We will cover these topics

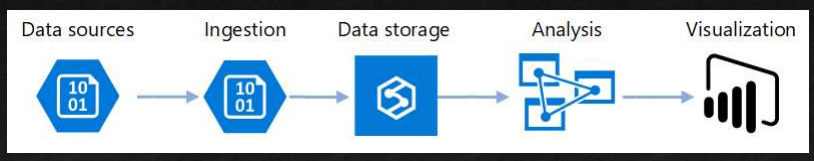
- Introduction
- What is ADF?
- How ADF Works?

Lesson 4

Azure Data Factory

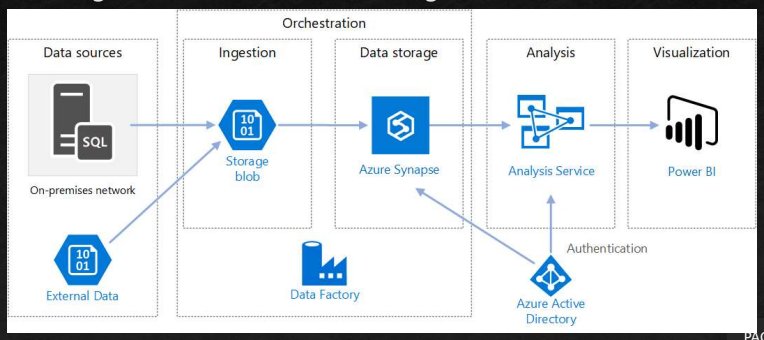
Introduction

- Microsoft Azure Data Factory is a managed cloud service that you can use to create actionable business insights from your unorganized data.
- It can help you manage complex hybrid extraction, transformation, and loading (ETL), extract-load-transform, and data-integration projects.



What is Azure Data Factory?

- Azure Data Factory is a cloud-based ETL and data integration service that enables you to create data-driven workflows (*also known as Pipelines*) to:
  - Orchestrate data movement.
  - Transform data at scale.
- By using Azure Data Factory, you can reorganize raw data into meaningful data stores and data lakes.

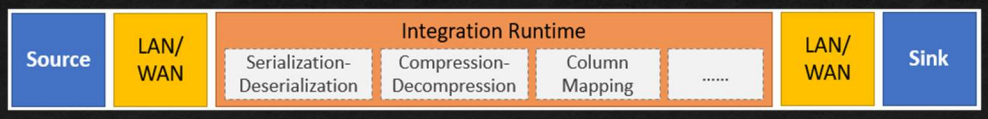


How ADF Works?

- Azure Data Factory is a collection of interconnected systems that combine to provide an end-to-end data analytics platform.
- Azure Data Factory consists of several functions such as:
  - Connect and collect
  - Transform and enrich
  - Continuous integration and delivery (CI/CD) and publish
  - Monitoring
- Key components of Azure Data Factory are:
  - Pipelines
  - Activities
  - Datasets
  - Linked services
  - Data flows
  - Integration runtimes

ADF Functions – Connect and Collect

- Collect the required data from the appropriate data sources, located in different locations (on-premises and cloud)
  - Structured
  - Unstructured
  - Semi-structured
- With ADF, you can use the copy activity to move data from various sources to a single, centralized data store in the cloud.
- After you've copied the data, you use other systems to transform and analyze it.
- Summary of **high-level steps:**



ADF Function – Transform & Enrich

- After you've successfully copied the data to a central cloud-based location, you can process and transform the data as needed.
- You'll use Azure Data Factory mapping data flows to achieve this.
- *Data flows* enable you to create data transformation graphs that run on Spark. However, you don't need to understand Spark clusters or Spark programming.

ADF Function – CI/CD and Publish

- Continuous integration (CI) means automatically testing each change made to your codebase as soon as possible.
- Continuous delivery (CD) follows this testing and pushes changes to a staging or production system.
- Support for CI/CD enables you to develop and deliver your ETL processes incrementally before you publish. Azure Data Factory provides for CI/CD of your data pipelines by using:
  - Azure DevOps
  - GitHub
- After Azure Data Factory has refined the raw data, you can load the data into whichever analytics engine your business users can access from their business intelligence tools, including:
  - Azure Synapse Analytics
  - Azure SQL Database
  - Azure Cosmos DB

ADF Function – Monitor

- After you've successfully built and deployed your data integration pipeline, it's important that you can monitor your scheduled activities and pipelines.
- This enables you to track success and failure rates.
- Azure Data Factory provides support for pipeline monitoring by using one of the following:
  - Azure Monitor
  - API
  - PowerShell
  - Azure Monitor logs
  - Health panels in the Azure portal

ADF Components – 1 of 2

- **Pipeline:**
  - A logical grouping of activities that perform a specific unit of work.
  - These activities together perform a task.
  - The advantage of using a pipeline is that you can more easily manage the activities as a set instead of as individual items.
- **Activities:**
  - A single processing step in a pipeline.
  - Azure Data Factory supports three types of activity: data movement, data transformation, and control activities.
- **Datasets:**
  - Represent data structures within your data stores.
  - These point to (or reference) the data that you want to use in your activities as either inputs or outputs.

PAGE 42

42

ADF Components – 2 of 2

- **Linked Services:**
  - Define the required connection information needed for Azure Data Factory to connect to external resources, such as a data source.
  - ADF uses these for two purposes: to represent a **data store** or a **compute resource**.
- **Data Flows:**
  - Enable your data engineers to develop data transformation logic without needing to write code.
  - Data flows are run as activities within Azure Data Factory pipelines that use scaled-out Apache Spark clusters.
- **Integration Runtimes:**
  - Azure Data Factory uses the compute infrastructure to provide the following data integration capabilities across different network environments:
    - data flow, data movement, activity dispatch, and SSIS package execution.
  - In Azure Data Factory, an integration runtime provides the bridge between the activity and linked services.

PAGE 43

43

Q&A

Clarification Session

PAGE 44

44

We will cover these topics

- Getting Started with Data Flow
- Data Flow Transformation
- How to use Data Flow

Lesson 5

Mapping Data Flow

PAGE 45

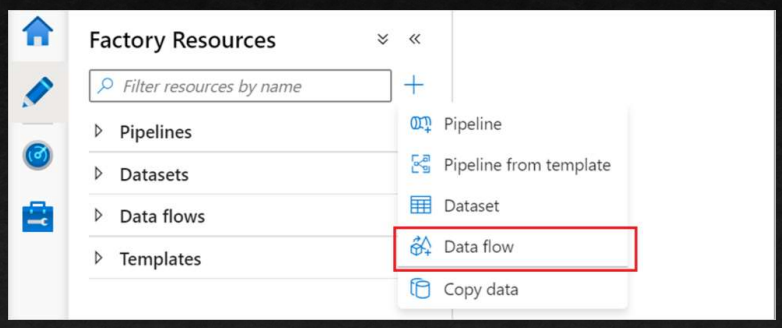
45

Introduction to Data Flow

- Mapping data flows are visually designed data transformations in Azure Data Factory.
- Data flows allow data engineers to develop data transformation logic without writing code.
- The resulting data flows are executed as activities within Azure Data Factory pipelines that use scaled-out Apache Spark clusters.
- Data flow activities can be operationalized using existing Azure Data Factory scheduling, control flow, and monitoring capabilities.
- Azure Data Factory handles all the code translation, path optimization, and execution of your data flow jobs.

Getting Started – Create a Data Flow

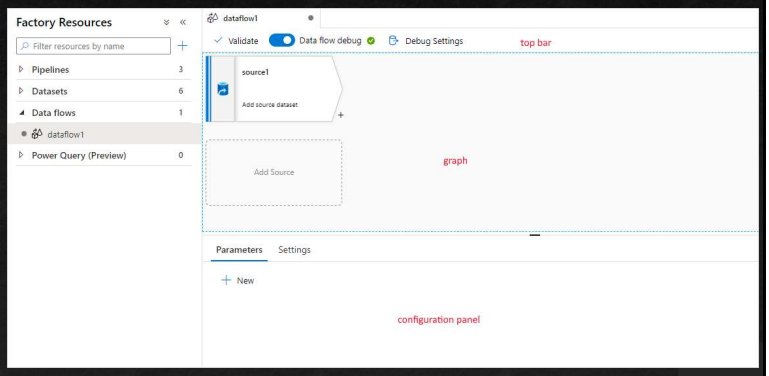
- Data flows are created from the factory resources pane like pipelines and datasets.



Getting Started – Authoring a Data Flow

- Mapping data flow has a unique authoring canvas designed to make building transformation logic easy.
- The data flow canvas is separated into three parts: the top bar, the graph, and the configuration panel.

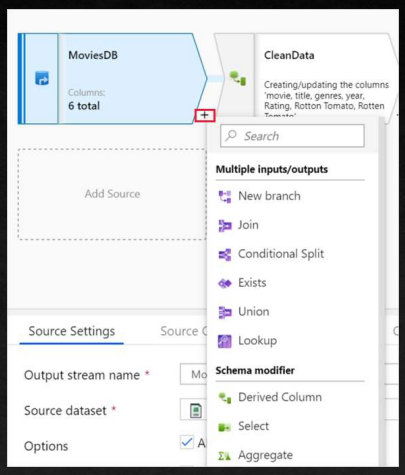
The top bar contains actions that affect the whole data flow, like saving and validation. You can view the underlying JSON code and data flow script of your transformation logic as well.



48

Getting Started – Graph

- The graph displays the transformation stream.
- It shows the lineage of source data as it flows into one or more sinks.
- To add a new source, select **Add source**.
- To add a new transformation, select the plus sign on the lower right of an existing transformation.

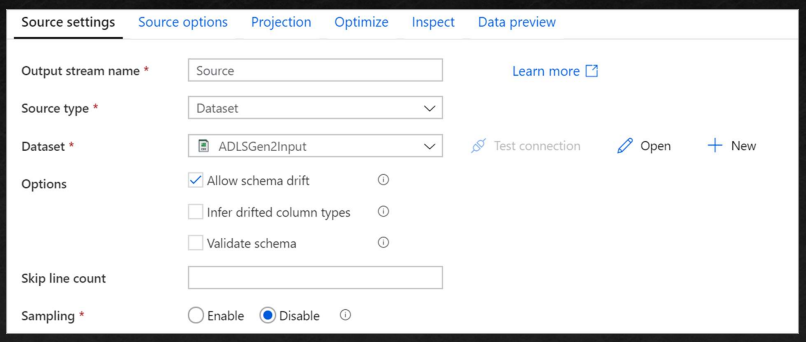


PAGE 49

49

Getting Started – Configuration Panel (1 of 4)

- The configuration panel shows the settings specific to the currently selected transformation.
- Each transformation contains at least four configuration tabs.
- The first tab in each transformation's configuration pane contains the settings specific to that transformation.

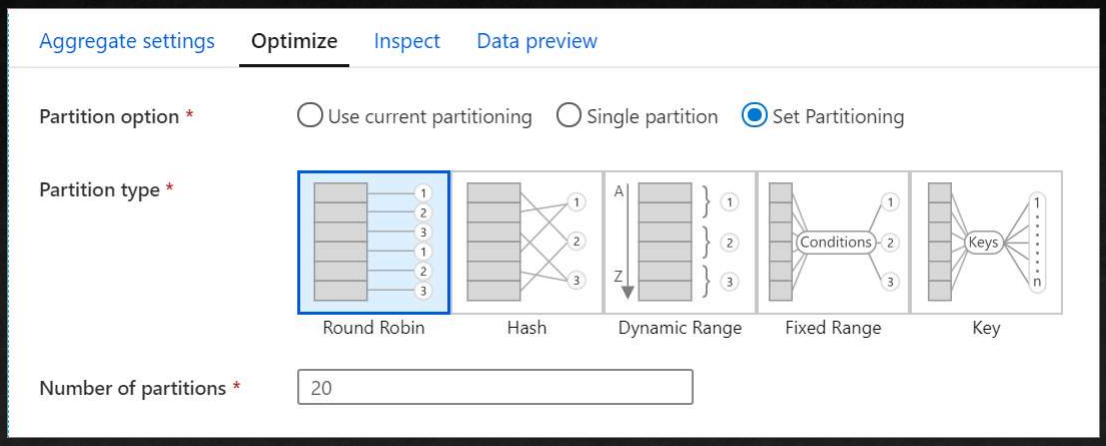


PAGE 50

50

Getting Started – Configuration Panel (2 of 4)

- **Optimize:** The **Optimize** tab contains settings to configure partitioning schemes.

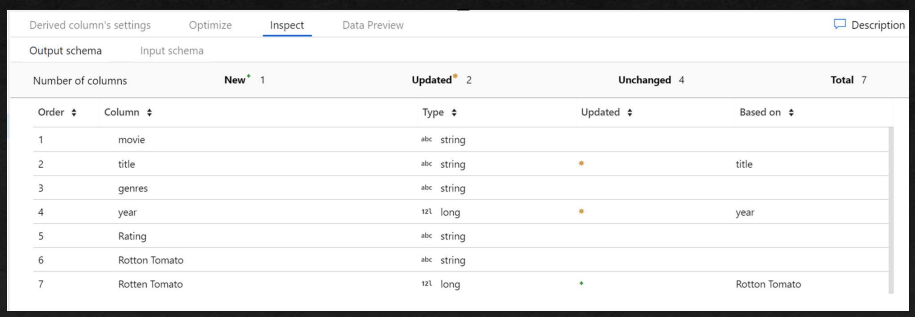


PAGE 51

51

Getting Started – Configuration Panel (3 of 4)

- **Inspect:** The **Inspect** tab provides a view into the metadata of the data stream that you're transforming. You can see column counts, the columns changed, the columns added, data types, the column order, and column references.
- **Inspect** is a read-only view of your metadata. You don't need to have debug mode enabled to see metadata in the **Inspect** pane.

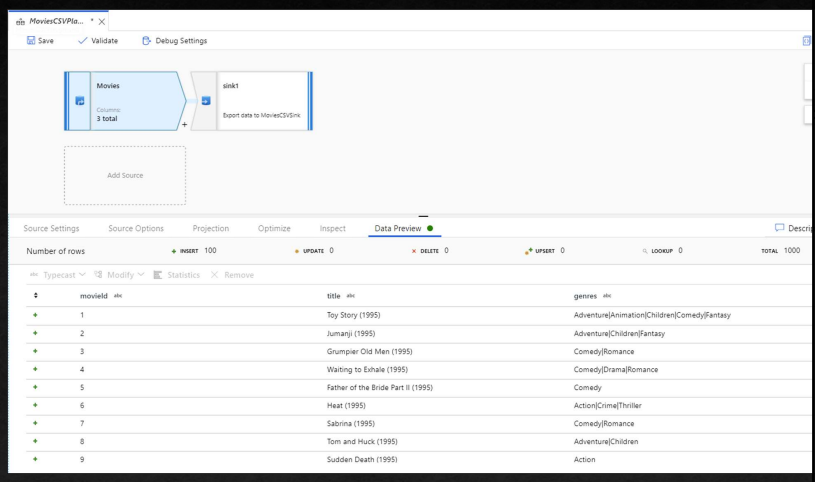


PAGE 52

52

Getting Started – Configuration Panel (4 of 4)

Data preview: With debug on, the Data Preview tab will light-up on the bottom panel. The data preview will only query the number of rows that you have set as your limit in your debug settings.
Click **Refresh** to fetch the data preview.



PAGE 53

53

Data Flow Transformations

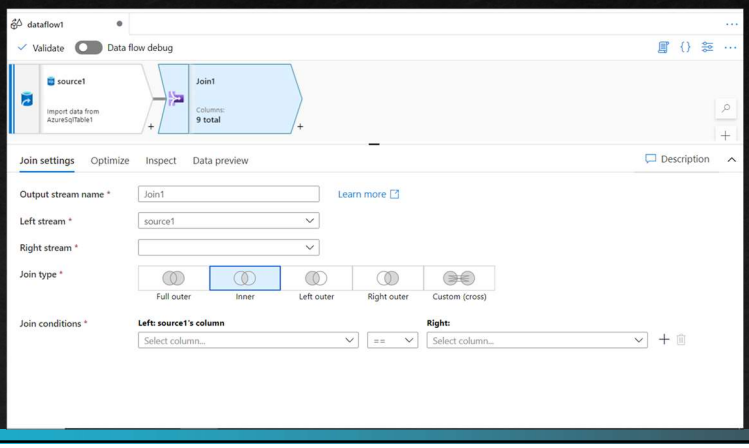
- To access the list of transforms in Data Factory, one needs to have an instance of it created using which we can create data pipelines.
- We can create Data Flows which can be used as a part of the data pipeline.
- In the Data Flow graph, once you have added one or multiple data sources, then the next logical step is to add one or more transformations to it.
- Data flow provides several transformations that we can apply to the data.
- New transformations are being added with each release.

PAGE 54

54

Data Flow Transformations – JOIN

- Typically, when you have data from one or more data sources, there is a need to bind this data into a common stream and for such use-cases, this transform can be used.

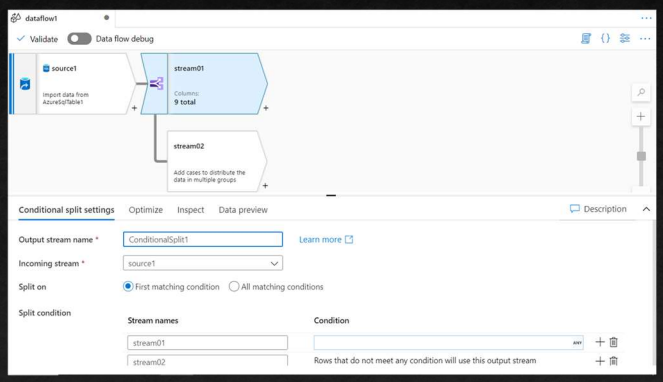


PAGE 55

55

Data Flow Transformations – SPLIT

- In Azure Data Factory, the split transform can be used to divide the data into two streams based on a criterion.
- The data can be split based on the first matching criteria or all the matching criteria as desired.
- This facilitates discrete types of data processing on data divided categorically into different streams using this transform.

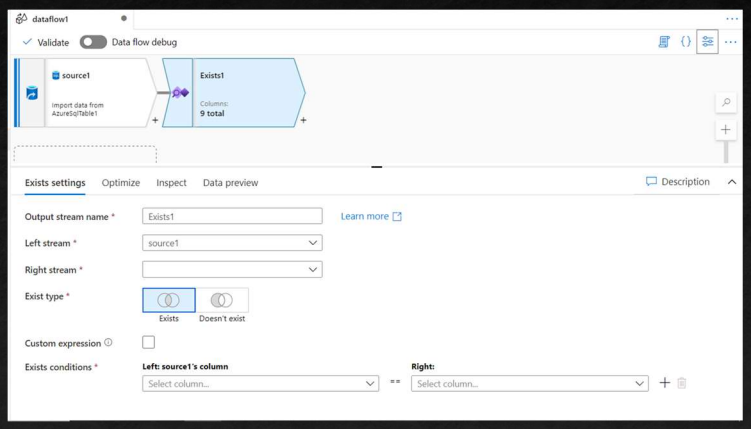


PAGE 56

56

Data Flow Transformations – EXISTS

- The Exists transform in Azure Data Factory is an equivalent of SQL EXISTS clause.
- It can be used to compare data from one stream with data in another stream using one or multiple conditions.

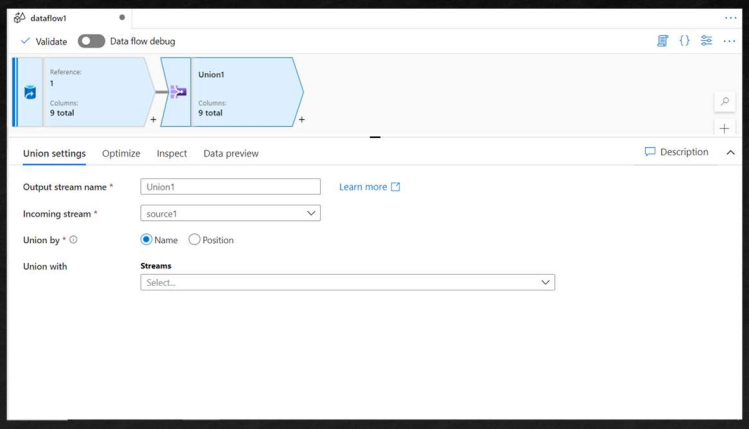


PAGE 57

57

Data Flow Transformations – UNION

- It can be used to merge data from two data streams that have identical or compatible schema into a single data stream.
- The schema from two streams can be mapped by name or ordinal position of the columns.

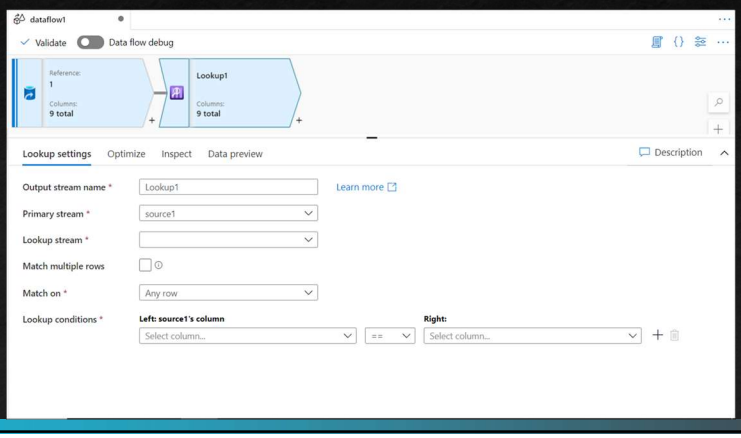


PAGE 58

58

Data Flow Transformations – LOOKUP

- While loading data into dimension or facts, one needs to validate if the data already exists to take a corresponding action of updating or inserting data.
- The lookup transform takes data from an incoming stream and matches it with data from the lookup stream and appends columns from the lookup stream into the primary stream.

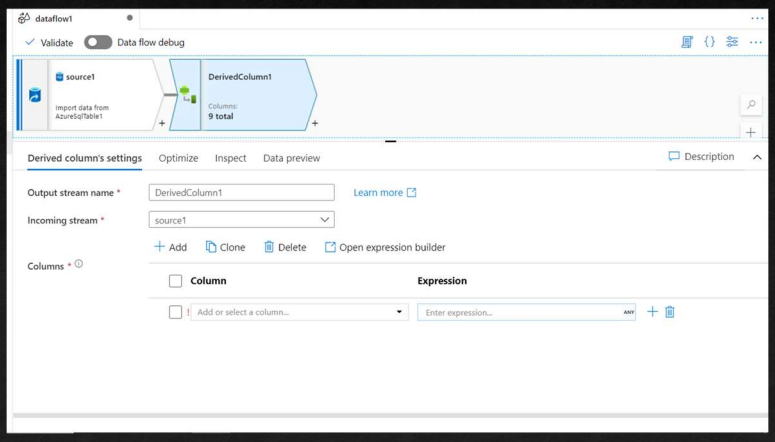


PAGE 59

59

Data Flow Transformations – DERIVED COLUMN

- With this transformation, we are starting with the schema modifier category of transforms.
- Often there is a need to create new calculated fields or update data in the existing fields in a data stream. Derived column transformation can be used in such cases.

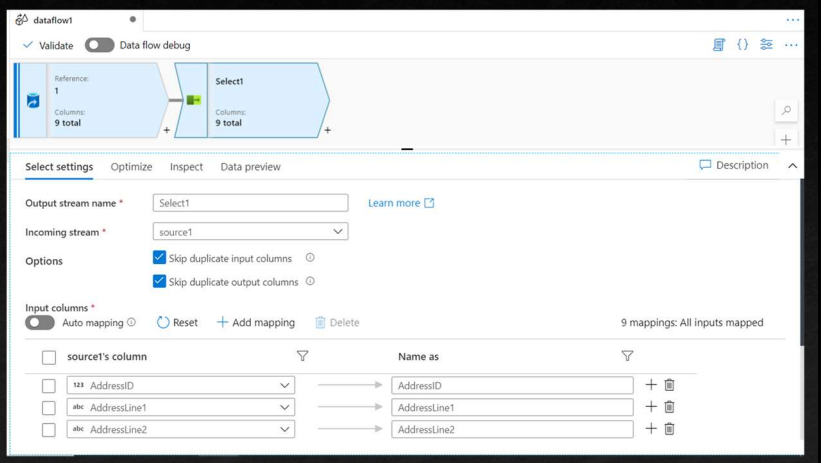


PAGE 60

60

Data Flow Transformations – SELECT

- While data in the input stream is being processed, there may be cases that while joining, merging, splitting, and creating calculated fields, it may result in some unnecessary or duplicate fields.
- To remove such fields or to rename the fields & change the mappings or to remove the undesired fields The SELECT transform can be facilitated.

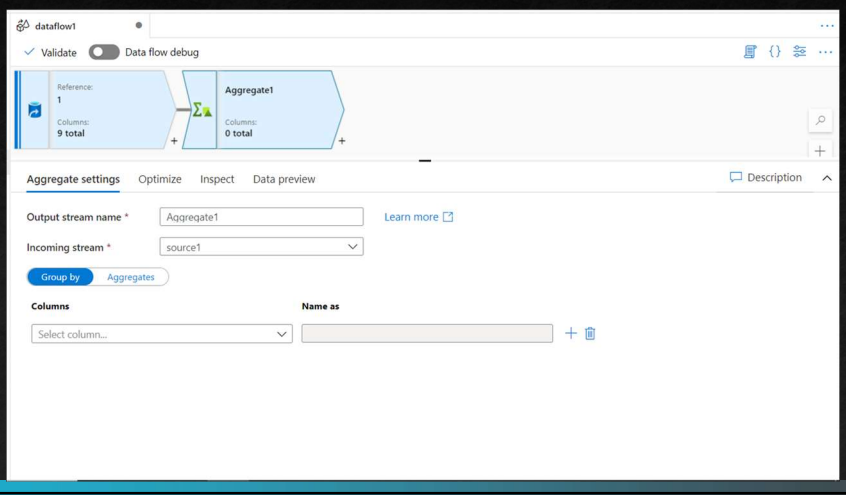


PAGE 61

61

Data Flow Transformations – AGGREGATE

- One of the obvious mechanisms of aggregating or rolling up data in SQL is by using the GROUP BY clause. This functionality can be exercised by using the Aggregate transform in Azure Data Factory.

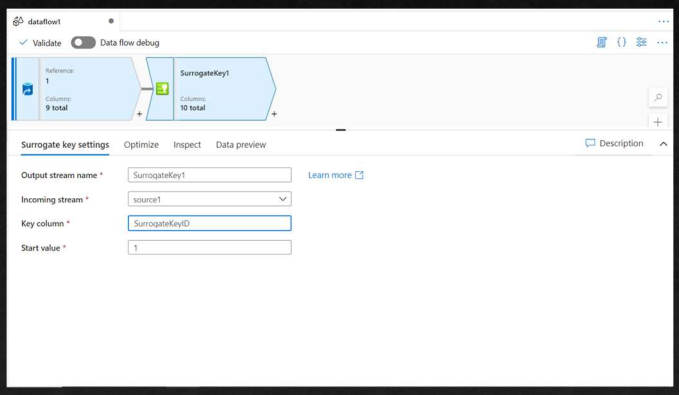


PAGE 62

62

Data Flow Transformations – SURROGATE KEY

- In a data warehousing scenario, typically in slowly changing dimensions (SCD) where one cannot use the business key as the primary key, surrogate keys are created that act as a unique identified for the record.
- Azure Data Factory provides a transform to generate these surrogate keys as well using the Surrogate Key transform.

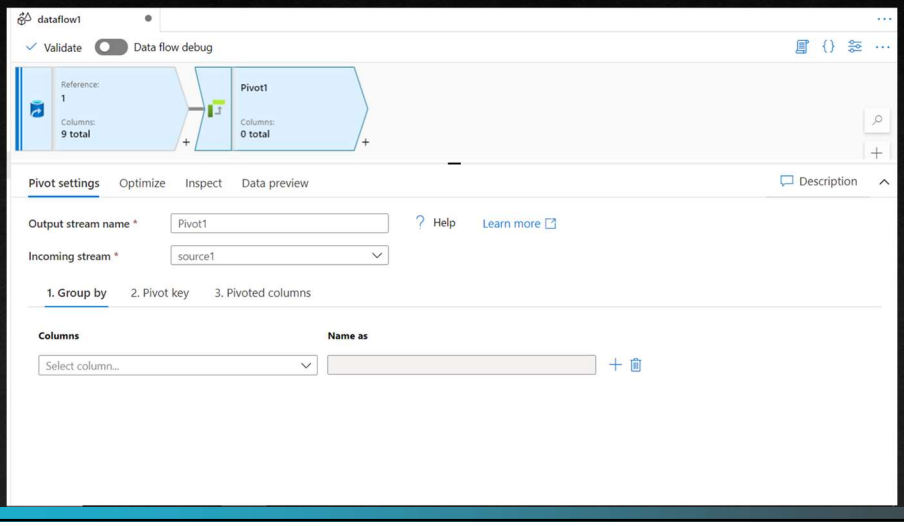


PAGE 63

63

Data Flow Transformations – PIVOT

- Converting unique values of rows from a field as columns is known as pivoting of data.

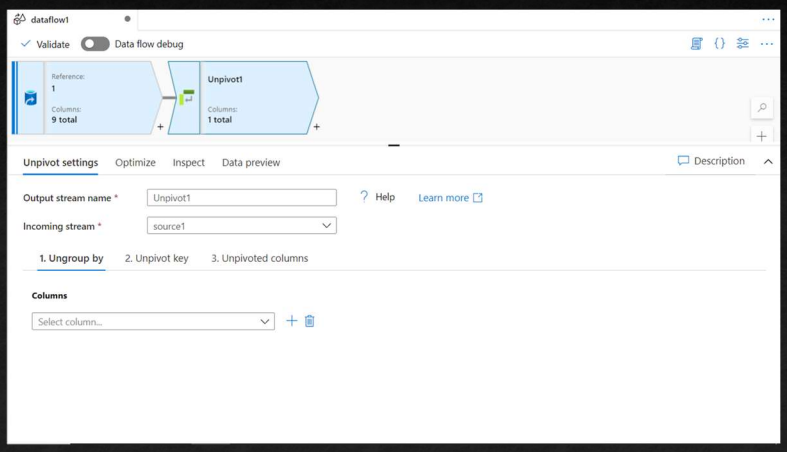


PAGE 64

64

Data Flow Transformations – UNPIVOT

- Unpivot transform is the inverse of pivot transform where it converts columns into rows.
- In pivot, the data is grouped as per the criterion, and in this case, the data is ungrouped and unpivoted.

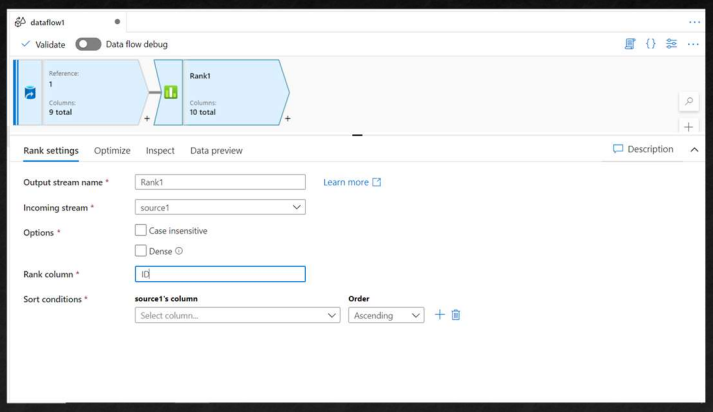


PAGE 65

65

Data Flow Transformations – RANK

- Once the data is channeled into different streams, validated with different source and destination repositories, calculated, and aggregated using custom expressions, towards the end of the data pipeline, one would typically sort the data.
- With this, there may be a need to rank the data as per a specific sorting criterion.

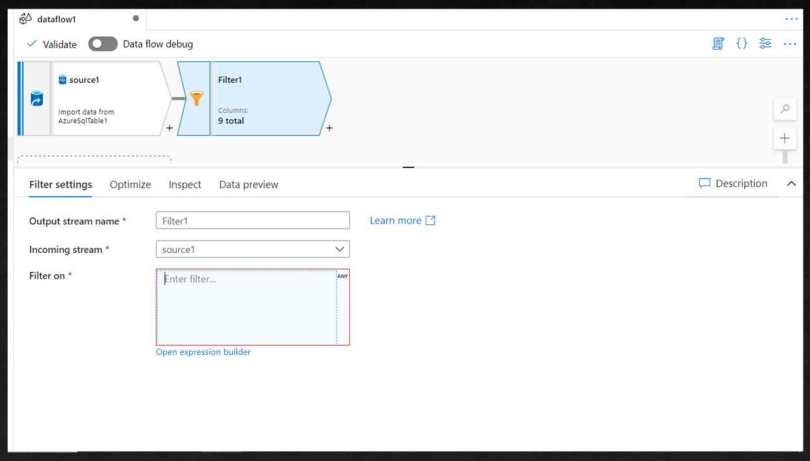


PAGE 66

66

Data Flow Transformations – FILTER

- One of the most common parts of data processing data is filtering the data to limit the scope of data and process it conditionally.

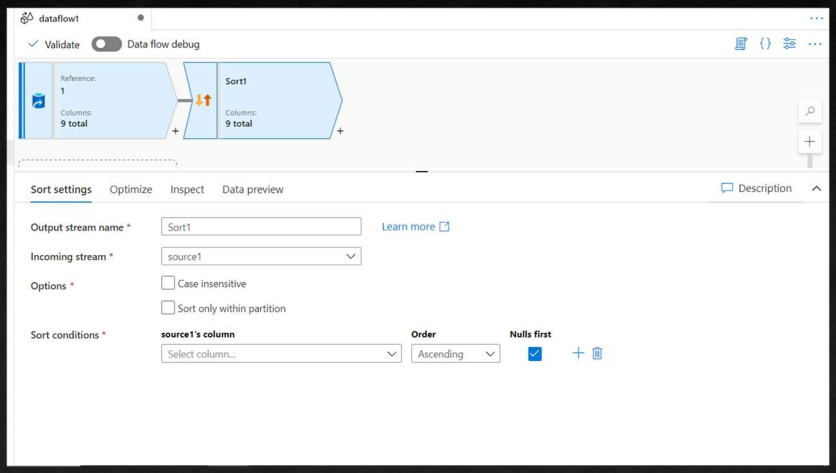


PAGE 67

67

Data Flow Transformations – SORT

- Another transform that goes together with filter transform is the sort transform.

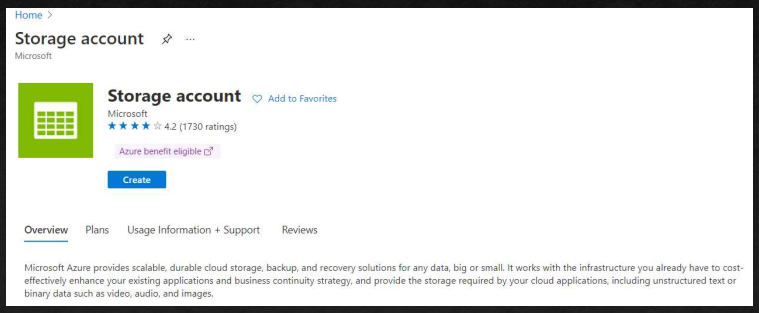


PAGE 68

68

Exercise: Create an Azure Storage Account (1 of 3)

- Sign into the Azure portal
- Select Create a resource and in the textbox that states "Search the Marketplace type Storage account and click on Storage account.
- In Storage account screen, click Create.

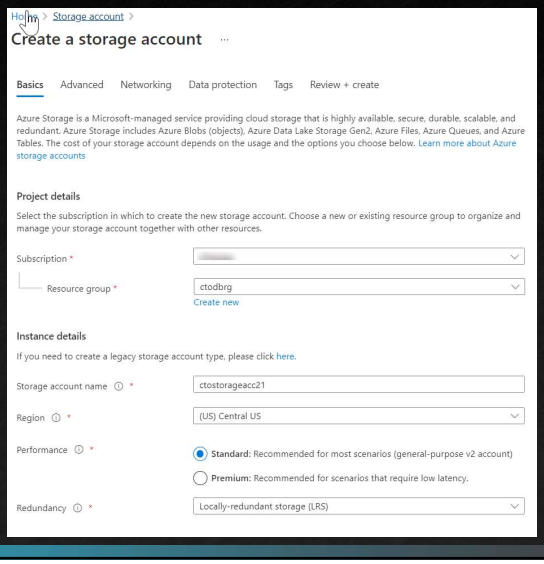


PAGE 75

75

Exercise: Create an Azure Storage Account (2 of 3)

- Next, in the **Create storage account** window, in the **Basics** tab, under Project details section, ensure that your subscription is selected, and the appropriate resource group.
- Under the Instance section, define a **storage account name**. Set the **Region** to **Central US**. In the **Performance** radio button list, select **Standard**, and set the Redundancy to **Locally redundant storage (LRS)**.



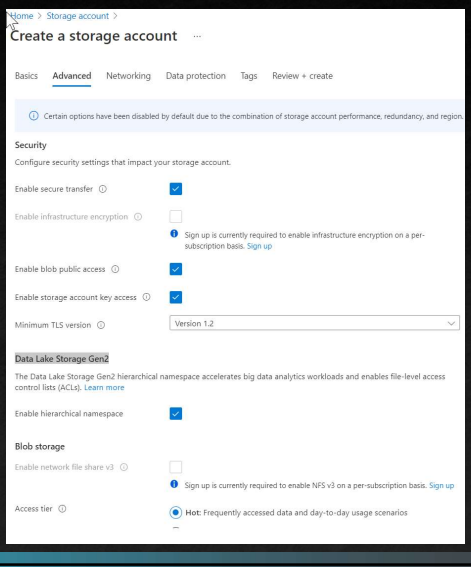
PAGE 76

76

Exercise: Create an Azure Storage Account (3 of 3)

- Select the **Advanced** tab. Under the section Data Lake Storage Gen2, click the checkbox next to **Enable hierarchical namespace**, as shown below.
- Select the Review + create tab, and click Create.

This new Azure Storage account is now set up to host data for an Azure Data Lake. After the account has deployed, you will find options related to Azure Data Lake in the Overview page.



PAGE 77

77

Introduction to Azure Data Lake Storage

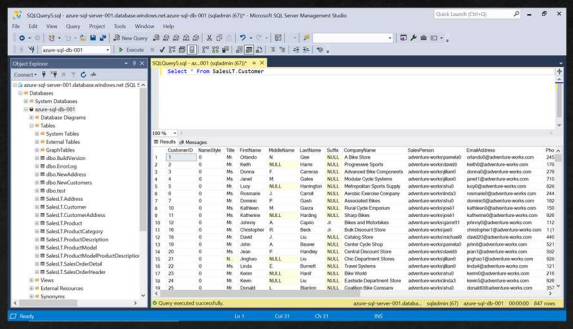
- Azure SQL Database is one of the most popular repositories for hosting transactional data.
- Azure Data Lake Storage Gen 2 forms the data lake storage layer, which is integrated with numerous data and analytics services on Azure like Azure Synapse Analytics, Azure Databricks, Azure Cognitive Services, and many more.
- Often there may be a need to export data out of the transactional databases to data lakes for different purposes.
- There are different ways of importing and exporting data out of the Azure SQL Database.
- One of the recommended ways of moving data within the Azure data ecosystem is by using Azure Data Factory.

PAGE 84

84

Initial Setup – Azure SQL Database Instance

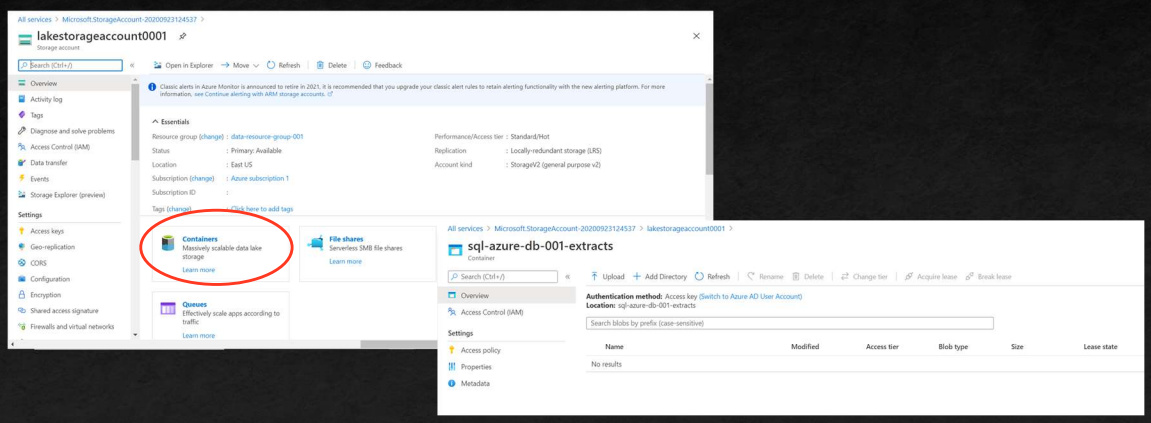
- There are a few pre-requisites that need to be in place before we can start working on the Azure Data Factory to export data from Azure SQL Database to Azure Data Lake Storage.
- As we are going to use Azure SQL Database as the data source, we need to have a database instance with some sample data in it, so that the same can be exported.



85

Initial Setup – ADLS Account & Containers

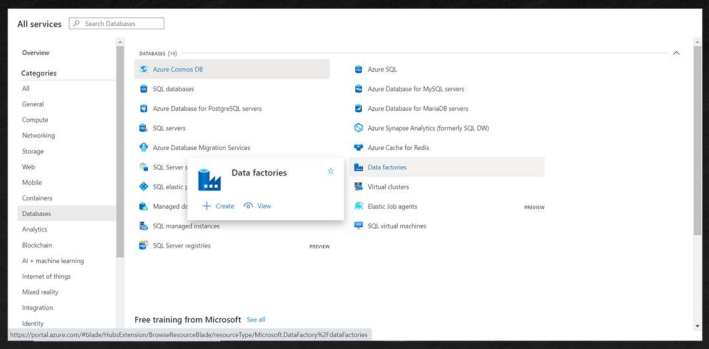
- As we intend to export the data into a container in a lake storage account, click on Containers and create a container in which the exported data would be stored.



86

Exercise: Creating an Instance in ADF

- To create a data pipeline in Azure Data Factory, we need to create an instance of Data Factory.
- Navigate to the All-services menu option, click on Databases and Click on Data factories. This would open the Data factory dashboard page.

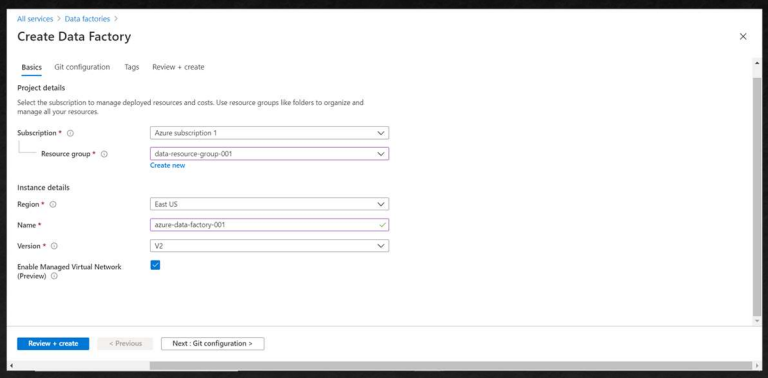


PAGE 87

87

Exercise: Creating an Instance in ADF

- Click on the Create Data Factory button and Fill up the basic details like the Subscription name, Resource group, Region, Name of the instance, and Version.
- The rest of the option can be the default value. Click on the Review + create button to create an Azure Data Factory instance.

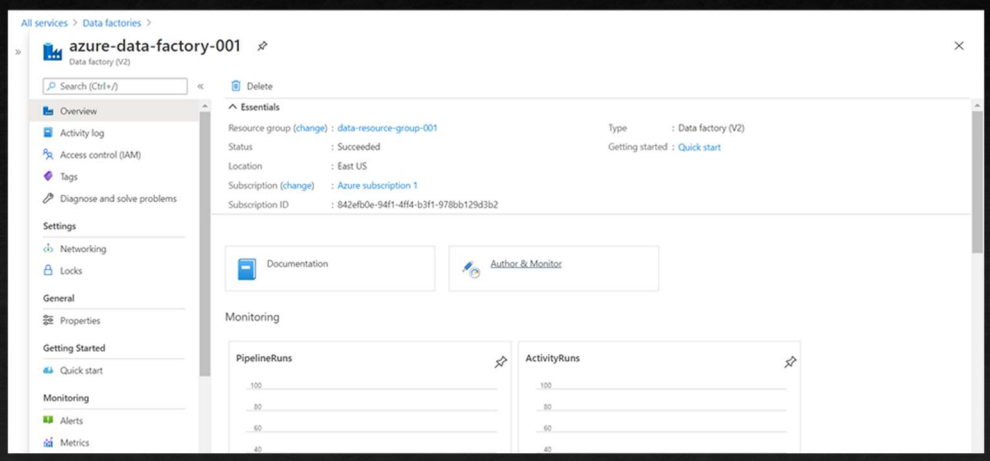


PAGE 88

88

Exercise: Creating Pipeline with ADF

- Open the instance and click on Author & Monitor button to open the Azure Data Factory portal.

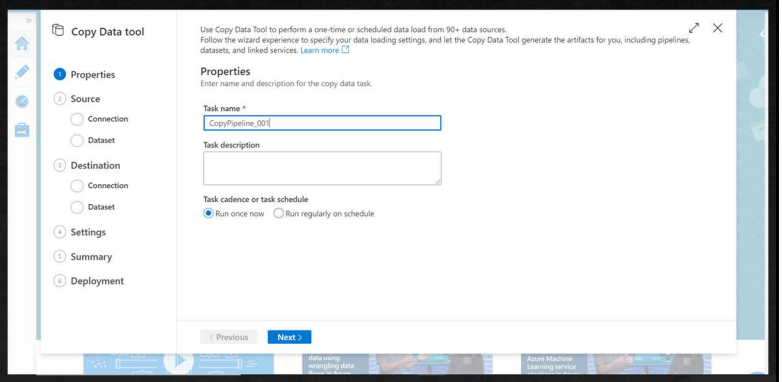


PAGE 89

89

Exercise: Creating Pipeline with ADF

- Once the portal opens, from the home page click on the Copy Data button to start the Copy Data tool or wizard.
- Provide the name of the data pipeline.
- Select the frequency of execution.

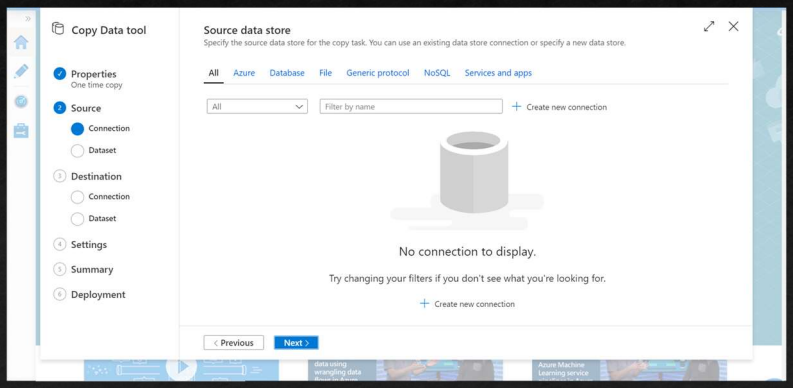


PAGE 90

90

Exercise: Creating Pipeline with ADF

- If you have not registered a linked service earlier, you may not have any data source connection listed
- Click on Create a new connection button to register Azure SQL Database as the data source.

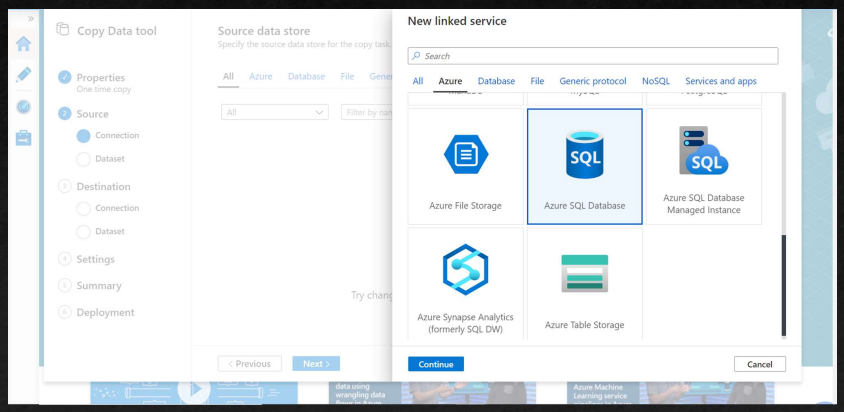


PAGE 91

91

Exercise: Exporting Data with ADF

- You would find the Azure SQL Database in the Azure list of data sources as shown below. Select the same and click on the Continue button.

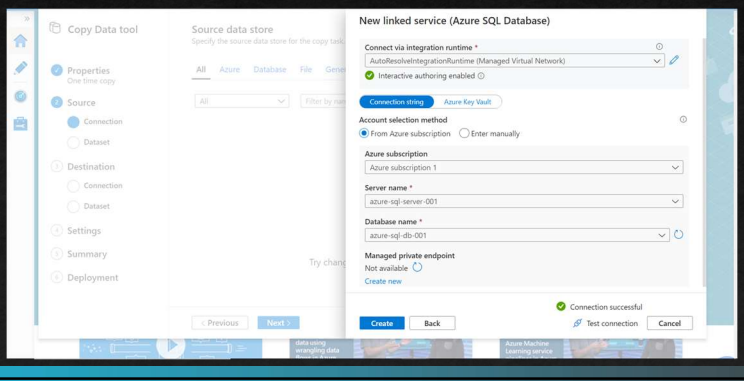


PAGE 92

92

Exercise: Exporting Data with ADF

- Provide connectivity details of the database
- If you intend to test the connection, add the integration runtime as well.
- After providing the connection credentials, click on the Test connection button to test the connectivity. Once the connection is successful, click on the Create button.

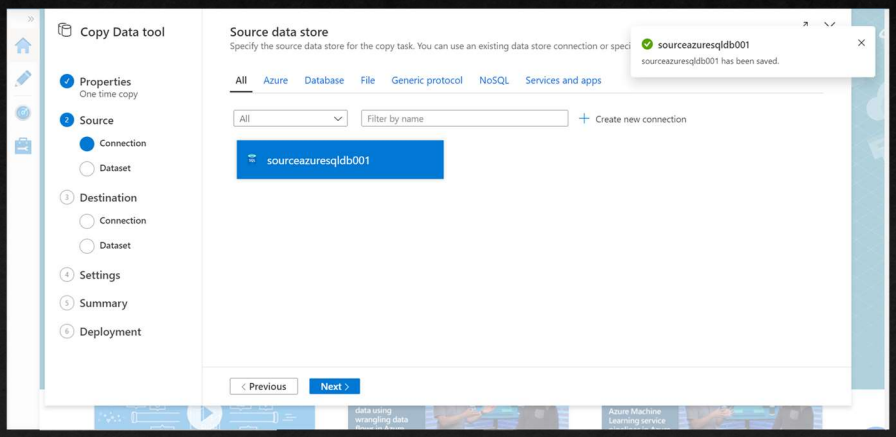


PAGE 93

93

Exercise: Exporting Data with ADF

- Once the data source is registered as a linked service, you would be able to find the same in the data source list. Click on the Next button.

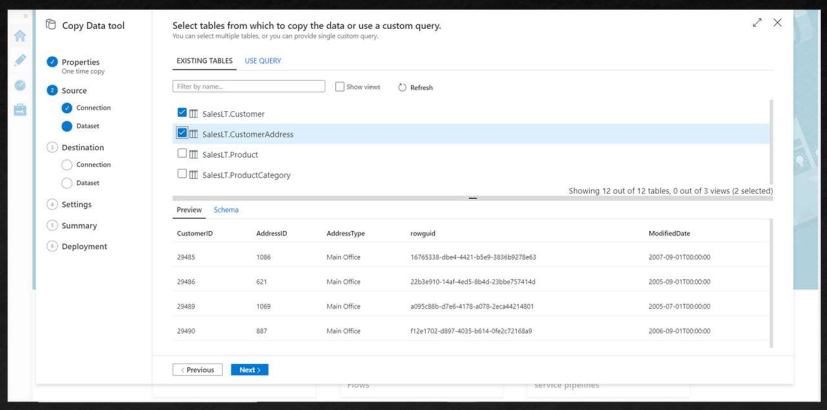


PAGE 94

94

Exercise: Exporting Data with ADF

- You would find the list of tables in the selected SQL database.
- In this case, we have selected two tables. Select the desired tables and click on the Next button.

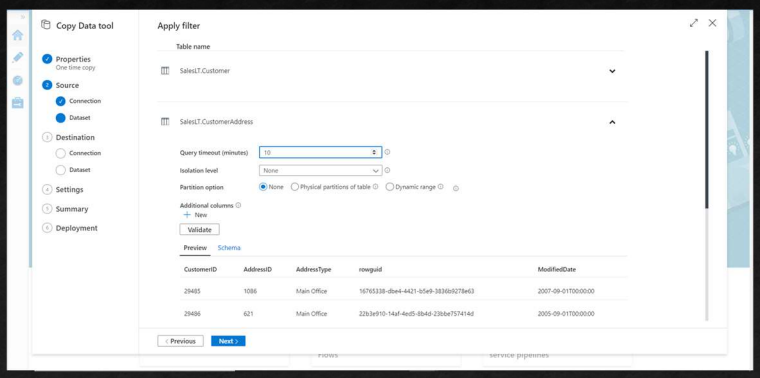


PAGE 95

95

Exercise: Exporting Data with ADF

- In this step, we can apply desired filters on each table separately.
- We can also add additional columns to the schema by clicking on the New button and then defining the details of the new columns.

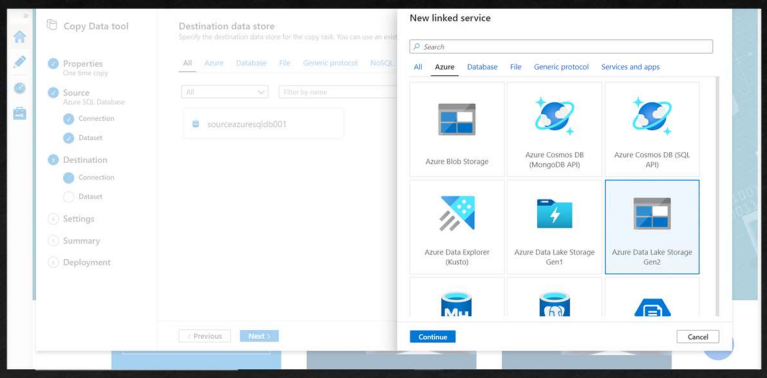


PAGE 96

96

Exercise: Exporting Data with ADF

- After that, we need to register the destination as a linked service.
- Repeat the steps that we followed when we registered the data source, and selected Azure Data Lake Storage Gen2.

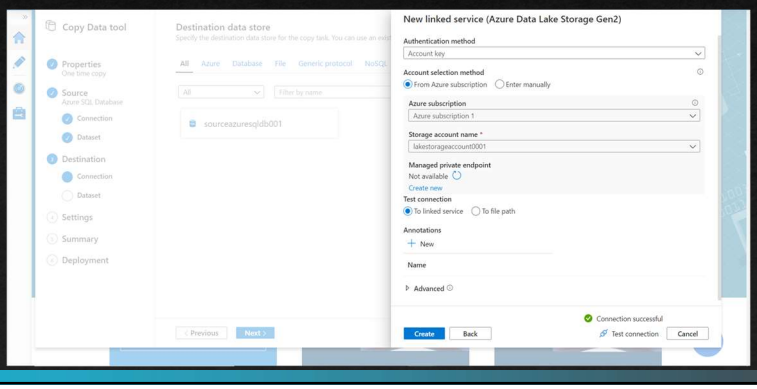


PAGE 97

97

Exercise: Exporting Data with ADF

- After that, provide the details of the storage account that we created in the pre-requisite section.
- You can select Account Key or Azure AD as the authentication method. We can test the connection to the linked service or to the exact file path.
- Click on the Create button to register the destination of the data pipeline.

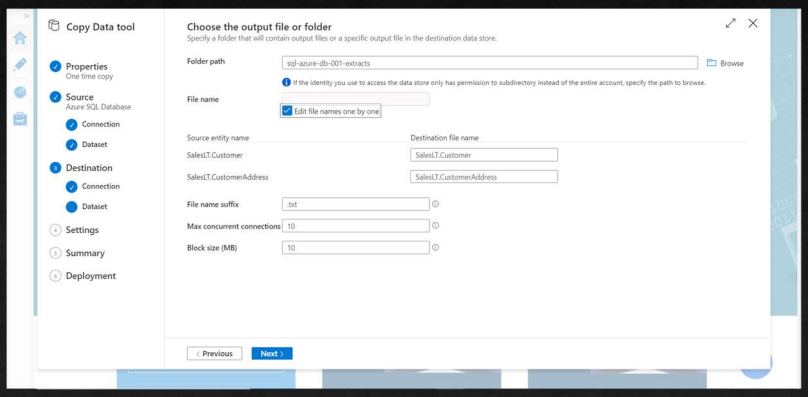


PAGE 98

98

Exercise: Exporting Data with ADF

- Under the data lake storage account, we need to specify the container i.e. the folder where we intend to save the exported result.
- Select the folder path and specify the file name for each exported table.

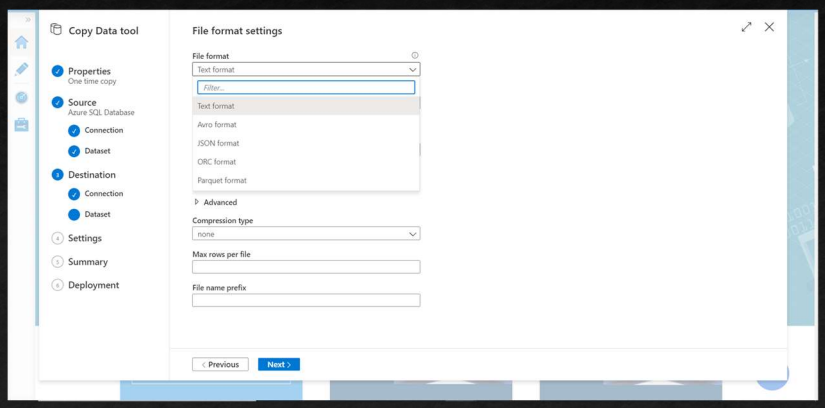


PAGE 99

99

Exercise: Exporting Data with ADF

- In the next step, specify file format settings like the file format, separator, compression etc. as shown below and click on the Next button.

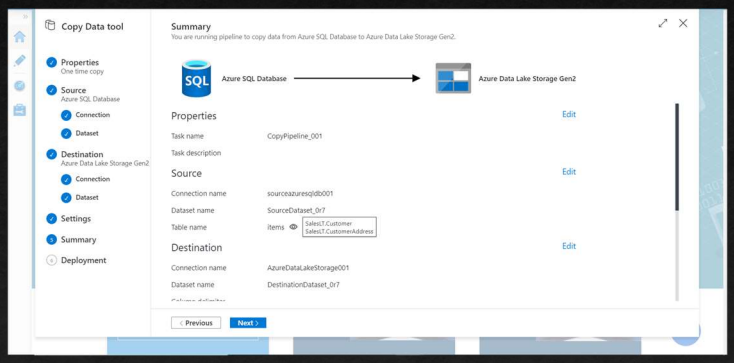


PAGE 100

100

Exercise: Exporting Data with ADF

- For the rest of the settings, we can continue with the default options.
- In the final step, we would find a Summary screen that would list all the details of the configuration we have specified so far as shown below.
- Click on the Next button the create as well as execute the pipeline.

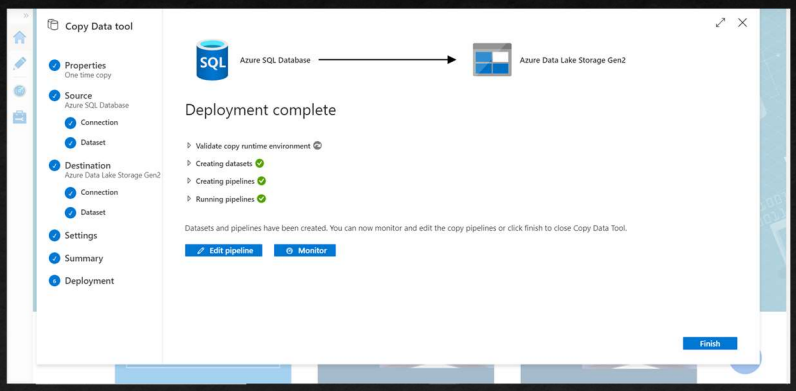


PAGE 101

101

Exercise: Exporting Data with ADF

- After the deployment is complete and the pipeline is executed, the successful confirmation status of the same should be available on the final step.
- This concludes the entire export process in case of a one-time export of the data.

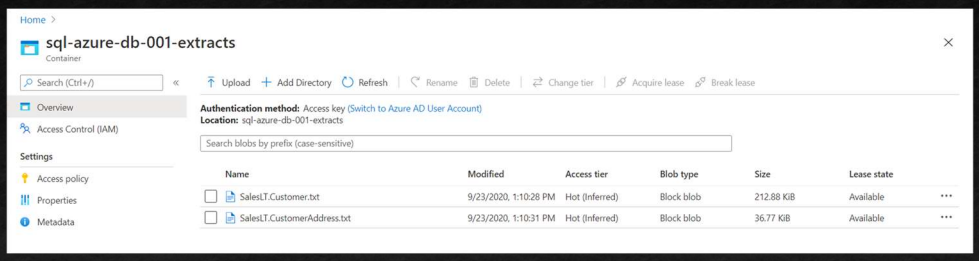


PAGE 102

102

Exercise: Exporting Data with ADF

- To confirm the successful export process, navigate to the container in the Azure Data Lake account and you should be able to find two files created with the same name that we specified while defining the pipeline.



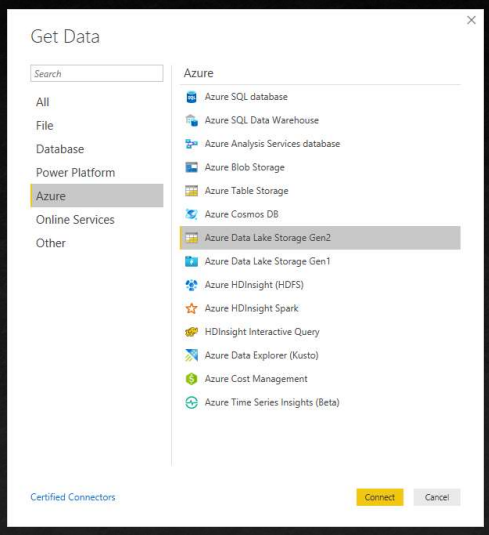
- Open any file to verify the data, and you should be able to see that the data got exported in a text file format with semi-comma as the separator, as per the configuration that we specified while defining the pipeline.

PAGE 103

103

Exercise: Load Data from ADLS to Power BI

- Launch Power BI Desktop on your computer.
- From the Home tab of the Ribbon, select Get Data, and then select More.
- In the Get Data dialog box, select Azure > Azure Data Lake Store Gen2, and then select Connect.

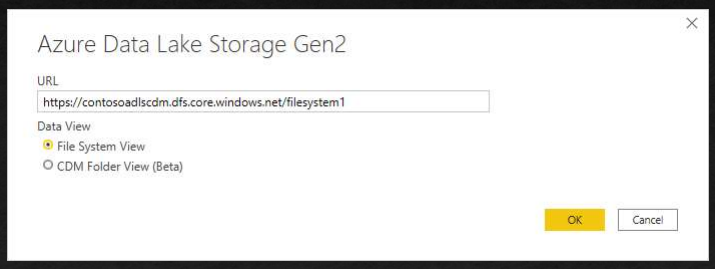


PAGE 107

107

Exercise: Load Data from ADLS to Power BI

- In the Azure Data Lake Storage Gen2 dialog box, you can provide the URL to your Azure Data Lake Storage Gen2 account, filesystem, or subfolder using the container endpoint format.
- URLs for Data Lake Storage Gen2 have the following pattern:
  - https://<accountname>.dfs.core.windows.net/<filesystemname>/<subfolder>
- Select **OK** to continue.

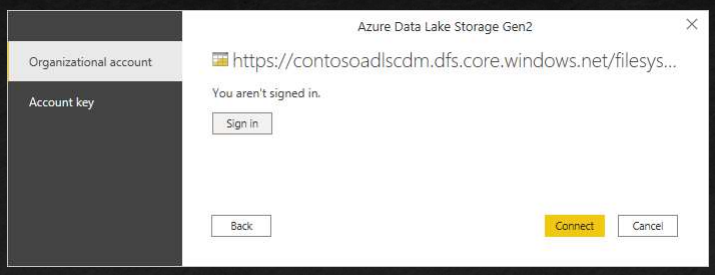


PAGE 108

108

Exercise: Load Data from ADLS to Power BI

- If this is the first time you're using this URL address, you'll be asked to select the authentication method.
- If you select the Organizational account method, select **Sign in** to sign into your storage account. You'll be redirected to your organization's sign in page. Follow the prompts to sign into the account. After you've successfully signed in, select **Connect**.
- If you select the Account key method, enter your account key and then select **Connect**.

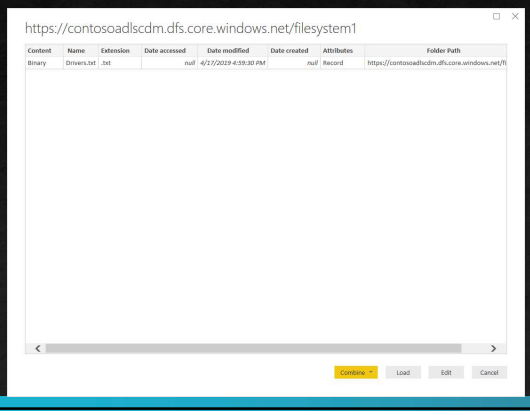


PAGE 109

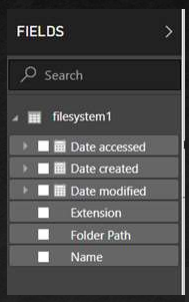
109

Exercise: Load Data from ADLS to Power BI

- The next dialog box shows all files under the URL you provided in step 4 above, including the file that you uploaded to your storage account. Verify the information, and then select **Load**.



- After the data has been successfully loaded into Power BI, you'll see the following fields in the Fields tab.



PAGE 110

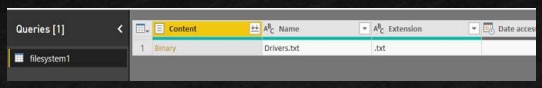
110

Exercise: Load Data from ADLS to Power BI

- However, to visualize and analyze the data, you might prefer the data to be available using the following fields.

0	1	2	3	4	5	6
1	Maria Anders	Obere Str. 57	Berlin		12209	Germany
2	Ana Trujillo	Avda. de la Constitució...	México D.F.		5021	Mexico

- In the next steps, you'll update the query to convert the imported data to the desired format. From the Home tab on the ribbon, select Edit Queries.



- In the **Query Editor**, under the **Content** column, select **Binary**. The file will automatically be detected as CSV and you should see an output as shown below. Your data is now available in a format that you can use to create visualizations.

PAGE 111

111