SQL Data Warehouse
Loading Data

Chris Testa-O'Neill
Features Engineer
Analytics and Data Science Team

Microsoft

This session is brought to you by Microsoft's Analytics and Data Science Team.

Agenda

The "load user"
Loading Data
Using PolyBase
Importing and Exporting Data Loads
Monitoring data loads
Azure Data Factory Integration
Consuming data from SQL DW
Recommendations

The agenda for this session includes:

- Load user
- Loading data
- Using PolyBase
- Importing and Exporting data
- Monitoring data loads
- Azure Data Factory Integration
- Recommendations

The “load user”

This section of the course will cover:

- Creating a dedicated user for data loading.
- Benefits of the load user.
- Creating a “load user” login.
- Resource Class roles.
- Creating a database user.
- Identifying users with elevated users.
- Memory Management.

Why create a dedicate user for data loading?

Post Provisioning

1 Login

Service admin

Full “sa” permissions

Fixed memory assignment

It is best practise to create a dedicated database user for the purpose of loading data. This is typically the first activity performed after creating an Azure SQL Data Warehouse and creating a database.

What benefits do I get?

More granular permissions model
Flexible memory management
Easier to identify requests

A dedicated user provides the following benefits

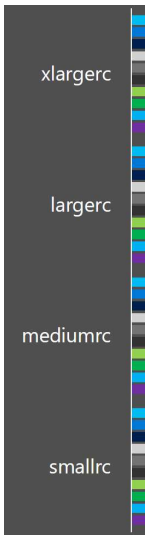
- More granular permissions model
- Flexible memory management
- Easier to identify requests

Create Login (master)

```
-- Run this against the master database

CREATE LOGIN SQLDWLoad WITH PASSWORD = 'SQLB1ts!';
CREATE USER SQLDWDBLoad for LOGIN SQLDWLoad;


EXEC sp_addrolemember 'loginmanager','SQLDWDBLoad';
EXEC sp_addrolemember 'dbmanager','SQLDWDBLoad';
```

Create a Server login first, add the login as a database user and then add the login to the loginmanager and dbmanager roles to give the appropriate access at the SQL Server level.

Resource class roles




```
SELECT  ro.[name]           AS [db_role_name]
FROM    sys.database_principals ro
WHERE   ro.[type_desc]      = 'DATABASE_ROLE'
AND     ro.[is_fixed_role]  = 0
;
```

Resource Class database roles impact the concurrency and memory limits within an Azure SQL Data Warehouse. You can identify the Resource Class Roles available in a database with the query in the slide.

https://docs.microsoft.com/en-gb/azure/sql-data-warehouse/sql-data-warehouse-develop-concurrency#concurrency-limits

Create user (user db)

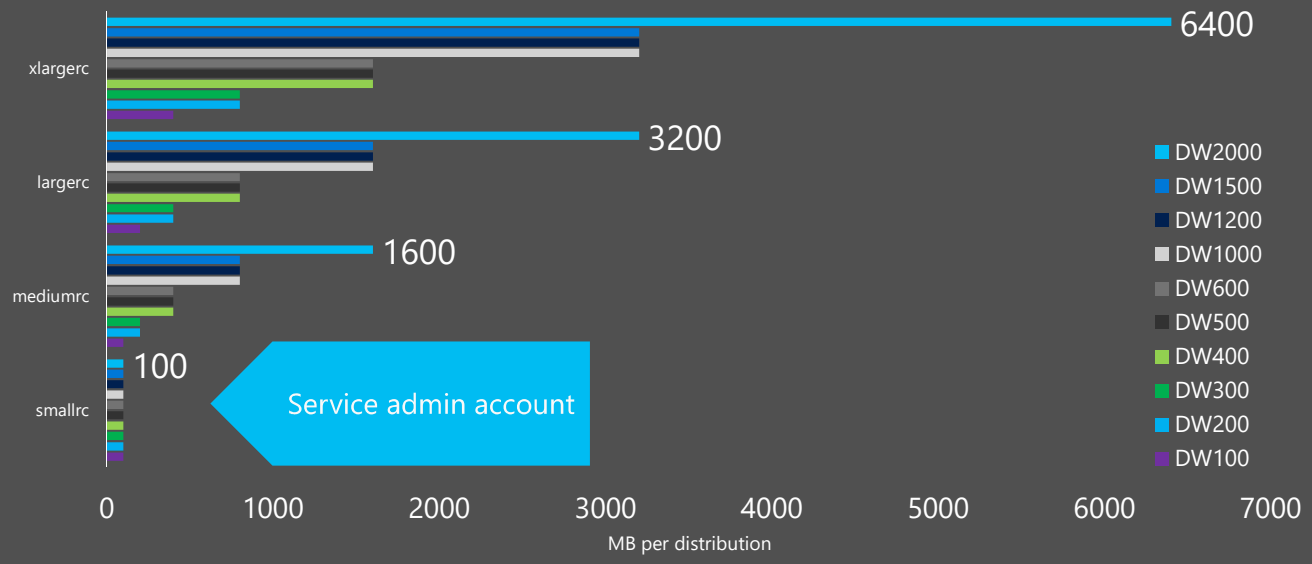
```
-- Run this against the user defined database
CREATE USER SQLDWDBLoad for LOGIN SQLDWLoad
;
GRANT CONTROL ON DATABASE::EquityDB TO SQLDWDBLoad
;

--use the select query to determine the role assignment
SELECT  r.[name] AS role_principal_name
,       m.[name] AS member_principal_name
FROM    sys.database_role_members rm
JOIN    sys.database_principals AS r    ON rm.[role_principal_id]     = r.[principal_id]
JOIN    sys.database_principals AS m    ON rm.[member_principal_id]   = m.[principal_id]
WHERE   r.[name] IN ('mediumrc','largerc', 'xlargerc')
;


EXEC sp_addrolemember 'mediumrc','SQLDWDBLoad'
;
```

To add a user to a resource class role.

First grant the database user control of the database.
Optionally, you can view which database user account are members of the resource class roles.
The important part is to add the database user to the resource class role that meets your workload requirements

https://docs.microsoft.com/en-gb/azure/sql-data-warehouse/sql-data-warehouse-develop-concurrency#resource-classes

Memory Management (MB per distribution)



Difference resource classes can determine the amount of memory that is granted to each distribution in an Azure SQL Data Warehouse.

https://docs.microsoft.com/en-gb/azure/sql-data-warehouse/sql-data-warehouse-develop-concurrency#memory-allocation

Loading

This section of the course will cover:

- Loading options
- Single gated clients
- Single gated clients parallelised
- Parallel Loading with PolyBase
- Demo: Loading data with a single gated client

Loading options

Parallel

PolyBase

Azure Data Factory

Single Gated Client

bcp / Insert Bulk

SQLBulkCopy

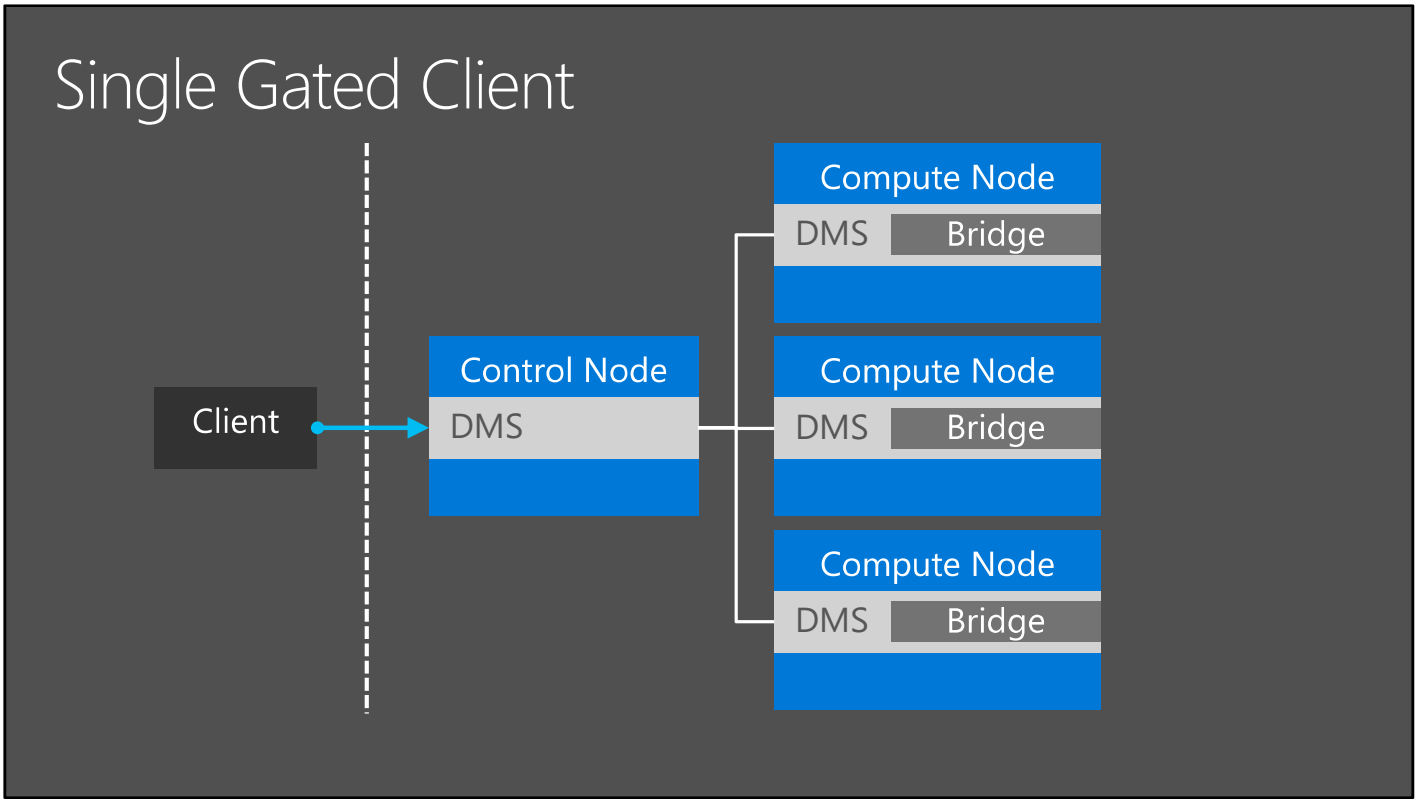
SSIS (data flow)

Azure Data Factory

There are a wide range of technologies that can be used to load data into the Data Warehouse.

https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-overview-load

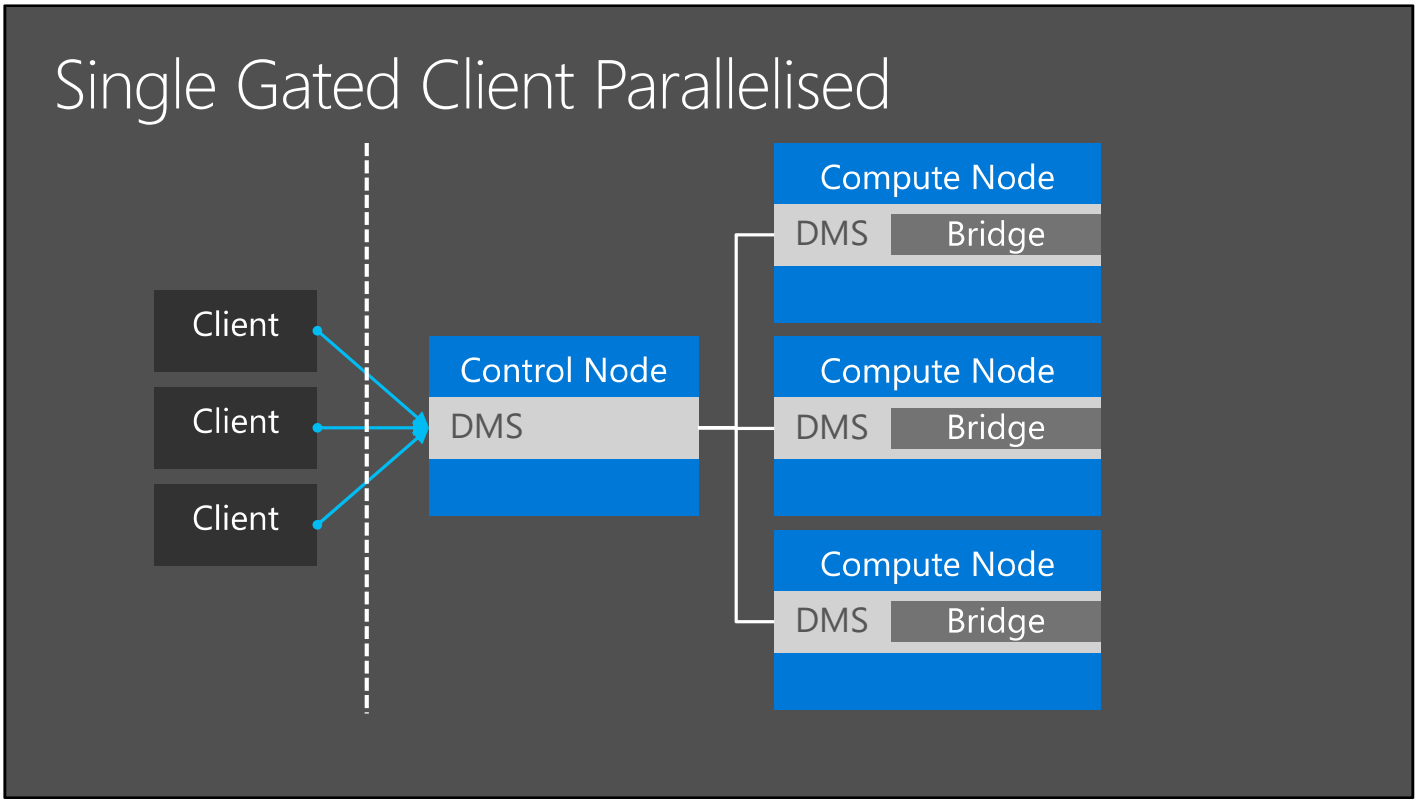
Single Gated Client



Single Gated Clients can include:

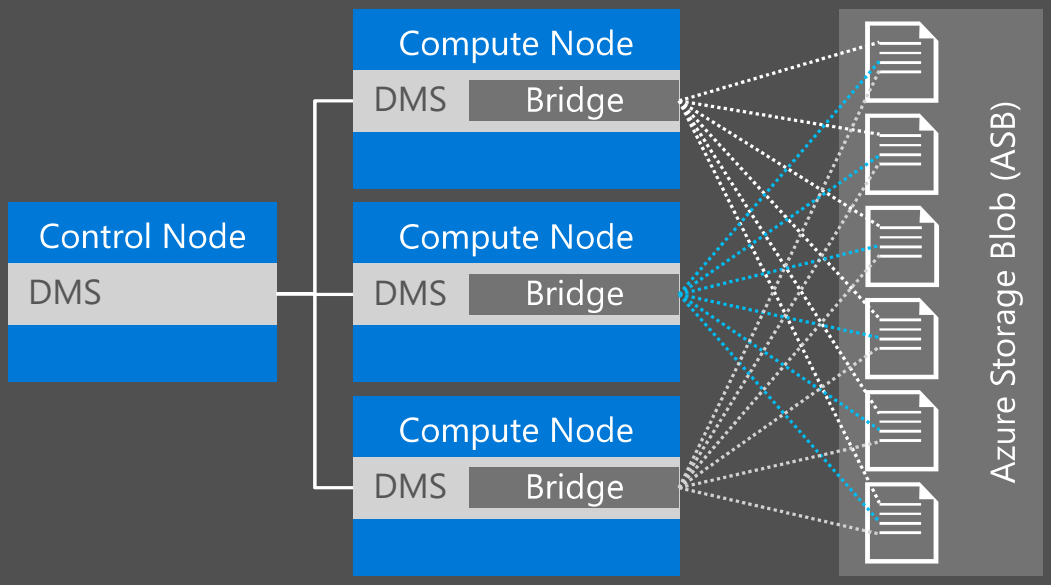
- bcp / Insert Bulk
- SQLBulkCopy
- SSIS (data flow)
- Azure Data Factory

Single Gated Client Parallelised



Single gated clients can operate in parallel against a control node in an Azure SQL Data Warehouse.

Parallel Loading with PolyBase



For fast data loads, PolyBase should be used to meet this objective. This takes full advantage of parallelism for fast loads.

https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-best-practices#use-polybase-to-load-and-export-data-quickly

Demo: Loading data with a single gated client.

SSIS will be used to demonstrate how a single gated client can load data into the Data Warehouse.

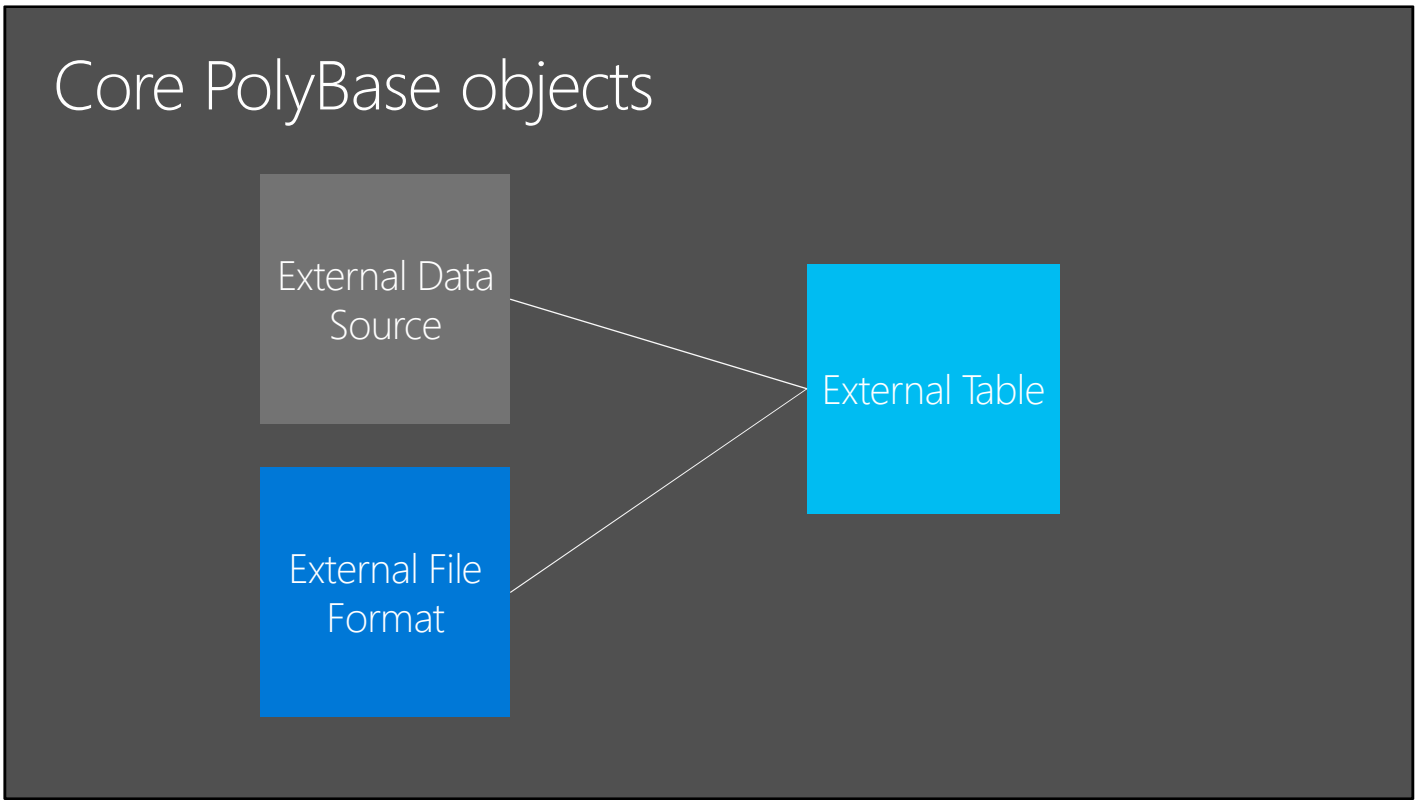
1. Load the Simple SSIS Package.dtsx in SQL Server Data Tools
2. Open the package, and change the connection managers to point to the EquityDB in the local instance of your SQL Server, and point the Azure SQL DW connection to the EquityDB in the Azure Data Warehouse
3. Run the package

PolyBase

This section of the course will cover:

- Core PolyBase objects
- External tables
- External table metadata
- Create External Table
- External Data Sources
- External File Formats

Core PolyBase objects



There are a number of key PolyBase objects to create

https://msdn.microsoft.com/en-us/library/mt652315.aspx

External Tables

Metadata used to describe external data

Enables data access outside the database

Never holds data

Does not delete data when dropped

Behaviour of an external table is very similar to Hive external tables

External Tables never hold data, they are metadata tables that describe a structure over semi or unstructured data

https://msdn.microsoft.com/en-us/library/dn935021.aspx

External table metadata

sys.external_tables
sys.tables

You can query information about external table metadata just like sql server tables

Create External Table

```
CREATE EXTERNAL TABLE [asb].[FactOnlineSales]
([ProductKey]    int      NOT NULL
,[StoreKey]      int      NOT NULL
,[DateKey]       int      NOT NULL
,[CustomerKey]   int      NOT NULL
,[PromotionKey]  int      NOT NULL
,[SalesQuantity] int      NOT NULL
,[UnitPrice]     money    NOT NULL
,[SalesAmount]   money    NOT NULL
)
```

Example creating an external table using T-SQL

https://msdn.microsoft.com/en-us/library/dn935021.aspx

External Tables (cont)

```
WITH
(LOCATION='wasbs://filepath_or_directory'
,DATA_SOURCE          = MyDataSourceName
,FILE_FORMAT          = MyFileFormatName
,REJECT_TYPE          = VALUE
,REJECT_VALUE         = 0
,REJECT_SAMPLE_VALUE  = 1000
)
;
```

The PolyBase aspects are contained within the WITH clause of the CREATE EXTERNAL TABLE statement

https://msdn.microsoft.com/en-us/library/dn935021.aspx

External Data Source

```
CREATE EXTERNAL DATA SOURCE MyAzureDataSource
WITH
(TYPE          = HADOOP
,LOCATION      =
'wasb[s]://[container@]account_name.blob.core.windows.net/path'
)
;
```

An External Data Source provides the connection information required for the external table

https://msdn.microsoft.com/en-us/library/dn935022.aspx

External File Format - ORC

```
CREATE EXTERNAL FILE FORMAT ORCFileFormat
WITH
(FORMAT_TYPE        =    ORC
,DATA_COMPRESSION   =
'org.apache.hadoop.io.compress.DefaultCodec'
| 'org.apache.hadoop.io.compress.SnappyCodec'
)
;
```

An external file format describes the format of the file being queried by the external table.

This is an example of an Optimized Record Column format

https://msdn.microsoft.com/en-us/library/dn935022.aspx

External File Format - Parquet

```
CREATE EXTERNAL FILE FORMAT ParquetFileFormat
WITH
(FORMAT_TYPE        =     PARQUET
,DATA_COMPRESSION   =
'org.apache.hadoop.io.compress.SnappyCodec'
| 'org.apache.hadoop.io.compress.GzipCodec'
)
;
```

An external file format describes the format of the file being queried by the external table.

This is an example of an Parquet file format

https://msdn.microsoft.com/en-us/library/dn935022.aspx

Hive Data Type Mapping

Missing Types in ORC / Parquet

SQL Type	Recommendation
DATE	Use TIMESTAMP

Different Ranges

Hive Type	Hive	SQL
TINYINT	-128 to +127	0 to 255
TIMESTAMP	1970 to 2039	0001-01-01 to 9999-12-31

Reference:
https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Types

Be mindful at times there can be data type mismatch to account for.

External File Format – Delimited Text

```
CREATE EXTERNAL FILE FORMAT MyTextFileFormat
WITH
(FORMAT_TYPE        =   DELIMITEDTEXT
,FORMAT_OPTIONS     (   FIELD_TERMINATOR= '|'
                    ,       STRING_DELIMITER= ','
                    ,       DATE_FORMAT= 'yyyy-MM-dd'
                    ,       USE_TYPE_DEFAULT= TRUE
                    )
,DATA_COMPRESSION   =
'org.apache.hadoop.io.compress.DefaultCodec'
|  'org.apache.hadoop.io.compress.GzipCodec'
)
;
```

An external file format describes the format of the file being queried by the external table.

This is an example of an Delimited Text format

https://msdn.microsoft.com/en-us/library/dn935022.aspx

Delimited text guidance

UTF-8 encode your files

Row delimiter is not configurable

No row delimiters in strings

GZIP not Winzip for compression

Delimiter	Description
\r	Carriage return {CR}
\n	Line Feed {LF}
\r\n	Carriage return linefeed {CR}{LF}

This page is left intentionally blank.

DATE_FORMAT

No DATE_FORMAT in EFF

DateTime: 'yyyy-MM-dd HH:mm:ss'

SmallDateTime: 'yyyy-MM-dd HH:mm'

Date: 'yyyy-MM-dd'

DateTime2: 'yyyy-MM-dd HH:mm:ss'

DateTimeOffset: 'yyyy-MM-dd HH:mm:ss'

Time: 'HH:mm:ss'

DATE_FORMAT in EFF

Same format used for all date typed fields

Cannot specify multiple date formats in the same EFF

One external file = one file format

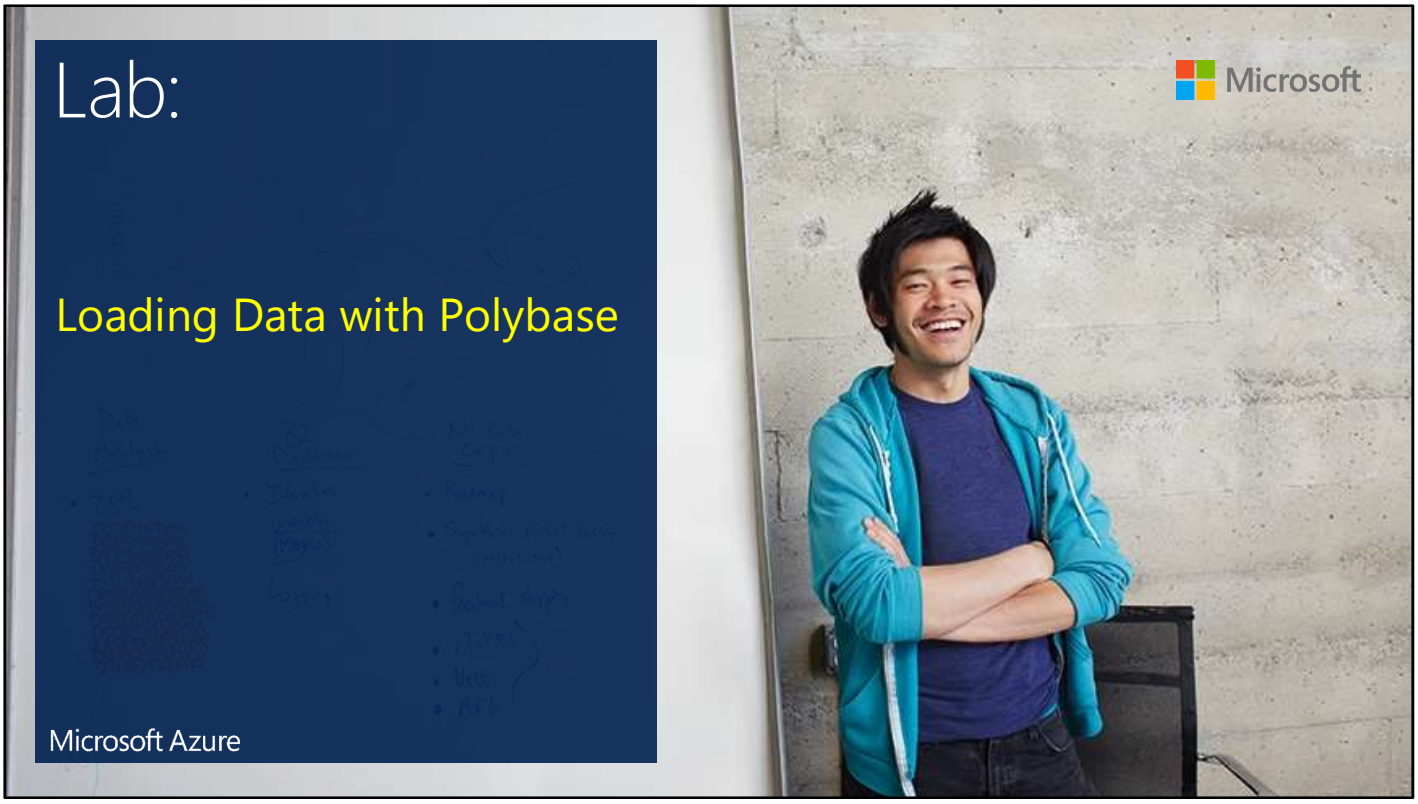
Specifies a custom format for all date and time data that might appear in a delimited text file. If the source file uses default datefime formats, this option is not necessary. Only one custom datetime format is allowed per file. You cannot specify multiple custom datetime formats per file. However, you can use multiple datetime formats if each one is the default format for its respective data type in the external table definition.

PolyBase only uses the custom date format for importing the data. It does not use the custom format for writing data to an external file format (EFF).

https://msdn.microsoft.com/en-us/library/dn935026.aspx#Arguments

Demo:
Loading data with PolyBase

This demo will show you how to create a table in Azure SQL Data Warehouse using data stored in an Azure BLOB store.



In this lab, you will create a table in Azure SQL Data Warehouse using data stored in an Azure BLOB store.

1. Ensure that you have an Azure Blob Storage Account with a container named "datacontainer"
2. Run the commands in 3. Azure_blob_AZCopy_Command.txt to load the dimdate2.txt file into the datacontainer of your storage account. Note that you will have to change some of the parameters to match your settings
3. Drop the Dates table on the Azure SQL Data Warehouse
4. Step through the code in 4. PolyBase_Load.sql to recreate the Date table in Azure SQL Data Warehouse using PolyBase

Importing and exporting data

This section of the course will cover:

- Importing with CTAS
- Creating a partitioned table with CTAS
- Exporting with CTAS
- Labelling your code

Importing with CTAS

```
CREATE TABLE [tmp].[FactOnlineSales]
WITH
(    DISTRIBUTION = HASH([ProductKey])
,    CLUSTERED COLUMNSTORE INDEX
)
AS
SELECT    *
FROM      [asb].[FactOnlineSales]
OPTION
(LABEL =  'CTAS : Import [cso].[FactOnlineSales]'
)
;
```

CTAS = Create Table As Select

You can import data into a table in Azure SQL Data Warehouse using a CTAS statement. This is a fully parallelised operation.

https://msdn.microsoft.com/en-us/library/mt204041.aspx

Creating a partitioned table with CTAS

```
CREATE TABLE [cso].FactOnlineSales_PTN
WITH
(   CLUSTERED COLUMNSTORE INDEX
,   DISTRIBUTION = HASH([ProductKey])
,   PARTITION
    (
        [DateKey] RANGE RIGHT FOR VALUES
        (
               '2007-01-01 00:00:00.000','2008-01-01 00:00:00.000'
         ,     '2009-01-01 00:00:00.000','2010-01-01 00:00:00.000'
        )
    )
)
AS
SELECT  *
FROM    [cso].[FactOnlineSales]
;
```

You can include partitioning (as well as indexing and distribution) options when using a CTAS statement

https://msdn.microsoft.com/en-us/library/mt204041.aspx

Exporting with CETAS

```
CREATE EXTERNAL TABLE [out].[dimProduct]
WITH(LOCATION = '/export/FactOnlineSales/'
,DATA_SOURCE = AzureStorage
,FILE_FORMAT = TextFileFormat
)
AS
SELECT *
FROM [cso].[dimProduct]
OPTION
(LABEL = 'CETAS : Export [cso].[FactOnlineSales]'
)
;
```

CETAS (CREATE EXTERNAL TABLE AS SELECT) can be used to export data from Azure SQL Data Warehouse to a file.

https://msdn.microsoft.com/en-us/library/mt204041.aspx

Labelling your code

Supported operations:

Select

Insert

Update

Delete

CTAS

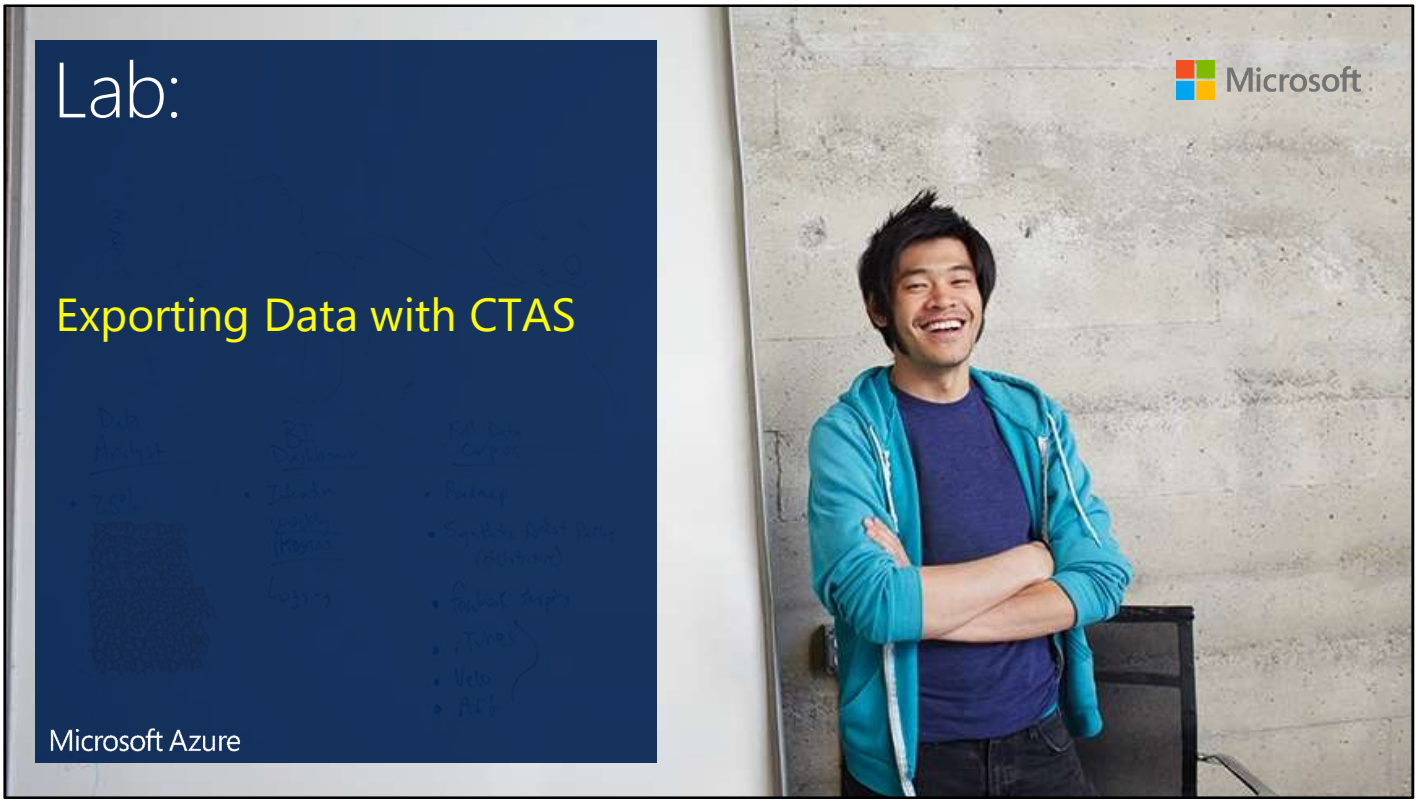
CETAS

```
SELECT  *
FROM    sys.dm_pdw_exec_requests
WHERE   [label] = 'SQLBits'
```

Tip: Use labels so you can view them in the sys.dm_pdw_exec_requests DMV in the Management Console

Demo:
Exporting data with CTAS

This demonstration will export data from Azure SQL Data Warehouse to Azure Blob store.



In this lab, you will export data from Azure SQL Data Warehouse to Azure Blob store.

1. Open up the script 5. ExportSQLDWData.sql
2. Run the code to create the schema
3. Run the code to export the data to the Azure Storage account.
4. Open up the portal and view the new files in the storage account

Monitoring data loads

This section of the course will cover:

- Monitoring execution requests.
- Monitoring execution request steps.
- Bringing execution requests and steps together.
- Monitoring worker activity.
- Monitoring data movement workers.

Monitoring execution requests

```
SELECT  'sys.dm_pdw_exec_requests'                  AS DMV
,       [label]                                     AS operation
,       NULL                                        AS location_type
,       NULL                                        AS step_index
,       DATEDIFF(ms ,MIN(req.[submit_time])
                    ,MAX(req.[end_time]))/1000.0    AS duration_sec
,       MIN(req.[submit_time])                      AS min_start_time
,       MAX(req.[end_time])                         AS max_end_Time
,       MIN(req.[total_elapsed_time])/1000.0        AS min_duration_sec
,       MAX(req.[total_elapsed_time])/1000.0        AS max_duration_sec
,       AVG(req.[total_elapsed_time])/1000.0        AS avg_duration_sec
,       NULL                                        AS row_count
,       [resource_class]                            AS resource_class
,       LEFT(command,50)                            AS command
FROM    sys.dm_pdw_exec_requests  AS req
WHERE   [request_id] = @req
GROUP BY [label]
,        [resource_class]
,        [command]
;
```

sys.dm_pdw_exec_requests provides information about execution requests

https://msdn.microsoft.com/en-us/library/mt203887.aspx

Monitoring execution request steps

```
SELECT      'sys.dm_pdw_request_steps'              AS DMV
,           step.[operation_type]                   AS operation_type
,           step.[location_type]                    AS location_type
,           step.[step_index]                       AS step_index
,           DATEDIFF(ms ,MIN([start_time])
                        ,max([end_time]))/1000.0    AS duration_sec
,           MIN([start_time])                       AS min_start_time
,           MAX([end_time])                         AS max_end_Time
,           MIN([total_elapsed_time])/1000.0        AS min_duration_sec
,           MAX([total_elapsed_time])/1000.0        AS max_duration_sec
,           AVG([total_elapsed_time])/1000.0        AS avg_duration_sec
,           SUM([row_count])                        AS row_count
,           NULL                                    AS resource_class
,           LEFT(step.[command],50)                 AS command
FROM        sys.dm_pdw_request_steps step
WHERE       [request_id] = @req
GROUP BY step.[operation_type]
,        step.[location_type]
,        step.[step_index]
,        step.[command]
;
```

sys.dm_pdw_request_steps provides information about the steps taken in an execution request

https://msdn.microsoft.com/en-us/library/mt203913.aspx

Bringing execution requests and steps together

```
SELECT      'sys.dm_pdw_sql_requests'                   AS DMV
,           step.[operation_type]                       AS operation_type
,           step.[location_type]                        AS location_type
,           step.[step_index]                           AS step_index
,           DATEDIFF(ms ,MIN(sreq.[start_time])
                        ,MAX(sreq.[end_time]))/1000.0   AS duration_sec
,           MIN(sreq.[start_time])                      AS min_start_time
,           MAX(sreq.[end_time])                        AS max_end_Time
,           MIN(sreq.[total_elapsed_time])/1000.0       AS min_duration_sec
,           MAX(sreq.[total_elapsed_time])/1000.0       AS max_duration_sec
,           AVG(sreq.[total_elapsed_time])/1000.0       AS avg_duration_sec
,           SUM(sreq.[row_count])                       AS row_count
,           NULL                                        AS resource_class
,           LEFT(step.[command],50)                     AS command
FROM        sys.dm_pdw_sql_requests  sreq
JOIN        sys.dm_pdw_request_steps step   ON  sreq.[step_index]      = step.[step_index]
                                            AND sreq.[request_id]      = step.[request_id]
WHERE       step.[request_id] = @req
GROUP BY    step.[operation_type]
,           step.[location_type]
,           step.[step_index]
,           step.[command]
;
```

You can bring these DMV's together

Monitoring worker activity

```
SELECT      'sys.dm_pdw_dms_external_work'          AS DMV
,           [type]                                  AS worker
,           DATEDIFF(ms ,MIN([start_time])
                        ,max([end_time]))/1000.0    AS duration_sec
,           MIN([start_time])                       AS min_start_time
,           MAX([end_time])                         AS max_end_Time
,           SUM([bytes_processed])/1000000000.0     AS sum_GB_processe
,           NULL                                    AS AVG_throughput_MB_sec
,           NULL                                    AS SUM_throughput_MB_sec
,           MIN([total_elapsed_time])/1000.0        AS min_duration_sec
,           MAX([total_elapsed_time])/1000.0        AS max_duration_sec
,           AVG([total_elapsed_time])/1000.0        AS avg_duration_sec
FROM        sys.dm_pdw_dms_external_work
WHERE       [request_id] = @req
GROUP BY    [type]
;
```

sys.dm_pdw_dms_external_work provides information about the worker loads that are operating against the Azure SQL Data Warehouse.

https://msdn.microsoft.com/en-us/library/mt204024.aspx

Monitoring data movement workers

```
SELECT      'sys.dm_pdw_dms_workers'                AS DMV
,           [type]                                  AS worker
,           DATEDIFF(ms ,MIN([start_time])
                        ,max([end_time]))/1000.0    AS duration_sec
,           MIN([start_time])                       AS min_start_time
,           MAX([end_time])                         AS max_end_Time
,           SUM([bytes_processed])/1000000000.0     AS sum_GB_processed
,           AVG([bytes_per_sec])/1000000.0          AS AVG_throughput_MB_sec
,           SUM([bytes_per_sec])/1000000.0          AS SUM_throughput_MB_sec
,           MIN([total_elapsed_time])/1000.0        AS min_duration_sec
,           MAX([total_elapsed_time])/1000.0        AS max_duration_sec
,           AVG([total_elapsed_time])/1000.0        AS avg_duration_sec
FROM        sys.dm_pdw_dms_workers
WHERE       [request_id] = @req
GROUP BY    [type]
;
```

sys.dm_pdw_dms_workers provides monitoring for the data movement workers

https://msdn.microsoft.com/en-us/library/mt203878.aspx

Azure Data Factory Integration

This section of the course will cover:

- Azure Data Factory Components
- PolyBase Prerequisites: Linked Service
- Azure Storage Linked Service
- SQL DW Linked Service
- PolyBase Prerequisites: Input Datasets
- Input Dataset
- PolyBase Prerequisites: Copy Activity
- Copy Activity
- Copy Activity Wizard
- ADF Limitations

ADF components

Linked services
Datasets
Activities

} Pipeline

Data Factory is a cloud-based data integration service that orchestrates and automates the **movement** and **transformation** of data.

https://docs.microsoft.com/en-gb/azure/data-factory/data-factory-introduction

PolyBase Pre-requisites: Linked Service

Azure Storage source only
No SAS authentication

PolyBase as a Linked Service can only access Azure Storage only, with access using the API key only. Shared Access Signatures are not supported as an access method.

Azure Storage Linked Service

```
{
    "name": "<ASBLinkedServiceName>"
,   "properties":
    {   "hubName": "DWfactory_hub"
    ,   "type": "AzureStorage"
    ,   "typeProperties":
        {
            "connectionString":
"DefaultEndpointsProtocol=https;AccountName=jrjtrip2015;Accoun
tKey=**********"
        }
    }
}
```

This is the example code for creating a linked service for an Azure Blob storage. Linked services define the information needed for Data Factory to connect to external resources.

https://docs.microsoft.com/en-gb/azure/data-factory/data-factory-introduction#linked-services

SQLDW Linked Service

```
{
    "name": "<SQLDWLinkedServiceName>"
,   "properties":
    {   "description": ""
    ,   "hubName": "DWfactory_hub"
    ,   "type": "AzureSqlDW"
    ,   "typeProperties":
        {
            "connectionString": "Data
Source=<server>.database.windows.net;Initial Catalog=<db>;Integrated
Security=False;User ID=<user>;Password=**********;Connect
Timeout=30;Encrypt=True"
        }
    }
}
```

This is an example of a SQL DW Linked Service

https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-load-with-data-factory#configure-the-destination-your-sql-data-warehouse

PolyBase Pre-requisites: Input Dataset

Azure Blob Properties:
Input Dataset: Azure Blob
Type: TextFormat
rowDelimiter: \n
nullValue: “”
encodingName: utf-8 (default)
escapeChar: not specified in activity
quoteChar: not specified in activity

https://docs.microsoft.com/en-us/azure/data-factory/data-factory-azure-sql-data-warehouse-connector#dataset-type-properties

Input Dataset

```
 "typeProperties":
{   "folderPath": "<blob_path>"
,   "format":
    {
        "type": "TextFormat"
    ,   "columnDelimiter": "<any delimiter>"
    ,   "rowDelimiter": "\n"
    ,   "nullValue": ""
    ,   "encodingName": "utf-8"
    }
,   "compression":
    {
        "type": "GZip"
    ,   "level": "Optimal"
    }
}
```

Azure Data Factory datasets can be both input and output datasets. The example relates to sourcing input data from a file in Azure Blob store

https://docs.microsoft.com/en-gb/azure/data-factory/data-factory-create-datasets

PolyBase Pre-Requisites: Copy Activity

Blob Source Properties:
skipHeaderLineCount: not specified
SqlDWSink:
sliceIdentifierColumnName: not specified
Copy Activity:
columnMapping: not specified

The Copy Activity can be created with a JSON file, or a Copy Activity Wizard.

https://docs.microsoft.com/en-us/azure/data-factory/data-factory-data-movement-activities

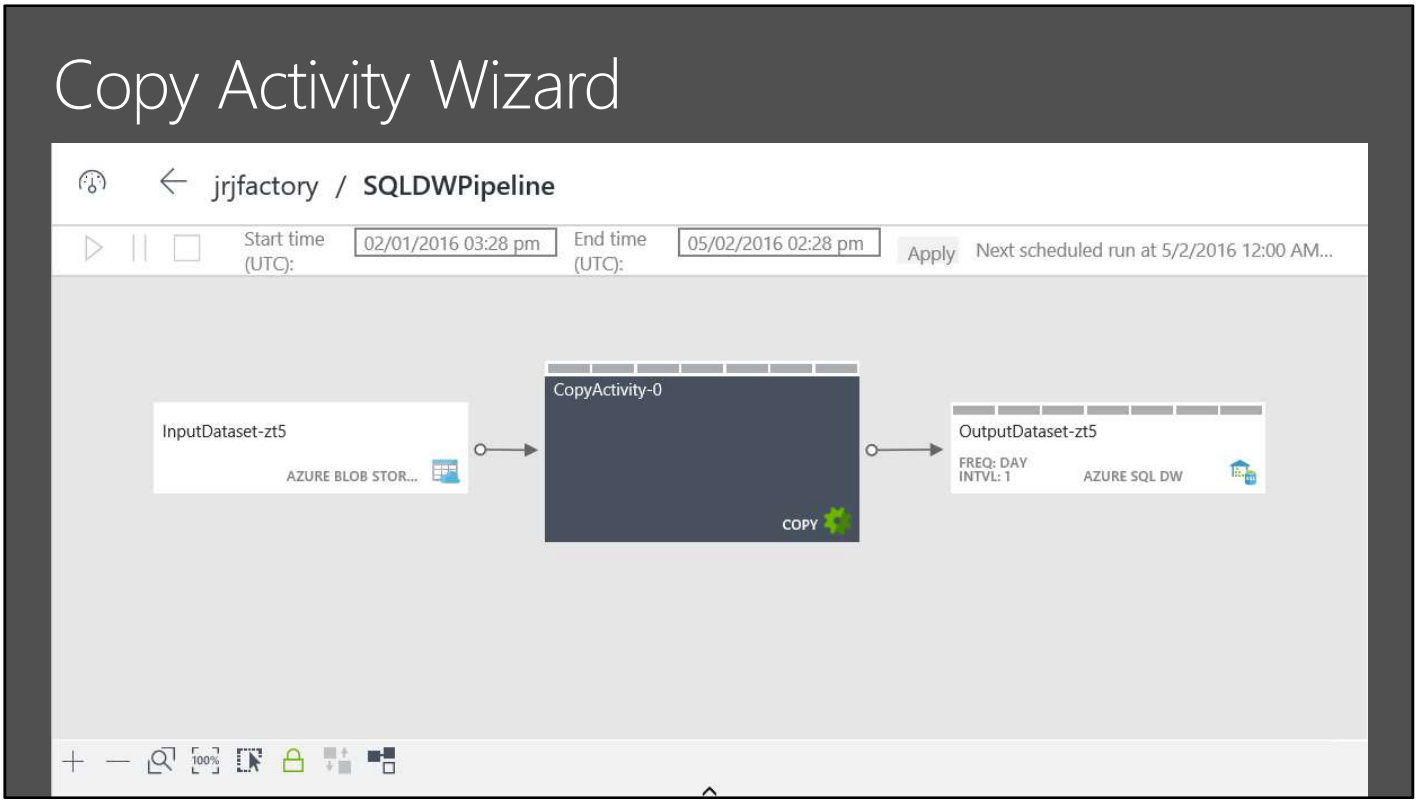
Copy Activity

```
 "sink":
{ "type": "SqlDWSink"
, "writeBatchSize": 1000000
, "writeBatchTimeout": "00:05:00"
, "allowPolyBase": true
, "polyBaseSettings":
    { "rejectType": "percentage"
    , "rejectValue": 10
    , "rejectSampleValue": 100
    , "useTypeDefault": true
    }
}
```

Copy Activity using a JSON file example

https://docs.microsoft.com/en-us/azure/data-factory/data-factory-azure-sql-data-warehouse-connector

Copy Activity Wizard



Copy Activity using the wizard

https://docs.microsoft.com/en-us/azure/data-factory/data-factory-azure-sql-data-warehouse-connector

ADF Limitations

Primary limitations

One time sync can't be edited

PolyBase can't be configured in Copy Wizard (today)

File headers must be addressed

ADF validates the data types of the data in the source

Fields must all map to string if headers are present

Use another copy activity (blob to blob) to remove the header from the source

Avoiding column mappings

Input names must equal output names

Data types must match

This page is left intentionally blank

Demo: Using ADF

This demonstration will export data from Azure SQL Data Warehouse to Azure Blob store using ADF.

1. Use the Azure Data Factory (ADF) that you create earlier. If you have not got an ADF instance, create one now.
2. In ADF, click on Copy Activity to copy the contents of the EquityTimeSeriesData Table to text files in Azure Blob

Azure Machine Learning Integration

This section of the course will cover:

- Azure Machine Learning Integration

Integrating Machine Learning to SQL DW

Use the Reader Module to read from Azure SQL DW

Use the Writer Module to write to Azure SQL DW

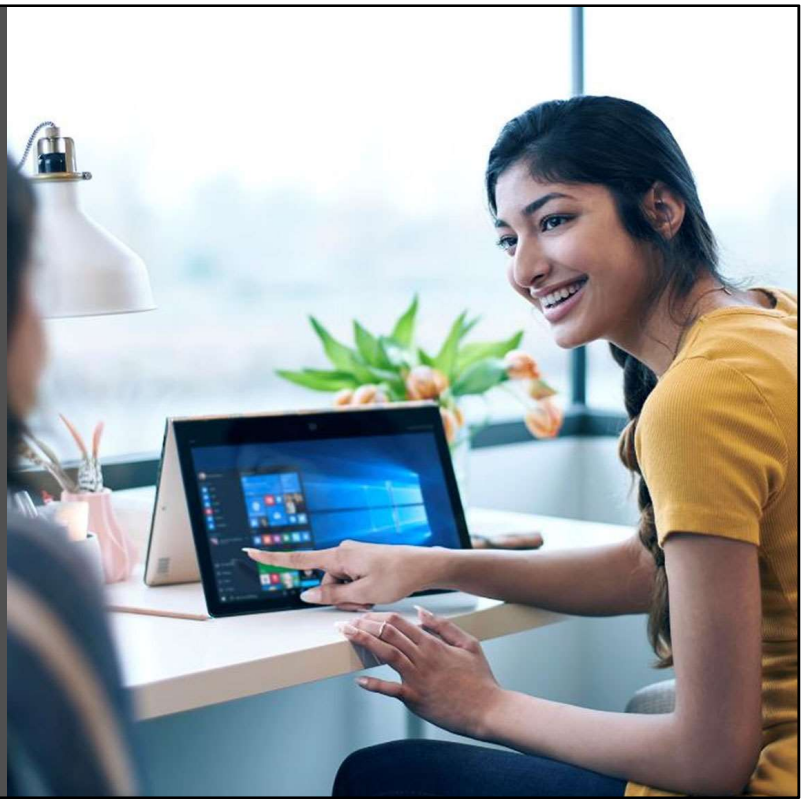
- You must define the table definition to write to

https://blogs.technet.microsoft.com/machinelearning/2016/03/08/how-to-use-azure-ml-with-azure-sql-data-warehouse/

https://blogs.technet.microsoft.com/machinelearning/2016/03/08/how-to-use-azure-ml-with-azure-sql-data-warehouse/

Demo:

Using Azure Machine Learning with Azure SQL Data Warehouse



In this lab, you will perform the following steps

1. Integrate Azure SQL Data Warehouse with Azure Machine Learning

Create an Azure SQL Data Warehouse

Open up the following link and perform all of the steps from one of the web page below to integrate an Azure SQL Data Warehouse with Azure Machine Learning.

Create an S1 Standard Machine Learning Workspace

1. Click on the following url:

2. https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-create-workspace

3. Perform the steps under **To create a workspace** only

Perform a Machine Learning Experiment

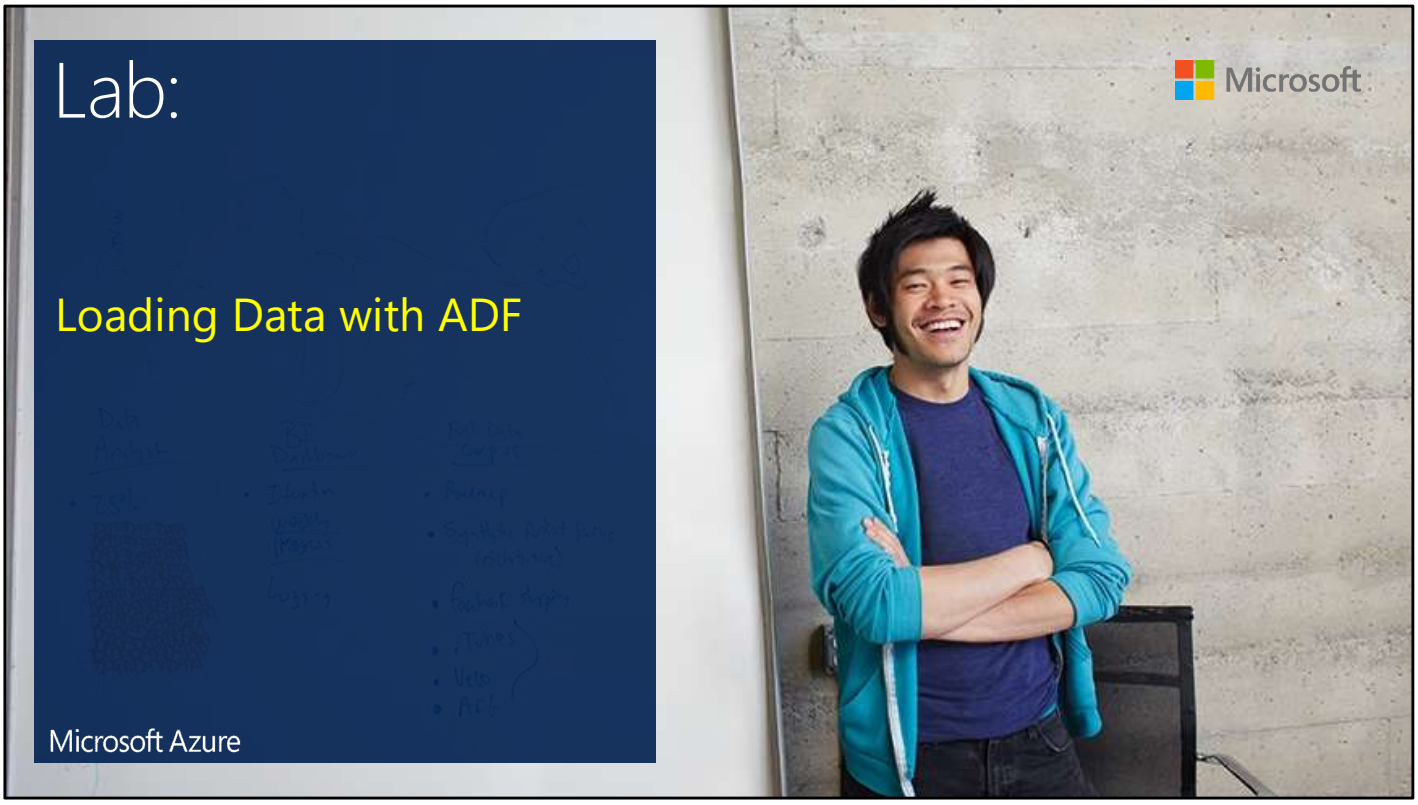
Use the Product table from your Azure SQL Data Warehouse Instance you have created to perform the Machine Learning experiment using the workspace created in the previous step

1. Go to the following url and sign in https://studio.azureml.net/

2. In the top toolbar, to the right of the **?** icon, ensure that the workspace you created in the previous step is selected

3. Perform the steps in the following url:
https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-integrate-azure-machine-learning

NOTE: The key objective here is to show how to import data from SQL Data Warehouse, not the model itself



In this lab, you will export data from Azure SQL Data Warehouse to Azure Blob store using ADF.

1. Use the Azure Data Factory (ADF) that you create earlier. If you have not got an ADF instance, create one now.
2. In ADF, click on Copy Activity
   - i. In the Properties pane
     - a. In the task name type "ADWtoBlob"
     - b. In the Task cadence (or) Task schedule, click the radio button next to Run Once now.
     - c. Leave the expiration time as "3.00:00:00"
     - d. Click Next.
   - ii. In the Source pane
     - a. Click Azure SQL Data Warehouse
     - b. In connection name, type ADWSource
     - c. In Account Selection method, select"Enter Manually"
     - d. In fully qualified domain name, type in the url for your Azure SQL Data Warehouse instance
     - e. In Database name, type "EquityDB"
     - f. In User name, type in your admin account.
     - g. In Password, type in the password, and click on Next.
   - iii. In the "Select tables from which to copy the data (or) use a custom query" pane
     - a. Click on the [dbo].[EquityTimeSeriesData], then click on Next.
   - iv. In the "Destination Data Store" pane.
     - a. Click Azure Blob Storage, and click next
     - b. In connection name, type BlobDest
     - c. In Account Selection method, leave as "From Azure Subscription"
     - d. In Azure Subscription, ensure your subscription name is selected
     - e. In Storage account name, select your Azure Blob Storage account name, click Next.
   - v. In the choose input file or folder pane
     - a. Browse to the folder that will store the data (i.e. datacontainer) ,click choose and click Next.
   - vi. In the "File Format settings" pane

a. Leave the default settings, and click Next

i. In the "Performance settings" pane

a. Expand Advanced Settings, and confirm the Parallel copy is set to "auto", click on Next., and then Finish

b. Click on "Click here to monitor the pipeline" hyperlink

c. After 1 minute, refresh the internet page, until the activity windows state "Ready"

1. Go to your storage account in the Azure Portal to confirm the presence of the [dbo].[EquityTimeSeriesData] file in the container of the storage account.

Recommendations

This section of the course will cover:

- Data Preparation
- Initial load
- Incremental load

Data preparation

Transfer data to blob storage

One root folder per table

Sub-folders for partitions / subset analysis

Split table data into multiple files: 1 file for each reader

Compress data to optimise transfer

This page is left intentionally blank

Initial load

CTAS data with PolyBase for max throughput

One external table definition per table

Configure load user

Size the rowgroup for memory grant
Set appropriate resource class

Maximise # readers to accelerate load

DWU1000+ for 60 readers
Multiply #files by readers for balanced throughput (i.e. 60,120,180 etc.)

This page is left intentionally blank

Lab review

1. What is the purpose of the load user in Azure SQL Data Warehouse?
2. What is the fastest method for loading data in Azure SQL Data Warehouse?
3. What is an external format file, and its' purpose
4. What does CTAS and CETAS stand for? What is the difference?
5. What is the wizard that can be used in Azure Data Factory to export data?

Take a moment to think about the following questions, and they will be reviewed as a class.

1. What is the purpose of the load user in Azure SQL Data Warehouse?
2. What is the fastest method for loading data in Azure SQL Data Warehouse?
3. What is an external format file, and its' purpose
4. What does CTAS and CETAS stand for? What is the difference?
5. What is the wizard that can be used in Azure Data Factory to export data?

Summary

Summary

The role of the load user.
The different methods for loading data.
How to use PolyBase.
Importing and Exporting Data.
Monitoring Data Loads.
Using ADF to load SQL DW.
Loading recommendations.

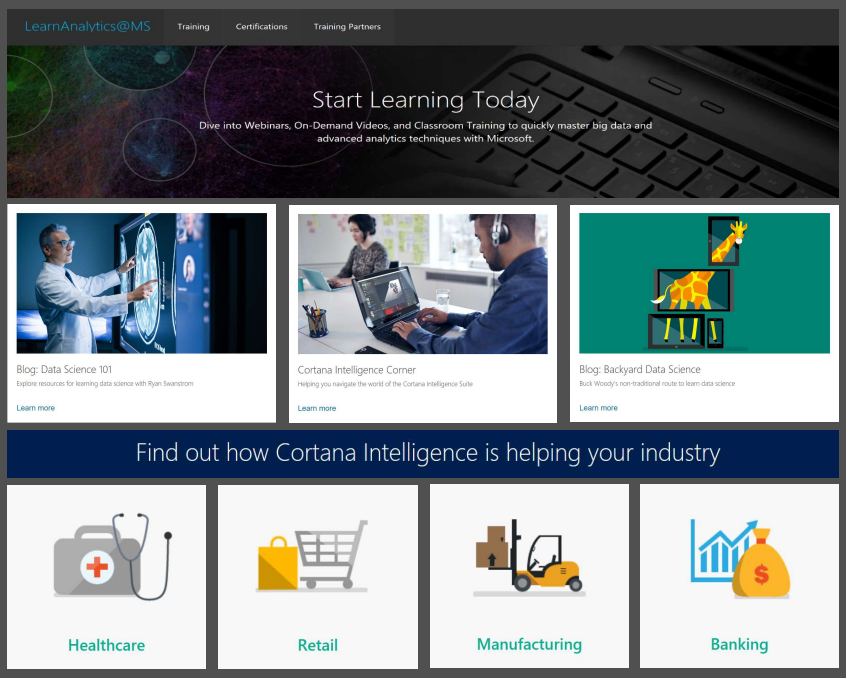
In this session, you have learned:

- The role of the load user.
- The different methods for loading data.
- How to use PolyBase.
- Importing and Exporting Data.
- Monitoring Data Loads.
- Using ADF to load SQL DW.
- Loading recommendations.

Microsoft

There are more learning options as shown in the links on the right, including:

- Online training
- Videos
- Instructor Led training
- Blogs
- Cortana Intelligence Gallery



Click on the graphics to explore more learning options from your Advanced Analytics and Data Science team, including:

- Online training
- Videos
- Instructor Led training
- Blogs
- Cortana Intelligence Gallery

Course Documentation

Accessing the course materials

1. Click on the picture on the left.
2. Sign in with your Live ID.
3. Look for the SQLW301 item.
4. Click on the course materials link.

Accessing the course materials

1. Click on the picture on the left or go to https://cisw-foundations.azurewebsites.net/
2. Sign in with your Live ID.
3. Look for the SQLW301 item.
4. Click on the course materials link

Microsoft

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see **Microsoft Copyright Permissions** at **http://www.microsoft.com/permission**

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.
The Microsoft company name and Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

This document reflects current views and assumptions as of the date of development and is subject to change. Actual and future results and trends may differ materially from any forward-looking statements. Microsoft assumes no responsibility for errors or omissions in the materials.

THIS DOCUMENT IS FOR INFORMATIONAL AND TRAINING PURPOSES ONLY AND IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT.