Azure Synapse Analytics

A data lakehouse

Ganesh R

Azure Data Engineer

Agenda

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

Multipleanalytics platforms

Big Data

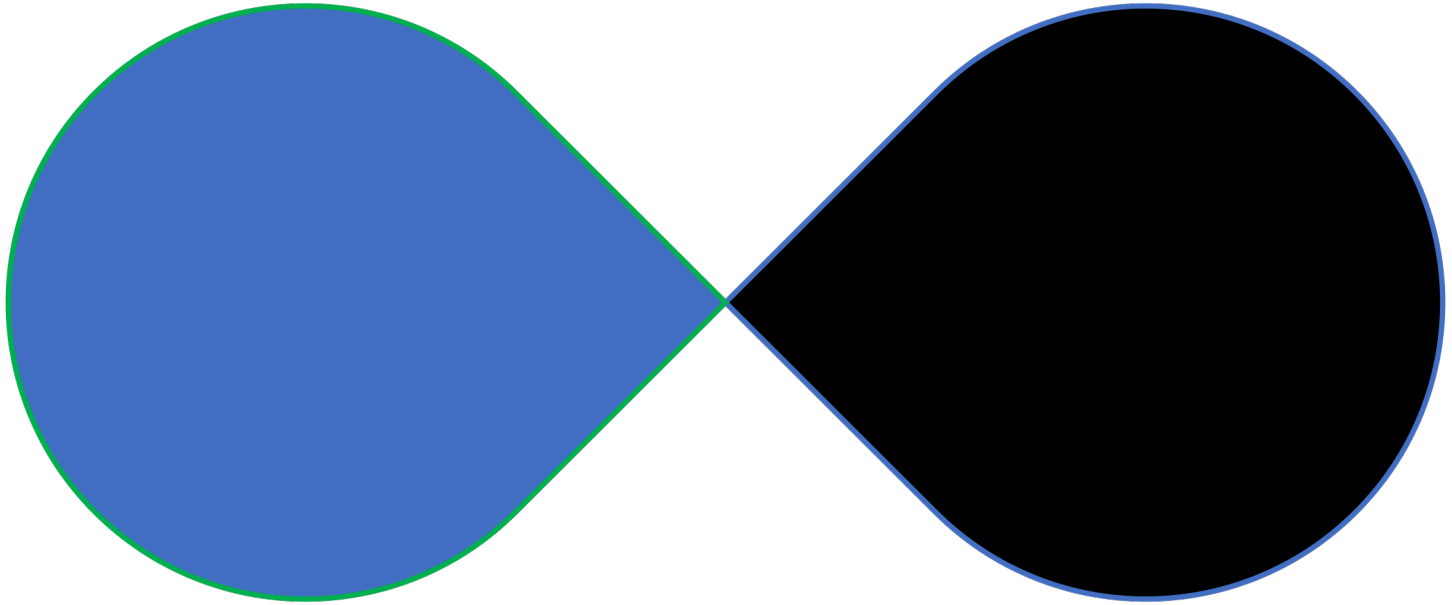
Data Lake

OR

Relational Data

Data Warehouse

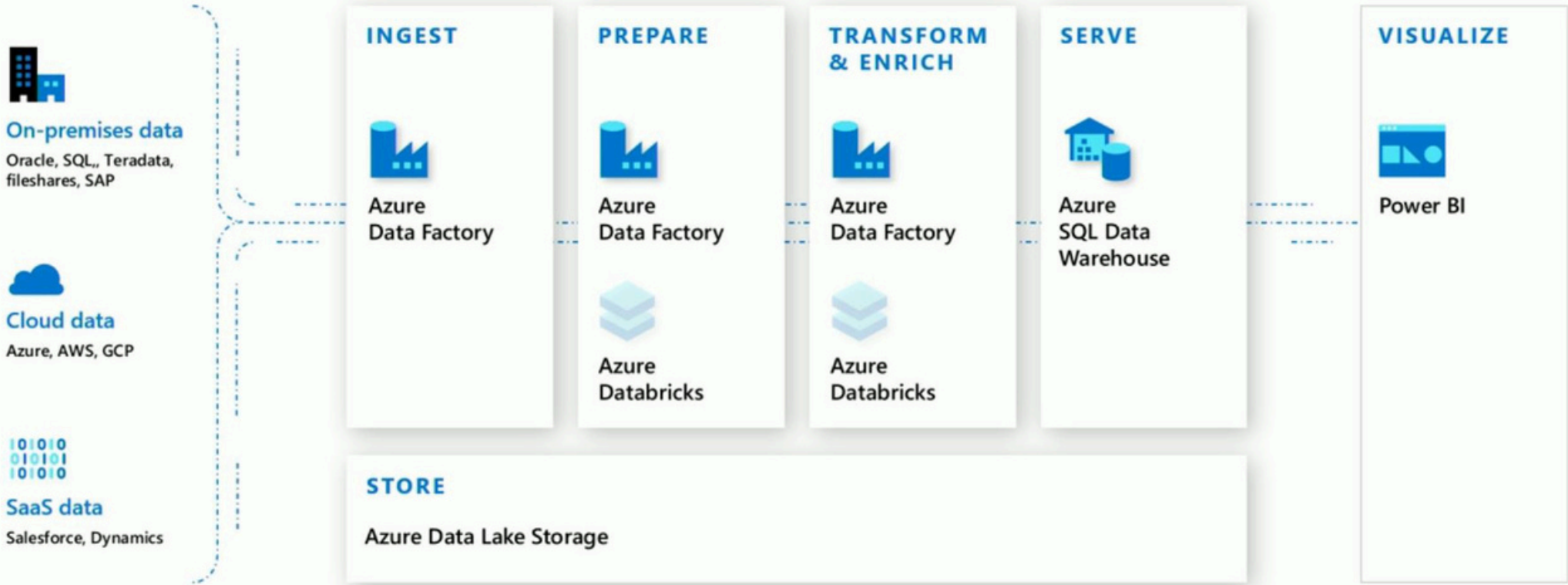
Azure brings these two worlds together



Welcome to Azure Synapse Analytics

Data warehousing & big data analytics—*Data Lakehouse*

Modern Data Warehouse



Azure Synapse Analytics - *Data Lakehouse*

Synapse Analytics (GA)

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

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

Synapse Analytics (PREVIEW)
"PP"

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

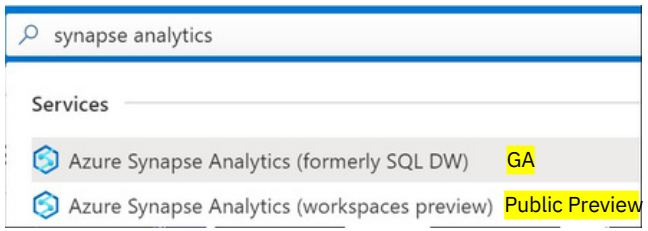
• DeltaLake support
- CDM support
- External table wizard

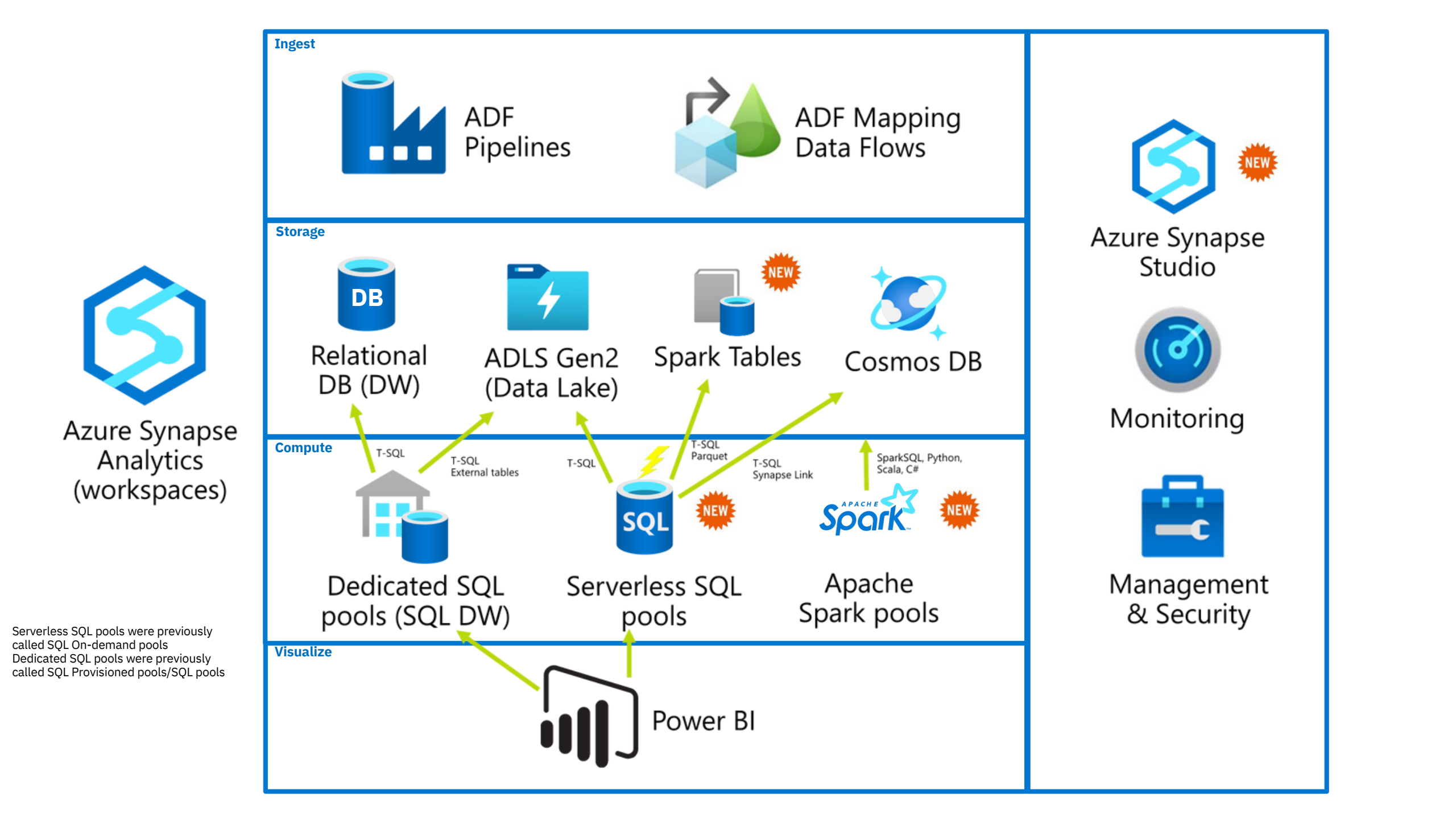
SQL ANALYTICS

APACHE SPARK

STUDIO

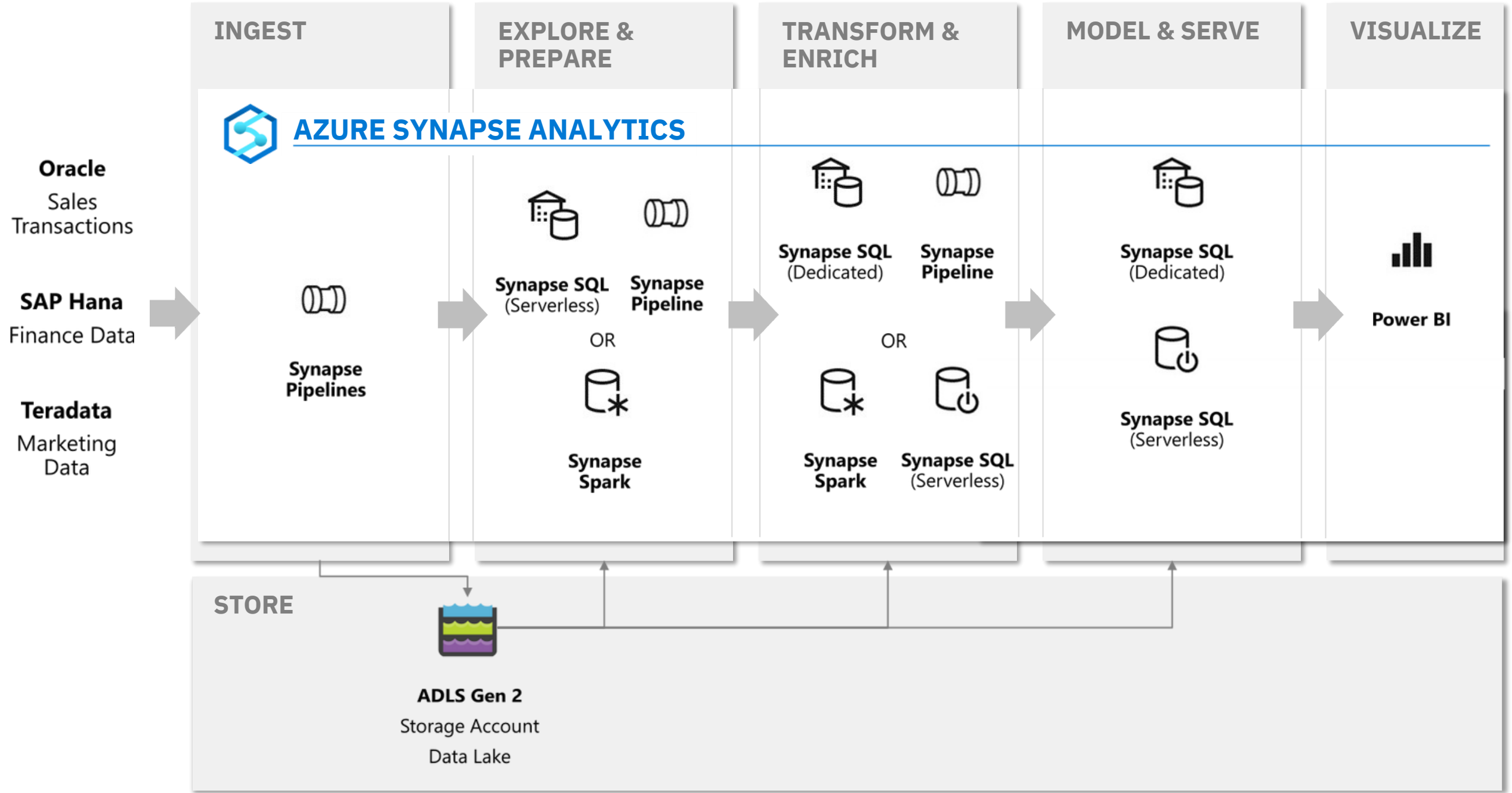
DATA INTEGRATION





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

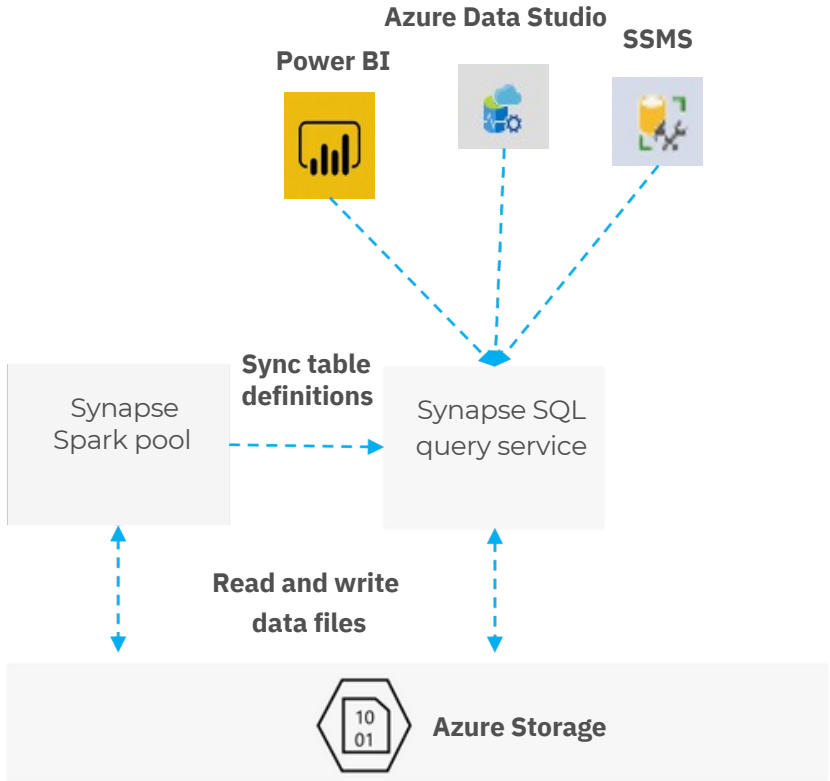
-Logical Data Warehouse on top of Azure storage

-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

Microsoft

Azure SynapseAnalytics

Massively Parallel Processing (MPP)

Intro

For Dedicated SQL pools

Parallelism

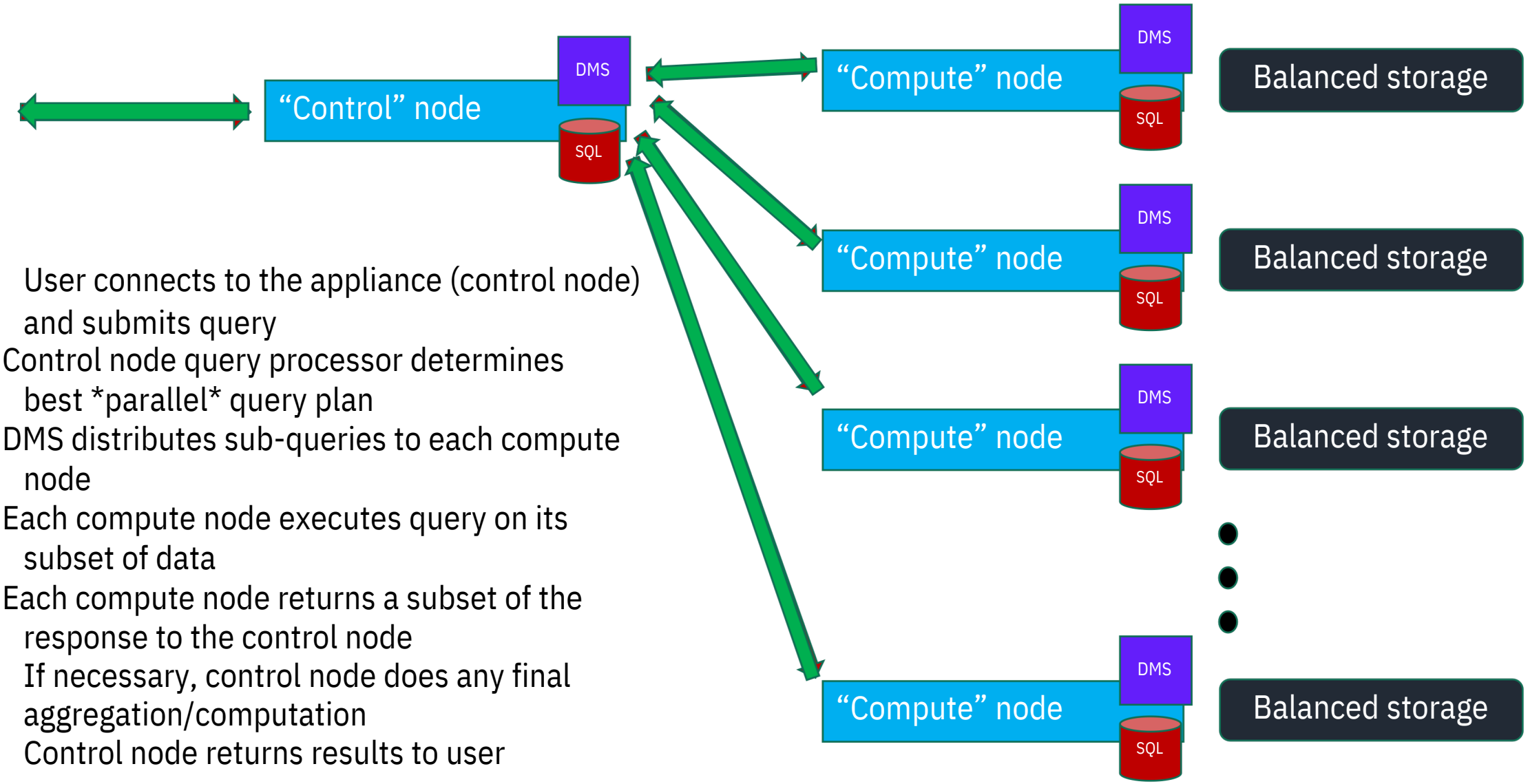
SMP -Symmetric Multiprocessing

- 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

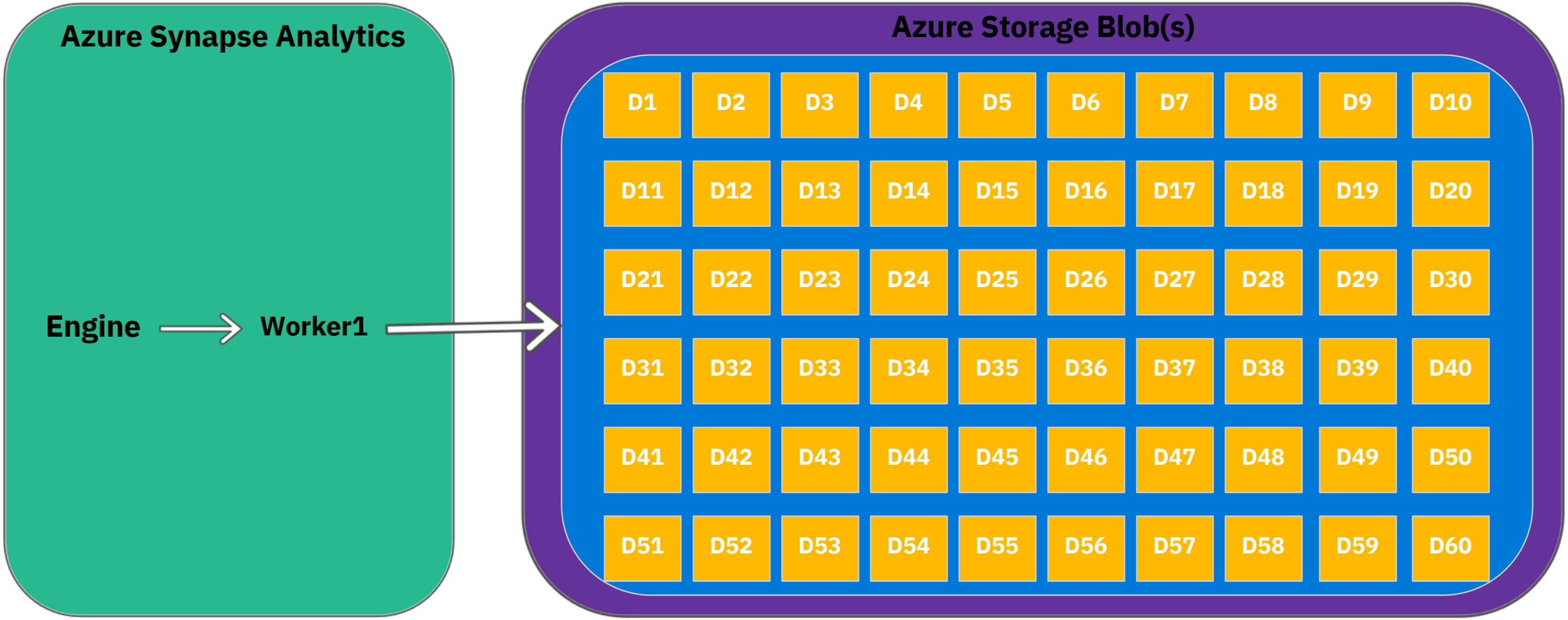
```
ALTER DATABASE ContosoDW MODIFY
(service_objective = 'DW1000');
```

CPU

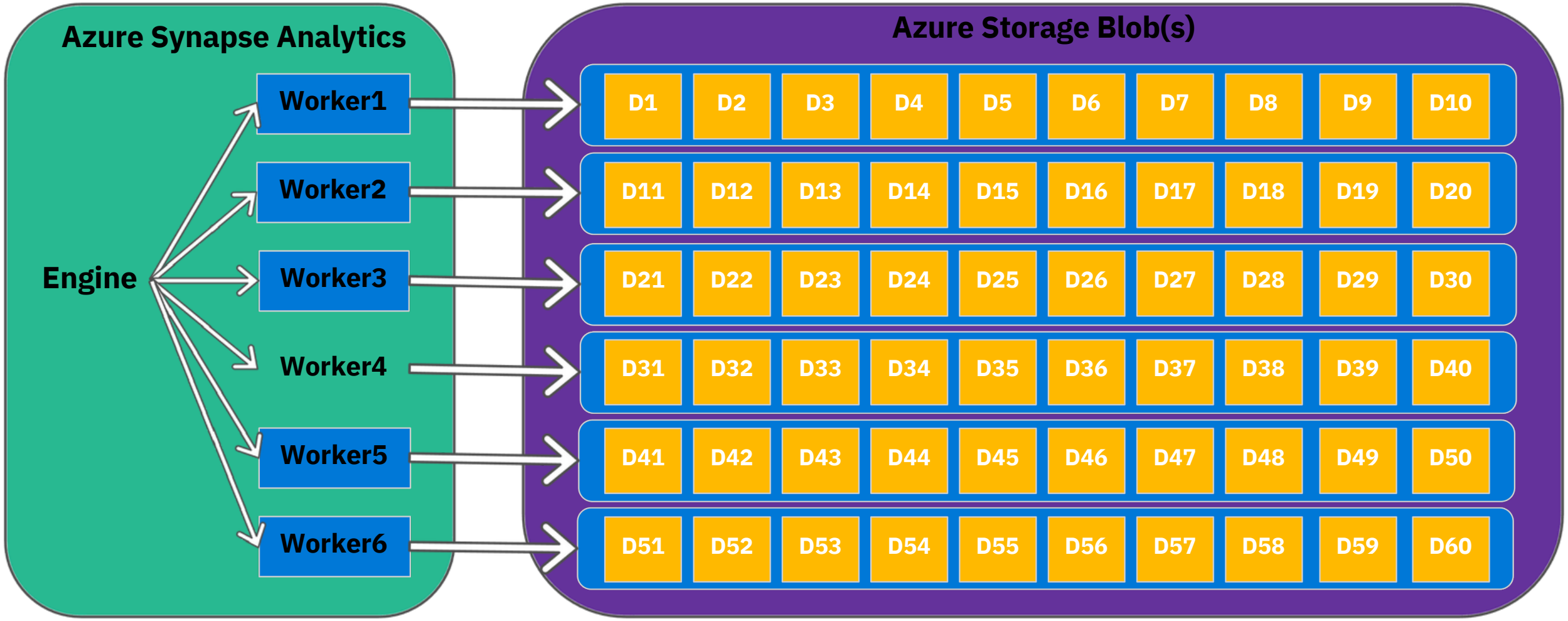
RAM

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				
Serverless SQL pool	Y	Y (external table)	*Soon (parquet format)*	*N*
Apache Spark pool	*N*	Y	Y (parquet format)	Y (Synapse link)
	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)

https://www.jamesserra.com/archive/2020/09/query-options-in-azure-synapse-analytics/

Query Demo with Azure Synapse Studio

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

https://www.jamesserra.com/archive/2020/09/query-options-in-azure-synapse-analytics/

Query Demo with Azure Synapse Studio

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

https://www.jamesserra.com/archive/2020/09/query-options-in-azure-synapse-analytics/

Query Demo with Azure Synapse Studio

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

https://www.jamesserra.com/archive/2020/09/query-options-in-azure-synapse-analytics/

Query Demo with Azure Synapse Studio

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

https://www.jamesserra.com/archive/2020/09/query-options-in-azure-synapse-analytics/

Query Demo with Azure Synapse Studio

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

https://www.jamesserra.com/archive/2020/09/query-options-in-azure-synapse-analytics/

Query Demo with Azure Synapse Studio

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

https://www.jamesserra.com/archive/2020/09/query-options-in-azure-synapse-analytics/

Query Demo with Azure Synapse Studio

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

https://www.jamesserra.com/archive/2020/09/query-options-in-azure-synapse-analytics/

Query Demo with Azure Synapse Studio

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

https://www.jamesserra.com/archive/2020/09/query-options-in-azure-synapse-analytics/

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.

https://www.jamesserra.com/archive/2020/09/query-options-in-azure-synapse-analytics/

DEMO

Query 30 billion rows

DEMO

COPY statement

DEMO

Spark notebooks

DEMO

Power BI

DEMO

SSMS

DEMO

Monitor

DEMO

Synapse Samples

DEMO

Federated Query

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

Microsoft

Azure SynapseAnalytics

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) isa 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.

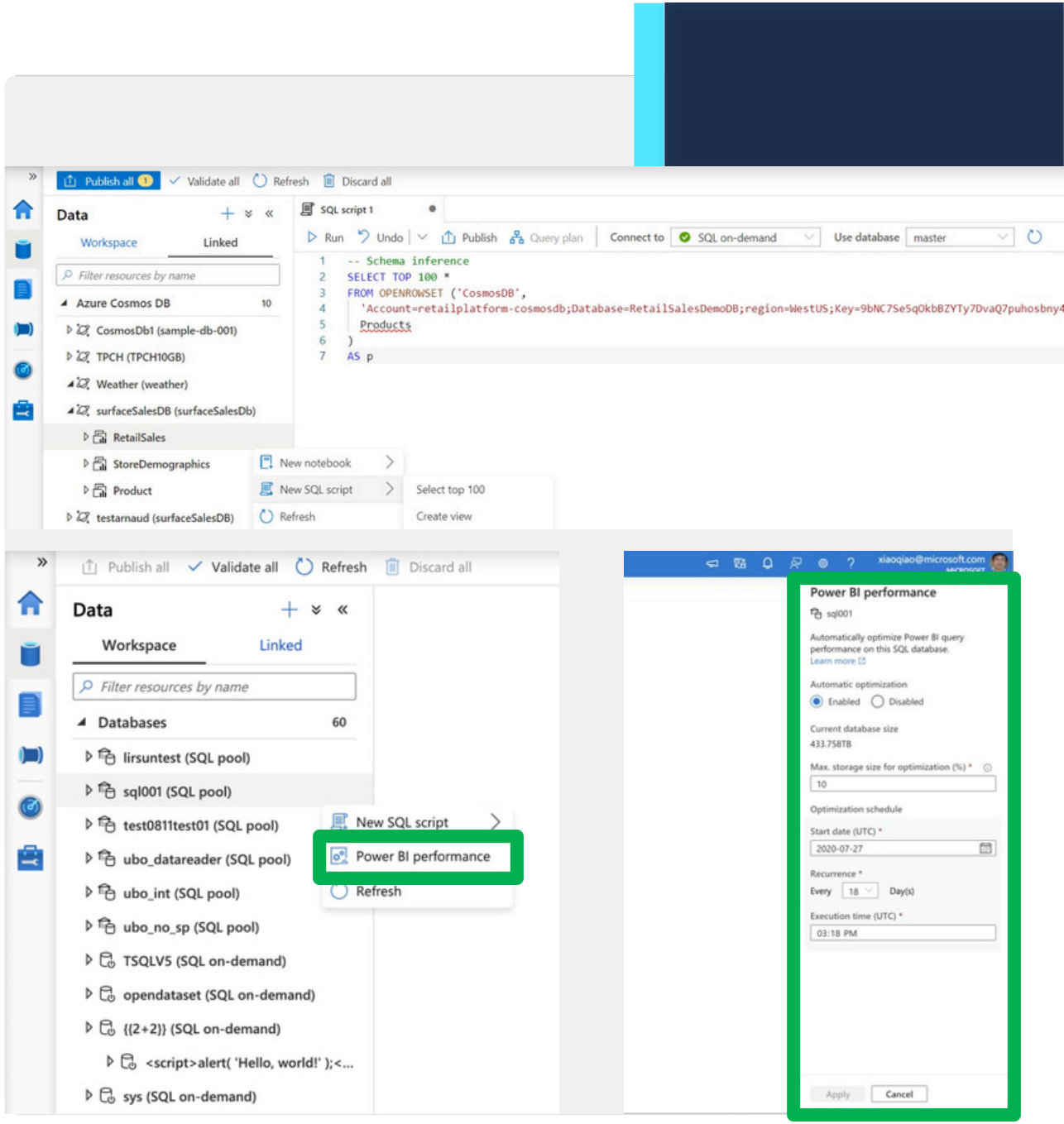
Column-level Encryption (public preview): With CLE, customers gain the ability to use different protection keys for different columns in a table, with each key having its own access permissions.

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



Microsoft

Azure Synapse Analytics

Data Storage and Performance Optimizations

Database Tables

Columnar Storage

Columnar Ordering

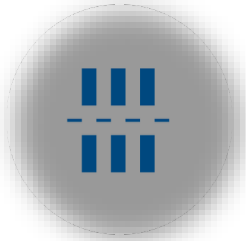


Table Partitioning

Hash Distribution

Nonclustered Indexes

Optimized Storage

Reduce Migration Risk

Less Data Scanned

Smaller Cache Required

Smaller Clusters

Faster Queries

Tables –Indexes

Clustered Columnstoreindex (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

Nonclusteredindexes (Secondary)

Enable ordering of multiple columns in a table Allows multiple nonclusteredon 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  INTNOT NULL,
    Date     DATENOT NULL,
    Name     VARCHAR(2),
    Country VARCHAR(2)
)
WITH
(
    CLUSTEREDCOLUMNSTOREINDEX |
     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

Over view

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

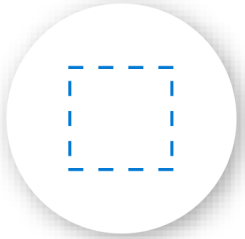
Improves efficiency and performance of loading and querying by limiting the scope to subset of data. Offers significant query performance enhancements where filtering on the partition key can eliminate unnecessary scans and eliminate IO.

```
CREATE TABLE partitionedOrderTable
(
    OrderId  INTNOT NULL,
    Date     DATENOT NULL,
    Name     VARCHAR(2),
    Country VARCHAR(2)
)
WITH
(
    CLUSTERED COLUMNSTORE INDEX,
    DISTRIBUTION = HASH([OrderId]),
    PARTITION (
    [Date] RANGERIGHTFORVALUES(
    '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
Fact	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. Operations that benefit: COUNT(DISTINCT(<hashed_key>)) OVER PARTITION BY <hashed_key> most JOIN <table_name> ON <hashed_key> GROUP BY <hashed_key>
Dimension	Use replicated for smaller tables. If tables are too large to store on each Compute node, use hash-distributed.
Staging	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.

Database Views



Views

Materialized Views

Scope: **Generally Available**

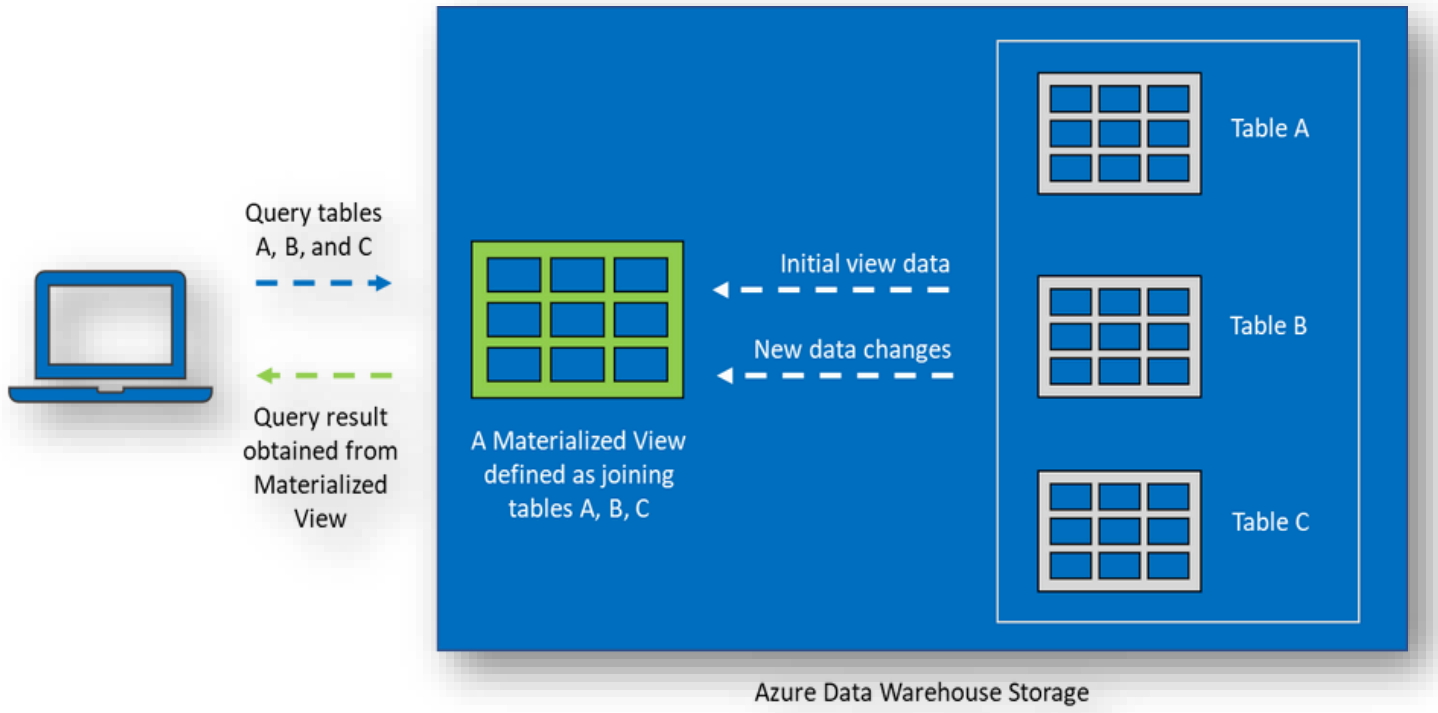
Best in class price performance

Interactive dashboarding with Materialized Views

-Automatic data refresh and maintenance

-Automatic query rewrites to improve performance

-Built-in advisor



Materialized views

Over view

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***

```
--Create indexed view
CREATEMATERIALIZED VIEWSales.vw_Orders
WITH
(
  DISTRIBUTION = ROUND_ROBIN |
  HASH(ProductID)
)
AS
   SELECTSUM(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_OrdersDISABLE;
--Re-enable index view by rebuilding it
ALTER INDEX ALL ON Sales.vw_OrdersREBUILD;
```

Benefits

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

High availability and resiliency as regular tables

COPY command

Over view

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**

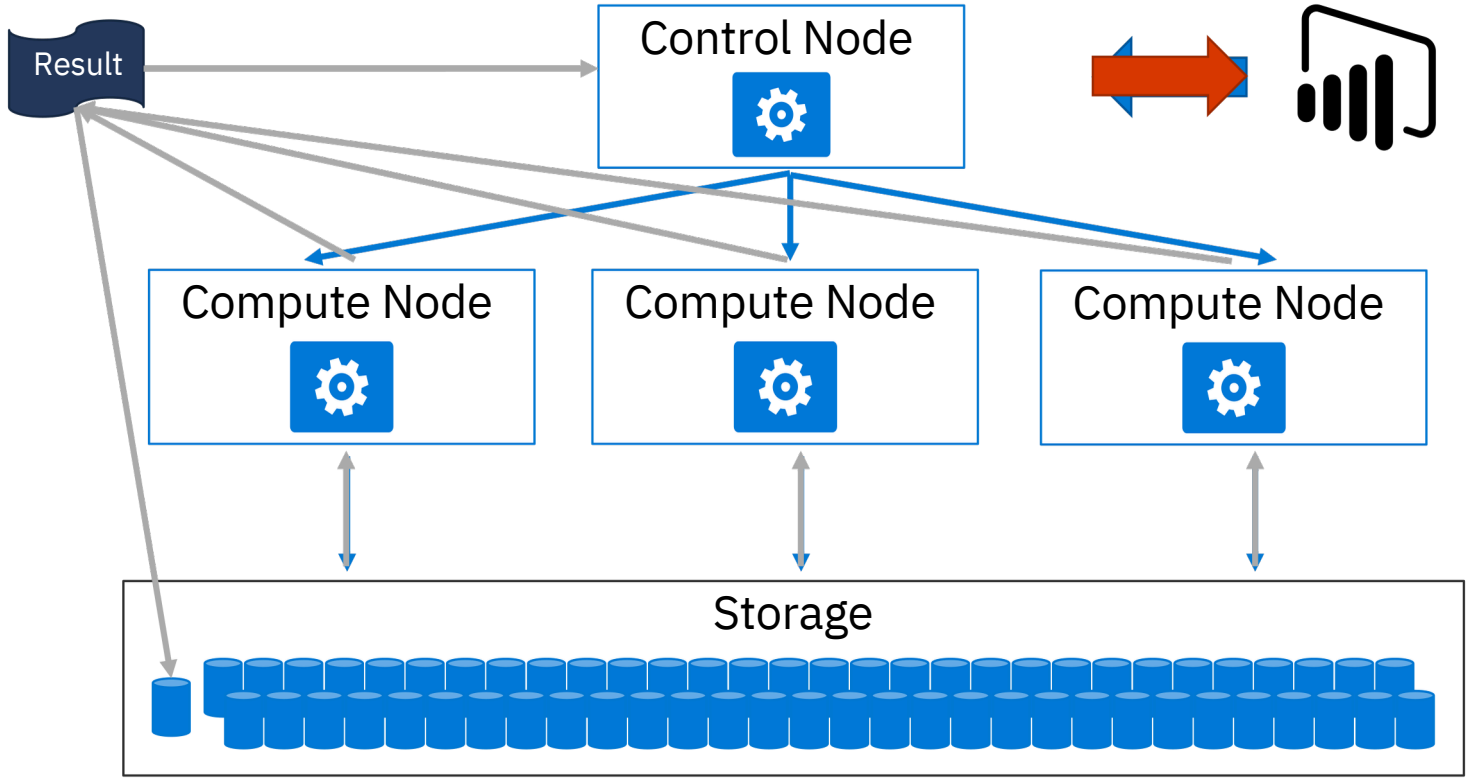
Best in class price performance

Interactive dashboarding with Resultset Caching

-Millisecondresponses with resultsetcaching

-Cache survives pause/resume/scale operations

-Fully managed cache (1TB in size)



Enable caching: **Alter Database** <DBNAME> **Set Result_Set_CachingON**

Purge cache: **DBCC DropResultSetCache**

Result-set caching

Over view

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

Over view

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

```sql
/* 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 Memor y	Max. Concurrent Queries
smallrc	3% 10% 22%	32 10 4 1
mediumrc	70%	
largerc		
xlargerc		

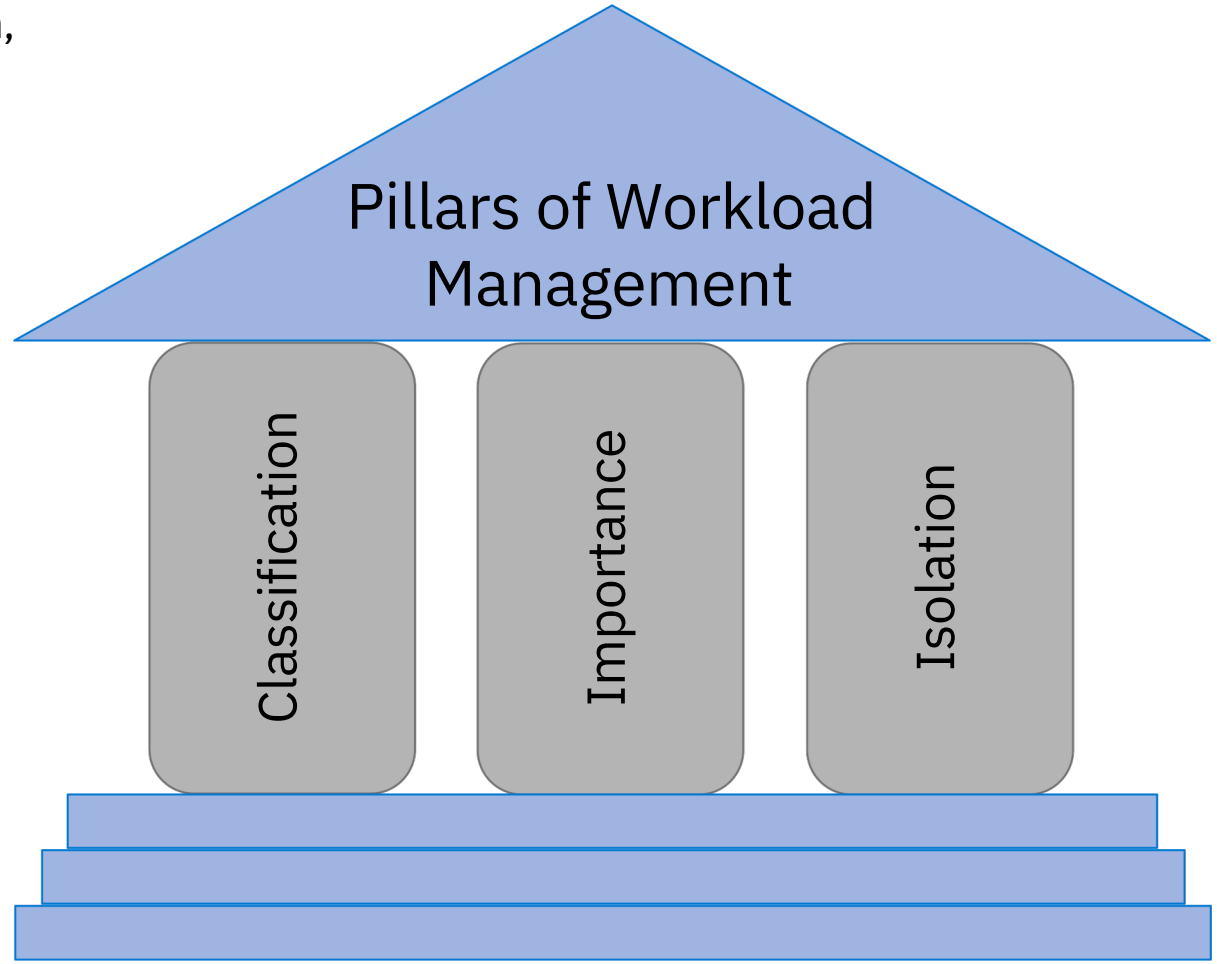
Workload Management

Over view

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

Over view

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.

```
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 NORMALvalue ]
  [ [ , ] IMPORTANCE = {LOW | BELOW_NORMAL |          ABOVE_NORMAL | HIGH} ]
  [ [ , ] QUERY_EXECUTION_TIMEOUT_SEC = value ]
)[ ; ]
```

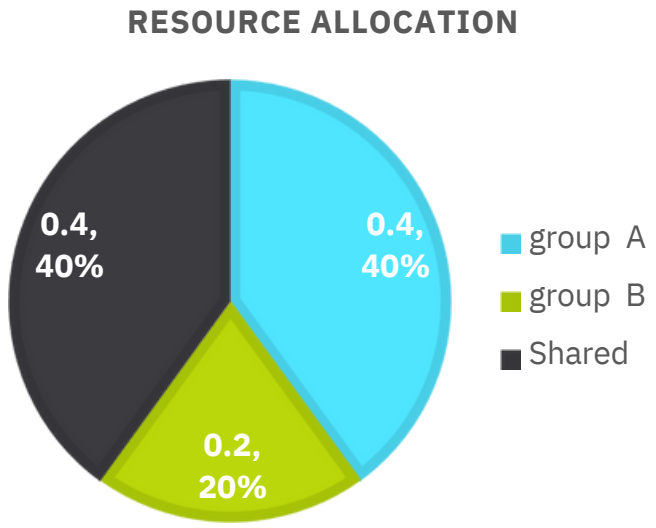
Benefits

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.

Workload importance

Over view

Queries past the concurrency limit enter a FiFoqueue

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

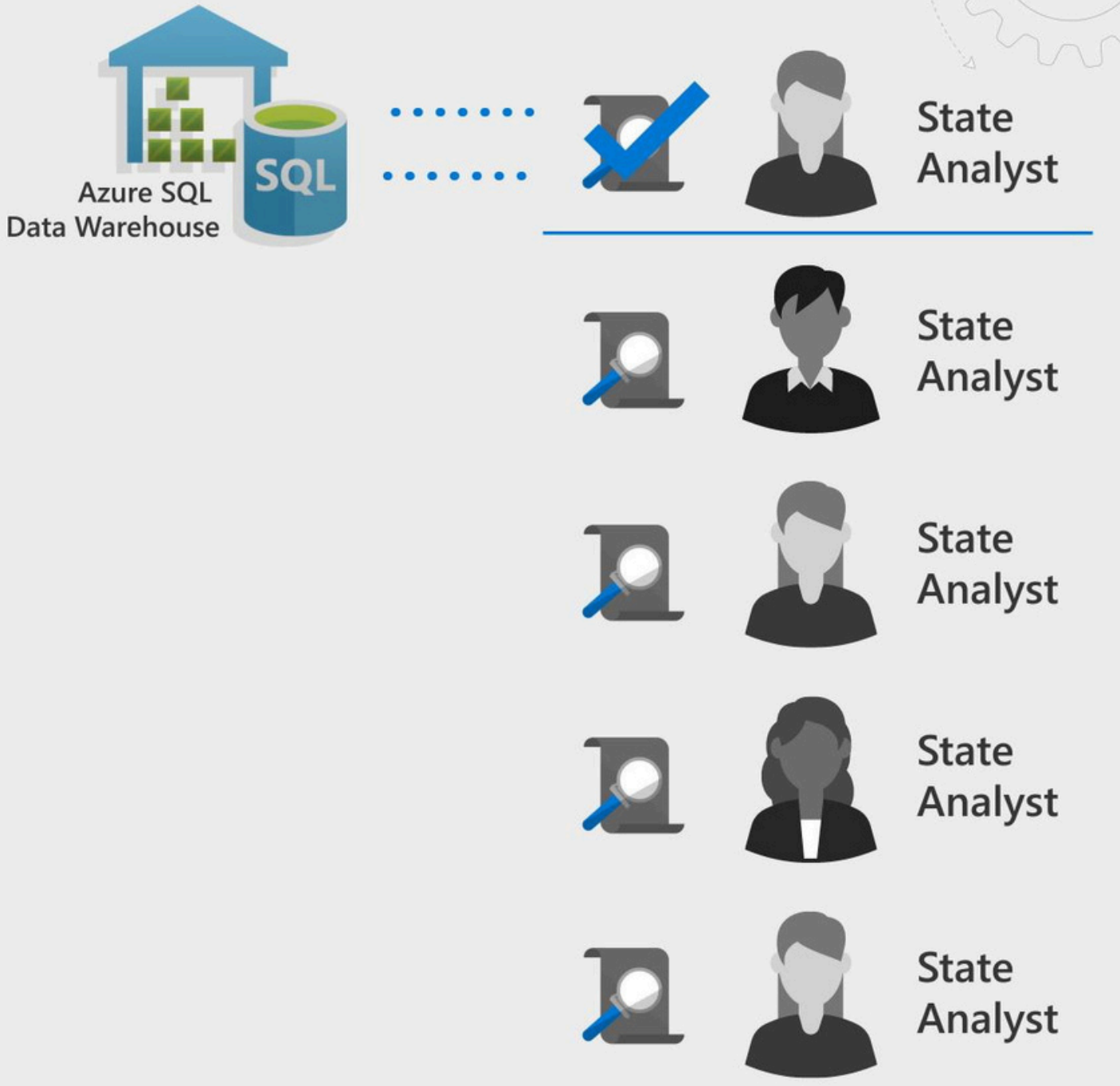
State analysts have normal importance.

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

Over view

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 (smallrcresource 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)

Over view

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

Over view

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

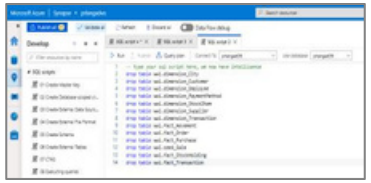
```
-- List all open sessions
SELECT * FROM sys.dm_pdw_exec_sessions where status <> 'Closed'
and session_id <> session_id();
```

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

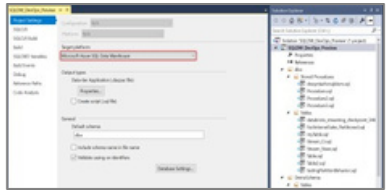


Azure Cloud Service

Offers end-to-end lifecycle for analytics

Connects to multiple ser vices

Visual Studio -SSDT database projects



Runs on Windows

Create, maintain database code, compile, code refactoring

Azure Data Studio

Runs on Windows, Linux, macOS

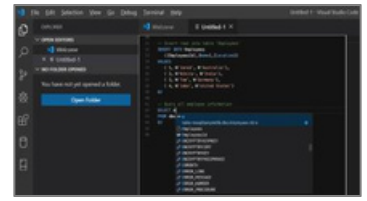
Light weight editor, (queries and extensions)

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

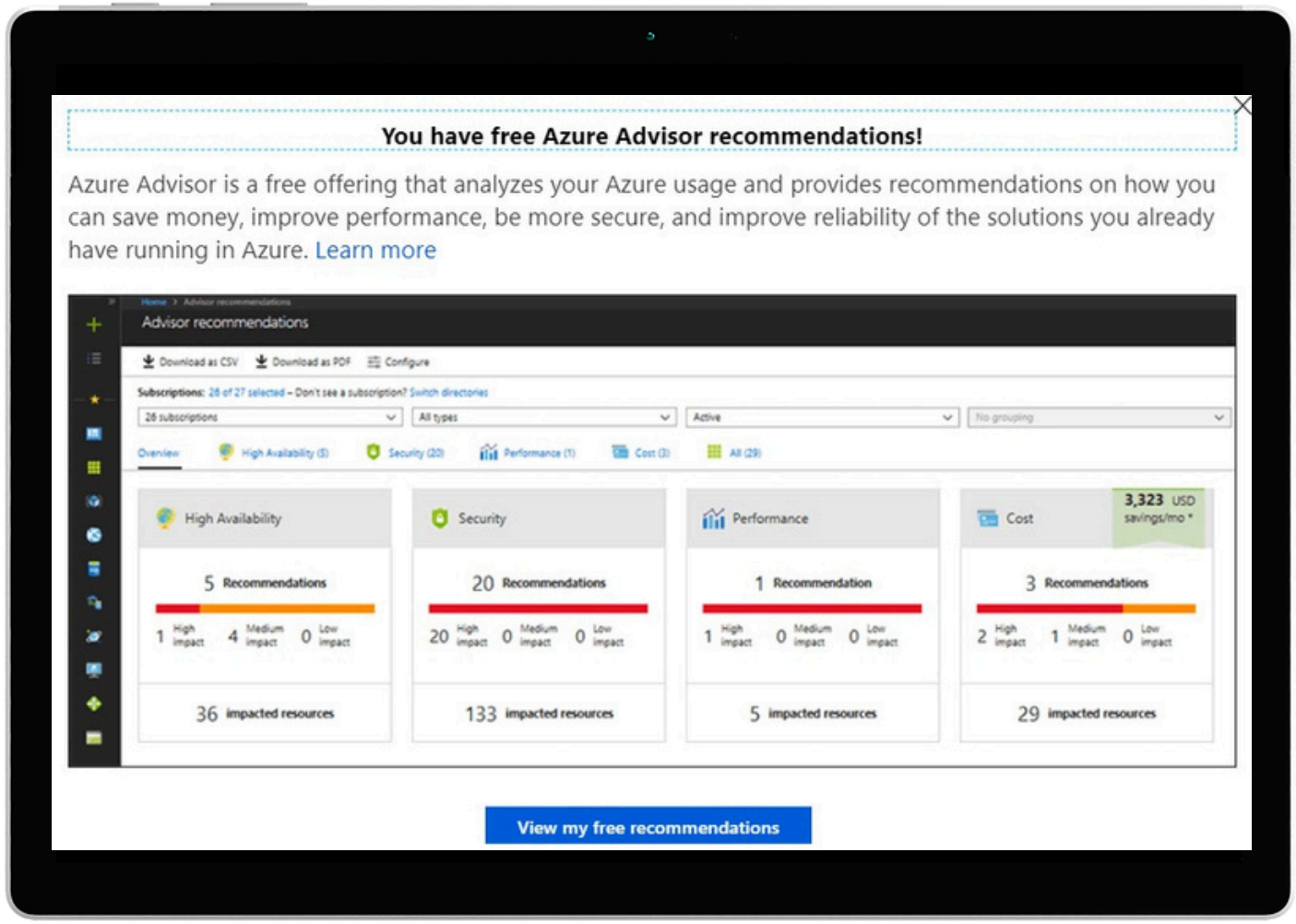
Provision additional capacity

TempdbContention

Scale or update user resource class

Suboptimal Plan Selection

Create or update table statistics



Maintenance schedule (preview)

Over view

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)

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

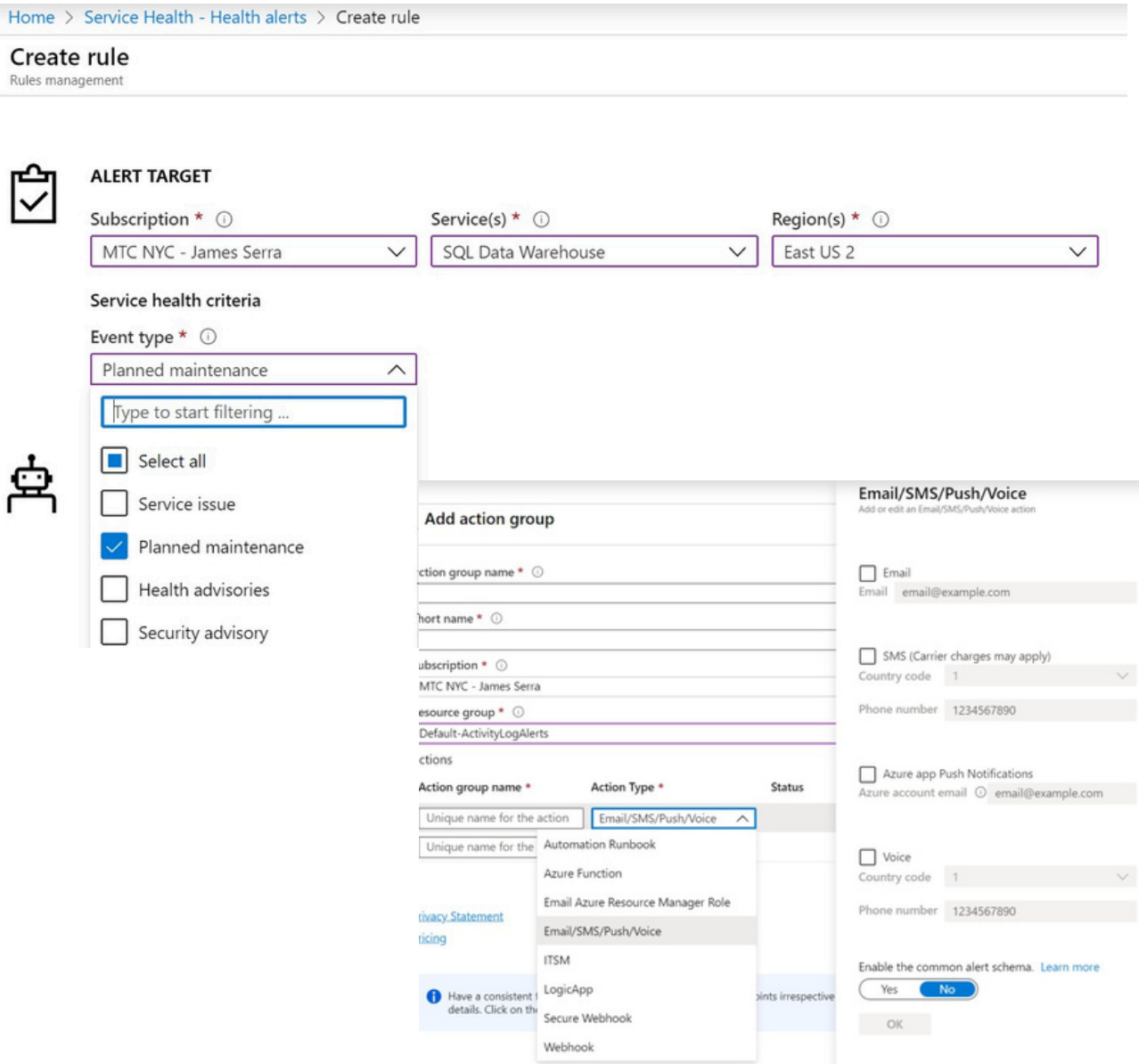
Choose primary window

(•) Saturday - Sunday ( ) Tuesday - Thursday

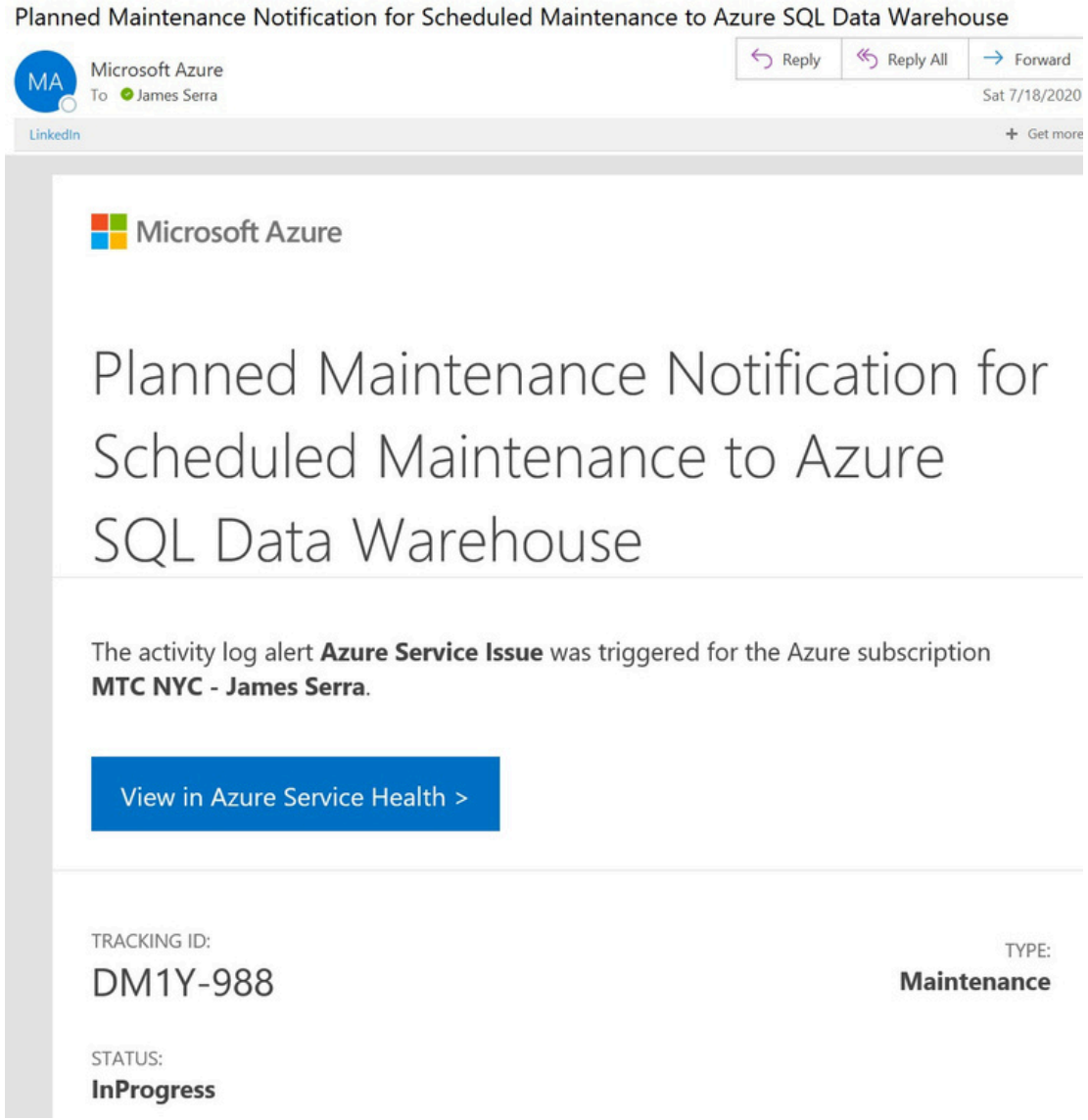
Primary maintenance window	Secondary maintenance window
Day: Saturday	Day: Wednesday
Start time: 05:00 UTC	Start time: 00:00 UTC
Time window: 7 hours	Time window: 8 hours

Schedule summary

Primary maintenance window	Secondary maintenance window
Saturday 05:00 UTC (7 hours)	**Wednesday** 00:00 UTC (8 hours)



Maintenance Schedule (preview)



Planned Maintenance Notification for Scheduled Maintenance to Azure SQL Data Warehouse

MA Microsoft Azure
To James Serra

Reply | Reply All | Forward

Sat 7/18/2020

LinkedIn + Get more

Microsoft Azure

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

Over view

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

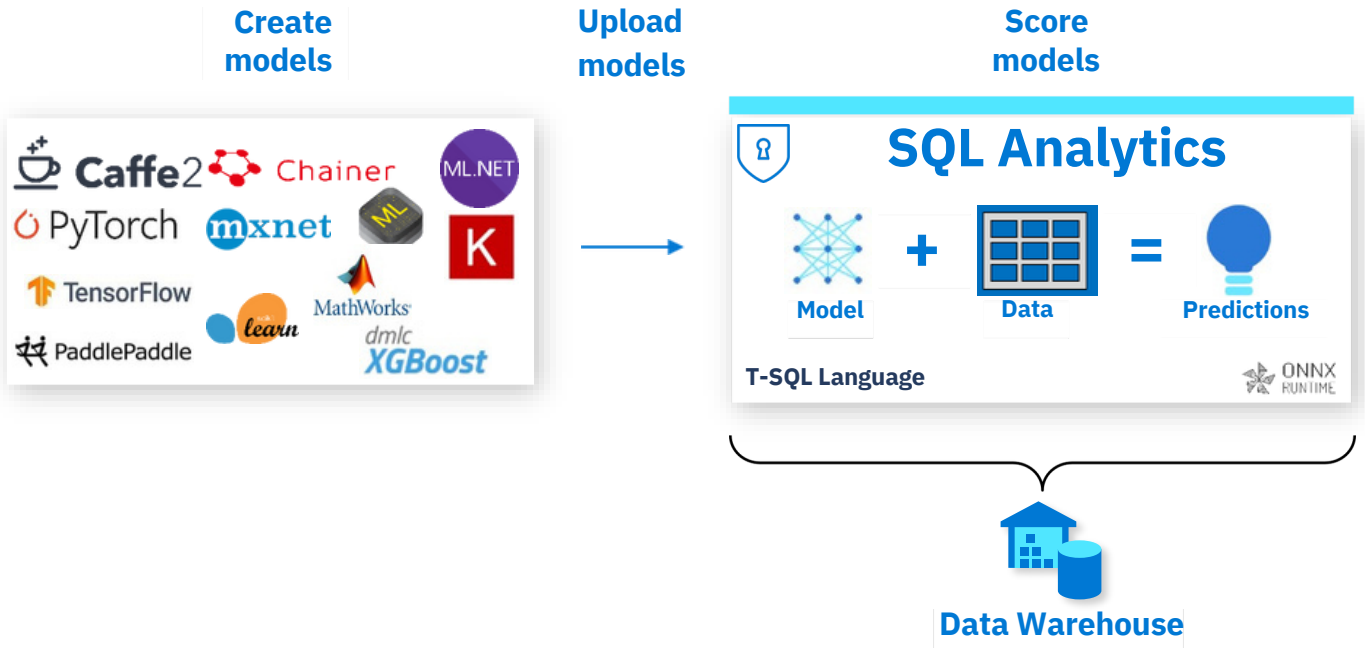
```
--Turn on/off auto-create statistics settings
ALTER DATABASE {database_name}
SET AUTO_CREATE_STATISTICS { ON | OFF }

--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
```

Scope: **Private Preview (whitelisting needed)**

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
SELECTd.*, p.Score
FROMPREDICT(MODEL= @onnx_model, DATA= dbo.mytableASd)
WITH(Score float) ASp;
```

SQL Pool: Snapshots and restores

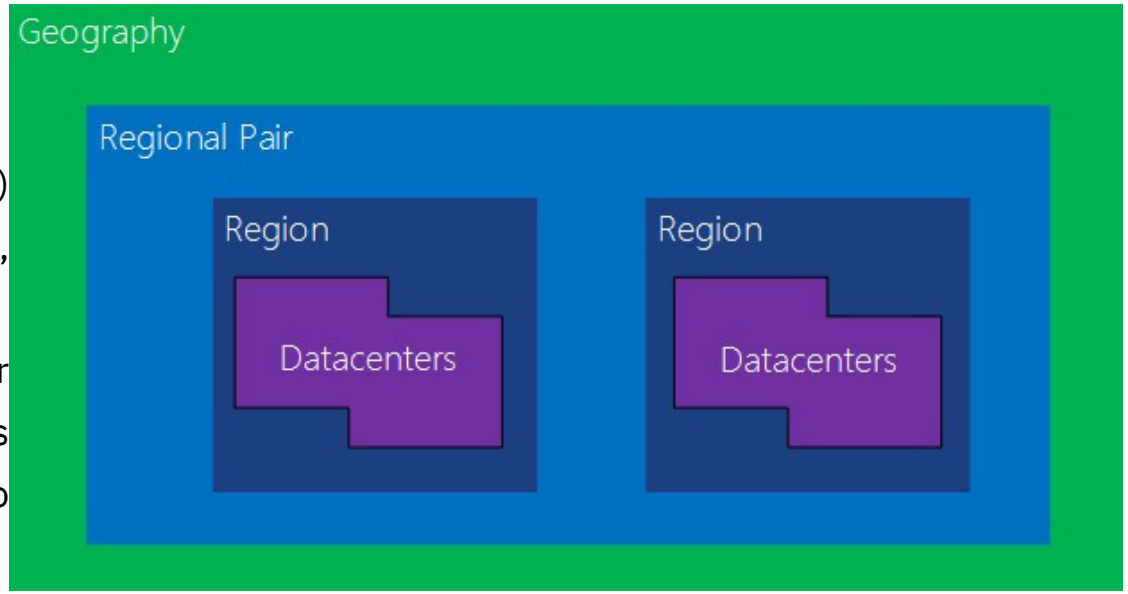
Over view

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.

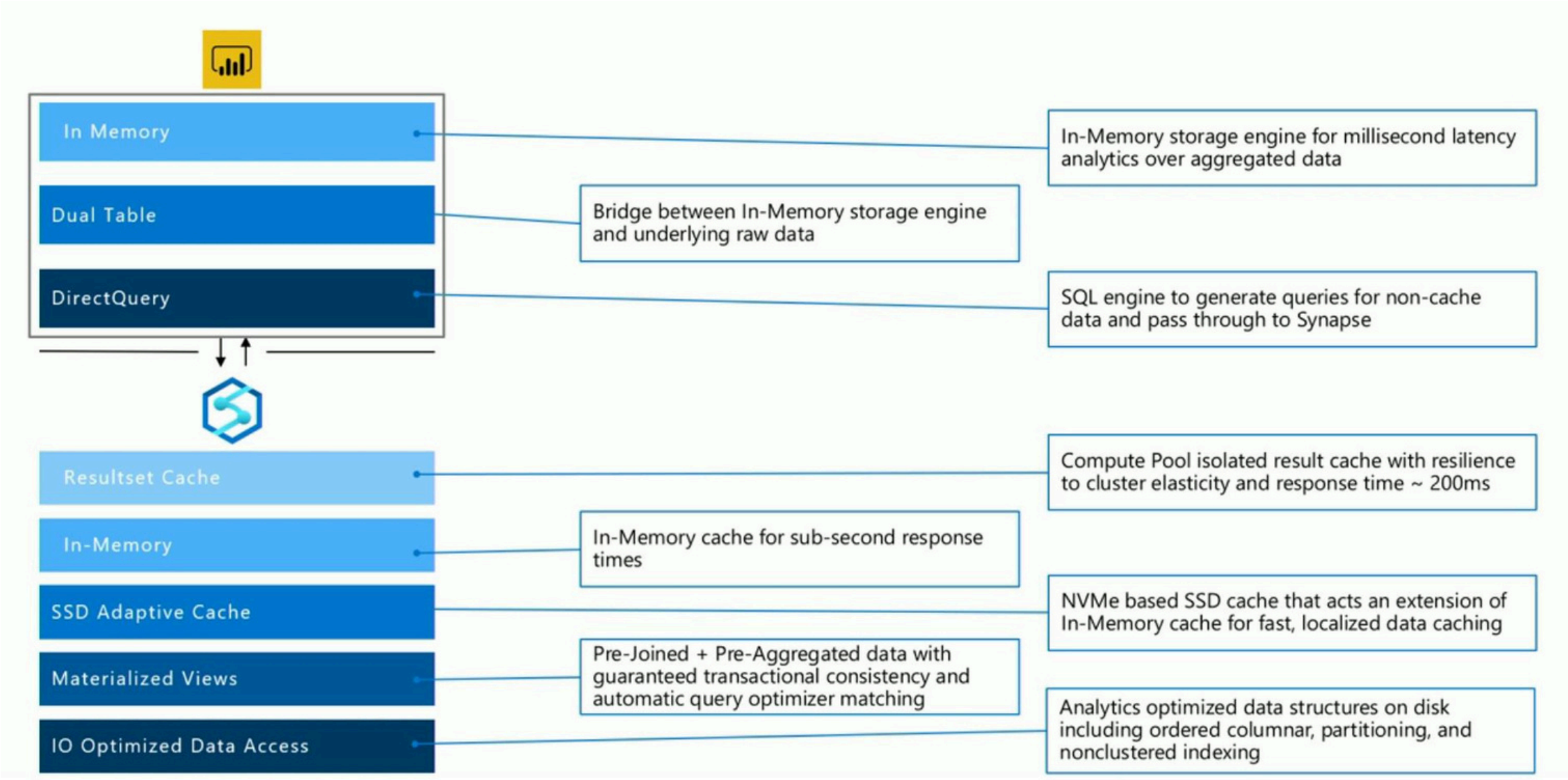


Benefits

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
SELECTtop1 *
FROM sys.pdw_loader_backup_runs
ORDER BY run_id DESC;
```

Power BI Aggregations and Synapse query performance



Follow for more content like this Azure Cloud for Data Engineering

Ganesh R

Azure Data Engineer

https://www.linkedin.com/in/rganesh0203/