PASS
Data Community
SUMMIT 2021

# Creating a Logical Data Warehouse using Azure Synapse Analytics Serverless SQL Pools


**Andy Cutler**

He/him

DW/BI Engineer

Datahai


# Andy Cutler

He/him

## DW/BI Engineer

Datahai

linkedin.com/in/andycutler/

@MrAndyCutler

Datahai.co.uk

ServerlessSQL.com

I currently work for Datahai, a Microsoft Silver Partner company I founded in 2020 as a Business Intelligence/Data Warehouse professional

I am a current Data Platform MVP

Microsoft MVP Most Valuable Professional

Microsoft Partner | Silver Data Analytics Silver Data Platform

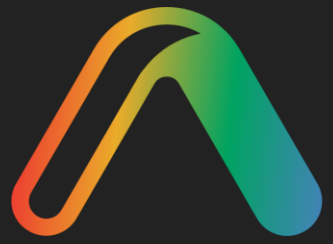
PASS Data Community SUMMIT 2021

# Session evaluation

Your feedback is important to us

**Evaluate this session at:**

www.PASSDataCommunitySummit.com/evaluation



PASS
**Data Community**
**SUMMIT** 2021

## Part 1: Overview

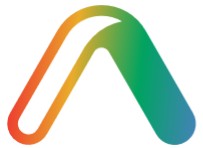
- Synapse Analytics
- Serverless SQL Pools
- Scenarios
- Logical Data Warehouse

## Part 2: Create

- Creating SQL objects
- Pipelines
- Database Designer

## Part 3: Discover

- Endpoint Connectivity
- Azure Purview
- Power BI



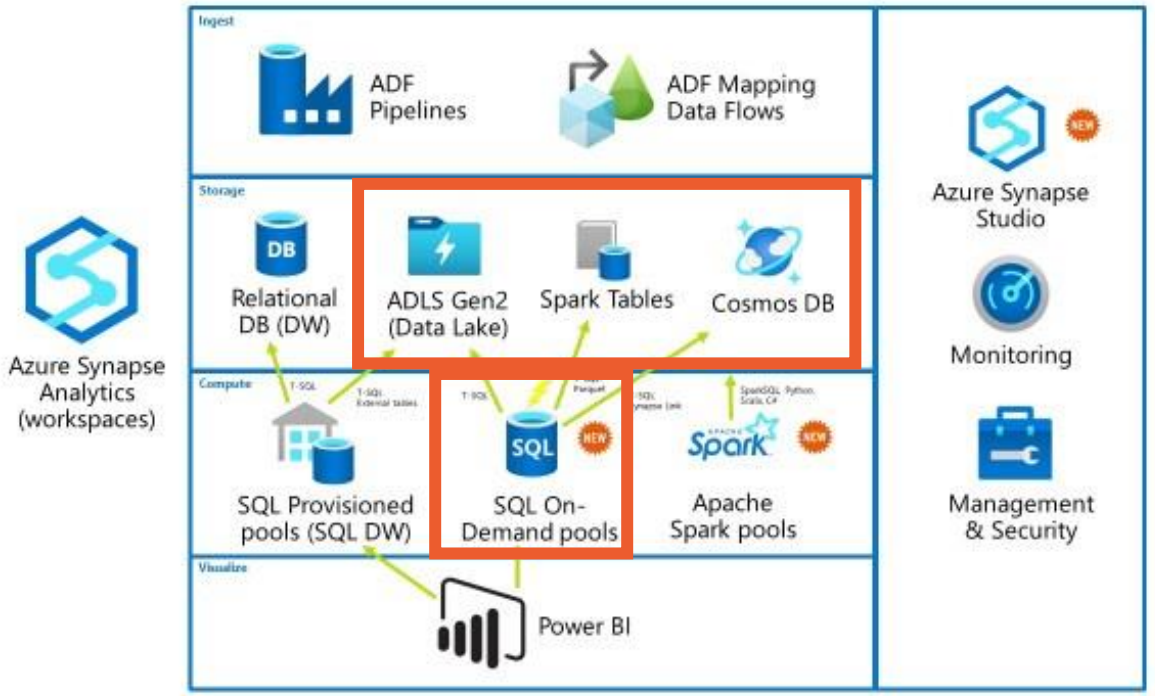
PASS
Data Community
SUMMIT 2021

# Overview

Introducing

# Azure Synapse Analytics

**Cloud Big Data and Analytics platform with several services**




- Dedicated SQL Pools
- Serverless SQL Pools
- Spark Pools
- Pipelines (Data Factory)
- Power BI integration

PASS Data Community SUMMIT 2021

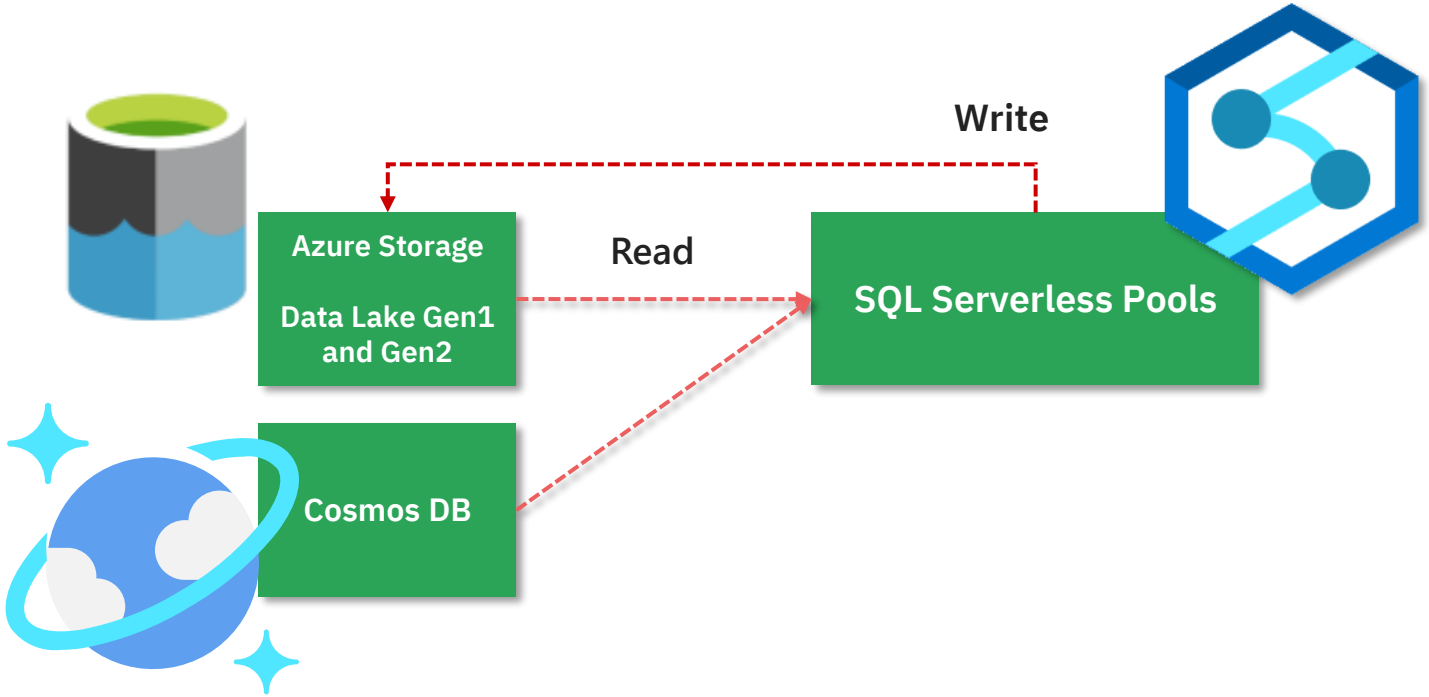
Introducing

# Serverless SQL Pools

## Query external data from Azure Storage, Cosmos DB and Dataverse

### Familiar SQL objects

- Databases
- Stored Procedures
- DMVs
- Views
- External Tables



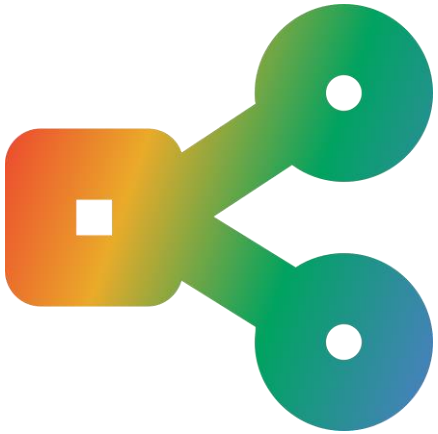

Serverless SQL Pools is a SQL-based service built into Azure Synapse Analytics that allows reading and writing CSV, Parquet and JSON data stored within Azure Storage, Data Lake Gen1 and Gen2, document data in Cosmos DB, and data from the Dataverse

No data is stored within Serverless SQL Pools

The Engine

# Polaris

**Polaris is the engine which underpins the Serverless SQL Pools service**



## No Compute To Configure

No clusters, or compute to scale/manage/automate

## Engine is Always Ready

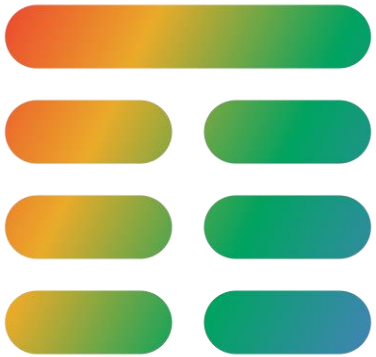
Can query the service at any point.  You are charged for the data processed, not query time/compute level

## Fault Tolerance

Query execution fault tolerance over large datasets

# Serverless SQL Scenarios

**Microsoft state 3 scenarios that Serverless SQL Pools can be useful for**



## Data Exploration

Analyse CSV, Parquet & JSON data stored in Azure Storage using common T-SQL commands. Query Cosmos DB in real-time.  Extract schema information to understand the structure of data.

## Logical Data Warehouse

Create a relational structure over disparate raw data stored in Azure Storage and Cosmos DB without transforming and moving data.  Data is available for immediate querying without further data loading.

## Data Transformation

Data stored in Azure Storage and Cosmos DB can be transformed at scale using T-SQL and the resultant dataset returned to BI tools or loaded into a data store (SQL Database, Dedicated SQL Pools, Data Lake).

What is a

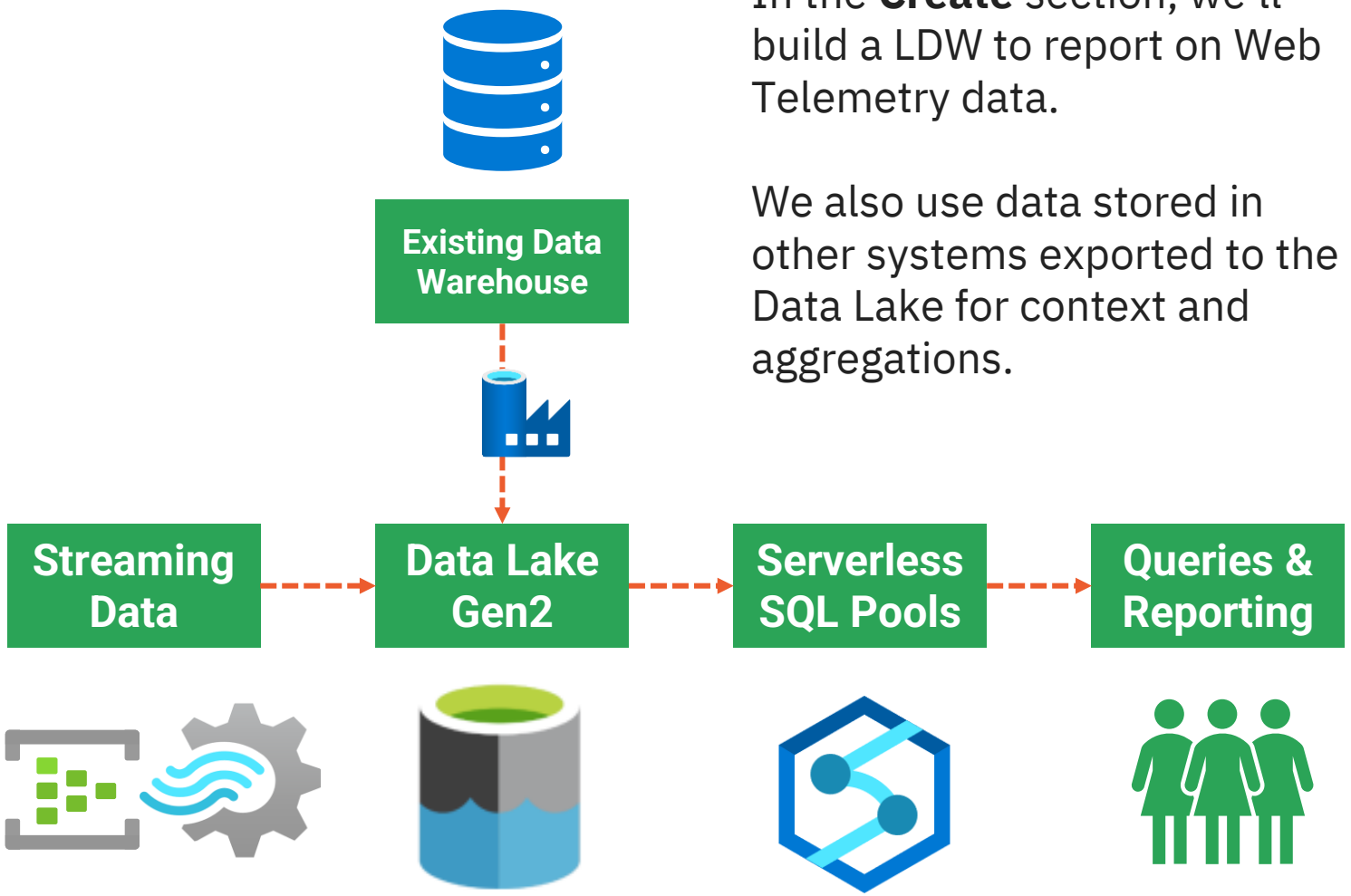
# Logical Data Warehouse?

- Architectural layer “sits” above existing data services
- Ability to view data without effort moving that data
- No/minimal ETL required to surface source data

# Logical Data Warehouse Scenario

Do we just throw away our existing Data Warehouse and use Serverless SQL for everything?

No! We can use Serverless SQL Pools alongside existing data warehouses




In the **Create** section, we'll build a LDW to report on Web Telemetry data.

We also use data stored in other systems exported to the Data Lake for context and aggregations.

# Create SQL Objects

PASS Data Community SUMMIT 2021

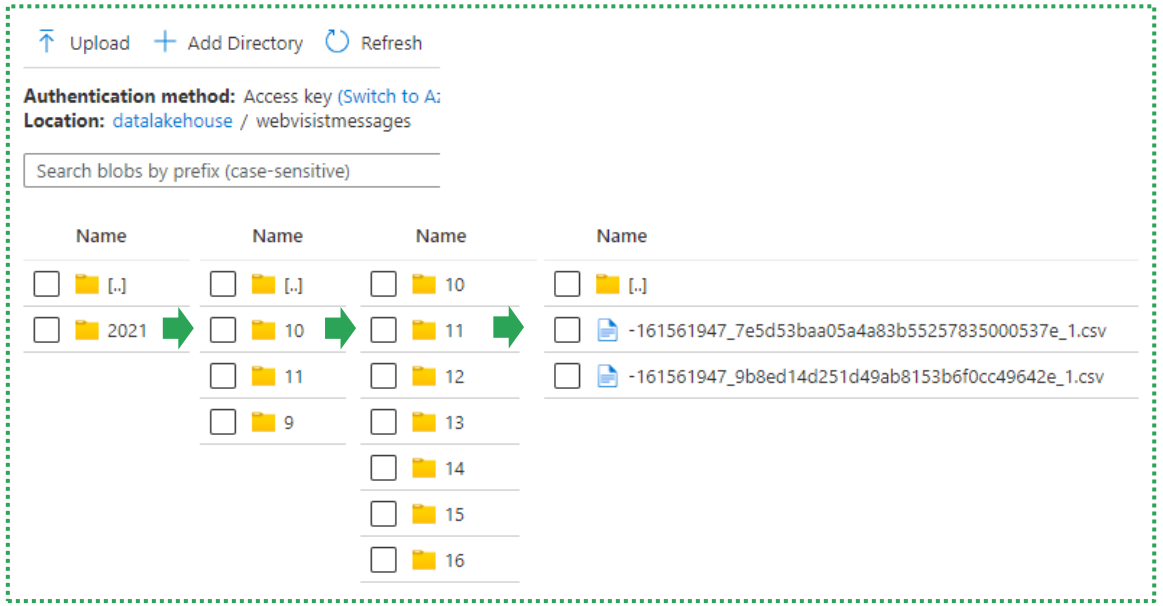
# External Tables vs Views

**We can create either an External Table or a View**

**Which should we use?**

| | External Table | Views |
|---|---|---|
| Must Define Schema (Column Names & Data Types) | Yes | No |
| Supports partition pruning | No (Yes if Spark table) | Yes |
| Can be used to write data back to storage | Yes (using CETAS) | No |
| Nested Types and JSON | No | Yes |

# Data in Azure Storage




Folder structure in Azure Data Lake Gen2

Partitioned by **Year** / **Month** / **Day**

Data generated by:

- Devices sending JSON messages to Event Hubs
- Output as CSV to Data Lake via Streaming Analytics into a partition scheme

| UserID | EventType | EventDateTime | ProductID | URL | Device | SessionViewSeconds | EventProcessedUtcTime | PartitionId | EventEnqueuedUtcTime | PartitionScheme |
|---|---|---|---|---|---|---|---|---|---|---|
| 29640 | browseproduct | 10/10/2021 09:08 | 998 | /product/998 | mobile | 60 | 08/11/2021 10:37 | 0 | 08/11/2021 10:37 | 10/10/2021 |
| 29853 | putinbasket | 10/10/2021 09:08 | 753 | /product/753 | pc | 49 | 08/11/2021 10:37 | 0 | 08/11/2021 10:37 | 10/10/2021 |
| 30071 | putinbasket | 10/10/2021 09:08 | 829 | /product/829 | tablet | 117 | 08/11/2021 10:37 | 0 | 08/11/2021 10:37 | 10/10/2021 |
| 29711 | browseproduct | 10/10/2021 09:08 | 899 | /product/899 | mobile | 98 | 08/11/2021 10:37 | 0 | 08/11/2021 10:37 | 10/10/2021 |
| 29733 | putinbasket | 10/10/2021 09:08 | 985 | /product/985 | tablet | 8 | 08/11/2021 10:37 | 0 | 08/11/2021 10:37 | 10/10/2021 |
| 30047 | browseproduct | 10/10/2021 09:08 | 996 | /product/996 | tablet | 37 | 08/11/2021 10:37 | 0 | 08/11/2021 10:37 | 10/10/2021 |
| 29873 | browseproduct | 10/10/2021 09:08 | 982 | /product/982 | tablet | 67 | 08/11/2021 10:37 | 0 | 08/11/2021 10:37 | 10/10/2021 |
| 29589 | purchasedproduct | 10/10/2021 09:08 | 886 | /product/886 | tablet | 13 | 08/11/2021 10:37 | 0 | 08/11/2021 10:37 | 10/10/2021 |
| 29925 | browseproduct | 10/10/2021 09:08 | 806 | /product/806 | mobile | 66 | 08/11/2021 10:37 | 0 | 08/11/2021 10:37 | 10/10/2021 |
| 29663 | browseproduct | 10/10/2021 09:08 | 915 | /product/915 | mobile | 44 | 08/11/2021 10:37 | 0 | 08/11/2021 10:37 | 10/10/2021 |

**File Preview**
Note that the last 4 columns are from Streaming Analytics

# Create Serverless SQL Pools Database

```
CREATE DATABASE PASSSummit;
```

Create a new Serverless SQL Pools database

```
CREATE EXTERNAL DATA SOURCE ExternalDataSourceDataLake
  WITH (
            LOCATION   = 'https://storageaccount.dfs.core.windows.net/container'
        );
```

Create a connection to the Azure Storage (Data Lake Gen2) account and container

```
CREATE SCHEMA LDW AUTHORIZATION dbo;
```

Create a schema to hold our objects

```
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'dfADFADf454325adsfadsf';
```

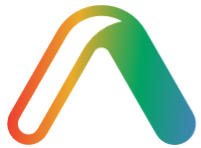
Create master key in database

```
CREATE DATABASE SCOPED CREDENTIAL SynapseUserIdentity
WITH IDENTITY = 'User Identity';
```

Create a credential to allow Active Directory authentication

```
ALTER DATABASE PASSSummit COLLATE Latin1_General_100_BIN2_UTF8;
```

Ensure database supports UTF8



# SELECTing Data

We can SELECT data using the full URL path to the Azure storage account, container, and folder

```
SELECT TOP 10 *
FROM OPENROWSET
(
    BULK 'https://storageaccount.blob.core.windows.net/datalakehouse/webvisistmessages/**',
    FORMAT = 'CSV',
    PARSER_VERSION = '2.0',
    HEADER_ROW = TRUE,
    FIELDTERMINATOR ='|',
    ROWSET_OPTIONS = '{"READ_OPTIONS":["ALLOW_INCONSISTENT_READS"]}'
) as web
```

We can append **/**** to the end of the root folder where the data resides and Serverless will recursively process the data

Results of the SELECT query in Synapse Studio

Results | Messages

View: Table | Chart — Export results

Search

| UserID | EventType | EventDateTime | ProductID | URL | Device | SessionViewSeconds | EventProcessedUtcTime | PartitionId | EventEnqueuedUtcTime | PartitionScheme |
|---|---|---|---|---|---|---|---|---|---|---|
| 29866 | putinbasket | 2021-10-01T09:06:35 | 755 | /product/755 | tablet | 109 | 2021-11-07T21:15:42.9039103Z | 0 | 2021-11-07T21:15:42.7060000Z | 2021/10/1 |
| 29853 | browseproduct | 2021-10-01T09:06:35 | 810 | /product/810 | tablet | 34 | 2021-11-07T21:15:43.2319913Z | 0 | 2021-11-07T21:15:43.0190000Z | 2021/10/1 |
| 29909 | browseproduct | 2021-10-01T09:06:35 | 828 | /product/828 | mobile | 72 | 2021-11-07T21:15:43.5601833Z | 0 | 2021-11-07T21:15:43.3470000Z | 2021/10/1 |

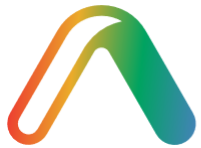
# Inferring Schema

```
EXEC sp_describe_first_result_set N'
    SELECT TOP 10 *
    FROM OPENROWSET
    (
        BULK ''https://storsynapsedemo.blob.core.windows.net/datalakehouse/webvisistmessages/**'',
        FORMAT = ''CSV'',
        PARSER_VERSION = ''2.0'',
        HEADER_ROW = TRUE,
        FIELDTERMINATOR =''|'',
        ROWSET_OPTIONS = ''{"READ_OPTIONS":["ALLOW_INCONSISTENT_READS"]}''
    ) as web';
```

| column_ordinal | name | is_nullable | system_type_id | system_type_name |
|---|---|---|---|---|
| 1 | UserID | True | 127 | bigint |
| 2 | EventType | True | 167 | varchar(8000) |
| 3 | EventDateTime | True | 167 | varchar(8000) |
| 4 | ProductID | True | 127 | bigint |
| 5 | URL | True | 167 | varchar(8000) |
| 6 | Device | True | 167 | varchar(8000) |
| 7 | SessionViewSeconds | True | 127 | bigint |
| 8 | EventProcessedUtcTime | True | 167 | varchar(8000) |
| 9 | PartitionId | True | 127 | bigint |
| 10 | EventEnqueuedUtcTime | True | 167 | varchar(8000) |
| 11 | PartitionScheme | True | 167 | varchar(8000) |

As we've seen, we can query the data simply using SELECT * without defining a schema.

However, if we run a schema inference procedure we can see data types that may not be optimal such as **bigint** and **varchar(8000)**



PASS
**Data Community**
**SUMMIT** 2021

# Creating Views Over Raw Data

```
CREATE VIEW LDW.vwRawWebTelemetry
AS
SELECT
    UserID,
    EventType,
    ProductID,
    [URL],
    Device,
    SessionViewSeconds,
    CAST(rwt.filepath(1) AS SMALLINT) AS FilePathYear,
    CAST(rwt.filepath(2) AS TINYINT) AS FilePathMonth,
    CAST(rwt.filepath(3) AS TINYINT) AS FilePathDay
FROM
OPENROWSET
(
    BULK 'webvisistmessages/*/*/*/*.csv',
    DATA_SOURCE = 'ExternalDataSourceDataLake',
    FORMAT = 'CSV',
    PARSER_VERSION = '2.0',
    HEADER_ROW = TRUE,
    FIELDTERMINATOR ='|',
    ROWSET_OPTIONS = '{"READ_OPTIONS":["ALLOW_INCONSISTENT_READS"]}'
)
WITH
(
    UserID INT,
    EventType VARCHAR(20),
    ProductID SMALLINT,
    [URL] VARCHAR(50),
    Device VARCHAR(10),
    SessionViewSeconds INT
)
AS rwt
```

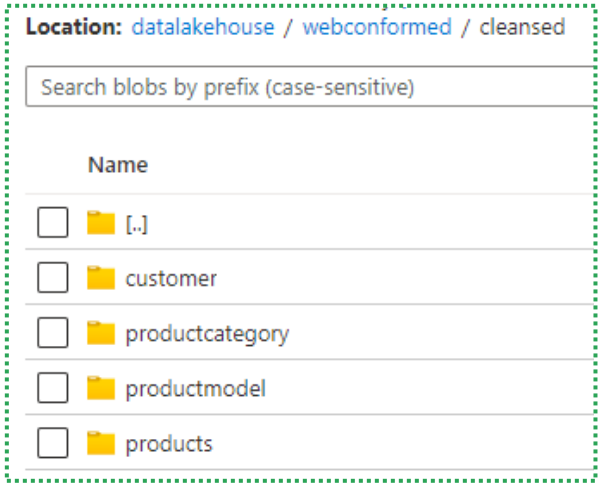
We can now create Views over the data stored in the Data Lake

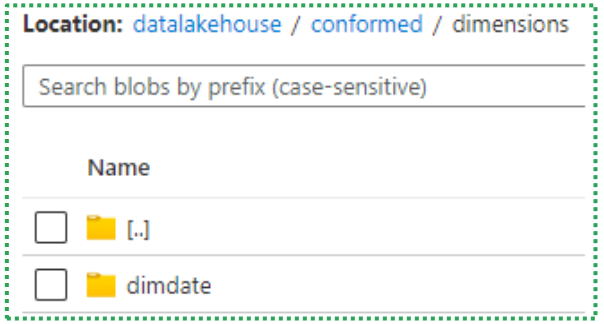
In the SQL we are defining which column we would like to select from the CSV by using the name of the column in the WITH command

We assign appropriate data types to ensure efficiency

Note the **filepath()** function, we are exposing the folder structure as columns within the view. We can use these columns to “partition prune”

# Creating Views Over Exported Data





```
CREATE VIEW LDW.vwCustomers
AS
SELECT
    CustomerID,
    Title,
    FirstName,
    MiddleName,
    LastName,
    CompanyName,
    SalesPerson,
    EmailAddress,
    Phone
FROM OPENROWSET
(
    BULK 'webconformed/cleansed/customer/01',
    DATA_SOURCE = 'ExternalDataSourceDataLake',
    FORMAT = 'Parquet'
) AS cst
```

We can use data from other systems to augment our incoming data

Data from our existing data systems

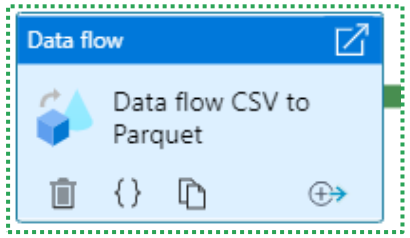
5 Views created:

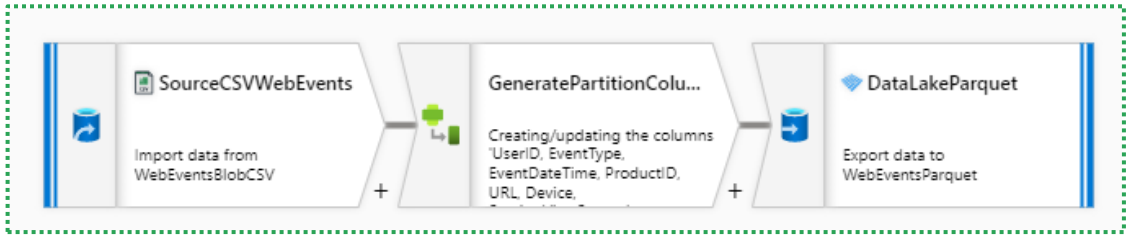
- Customer
- Product
- Product Category
- Product Model
- Date (from a global Dimension)

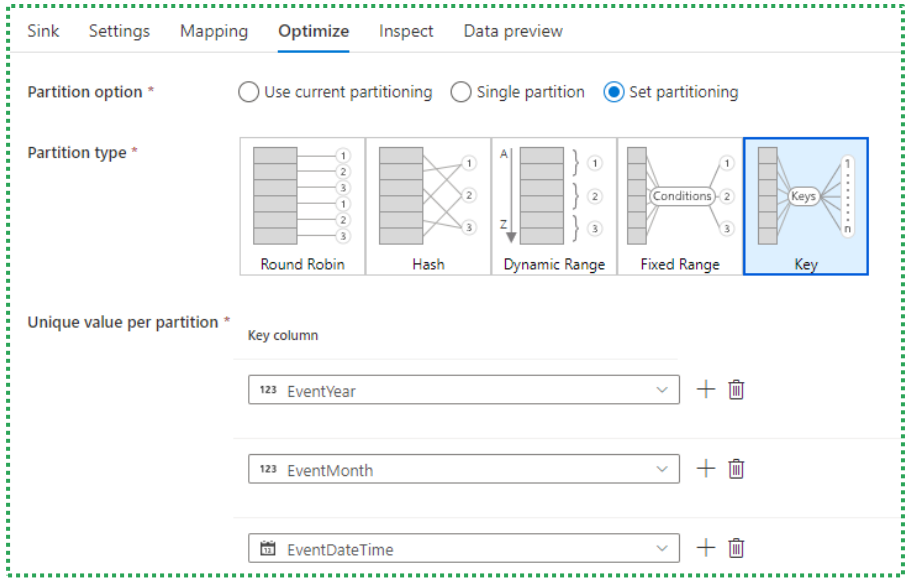
# Pipeline for CSV to Parquet Data Transformation

Source web telemetry data is in CSV, not particular optimal

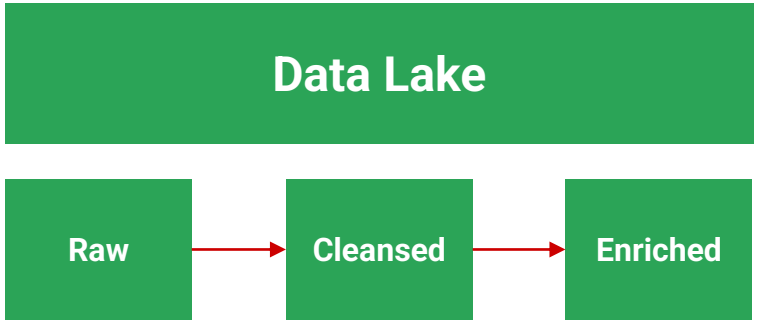
We can transform that data whilst keeping it within the data lake and accessible from other data services as well.




## Common Data Lake Pattern:

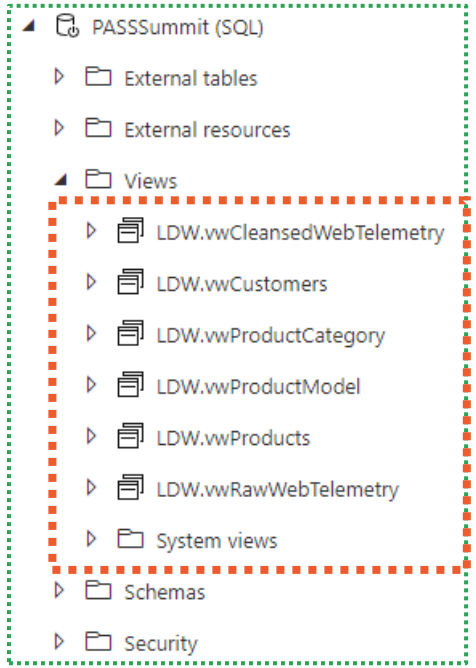



PASS Data Community SUMMIT 2021

Viewing

# Schema Objects

**Once we have created the Views, we can see these Views in the Data tab in Synapse Studio**




Other client tools such as **SSMS** and **Azure Data Studio** will show a similar structure

# Data Analysis Queries

We can now issue standard SQL queries, joining Views together to aggregate data

```
SELECT  DD.[Year] AS EventYear,
        DD.MonthName AS EventMonth,
        COUNT(WT.UserID) AS EventCount,
        SUM(WT.SessionViewSeconds) AS EventSessionViewSeconds
FROM LDW.vwCleansedWebTelemetry WT
LEFT JOIN LDW.vwDimDate DD ON DD.[Date] = WT.FilePathDate
WHERE   WT.FilePathYear = 2021
        AND WT.FilePathMonth = 10
GROUP BY DD.[Year],
         DD.MonthName
```

Results | Messages

View: Table | Chart  Export results

Search

| EventYear | EventMonth | EventCount | EventSessionViewSeconds |
|---|---|---|---|
| 2021 | October | 72004 | 4467267 |

We use the columns that are returning the **filepath function** results as a filter to partition prune

FilePathYear
FilePathMonth

PASS Data Community SUMMIT 2021

## Automating
# View Creation

**We can leverage dynamic SQL to create Views by passing in parameters to a custom stored procecure**

```
DECLARE @Location NVARCHAR(200) = 'sourcedatasystem/Sales_Customers/*.csv',
        @ViewName NVARCHAR(255) = 'LDW.vwSalesCustomers'

INSERT INTO #t EXEC sp_describe_first_result_set N'
   SELECT * FROM
    OPENROWSET
    (
        BULK ''' + @Location + ''',
        DATA_SOURCE = ''ExternalDataSourceDataLake'',
        FORMAT = ''CSV'',
        PARSER_VERSION = ''2.0'',
        HEADER_ROW = TRUE,
        FIELDTERMINATOR =''|''
    ) AS fct'

DECLARE @mincol INT,
        @maxcol INT,
        @sqltext NVARCHAR(4000)

SELECT @mincol = MIN(column_ordinal) FROM #t
SELECT @maxcol = MAX(column_ordinal) FROM #t

SET @sqltext = 'CREATE VIEW '+ @ViewName + CHAR(10) + CHAR(13) + 'AS' + CHAR(10) + CHAR(13) + '
SELECT '

WHILE @mincol <= @maxcol
BEGIN
    SELECT @sqltext = @sqltext + CONCAT('CAST(',[name],' AS ',system_type_name,') AS ',[name],C
ASE WHEN @mincol = @maxcol THEN '' ELSE ',' END)
    FROM #t
    WHERE column_ordinal = @mincol

    SET @mincol = @mincol +1
END
```

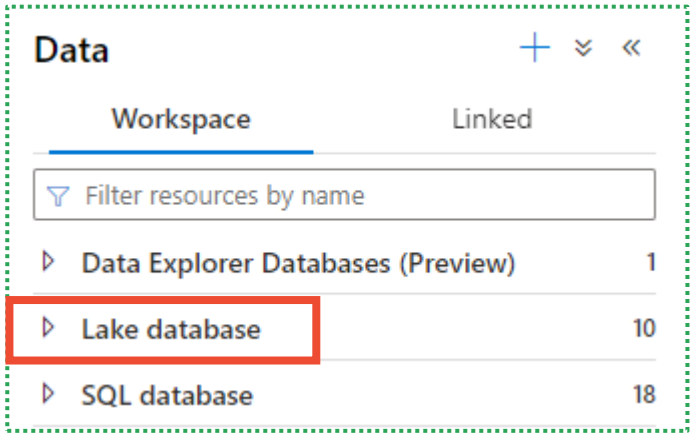
https://github.com/datahai/serverlesssqlpooltools

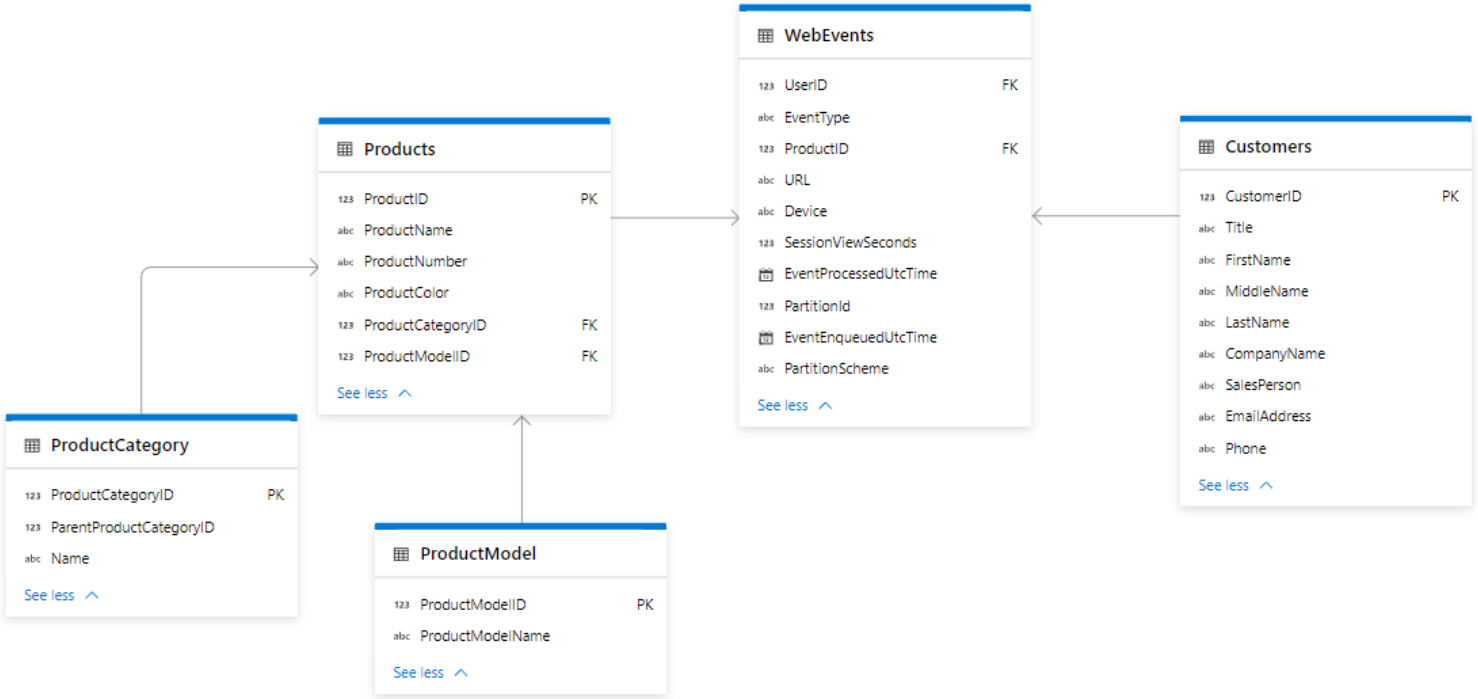
Preview Feature

# Database Designer

**New feature within Synapse Studio to create a Data Model**

**Introduces a new database called the “Lake Database”**




You can create External Tables that can be shared with both Serverless SQL Pools and Spark Pools

Allows for the creation of Partition Schemes

Ability to define table relationships, primary, and foreign keys

Power BI can connect and import the relationships into the Power BI data model (coming soon!)

# Discover

PASS
**Data Community**
**SUMMIT** 2021

# Connecting to
# Serverless SQL Pools

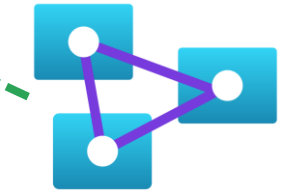
**Serverless SQL Pools has a separate endpoint which cother data services can connect to and issue SQL statements**

| | | |
|---|---|---|
| Networking | : | Show firewall settings |
| Primary ADLS Gen2 acco... | : | https:// .dfs.core.windows.net |
| Primary ADLS Gen2 file s... | : | users |
| SQL admin username | : | |
| SQL Active Directory ad... | : | andycutler@datahai.co.uk |
| Dedicated SQL endpoint | : | .sql.azuresynapse.net |
| Serverless SQL endpoint | : | -ondemand.sql.azuresynapse.net |
| Development endpoint | : | emodh.dev.azuresynapse.net |

Serverless SQL Pools
Endpoint

SQL Server Management Studio

Azure Data Studio

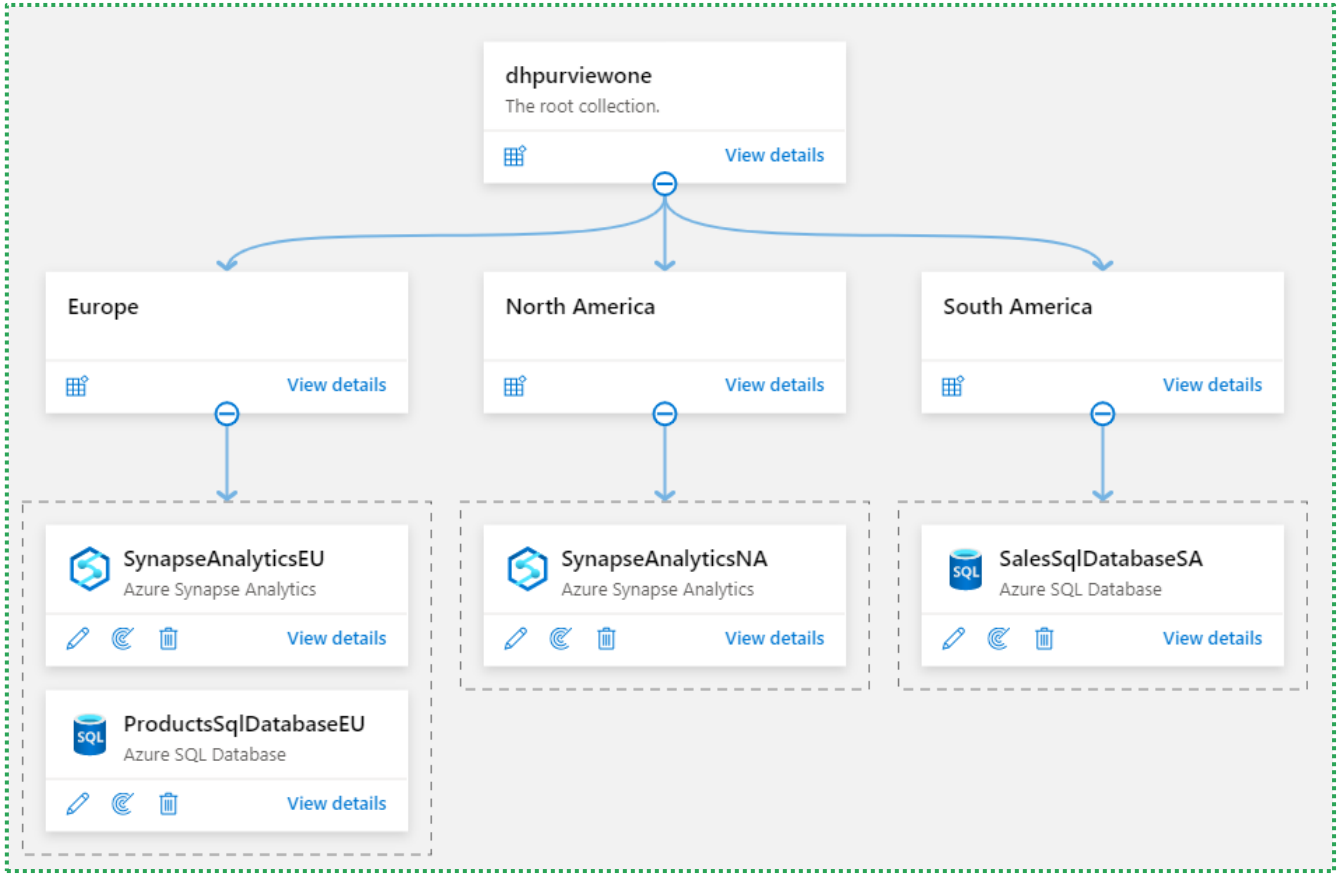


Azure Analysis Services

Power BI (Datasets & Dataflows)

## Data Discovery with
# Azure Purview

**Data Governance service to bring together data services into a holistic “one view”**

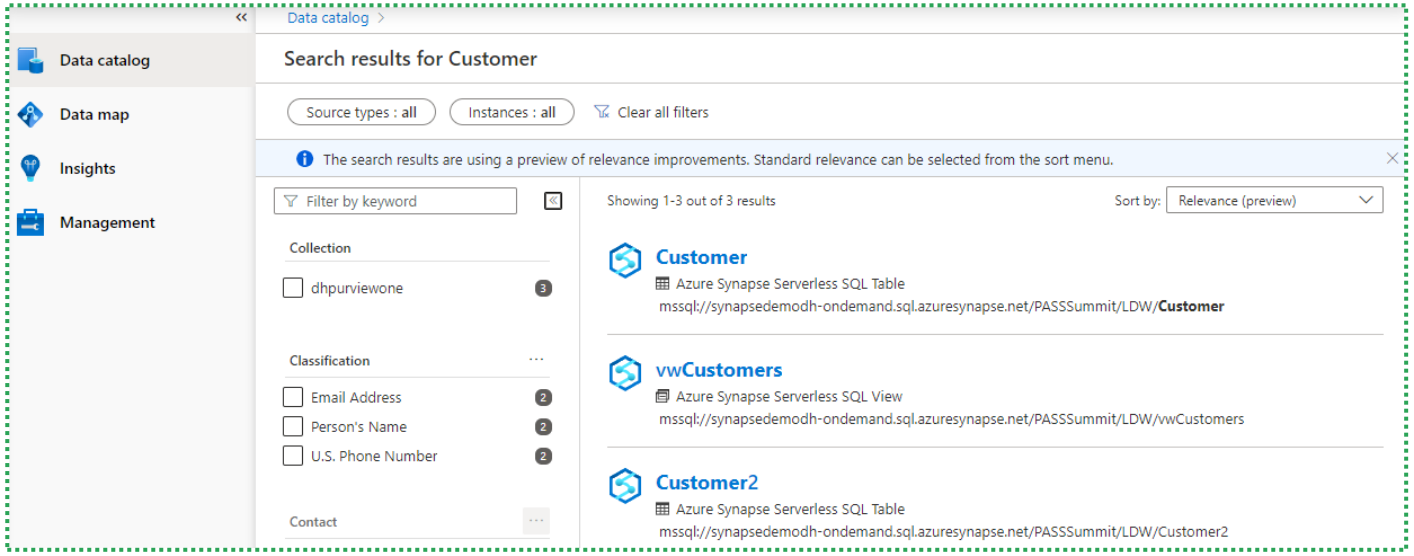


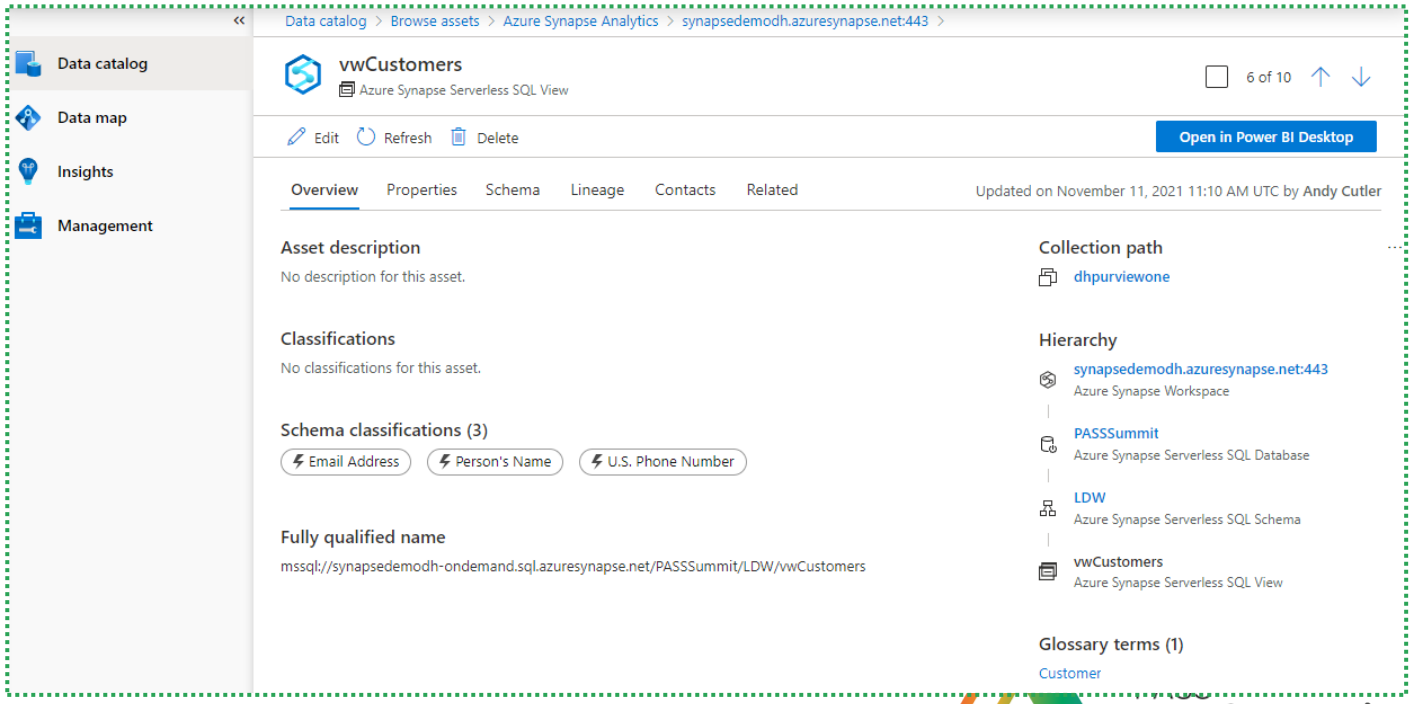

- Add Purview Managed Identity as a Reader to Synapse workspace
- Add Purview Managed Identity as a Storage Blob Data Reader to Azure Storage
- Add Purview Managed Identity as a user to Serverless SQL Pools databases:
  - Run on Master:
  ```
  CREATE LOGIN [dhpurviewone] FROM EXTERNAL PROVIDER;
  ```
  - Run on database:
  ```
  CREATE USER [dhpurviewone] FOR LOGIN [dhpurviewone];

  ALTER ROLE db_datareader ADD MEMBER [dhpurviewone];
  ```

# Data Discovery with
# Azure Purview

Users of Purview can now search for and see information about the Serverless SQL Pools objects including External Tables, Views, and the Columns defined.

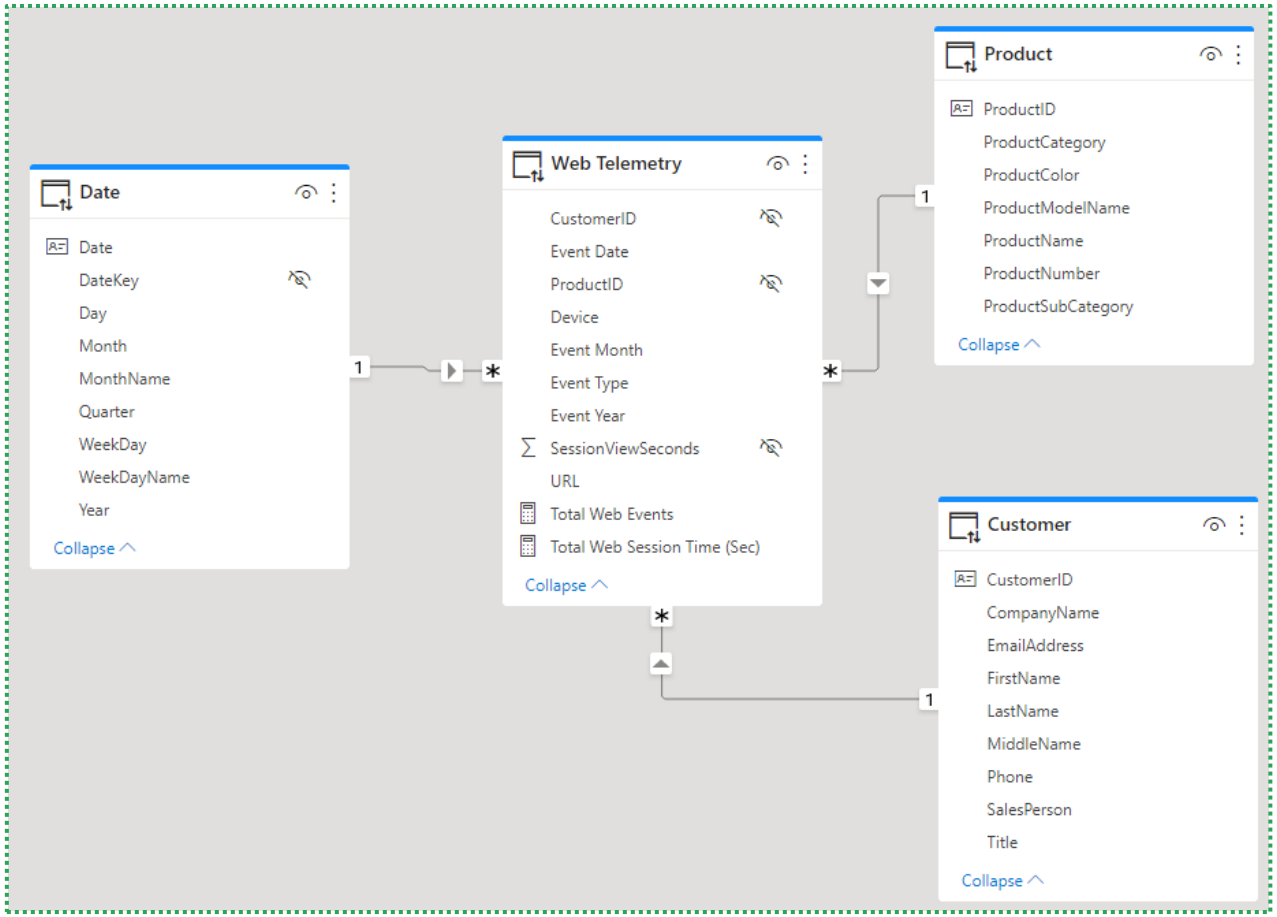


## Analytics using
# Power BI

Connect in DirectQuery for large datasets.

Be mindful of **data processed**, use filters on the **filepath** columns




Be mindful of DirectQuery on large data sources

Leverage **Power BI Aggregations** on large datasets in the Data Lake

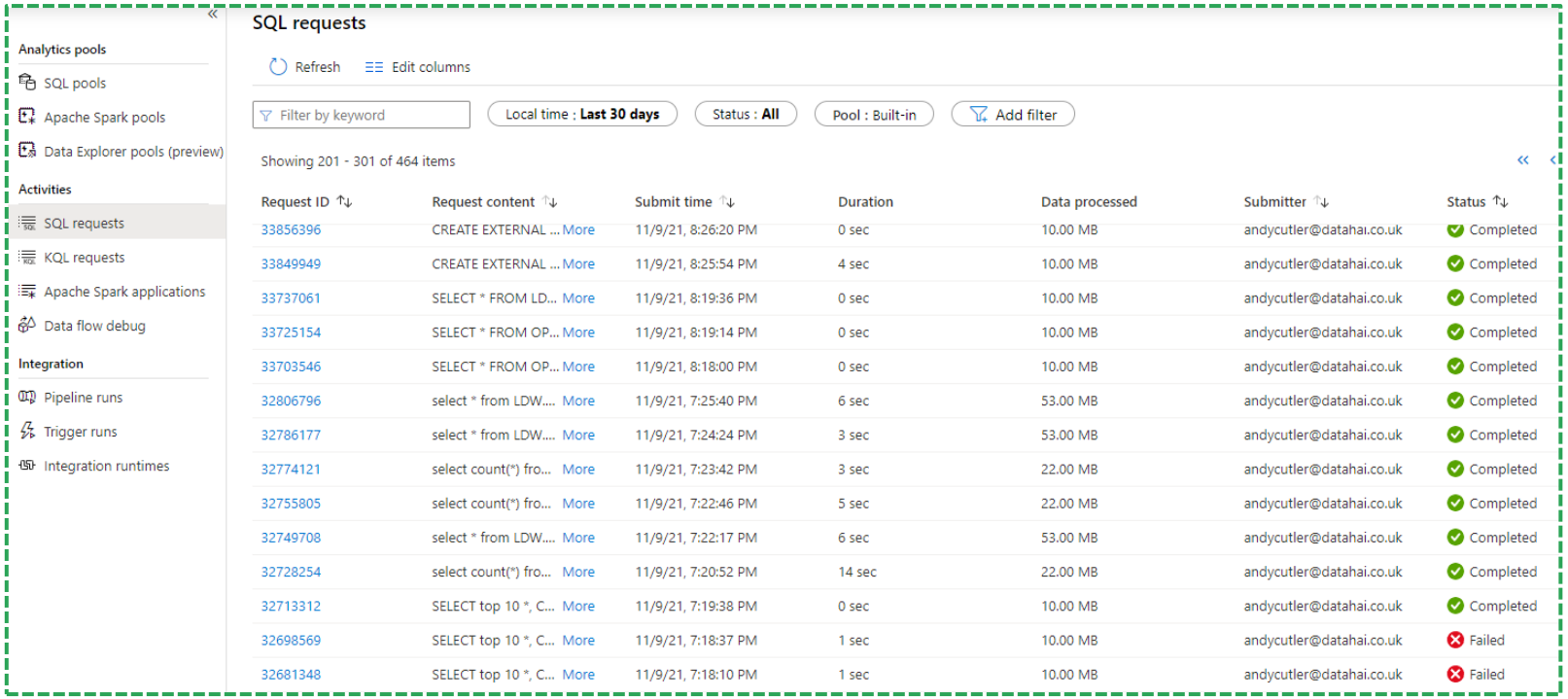
Set **Query Reduction** settings on slicers and filters

PASS Data Community SUMMIT 2021

# Monitoring

Monitoring SQL requests to determine Duration and Data processed

We can also set limits to how much data can be processed




How much data has been processed?

```
sys.dm_external_data_processed
```

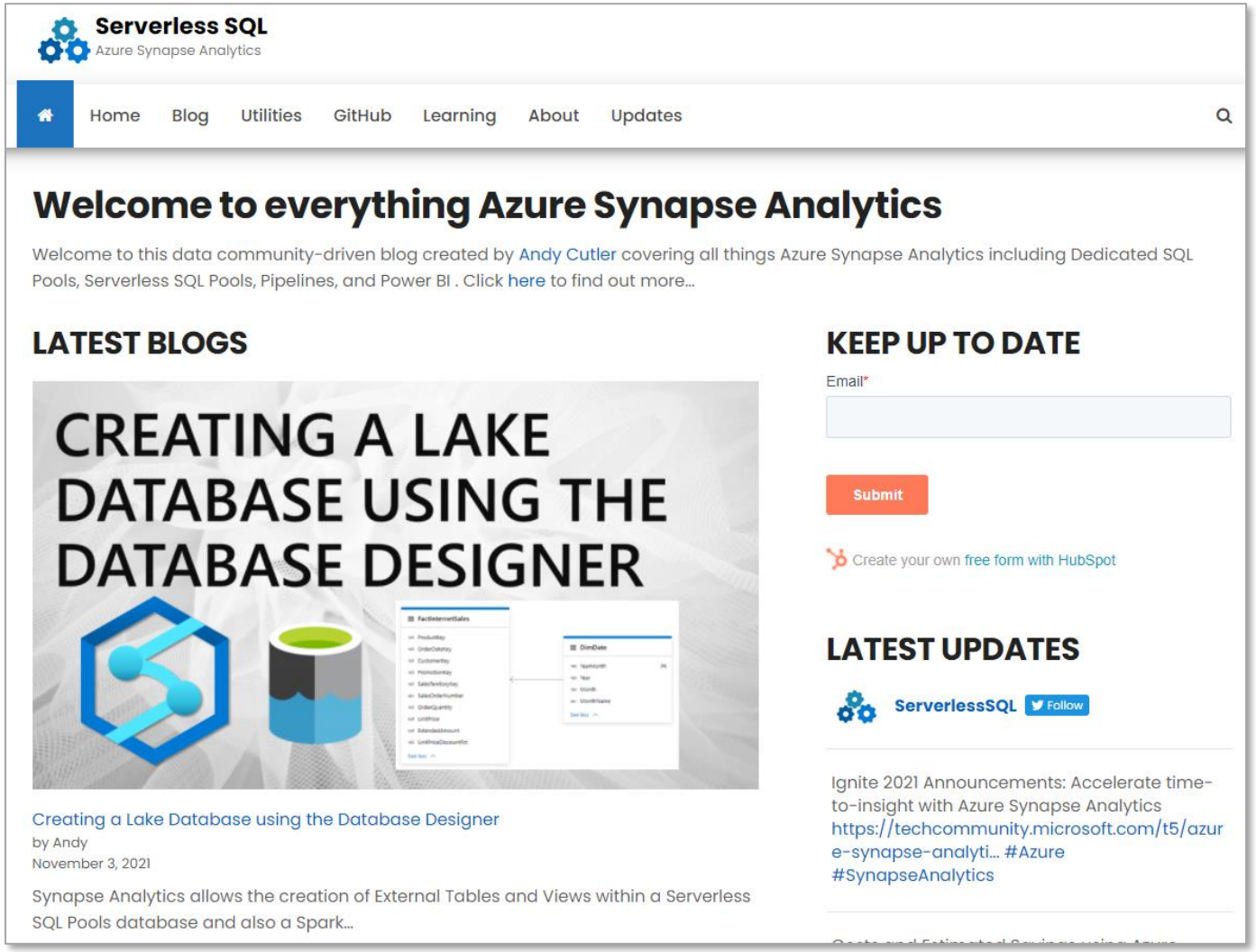
Show data processed limits for Daily, Weekly, and Monthly

```
sys.configurations WHERE [name] LIKE 'Data processed %';
```

# Demo

# Further Reading

If you'd like to read more on Serverless SQL Pools and Synapse Analytics in general, I blog at...




www.serverlesssql.com

PASS Data Community SUMMIT 2021

# Thank you

## Evaluate this session at:

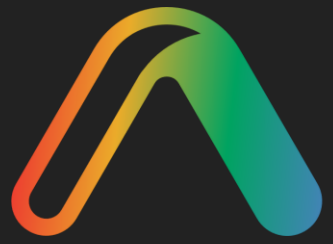
www.PASSDataCommunitySummit.com/evaluation

**Andy Cutler (he/him)**

@MrAndyCutler

Datahai.co.uk

ServerlessSQL.com



PASS
**Data Community**
**SUMMIT** 2021