

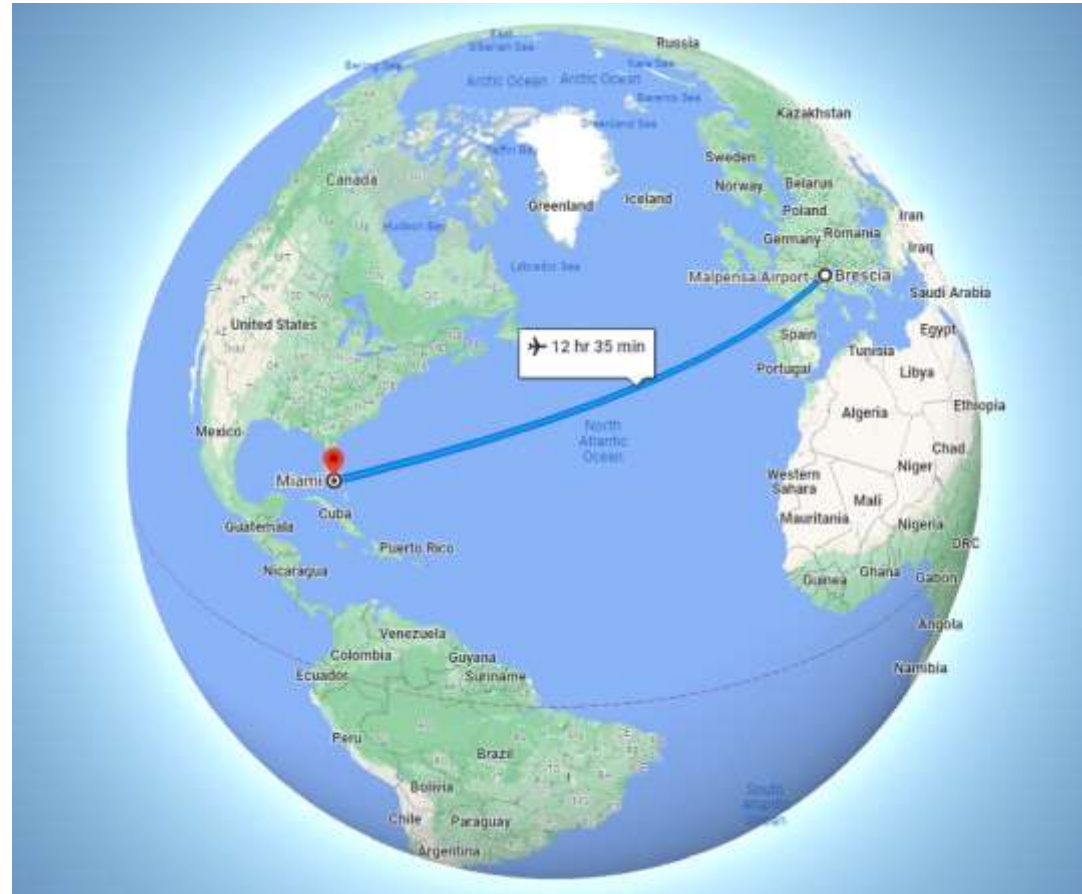


Azure Synapse Analytics SQL on-demand and Power BI

Azure Synapse Analytics SQL serverless pool & Power BI

Andrea Benedetti

Sr. Cloud Architect / Data & AI Engineer | Microsoft Italy



```
Declare @source_Brescia_Italy geography = geography::Point(45.541553, 10.211802, 4326);  
Declare @destination_Miami_Florida geography= geography::Point(25.761681, -80.191788, 4326);  
  
Select @source_Brescia_Italy.STDistance(@destination_Miami_Florida) / 1000 as Kilometers  
Select @source_Brescia_Italy.STDistance(@destination_Miami_Florida) / 1609.344 as Miles
```

100 %

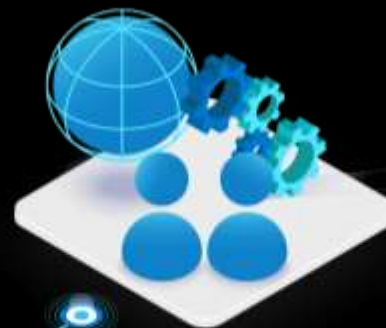
Results Messages

Kilometers	
1	8042.87736702563
Miles	
1	4997.61229856738

Azure Data Share



Ecosystem



Azure Synapse Analytics

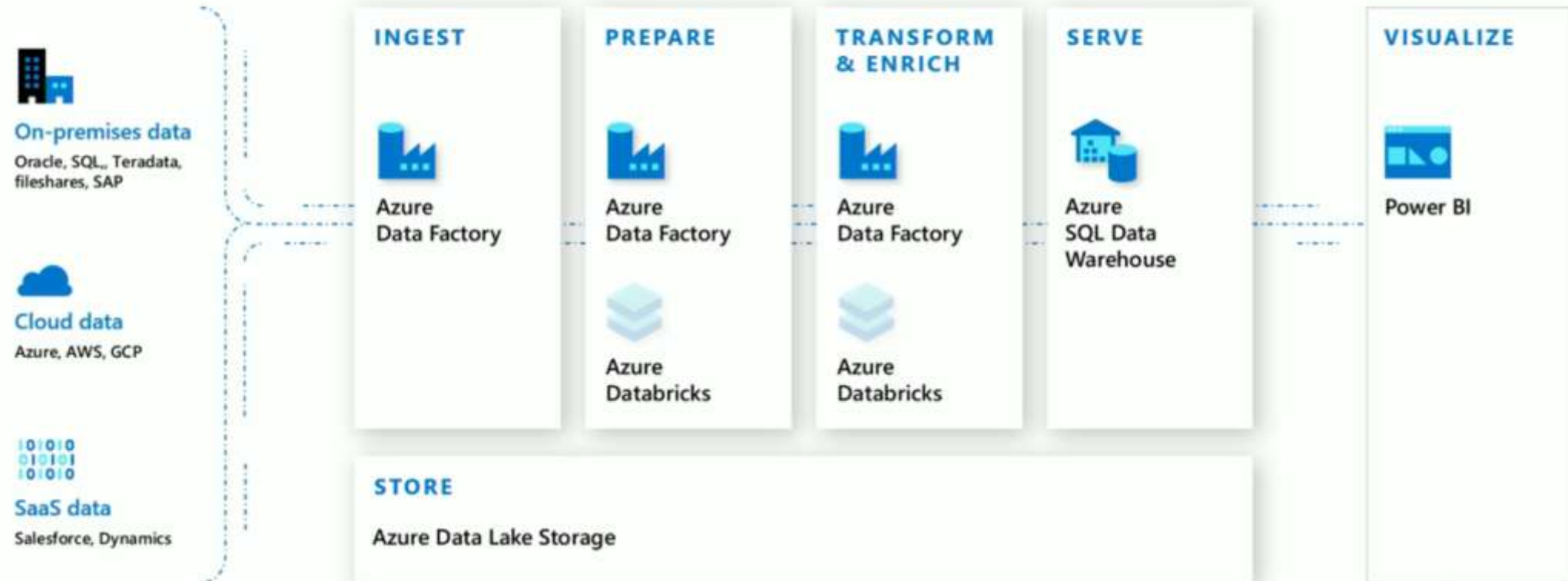


Power BI



Azure Machine Learning

Modern Data Warehouse



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

Businesses are forced to maintain two critical, yet independent analytics systems

Data science



Data lake

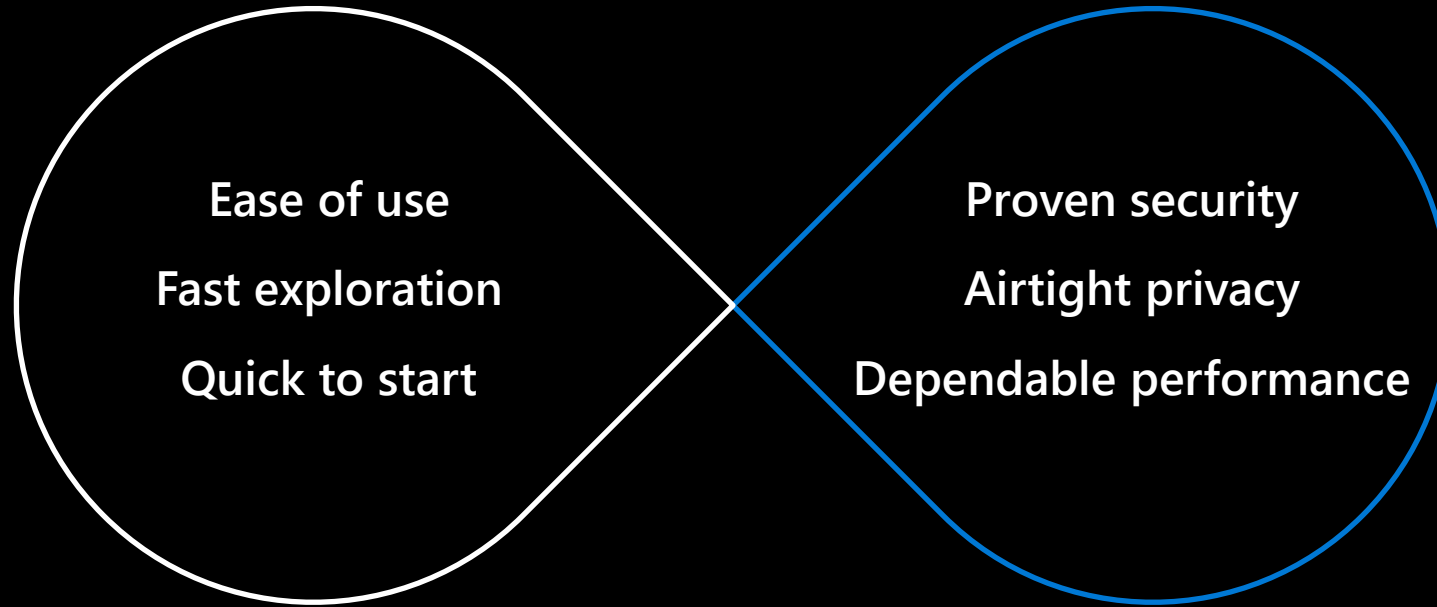
OR

Business analytics



Data warehouse

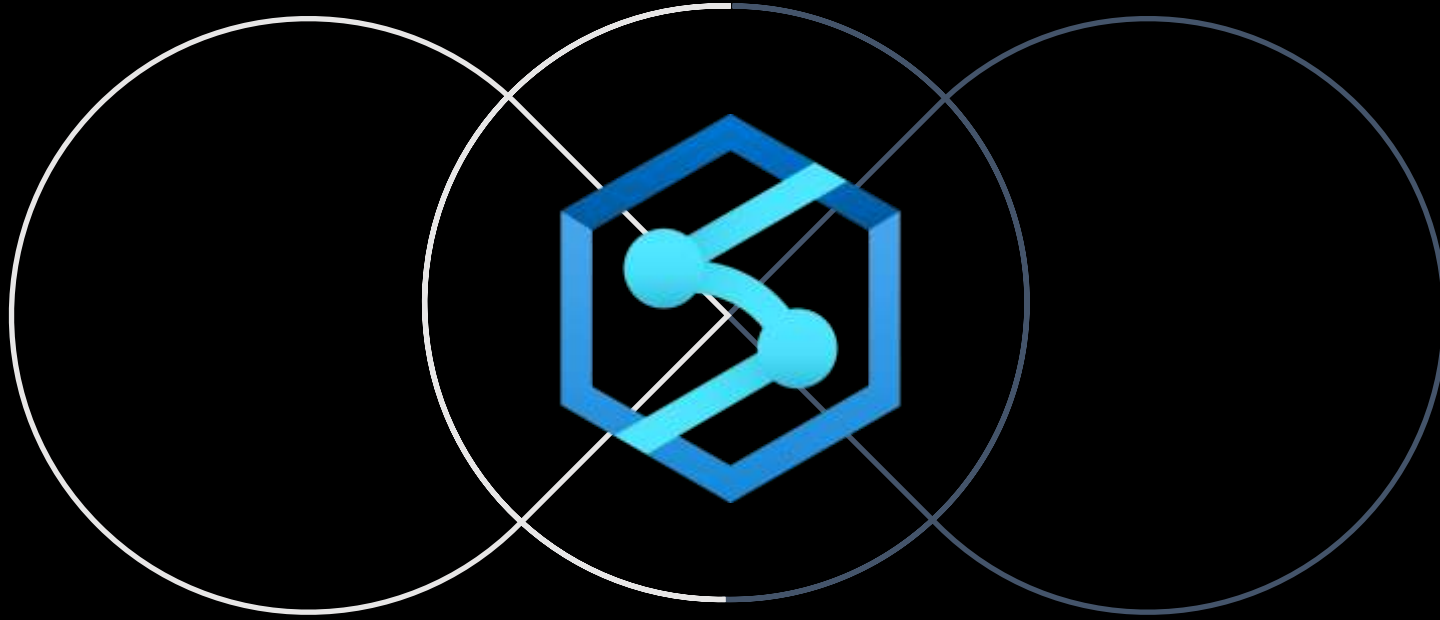
Azure meets these challenges with a single service to provide limitless analytics



Welcome to limitless

Data warehousing & big data analytics—all in one service

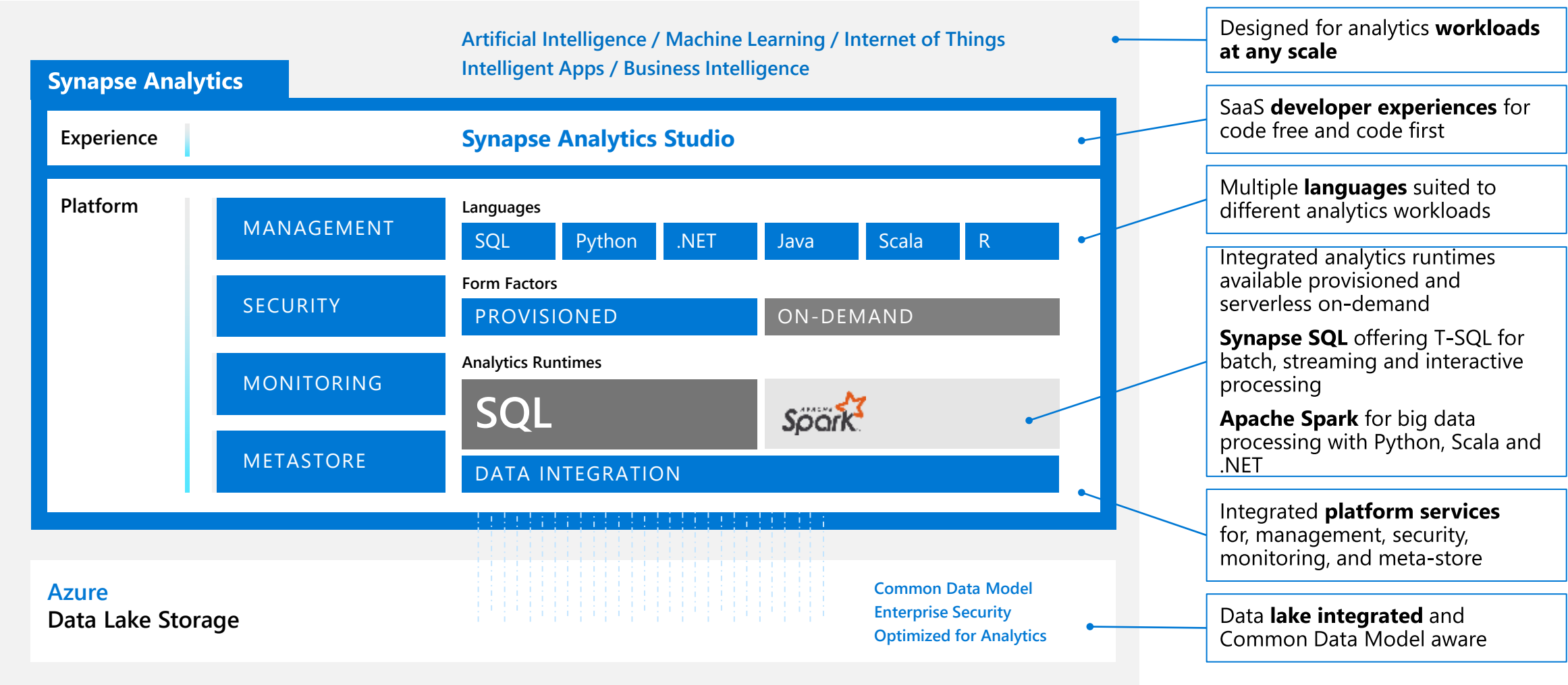
**Azure meets these challenges
with a single service to provide limitless analytics**



Azure Synapse Analytics

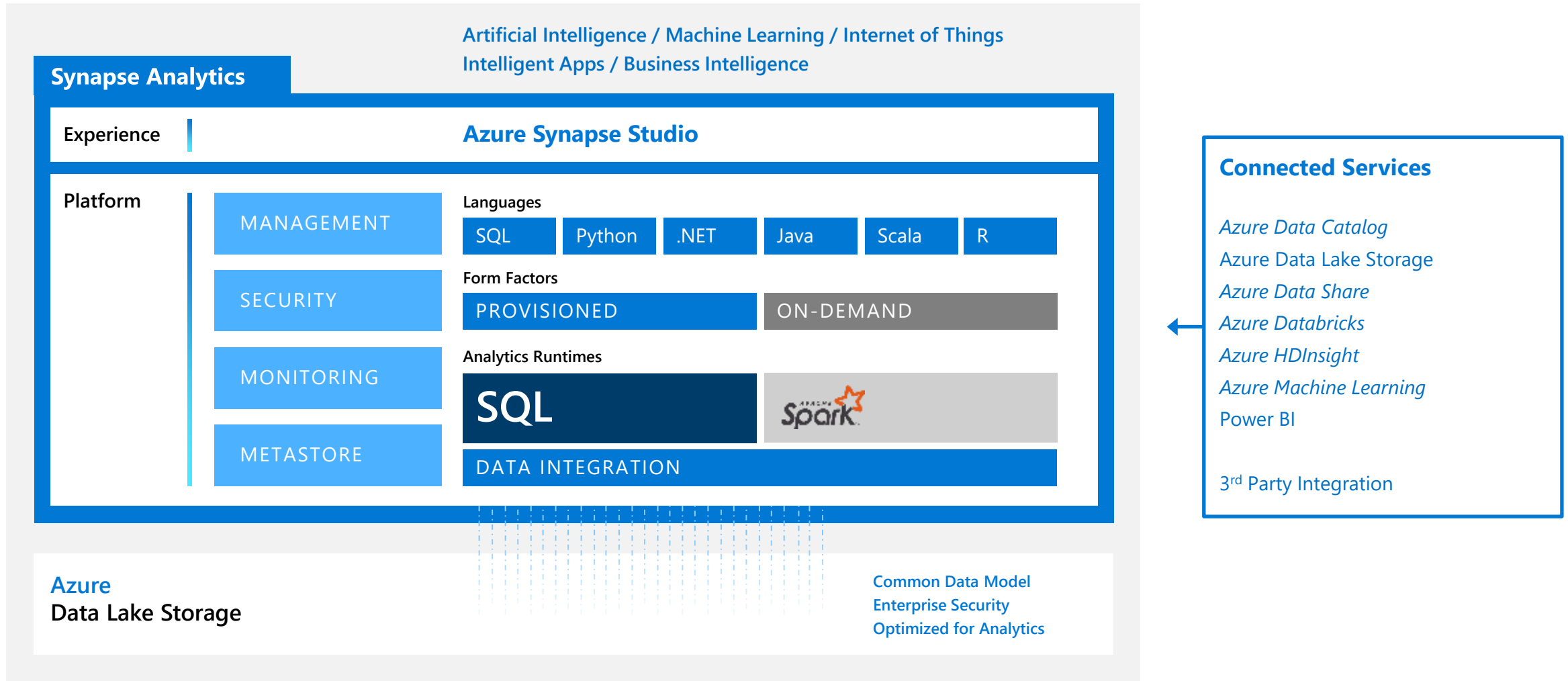
Azure Synapse Analytics

Limitless analytics service with unmatched time to insight



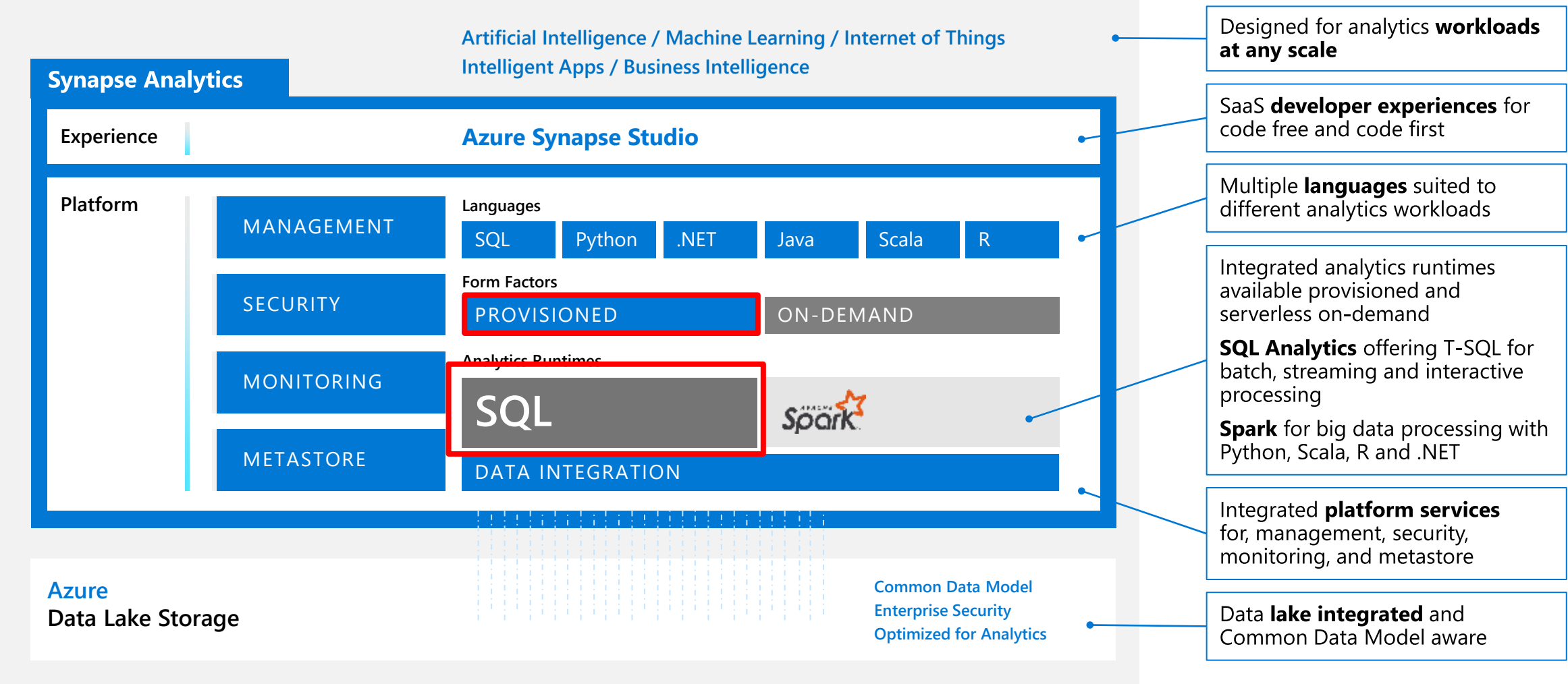
Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



Parallelism

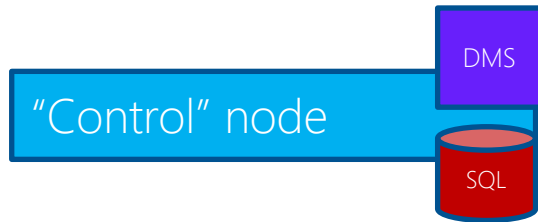
SMP - Symmetric Multiprocessing

- 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

SQL DW Logical Architecture (overview)



Compute Node – the “worker bee” of SQL DW

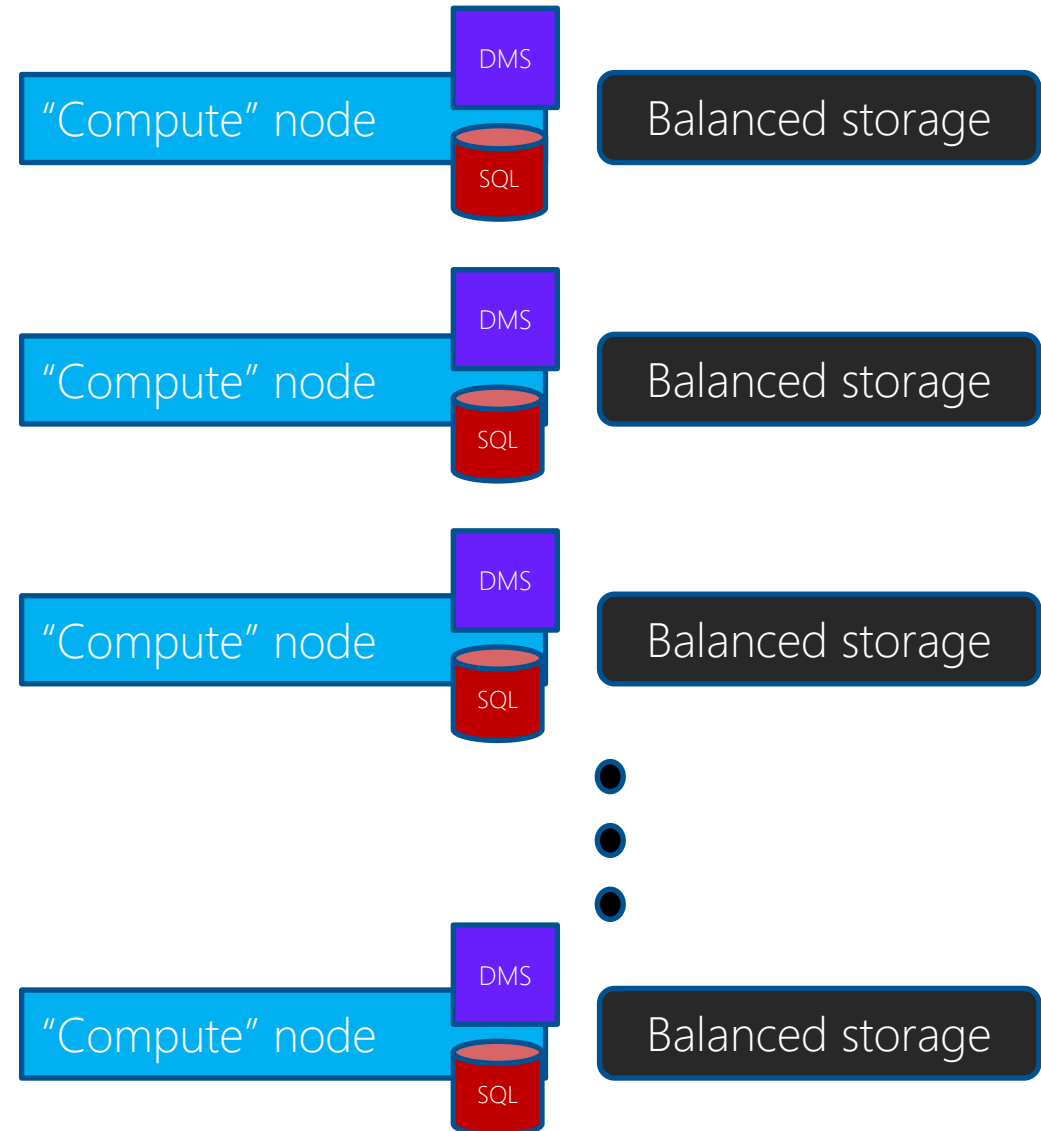
- Runs Azure SQL Server DB
- Contains a “slice” of each database
- CPU is saturated by storage

Control Node – the “brains” of the SQL DW

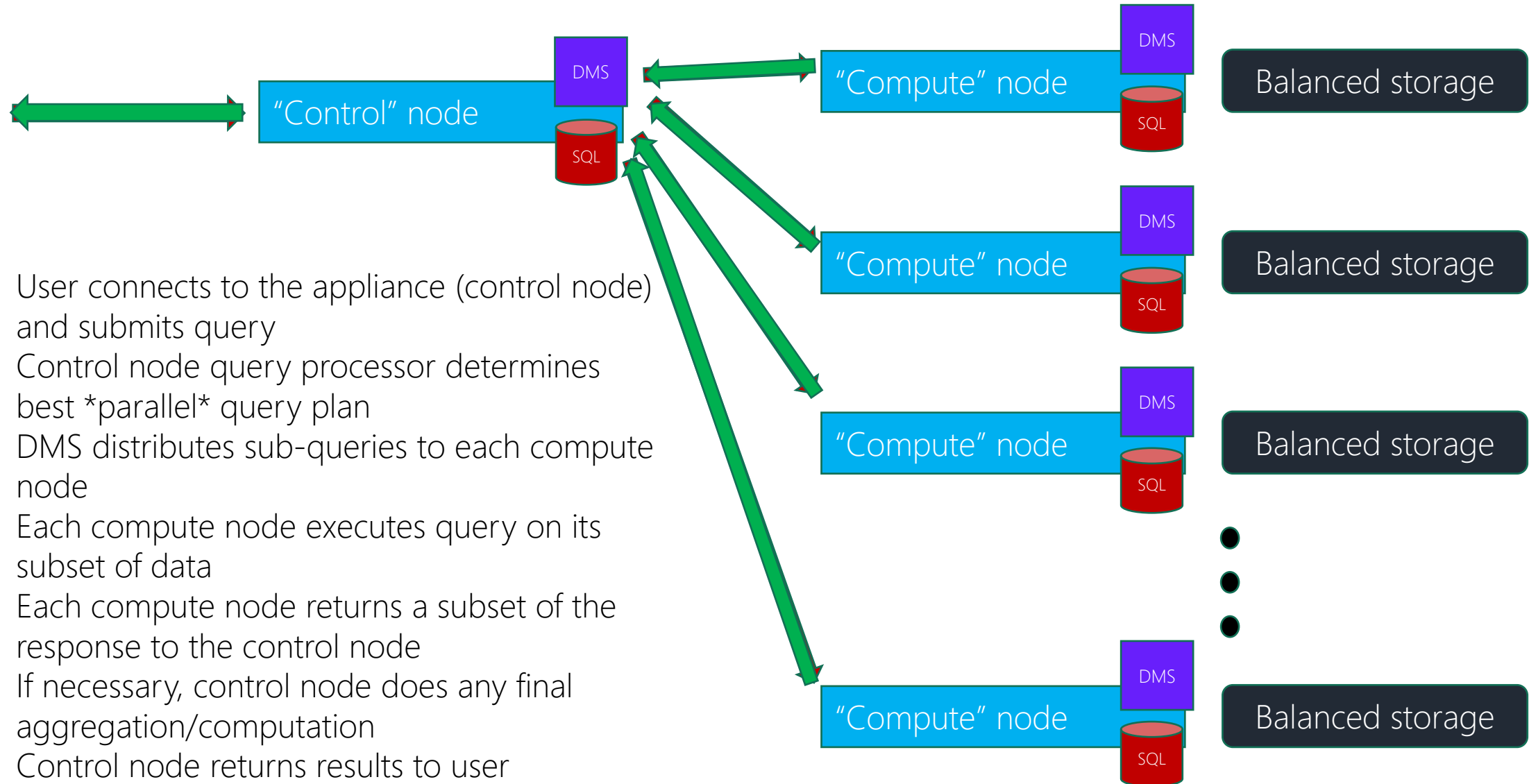
- Also runs Azure SQL Server DB
- Holds a “shell” copy of each database
 - Metadata, statistics, etc
- The “public face” of the appliance

Data Movement Services (DMS)

- Part of the “secret sauce” of SQL DW
- Moves data around as needed
- Enables parallel operations among the compute nodes (queries, loads, etc)

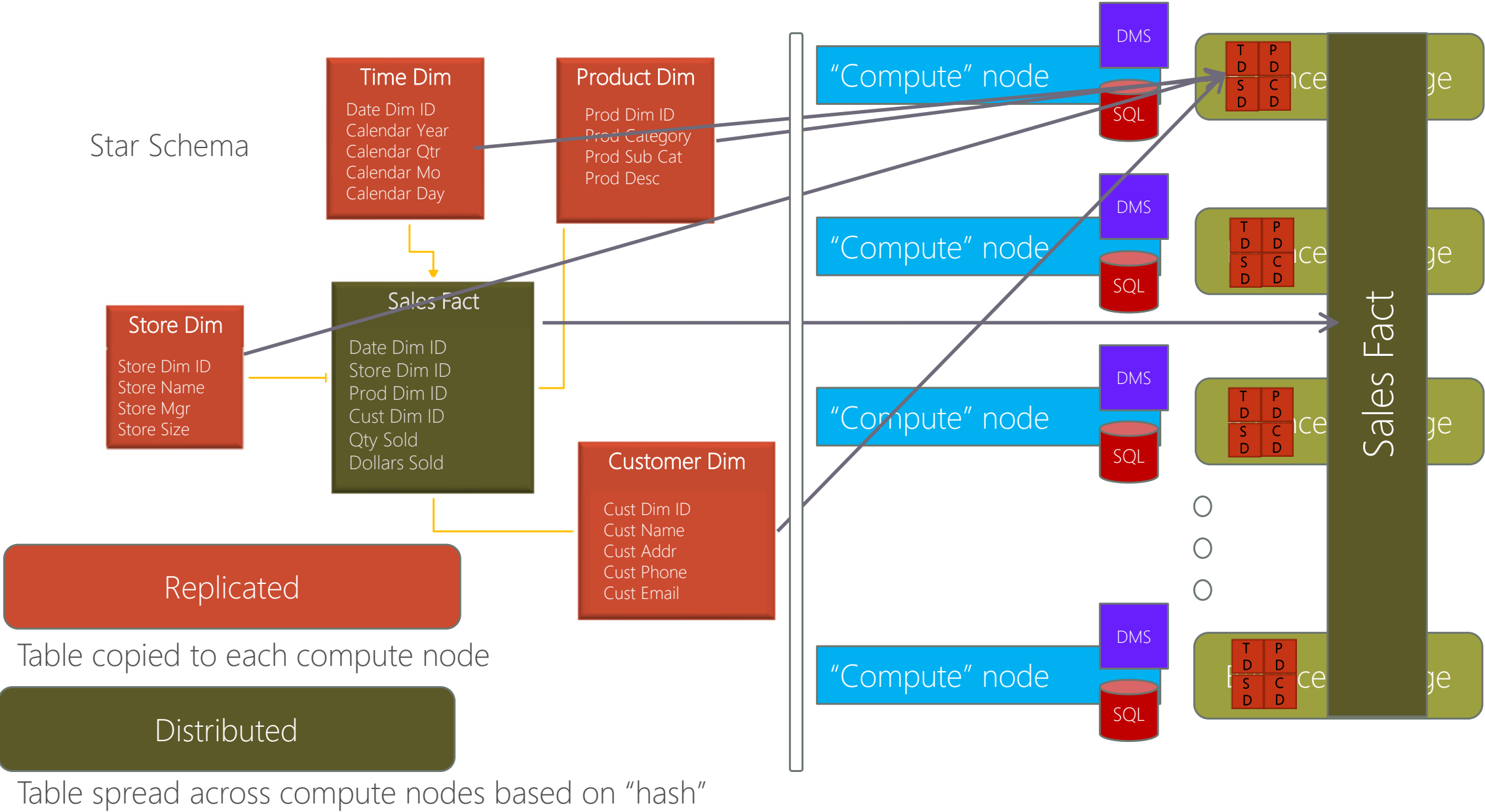


SQL DW Logical Architecture (overview)



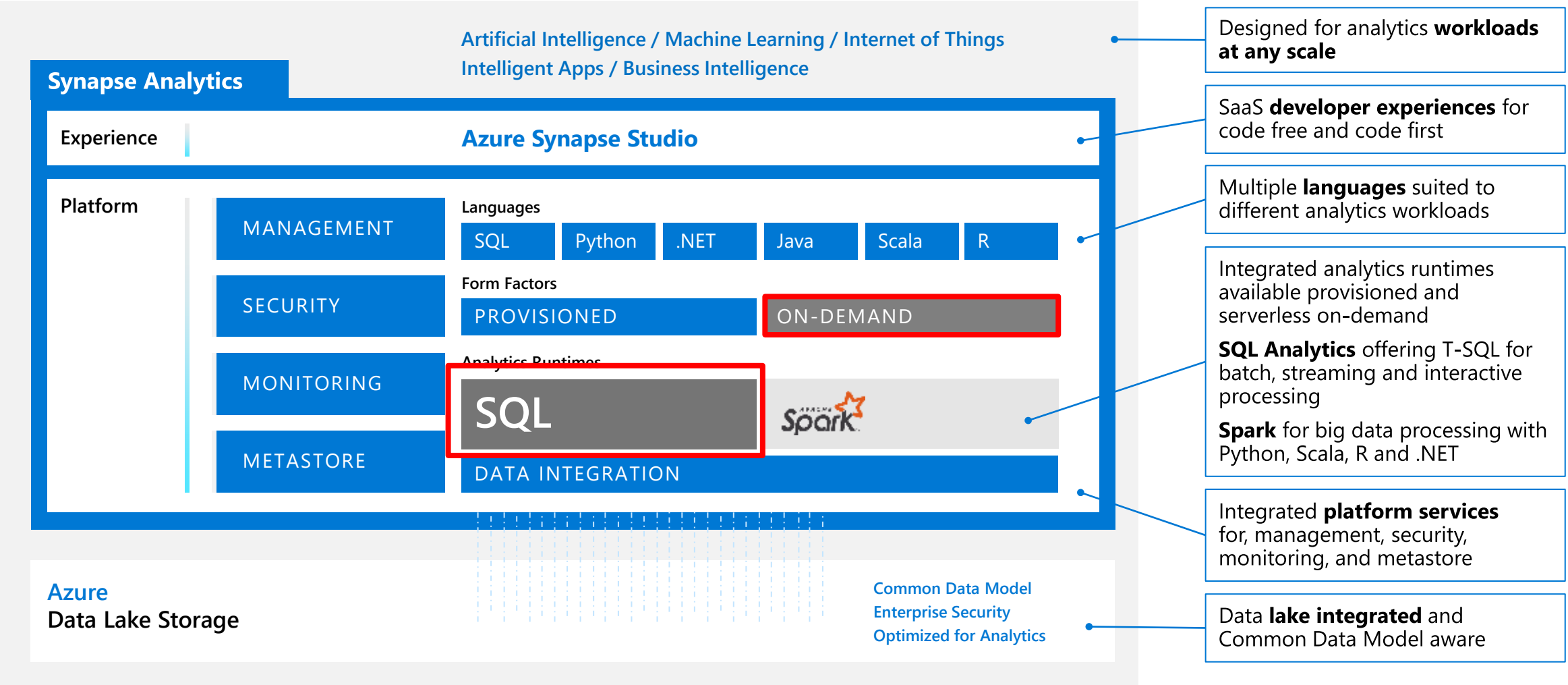
Queries running in parallel on a subset of the data, using separate pipes effectively making the pipe larger

SQL DW Data Layout Options



Azure Synapse Analytics

Integrated data platform for BI, AI and continuous intelligence



Synapse SQL on-demand scenarios

Discovery and exploration

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

SQL On-demand reduces data lake exploration to the right-click!

Data transformation

How to convert CSVs to Parquet quickly? How to transform the raw data?

Use the full power of T-SQL to transform the data in the data lake

3 main scenarios that SQL on-demand is great for



Basic discovery and exploration

Quickly view the data → extract insights



Logical data warehouse

Relational abstraction on top of raw

Always up-to-date view

T-SQL → blurring the line between a relational database and a data lake



Data transformation

Simple, scalable, and performant way to transform data in the lake using T-SQL

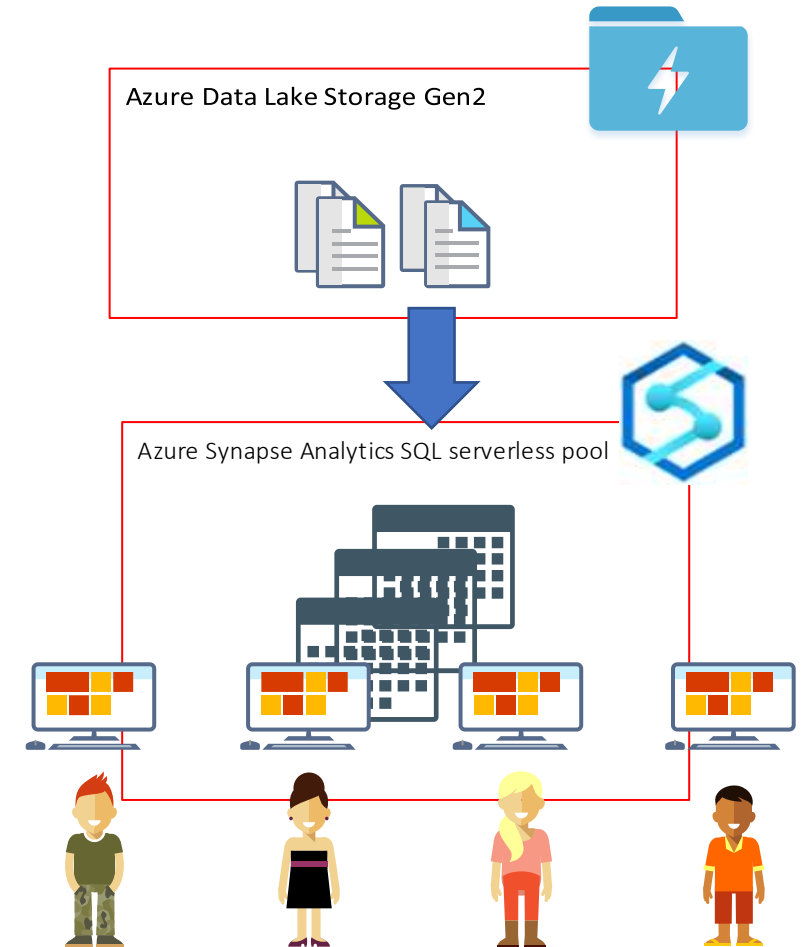
- For example, using the Copy activity in Azure Data Factory you can convert CSV files in the data lake (via T-SQL views in SQL on-demand) to Parquet files in the data lake



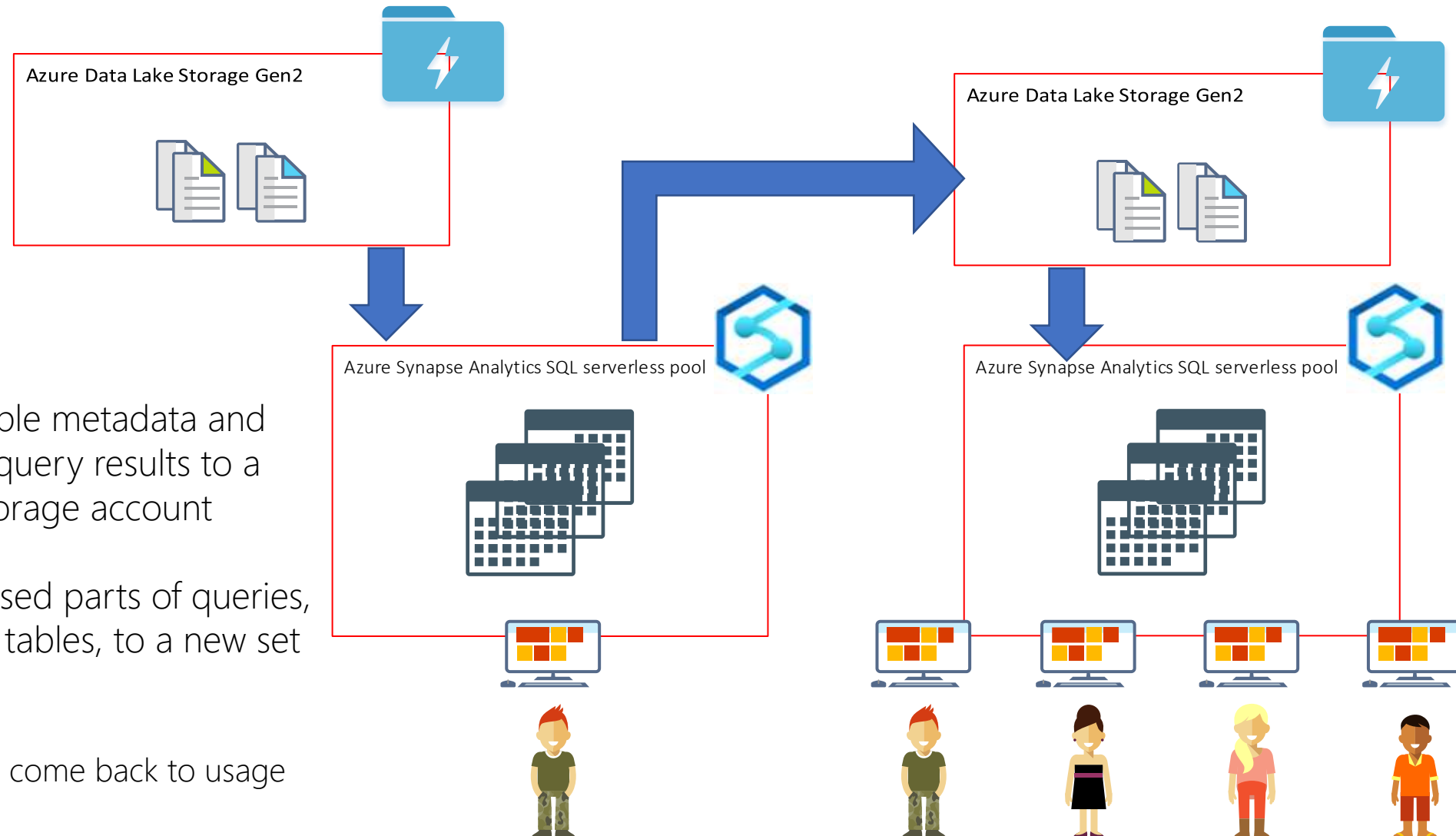
Usage Patterns

1) Discover / explore data in data lake

- Explore files with T-SQL
- Create External Tables / Views over files
- Join disparate data if needed
- Logical DWH by creating a relational abstraction on top of raw
- You can transform your data to satisfy whichever model you want
- Quickly create (Power BI) reports on top of data in the lake
- ...



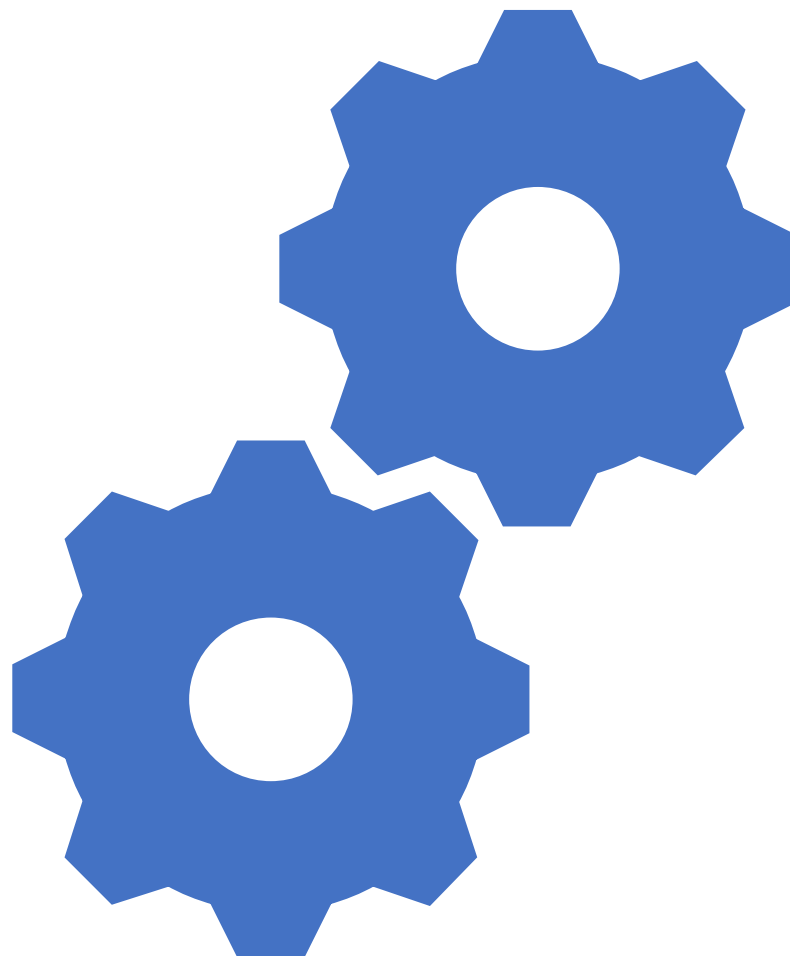
2) Store query results to storage



- To create external table metadata and exports the SELECT query results to a set of files in your storage account
- To store frequently used parts of queries, like joined reference tables, to a new set of files
- When stored... you can come back to usage pattern (1)



Setup



Setup

- Just 1 thing → provisioning Azure Synapse Analytics
- SQL on-demand is immediately available for your workspace
- SQL pools can be configured to adapt to team or organizational requirements and constraints
 - <https://github.com/Azure/azure-synapse-analytics/blob/master/docs/quickstart-create-a-sqlpool.md>

SQL on-demand Endpoint

+ New SQL pool + New Apache Spark pool Refresh Reset SQL admin password Delete Launch Synapse Studio

Resource group (change)	: anbenedsynapse	Firewalls	: Show firewall settings
Status	: Succeeded	Primary ADLS Gen2 acco...	: https://anbenedstorageaccount.dfs.core.windows.net
Location	: West Europe	Primary ADLS Gen2 file s...	: anbenedsynapseadls
Subscription (change)	: Microsoft Azure Internal Consumption	SQL admin username	: sqladminuser
Subscription ID	: 0e176cf5-f3c9-4a86-be16-208f112e4f4f	SQL Active Directory ad...	: anbened@microsoft.com
Managed virtual network	: No	SQL endpoint	: anbenedsynapse.sql.azure.synapse.net
Managed Identity object ...	: f6216298-6505-4568-9d3b-faa429539548	SQL on-demand endpoint	: anbenedsynapse-ondemand.sql.azure.synapse.net
Workspace web URL	: https://web.azure.synapse.net?workspace=%2fsubscriptions%2f0e176cf5-f3c9-4a86-be16-208f112e4f4f	Development endpoint	: https://anbenedsynapse.dev.azure.synapse.net
Tags (change)	: Click here to add tags		

Available resources



Object Explorer

Connect

- anbenedsynapse.sql.azure.synapse.net (SQL Server 12.0.2000.8 - sqladminuser)
- anbenedsynapse-ondemand.sql.azure.synapse.net (SQL Server 12.0.2000.8 - sqladminuser)
- Databases
 - System Databases
 - Security

File Edit View Help

CONNECTIONS

SERVICES

- Test
- Local
- Azure Synapse Analytics
 - anbenedsynapse-ondemand.sql.azure.synapse.net, <default> (sqla...

(Azure Synapse Analytics SQL serverless pool)

External Tables

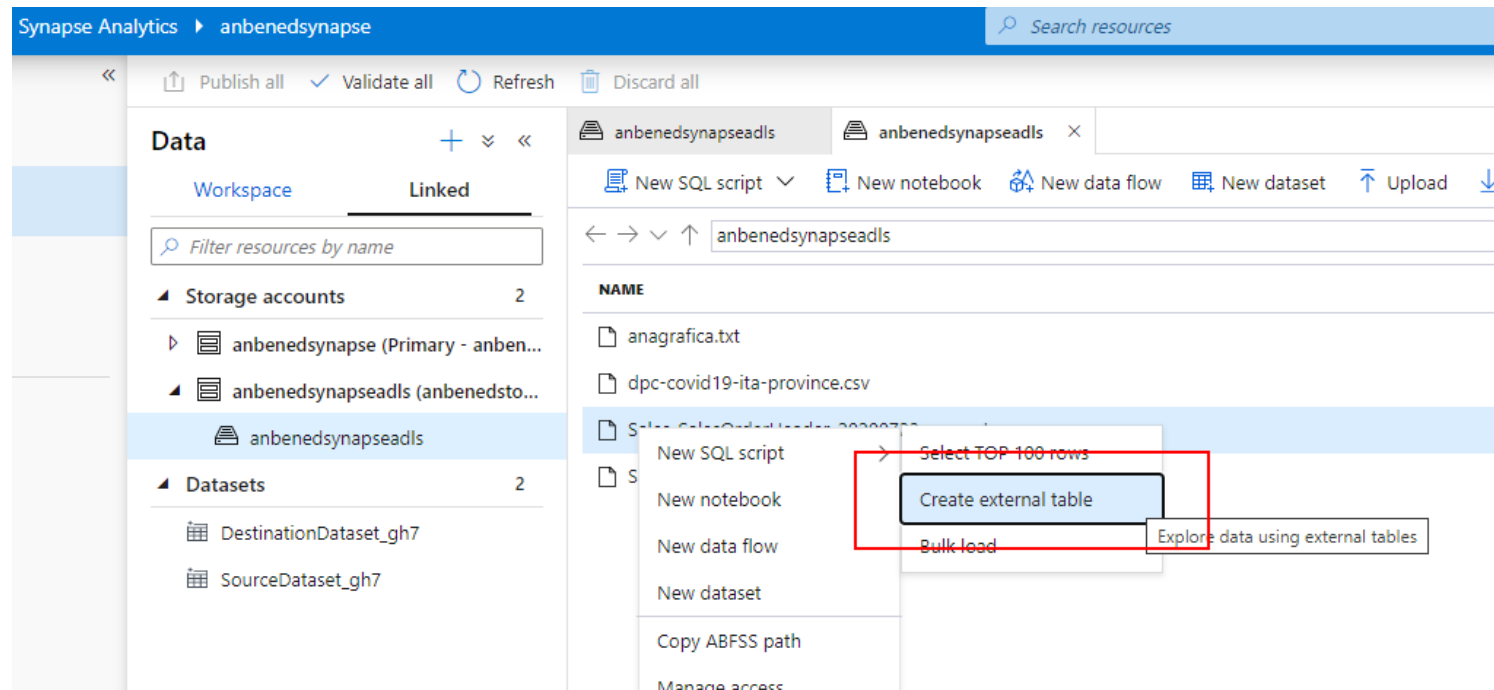
External Tables

- Used to read data from files or write data to files in Azure Storage
 - Query data in Azure Blob Storage or Azure Data Lake Storage with T-SQL
 - Store query results to files in Azure Blob Storage or Azure Data Lake Storage using CETAS
- Steps:
 - CREATE EXTERNAL DATA SOURCE
 - CREATE EXTERNAL FILE FORMAT
 - CREATE EXTERNAL TABLE

```
CREATE EXTERNAL TABLE abc
WITH
(
    LOCATION = 'myFolder',
    DATA_SOURCE = myDS,
    FILE_FORMAT = myFF
)
AS
    SELECT ...
GO
```

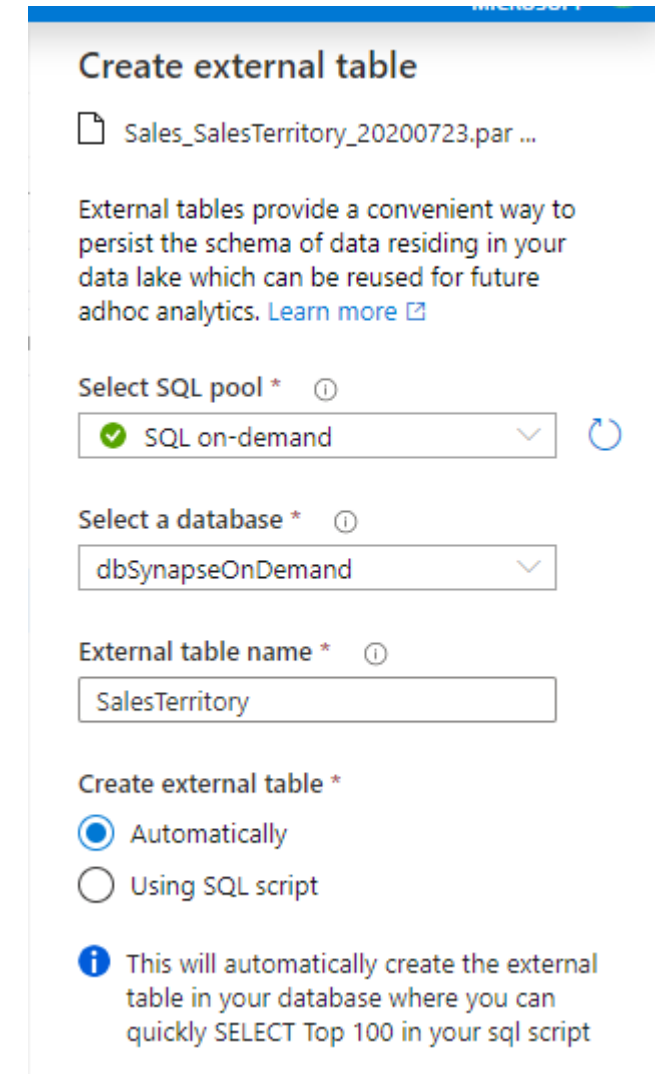
External Tables

- You can create them without writing any lines of codes
- We'll create a link definition between your Data Lake and the db
 - This takes advantage of the Synapse Analytics engine (MPP system)




External Tables

- Azure Synapse Analytics can create the external table for us
- We just need to provide the name of the database that we want to use, external table name, and the automatic option






The screenshot shows the 'Create external table' wizard in the Azure Synapse Analytics portal. At the top, the title 'Create external table' is displayed. Below it, a file icon and the name 'Sales_SalesTerritory_20200723.par ...' are shown. A descriptive text explains that external tables persist the schema of data in a data lake for future analytics, with a 'Learn more' link. The form includes three dropdown menus: 'Select SQL pool *' with 'SQL on-demand' selected, 'Select a database *' with 'dbSynapseOnDemand' selected, and 'External table name *' with 'SalesTerritory' entered. At the bottom, the 'Create external table *' section has two radio buttons: 'Automatically' (selected) and 'Using SQL script'. An information icon and a note state: 'This will automatically create the external table in your database where you can quickly SELECT Top 100 in your sql script'.


Create external table

 Sales_SalesTerritory_20200723.par ...


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 * 

 SQL on-demand 

Select a database * 

dbSynapseOnDemand


External table name * 

SalesTerritory

Create external table *

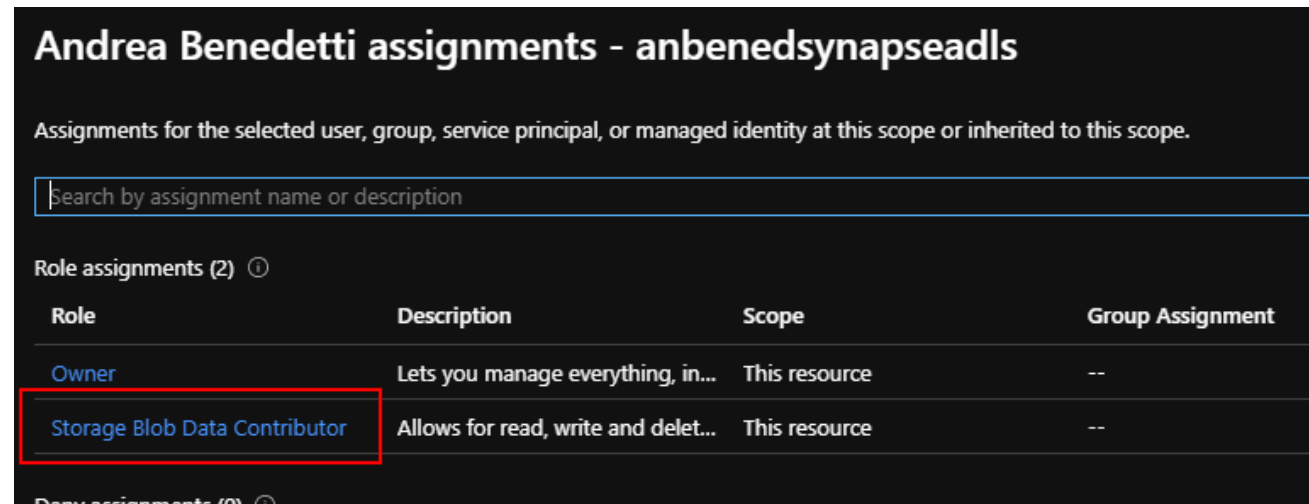
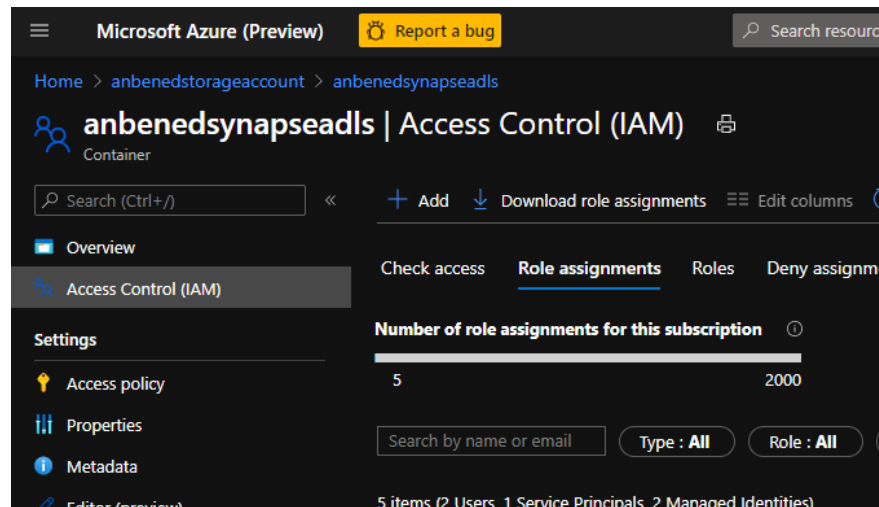
☒ Automatically

☐ Using SQL script

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

Security Tip

- Grant user / group 'Storage Blob Data Contributor' role on the storage account you're trying to query



External Tables

- Then, we'll be able to query the table

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Databases' tree is expanded to show the 'dbo.SalesTerritory' external table. The 'Columns' list for this table includes: TerritoryID (int, null), Name (varchar(8000), ...), CountryRegionCode (v...), Group (varchar(8000), ...), SalesYTD (numeric(38,...), SalesLastYear (numeric(38,...), CostYTD (numeric(38,...), CostLastYear (numeric(38,...), rowguid (varchar(8000,...), and ModifiedDate (datetime...). The main pane shows a SQL query in the 'Query Editor' window:

```
-- Step 6: Query TOP 100 rows
SELECT TOP 100 * FROM SalesTerritory
GO
```

Below the query editor, the 'Results' tab is active, showing a table view of the query results. The table has 6 columns: TERRITORYID, NAME, COUNTRYREGI..., GROUP, SALESYTD, and SALESLE... The results are as follows:

TERRITORYID	NAME	COUNTRYREGI...	GROUP	SALESYTD	SALESLE...
1	Northwest	US	North America	7887186.78820...	329869...
2	Northeast	US	North America	2402176.84760...	360714...
3	Central	US	North America	3072175.11800...	320501...
4	Southwest	US	North America	10510853.8739...	536657...
5	Southeast	US	North America	2538667.25150...	392507...
6	Canada	CA	North America	6771829.13760...	569398...
7	France	FR	Europe	4772398.30780...	239653...
8	Germany	DE	Europe	3805202.34780...	130794...

External Tables (manually)

- You can use external tables in your queries the same way you use them in SQL Server queries

```
CREATE EXTERNAL TABLE dbo.myFileParquet
(
    calendarDate varchar(50),
    country varchar(50),
    code varchar(50),
    ...
)
WITH
(
    LOCATION = 'anbenedsynapsene/myFile.parquet',
    DATA_SOURCE = [SqlOnDemandDemoNE],
    FILE_FORMAT = [SynapseParquetFormat]
)
GO

SELECT TOP 10 * FROM dbo.myFileParquet
GO

/*
DROP EXTERNAL TABLE dbo.myFileParquet
GO
*/
```



CETAS



CREATE EXTERNAL TABLE AS SELECT



CETAS

Overview

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

Benefits

Select any data set and store it in parquet format.

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

Use saved data using external table.

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

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

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

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

CETAS

```
CREATE EXTERNAL TABLE dataAggregatedByMonthParquet
WITH
(
    LOCATION = 'anbenedsynapsene/dataAggregatedByMonthParquet',
    DATA_SOURCE = sqlOnDemandDemoNE,
    FILE_FORMAT = SynapseParquetFormat
)
AS
SELECT
    MONTH(CAST(data as date)) as M,
    SUM(CAST(totale_casi as int)) as N,
    sigla_provincia as P
FROM OPENROWSET
(
    BULK 'anbenedsynapsene/myFile.parquet',
    DATA_SOURCE = 'SqlOnDemandDemoNE',
    FORMAT='PARQUET'
) AS myData
GROUP BY data, sigla_provincia
GO

SELECT * FROM dataAggregatedByMonthParquet
GO
```

CETAS - Note

- ORDER BY clause in SELECT is not supported for CETAS
- LOBs can't be used with CETAS
- At this time DROP TABLE don't delete folder / files
 - 2 separate process: one to drop the table and another one to drop ADLS file

Log

Logs Retention

- Log retention = 60 days

Integration

Pipeline runs

Trigger runs

Integration runtimes

Activities

Apache Spark applications

SQL requests

Data flow debug

Analytics pools

SQL pools

Apache Spark pools

SQL requests

Refresh

Edit columns

Amsterdam, Berlin, B...

Last 30 days

Status : All


Pool : Built-in

Add filter

Showing 1 - 52 of 52 items

Request ID ↑↓	Request content ↑↓	Submit time ↑↓	Duration	Data processed	Submitter ↑↓	Status ↑↓
19476203	select ROW_NUMBER() OVER(OI	12/3/20, 2:28:57 PM	5s	3 MiB	anbened@microsoft.com	Completed
19468315	select top (5) * from [dbo].[time	12/3/20, 2:27:09 PM	5s	1 MiB	anbened@microsoft.com	Completed
9662178	SELECT TOP 100 * FROM OPENF	12/1/20, 4:18:11 PM	4s	4 MiB	anbened@microsoft.com	Completed
3774043	select ROW_NUMBER() OVER(OI	11/30/20, 12:33:02 PM	0s	1 MiB	anbened@microsoft.com	Completed
3773189	select * from (select c11, c265 a	11/30/20, 12:32:42 PM	0s	2 MiB	anbened@microsoft.com	Completed
3770973	select top (5) * from [dbo].[time	11/30/20, 12:32:11 PM	2s	1 MiB	anbened@microsoft.com	Completed
3772896	select ROW_NUMBFR0 OVER(OI	11/30/20 12:32:11 PM	7s	3 MiB	anbened@microsoft.com	Completed

```
select * from sys.dm_exec_requests_history order by start_time desc
```



Best practices for SQL on-demand

Best Practices

- Minimize latency: **colocate** your Azure storage account and your SQL on-demand endpoint
 - Storage accounts and endpoints provisioned during workspace creation are located in the same region
- Optimal performance: if you access other storage accounts with SQL on-demand, make sure they're in the **same region**
 - Different region = increased latency for the data's network transfer between the remote region and the endpoint's region

Best Practices

- Optimal performance → **same region**

Different region → Workspace WE; Storage NE

Execution time: ~31 sec

```
PRINT 'Start time: ' + CAST(SYSDATETIMEOFFSET() as varchar)
SELECT TOP 10 *,
    JSON_VALUE(content, '$._id') as ID,
    JSON_VALUE(content, '$.type') as TypeObj,
    JSON_VALUE(content, '$.title') as Title
FROM json.Books
GO
```

Start time: 2020-10-21 18:52:43.0529942 +0
Statement ID: {3A9FC555-8769-46D2-86C0-F301AA7624C5} | Query has
h: 0xA3D53FC7C24A0A9C | Distributed request ID: {D0DF16F3-0B0E-4
670-9677-222065823D5F}. Total size of data scanned is 25 megabyt
es, total size of data moved is 1 megabytes, total size of data
written is 0 megabytes.
(10 rows affected)
Total execution time: 00:00:30.517

Same region → Workspace NE; Storage NE

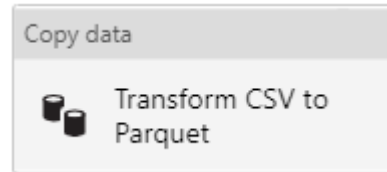
Execution time: ~13 sec

```
PRINT 'Start time: ' + CAST(SYSDATETIMEOFFSET() as varchar)
SELECT TOP 10 *,
    JSON_VALUE(content, '$._id') as ID,
    JSON_VALUE(content, '$.type') as TypeObj,
    JSON_VALUE(content, '$.title') as Title
FROM json.Books
GO
```

Start time: 2020-10-22 13:36:30.9970461 +0
Statement ID: {291E82F7-C734-4A36-A3B1-B76E765C4325} | Query has
h: 0xA3D53FC7C24A0A9C | Distributed request ID: {3314CA9A-C27B-4
FCB-B2A3-254AA49B0D4B}. Total size of data scanned is 25 megabyt
es, total size of data moved is 1 megabytes, total size of data
written is 0 megabytes.
(10 rows affected)
Total execution time: 00:00:12.961

Best Practices

- Multiple applications and services might access your storage account
 - **don't stress the storage** with other workloads during query execution
- If possible, you can **prepare files** for better performance
 - Convert CSV and JSON → Parquet (data scanned reduced)
 - Simple idea: copy data in ADF



Best Practices (csv vs. parquet)

Name	Size
dpc-covid19-ita-province.csv	34.8 MB
dpc-covid19-ita-province.parquet	3.7 MB

	data	stato	codice_regione	denominazione_regione	codice_provincia	denominazione_provincia	sigla_provincia	lat	long	totale_casi	note
1	2020-02-24T18:00:00	ITA	13	Abruzzo	066	L'Aquila	AQ	42.35122196	13.39843823	0	NULL
2	2020-02-24T18:00:00	ITA	13	Abruzzo	067	Teramo	TE	42.6589177	13.70439971	0	NULL
3	2020-02-24T18:00:00	ITA	13	Abruzzo	068	Pescara	PE	42.46458398	14.21364822	0	NULL
4	2020-02-24T18:00:00	ITA	13	Abruzzo	069	Chieti	CH	42.35103167	14.16754574	0	NULL
5	2020-02-24T18:00:00	ITA	13	Abruzzo	979	In fase di definizione/aggiornamento	NULL	NULL	NULL	0	NULL
6	2020-02-24T18:00:00	ITA	17	Basilicata	076	Potenza	PZ	40.63947052	15.80514834	0	NULL

Statement test	Format	Duration	Rows	Data Scanned	Data Moved
SELECT *	CSV	~20 sec	441259	35 MB	38 MB
	Parquet	~18 sec	441259	4 MB	38 MB
COUNT(*)	CSV	~0 sec	1	35 MB	1 MB
	Parquet	~0 sec	1	1 MB	1 MB
SELECT sum(cast(totale_casi as int))	CSV	~1 sec	1	35 MB	1 MB
	Parquet	~0 sec	1	2 MB	1 MB
SELECT data, codice_provincia, totale_casi	CSV	~8 sec	441259	35 MB	14 MB
	Parquet	~6 sec	441259	2 MB	14 MB

Best Practices

- **Data types** affect performance
 - Use smallest data size that will accommodate the largest possible value
 - If the maximum character value length is 30 characters, use a character data type of length 30.
 - If all character column values are of fixed size, use char or nchar. Otherwise, use varchar or nvarchar.
 - If the maximum integer column value is 500, use smallint because it's the smallest data type that can accommodate this value. You can find integer data type ranges in this [article](#).
 - If possible, use varchar and char instead of nvarchar and nchar
 - Use integer-based data types if possible. SORT, JOIN, and GROUP BY operations complete faster on integers than on character data



Power BI and SQL on-demand

Power BI

- There's not Power BI by default
- If you want to use Power BI:
 - Sign in to powerbi.microsoft.com
 - Create a new Power BI workspace
 - In Synapse Studio, go to Manage > Linked Services

New linked service (Power BI)

i Choose a name for your linked service. This name cannot be updated later.

Name *

PowerBIAnbenedSynapse

Description

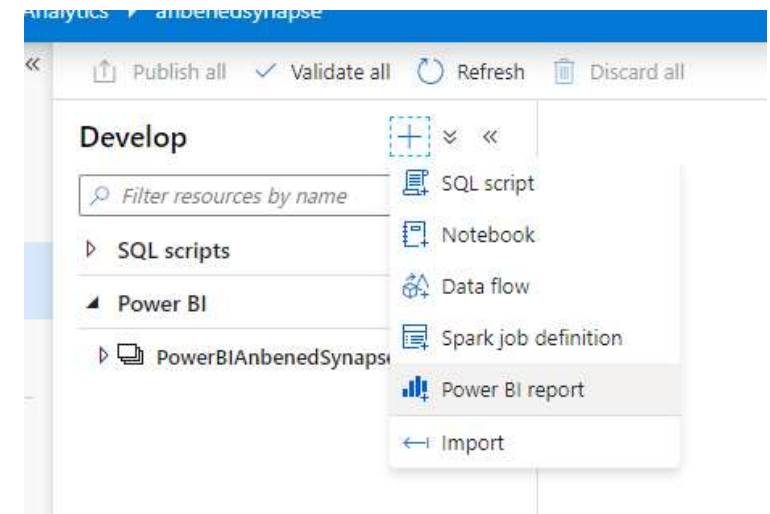
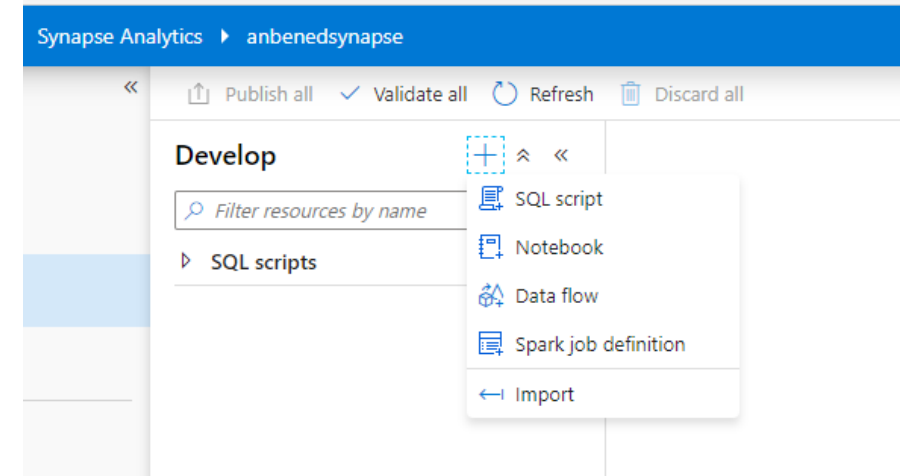
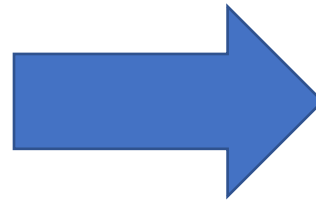
Workspace name *

PowerBIAnbenedSynapse (8c0ccb23-13ee-49d1-81c0-2c2b9d0616eb)

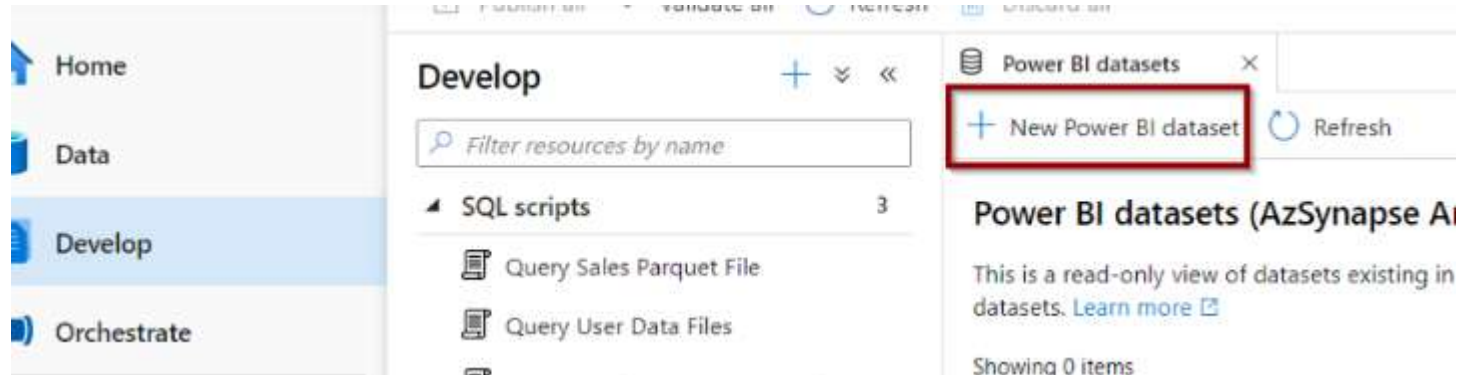
Annotations

+ New

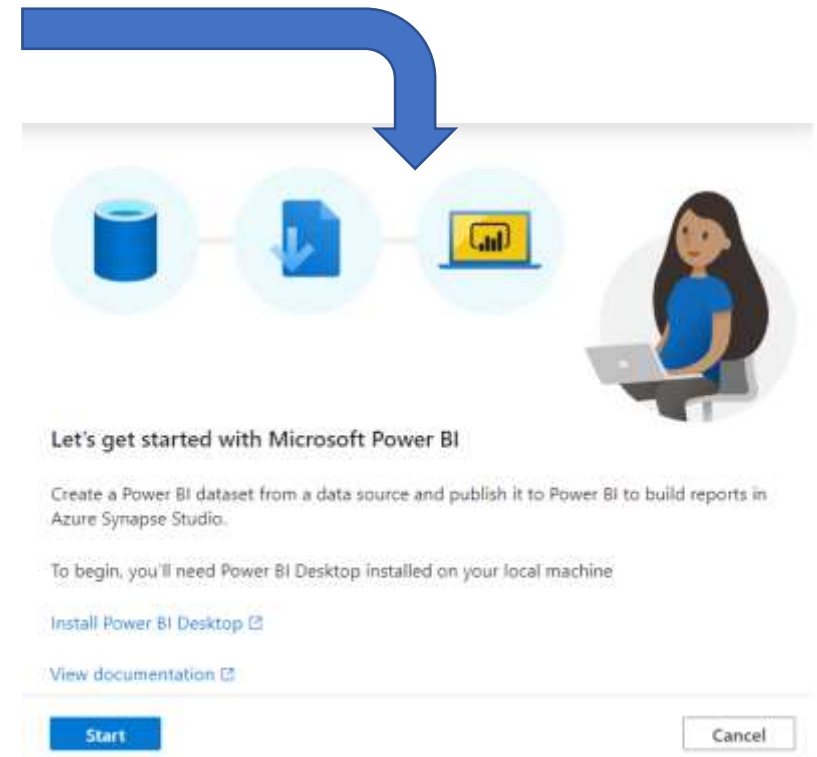
Advanced ⓘ



Create Power BI Dataset

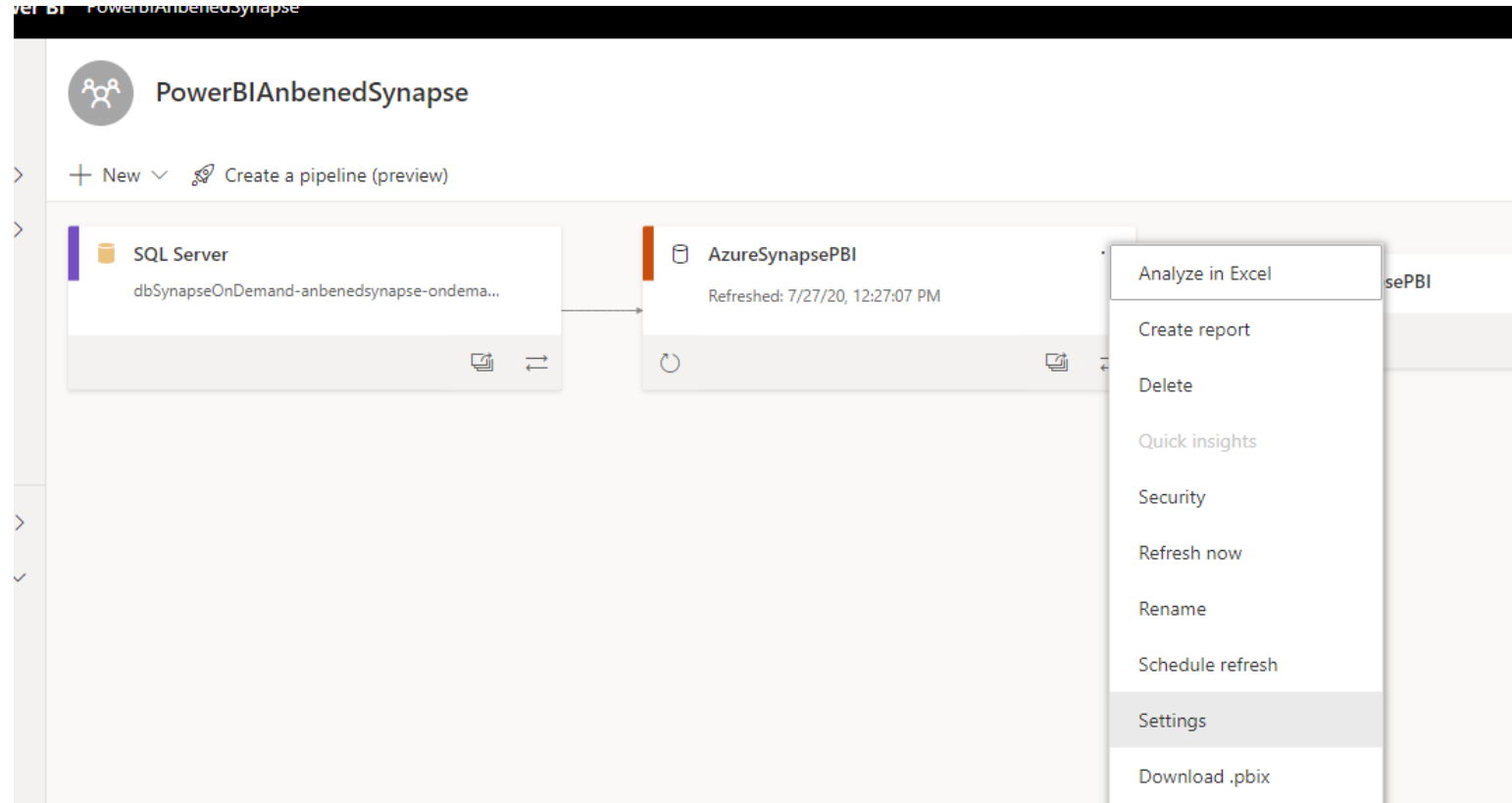


- Like a tutorial:
 - Select your database
 - Select your data source (serverless)
 - Download the Power BI template
 - Open it with Power BI Desktop
 - Select your dataset
 - Choose DirectQuery
 - Create the relationship
 - Publish to Power BI service
 - The dataset will become available in your Azure Synapse Analytics workspace



Security 1/2

- Configure the credentials for the Power BI dataset in the Power BI service



Security 2/2

- Edit the credentials

Settings for AzureSynapsePBI

This dataset has been configured by [anbened@microsoft.com](#).

[Refresh history](#)

► Gateway connection

◄ Data source credentials

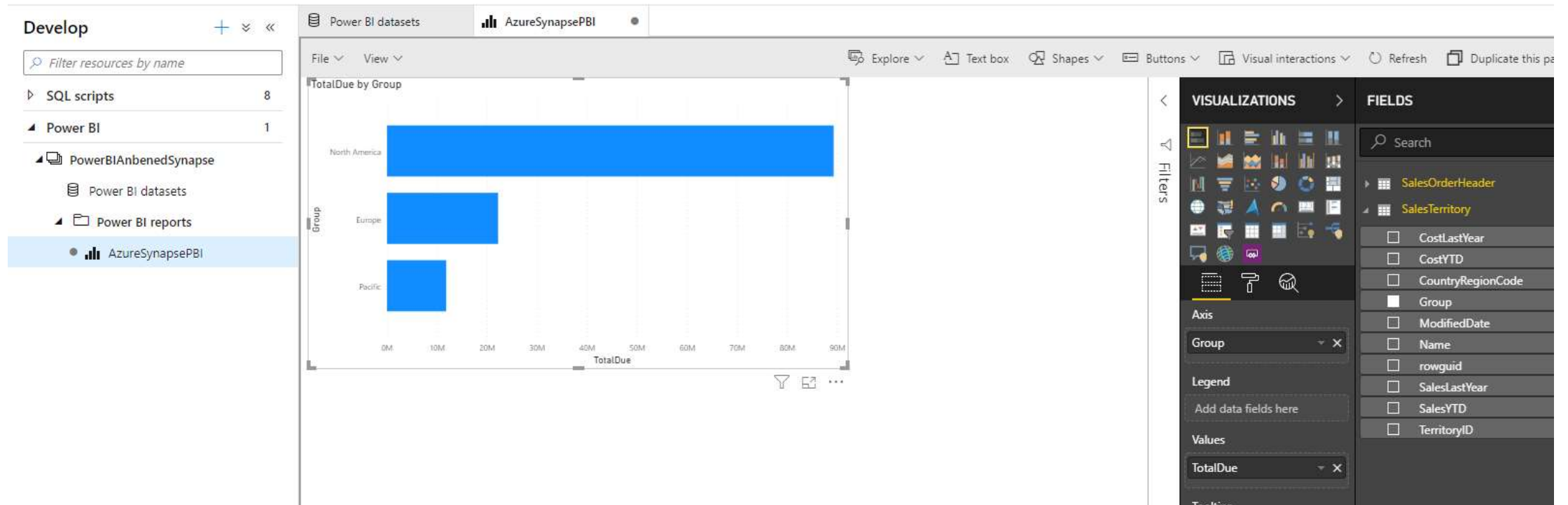
⊗ Your data source can't be refreshed because the credentials are invalid. Please update your credentials and try again.

dbSynapseOnDemand-anbenedsynapse-ondemand.sql.azuresynapse.net⊗ [Edit credentials](#)

► Sensitivity label

Power BI Report

- When you save the report, it will become available in your workspace. You can also share it.





Pricing

Pricing

*Does **Azure Synapse SQL On Demand** pricing have provision for Azure reserved capacity (e.g. 1 year or 3 year reserved) like how it is present in Azure SQL Pool or it is flat as per the rate described above?*

At the moment, there is no reserved capacity payment model for the SQL On-demand.

*Suppose if the data queried through **Azure Synapse SQL On Demand** is 1 MB , so price will be calculated according to pro-rate usage or any rounding logic involved?e.g. with a rate of €4.217 per TB, 1 MB data query cost will be € 0.000004217?*

The floor for charging is actually 10 MB, so the minimum charge is for 10 MB, but otherwise the math you're showing above is good.

For Azure Synapse SQL On Demand, the pricing is €4.217 per TB of data processed in West EU region.

Pricing example

- How to calculate price for Azure Synapse SQL (Serverless) as below?

Query Scenario: 30 execution times /month; 0.5 TB per execution times

Calculation: $30 * 0.5 * 4.217 = 63.255$ € (Monthly cost)

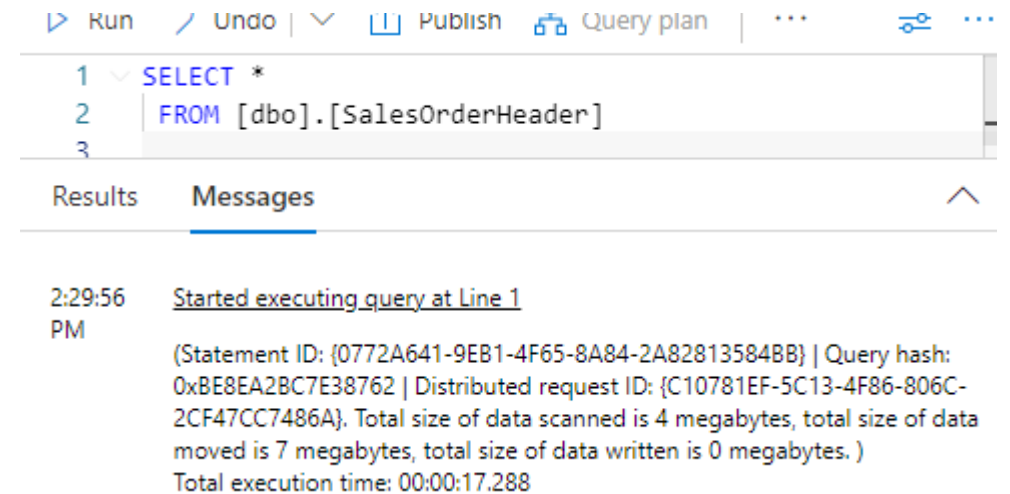
Pricing Tips

- Charging is per data processed
- If data is in Parquet format which is compressed, it will be cheaper
- Also, Parquet format is columnar so you will be charged only for columns you need in your query, not all columns, making it even cheaper
- In case of CSV, data is not only uncompressed, but SQL on-demand would need to read whole rows for you to extract columns you target

Billing: “total processed” volume

Each query:

- Total size of data scanned is XX megabytes
- Total size of data moved is XX megabytes
- Total size of data written is XX megabytes



The screenshot shows the SQL Server Enterprise Manager interface. At the top, there is a toolbar with buttons for 'Run', 'Undo', 'Publish', 'Query plan', and others. Below the toolbar, a query window displays the following SQL statement:

```
1 SELECT *
2 FROM [dbo].[SalesOrderHeader]
3
```

Below the query window, there are two tabs: 'Results' and 'Messages'. The 'Messages' tab is currently selected, showing the following message:


2:29:56 PM Started executing query at Line 1
(Statement ID: {0772A641-9EB1-4F65-8A84-2A82813584BB} | Query hash: 0xBE8EA2BC7E38762 | Distributed request ID: {C10781EF-5C13-4F86-806C-2CF47CC7486A}). Total size of data scanned is 4 megabytes, total size of data moved is 7 megabytes, total size of data written is 0 megabytes.)
Total execution time: 00:00:17.288

Total data processed = data scanned + data moved + data written

- Data processed = data stored internally while executing query
 - Data read (compressed data + metadata reads) + intermediate results (data shuffled, uncompressed format always)
 - + data transferred to node you connected to before returning results to client
 - (this transfer is accounted for as data processed also)
 - + in general case: autostats and read-ahead.

Billing & global stats query

- Global stats query = queries that system automatically executes to figure out what are the statistics in the data
 - You are charged for that query as well
 - Without statistics execution plan would be suboptimal and would lead to more data processed by the user query itself and worst performance

88872389		SELECT C1 FROM OPENROWSET	12/11/20, 4:56:06 PM	7s	11 MiB
88871523		SELECT C1 FROM OPENROWSET	12/11/20, 4:56:03 PM	8s	15 MiB
88871442		SELECT C1 FROM OPENROWSET	12/11/20, 4:56:01 PM	7s	12 MiB
88870508		*** Global stats query ***	12/11/20, 4:55:59 PM	7s	12 MiB
88872090		SELECT C1 FROM OPENROWSET	12/11/20, 4:55:59 PM	11s	12 MiB
88870465		*** Global stats query ***	12/11/20, 4:55:57 PM	9s	9 MiB
88872005		*** Global stats query ***	12/11/20, 4:55:57 PM	12s	9 MiB

Cost control

Cost control

```
exec sp_set_data_processed_limit
    @type = N'daily', @limit_tb = 5
exec sp_set_data_processed_limit
    @type= N'weekly', @limit_tb = 10
exec sp_set_data_processed_limit
    @type= N'monthly', @limit_tb = 50
```

```
SELECT * FROM sys.dm_external_data_processed
```

	type	data_processed_mb
1	daily	17
2	weekly	17
3	monthly	3065

The screenshot displays the Microsoft Azure Synapse Analytics portal. The left sidebar shows the 'Manage' tab selected. The main content area shows the 'SQL pools' section with a table listing pools. A red box labeled '2' points to the 'Type' column of the table. Below the table, the 'Cost Control' section is visible, showing options to enable or disable limits for daily, weekly, and monthly data usage. A red box labeled '1' points to the 'Manage' tab in the sidebar.

Name	Type
Built-in	Serverless
SQL DW Nikola	Dedicated

Cost Control

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

Daily limit ☐ Enable ☒ Disable

Data used today: 17 MB

Enter Daily limit in TB: TB

Weekly limit ☐ Enable ☒ Disable

Data used this week: 17 MB

Enter Weekly limit in TB: TB

Monthly limit ☐ Enable ☒ Disable

Data used this month: 2 GB

Enter Monthly limit in TB: TB



Notes

- There is **no cache** yet in SQL on-demand so the queries won't run faster after the first run
- There is **no way for a user to pre-allocate more resources** to SQL on-demand at this moment
- Everything related to reading files from storage might have an impact on query performance
 - SQL on-demand allows you to query files in your Azure storage accounts. It doesn't have local storage or ingestion capabilities. So, all files that the query targets are external to SQL on-demand.

```
...for object to mirror...  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod...
```

Demo

```
...types.Operator):  
...X mirror to the selected  
...object.mirror_mirror_x"  
...mirror X"
```




Q&A



