synapse Studio
  - Collaborative *workspaces*
  - Distributed T-SQL Query service
  - SQL Script editor
  - Unified security model
  - Notebooks
  - Apache Spark
  - On-demand T-SQL
  - Code-free data flows
  - Orchestration Pipelines
  - Data movement
  - Integrated Power BI
  - Bulk load wizard
  - DeltaLake support
  - CDM support
  - External table wizard



SQL ANALYTICS



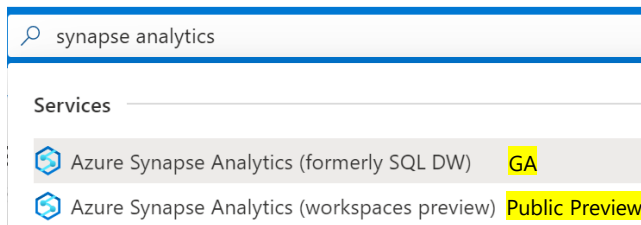
APACHE SPARK



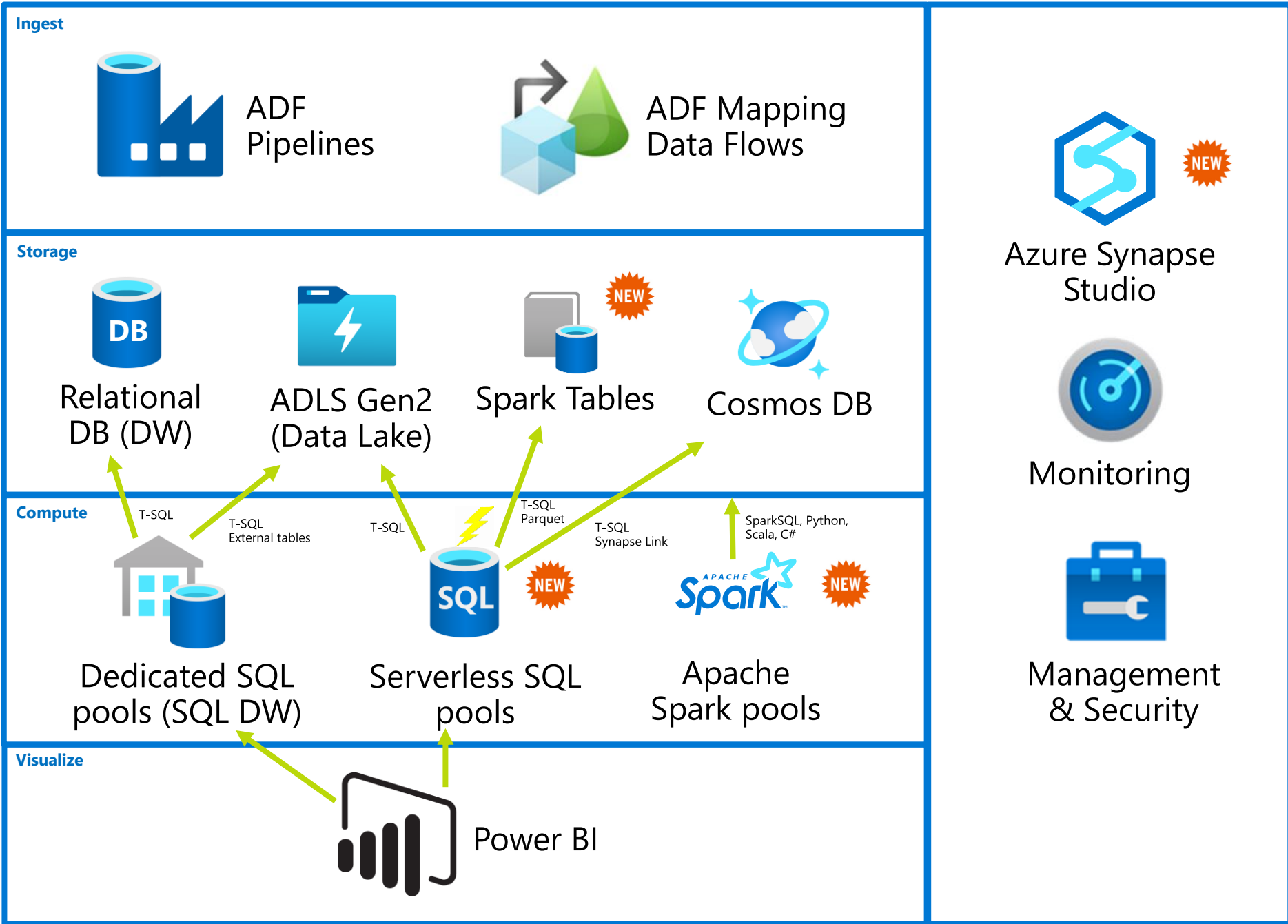
STUDIO



DATA INTEGRATION



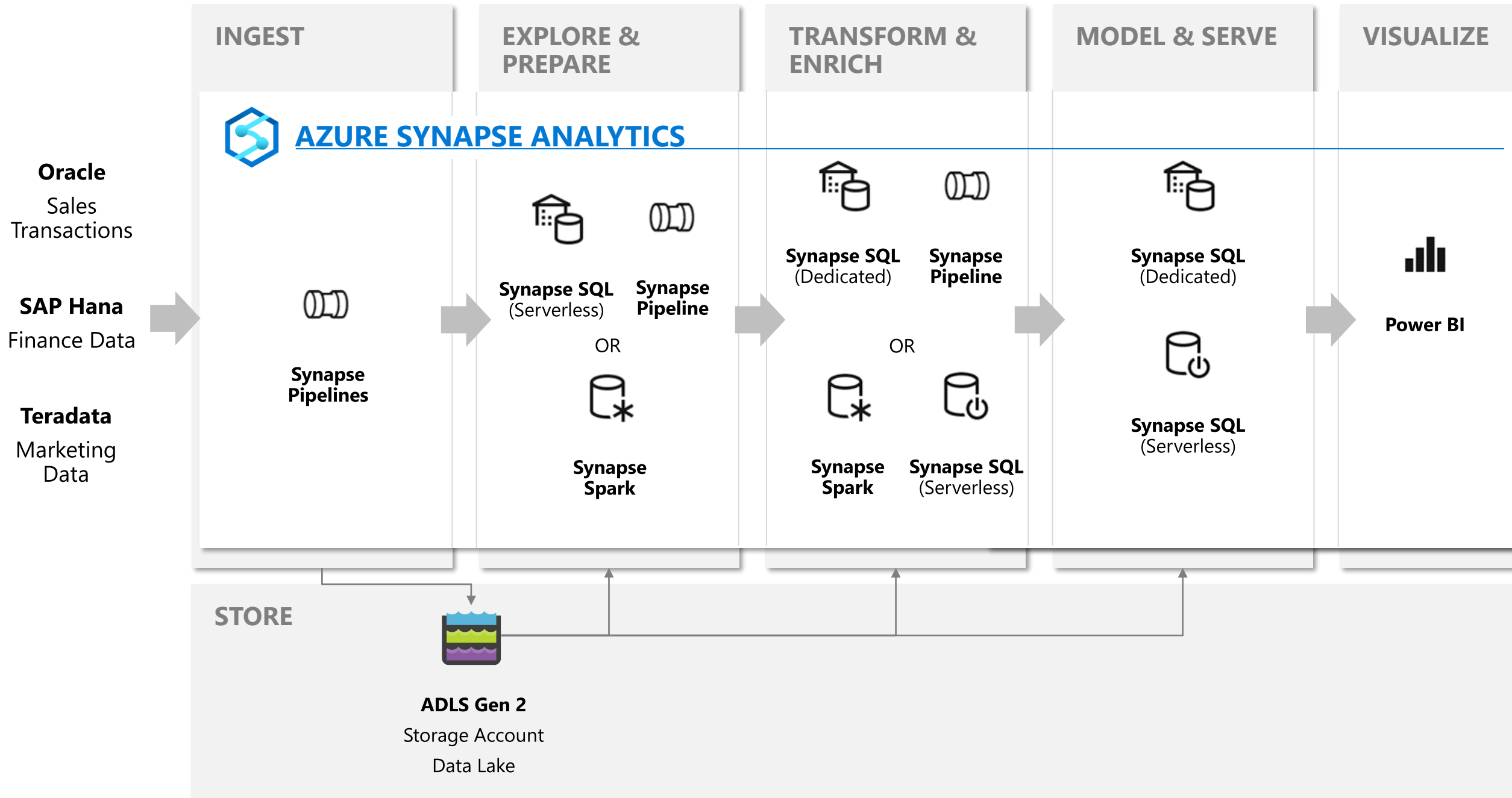




Azure Synapse Analytics (workspaces)

Serverless SQL pools were previously called SQL On-demand pools  
 Dedicated SQL pools were previously called SQL Provisioned pools/SQL pools

# Modern Data Warehouse



# Synapse SQL serverless

## Overview

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

## Benefits

Use SQL to work with files on Azure storage

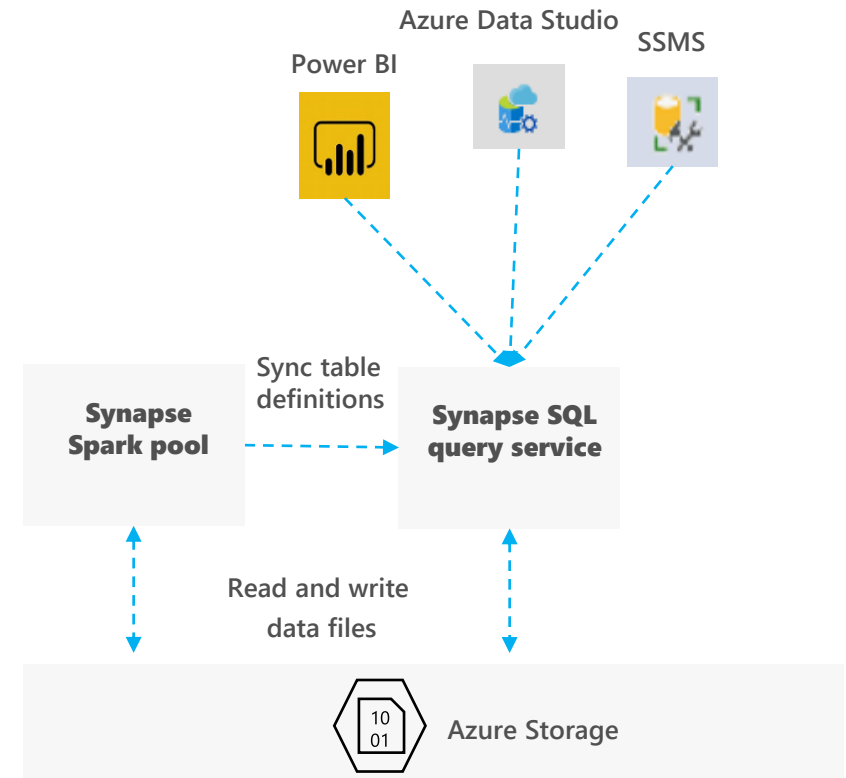
- Directly query files on Azure storage using T-SQL
- Logical Data Warehouse on top of Azure storage
- Easy data transformation of Azure storage files

Supports any tool or library that uses T-SQL to query data

Automatically synchronize tables from Spark pool

## Serverless

- No infrastructure, no upfront cost, no resource reservation
- Auto scale
- Automatic scheme inference
- Pay only for query execution (per data processed)



# Recommended scenarios for serverless

## Basic data exploration

What's in this file? How many rows are there? What's the max value?

Explore with **schema-on-read**, **automatic schema inference** and more...

## Logical data warehouse

How to join data produced by various apps, IoT events, logs, ...?

Create a **virtual SQL layer** on top of heterogeneous data in the lake

## Data transformation

How to transform raw data and prepare it for ingestion into a data store?

Run **batches of distributed SQL queries**, and write-out optimized Parquet



# Azure Synapse Analytics

## Massively Parallel Processing (MPP) Intro

*For Dedicated SQL pools*



# Parallelism

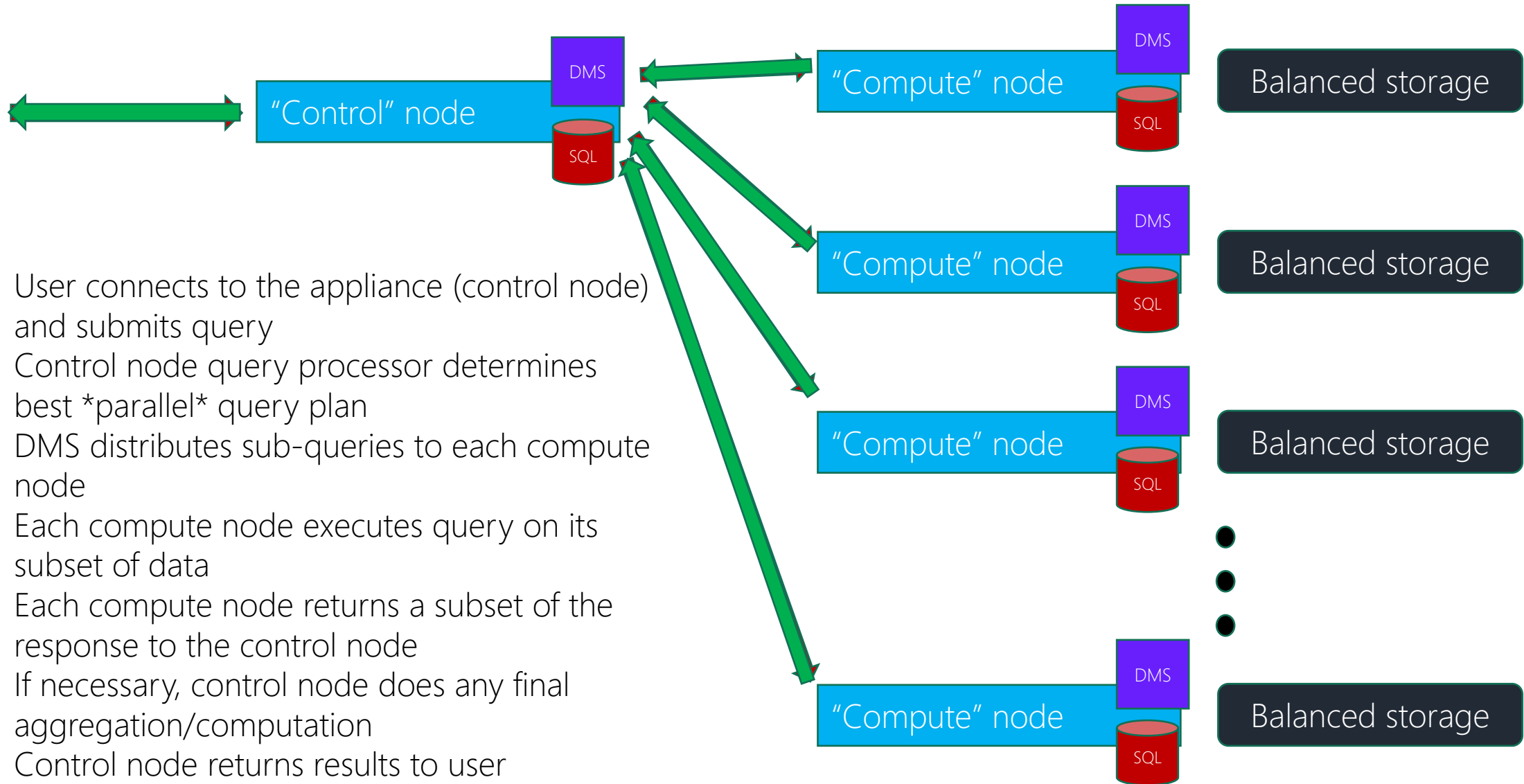
## SMP - Symmetric Multiprocessing

- Multiple CPUs used to complete individual processes simultaneously
- All CPUs share the same memory, disks, and network controllers (scale-up)
- All SQL Server implementations up until now have been SMP
- Mostly, the solution is housed on a shared SAN

## MPP - Massively Parallel Processing

- Uses many separate CPUs running in parallel to execute a single program
- Shared Nothing: Each CPU has its own memory and disk (scale-out)
- Segments communicate using high-speed network between nodes

# Synapse Analytics Logical Architecture (overview)



- 1) User connects to the appliance (control node) and submits query
- 2) Control node query processor determines best \*parallel\* query plan
- 3) DMS distributes sub-queries to each compute node
- 4) Each compute node executes query on its subset of data
- 5) Each compute node returns a subset of the response to the control node
- 6) If necessary, control node does any final aggregation/computation
- 7) Control node returns results to user

*Queries running in parallel on a subset of the data, using separate pipes effectively making the pipe larger*

# Data Warehouse Units (DWU)

## DWU

DW100
DW200
DW300
DW400
DW500
DW1000
DW1500
DW2000
DW2500
DW3000
DW5000
DW6000
DW7500
DW10000
DW15000
DW30000

```
ALTER DATABASE ContosoDW MODIFY  
(service_objective = 'DW1000');
```

Scale your system ⓘ



DW100c

Estimated price ⓘ

**Est. Cost Per Hour**

1.51 USD

[View pricing details](#)

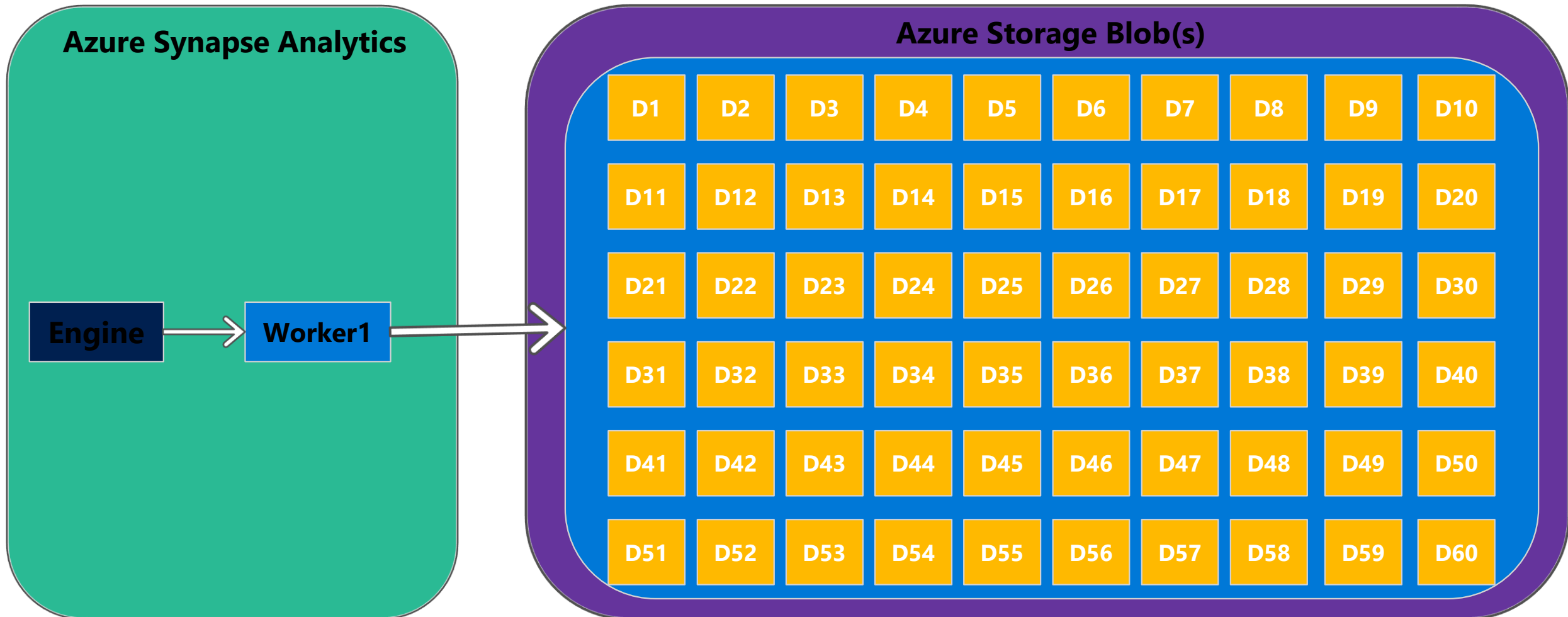
CPU

RAM

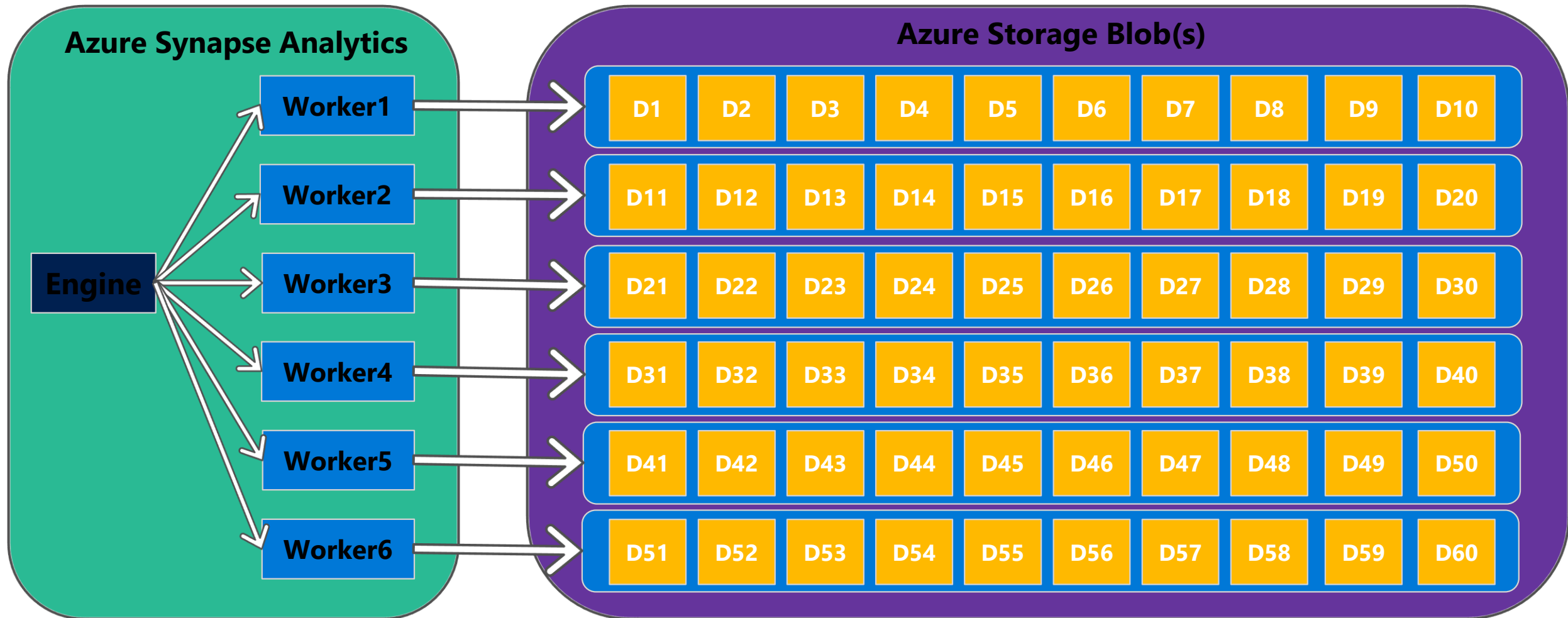
I/O



# Architecture for DW100




# Architecture for DW600



**DEMO**



**Create workspace,  
SQL Pools, Spark  
Pools**



DEMO



# Copy Data Tool, Pipelines, Data Flows (ADF)

**DEMO**



# Linked services



# Query Demo with Azure Synapse Studio

Storage > Compute	Relational Database	ADLS Gen2	Spark Table	Cosmos DB
Dedicated SQL pool	Y	Y (external table)	<i>Soon (parquet format)</i>	N
Serverless SQL pool	N	Y	Y (parquet format)	Y (Synapse link)
Apache Spark pool	Y	Y	Y	Y (Synapse link)

Right-click object type:	Preview	New SQL script -> Select TOP 100 rows	New notebook -> Load to DataFrame
Parquet	N	Y	Y
CSV	Y	Y	Y
JSON	Y	Y	Y
Relational Tbl	N	Y	Y
External Tbl	N	Y	Y
Spark Tbl	N	Y	Y
Cosmos DB	N	Y (add key)	Y

SQL pool supported file formats in ADLS Gen2 are parquet, csv, json (Spark supports many more formats)

# Query Demo with Azure Synapse Studio

Storage > Compute	Relational Database	ADLS Gen2	Spark Table	Cosmos DB
Dedicated SQL pool	Y	Y (external table)	Soon (parquet format)	N
Serverless SQL pool	N	Y	Y (parquet format)	Y (Synapse link)
Apache Spark pool	Y	Y	Y	Y (Synapse link)

Right-click object type:	Preview	New SQL script -> Select TOP 100 rows	New notebook -> Load to DataFrame
Parquet	N	Y	Y
CSV	Y	Y	Y
JSON	Y	Y	Y
Relational Tbl	N	Y	Y
External Tbl	N	Y	Y
Spark Tbl	N	Y	Y
Cosmos DB	N	Y (add key)	Y

SQL pool supported file formats in ADLS Gen2 are parquet, csv, json (Spark supports many more formats)

# Query Demo with Azure Synapse Studio

Storage > Compute	Relational Database	ADLS Gen2	Spark Table	Cosmos DB
Dedicated SQL pool	Y	Y (external table)	Soon (parquet format)	N
Serverless SQL pool	N	Y	Y (parquet format)	Y (Synapse link)
Apache Spark pool	Y	Y	Y	Y (Synapse link)

Right-click object type:	Preview	New SQL script -> Select TOP 100 rows	New notebook -> Load to DataFrame
Parquet	N	Y	Y
CSV	Y	Y	Y
JSON	Y	Y	Y
Relational Tbl	N	Y	Y
External Tbl	N	Y	Y
Spark Tbl	N	Y	Y
Cosmos DB	N	Y (add key)	Y

SQL pool supported file formats in ADLS Gen2 are parquet, csv, json (Spark supports many more formats)



# Query Demo with Azure Synapse Studio

Storage > Compute	Relational Database	ADLS Gen2	Spark Table	Cosmos DB
Dedicated SQL pool	Y	Y (external table)	Soon (parquet format)	N
Serverless SQL pool	N	Y	Y (parquet format)	Y (Synapse link)
Apache Spark pool	Y	Y	Y	Y (Synapse link)

Right-click object type:	Preview	New SQL script -> Select TOP 100 rows	New notebook -> Load to DataFrame
Parquet	N	Y	Y
CSV	Y	Y	Y
JSON	Y	Y	Y
Relational Tbl	N	Y	Y
External Tbl	N	Y	Y
Spark Tbl	N	Y	Y
Cosmos DB	N	Y (add key)	Y

SQL pool supported file formats in ADLS Gen2 are parquet, csv, json (Spark supports many more formats)

# Query Demo with Azure Synapse Studio

Storage > Compute	Relational Database	ADLS Gen2	Spark Table	Cosmos DB
Dedicated SQL pool	Y	Y (external table)	Soon (parquet format)	N
Serverless SQL pool	N	Y	Y (parquet format)	Y (Synapse link)
Apache Spark pool	Y	Y	Y	Y (Synapse link)

Right-click object type:	Preview	New SQL script -> Select TOP 100 rows	New notebook -> Load to DataFrame
Parquet	N	Y	Y
CSV	Y	Y	Y
JSON	Y	Y	Y
Relational Tbl	N	Y	Y
External Tbl	N	Y	Y
Spark Tbl	N	Y	Y
Cosmos DB	N	Y (add key)	Y

SQL pool supported file formats in ADLS Gen2 are parquet, csv, json (Spark supports many more formats)

# Query Demo with Azure Synapse Studio

Storage > Compute	Relational Database	ADLS Gen2	Spark Table	Cosmos DB
Dedicated SQL pool	Y	Y (external table)	<i>Soon (parquet format)</i>	N
Serverless SQL pool	N	Y	Y (parquet format)	Y (Synapse link)
Apache Spark pool	Y	Y	Y	Y (Synapse link)

Right-click object type:	Preview	New SQL script -> Select TOP 100 rows	New notebook -> Load to DataFrame
Parquet	N	Y	Y
CSV	Y	Y	Y
JSON	Y	Y	Y
Relational Tbl	N	Y	Y
External Tbl	N	Y	Y
Spark Tbl	N	Y	Y
Cosmos DB	N	Y (add key)	Y

SQL pool supported file formats in ADLS Gen2 are parquet, csv, json (Spark supports many more formats)

# Query Demo with Azure Synapse Studio

Storage > Compute	Relational Database	ADLS Gen2	Spark Table	Cosmos DB
Dedicated SQL pool	Y	Y (external table)	Soon (parquet format)	N
Serverless SQL pool	N	Y	Y (parquet format)	Y (Synapse link)
Apache Spark pool	Y	Y	Y	Y (Synapse link)

Right-click object type:	Preview	New SQL script -> Select TOP 100 rows	New notebook -> Load to DataFrame
Parquet	N	Y	Y
CSV	Y	Y	Y
JSON	Y	Y	Y
Relational Tbl	N	Y	Y
External Tbl	N	Y	Y
Spark Tbl	N	Y	Y
Cosmos DB	N	Y (add key)	Y

SQL pool supported file formats in ADLS Gen2 are parquet, csv, json (Spark supports many more formats)

# Query Demo with Azure Synapse Studio

Storage > Compute	Relational Database	ADLS Gen2	Spark Table	Cosmos DB
Dedicated SQL pool	Y	Y (external table)	Soon (parquet format)	N
Serverless SQL pool	N	Y	Y (parquet format)	Y (Synapse link)
Apache Spark pool	Y	Y	Y	Y (Synapse link)

Right-click object type:	Preview	New SQL script -> Select TOP 100 rows	New notebook -> Load to DataFrame
Parquet	N	Y	Y
CSV	Y	Y	Y
JSON	Y	Y	Y
Relational Tbl	N	Y	Y
External Tbl	N	Y	Y
Spark Tbl	N	Y	Y
Cosmos DB	N	Y (add key)	Y

SQL pool supported file formats in ADLS Gen2 are parquet, csv, json (Spark supports many more formats)

# Query Demo with Azure Synapse Studio

Storage > Compute	Relational Database	ADLS Gen2	Spark Table	Cosmos DB
Dedicated SQL pool	Y	Y (external table)	Soon (parquet format)	N
Serverless SQL pool	N	Y	Y (parquet format)	Y (Synapse link)
Apache Spark pool	Y	Y	Y	Y (Synapse link)

Right-click object type:	Preview	New SQL script -> Select TOP 100 rows	New notebook -> Load to DataFrame
Parquet	N	Y	Y
CSV	Y	Y	Y
JSON	Y	Y	Y
Relational Tbl	N	Y	Y
External Tbl	N	Y	Y
Spark Tbl	N	Y	Y
Cosmos DB	N	Y (add key)	Y

SQL pool supported file formats in ADLS Gen2 are parquet, csv, json (Spark supports many more formats)

# Query Demo with Azure Synapse Studio

If you have data in ADLS Gen2 that is stored in [Delta Lake](#) format (via methods such as a [Spark Table](#) or [Azure Data Factory](#)), that data can be read via the Apache Spark pool (no matter if the Delta Lake is built via [open source](#) or [Databricks](#)), soon via Serverless SQL (a workaround for now is [here](#)), and not via a Dedicated SQL pool. Note that Power BI can [read the Delta Lake format](#) but requires you to use the Power BI Databricks connector.

If you have data in ADLS Gen2 that is in the [Common Data Model \(CDM\)](#) format (via methods such as [Dynamics 365 CDS export to ADLS Gen2](#) or stored [directly](#), or via [Azure Data Factory](#)), Serverless SQL is not yet able to read it, but you can via the Apache Spark pool (see [Spark CDM connector](#)). Note that Power BI can read CDM if using Power BI dataflows.

**DEMO**

The slide features a white background with decorative elements. At the top left, the word "DEMO" is written in a bold, black, sans-serif font. Below it are two horizontal red lines of varying lengths. In the top right corner, there is a faint, dotted red arc. In the bottom left corner, there is a more prominent dotted red arc that curves downwards and to the right.

**Query 30 billion  
rows**



**DEMO**

The slide features a minimalist design with a white background. At the top left, the word "DEMO" is written in a bold, black, sans-serif font. Below it are two horizontal red lines of varying lengths. In the bottom left corner, a dotted red line forms a curved shape. In the top right corner, a dotted grey line forms a curved shape.

**COPY statement**

**DEMO**

The slide features a minimalist design with two horizontal red lines of varying lengths extending from the left edge. A dotted red curve is visible in the top right corner, and another dotted red curve is in the bottom left corner. The text 'DEMO' is positioned above the top line, and 'Spark notebooks' is positioned below the bottom line.

# Spark notebooks

**DEMO**

The slide features a minimalist design with a white background. At the top left, the word "DEMO" is written in a bold, black, sans-serif font. Below it are two horizontal red lines of varying lengths. In the bottom left corner, a decorative curve of red dots arcs upwards and to the right. In the top right corner, a similar decorative curve of grey dots arcs downwards and to the left.

**Power BI**

**DEMO**



**SSMS**



**DEMO**



**Monitor**



DEMO

The slide features a minimalist design with a white background. At the top left, the word "DEMO" is written in a bold, black, sans-serif font. Below it, two horizontal red lines of varying lengths extend to the right. In the bottom left corner, a series of red dots form a curved path that starts horizontally and then curves downwards. In the top right corner, a series of grey dots form a curved path that starts vertically and then curves to the right.

# Synapse Samples

DEMO

The slide features a minimalist design with a white background. At the top left, the word "DEMO" is written in a bold, black, sans-serif font. Below it, two horizontal red lines of varying lengths extend to the right. In the top right corner, a faint, dotted red arc is visible. In the bottom left corner, another dotted red arc is present. The main title "Federated Query" is centered on the left side in a large, bold, black font. At the bottom right, a URL is displayed in a smaller, black font.

# Federated Query

<https://www.jamesserra.com/archive/2020/10/synapse-and-federated-queries/>

# Thank you

---

James Serra

Blog: [JamesSerra.com](http://JamesSerra.com)



@jameserra



[JamesSerra3@gmail.com](mailto:JamesSerra3@gmail.com)

**BIG DATA CONFERENCE**  
**EUROPE 2020**





# Azure Synapse Analytics

## APPENDIX



# Azure Synapse Analytics

**Azure Synapse Analytics** added multiple new features during Microsoft Ignite 2020. These new features bring accelerated time to insight with new built-in capabilities for both data exploration and data warehousing.

**Power BI performance accelerator for Azure Synapse Analytics (private preview)** is a new self-managed process that enables automatic performance tuning for workloads and queries ran in Power BI.

**Azure Synapse Link now includes Synapse SQL (public preview):** Customers can now use a serverless SQL pool in Azure Synapse Analytics to perform interactive analytics over Azure Cosmos DB data enabling quick insights and exploratory analysis without the need to employ complex data movement steps

**Enhanced support for analyzing text delimited files (public preview):** Includes fast parser, automatic schema discovery, and transform as CSV.

**COPY command (Generally Available):** Includes updates for automatic file splitting, automatic schema discovery, and complex data type support.

**Fast streaming ingestion (Generally Available):** This new connector can handle ingestion rates exceeding 200MB/sec while ensuring very low latencies.

**Column-level Encryption (public preview):** With CLE, customers gain the ability to use different protection keys for different columns in a table, with each key having its own access permissions.

**MERGE support (public preview):** Users can now synchronize two tables in a single step, streamlining the data processing using a single step statement while improving code readability and debugging.

**Inline Table-Valued Functions (public preview):** By extending support for inline table-valued functions (TVFs), users can now return a table result set based on specified parameters

**Stored procedures support (public preview):** To enable customers to operationalize their SQL transformation logic over the data residing in their data lakes, we have added stored procedures support to our serverless model

[Learn more](#)

Updates  
Announced @ Ignite  
September 2020

The image displays two screenshots of the Azure Synapse Analytics user interface. The top screenshot shows a workspace with a list of data sources on the left, including 'Azure Cosmos DB' which is highlighted with a green box. A SQL script editor is open on the right, showing a query for schema inference from an Azure Cosmos DB database. The bottom screenshot shows the 'Data' workspace with a list of SQL pools. A 'Power BI performance' settings panel is open, also highlighted with a green box, showing options to enable automatic optimization, set the current database size (433.758TB), and configure an optimization schedule (every 18 days).



# Azure Synapse Analytics

## Data Storage and Performance Optimizations



# Database Tables

## Optimized Storage

Reduce Migration Risk

Less Data Scanned

Smaller Cache Required

Smaller Clusters

Faster Queries



Columnar Storage



Columnar Ordering

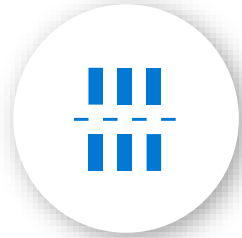
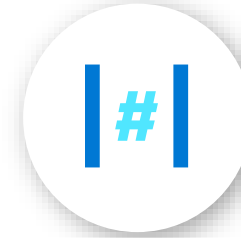


Table Partitioning



Hash Distribution



Nonclustered Indexes

# Tables – Indexes

## Clustered Columnstore index (Default Primary)

Highest level of data compression

Best overall query performance

## Clustered index (Primary)

Performant for looking up a single to few rows

## Heap (Primary)

Faster loading and landing temporary data

Best for small lookup tables

## Nonclustered indexes (Secondary)

Enable ordering of multiple columns in a table

Allows multiple nonclustered on a single table

Can be created on any of the above primary indexes

More performant lookup queries

```
-- 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);
```

# Tables – Distributions

## Round-robin distributed

Distributes table rows evenly across all distributions at random.

## Hash distributed

Distributes table rows across the Compute nodes by using a deterministic hash function to assign each row to one distribution.

## Replicated

Full copy of table accessible on each Compute node.

```
CREATE TABLE dbo.OrderTable
(
    OrderId INT NOT NULL,
    Date DATE NOT NULL,
    Name VARCHAR(2),
    Country VARCHAR(2)
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX,
    DISTRIBUTION = HASH([OrderId]) |
                                ROUND ROBIN |
                                REPLICATED
);
```

# Tables – Partitions

## Overview

Table partitions divide data into smaller groups

In most cases, partitions are created on a date column

Supported on all table types

RANGE RIGHT – Used for time partitions

RANGE LEFT – Used for number partitions

## Benefits

Improves efficiency and performance of loading and querying by limiting the scope to subset of data.

Offers significant query performance enhancements where filtering on the partition key can eliminate unnecessary scans and eliminate IO.

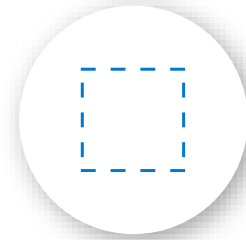
```
CREATE TABLE partitionedOrderTable
(
    OrderId INT NOT NULL,
    Date DATE NOT NULL,
    Name VARCHAR(2),
    Country VARCHAR(2)
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX,
    DISTRIBUTION = HASH([OrderId]),
    PARTITION (
        [Date] RANGE RIGHT FOR VALUES (
            '2000-01-01', '2001-01-01',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

# Common table distribution methods

Table Category	Recommended Distribution Option
<b>Fact</b>	<p>Use hash-distribution with clustered columnstore index. Performance improves because hashing enables the platform to localize certain operations within the node itself during query execution.</p> <p>Operations that benefit:</p> <pre>COUNT(DISTINCT( &lt;hashed_key&gt; )) OVER PARTITION BY &lt;hashed_key&gt; most JOIN &lt;table_name&gt; ON &lt;hashed_key&gt; GROUP BY &lt;hashed_key&gt;</pre>
<b>Dimension</b>	<p>Use replicated for smaller tables. If tables are too large to store on each Compute node, use hash-distributed.</p>
<b>Staging</b>	<p>Use round-robin for the staging table. The load with CTAS is faster. Once the data is in the staging table, use INSERT...SELECT to move the data to production tables.</p>



# Database Views



Views



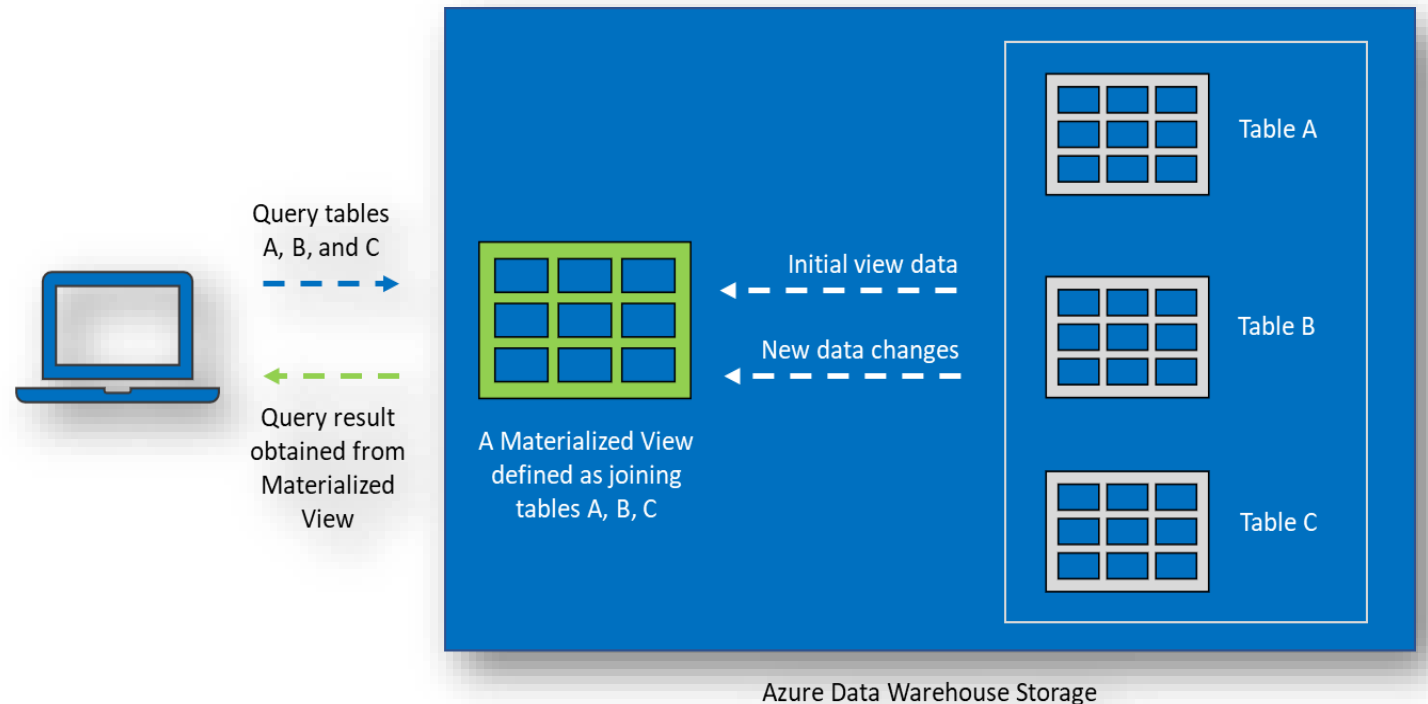
Materialized Views

Scope: **Generally Available**

# Best in class price performance

## Interactive dashboarding with Materialized Views

- Automatic data refresh and maintenance
- Automatic query rewrites to improve performance
- Built-in advisor



# Materialized views

## Overview

A materialized view pre-computes, stores, and maintains its data like a table.

Materialized views are automatically updated when data in underlying tables are changed. This is a synchronous operation that occurs as soon as the data is changed.

The auto caching functionality allows Azure Synapse Analytics Query Optimizer to consider using the indexed view even if the view is not referenced in the query.

Supported aggregations: MAX, MIN, AVG, COUNT, COUNT\_BIG, SUM, VAR, STDEV. **Note: Aggregate functions are required in the SELECT list of the materialized view definition**

## Benefits

Automatic and synchronous data refresh with data changes in base tables. No user action is required.

High availability and resiliency as regular tables

```
-- Create indexed view
CREATE MATERIALIZED VIEW Sales.vw_Orders
WITH
(
  DISTRIBUTION = ROUND_ROBIN |
  HASH(ProductID)
)
AS
  SELECT SUM(UnitPrice*OrderQty) AS Revenue,
         OrderDate,
         ProductID,
         COUNT_BIG(*) AS OrderCount
  FROM   Sales.SalesOrderDetail
  GROUP BY OrderDate, ProductID;
GO

-- Disable index view and put it in suspended mode
ALTER INDEX ALL ON Sales.vw_Orders DISABLE;

-- Re-enable index view by rebuilding it
ALTER INDEX ALL ON Sales.vw_Orders REBUILD;
```

# COPY command

## Overview

Copies data from source to destination

## Benefits

Retrieves data from all files from the folder and all its subfolders.

Supports multiple locations from the same storage account, separated by comma

Supports Azure Data Lake Storage (ADLS) Gen 2 and Azure Blob Storage.

Supports CSV, PARQUET, ORC file formats

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

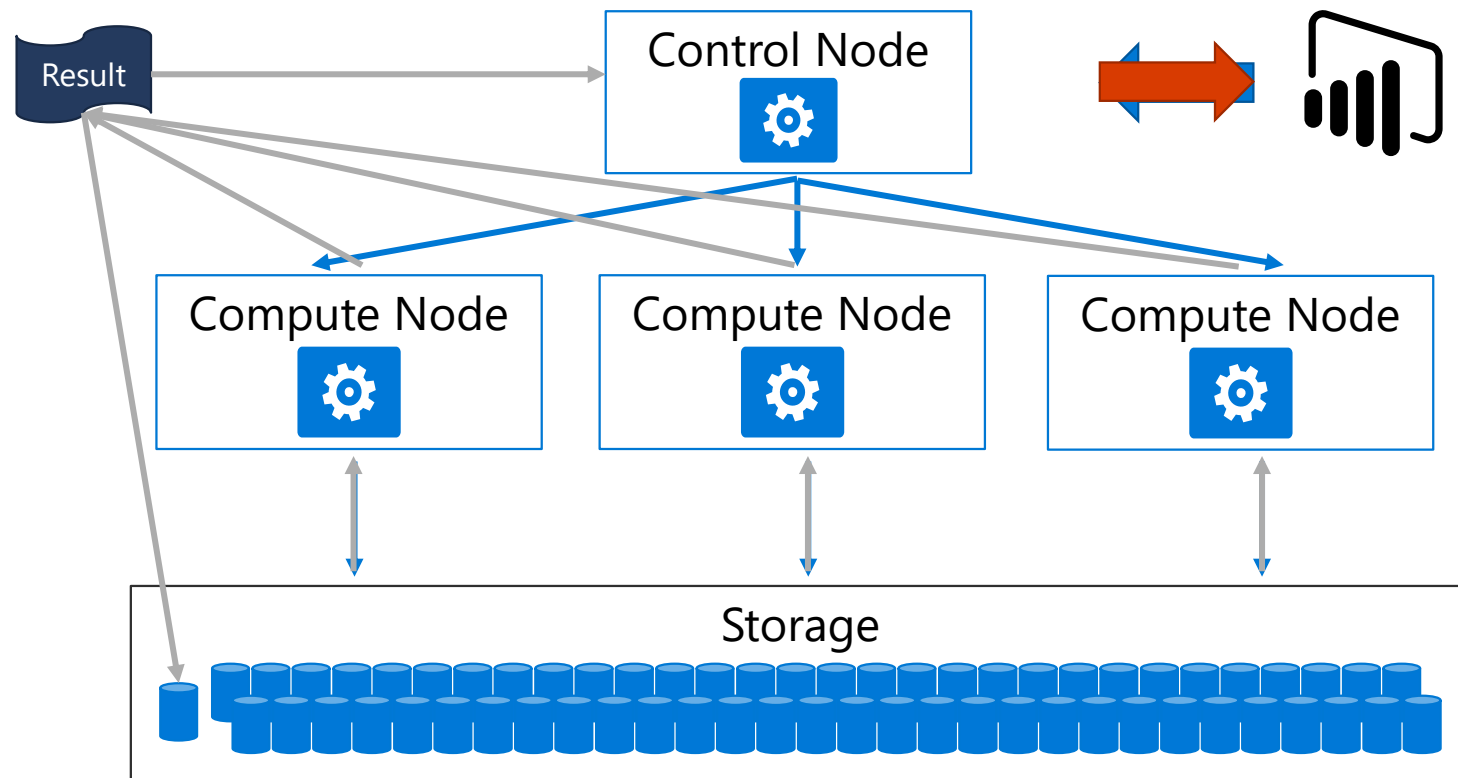
```
COPY INTO test_parquet
FROM
'https://XXX.blob.core.windows.net/customerdatasets/test.parquet'
WITH (
    FILE_FORMAT = myFileFormat
    CREDENTIAL=(IDENTITY= 'Shared Access Signature',
SECRET='<Your_SAS-Token>')
)
```

Scope: **Generally Available**

## Best in class price performance

## Interactive dashboarding with Resultset Caching

- Millisecond responses with resultset caching
- Cache survives pause/resume/scale operations
- Fully managed cache (1TB in size)



Enable caching: **Alter Database <DBNAME> Set Result\_Set\_Caching ON**  
Purge cache: **DBCC DropResultSetCache**

# Result-set caching

## Overview

Cache the results of a query in DW storage. This enables interactive response times for repetitive queries against tables with infrequent data changes.

The result-set cache persists even if a data warehouse is paused and resumed later.

Query cache is invalidated and refreshed when underlying table data or query code changes.

Result cache is evicted regularly based on a time-aware least recently used algorithm (TLRU).

## Benefits

Enhances performance when same result is requested repetitively

Reduced load on server for repeated queries

Offers monitoring of query execution with a result cache hit or miss

```
-- Turn on/off result-set caching for a database
-- Must be run on the MASTER database
ALTER DATABASE {database_name}
SET RESULT_SET_CACHING { ON | OFF }

-- Turn on/off result-set caching for a client session
-- Run on target data warehouse
SET RESULT_SET_CACHING {ON | OFF}

-- Check result-set caching setting for a database
-- Run on target data warehouse
SELECT is_result_set_caching_on
FROM sys.databases
WHERE name = {database_name}

-- Return all query requests with cache hits
-- Run on target data warehouse
SELECT *
FROM sys.dm_pdw_request_steps
WHERE command like '%DWResultCacheDb%'
AND step_index = 0
```

# Resource classes

## Overview

Pre-determined resource limits defined for a user or role.

## Benefits

Govern the system memory assigned to each query.

Effectively used to control the number of concurrent queries that can run on a data warehouse.

## Exemptions to concurrency limit:

CREATE|ALTER|DROP (TABLE|USER|PROCEDURE|VIEW|LOGIN)

CREATE|UPDATE|DROP (STATISTICS|INDEX)

SELECT from system views and DMVs

EXPLAIN

Result-Set Cache

TRUNCATE TABLE

ALTER AUTHORIZATION

CREATE|UPDATE|DROP STATISTICS

```
/* View resource classes in the data warehouse */  
SELECT name  
FROM sys.database_principals  
WHERE name LIKE '%rc%' AND type_desc = 'DATABASE_ROLE';  
  
/* Change user's resource class to 'largerc' */  
EXEC sp_addrolemember 'largerc', 'loaduser';  
  
/* Decrease the loading user's resource class */  
EXEC sp_droprolemember 'largerc', 'loaduser';
```

# Resource class types

## Static Resource Classes

Allocate the same amount of memory independent of the current service-level objective (SLO).

Well-suited for fixed data sizes and loading jobs.

## Dynamic Resource Classes

Allocate a variable amount of memory depending on the current SLO.

Well-suited for growing or variable datasets.

All users default to the *smallrc* dynamic resource class.

## Static resource classes:

```
staticrc10 | staticrc20 | staticrc30 |
staticrc40 | staticrc50 | staticrc60 |
staticrc70 | staticrc80
```

## Dynamic resource classes:

```
smallrc | mediumrc | largerc | xlargerc
```

Resource Class	Percentage Memory	Max. Concurrent Queries
smallrc	3%	32
mediumrc	10%	10
largerc	22%	4
xlargerc	70%	1



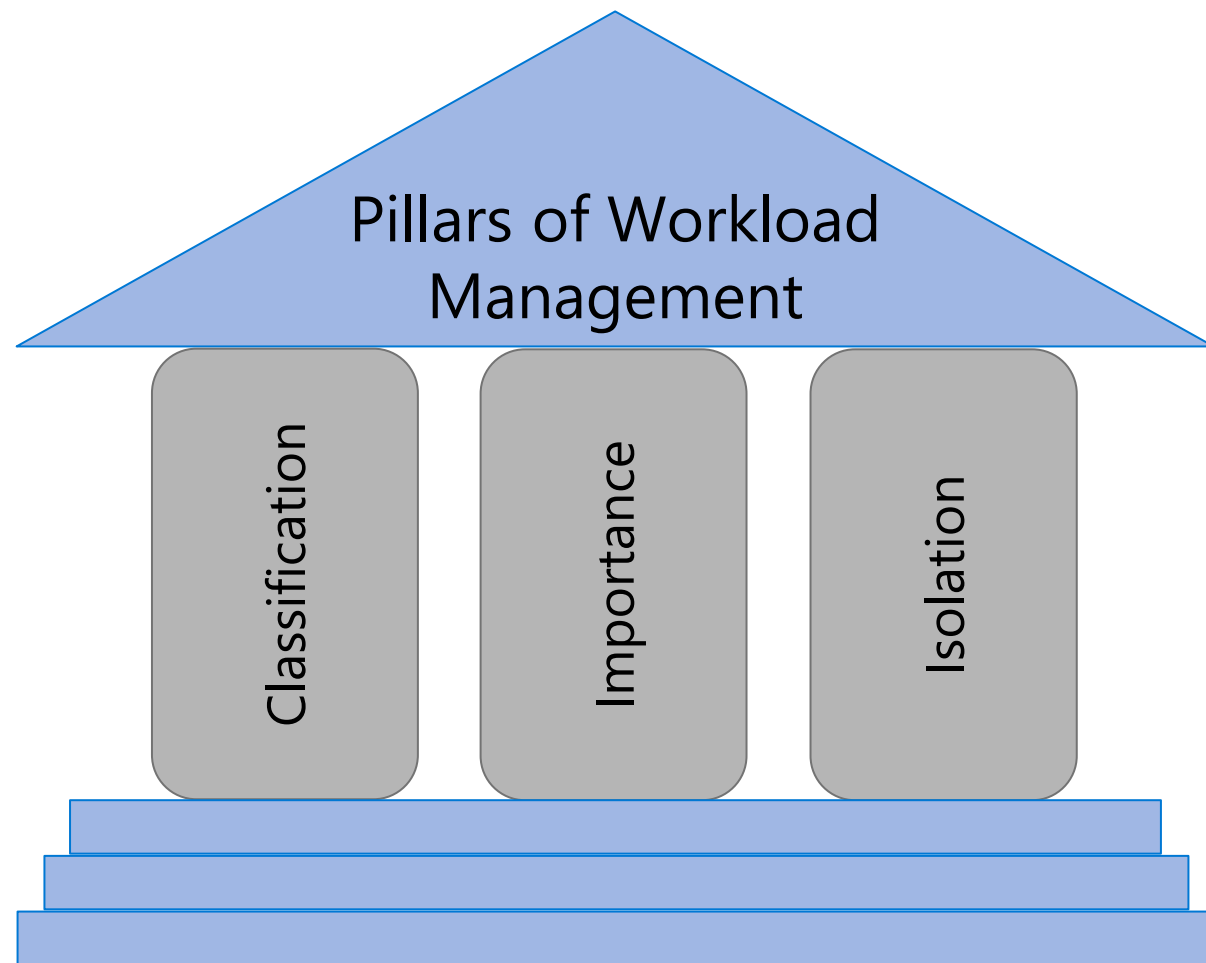
# Workload Management

## Overview

It manages resources, ensures highly efficient resource utilization, and maximizes return on investment (ROI).

The three pillars of workload management are

1. Workload Classification – To assign a request to a workload group and setting importance levels. Example: Loads and Queries
2. Workload Importance – To influence the order in which a request gets access to resources.
3. Workload Isolation – To reserve resources for a workload group.



# Workload Isolation

## Overview

**Allocate fixed resources to workload group (ie. "loading data").**

Assign maximum and minimum usage for varying resources under load. These adjustments can be done live without having to SQL Analytics offline.

## Benefits

Reserve resources for a group of requests

Limit the amount of resources a group of requests can consume

Shared resources accessed based on importance level

Set Query timeout value. Get DBAs out of the business of killing runaway queries

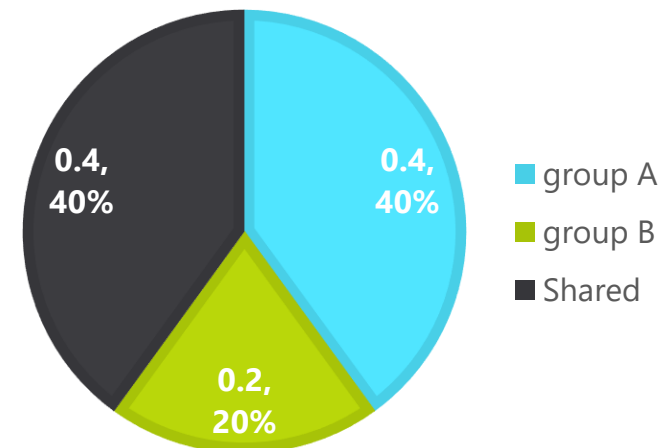
## Monitoring DMVs

`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 ]
)[ ; ]
```

RESOURCE ALLOCATION



# Workload importance

## Overview

Queries past the concurrency limit enter a FiFo queue

By default, queries are released from the queue on a first-in, first-out basis as resources become available

**Workload importance allows higher priority queries to receive resources immediately regardless of queue (ie. "company CEO")**

## Example Video

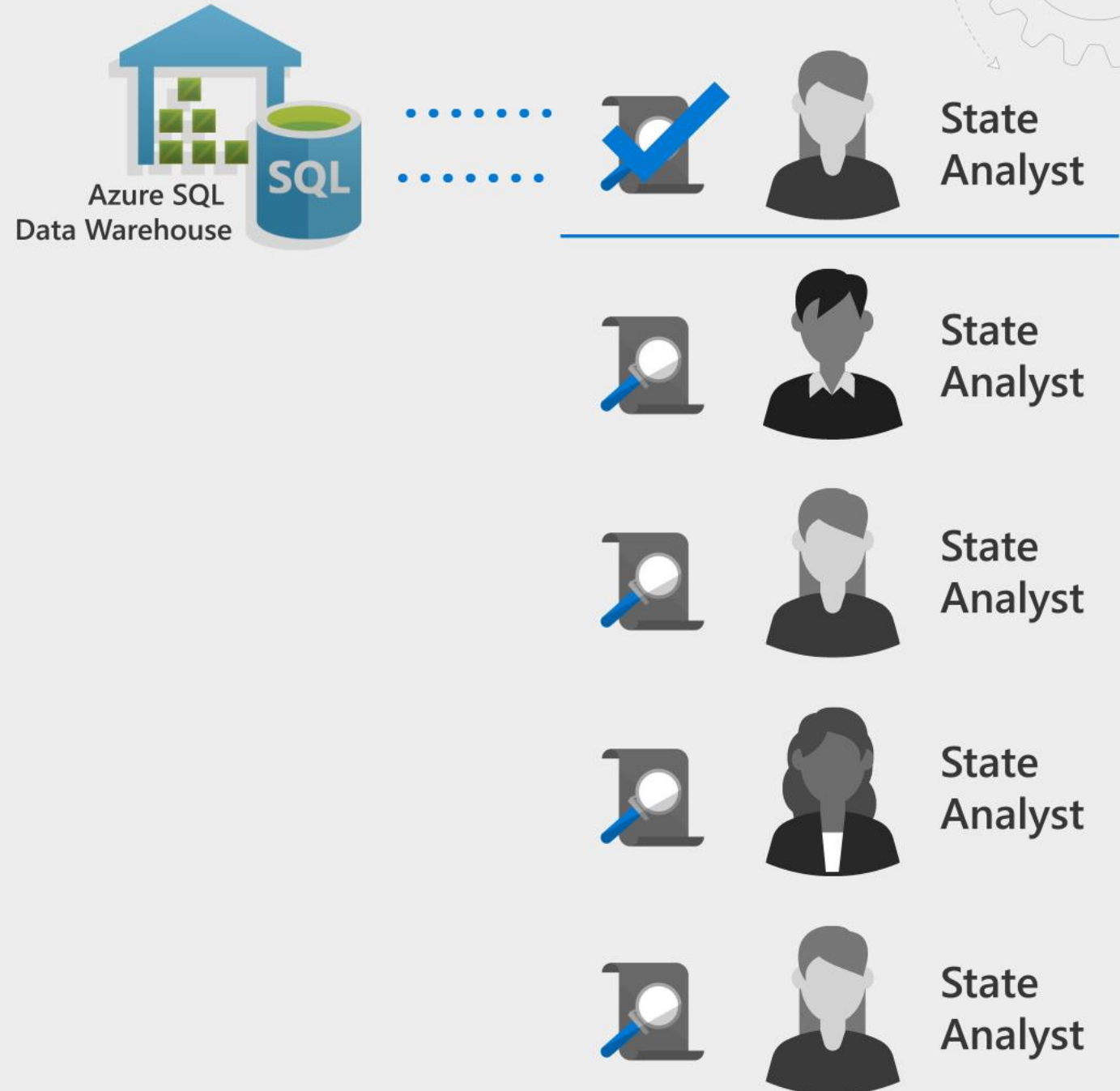
State analysts have normal importance.

National analyst is assigned high importance.

State analyst queries execute in order of arrival

When the national analyst's query arrives, it jumps to the top of the queue

```
CREATE WORKLOAD CLASSIFIER National_Analyst
WITH
(
  [WORKLOAD_GROUP] = 'smallrc' ]
  [IMPORTANCE] = HIGH]
  [MEMBERNAME] = 'National_Analyst_Login' ]
```



# Workload classification

## Overview

Map queries to allocations of resources via pre-determined rules.

***Use with workload importance to effectively share resources across different workload types (ie. "ETL login").***

If a query request is not matched to a classifier, it is assigned to the default workload group (smallrc resource class).

## Benefits

Map queries to both Resource Management and Workload Isolation concepts.

Manage groups of users with only a few classifiers.

## 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 = '<Resource Class>' ]
  [IMPORTANCE = { LOW
                  BELOW_NORMAL
                  NORMAL
                  ABOVE_NORMAL
                  HIGH
                }
  ]
  [MEMBERNAME = 'security_account']
)
```

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

*IMPORTANCE: specifies relative importance of request*

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

# Workload Management Sample

---

```
Use master;
```

```
IF NOT EXISTS (SELECT * FROM sys.sql_logins WHERE name = 'ELTLogin')
BEGIN
CREATE LOGIN [ELTLogin] WITH PASSWORD='Password123'
END
;
```

```
Use SQLPool01;
```

```
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = 'ELTLogin')
BEGIN
CREATE USER [ELTLogin] FOR LOGIN [ELTLogin]
END
;
```

```
CREATE WORKLOAD GROUP DataLoads Isolation
WITH ( MIN_PERCENTAGE_RESOURCE = 20
      ,CAP_PERCENTAGE_RESOURCE = 100
      ,REQUEST_MIN_RESOURCE_GRANT_PERCENT = 5)
;
```

```
CREATE WORKLOAD CLASSIFIER [wgcELTLogin] Classification
WITH (WORKLOAD_GROUP = 'DataLoads'
      ,IMPORTANCE = HIGH Importance
      ,MEMBERNAME = 'ELTLogin')
```

```
CREATE WORKLOAD CLASSIFIER [wgcELTLogin]
WITH (WORKLOAD_GROUP = 'xlargerc'
      ,IMPORTANCE = HIGH
      ,MEMBERNAME = 'ELTLogin')
```

```
;
```

# Dynamic Management Views (DMVs)

## Overview

Dynamic Management Views (DMV) are queries that return information about model objects, server operations, and server health.

## Benefits:

Simple SQL syntax

Returns result in table format

Easier to read and copy result

# SQL Monitor with DMVs

## Overview

Offers monitoring of

- all open, closed sessions
- count sessions by user
- count completed queries by user
- all active, complete queries
- longest running queries
- memory consumption

Count sessions by user

```
--count sessions by user
SELECT login_name, COUNT(*) as session_count FROM
sys.dm_pdw_exec_sessions where status = 'Closed' and session_id
<> session_id() GROUP BY login_name;
```

List all open sessions

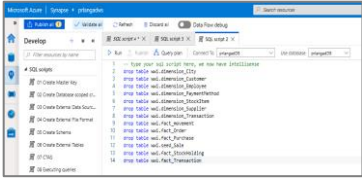
```
-- List all open sessions
SELECT * FROM sys.dm_pdw_exec_sessions where status <> 'Closed'
and session_id <> session_id();
```

List all active queries

```
-- List all active queries
SELECT * FROM sys.dm_pdw_exec_requests WHERE status not in
('Completed','Failed','Cancelled') AND session_id <> session_id()
ORDER BY submit_time DESC;
```

# Developer Tools

Azure Synapse Analytics

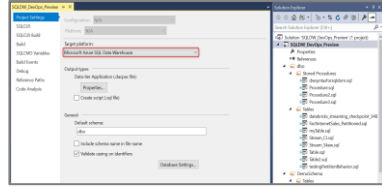


Azure Cloud Service

Offers end-to-end lifecycle for analytics

Connects to multiple services

Visual Studio - SSDT database projects



Runs on Windows

Create, maintain database code, compile, code refactoring

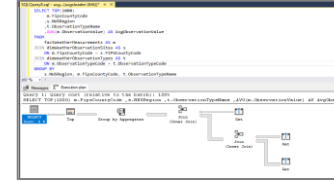
Azure Data Studio



Runs on Windows, Linux, macOS

Light weight editor, (queries and extensions)

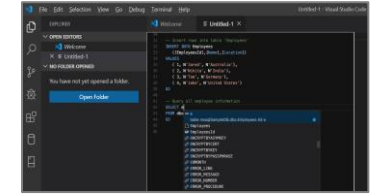
SQL Server Management Studio



Runs on Windows

Offers GUI support to query, design and manage

Visual Studio Code



Runs on Windows, Linux, macOS

Offers development experience with light-weight code editor



# Azure Advisor recommendations

## Suboptimal Table Distribution

Reduce data movement by replicating tables

## Data Skew

Choose new hash-distribution key

Slowest distribution limits performance

## Cache Misses

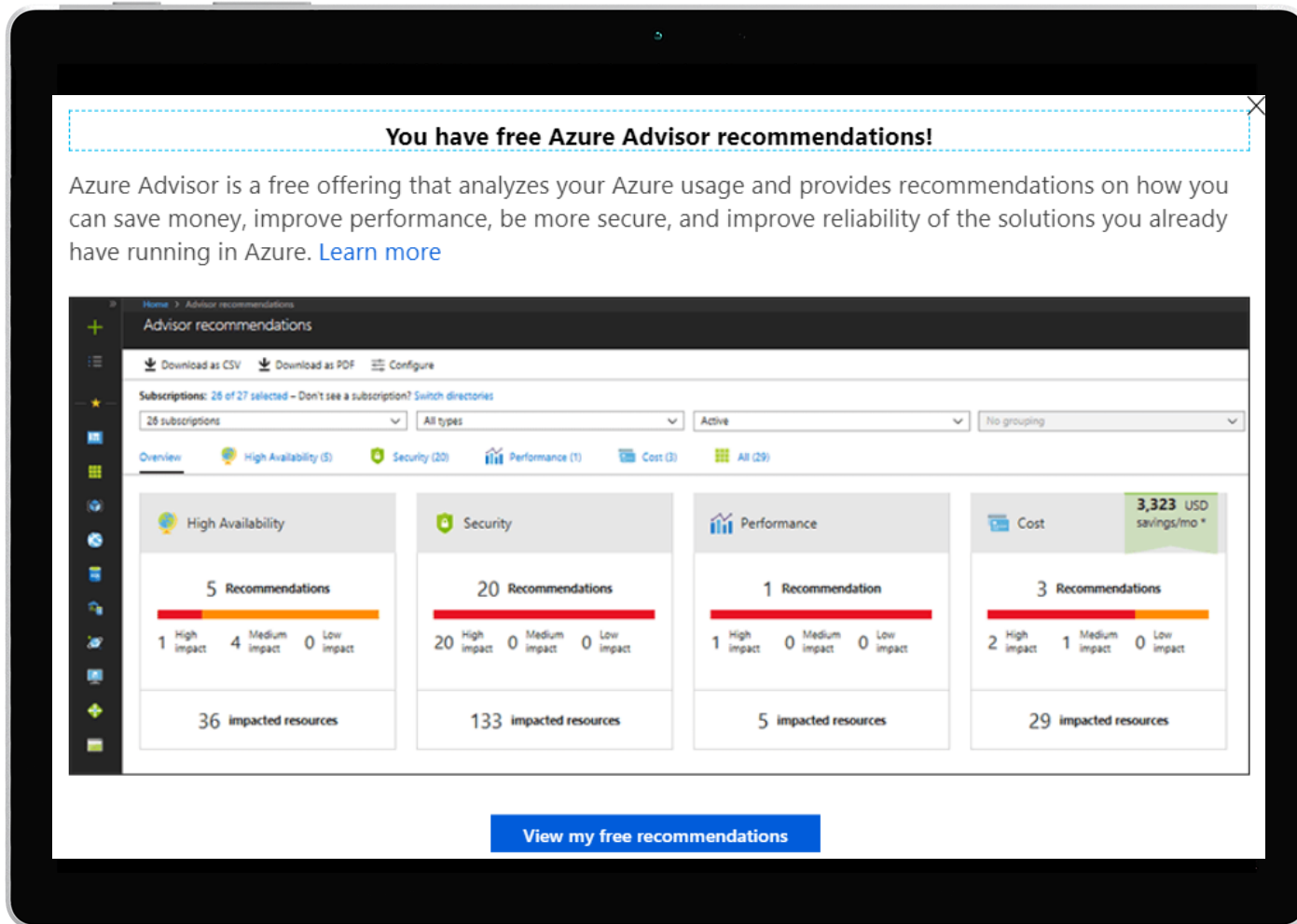
Provision additional capacity

## Tempdb Contention

Scale or update user resource class

## Suboptimal Plan Selection

Create or update table statistics



# Maintenance schedule (preview)

## Overview

Choose a time window for your upgrades.

Select a primary and secondary window within a seven-day period.

Windows can be from 3 to 8 hours.

24-hour advance notification for maintenance events.

## Benefits

Ensure upgrades happen on your schedule.

Predictable planning for long-running jobs.

Stay informed of start and end of maintenance.

# Maintenance Schedule (preview)

**i** The following maintenance schedule is currently active on this data warehouse. Maintenance may occur during both the primary and the secondary windows. DW400c and lower performance levels could experience maintenance outside of a designated maintenance window. Find out more about [maintenance schedules](#).

[View maintenance notifications and create alerts](#)

## Choose primary window <sup>i</sup>

Saturday - Sunday  Tuesday - Thursday

## Primary maintenance window

Day <sup>i</sup>  
Saturday

Start time <sup>i</sup>  
05:00 UTC

Time window <sup>i</sup>  
7 hours

## Secondary maintenance window

Day <sup>i</sup>  
Wednesday

Start time <sup>i</sup>  
00:00 UTC

Time window <sup>i</sup>  
8 hours

## Schedule summary

Primary maintenance window  
Saturday 05:00 UTC (7 hours)

Secondary maintenance window  
Wednesday 00:00 UTC (8 hours)

[Home](#) > [Service Health - Health alerts](#) > [Create rule](#)

## Create rule

Rules management



### ALERT TARGET

Subscription \* <sup>i</sup>  Service(s) \* <sup>i</sup>  Region(s) \* <sup>i</sup>

### Service health criteria

Event type \* <sup>i</sup>



- Select all
- Service issue
- Planned maintenance
- Health advisories
- Security advisory

### Add action group

Action group name \* <sup>i</sup>

Subscription \* <sup>i</sup>  
MTC NYC - James Serra

Resource group \* <sup>i</sup>  
Default-ActivityLogAlerts

Action group name *	Action Type *	Status
<input type="text" value="Unique name for the action"/>	<input type="text" value="Email/SMS/Push/Voice"/>	
<input type="text" value="Unique name for the action"/>	Automation Runbook	
<input type="text" value="Unique name for the action"/>	Azure Function	
<input type="text" value="Unique name for the action"/>	Email Azure Resource Manager Role	
<input type="text" value="Unique name for the action"/>	Email/SMS/Push/Voice	
<input type="text" value="Unique name for the action"/>	ITSM	
<input type="text" value="Unique name for the action"/>	LogicApp	
<input type="text" value="Unique name for the action"/>	Secure Webhook	
<input type="text" value="Unique name for the action"/>	Webhook	

### Email/SMS/Push/Voice

Add or edit an Email/SMS/Push/Voice action

Email  
Email

SMS (Carrier charges may apply)  
Country code

Phone number

Azure app Push Notifications  
Azure account email

Voice  
Country code

Phone number

Enable the common alert schema. [Learn more](#)

Yes  No

OK

**i** Have a consistent naming convention for action group names. Click on the details link for more information.

# Maintenance Schedule (preview)

Planned Maintenance Notification for Scheduled Maintenance to Azure SQL Data Warehouse



Microsoft Azure  
To James Serra

Reply

Reply All

Forward

Sat 7/18/2020

LinkedIn

Get more



## Planned Maintenance Notification for Scheduled Maintenance to Azure SQL Data Warehouse

The activity log alert **Azure Service Issue** was triggered for the Azure subscription **MTC NYC - James Serra**.

[View in Azure Service Health >](#)

TRACKING ID:  
**DM1Y-988**

TYPE:  
**Maintenance**

STATUS:  
**InProgress**

COMMUNICATION:

This notification is for previously scheduled maintenance to your Azure SQL Data Warehouse instance(s) in East US 2. Maintenance will start at 07:00 UTC on 18 Jul 2020 and expected to complete by 12:00 UTC on 18 Jul 2020. During maintenance, your Azure SQL Data Warehouse instance(s) may experience a brief drop in connectivity for a short number of seconds. Feel free to refer to your portal for more information (<https://aka.ms/servicehealthpm>) and for any additional questions, please [contact support](#).

Resource Group	Server Name	Database Name / Elastic Pool
workspacemanagedrg-cf13ef62-5454-4eac-a642-84c4087e758a	serrademoworkspace	AdventureWorksD

IMPACTED SERVICE(S) AND REGION(S)

Service Name	Region
SQL Data Warehouse	East US 2

You are receiving this alert notification from Microsoft Azure as a member of the 'SerraEmail' action group. To unsubscribe from emails directed to this action group [click here](#).

# Automatic statistics management

## Overview

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

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

Statistics are considered out of date when:

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

-- Turn on/off auto-create statistics settings

```
ALTER DATABASE {database_name}
SET AUTO_CREATE_STATISTICS { ON | OFF }
```

-- Turn on/off auto-update statistics settings

```
ALTER DATABASE {database_name}
SET AUTO_UPDATE_STATISTICS { ON | OFF }
```

-- Configure synchronous/asynchronous update

```
ALTER DATABASE {database_name}
SET AUTO_UPDATE_STATISTICS_ASYNC { ON | OFF }
```

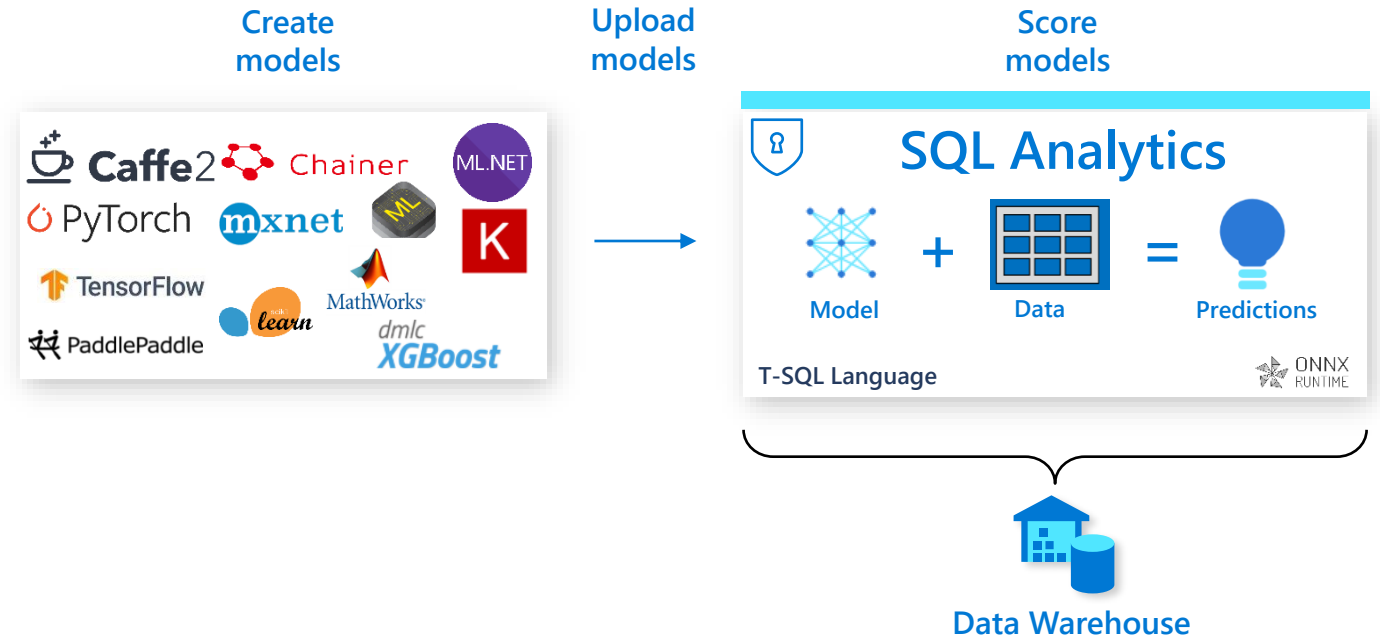
-- Check statistics settings for a database

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

# Machine Learning enabled DW

## Native PREDICT-ion

- T-SQL based experience (interactive./batch scoring)
- Interoperability with other models built elsewhere
- Execute scoring where the data lives



```
--T-SQL syntax for scoring data in SQL DW
SELECT d.*, p.Score
FROM PREDICT(MODEL = @onnx_model, DATA = dbo.mytable AS d)
WITH (Score float) AS p;
```

# SQL Pool: Snapshots and restores

## Overview

Auto backups (snapshots), at least every 8 hours (usually every 4 hours)

Taken throughout the day, or triggered manually via REST API, PowerShell or Azure Portal.

Available for up to 7 days, even after data warehouse deletion. 8-hour RPO (data loss) for restores from snapshot

42 user-defined restore points (manually triggered snapshots).

Regional restore in under 20 minutes, no matter data size.

Snapshots and geo-backups allow cross-region restores.

Automatic snapshots and geo-backups on by default.

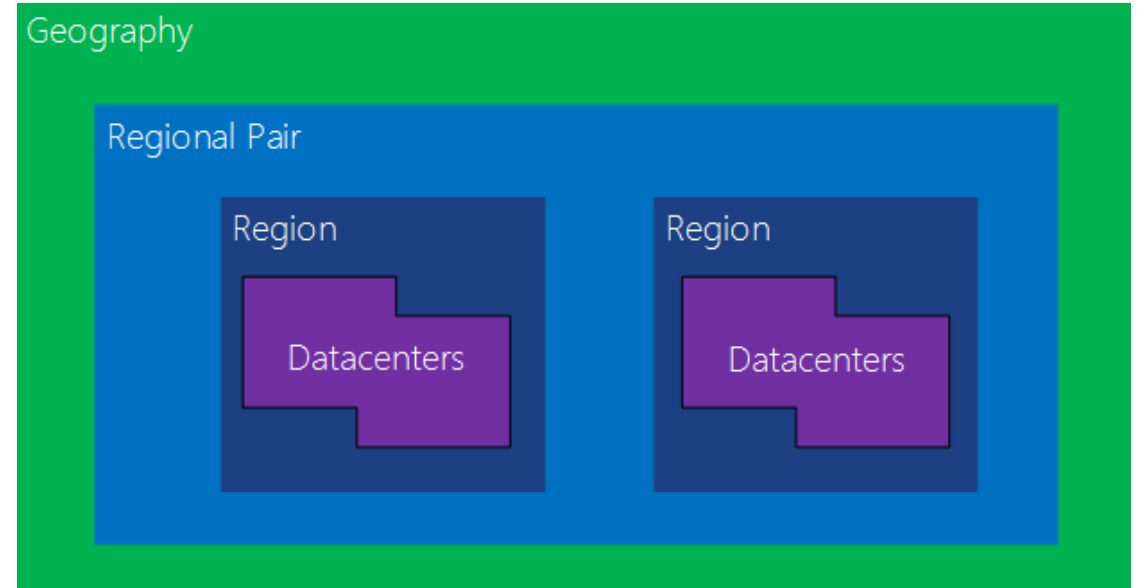
## Benefits

Snapshots protect against data corruption and deletion.

Restore to quickly create dev/test copies of data.

Manual snapshots protect large modifications.

Geo-backup copies one of the automatic snapshots each day to RA-GRS storage. This can be used in an event of a disaster to recover your Azure Synapse Analytics to a new region. 24-hour RPO for a geo-restore



--View most recent snapshot time

```
SELECT top 1 *  
FROM sys.pdw_loader_backup_runs  
ORDER BY run_id DESC;
```

# Power BI Aggregations and Synapse query performance

