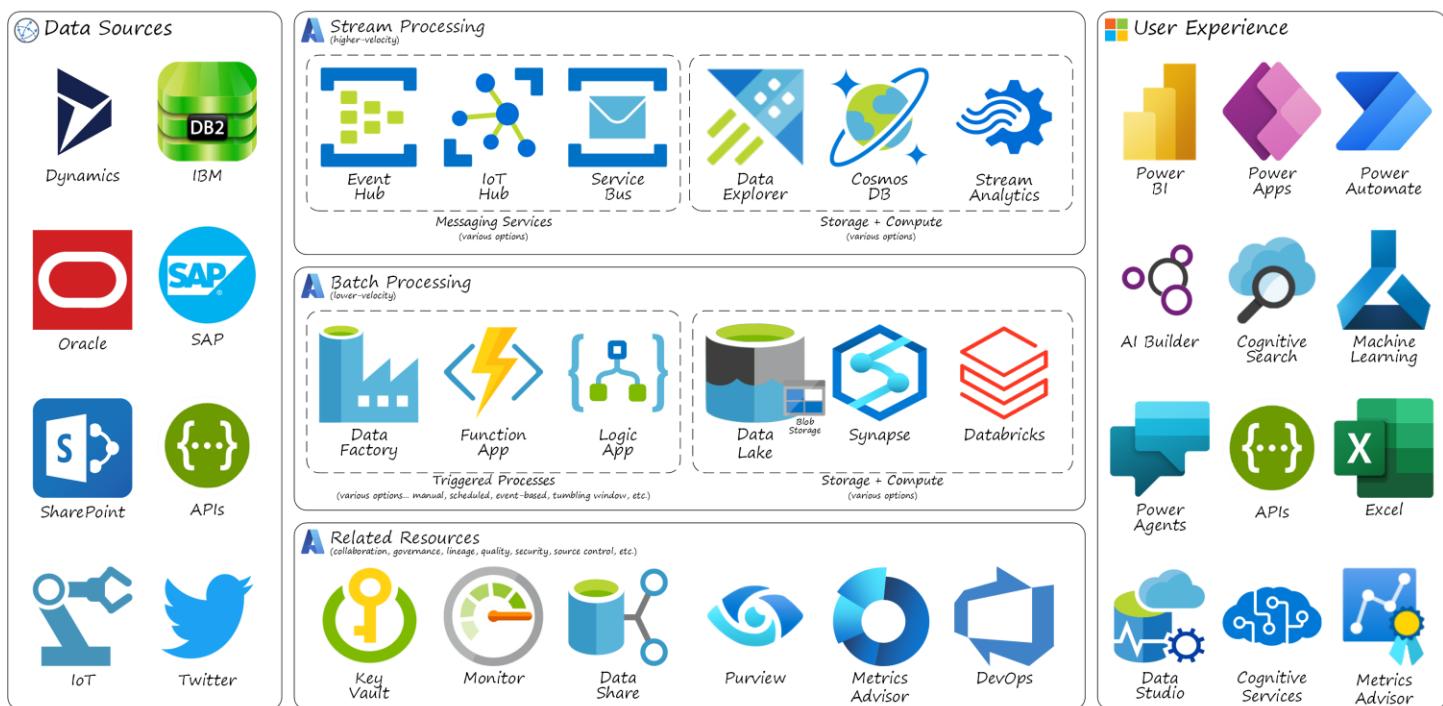


# Azure Data Practicum

This document is best viewed in Microsoft Word, "Web Layout" view with Navigation pane turned on

This practicum is an opportunistic capture of things I have learned while working with a variety of amazing solutions.



This document contains many **product | activity-focused objectives** organized into **discipline-specific sections**.

Consider navigating directly to those sections and objectives that align with your focus and interest.

## Table of Contents

DevOps.....	3
Objective: Synapse + DevOps   Source Control .....	3
Objective: Synapse   Schema Comparison .....	10
Engineering .....	14
Objective: Synapse   Use External Data.....	14
Objective: Synapse + Data Lake   Simple Data Ingestion.....	23
Objective: Synapse + SQL + Data Explorer   Simple Data Pipeline.....	36
Objective: Synapse   Simple Conditional Pipeline .....	44
Objective: Synapse (pipeline) + Data Explorer   Incremental Load .....	50
Objective: Synapse (data flow) + Delta Lake   Incremental Load .....	59
Objective: Databricks + Data Lake   Mount Data Lake .....	101
Objective: Databricks   Localize Mounted Data.....	105
Objective: Databricks   Source from APIs .....	107
Objective: Databricks   Batch Upsert Data .....	111
Objective: Data Explorer   Query from On-Prem.....	114
Analysis .....	118

Objective: Metrics Advisor + Data Explorer   Onboard and Investigate .....	118
Application .....	126
Objective: SQL + Power Apps + AI Builder   Capture Data .....	126
Governance .....	149
Objective: Purview   Discover Data .....	149
Objective: Purview   Classify Data .....	163
Objective: Purview   Understand Lineage .....	170
Monitoring .....	175
Objective: Synapse   Audit Data Usage .....	175
Appendix .....	179
Application Registration (aka Service Principal) .....	179
CLI .....	186
Data Explorer .....	187
Data Factory .....	191
Data Lake .....	195
Data Share .....	200
Data Studio .....	213
Databricks .....	215
DevOps .....	221
Key Vault .....	225
Log Analytics .....	226
Metrics Advisor .....	228
Postman .....	229
Power Apps .....	229
Purview .....	229
Resource Group .....	235
SQL .....	235
Storage Account .....	242
Synapse .....	243

# DevOps

Objective(s) in this section speak to use of Azure DevOps, Data Studio, etc. for source control and release management.

*Note: The DevOps section is intentionally first; DevOps is a critical foundation for effective collaboration, source control, etc.*

## Objective: Synapse + DevOps | Source Control

*Note: These instructions also apply (with minor differences) to Azure Data Factory.*

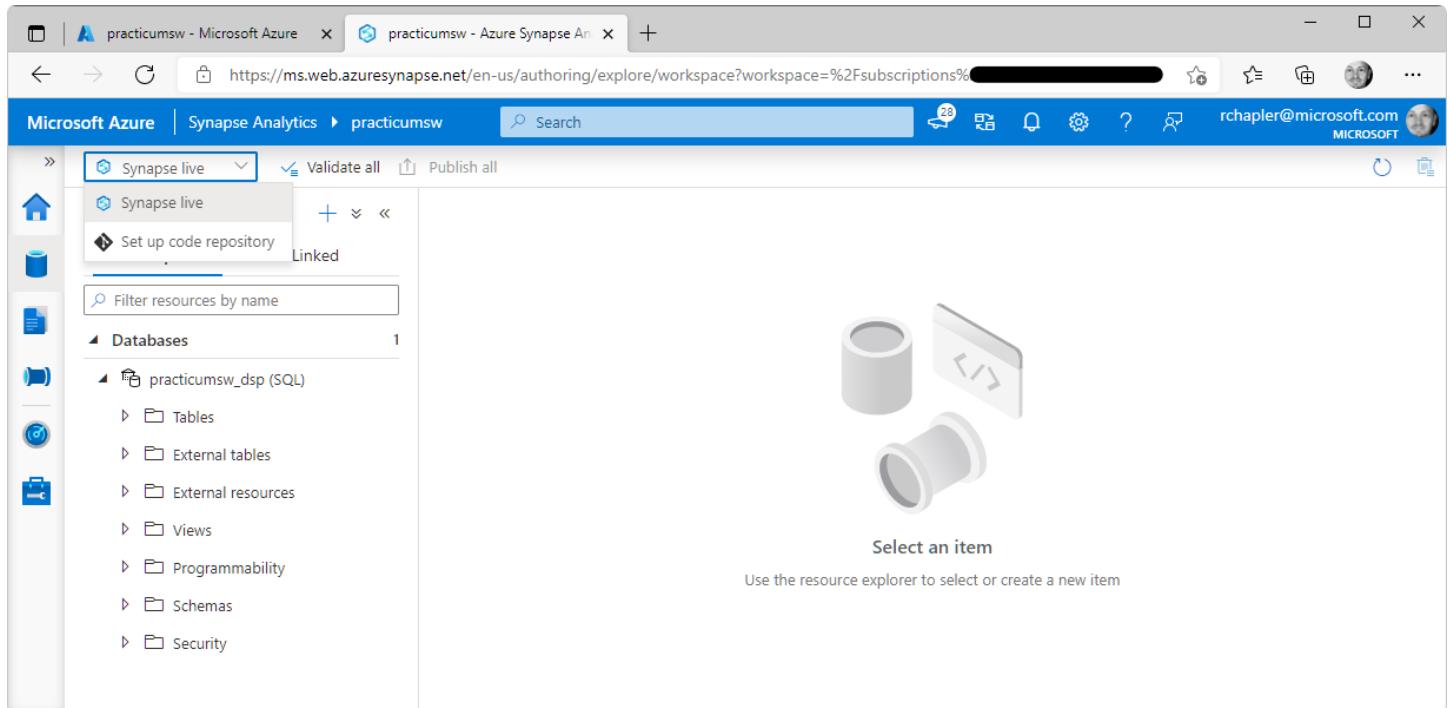
Follow the instructions in this section to **connect Synapse to DevOps, create branches, and pull requests**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- DevOps
- Synapse (with Dedicated SQL Pool)

### Setup Code Repository

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.



Click “**Synapse live**” and then select “**Set up code repository**” from the resulting drop-down.

The screenshot shows a Microsoft Azure browser interface with two tabs open: 'practicumsw - Microsoft Azure' and 'practicumsw - Azure Synapse Analytics'. The main content area displays a 'Configure a repository' dialog. On the left, there's a sidebar with icons for Home, Data, Machine Learning, and More, followed by a 'Data' section with 'Workspace' selected. The 'Workspace' section lists 'practicumsw\_dsp (SQL)' with sub-options like Tables, External tables, External resources, Views, Programmability, Schemas, and Security. At the top right of the main area, there's a search bar and a user profile for 'rchapler@microsoft.com MICROSOFT'. The 'Configure a repository' dialog has a title 'Configure a repository' and a sub-instruction 'Specify the settings that you want to use when connecting to your repository.' It contains two dropdown menus: 'Repository type \*' set to 'Azure DevOps Git' and 'Azure Active Directory' set to 'Microsoft'. At the bottom are 'Continue' and 'Cancel' buttons.

On the “Configure a repository” pop-out, enter values for the following items:

---

**Repository Type** Select “**Azure DevOps Git**” from the drop-down

**Azure Active Directory** Select the value appropriate for your organization

---

Click the **Continue** button.

The screenshot shows the Microsoft Azure portal interface. In the top navigation bar, there are two tabs: "practicumsw - Microsoft Azure" and "practicumsw - Azure Synapse Analytics". The URL in the address bar is <https://ms.web.azuresynthesize.net/en-us/authoring/explore/workspace?workspace=%2Fsubscriptions%2F...>. The main content area is titled "Configure a repository". On the left, there is a sidebar with icons for Home, Data, Workspace, Databases, Tables, External tables, External resources, Views, Programmability, Schemas, and Security. The "Data" section is selected, showing a "Workspace" tab with "Linked" status. A search bar at the top right contains the text "Search". The "Configure a repository" dialog has the following fields:

- Microsoft [redacted]**
- Select repository** (radio button selected)
- Azure DevOps organization name \***:
- Project name \***:
- Repository name \***:
- Collaboration branch \***:
- Publish branch \***:
- Root folder \***:
- Import existing resource**  
 Import existing resources to repository

At the bottom of the dialog are three buttons: "Apply" (blue), "Back", and "Cancel".

On the second “Configure a repository” pop-out, enter values for the following items:

Azure DevOps Organization...	Select your DevOps account
Project Name	Select your DevOps project
Repository Name	Select the repo created with your DevOps project
Collaboration Branch	Dropdown, click “+ Create new”, enter an appropriate name in the resulting pop-up, and click Create
Publish Branch	Confirm default, <code>workspace_publish</code>
Root Folder	Confirm default, <code>/</code>
Import Existing Resources...	Unchecked

Click the **Apply** button.

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. On the left, there's a sidebar with icons for Home, Data, Machine Learning, Databricks, and Pipelines. The main area shows a workspace named 'collaboration\_synapse' containing one database: 'practicumsw\_dsp (SQL)'. At the top right, there's a 'Set working branch' dialog box. It has two radio buttons: 'Create new' (unchecked) and 'Use existing' (checked). Below the radio buttons is a dropdown menu showing 'collaboration\_synapse'. At the bottom of the dialog is a blue 'Save' button.

Confirm values and then click the **Save** button.

When processing is complete, you will note that we are now working in the new branch.

## Confirm Success

In this section, we will confirm success by simulating a deployment workflow {i.e., modification, and then pull request}.

Modify one or more of the following:

- SQL Script
- Spark Notebook
- Data Flow
- Pipeline

## Pull Request

Next, we will promote changes from the collaboration branch to the master branch via Pull Request.

Click on the “**{collaboration} branch**” drop-down control and select “**Create pull request...**” from the resulting drop-down.

A third tab in your browser (pointing to <https://dev.azure.com...>) will open.

The screenshot shows the 'New pull request' interface in Azure DevOps. On the left, a sidebar lists various project management and CI/CD options like Overview, Boards, Repos, Pull requests, Pipelines, Test Plans, Artifacts, and Compliance. The 'Pull requests' option is currently selected. The main content area displays a form for creating a new pull request. At the top, it shows the source branch as 'collaboration\_synapse' and the target branch as 'master'. Below this, tabs for 'Overview', 'Files' (6), and 'Commits' (4) are visible. The 'Overview' tab is active. The main form fields include:

- Title:** A text input field containing 'My Pull Request'.
- Description:** A large text area with placeholder text 'Describe the code that is being reviewed'.
- Markdown supported:** A note indicating that Markdown is supported, with a link to 'Link work items'.
- Reviewers:** A search bar labeled 'Search users and groups to add as reviewers'.
- Work items to link:** A dropdown menu labeled 'Search work items by ID or title'.
- Tags:** An empty text input field.

A prominent blue 'Create' button is located at the bottom right of the form.

On the second “**New pull request**” page, enter values for the following items:

<b>Branches</b>	{collaboration branch} into {master branch} ... you may have to create a master branch if one does not exist
<b>Title</b>	Enter a title that is meaningful for you (and aligned with your naming standards)

Enter values for items like **Reviewers** and “**Work items...**” as appropriate.

Review settings on remaining tabs {e.g., **Files** and **Commits**}.

No additional changes are required.

Click the **Create** button.

The screenshot shows the Azure DevOps interface for a pull request. The left sidebar is titled 'practicumadop' and includes links for Overview, Boards, Repos, Files, Commits, Pushes, Branches, Tags, Pull requests, Pipelines, Test Plans, Artifacts, and Compliance. The 'Pull requests' link is currently selected. The main content area is titled 'My Pull Request' and shows an active pull request from Rich Chapler for 'collaboration\_synapse' into 'master'. The pull request has been checked for merge conflicts, which are listed as 'No merge conflicts' (Last checked Just now). Below this is a 'Description' section with a placeholder 'Add a comment...'. A message at the bottom indicates that Rich Chapler created the pull request 'Just now'. To the right of the pull request details are sections for 'Reviewers' (Required: No required reviewers; Optional: No optional reviewers), 'Tags' (No tags), and 'Work items' (No work items). At the top right of the main content area are buttons for 'Approve' and 'Complete'.

At this point (and if specified), designated reviewers would be notified by email, and they will need to review and approve for the Pull Request to move forward.

Additional configuration can provide for validation of other gating criteria {e.g., inclusion of text patterns that might be secret like an account key}.

We have not included gating factors in this example.

Click the **Complete** button.

The screenshot shows the same Azure DevOps interface as before, but with a modal dialog box titled 'Complete pull request' overlaid. The dialog contains fields for 'Merge type' (set to 'Merge (no fast forward)'), a diagram illustrating the merge process, and 'Post-completion options' with three checkboxes: 'Complete associated work items after merging' (checked), 'Delete collaboration\_synapse after merging' (checked), and 'Customize merge commit message' (unchecked). At the bottom of the dialog are 'Cancel' and 'Complete merge' buttons.

Confirm selections in the “Complete pull request” pop-out.

Click the “Complete merge” button.

The screenshot shows the Azure DevOps interface for a pull request titled "My Pull Request". The status bar at the top indicates "Completed" with a timestamp "Just now". Below the title, it says "Rich Chapler collaboration\_synapse into master". The "Overview" tab is selected. A message box displays "Rich Chapler completed this pull request Just now" with "Cherry-pick" and "Revert" buttons. Another message box shows "Merged PR 2: My Pull Request eda684b1 Rich Chapler Just now". A "Show details" link is present. A green checkmark indicates "No merge conflicts Last checked Just now". The "Reviewers" section shows "Required" and "Optional" sections both with "No required reviewers" and "No optional reviewers". The "Tags" section shows "No tags". The "Work items" section shows "No work items". At the bottom, there is a comment input field "Add a comment..." and a message from "Rich Chapler completed the pull request Just now".

Note: Post-completion options such as “**Cherry-pick**” and **Revert** provide for operational control even after a code merge.

## Objective: Synapse | Schema Comparison

Follow the instructions in this section to **compare two database schemas** {e.g., old, and new}.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Studio
- Synapse

Although this example focuses on Synapse, the same solution applies to SQL.

### Stage Resources

Navigate to your Synapse Workspace.

The screenshot shows the Microsoft Azure portal interface for a Synapse workspace named 'practicumsaw'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (with options for Active Directory admin, Properties, and Locks), Analytics pools (SQL pools, Apache Spark pools, Scope pools), and a 'Getting started' section with links to Open Synapse Studio and Read documentation. The main content area displays workspace details such as Resource group (practicumrg), Status (Succeeded), Location (West US 2), Subscription (rchapler), Subscription ID (91e9fddc-ef15-416c-9be1-085f8b1b46ed), Managed virtual network (No), Managed Identity object ID (78190823-9f09-4ed0-94af-118b5e80a55b), Workspace web URL (https://web.azure-synapse.net?workspace=%2...), and various endpoint URLs. A 'Delete' button is visible at the top right of the main content area.

Follow the directions at [Instantiate Resources | Synapse | SQL Pool](#) to create two SQL Pools named **practicumdsp\_dev** and **practicumdsp\_prod**.

We will delete both SQL Pools at the end of this objective.

Open Azure Data Studio.

Azure Data Studio interface showing a query window. The connection is set to 'practicumsaw.sql.azuresynapse.net,practicumdsp\_prod'. The query is:

```
CREATE TABLE [dbo].[table1]
(
    [column1] [nvarchar](128) NULL,
    [column2] [nvarchar](128) NULL,
    [column3] [nvarchar](128) NULL
```

The 'Messages' pane shows the execution results:

```
9:29:47 AM Started executing_query_at Line 1
Commands completed successfully.
Total execution time: 00:00:01.257
```

Bottom status bar: Ln 6, Col 2, Spaces: 4, UTF-8, CRLF, SQL, MSSQL, 0 rows, 00:00:01, practicumsaw.sql.azuresynapse.net : practicumdsp\_prod.

Execute the following T-SQL on practicumdsp\_dev:

```
CREATE TABLE [dbo].[table1]
(
    [column1] [nvarchar](128) NULL,
    [column2] [nvarchar](128) NULL,
    [column3] [nvarchar](128) NULL
)
```

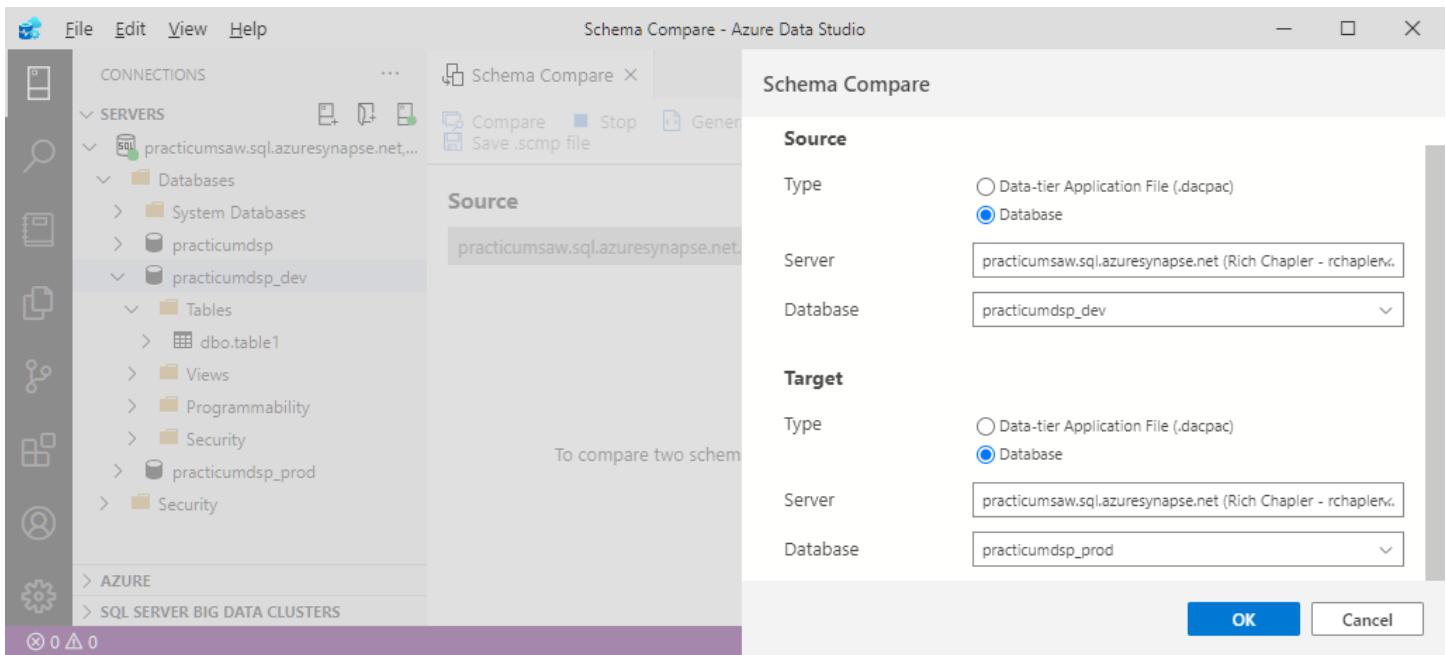
## Schema Compare

Azure Data Studio interface showing the 'Schema Compare' feature. The connection is set to 'practicumsaw.sql.azuresynapse.net,practicumdsp\_prod'. The 'Source' dropdown is set to 'practicumsaw.sql.azuresynapse.net.practicu...'. A context menu is open over the 'Tables' node under 'practicumdsp\_dev', with the 'Schema Compare' option highlighted.

To compare two schemas, first select a source schema and target schema, then press Compare.

Bottom status bar: Choose SQL Language.

Right-click on the SQL Pool and click “**Schema Compare**” in the resulting drop-down.

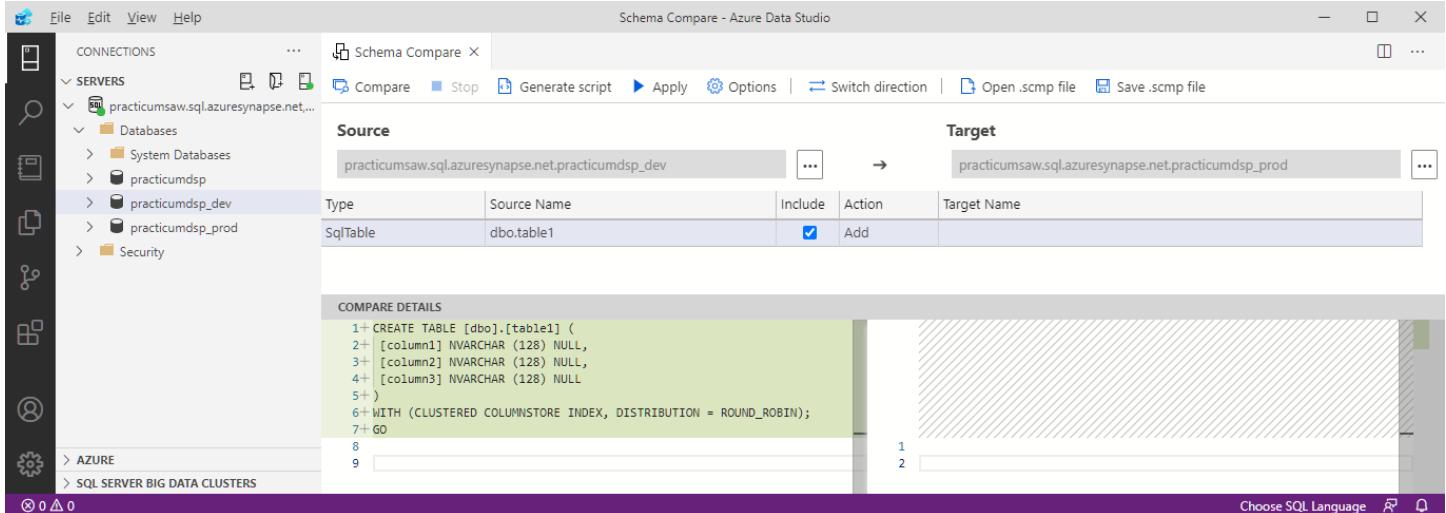


Click the ellipses button.

On the “Schema Compare” pop-out, enter values for the following items:

<b>Source   Type</b>	Select the <b>Database</b> radio button
<b>Source   Server and Database</b>	Confirm selection of your Synapse server and the <b>practicumdsp_dev</b> database
<b>Target   Type</b>	Select the <b>Database</b> radio button
<b>Target   Server and Database</b>	Confirm selection of your Synapse server and the <b>practicumdsp_prod</b> database

Click the **OK** button.



On the “Schema Compare” tab, click the **Compare** button.

We only created a table on the development database, so Schema Compare should surface that table as something missing from the production database.

Click on the row to see T-SQL for the identified item.

Click the **Apply** button to update the target.

File Edit View Help

Schema Compare - Azure Data Studio

Servers: practicumsaw.sql.azuresynapse.net...

Databases: System Databases, practicumdsp, practicumdsp\_dev, practicumdsp\_prod, Security

AZURE: SQL SERVER BIG DATA CLUSTERS

Choose SQL Language: Choose SQL Language

Schema Compare X

Compare Stop Generate script Apply Options Switch direction Open .scmp file Save .scmp file

Source: practicumsaw.sql.azuresynapse.net.practicumdsp\_dev

Target: practicumsaw.sql.azuresynapse.net.practicumdsp\_prod

Type: Source Name Include Action Target Name

PROBLEMS OUTPUT TERMINAL TASKS

✓ Apply schema compare changes succeeded practicumsaw.sql.azuresynapse.net | practicumdsp\_prod 9:47:58 AM - 9:48:09 AM (00:00:10)

# Engineering

Objectives in this section describe methods for extracting, transforming, and loading data.

## Objective: Synapse | Use External Data

Follow the instructions in this section to **explore methods of using data from assets external to Synapse**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Lake (with sample data)
- Synapse (with Serverless SQL Database and Apache Spark Pool)

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Data** icon in the left-hand navigation.

The screenshot shows the Microsoft Azure portal interface for a Synapse Analytics workspace named "practicumsw". The left sidebar has a "Data" icon selected. The main content area is titled "practicumdlc" and shows a list of files. The table below lists the files:

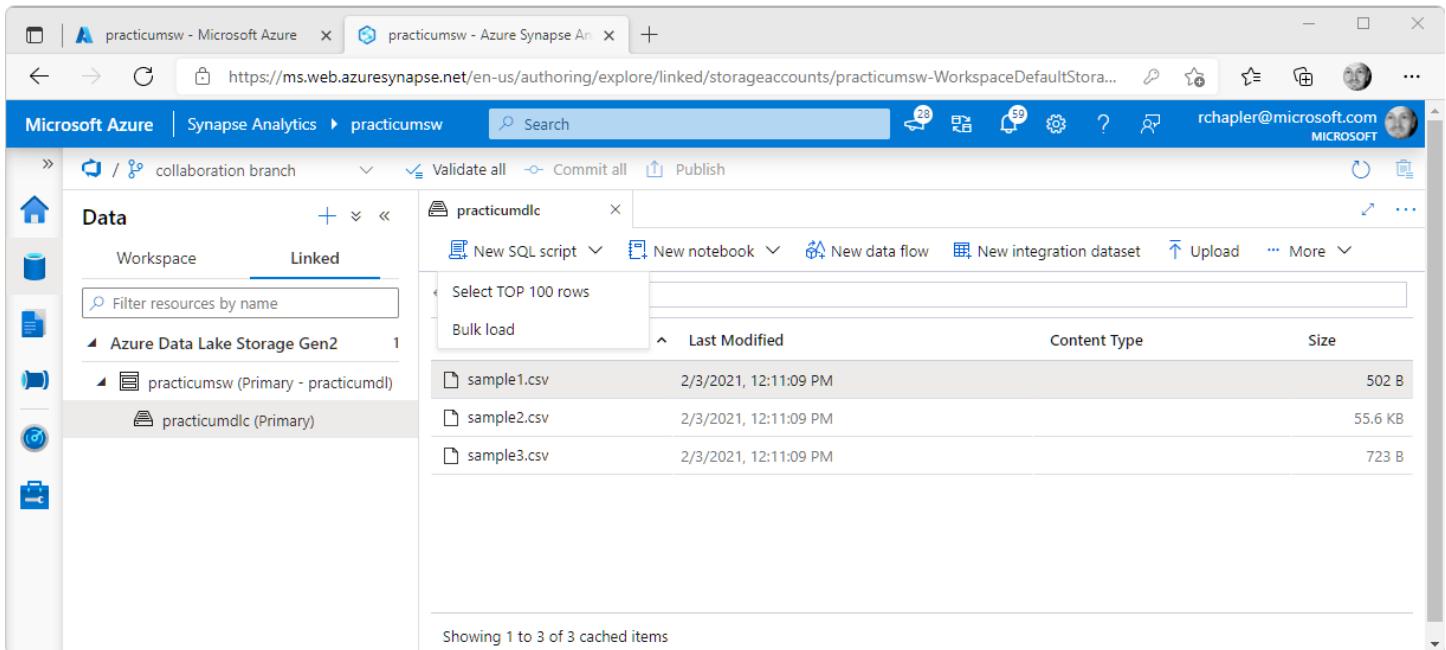
Name	Last Modified	Content Type	Size
sample1.csv	2/3/2021, 12:11:09 PM		502 B
sample2.csv	2/3/2021, 12:11:09 PM		55.6 KB
sample3.csv	2/3/2021, 12:11:09 PM		723 B

At the bottom of the list, it says "Showing 1 to 3 of 3 cached items".

Click the **Linked** tab and Expand navigation.

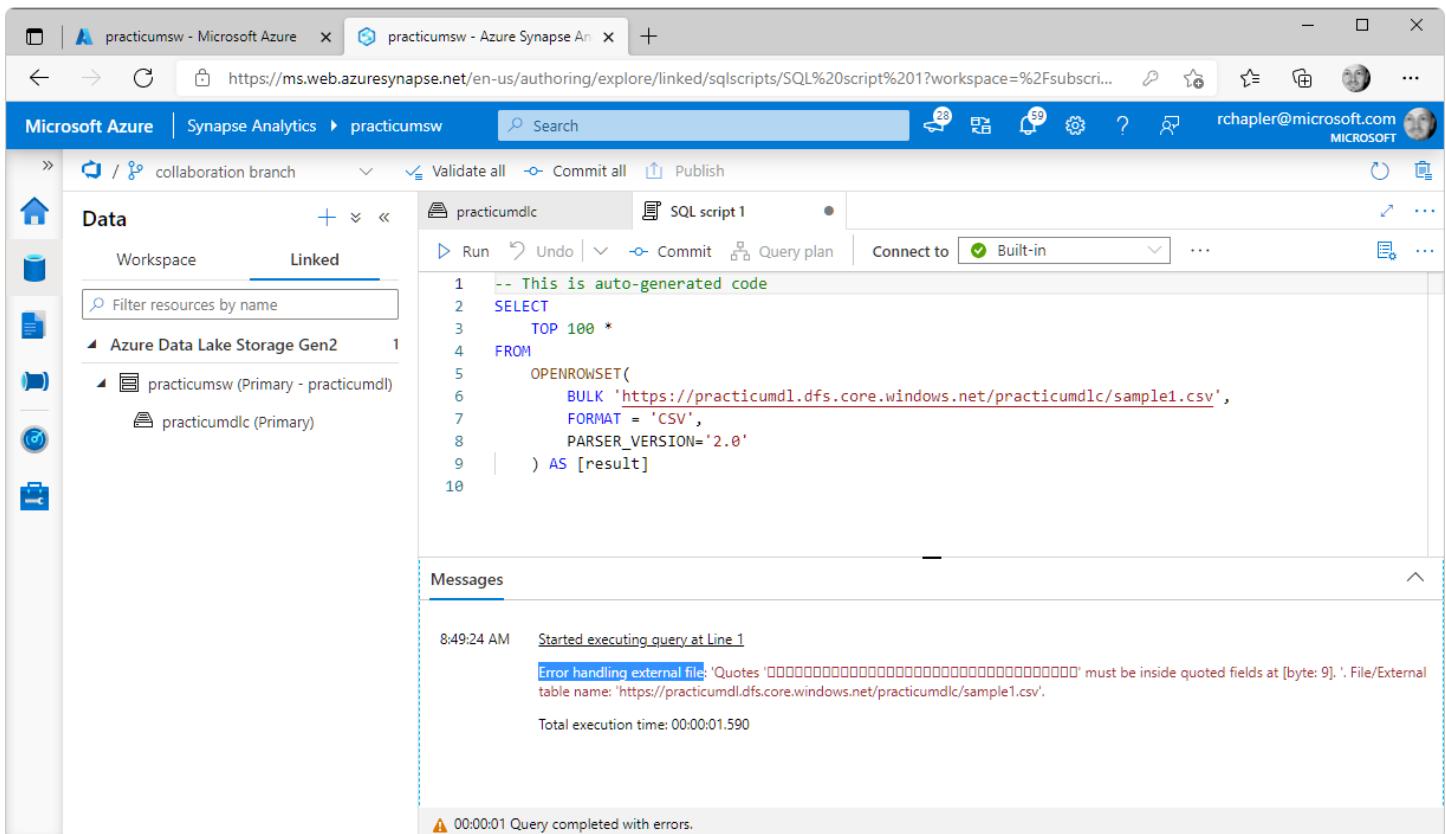
You will see that the Azure Data Lake Storage Gen2 specified during Synapse instantiation {i.e., `practicumdlc`} is surfaced with no additional configuration.

## Method #1: SQL Script (OPENROWSET)



The screenshot shows the Microsoft Azure Synapse Analytics Data blade. On the left, there's a navigation pane with icons for Home, Data, Workspace, and Linked. Under 'Linked', it shows 'practicumdlc' with three CSV files: 'sample1.csv', 'sample2.csv', and 'sample3.csv'. Each file has a timestamp of '2/3/2021, 12:11:09 PM' and a size: 502 B, 55.6 KB, and 723 B respectively. Below the table, it says 'Showing 1 to 3 of 3 cached items'.

Click the “New SQL script” button and “SELECT TOP 100 rows” from the resulting drop-down.



The screenshot shows the Microsoft Azure Synapse Analytics SQL script editor. The top bar shows the URL 'https://ms.web.azure-synapse.net/en-us/authoring/explore/linked/sqlscripts/SQL%20script%201?workspace=%2Fsubscriptions%2F...'. The main area contains a SQL script named 'SQL script 1' with the following code:

```
1 -- This is auto-generated code
2 SELECT
3     TOP 100 *
4 FROM
5     OPENROWSET(
6         BULK 'https://practicumdl.dfs.core.windows.net/practicumdlc/sample1.csv',
7         FORMAT = 'CSV',
8         PARSE_DATE='2021-02-03T12:11:09.000Z'
9     ) AS [result]
10
```

Below the code, the 'Messages' section shows:

- 8:49:24 AM Started executing query at Line 1
- Error handling external file: 'Quotes '\u00e2\u20ac\u201d must be inside quoted fields at [byte: 9]. ' . File/External table name: 'https://practicumdl.dfs.core.windows.net/practicumdlc/sample1.csv'.
- Total execution time: 00:00:01.590

At the bottom, a message states: 00:00:01 Query completed with errors.

The resulting auto-generated code uses OPENROWSET(...) and a built-in, serverless SQL Pool to pull data directly from the CSV file in the ADLS container.

When we run the code (assuming you are using the same sample files, of course), we get an error about quote-handling.

Given the formatting of the sample file, we cannot run this without changes to the code.

Update the code to:

```

SELECT TOP 100 *
FROM OPENROWSET(
    BULK 'https://practicumdl.dfs.core.windows.net/practicumdlc/sample3.csv',
    FORMAT = 'CSV',
    PARSER_VERSION='2.0',
    HEADER_ROW = TRUE,
    FIELDQUOTE = ''
) AS [result]

```

... and then, re-Run...

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. On the left, there's a navigation sidebar with icons for Home, Data, Workspace, and Linked. The 'Linked' tab is selected. Below it, under 'Azure Data Lake Storage Gen2', is a list with 'practicumsw (Primary - practicumdl)' expanded, showing 'practicumdlc (Primary)' as a child item. The main area has a title bar 'practicumsw - Microsoft Azure' and 'practicumsw - Azure Synapse An'. The URL in the address bar is 'https://ms.web.azuresynapse.net/en-us/authoring/explore/linked/sqlscripts/SQL%20script%201?workspace=%2Fsubscr...'. The workspace name 'practicumsw' is also visible in the top right. The central workspace area contains a code editor with a SQL script titled 'practicumdlc' and 'SQL script 1'. The script is identical to the one above. Below the code editor is a results pane with tabs for 'Results' (selected) and 'Messages'. The 'Results' tab shows a table view with two columns: 'Game Number' and 'Game Length'. The data rows are: Game Number 1, Game Length 30; Game Number 2, Game Length 29; Game Number 3, Game Length 31. At the bottom of the results pane, a message says '00:00:11 Query executed successfully.'

## Method #2: External Table (Serverless SQL Database)

Click the **Workspace** tab.

External Data Source

Expand navigation to and then right-click on “External data sources”.

The screenshot shows the Microsoft Azure Synapse Analytics workspace Data blade. In the left sidebar, under 'External resources', the 'External data sources' item is selected. A context menu is open over this item, with 'New SQL script' highlighted. Other options in the menu include 'New external data source' and 'Refresh'. The main pane displays a placeholder message: 'Select an item' with the sub-instruction 'Use the resource explorer to select or create a new item'. There are icons for databases and external tables.

Click “New SQL script”, and then “New external data source” in the drop-down.

The screenshot shows the Microsoft Azure Synapse Analytics workspace Data blade. The SQL script editor is open, displaying the following T-SQL code:

```
1 CREATE EXTERNAL DATA SOURCE [ExternalDataSource] WITH
2 (
3     LOCATION = 'https://practicumdl1.blob.core.windows.net/practicumdlc'
4 )
5
```

The results pane shows the execution log:

- 9:29:07 AM Started executing query at Line 1
- Total execution time: 00:00:03.449
- 00:00:03 Query executed successfully.

Replace <STORAGEACCOUNT> and <CONTAINER> in the auto-generated code and then Run.

## External File Format

Expand navigation to and then right-click on “External file formats”.

The screenshot shows the Microsoft Azure Synapse Analytics workspace Data blade. In the left sidebar, under the 'Data' category, 'External resources' is expanded, and 'External file formats' is selected. A context menu is open over this item, with 'New SQL script' highlighted. Other options in the menu include 'New external file format', 'Refresh', and 'Select an item'. The main pane displays a large icon of two pipes and a code editor window.

Click “New SQL script”, and then “New external file format” in the drop-down.

The screenshot shows the Microsoft Azure Synapse Analytics workspace Data blade. In the left sidebar, under the 'Data' category, 'External resources' is expanded, and 'External file formats' is selected. A SQL script titled 'SQL script 1' is displayed in the main pane, containing the following code:

```
1 CREATE EXTERNAL FILE FORMAT [ExternalFileFormat] WITH
2 (
3     FORMAT_TYPE = DELIMITEDTEXT, FORMAT_OPTIONS ( FIELD_TERMINATOR = ',', FIRST_ROW = 2 )
4 )
```

The 'Results' tab at the bottom shows the execution log:

```
10:24:53 AM Started executing query at Line 1
Total execution time: 00:00:02.705
```

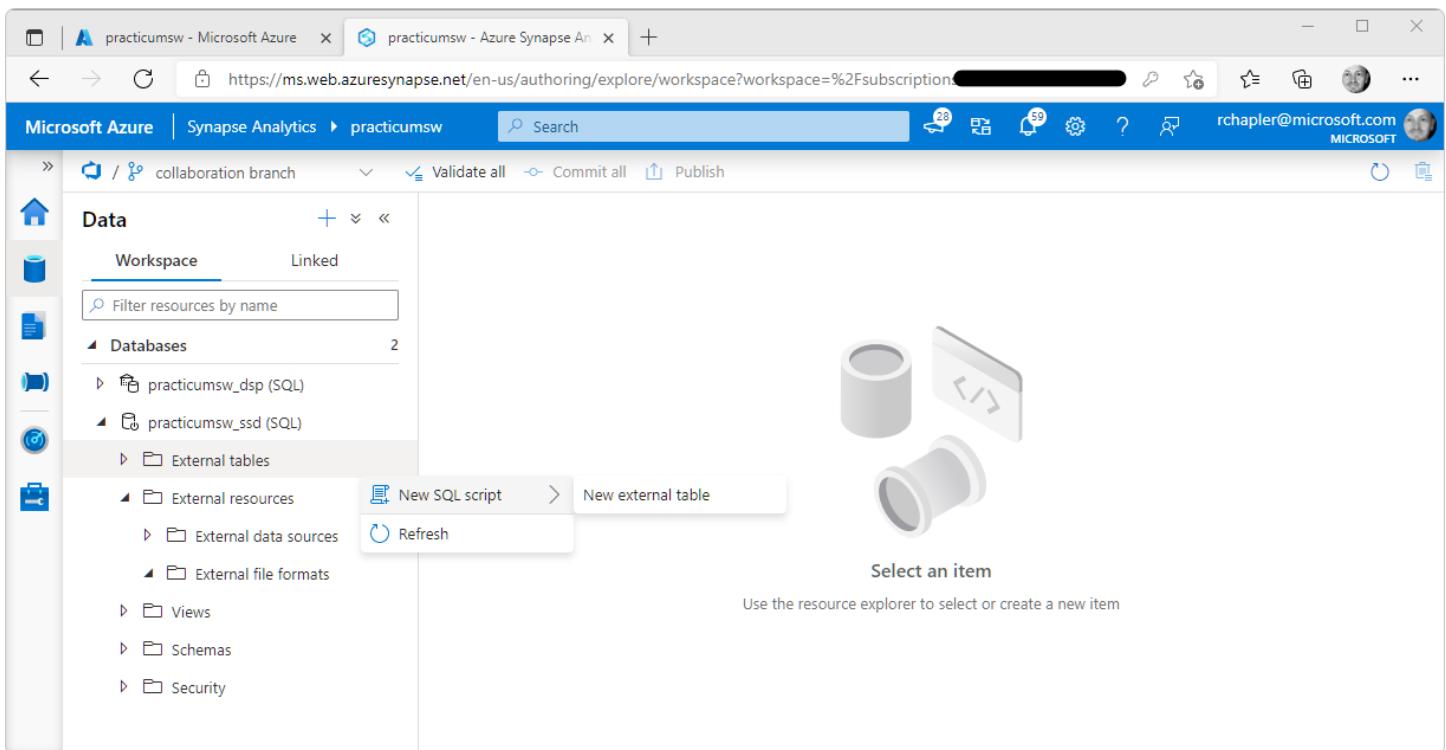
A message at the bottom indicates the query was executed successfully:

```
00:00:02 Query executed successfully.
```

Append FORMAT\_OPTIONS arguments FIELD\_TERMINATOR and FIRST\_ROW to the default FORMAT\_TYPE argument and then Run.

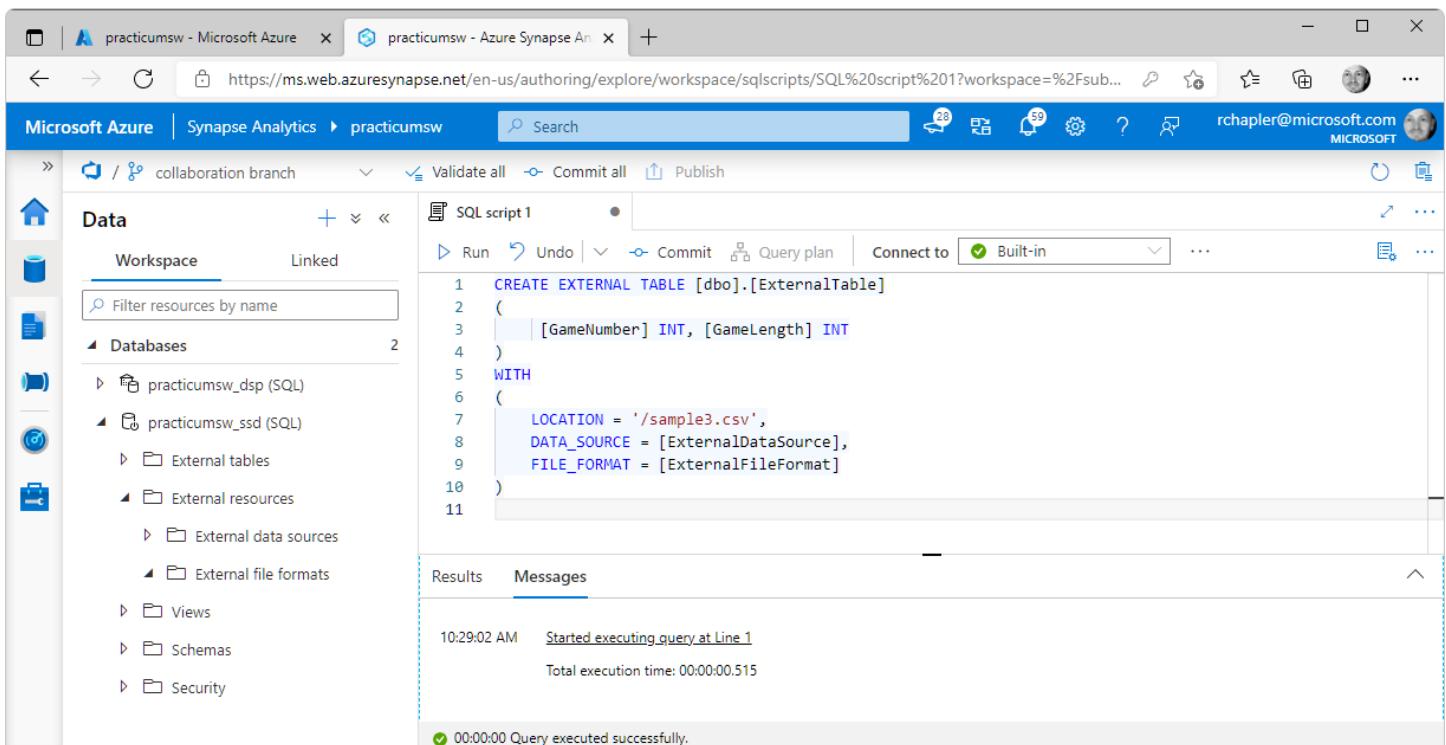
## External Table

Expand navigation to and then right-click on “External tables”.



The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. On the left, the 'Data' blade is open, showing a tree view of resources under 'Workspace'. The 'External tables' folder is currently selected. A context menu is open at this location, with 'New SQL script' highlighted. Other options in the menu include 'New external table', 'Refresh', and a separator line. To the right of the menu, there is a placeholder area with icons for databases and tables, and a message saying 'Select an item'.

Click “New SQL script”, and then “New external table” in the drop-down.



The screenshot shows the Microsoft Azure Synapse Analytics workspace interface with the 'SQL script' editor open. The editor contains a 'CREATE EXTERNAL TABLE' statement:

```
1 CREATE EXTERNAL TABLE [dbo].[ExternalTable]
2 (
3     [GameNumber] INT, [GameLength] INT
4 )
5 WITH
6 (
7     LOCATION = '/sample3.csv',
8     DATA_SOURCE = [ExternalDataSource],
9     FILE_FORMAT = [ExternalFileFormat]
10 )
```

The 'Run' button in the toolbar above the editor is highlighted. Below the editor, the 'Messages' tab shows the execution status:

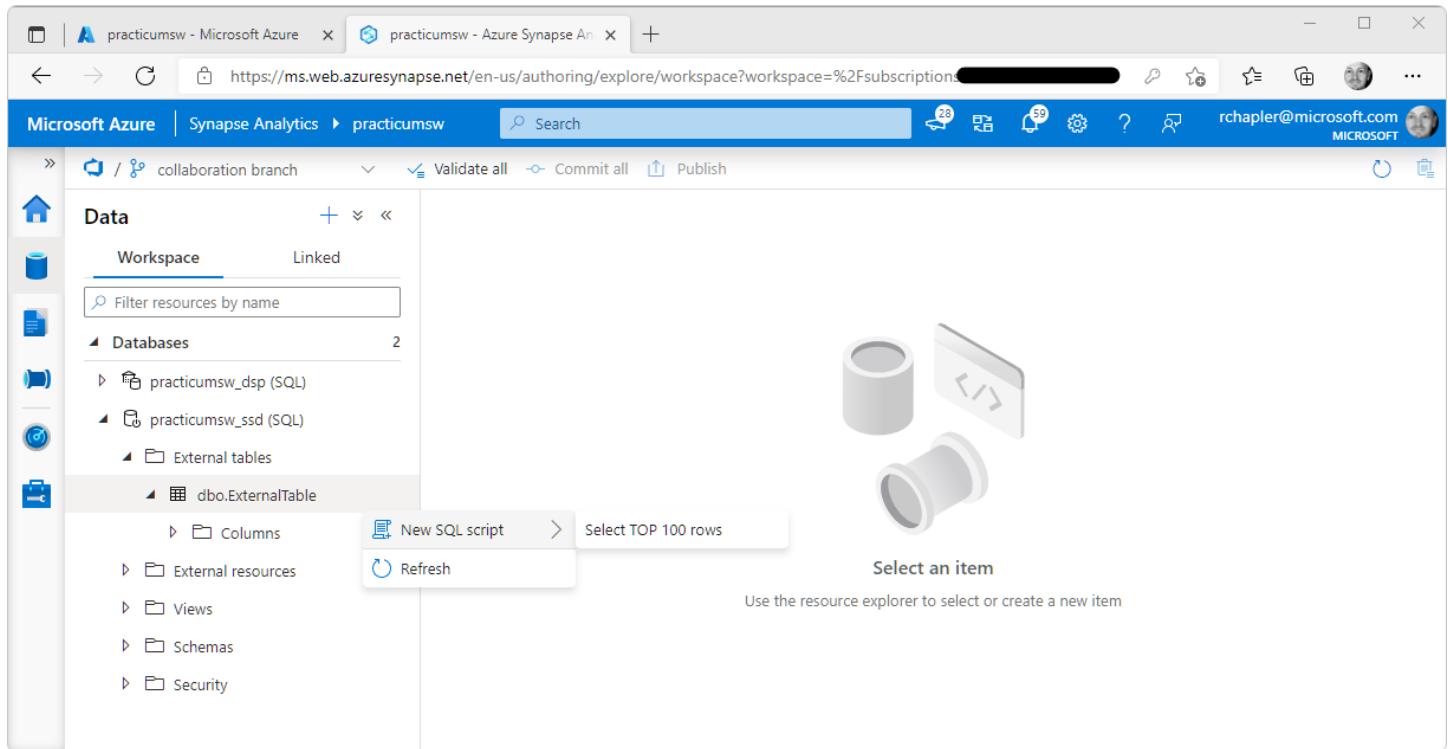
10:29:02 AM Started executing query at Line 1  
Total execution time: 00:00:00.515  
00:00:00 Query executed successfully.

Update the following items in the default code and then Run.

- **Schema Definition** ... replace `[Id] INT` with columns matching the external data source
- **LOCATION** ... replace `'/folder/file'` with values matching those in your container
- **DATA\_SOURCE** ... replace `[DataSource1]` with the name used in [Create External Data Source](#)
- **FILE\_FORMAT** ... replace `[FileFormat1]` with the name used in [Create External File Format](#)

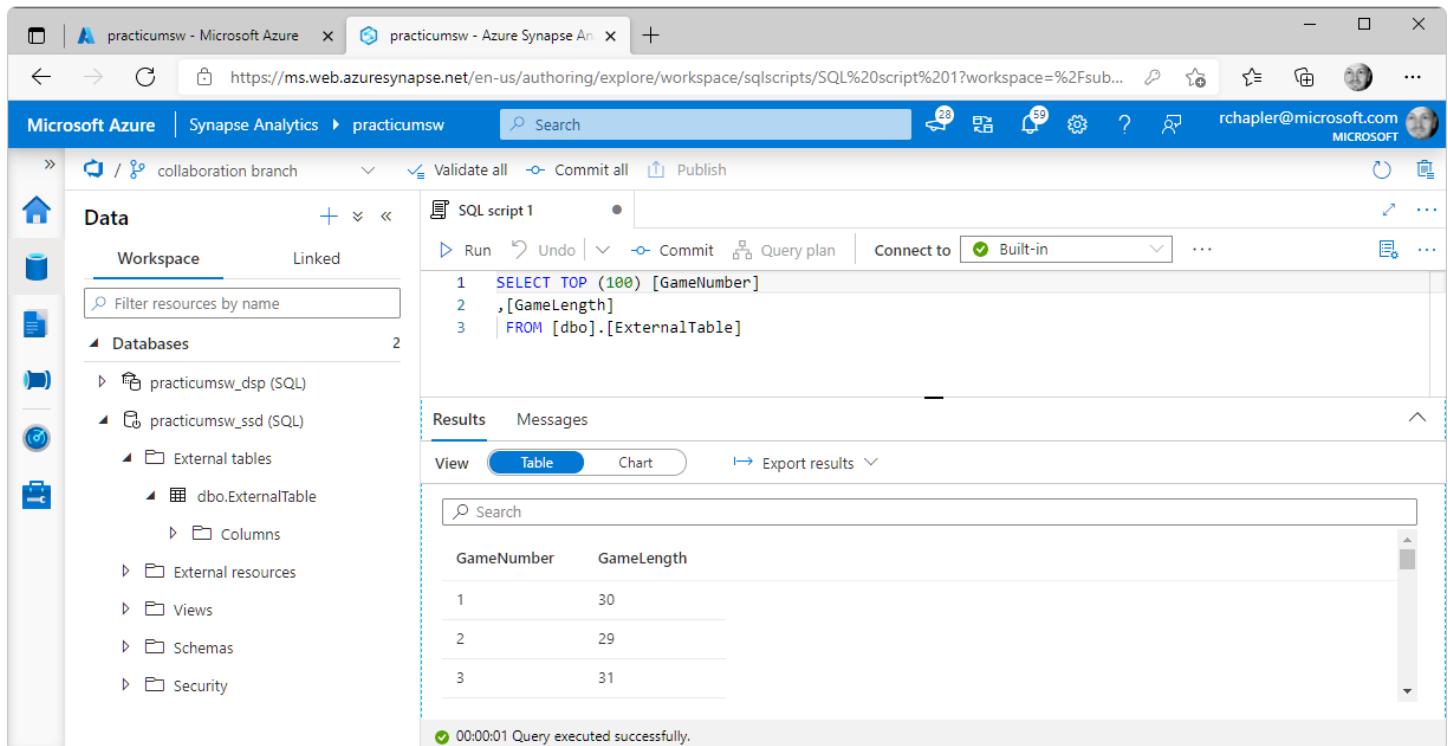
## Confirm Success

Right-click on “External tables” and click Refresh.



The screenshot shows the Microsoft Azure Synapse Analytics Data blade. In the left sidebar, under the 'Data' category, the 'Workspace' tab is selected. Under 'External tables', there is a single item named 'dbo.ExternalTable'. A context menu is open over this item, with the 'Refresh' option highlighted. Other options in the menu include 'New SQL script' and 'Select TOP 100 rows'. To the right of the menu, there is a placeholder area with the text 'Select an item' and a small icon of two cylinders.

Right-click on “dbo.ExternalTable”, click on “New SQL script”, and finally click on “Select TOP 100 rows” in the drop-down menu.



The screenshot shows the Microsoft Azure Synapse Analytics Data blade with a SQL script editor open. The script contains the following T-SQL code:

```
1 SELECT TOP (100) [GameNumber]
2 ,[GameLength]
3 FROM [dbo].[ExternalTable]
```

Below the script, the 'Results' pane is visible, showing the output of the query. The results are presented in a table with two columns: 'GameNumber' and 'GameLength'. The data is as follows:

GameNumber	GameLength
1	30
2	29
3	31

At the bottom of the results pane, a message indicates: '00:00:01 Query executed successfully.'

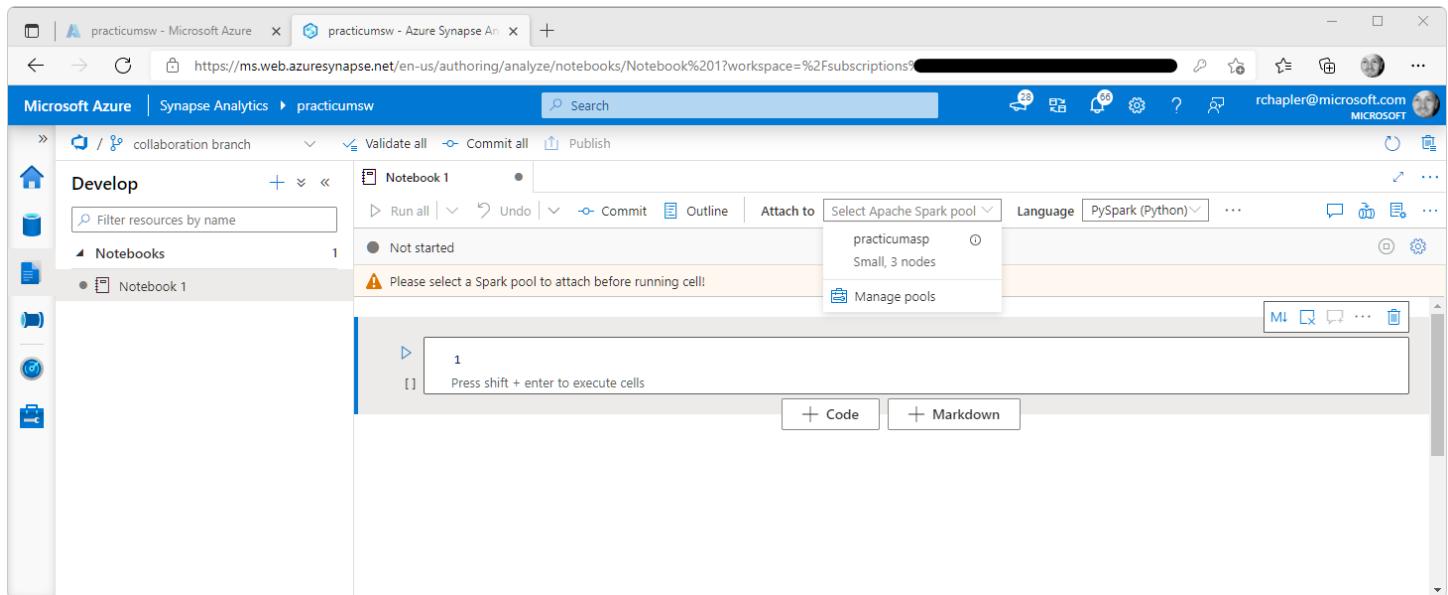
Click Run.

## Method #3: Spark Notebook (Python)

Click the **Develop** tab.

Click the + icon just above and to the right of the “**Filter resources by name**” input.

Select **Notebook** from the resulting drop-down.



Select your Apache Spark Pool from the “**Attach to**” drop-down.

Paste the following code to Cell 1 and then click “**Run Cell**”:

```
%pyspark  
theData = spark.read.load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/sample1.csv', format='csv', header=True)  
theData.show(10)  
  
print('Converted to Pandas...')  
print(theData.toPandas())
```

practicumsw - Microsoft Azure practicumsw - Azure Synapse Analytics

Microsoft Azure | Synapse Analytics > practicumsw

Search

collaboration branch | Validate all | Commit all | Publish

Develop

Notebook 1

Filter resources by name

Notebooks

Notebook 1

Ready

```
1 %%pyspark
2 theData = spark.read.load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/sample1.csv', format='csv', header=True)
3 theData.show(10)
4
5 print('Converted to Pandas...')
6 print(theData.toPandas())
7
```

[1] ✓ 2 min 29 sec - Apache Spark session started in 2 min 6 sec 348 ms. Command executed in 23 sec 297 ms by rchaper on 1:21:45 PM, 9/23/21

> Job execution Succeeded | Spark 2 executors 8 cores

View in monitoring | Open Spark UI

...  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
|Month| "Average" | "2005" | "2006" | "2007" | "2008" | "2009" | "2010" | "2011" | "2012" | "2013" | "2014" | "2015"  
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
| May | 0.1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |  
| Jun | 0.5 | 2 | 1 | 1 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 0 | 1  
| Jul | 0.7 | 5 | 1 | 1 | 2 | 0 | 1 | 3 | 0 | 2 | 2 | 2 | 1  
| Aug | 2.3 | 6 | 3 | 2 | 4 | 4 | 4 | 7 | 8 | 2 | 2 | 2 | 3  
Sep	3.5	6	4	7	4	2	8	5	2	5	2	5
Oct	2.0	8	0	1	3	2	5	1	5	2	3	0
Nov	0.5	3	0	0	1	1	0	1	0	1	0	1
Dec	0.0	1	0	1	0	0	0	0	0	0	0	1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

Converted to Pandas...

	Month	"Average"	"2005"	"2006"	"2007"	"2008"	"2009"	"2010"	"2011"	"2012"	"2013"	"2014"	"2015"
0	May	0.1	0	0	...	2	0	0	0	0	0	0	0
1	Jun	0.5	2	1	...	2	2	0	1	2	2	2	0
2	Jul	0.7	5	1	...	0	2	2	1	2	2	2	1
3	Aug	2.3	6	3	...	8	2	2	3	2	2	2	3
4	Sep	3.5	6	4	...	2	5	2	5	2	5	2	5
5	Oct	2.0	8	0	...	5	2	3	0	1	0	1	0
6	Nov	0.5	3	0	...	0	1	0	1	0	1	0	1
7	Dec	0.0	1	0	...	0	0	0	0	0	0	0	1

[8 rows x 13 columns]

+ Code | + Markdown

## Objective: Synapse + Data Lake | Simple Data Ingestion

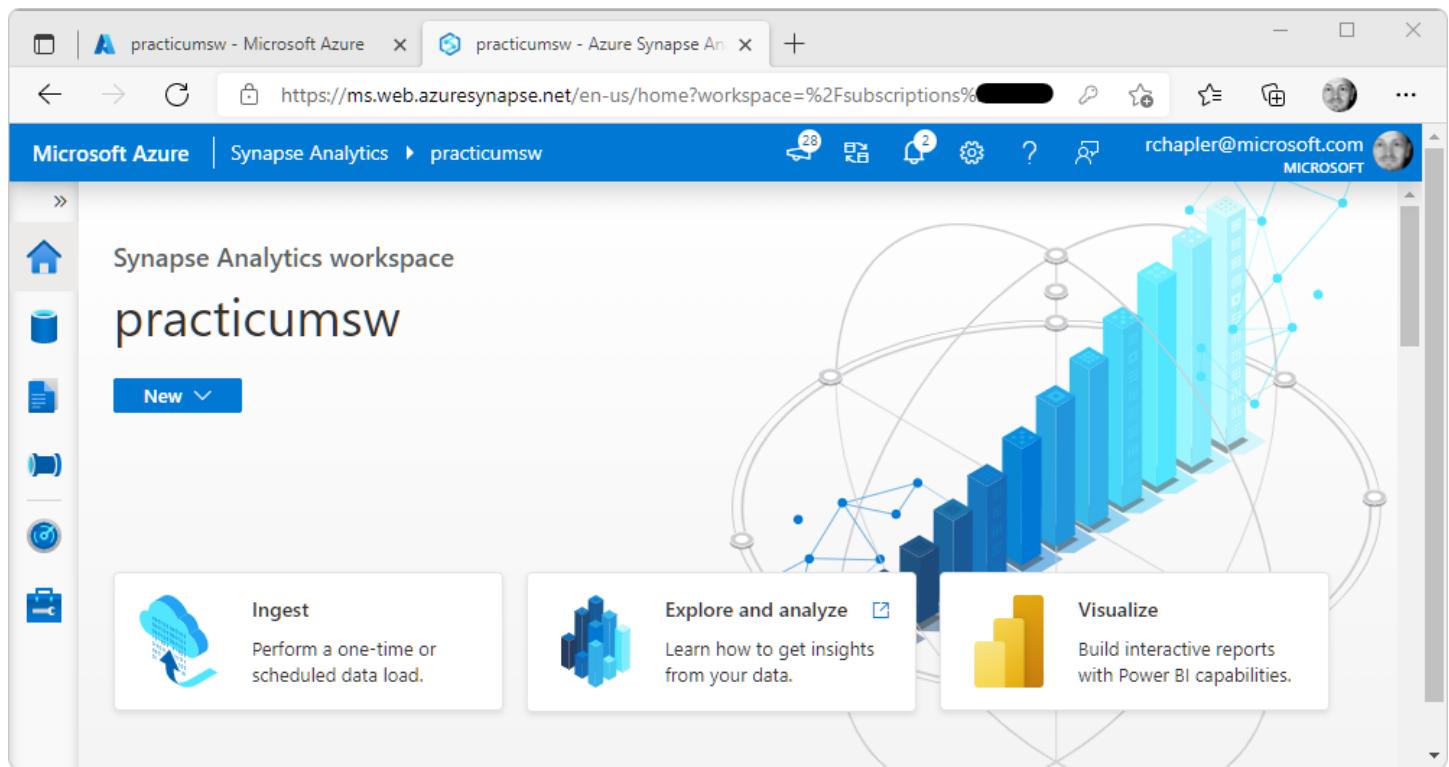
*Note: These instructions also apply (with minor differences) to Azure Data Factory.*

Follow the instructions in this section to **load data from Data Lake to Synapse**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Lake (with container and sample data)
- Synapse (with Dedicated SQL Pool)

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.



Click the **Ingest** button.

## Properties

Screenshot of the Microsoft Azure Copy Data tool wizard. The left sidebar shows steps 1 through 5: Properties, Source, Target, Settings, and Review and finish. Step 1 is selected and highlighted.

**Properties**

Use Copy Data Tool to perform a one-time or scheduled data load from 90+ data sources. Follow the wizard experience to specify your data loading settings, and let the Copy Data Tool generate the artifacts for you, including pipelines, datasets, and linked services. [Learn more](#)

**Properties**

Select copy data task type and configure task schedule

**Task type**

**Built-in copy task**  
You will get single pipeline to copy data from 90+ data source easily.

**Metadata-driven copy task (Preview)**  
Metadata is required to be stored in external control tables to load data at large-scale.

You will get single pipeline to quickly copy objects from data source store to destination in a very intuitive manner.

**Task cadence or task schedule \***

Run once now    Schedule    Tumbling window

[< Previous](#)   [Next >](#)   [Cancel](#)

Confirm default select, “Built-in copy task” and “Run once now”, then click the “**Next >**” button.

## Source > Dataset

The screenshot shows the 'Copy Data tool' interface in the Microsoft Azure portal. The left sidebar lists steps: Properties (selected), Source, Dataset, Configuration, Target, Settings, and Review and finish. The main panel is titled 'Source data store' and contains the following fields:

- Source type:** Azure Data Lake Storage Gen2
- Connection \***: practicumdl (with Edit and New connection buttons)
- Integration runtime \***: AutoResolveIntegrationRuntime (with Edit button)
- File or folder:** practicumdlc/sample1.csv (with Browse button)
- Options:** Binary copy (unchecked), Recursively (checked), Enable partition discovery (unchecked)
- Max concurrent connections:** (empty input field)
- Filter by last modified:** Start time (UTC) and End time (UTC) input fields

At the bottom are 'Previous' and 'Next >' buttons, and a 'Cancel' button.

On the “Source data store” page, enter values for the following items:

**Source Type** Select “Azure Data Lake Storage Gen2”

**Connection** Click the “+ New connection” button

The screenshot shows the Microsoft Azure Synapse Analytics Copy Data tool interface. On the left, a vertical navigation pane lists steps: Properties (selected), Source, Dataset, Configuration, Target, Settings, and Review and finish. The main area is titled "New connection (Azure Data Lake Storage Gen2)". It includes fields for Name (practicumdl), Description, Connect via integration runtime (AutoResolveIntegrationRuntime), Authentication method (Managed Identity), Account selection method (From Azure subscription selected), Azure subscription (rchapler), Storage account name (practicumdl), and Test connection (To linked service selected). At the bottom right, there are "Connection successful" and "Test connection" buttons.

On the “New connection (Azure Data Lake Storage Gen2)” page, enter values for the following items:

Name	Self-Explanatory
Connect via...	Confirm default selection, <b>AutoResolveIntegrationRuntime</b>
Account Selection Method	Confirm default selection, “ <b>From Azure subscription</b> ”
Azure Subscription	Select your subscription
Storage Account Name	Select your instance of Data Lake

Click “**Test connection**” to confirm successful connection and then click the **Create** (or **Commit**) button.

Back on the “Source data store” page, click **Browse** to select container/file {e.g., “practicumdlc/sample1.csv”}.

No additional changes are required.

Click the “**Next >**” button.

## Source > Configuration

The screenshot shows the 'Copy Data tool' configuration interface. On the left, a vertical navigation pane lists five steps: 'Properties' (selected), 'Source', 'Dataset', 'Configuration', 'Target', 'Settings', and 'Review and finish'. The main panel is titled 'File format settings' and contains the following configuration options:

- File format**: Set to 'Text format' with buttons for 'Detect text format' and 'Preview data'.
- Column delimiter**: Set to 'Comma (,)'. There is an 'Edit' link below it.
- Row delimiter**: Set to 'Line feed (\n)'. There is an 'Edit' link below it.
- First row as header**: A checked checkbox.
- Advanced**: A collapsed section containing:
  - Compression type**: Set to 'None'.
  - Additional columns**: A section with a '+ New' button.

At the bottom of the main panel are 'Previous' and 'Next >' buttons, and a 'Cancel' button on the right.

On the “**File format settings**” page, confirm default configuration and the resulting data preview.

Make changes if required {e.g., in this example, I check “**First row as header**” to correct handling of incoming data}.

Click the “**Next >**” button.

## Target > Dataset

The screenshot shows the 'Copy Data tool' interface in the Microsoft Azure portal. The left sidebar lists steps: Properties (checked), Source (checked), Target (selected), Dataset, Configuration, Settings, and Review and finish. The main panel is titled 'Destination data store' with the sub-instruction: 'Specify the destination data store for the copy task. You can use an existing data store connection or specify a new data store.' It includes a 'Target type' dropdown set to 'Azure Synapse Analytics' and a 'Connection \*' dropdown with a 'Select...' option and a '+ New connection' button. Navigation buttons at the bottom include '< Previous', 'Next >', and 'Cancel'.

On the “**Destination data store**” page, enter values for the following items:

**Source Type** Select “**Azure Synapse Analytics**”

**Connection** Click the “**+ New connection**” button

practicumsw - Microsoft Azure x practicumsw - Azure Synapse An x + <https://ms.web.azuresynthesize.net/en-us/authoring/orchestrate/pipeline/Pipeline%201?...> 28 4 ⚡ 🌐 ? 🔍 🗃 🏷️ 🎯 🏹 rchapler@microsoft.com MICROSOFT

Microsoft Azure | Synapse Analytics > practicum Search

Copy Data tool

Properties  
Source  
Target  
Dataset  
Configuration  
Settings  
Review and finish

Destination data store  
Specify the destination data store.  
Target type  
Connection \*

New connection (Azure Synapse Analytics)

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

Name \*

Description

Connect via integration runtime \*  [Edit](#)

Connection string [Azure Key Vault](#)

Account selection method [?](#)  
 From Azure subscription  Enter manually

Azure subscription

Server name \*

Database name \*

SQL pool \*  [Edit](#)

Authentication type \*

User name \*

Add dynamic content [Alt+Shift+D]

Password [Azure Key Vault](#)

AKV linked service \* [?](#)  [Edit](#)

Secret name \*

Secret version [?](#)

Additional connection properties [New](#)

Annotations [New](#)

Parameters [Advanced](#)

Previous Next Commit Cancel [Test connection](#)

On the “New connection (Azure Synapse Analytics)” pop-out, enter values for the following items:

Name	Self-Explanatory
Connect via...	Confirm default selection, <b>AutoResolveIntegrationRuntime</b>
Account Selection Method	Confirm default selection, “From Azure subscription”
Azure Subscription	Select your subscription
Server Name	Select your instance of Synapse Workspace
Database Name	Select your Dedicated SQL Pool
Authentication Type	Confirm default value, “SQL authentication”
User Name	Enter the “SQL Server admin login” value used during instantiation of Synapse
AKV Linked Service	Select the name of the Linked Service created for the Key Vault
Secret Name	Enter the Secret Name used to capture the Synapse Workspace administrator password

Click “Test connection” to confirm successful connection and then click the **Create** (or **Commit**) button.

The screenshot shows the Microsoft Azure Copy Data tool interface. On the left, a vertical navigation pane lists steps: Properties (checked), Source (checked), Target (selected), Dataset, Configuration, Settings, and Review and finish. The main panel is titled "Destination data store". It asks to specify the destination data store for the copy task, using an existing data store connection or specifying a new one. The "Target type" dropdown is set to "Azure Synapse Analytics". The "Connection" dropdown shows "practicumsa". Below it, the "Integration runtime" dropdown is set to "AutoResolveIntegrationRuntime". A table below maps a source table ("Azure Data Lake Storage Gen2 file") to a target table ("dbo.Sample1"). The target table has a note "(auto-create)". A checkbox "Skip column mapping for all tables" is present. At the bottom are "Previous", "Next >" (highlighted in blue), and "Cancel" buttons.

Back on the “Destination data source” page, enter values for target schema name {e.g., “dbo”} and target table name {e.g., “Sample1”}.

Click the “Next >” button.

## Target > Configuration

The screenshot shows the 'Copy Data tool' configuration page in the Microsoft Azure portal. The left sidebar lists steps: Properties, Source, Target, Dataset, Configuration, Settings, and Review and finish. The 'Target' step is selected. The main area shows 'Column mapping' for a 'Table mappings (1)' entry. It details a single mapping from a 'Source' (Azure Data Lake Storage Gen2 file) column named 'Month' to a 'Destination' (Azure Synapse Analytics) column also named 'Month'. Both columns are of type String. Below this, there's an 'Azure Synapse Analytics sink properties' section with a 'Pre-copy script' input field and an 'Advanced' link. At the bottom are navigation buttons: '< Previous', 'Next >', and 'Cancel'.

Review default column mappings; correct if required.

Click the “**Next >**” button.

## Settings

The screenshot shows the 'Copy Data tool' settings page in the Microsoft Azure portal. The left sidebar lists steps: Properties (checked), Source (checked), Target (checked), Settings (checked), and Review and finish (unchecked). The main area is titled 'Settings' and contains the following fields:

- Task name \***: CopyPipeline\_8th
- Task description**: (empty text area)
- Data consistency verification**: (radio button)
- Fault tolerance**: (dropdown menu)
- Enable logging**: (checkbox)
- Enable staging**: (checkbox)
- Advanced** section:
  - Copy method**: Bulk insert (selected radio button)
  - Bulk insert table lock**: No (radio button)
- Data integration unit**: Auto (dropdown menu)
- Degree of copy parallelism**: (dropdown menu)

At the bottom are buttons for < Previous, Next >, and Cancel.

On the **Settings** page, enter values for the following items:

Enable Staging	Unchecked
Copy Method	Bulk insert

Click the “**Next >**” button.

## Summary

Screenshot of the Microsoft Azure Copy Data tool interface showing a pipeline configuration for copying data from Azure Data Lake Storage Gen2 to Azure Synapse Analytics.

The pipeline steps completed so far are:

- Properties
- Source
- Target
- Settings

The current step is:

- Review and finish

**Summary**

You are running pipeline to copy data from Azure Data Lake Storage Gen2 to Azure Synapse Analytics.

 Azure Data Lake Storage Gen2 →  Azure Synapse Analytics

**Properties**

Task name: CopyPipeline\_8th

Task description:

**Source**

Connection name: practicuml  
Dataset name: SourceDataset\_8th  
Column delimiter:  
Row delimiter:  
Escape character:  
Quote char:  
First row as header: true  
File name: sample1.csv

**Target**

Connection name: practicumsa  
Dataset name: DestinationDataset\_8th  
Table name: dbo.Sample1

**Copy settings**

Timeout: 7:00:00:00  
Retry: 0  
Retry interval: 30  
Secure output: false  
Secure input: false

Buttons at the bottom:

- < Previous
- Next >**
- Cancel

Review configuration and then click the “**Next >**” button.

## Deployment

The screenshot shows the Microsoft Azure Copy Data tool interface. On the left, a vertical navigation bar lists steps: Properties, Source, Target, Settings, Review and finish, Review, and Deployment. The first four steps have green checkmarks. The 'Review and finish' step is currently selected. In the center, there's a diagram showing an arrow pointing from 'Azure Data Lake Storage Gen2' to 'Azure Synapse Analytics'. Below the diagram, the text 'Deployment complete' is displayed. To the right of the diagram, a table titled 'Deployment step' shows two rows: 'Creating datasets' (Status: Succeeded) and 'Creating pipelines' (Status: Succeeded). At the bottom of the screen, there are three buttons: 'Finish' (highlighted in blue), 'Edit pipeline', and 'Monitor'.

Monitor progress.

Click the **Finish** button on success.

## Confirm Success

“CopyPipeline...” will be added to the list of **Pipelines** in the Synapse, **Integrate** section.

The screenshot shows the Microsoft Azure Synapse Analytics Integrate section. On the left, a sidebar shows a 'collaboration branch' and a list of 'Pipelines' with one item named 'CopyPipeline\_8th'. The main area displays the 'CopyPipeline\_8th' pipeline details. It shows the 'Activities' list with 'Copy data' selected. Below the activities, there are tabs for 'Parameters', 'Variables', 'Settings', and 'Output'. The 'Output' tab is active, showing a table with one row: 'Copy\_8th' (Type: Copy data, Run start: 2021-09-22T17:38:30.320, Duration: 00:00:10, Status: Succeeded, Integration runtime: DefaultIntegrationRuntime). At the bottom right, there's a link 'View debug run consumption'.

Click **Debug** and confirm successful execution.

Click the **Data** icon in the left-hand navigation.

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar is titled 'Data' and lists options like Workspace, Databases, Tables, Columns, External tables, External resources, Views, Programmability, Schemas, and Security. Under 'Tables', a table named 'dbo.Sample1' is selected. A context menu is open over this table, showing options: New SQL script (highlighted), Select TOP 100 rows, New notebook, CREATE, DROP, New data flow, DROP and CREATE, and Refresh. The main area displays a SQL script window with the following code:

```
1 SELECT TOP (100) [Month]
2 , [ "Average"]
3 , [ "2005"]
```

Below the script, there are tabs for Results and Messages, and a 'Table' tab is selected. The results pane shows a table with data for years 2006 through 2011. The message pane at the bottom right says '00:00:00 Query executed successfully.'

Expand **Databases > {Dedicated SQL Pool} > Tables** and confirm that you can see your new table.

Right-click on your table, select “**New SQL script**”, and then “**Select TOP 100 rows**” in the drop-down menu.

## Objective: Synapse + SQL + Data Explorer | Simple Data Pipeline

Note: These instructions also apply (with minor differences) to **Azure Data Factory**.

Follow the instructions in this section to **load data from Data Explorer to Synapse**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Explorer (with cluster, database, and sample Product table)
- SQL (with sample data)
- Synapse (with linked services)

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

### Linked Services

Select the **Manage** icon on the left-hand navigation and then “**Linked Services**” in the resulting navigation menu.

Confirm linked service instantiation for: 1) Key Vault, 2) SQL and 3) Data Explorer.

The screenshot shows the Microsoft Azure portal interface for a Synapse Analytics workspace named 'practicumsw'. The left sidebar contains a navigation menu with items like 'Analytics pools', 'External connections', 'Integration', 'Security', 'Access control', 'Credentials', 'Managed private endpoints', 'Code libraries', 'Workspace packages', 'Source control', and 'Git configuration'. The 'External connections' section is currently selected. The main content area is titled 'Linked services' and contains a sub-header: 'Linked services are much like connection strings, which define the connection information needed for Azure Synapse Analytics to connect to external resources.' Below this is a 'New' button and a search/filter bar with 'Annotations : Any'. A table lists five instantiated linked services: 'practicumded' (Azure Data Explorer (Kusto)), 'practicumkv' (Azure Key Vault), 'practicumsd' (Azure SQL Database), 'practicumsw-WorkspaceDefaultSqlServer' (Azure Synapse Analytics), and 'practicumsw-WorkspaceDefaultStorage' (Azure Data Lake Storage Gen2). The table has columns for Name, Type, Related, and Annotations.

Name ↑	Type ↑	Related ↑	Annotations ↑
practicumded	Azure Data Explorer (Kusto)	0	
practicumkv	Azure Key Vault	1	
practicumsd	Azure SQL Database	0	
practicumsw-WorkspaceDefaultSqlServer	Azure Synapse Analytics	0	
practicumsw-WorkspaceDefaultStorage	Azure Data Lake Storage Gen2	0	

Note: you are likely to see system-generated linked services {e.g., “practicumsw-WorkspaceDefault...”} in addition to instantiated items.

### Pipeline

Click the **Integrate** icon in the left-hand navigation.

No items to show  
Try creating a new item using the + button above. [Learn more](#)

Select an item  
Use the resource explorer to select or create a new item

Click the + icon just above and to the right of the “**Filter resources by name**” input.

Select **Pipeline** from the resulting drop-down.

Activities

Source dataset \*

Select... New

Add a “**Copy data**” activity.

## Source

On the **Source** tab, click the “**+ New**” button.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline creation interface. On the left, there's a sidebar with icons for Home, Pipelines, Databricks, Data Lake Analytics, Functions, Batch Service, Data Explorer, and General. The main area shows a pipeline named "Pipeline 1" with one step. A search bar at the top right has "azuresql" typed into it. Below the search bar, there are tabs for All, Azure, Database, File, Generic protocol, NoSQL, and Services and apps. Under the "Database" tab, two options are visible: "Azure SQL Database" and "Azure SQL Database Managed Instance". A "Continue" button is at the bottom of the dialog, and a "Cancel" button is on the far right.

On the “**New integration dataset**” pop-out, search for and then select “**Azure SQL Database**”.

Click the **Continue** button.

The screenshot shows the Microsoft Azure Synapse Analytics interface. On the left, there's a sidebar with icons for Home, Pipelines, Data Flow, Copy Data, and others. The main area shows a pipeline named 'Pipeline 1'. A 'Set properties' dialog is open on the right, titled 'Set properties'. It contains the following fields:

- Name:** AzureSqlTable1
- Linked service:** practicumsd
- Connect via integration runtime:** AutoResolveIntegrationRuntime
- Table name:** SalesLT.Product
- Import schema:** From connection/store (radio button selected)

At the bottom of the dialog are buttons for OK, Back, and Cancel.

On the “Set properties” pop-out, enter values for the following items:

<b>Name</b>	Self-Explanatory
<b>Linked Service</b>	Select your Azure SQL linked service
<b>Table Name</b>	Select “SalesLT.Product”
<b>Import Schema</b>	Confirm default selection, “From connection/store”

Leave all other settings with default values.

Click the **OK** button.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor interface. On the left, there's a sidebar with icons for Home, Pipelines, Databricks, Data Lake Analytics, General, HDInsight, Iteration & conditionals, and Machine Learning. The main area shows a pipeline named "Pipeline 1" with one activity: "Copy data1". The activity is set to copy data from an "AzureSqlTable1" source dataset. The "Source" tab is selected, showing options for "Use query" (set to "Table"), "Query timeout (minutes)" (set to 120), and "Isolation level" (set to "None"). The "Sink" tab is also visible. At the bottom, there's a note: "Please preview data to validate the partition settings are correct before you trigger a run or publish the pipeline." Below the activity, there are buttons for "Open", "+ New", "Preview data", and "Learn more".

## Sink

Click on the **Sink** tab and then click the “**+ New**” button.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline creation interface. On the left, there's a sidebar with icons for Home, Pipelines, Databricks, Data Lake Analytics, and General. The main area shows a pipeline named 'Pipeline 1'. A search bar at the top right has 'practicumsw' typed into it. A 'New integration dataset' dialog is open, prompting the user to select a data store. The search bar in this dialog also has 'data explorer' typed into it. Below the search bar, tabs for All, Azure, Database, File, Generic protocol, NoSQL, and Services and apps are visible. Under the 'All' tab, a box highlights 'Azure Data Explorer (Kusto)' with its icon and name. At the bottom of the dialog are 'Continue' and 'Cancel' buttons.

On the “**New integration dataset**” pop-out, search for and then select “**Azure Data Explorer (Kusto)**”.

Click the **Continue** button.

The screenshot shows the 'Set properties' dialog for an integration dataset. The 'Name' field is set to 'AzureDataExplorerTable1'. The 'Linked service' dropdown is set to 'practicumded'. The 'Connect via integration runtime' dropdown is set to 'AutoResolveIntegrationRuntime'. The 'Table' dropdown is set to 'None'. There's an 'Edit' checkbox and an 'Advanced' section. At the bottom are 'OK', 'Back', and 'Cancel' buttons.

On the “**Set properties**” pop-out, enter values for the following items:

<b>Name</b>	Self-Explanatory
<b>Linked Service</b>	Select your Azure Data Explorer linked service
<b>Table Name</b>	Select <b>Product</b>

Leave all other settings with default values.

Click the **OK** button.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor interface. On the left, the navigation pane is visible with sections like 'Integrate', 'Pipelines', and 'Pipeline 1'. The main workspace displays a pipeline named 'Pipeline 1' containing a single activity: 'Copy data'. This activity is highlighted with a red circle. Below the activity, there are tabs for 'General', 'Source', 'Sink', 'Mapping', 'Settings', and 'User properties'. The 'Sink' tab is selected. Under the 'Sink' tab, the 'Sink dataset' dropdown is set to 'AzureDataExplorerTable1'. Other fields include 'Database' (set to 'practicumded') and 'Ingestion mapping name' (empty). There are also buttons for 'Open', 'New', and 'Learn more'.

## Mapping

Click on the **Mapping** tab and then click the “Import schemas” button.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor. On the left, the navigation pane shows 'practicumsw' workspace, 'Integrate' section, and 'Pipelines' list with 'Pipeline 1'. The main area displays 'Pipeline 1' with a single 'Copy data' activity named 'Copy data1'. The 'Mapping' tab is selected, showing type conversion settings for 'ProductNumber' and 'ListPrice' from source to destination. The destination types are 'String' and 'Decimal' respectively. Below the mapping tab, the 'Output' tab shows a table of pipeline run details.

Name	Type	Run start	Duration	Status	Integration runtime
Copy data1	Copy data	2021-10-04T20:59:21.848	00:00:15	Succeeded	DefaultIntegrationRuntime

## Confirm Success

Click **Validate** to confirm that there are no errors, and then click **Debug** to confirm that the pipeline runs successfully.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor. The interface is identical to the previous one, but the 'Output' tab is now selected. It displays a table of pipeline run details, showing a single run that has completed successfully ('Succeeded') with a duration of 00:00:15. The integration runtime used was 'DefaultIntegrationRuntime'.

Name	Type	Run start	Duration	Status	Integration runtime
Copy data1	Copy data	2021-10-04T20:59:21.848	00:00:15	Succeeded	DefaultIntegrationRuntime

## Objective: Synapse | Simple Conditional Pipeline

Note: These instructions also apply (with minor differences) to **Azure Data Factory**.

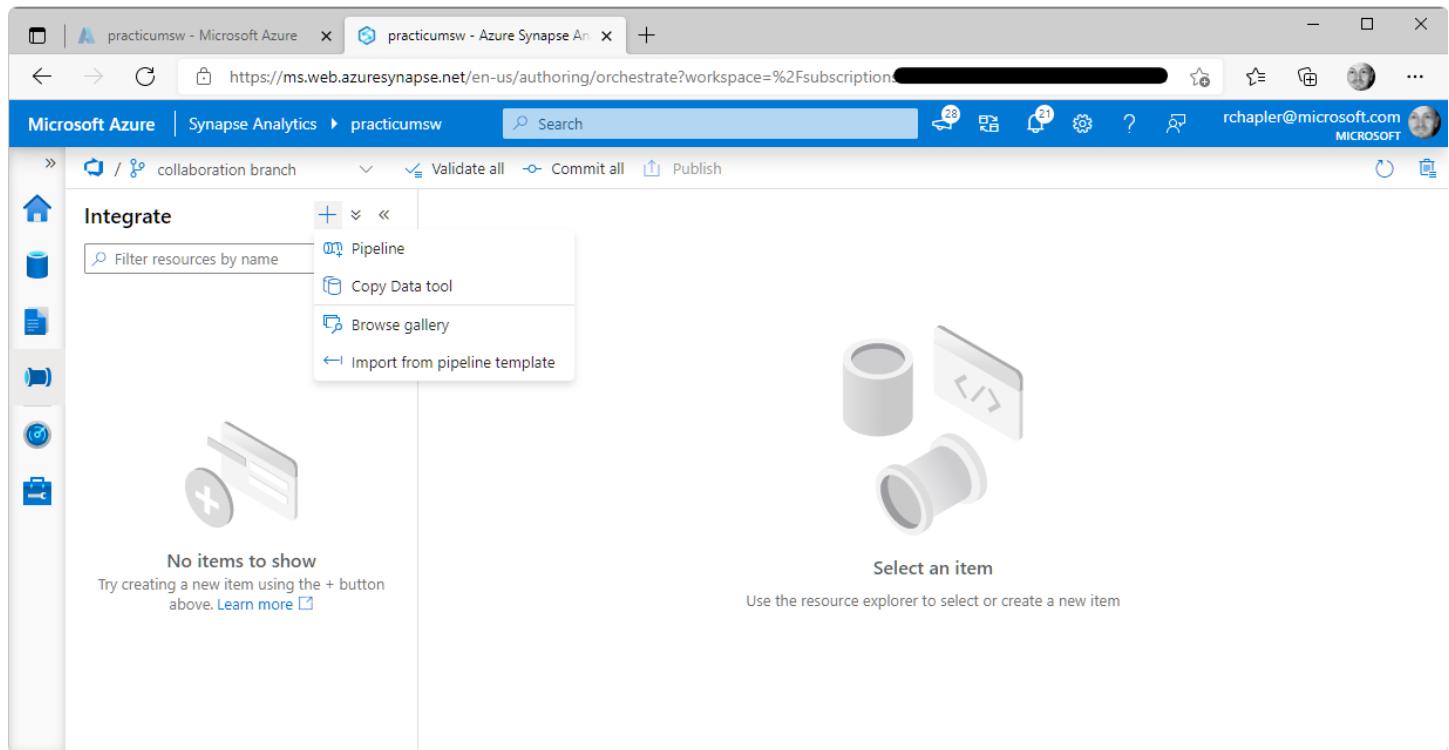
Follow the instructions in this section to **demonstrate conditional logic in a pipeline**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Synapse

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

### Pipeline



Click the **Integrate** icon in the left-hand navigation.

Click the **+** icon just above and to the right of the “**Filter resources by name**” input.

Select **Pipeline** from the resulting drop-down.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor. On the left, the navigation pane shows 'Integrate' selected, with 'Pipelines' expanded and 'Pipeline 1' selected. The main area displays the 'Activities' bar with 'General' expanded, showing options like 'Append variable', 'Delete', 'Execute Pipeline', etc. Below the activities bar, the 'Variables' tab is selected in the 'Parameters' section. It contains two variables: 'Flag' (Boolean type) with a value of 'true', and 'Result' (String type) with a value of 'Value'. There are buttons for '+ New' and 'Delete'.

Click on the **Variables** tab.

Click “+ New” to add each of the following variables:

Flag (Boolean)	Will serve as a TRUE / FALSE trigger for the conditional result
Result (String)	For capturing an anecdotal result {e.g., “Success!”}

## Activity #1, Set Variable

Expand **General** in the **Activities** bar, and then drag-and-drop a “Set variable” component into the activity window.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor. The 'Activities' bar has 'General' expanded, and a 'Set variable' component is selected and highlighted with a red border. This component is being dragged into the pipeline canvas. The pipeline canvas currently contains a single 'Set variable' activity labeled 'Set variable1'. The 'Variables' tab is selected in the properties pane on the right, showing a variable named 'Flag' with a value of 'true'.

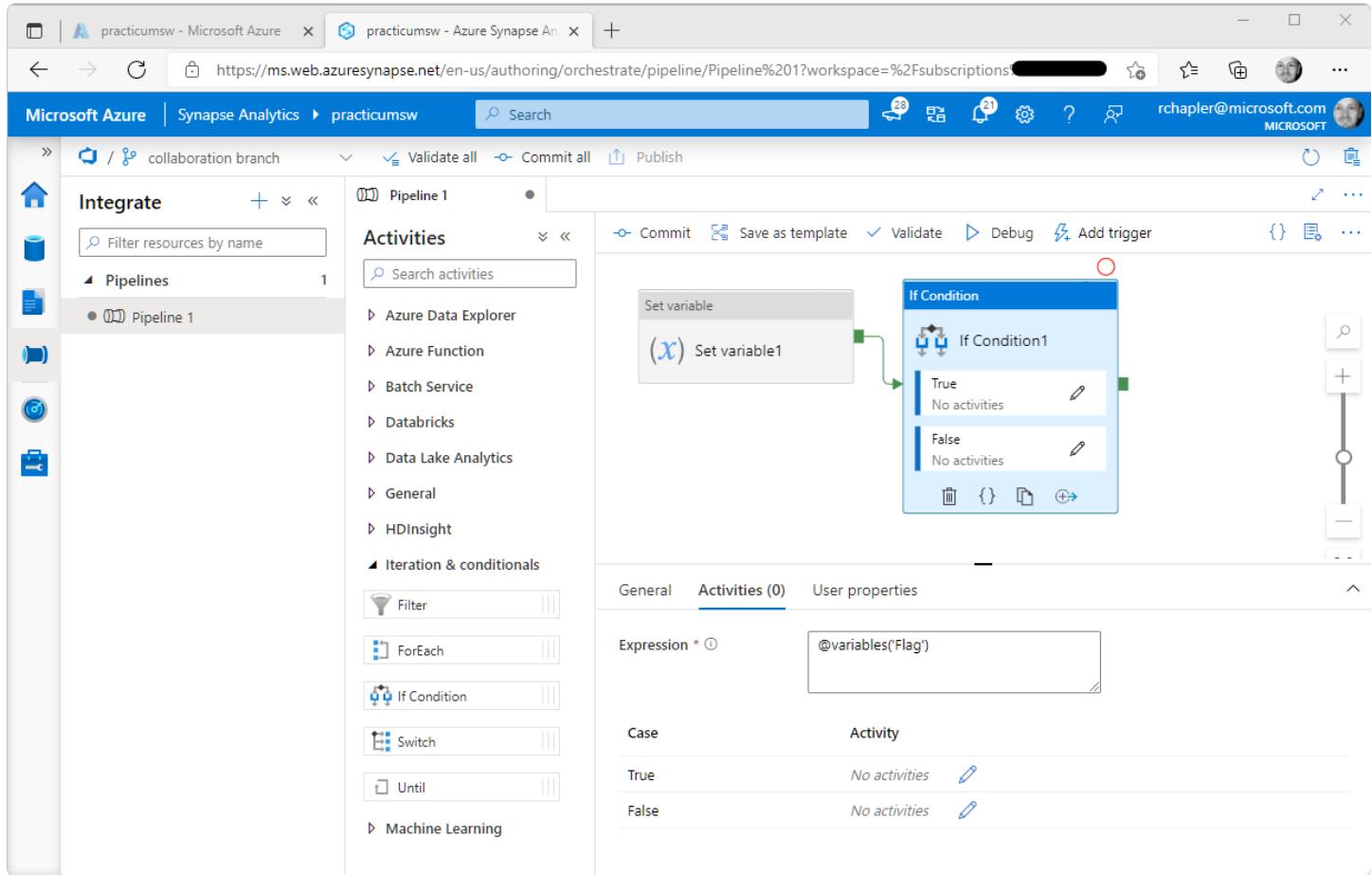
On the **Variables** tab of the “Set Variable...” component, enter values for the following items:

Name	Select <b>Flag</b> from the drop-down
Value	Enter <b>true</b>

## Activity #2, If Condition

Expand “Iteration & conditionals” in the **Activities** bar.

Drag-and-drop an “If Condition” component into the pipeline.



Create a dependency between the “Set variable” and “If Condition” components.

Click on the “Activities (0)” tab.

Click into the **Expression** textbox, and then enter the following: `@variables('Flag')`

Click on the pencil icon in the “If Condition” component, **True** selection.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor. On the left, the navigation pane shows 'Integrate' and 'Pipelines'. Under 'Pipelines', 'Pipeline 1' is selected. The main area displays the pipeline activities. A 'Set variable' activity is currently selected. The 'Variables' tab of the activity configuration pane is open, showing a 'Name' dropdown set to 'Result' and a 'Value' text input set to '"Success!"'. The pipeline structure above shows an 'If Condition' activity followed by a 'True activities' section containing the 'Set variable' activity.

Expand **General** in the **Activities** bar, and then drag-and-drop a “Set variable” component into the activity window.

On the **Variables** tab of the “Set Variable...” component, enter values for the following items:

<b>Name</b>	Select <b>Result</b> from the drop-down
<b>Value</b>	Enter “Success!”

Use cookie crumbs to navigate back to the main window.

Click on the pencil icon in the “If Condition” component, **False** selection.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor. The pipeline structure has been modified. The 'If Condition' activity now has two branches: a 'True activities' section containing the previously configured 'Set variable' activity, and a 'False activities' section. In the 'False activities' section, there is a new 'Set variable' activity with its 'Variables' tab configuration: 'Name' is 'Result' and 'Value' is '@div(1,0)'. The pipeline structure above shows an 'If Condition' activity followed by a 'True activities' section and a 'False activities' section, each containing a 'Set variable' activity.

Expand **General** in the **Activities** bar, and then drag-and-drop a “Set variable” component into the activity window.

On the **Variables** tab of the “Set Variable...” component, enter values for the following items:

Name	Select <b>Result</b> from the drop-down
Value	Enter <code>@div(1,0)</code> ... this will force an error result

Confirm Success

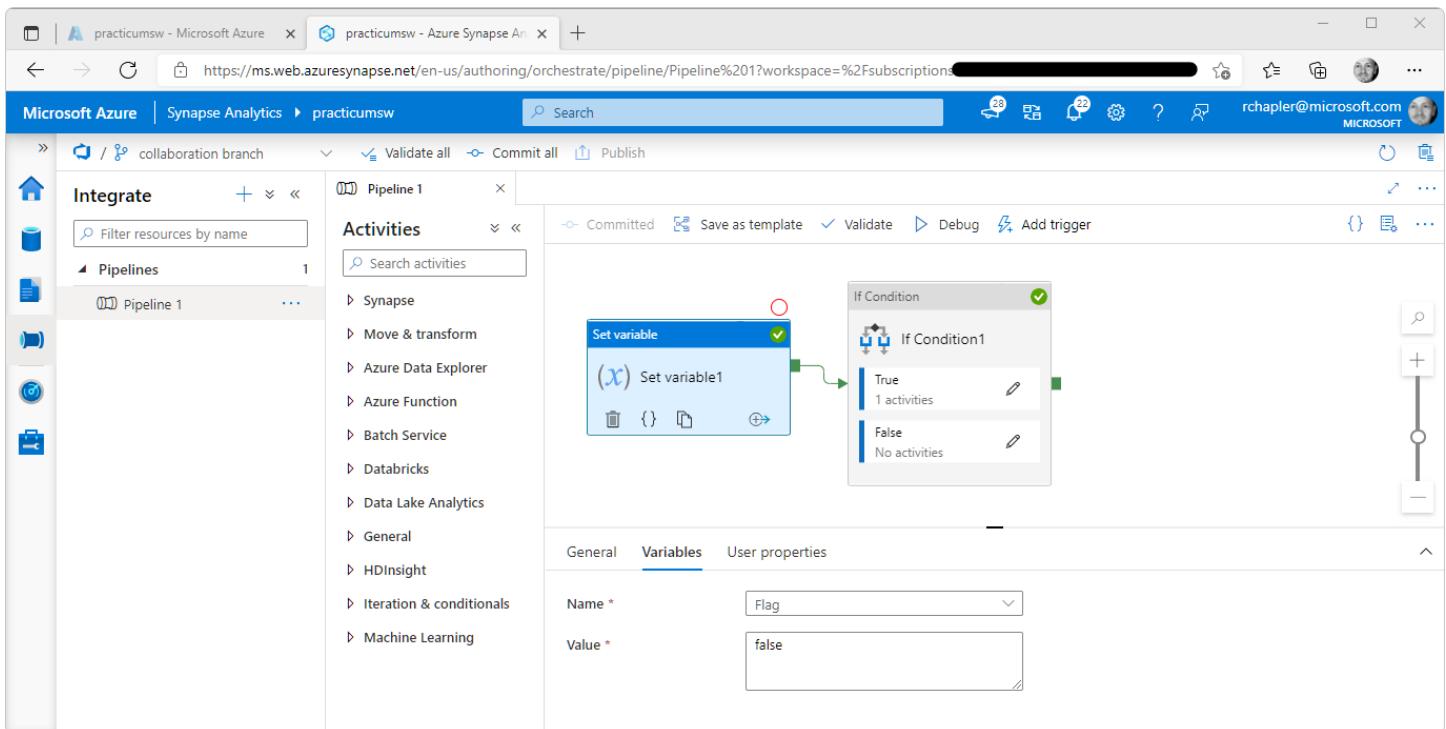
Click **Debug**.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor interface. On the left, the sidebar displays 'Integrate' and 'Pipeline 1'. The main area shows a pipeline diagram with an 'If Condition' activity followed by a 'Set variable' activity. The 'Output' tab at the bottom shows a table of pipeline runs with three entries: 'Set variable2', 'If Condition1', and 'Set variable1', all marked as 'Succeeded'. The URL in the browser is <https://ms.web.azuresynthesize.net/en-us/authoring/orchestrate/pipeline/Pipeline%201?workspace=%2Fsubscriptions%2F...>.

Name	Type	Run start	Duration	Status	Integral
Set variable2	Set variable	2021-09-22T20:13:20.675	00:00:01	Succeeded	
If Condition1	If Condition	2021-09-22T20:13:20.284	00:00:01	Succeeded	
Set variable1	Set variable	2021-09-22T20:13:19.847	00:00:01	Succeeded	

You should see **Succeeded** messages given current variable settings.

Next, we will make a pipeline change that will force pipeline failure.



Navigate to the “**Set variable1**” component, **Variables** tab.

Replace the **true** value with **false**.

Click **Debug**.

You will see **Failed** messages given current variable settings.

# Objective: Synapse (pipeline) + Data Explorer | Incremental Load

Note: These instructions also apply (with minor differences) to Azure Data Factory.

Scenario:

- You want to efficiently load data from an operational data source
- You have a data source that isn't available as a connector in Data Flow {e.g., Oracle}, so you must use a Pipeline
- You want to capture a time series of delete, insert, and update events and provide for "time travel" through data

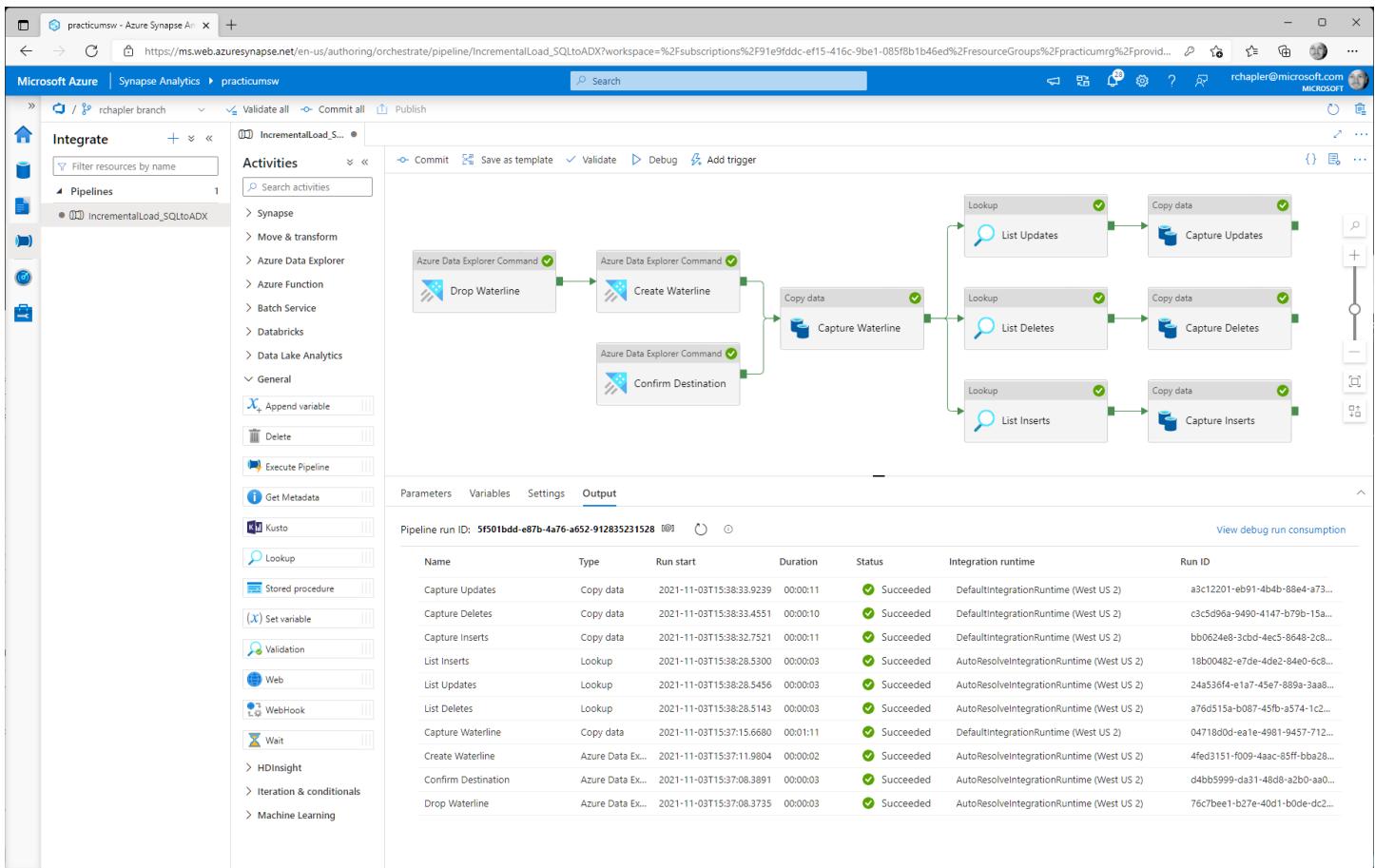
Follow the instructions in this section to **establish incremental load using a Synapse Pipeline to move data from SQL to Data Explorer**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Explorer (with cluster, database, and sample Product table)
- SQL (with sample Northwind database)
- Synapse with linked service **practicumde** (Data Explorer)

## Create Pipeline

When we are finished, our pipeline will look like and function as snipped below.



Navigate to your Synapse Analytics workspace and then click the **Open** link on the “Open Synapse Studio” rectangle.

Click the **Integrate** icon in the left-hand navigation.

Click the **+** icon just above and to the right of the “Filter resources by name” input.

Select “**Pipeline**” from the resulting drop-down.

## Drop Waterline

Execution of this component will drop any previously created waterline table and data.

Expand “**Azure Data Explorer**” in the **Activities** bar, and then drag-and-drop an “**Azure Data Explorer Command**” component into the activity window.

On the **Connection** tab of the new component, enter values for the following items:

...Linked Service	Select the Linked Service for Data Explorer from the drop-down
-------------------	--

On the **Command** tab of the new component, paste the following KQL in the Command textbox.

```
.drop table Product_Waterline ifexists;
```

Click Debug and monitor result to confirm successful progress.

## Create Waterline

Execution of this component will create a new (and empty) waterline table.

Expand “**Azure Data Explorer**” in the **Activities** bar, and then drag-and-drop an “**Azure Data Explorer Command**” component into the activity window.

Create a dependency from the “**Drop Waterline**” component to the new component.

On the **Connection** tab of the new component, enter values for the following items:

...Linked Service	Select the Linked Service for Data Explorer from the drop-down
-------------------	--

On the **Command** tab of the new component, paste the following KQL in the Command textbox.

```
.create table Product_Waterline ( ProductId:int, ModifiedDate:datetime );
```

Click Debug and monitor result to confirm successful progress.

## Confirm Destination

Execution of this component will confirm that the final data destination exists.

Expand “**Azure Data Explorer**” in the **Activities** bar, and then drag-and-drop an “**Azure Data Explorer Command**” component into the activity window.

On the **Connection** tab of the new component, enter values for the following items:

...Linked Service	Select the Linked Service for Data Explorer from the drop-down
-------------------	--

On the **Command** tab of the new component, paste the following KQL in the Command textbox.

```
.create-merge table Product ( ProductId:int, ProductNumber: string, Name:string, ListPrice: decimal, ModifiedDate:datetime, wasDeleted:bool );
```

*Note: The data destination includes column **wasDeleted**, which will be used to provide for capture of delete events as “soft deletes”.*

Click Debug and monitor result to confirm successful progress.

## Capture Waterline

Execution of this component will copy all current identifier and waterline fields from the source.

We will localize a waterline dataset to Data Explorer to provide for identification of source records that have been deleted, inserted, and updated.

This is necessary in Pipelines because unlike Data Flow, there is no in-flow use of data.

Expand “**Move & Transform**” in the **Activities** bar, and then drag-and-drop a “**Copy data**” component into the activity window.

Create one dependency from the “**Create Waterline**” component to the new component and a second dependency from the “**Confirm Destination**” component to the new component.

On the **Source** tab of the new component, enter values for the following items:

<b>Source Dataset</b>	Create a new (or use an existing) dataset using the SQL linked service and <b>SalesLT.Product</b> table
<b>Use Query</b>	Select the <b>Query</b> radio button
<b>Query</b>	Paste the following T-SQL:

```
SELECT ProductId,MAX(ModifiedDate) ModifiedDate FROM SalesLT.Product GROUP BY ProductId
```

No additional changes are required on this tab.

On the **Sink** tab of the new component, enter values for the following items:

<b>Sink Dataset</b>	Create a dataset using the Linked Service for Data Explorer and pointing at the <b>Product_Waterline</b> table
<b>Database and Table</b>	Confirm values

No additional changes are required on this tab.

On the **Mapping** tab of the new component, click the “**Import schemas**” button.

Click Debug and monitor result to confirm successful progress.

## List Deletes

Execution of this component will lookup items that exist in captured data but not in waterline {i.e., deleted rows}.

Expand **General** in the **Activities** bar, and then drag-and-drop a **Lookup** component into the activity window.

Create a dependency from the “**Capture Waterline**” component to the new component.

On the **Settings** tab of the new component, enter values for the following items:

<b>Source Dataset</b>	Create a new (or use an existing) dataset using the Data Explorer linked service and <b>Product</b> table
<b>Query</b>	Paste the following KQL:
<pre>Product   join ( Product   summarize ModifiedDate=max(ModifiedDate) by ProductId ) on ProductId, ModifiedDate   join kind=fullouter Product_Waterline on ProductId   project ProductId = iif(isnull(ProductId),ProductId2,ProductId), ModifiedDate_Waterline = ModifiedDate2, wasDeleted   where isnull(ModifiedDate_Waterline) and not(wasDeleted)   distinct ProductId</pre>	

---

```
| summarize  
Ids=replace_string(replace_string(tostring(make_list(ProductId)), '[',''), ']', ''))
```

Logic explained...

- The first **join** ensures that we are comparing to the most current version of the Sink data
- The second **join** is full outer to ensure that all columns from source and sink are included
- The **project** clause is used to make the result of the joins more understandable
- The first **where** clause filters to sink records that no longer exist at the source
- The second **where** clause ensures that this item hasn't already been flagged wasDeleted (and prevents duplicate capture)
- The **distinct** clause eliminates any duplicates that might flow through
- **make\_list** prepares a concatenated string of values that can be parameterized for later use

---

No additional changes are required on this tab.

Click Debug and monitor result to confirm successful progress.

## Capture Deletes

Execution of this component will capture information about deleted rows.

Expand “**Move & Transform**” in the **Activities** bar, and then drag-and-drop a “**Copy data**” component into the activity window. Create a dependency from the “**List Deletes**” component to the new component.

On the **Source** tab of the new component, enter values for the following items:

<b>Source Dataset</b>	Create a new (or use an existing) dataset using the Data Explorer linked service and <b>Product</b> table
<b>Query</b>	Paste the following parameterized KQL:

```
@concat(  
'Product  
| join ( Product | summarize ModifiedDate=max(ModifiedDate) by ProductId )  
      on ProductId, ModifiedDate  
| where ProductId in (',  
    if(  
      empty(activity('List Deletes').output.firstrow.Ids),  
      '0',  
      activity('List Deletes').output.firstrow.Ids  
    ),  
    ')  
| project ProductId, ProductNumber, Name, ListPrice, ModifiedDate, wasDeleted=true'
```

Logic explained...

- Last-known values are pulled from the sink {i.e., Data Explorer} because the data is unavailable from the source
- **@concat** is used to concatenate the various used in composition of the query string
- The **where ... in** clause provides for use of the parameterized value produced in the “**List Deletes**” component
- The ...**Ids** parameterization is qualified with **if(empty...)** to prevent pipeline error when no values are passed from the “**List Deletes**” component
- The **project** clause includes the **soft-delete flag** and is set to **true**

---

No additional changes are required on this tab.

On the **Sink** tab of the new component, enter values for the following items:

---

<b>Sink Dataset</b>	Create a dataset using the Linked Service for Data Explorer and pointing at the <b>Product</b> table
<b>Database and Table</b>	Confirm values

---

No additional changes are required on this tab.

On the **Mapping** tab of the new component, click the “Import schemas” button and pass 0 when prompted for a parameter value.

Click Debug and monitor result to confirm successful progress.

## List Inserts

Execution of this component will lookup items that exist in the waterline but not in captured data {i.e., inserted rows}.

Expand **General** in the **Activities** bar, and then drag-and-drop a **Lookup** component into the activity window.

Create a dependency from the “**Capture Waterline**” component to the new component.

On the **Settings** tab of the new component, enter values for the following items:

---

<b>Source Dataset</b>	Create a new (or use an existing) dataset using the Data Explorer linked service and <b>Product</b> table
<b>Query</b>	Paste the following KQL:

---

```
Product
| join ( Product | summarize ModifiedDate=max(ModifiedDate) by ProductId ) on ProductId,
ModifiedDate
| join kind=fullouter Product_Waterline on ProductId
| project ProductId = iif(isnull(ProductId),ProductId2,ProductId), ModifiedDate =
ModifiedDate1
| where isnull(ModifiedDate)
| distinct ProductId
| summarize
Ids=replace_string(replace_string(tostring(make_list(ProductId)),',''),']',''))
```

Logic explained...

- The first **join** ensures that we are comparing to the most current version of the Sink data
- The second **join** is full outer to ensure that all columns from source and sink are included
- The **project** clause is used to make the result of the joins more understandable
- The **where** clause filters to source records that do not exist at the sink
- The **distinct** clause eliminates any duplicates that might flow through
- **make\_list** prepares a concatenated string of values that can be parameterized for later use

---

No additional changes are required on this tab.

Click Debug and monitor result to confirm successful progress.

## Capture Inserts

Execution of this component will capture information about inserted rows.

Expand “**Move & Transform**” in the **Activities** bar, and then drag-and-drop a “**Copy data**” component into the activity window.

Create a dependency from the “**List Inserts**” component to the new component.

On the **Source** tab of the new component, enter values for the following items:

<b>Source Dataset</b>	Create a new (or use an existing) dataset using the SQL linked service and <b>SalesLT.Product</b> table
<b>Query</b>	Paste the following parameterized KQL:  @concat( 'SELECT *, cast(0 as bit) [wasDeleted] FROM SalesLT.Product WHERE ProductId IN (', if( empty(activity('List Inserts').output.firstrow.Ids), '0', activity('List Inserts').output.firstrow.Ids ), ')');')  Logic explained... <ul style="list-style-type: none"><li>• <b>@concat</b> is used to concatenate the various used in composition of the query string</li><li>• <b>cast(0 as bit)</b> forces a false value into the soft-delete flag (this logic will vary by source)</li><li>• The <b>where ... in</b> clause provides for use of the parameterized value produced in the “<b>List Inserts</b>” component</li><li>• The ...<b>Ids</b> parameterization is qualified with <b>if(empty...)</b> to prevent pipeline error when no values are passed from the “<b>List Inserts</b>” component</li></ul>

No additional changes are required on this tab.

On the **Sink** tab of the new component, enter values for the following items:

<b>Sink Dataset</b>	Create a new (or use an existing) dataset using the Data Explorer linked service and <b>Product</b> table
<b>Database and Table</b>	Confirm values

No additional changes are required on this tab.

On the **Mapping** tab of the new component, click the “**Import schemas**” button and pass 0 when prompted for a parameter value.

Click Debug and monitor result to confirm successful progress.

## List Updates

Execution of this component will lookup items that have a higher waterline at the source than the sink {i.e., updated rows}.

Expand **General** in the **Activities** bar, and then drag-and-drop a **Lookup** component into the activity window.

Create a dependency from the “**Capture Waterline**” component to the new component.

On the **Settings** tab of the new component, enter values for the following items:

<b>Source Dataset</b>	Create a new (or use an existing) dataset using the Data Explorer linked service and <b>Product</b> table
<b>Query</b>	Paste the following KQL:  Product   join ( Product   summarize ModifiedDate=max(ModifiedDate) by ProductId ) on ProductId, ModifiedDate   join kind=fullouter Product_Waterline on ProductId   project ProductId = iif(isnull(ProductId),ProductId2,ProductId), ModifiedDate = ModifiedDate1, ModifiedDate_Waterline = ModifiedDate2   where ModifiedDate < ModifiedDate_Waterline   distinct ProductId

---

```
| summarize  
Ids=replace_string(replace_string(tostring(make_list(ProductId)), '[',''), ']',''))
```

Logic explained...

- The first **join** ensures that we are comparing to the most current version of the Sink data
- The second **join** is full outer to ensure that all columns from source and sink are included
- The **project** clause is used to make the result of the joins more understandable
- The **where** clause filters to source records with a higher waterline captured sink data
- The **distinct** clause eliminates any duplicates that might flow through
- **make\_list** prepares a concatenated string of values that can be parameterized for later use

---

No additional changes are required on this tab.

Click Debug and monitor result to confirm successful progress.

## Capture Updates

Execution of this component will capture information about updated rows.

Expand “**Move & Transform**” in the **Activities** bar, and then drag-and-drop a “**Copy data**” component into the activity window. Create a dependency from the “**List Updates**” component to the new component.

On the **Source** tab of the new component, enter values for the following items:

<b>Source Dataset</b>	Create a new (or use an existing) dataset using the Data Explorer linked service and <b>Product</b> table
<b>Query</b>	Paste the following parameterized KQL:  <pre>@concat(     'SELECT *, cast(0 as bit) [wasDeleted] FROM SalesLT.Product WHERE ProductId IN (',     if(         empty(activity('List Updates').output.firstrow.Ids),         '0',         activity('List Updates').output.firstrow.Ids     ),     ');')</pre>

Logic explained...

- **@concat** is used to concatenate the various used in composition of the query string
- **cast(0 as bit)** forces a false value into the soft-delete flag (this logic will vary by source)
- The **where ... in** clause provides for use of the parameterized value produced in the “**List Updates**” component
- The ...**Ids** parameterization is qualified with **if(empty...)** to prevent pipeline error when no values are passed from the “**List Updates**” component

---

No additional changes are required on this tab.

On the **Sink** tab of the new component, enter values for the following items:

<b>Sink Dataset</b>	Create a new (or use an existing) dataset using the Data Explorer linked service and <b>Product</b> table
<b>Database and Table</b>	Confirm values

No additional changes are required on this tab.

On the **Mapping** tab of the new component, click the “**Import schemas**” button and pass 0 when prompted for a parameter value.

Click Debug and monitor result to confirm successful progress.

## Confirm Success

The first and most obvious confirmation is that your completed pipeline runs successfully.

Then, confirm that data was captured.

### Confirmation #1, Record Count

Navigate to your instance of Data Explorer, and then click **Query** in the left-hand navigation.

Execute the following KQL query:

```
Product | summarize count()
```

### Confirmation #2, Delete Action

Navigate to your instance of SQL, click “**Query Editor**” in the left-hand navigation, and then login / whitelist (if required).

Execute the following T-SQL query to delete a record:

```
DELETE FROM [SalesLT].[Product] WHERE [ProductID]=(SELECT MIN([ProductID]) FROM [SalesLT].[Product])
```

Navigate to your “**Incremental Load**” pipeline in Synapse.

Click Debug and monitor result to confirm successful progress.

Navigate to your instance of Data Explorer, and then click **Query** in the left-hand navigation.

Execute the following KQL query: `Product | where wasDeleted == true`

Confirm that the deleted record is included in your query results.

Execute the following KQL query: `Product | where ProductId == {deleted ProductId value}`

Confirm that you can see the full history for the record that was deleted.

	ProductId	ProductNumber	Name	ListPrice	ModifiedDate	wasDeleted
>	680	FR-R92B-58	HL Road Frame - Black,...	999.99	2021-11-03 16:22:09.7170	true
>	680	FR-R92B-58	HL Road Frame - Black,...	999.99	2021-11-03 16:22:09.7170	false

### Confirmation #3, Insert Action

Navigate to your instance of SQL, click “**Query Editor**” in the left-hand navigation, and then login / whitelist (if required).

Execute the following T-SQL query to insert a record:

```
INSERT INTO [SalesLT].[Product]
([Name],[ProductNumber],[Color],[StandardCost],[ListPrice],[SellStartDate],[rowguid],[ModifiedDate])
VALUES ('New Product ABC','ABC-123','Bronze',1000,1500,getdate(),newid(),getdate())
```

Navigate to your “**Incremental Load**” pipeline in Synapse.

Click Debug and monitor result to confirm successful progress.

Navigate to your instance of Data Explorer, and then click **Query** in the left-hand navigation.

Execute the following KQL query: `Product | where Name contains 'New Product'`

Confirm that the inserted record is included in your query results.

ProductID	ProductNumber	Name	ListPrice	ModifiedDate	wasDeleted
> 1,015	ABC-123	New Product ABC	1,500	2021-11-03 20:27:01.7530	false

#### Confirmation #4, Update Action

Navigate to your instance of SQL, click “**Query Editor**” in the left-hand navigation, and then login / whitelist (if required).

Execute the following T-SQL query to insert a record:

```
UPDATE SalesLT.Product SET ModifiedDate=GETDATE() WHERE ProductID=(SELECT MAX([ProductID]) FROM [SalesLT].[Product])
```

Navigate to your “**Incremental Load**” pipeline in Synapse.

Click Debug and monitor result to confirm successful progress.

Navigate to your instance of Data Explorer, and then click **Query** in the left-hand navigation.

Execute the following KQL query: `Product | where Name contains 'New Product'`

Confirm that the updated record is included in your query results.

ProductID	ProductNumber	Name	ListPrice	ModifiedDate	wasDeleted
> 1,015	ABC-123	New Product ABC	1,500	2021-11-03 20:27:01.7530	false
> 1,015	ABC-123	New Product ABC	1,500	2021-11-03 20:38:09.5870	false

#### Create Trigger

Establish a recurring schedule for your completed pipeline.

Navigate to your “**Incremental Load**” pipeline in Synapse.

Click the “**Add trigger**” button and click “**New/Edit**” in the resulting drop-down.

On the resulting pop-out, click the “**Choose trigger...**” drop-down and select “**+ New**”.

On the “**New trigger**” pop-out, enter values for the following items:

Name	Self-explanatory
Type	Select “ <b>Schedule</b> ”
Start Date	Confirm default value (consider using round hour / minute to avoid odd timing)
Time Zone	Confirm default value
Recurrence	Choose appropriate recurrence

No additional changes are required.

Click the **Commit** button.

## Objective: Synapse (data flow) + Delta Lake | Incremental Load

*Note: These instructions also apply (with minor differences) to Azure Data Factory.*

Follow the instructions in this section to **demonstrate incremental load using Synapse Data Flows and Delta Lake**.

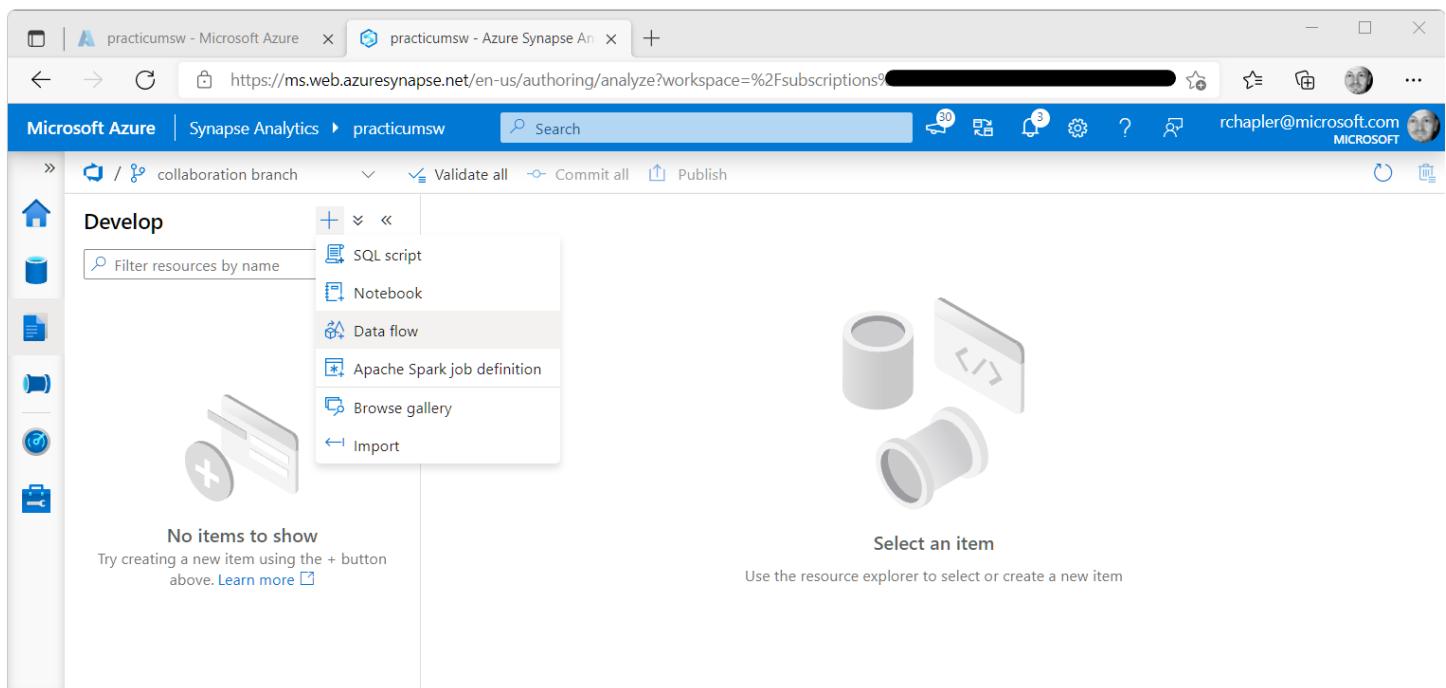
To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Lake
- SQL (with sample database)
- Synapse (with linked service for SQL and Apache Spark Pool)

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

### Initial Load

This section describes how to load existing source data into Delta Lake (as a primer for Incremental Load).



### Create Data Flow

Click the **Develop** icon in the left-hand navigation.

Click the **+** icon just above and to the right of the “**Filter resources by name**” input.

Select “**Data flow**” from the resulting drop-down.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. At the top, there are two tabs: 'practicumsw - Microsoft Azure' and 'practicumsw - Azure Synapse Analytics'. The URL in the address bar is <https://ms.web.azuresynapse.net/en-us/authoring/analyze/dataflow/Dataflow1?workspace=%2Fsubscriptions%2F...>. The main area displays a data flow named 'Initial Load'. The data flow consists of a single step labeled 'Add Source' enclosed in a dashed rectangle. To the right of the data flow, there is a 'Properties' panel. The 'General' tab is selected, showing the 'Name' field set to 'Initial Load'. Below the properties panel, there are tabs for 'Parameters' and 'Settings', with 'Parameters' currently selected.

On the **Properties** pop-out, enter values for the following items:

Name	Self-Explanatory
------	------------------

Consider activating "**Data flow debug**" to avoid wait time later in the process.

Source (sample database)

Click on "**Add Source**" in the dashed rectangle.

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow blade. At the top, there are two tabs: 'practicumsw - Microsoft Azure' and 'practicumsw - Azure Synapse Analytics'. The URL in the address bar is <https://ms.web.azuresynapse.net/en-us/authoring/analyze/dataflow/Dataflow1?workspace=%2Fsubscriptions%2F...>. The main area displays a pipeline named 'Initial Load' under the 'Develop' branch. The pipeline consists of one step: 'practicumsd' (Integration dataset). Below the pipeline, the 'Source settings' tab is selected, showing configuration for the output stream name 'practicumsd', source type 'Inline', inline dataset type 'Azure SQL Database', linked service 'practicumsd' (with a successful test connection), and sampling settings.

On the “Source settings” tab, enter values for the following items:

Output Stream Name	Self-Explanatory
Source Type	Select <b>Inline</b>
Inline Dataset Type	Select “ <b>Azure SQL Database</b> ”
Linked Service	Select <b>practicumsd</b>
Sampling	Confirm default setting, <b>Disable</b>

Click “Test connection” to confirm successful configuration.

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow blade. In the center, there is a data flow named "Initial Load". Below it, the "Source options" tab is selected, showing the following configuration:

- Input:** Table (radio button selected)
- Schema name \***: SalesLT
- Table name \***: Product
- Isolation level**: Read uncommitted

At the top right of the data flow editor, there is a "Data flow debug" toggle switch which is turned on.

On the “Source options” tab, enter values for the following items:

<b>Input</b>	Confirm default setting, <b>Table</b>
<b>Schema Name</b>	Enter <b>SalesLT</b>
<b>Table Name</b>	Enter <b>Product</b>

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow blade. The "Projection" tab is selected. At the top, it says "Import schema" (which is highlighted in blue) and "Clear schema". Below that is a table titled "Column name" and "Type". The data in the table is:

Column name	Type
ProductID	integer
Name	string
ProductNumber	string

On the **Projection** tab, click “**Import schema**” and then **Import** on the resulting pop-out.

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow interface. In the center, there is a data preview pane for a flow named 'Initial Load'. The preview shows 295 rows of data from a source named 'practicumsd'. The columns listed are ProductID, Name, ProductNumber, and Color. The data includes entries for various products like HL Road Frame - Black, Sport-100 Helmet, and Mountain Bike Saddle. At the top of the preview pane, there are tabs for Source settings, Source options, Projection, Optimize, Inspect, and Data preview. The 'Data preview' tab is selected. Below the tabs, there are buttons for Refresh, Typecast, Modify, Map drifted, Statistics, and Remove. The 'Refresh' button is highlighted.

Navigate to the “**Data preview**” tab and then click **Refresh**.

Click the **Commit** button.

## Sink (delta lake)

This screenshot is similar to the previous one but shows a 'sink' destination selected in the 'Destination' dropdown. The rest of the interface and data preview are identical to the first screenshot, showing the same 295 rows of product data from the source 'practicumsd'.

Click the + in the bottom right of the Data Lake source, and then search for and select **Sink** from the resulting pop-up list.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. At the top, there are two tabs: 'Initial Load' and 'Initial Load'. Below these are several buttons: 'Commit', 'Validate', 'Data flow debug' (which is turned on), and 'Debug Settings'. On the left sidebar, under 'Develop', there is a section for 'Data flows' which contains one item: 'Initial Load'. The main area shows a data flow diagram with two datasets: 'practicumsd' (source) and 'practicumdl' (sink). A plus sign (+) is located at the bottom right of the sink dataset box. The 'Sink' tab is currently selected, showing the following configuration:

- Output stream name \***: practicumdl
- Incoming stream \***: practicumsd
- Sink type \***: **Inline** (selected)
- Inline dataset type \***: Delta
- Linked service \***: practicumsw-WorkspaceDefaultStorage (with 'Test connection' and 'Edit' buttons)
- Options**:
  - Allow schema drift
  - Validate schema

On the **Sink** tab, enter values for the following items:

<b>Output Stream Name</b>	Self-Explanatory
<b>Incoming Stream</b>	Confirm default, <b>practicumsd</b>
<b>Sink Type</b>	Select <b>Inline</b>
<b>Linked Service</b>	Select <b>practicumsw-WorkspaceDefaultStorage</b>
<b>Options</b>	Confirm that " <b>Allow schema drift</b> " is checked

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. At the top, there are two tabs: 'practicumsw - Microsoft Azure' and 'practicumsw - Azure Synapse Analytics'. The URL in the address bar is <https://ms.web.azuresynapse.net/en-us/authoring/analyze/dataflow/Dataflow1?workspace=%2Fsubscriptions%2F...>. The main area is titled 'Develop' and shows a pipeline named 'Initial Load'. The pipeline consists of a source dataset 'practicumsd' and a sink dataset 'practicumdlc'. The sink dataset has 17 total columns. Below the pipeline diagram, there are several tabs: Sink, Settings (which is selected), Mapping, Optimize, Inspect, and Data preview. On the right side of the pipeline diagram, there are buttons for Commit, Validate, Data flow debug, and Debug Settings. The 'Settings' tab contains the following configuration:

Folder path *	practicumdlc / bronze/northwind/prc	<input type="button" value="Browse"/>
Compression type	snappy	<input type="button" value=""/>
Compression level	Fastest	<input type="button" value=""/>
Vacuum	0	<input type="button" value=""/>
Table action	<input checked="" type="radio"/> None <input type="radio"/> Overwrite <input type="radio"/> Truncate	<input type="button" value=""/>
Update method	<input checked="" type="checkbox"/> Allow insert <input type="checkbox"/> Allow delete <input type="checkbox"/> Allow upsert <input type="checkbox"/> Allow update	<input type="button" value=""/>
<b>Delta options</b>		
Merge schema	<input type="checkbox"/>	<input type="button" value=""/>
Auto compact	<input checked="" type="checkbox"/>	<input type="button" value=""/>
Optimize write	<input checked="" type="checkbox"/>	<input type="button" value=""/>

On the **Settings** tab, enter values for the following items:

File System	Enter <b>practicumdlc</b>
Folder Path	Enter " <b>bronze/northwind/product</b> "
Compression Type	Select <b>snappy</b>
Compression Level	Select <b>Fastest</b>
Auto Compact	Checked
Optimize Write	Checked

No additional changes are required.

Click the **Commit** button.

## Create Pipeline

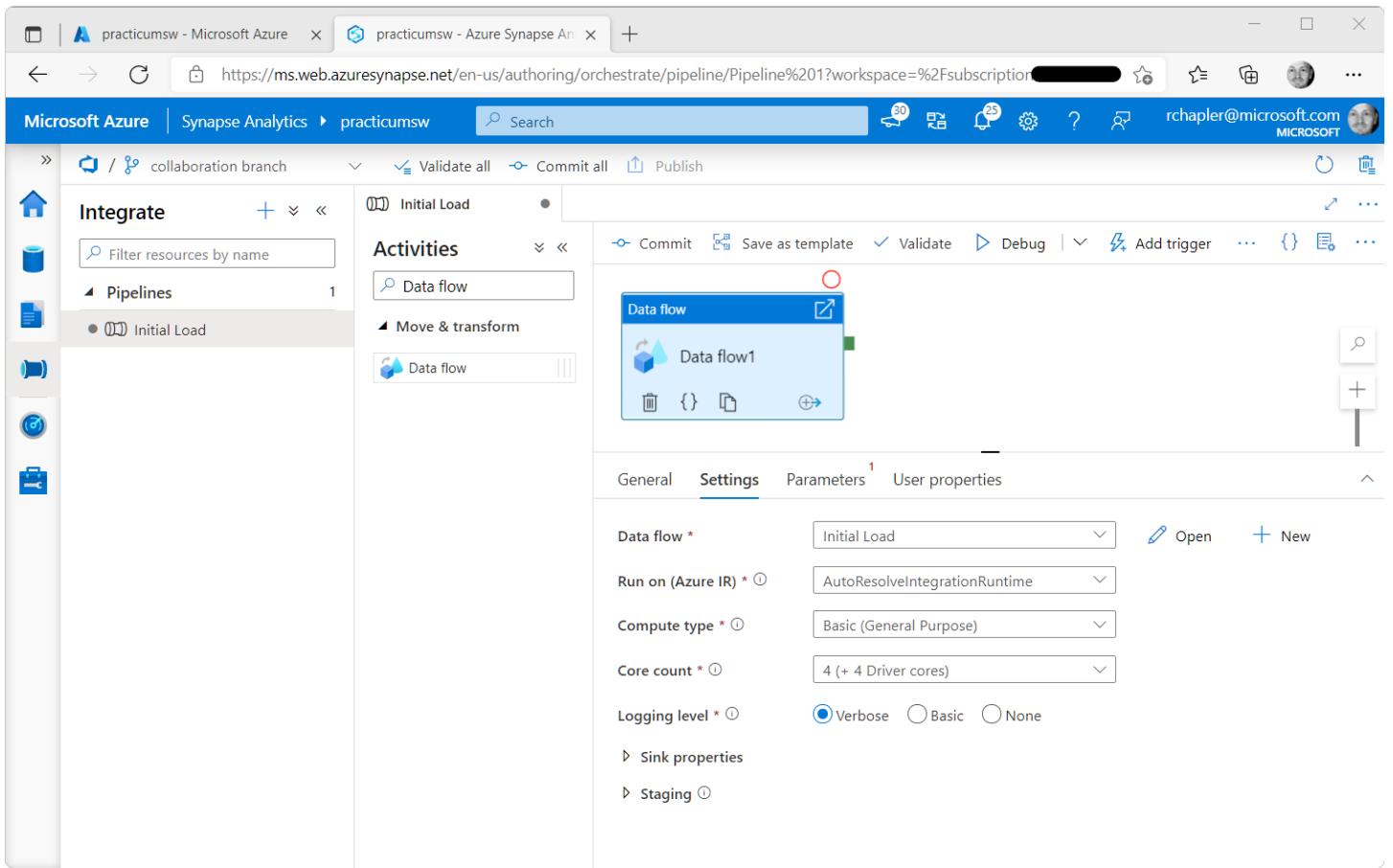
The screenshot shows the Microsoft Azure Synapse Analytics portal interface. The top navigation bar includes tabs for 'practicumsw - Microsoft Azure' and 'practicumsw - Azure Synapse Analytics'. The main area has a breadcrumb path: 'Microsoft Azure | Synapse Analytics > practicumsw'. A search bar and various navigation icons are at the top right. On the left, there's a vertical sidebar with icons for Home, Collaboration branch, Pipelines, Copy Data tool, Browse gallery, and Import from pipeline template. The main content area has a title 'Integrate' with a plus sign icon. A dropdown menu is open, showing options: 'Pipeline' (which is selected), 'Copy Data tool', 'Browse gallery', and 'Import from pipeline template'. Below this, a message says 'No items to show' and 'Try creating a new item using the + button above.' followed by a 'Learn more' link. To the right, there's a large 'Select an item' section with a 'Select' button and a note: 'Use the resource explorer to select or create a new item'. There are also two decorative icons: one showing a cylinder and a screen with code, and another showing a cylinder and a magnifying glass.

Click the **Integrate** icon in the left-hand navigation.

Click the **+** icon just above and to the right of the “**Filter resources by name**” input.

Select “**Pipeline**” from the resulting drop-down.

On the **Properties** pop-out, enter Name, “**Initial Load**”.



Search for, and then drag-and-drop a “**Data flow**” component into the activity window.

On the **Settings** tab of the “**Data flow...**” component, select your data flow.

No additional changes are required.

Click the **Commit** button.

Confirm Success

Debug Pipeline / Data Flow

Click **Debug** and confirm successful execution.

The screenshot shows the Microsoft Azure Synapse Analytics interface. On the left, the navigation bar includes 'Microsoft Azure', 'Synapse Analytics', and 'practicumsw'. The main area is titled 'Integrate' and shows a pipeline named 'Initial Load'. The pipeline details pane shows a single activity named 'Data flow1'. The pipeline run ID is cf8046e2-f692-44ce-bb3f-05f21c00ff03, and the status is 'Succeeded'.

## Query Delta Lake

Navigate to your instance of Synapse, and then click the **Develop** icon in the left-hand navigation.

Click the + icon just above and to the right of the “**Filter resources by name**” input.

Select “**Notebook**” from the resulting drop-down.

In your new notebook, select your Apache Spark pool (if required) from the “**Attach to**” drop-down.

The screenshot shows the Microsoft Azure Synapse Analytics interface in 'Develop' mode. A notebook named 'Notebook 1' is open, showing PySpark code. The code reads data from a Delta Lake table ('northwind/product') and prints the count. The output window shows the job completed successfully with 1 executor and 4 cores.

Run the following code to produce a count of records in the new Delta Lake:

```
%pyspark  
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')  
print(df.count())
```

The screenshot shows the Microsoft Azure Synapse Analytics notebook interface. On the left, there's a navigation bar with icons for Home, Data flows, Notebooks, and more. The main area is titled 'Develop' and shows 'Notebook 1'. A code editor window displays the following PySpark code:

```
1 %%pyspark
2 df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
3 display(df.limit(3))
4
```

Below the code, a message indicates a successful execution: '✓ 6 sec - Command executed in 5 sec 906 ms by rchapler on 7:24:44 AM, 10/13/21'. It also shows 'Job execution Succeeded' with 'Spark 1 executors 4 cores'. There are 'View in monitoring' and 'Open Spark UI' links. The results are presented in a table view:

ProductID	Name	ProductNumber	Color	Sta
680	HL Road Frame - Black, 58	FR-R92B-58	Black	10!
707	Sport-100 Helmet, Red	HL-U509-R	Red	13.
708	Sport-100 Helmet, Black	HL-U509	Black	13.

Run the following code to display a few records in the new Delta Lake:

```
%%pyspark
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
display(df.limit(3))
```

## Incremental Load

This section describes how to provide for recurring capture of changes to source data.

### Create Data Flow

Click the **Develop** icon in the left-hand navigation.

Click the + icon just above and to the right of the “**Filter resources by name**” input.

Select “**Data flow**” from the resulting drop-down.

Name your data flow “**Incremental Load**”

### Source (delta lake)

In this step, we will add the Delta Lake (created in the previous section) as a source.

Click on “**Add Source**” in the dashed rectangle.

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow blade. On the left, there's a navigation pane with 'Develop' selected. In the center, under 'Data flows', 'Incremental Load' is selected. A preview window shows a single row with the identifier 'practicumndl' and '0 total' columns. Below the preview, the 'Source settings' tab is active, displaying the following configuration:

- Output stream name \***: practicumndl
- Source type \***: Inline (selected)
- Inline dataset type \***: Delta
- Linked service \***: practicumsw-WorkspaceDefaultStorage
- Sampling \* ⓘ**: Disable (radio button selected)

On the “Source settings” tab, enter values for the following items:

Output Stream Name	Self-Explanatory
Source Type	Select <b>Inline</b>
Inline Dataset Type	Select <b>Delta</b>
Linked Service	Select <b>practicumsw-WorkspaceDefaultStorage</b>
Sampling	Confirm default setting, <b>Disable</b>

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow blade. In the left sidebar under 'Develop', 'Incremental Load' is selected. The main area displays a preview of a folder named 'practicumdlc' containing 0 total files. Below the preview, the 'Source options' tab is active, showing the following configuration:

- Folder path \***: practicumdlc / bronze/northwind/prc
- Allow no files found**: Checked
- Compression type**: snappy
- Compression level**: Fastest
- Time travel \***: Disable (radio button selected)

On the “Source options” tab, enter values for the following items:

<b>File System</b>	Enter <b>practicumdlc</b>
<b>Folder Path</b>	Enter “ <b>bronze/northwind/product</b> ”
<b>Allow No Files Found</b>	<b>Checked</b>
<b>Compression Type</b>	Select <b>snappy</b>
<b>Compression Level</b>	Select <b>Fastest</b>
<b>Time Travel</b>	Confirm default, <b>Disable</b>

The screenshot shows the 'Import schema' dialog box in the Microsoft Azure Synapse Analytics interface. On the left, there's a navigation sidebar with icons for Home, Databases, Tables, Scripts, Data flows, Notebooks, and Initial Load. The 'Data flows' section is expanded, showing a single item named 'practicumdl'. The main area displays 'Source settings' and 'Import schema' tabs. Under 'Import schema', it says 'No columns exist in the source'. To the right, there are configuration sections for Date, Time, Numerical whole number, Numerical fraction, Boolean true, and Boolean false. At the bottom are 'Import' and 'Cancel' buttons.

On the **Projection** tab, click “**Import schema**” and then **Import** on the resulting pop-out.

Navigate to the “**Data preview**” tab and then click **Refresh**.

The screenshot shows the 'Data preview' tab in the Microsoft Azure Synapse Analytics interface. The top navigation bar includes 'Validate all', 'Commit all', 'Publish', 'Data flow debug' (which is turned on), and 'Debug Settings'. The left sidebar is identical to the previous screenshot. The main area shows a data preview of the 'practicumdl' dataset. It indicates 17 total columns. Below the preview, there are buttons for Refresh, Typecast, Modify, Map drifted, Statistics, and Remove. A table below shows four rows of data:

	ProductID	Name	ProductNumber	Color
+	680	HL Road Frame - Black, 58	FR-R92B-58	Black
+	707	Sport-100 Helmet, Red	HL-U509-R	Red
+	708	Sport-100 Helmet, Black	HL-U509	Black

Click the **Commit** button.

## Source (sample database for this exercise)

In this step, we will add the SQL Northwind sample database (our primary source for this exercise).

Click on “Add Source” in the dashed rectangle.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. The left sidebar shows 'Develop' mode with 'Data flows' selected. The main area shows two datasets: 'practicumdl' (Add source dataset) and 'practicumsd' (Columns: 0 total). The 'practicumsd' dataset is currently selected. The 'Source settings' tab is active, displaying the following configuration:

- Output stream name: practicumsd
- Source type: Inline (selected)
- Inline dataset type: Azure SQL Database
- Linked service: practicumsd (Connection successful)
- Sampling: Disable (selected)

On the “Source settings” tab, enter values for the following items:

Output Stream Name	Self-Explanatory
Source Type	Select <b>Inline</b>
Inline Dataset Type	Select <b>“Azure SQL Database”</b>
Linked Service	Select <b>practicumsd</b>
Sampling	Confirm default setting, <b>Disable</b>

Click “Test connection” to confirm successful configuration.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. On the left, there's a navigation pane with icons for Home, Databases, Notebooks, Data flows, and Incremental Load. The 'Data flows' section is expanded, showing two items: 'Incremental Load' and 'Initial Load'. The 'Incremental Load' item is selected. The main workspace displays two stages: 'practicumdl' (Incremental Load) and 'practicumsd' (Initial Load). The 'practicumdl' stage has an 'Add source dataset' button. Below the stages, there are tabs for 'Source settings', 'Source options' (which is selected), 'Projection', 'Optimize', 'Inspect', 'Data preview', and 'Description'. Under the 'Source options' tab, the 'Input' section is set to 'Table' (selected via radio button), with 'Schema name' set to 'SalesLT', 'Table name' set to 'Product', and 'Isolation level' set to 'Read uncommitted'. There are also 'Refresh' and 'Description' buttons.

On the “Source options” tab, enter values for the following items:

<b>Input</b>	Confirm default setting, <b>Table</b>
<b>Schema Name</b>	Enter <b>SalesLT</b>
<b>Table Name</b>	Enter <b>Product</b>

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow workspace. The left sidebar lists 'Develop' branches: 'Notebooks' (1), 'Data flows' (2), and 'Incremental Load'. The 'Incremental Load' branch is selected. The main area displays a data flow diagram with two stages: 'practicumdl' (Add source dataset) and 'practicumsd' (Columns: 17 total). The 'Projection' tab is active, showing the schema:

Column name	Type
ProductID	integer
Name	string
ProductNumber	string
Color	string

On the **Projection** tab, click “**Import schema**” and then **Import** on the resulting pop-out.

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow interface. On the left, the navigation pane is visible with sections for Develop, Notebooks, Data flows, and Incremental Load. The 'Incremental Load' section is currently selected. In the main workspace, there are two datasets: 'practicumdl' (Add source dataset) and 'practicumsd' (Columns: 17 total). A join operation is being performed between them. The 'Data preview' tab is selected at the bottom, showing a table with three rows. The columns are ProductID, Name, ProductNumber, and Color. The data is as follows:

	ProductID	Name	ProductNumber	Color
+	680	HL Road Frame - Black, 58	FR-R92B-58	Black
+	707	Sport-100 Helmet, Red	HL-U509-R	Red

Navigate to the “**Data preview**” tab and then click **Refresh**.

Click the **Commit** button.

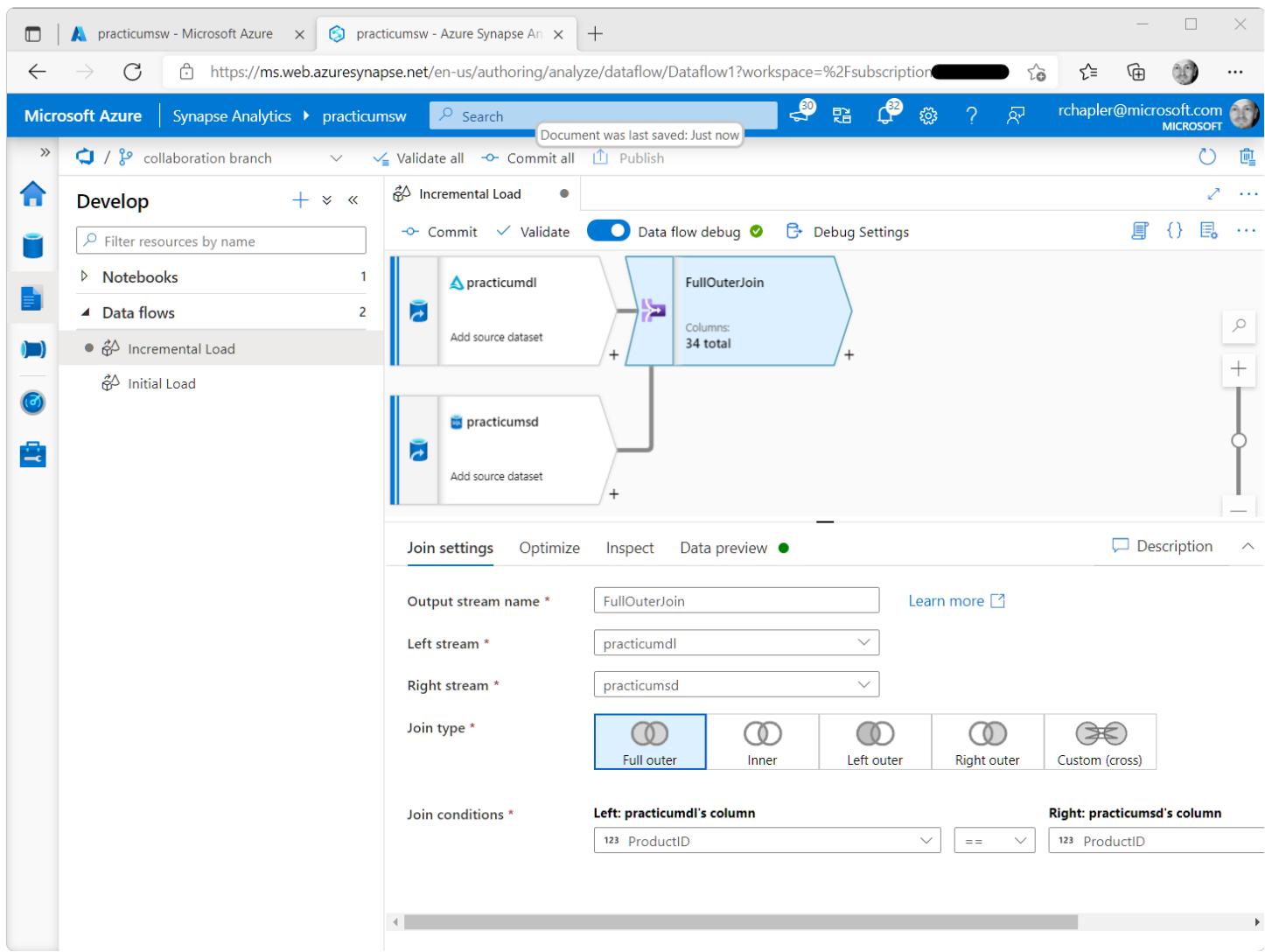
#### Join Sources

In this step, we will perform a full outer join between source datasets.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow interface. On the left, the navigation pane is visible with sections like 'Develop', 'Notebooks' (1 item), 'Data flows' (2 items), and 'Initial Load'. The 'Data flows' section has 'Incremental Load' selected. The main workspace displays a data flow diagram with two datasets: 'practicumdl' (17 total columns) and 'practicumsd' (multiple inputs/outputs). A 'join' operation is being configured between them. Below the diagram, the 'Data preview' tab is active, showing a table with three rows of data:

	ProductID	Name	ProductNumber	Color
+	680	HL Road Frame - Black, 58	FR-R92B-58	Black
+	707	Sport-100 Helmet, Red	HL-U509-R	Red

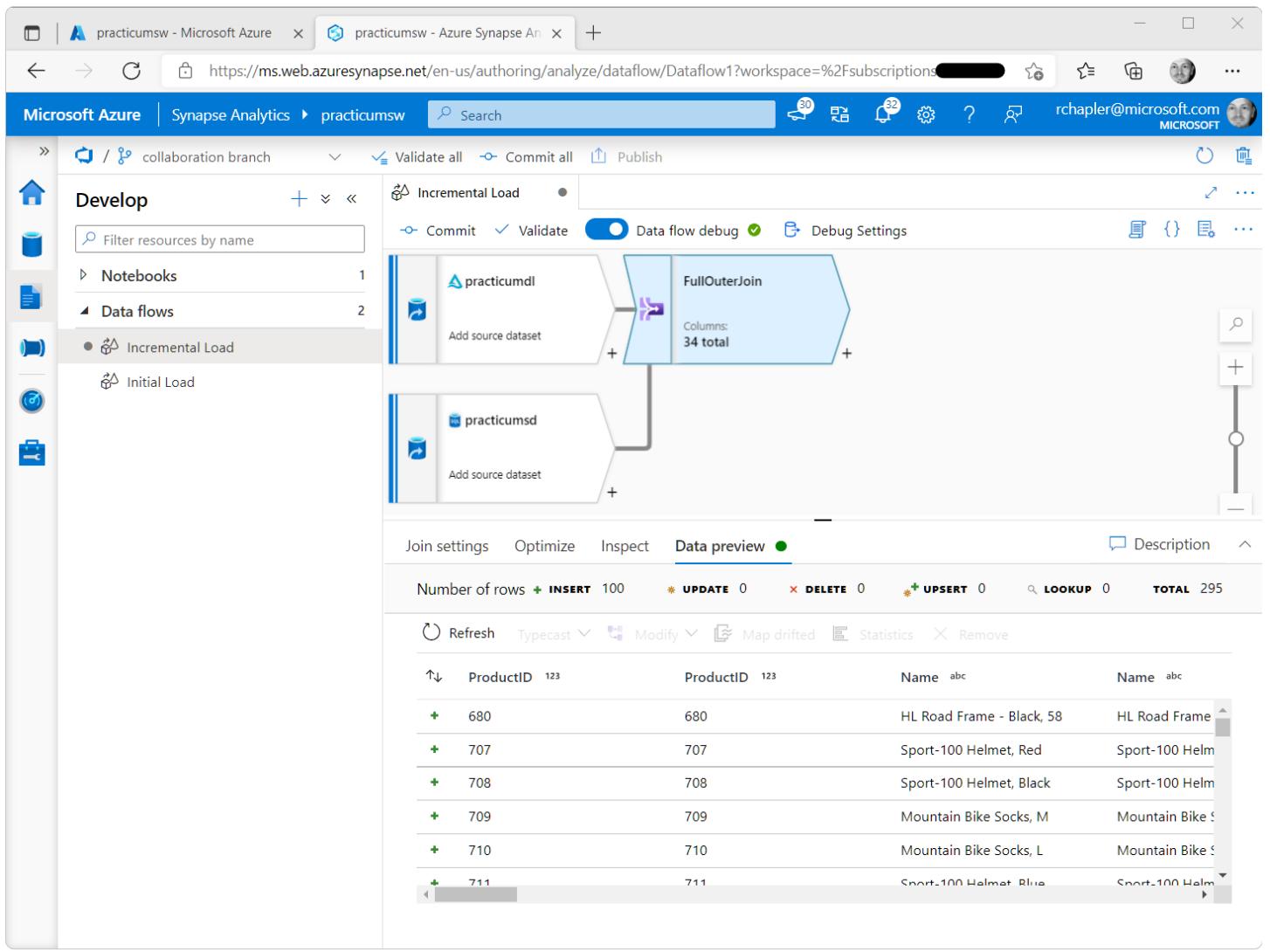
Click the **+** in the bottom right of the SQL source, and then select **Join** from the resulting pop-up list.



On the “Join settings” tab, enter values for the following items:

<b>Output Stream Name</b>	Self-Explanatory
<b>Left Stream</b>	Select <b>practicumsd</b>
<b>Right Stream</b>	Select <b>AggregateWaterline</b>
<b>Join Type</b>	Select “ <b>Full outer</b> ”
<b>Join Conditions</b>	Select <b>ProductID</b> for both Left... and Right... column

Navigate to the “Data preview” tab and then click **Refresh**.



Click the **Commit** button.

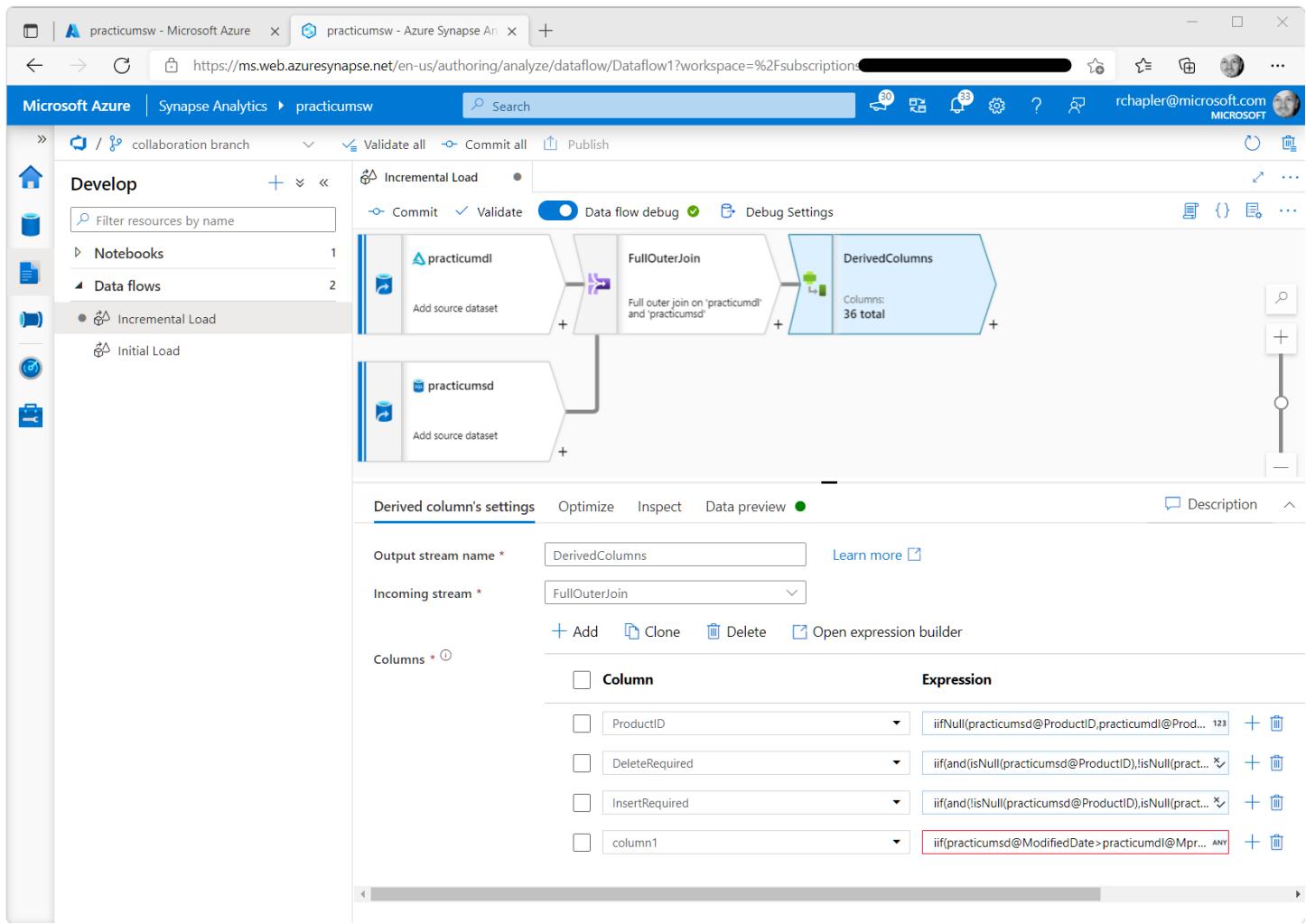
### Derive Columns

In this step, we will add columns designed to characterize required Delta Lake changes.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. On the left, the navigation pane is visible with sections like 'Develop', 'Notebooks', 'Data flows', and 'Initial Load'. The main workspace displays a data flow diagram titled 'Incremental Load'. It consists of two 'Add source dataset' components (labeled 'practicumdl' and 'practicumsd') connected to a 'FullOuterJoin' component. The 'FullOuterJoin' component has 34 total columns. A 'Schema modifier' panel is open on the right, showing a 'Derived Column' option. Below the diagram, the 'Data preview' tab is selected, showing a table with 100 rows. The table has columns: ProductID, ProductID, Name, Name, and ProductN. The data includes various product names and IDs.

	ProductID	ProductID	Name	Name	ProductN
+	680	680	HL Road Frame - Black, 58	HL Road Frame - Black, 58	FR-R9
+	707	707	Sport-100 Helmet, Red	Sport-100 Helmet, Red	HL-U5
+	708	708	Sport-100 Helmet, Black	Sport-100 Helmet, Black	HL-U5
+	709	709	Mountain Bike Socks, M	Mountain Bike Socks, M	SO-B9
+	710	710	Mountain Bike Socks, L	Mountain Bike Socks, L	SO-B9
+	711	711	Sport-100 Helmet, Blue	Sport-100 Helmet, Blue	HL-U5
+	712	712	AWC Logo Cap	AWC Logo Cap	CA-10

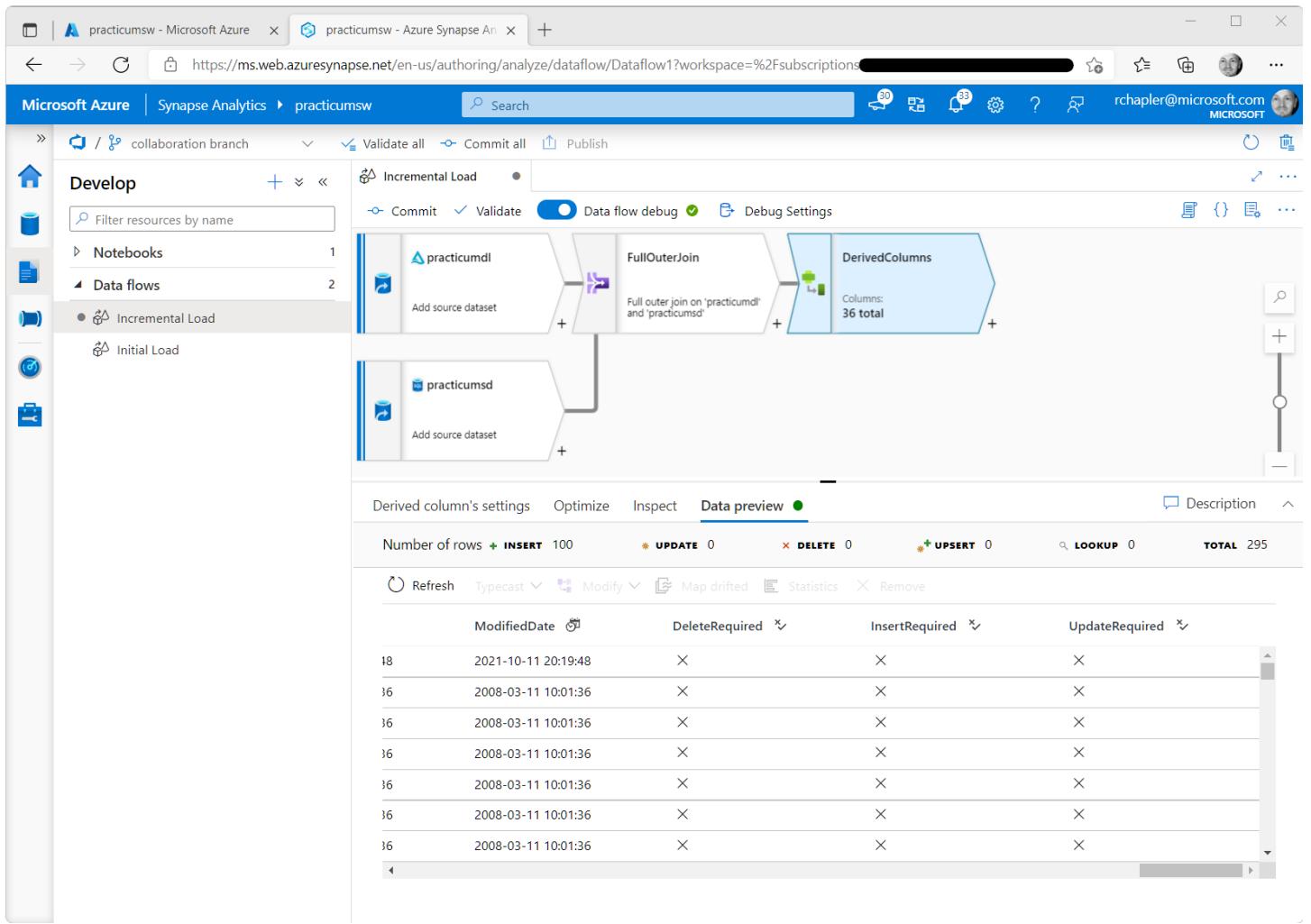
Click the **+** in the bottom right of the SQL source, and then select **Derived Column** from the resulting pop-up list.



On the “Derived column’s settings” tab, enter values for the following items:

<b>Output Stream Name</b>	Self-Explanatory										
<b>Incoming Stream</b>	Confirm default selection										
<b>Columns</b>	<table> <tr> <th>Column</th> <th>Expression</th> </tr> <tr> <td>ProductID</td> <td><code>iifNull(practicumsd@ProductID,practicumdl@ProductID)</code></td> </tr> <tr> <td>DeleteRequired</td> <td><code>iif(and(isNull(practicumsd@ProductID),!isNull(practicumdl@ProductID)),true(),false())</code></td> </tr> <tr> <td>InsertRequired</td> <td><code>iif(and(!isNull(practicumsd@ProductID),isNull(practicumdl@ProductID)),true(),false())</code></td> </tr> <tr> <td>UpdateRequired</td> <td><code>iif(practicumsd@ModifiedDate&gt;practicumdl@ModifiedDate,true(),false())</code></td> </tr> </table>	Column	Expression	ProductID	<code>iifNull(practicumsd@ProductID,practicumdl@ProductID)</code>	DeleteRequired	<code>iif(and(isNull(practicumsd@ProductID),!isNull(practicumdl@ProductID)),true(),false())</code>	InsertRequired	<code>iif(and(!isNull(practicumsd@ProductID),isNull(practicumdl@ProductID)),true(),false())</code>	UpdateRequired	<code>iif(practicumsd@ModifiedDate&gt;practicumdl@ModifiedDate,true(),false())</code>
Column	Expression										
ProductID	<code>iifNull(practicumsd@ProductID,practicumdl@ProductID)</code>										
DeleteRequired	<code>iif(and(isNull(practicumsd@ProductID),!isNull(practicumdl@ProductID)),true(),false())</code>										
InsertRequired	<code>iif(and(!isNull(practicumsd@ProductID),isNull(practicumdl@ProductID)),true(),false())</code>										
UpdateRequired	<code>iif(practicumsd@ModifiedDate&gt;practicumdl@ModifiedDate,true(),false())</code>										

Navigate to the “Data preview” tab and then click **Refresh**.

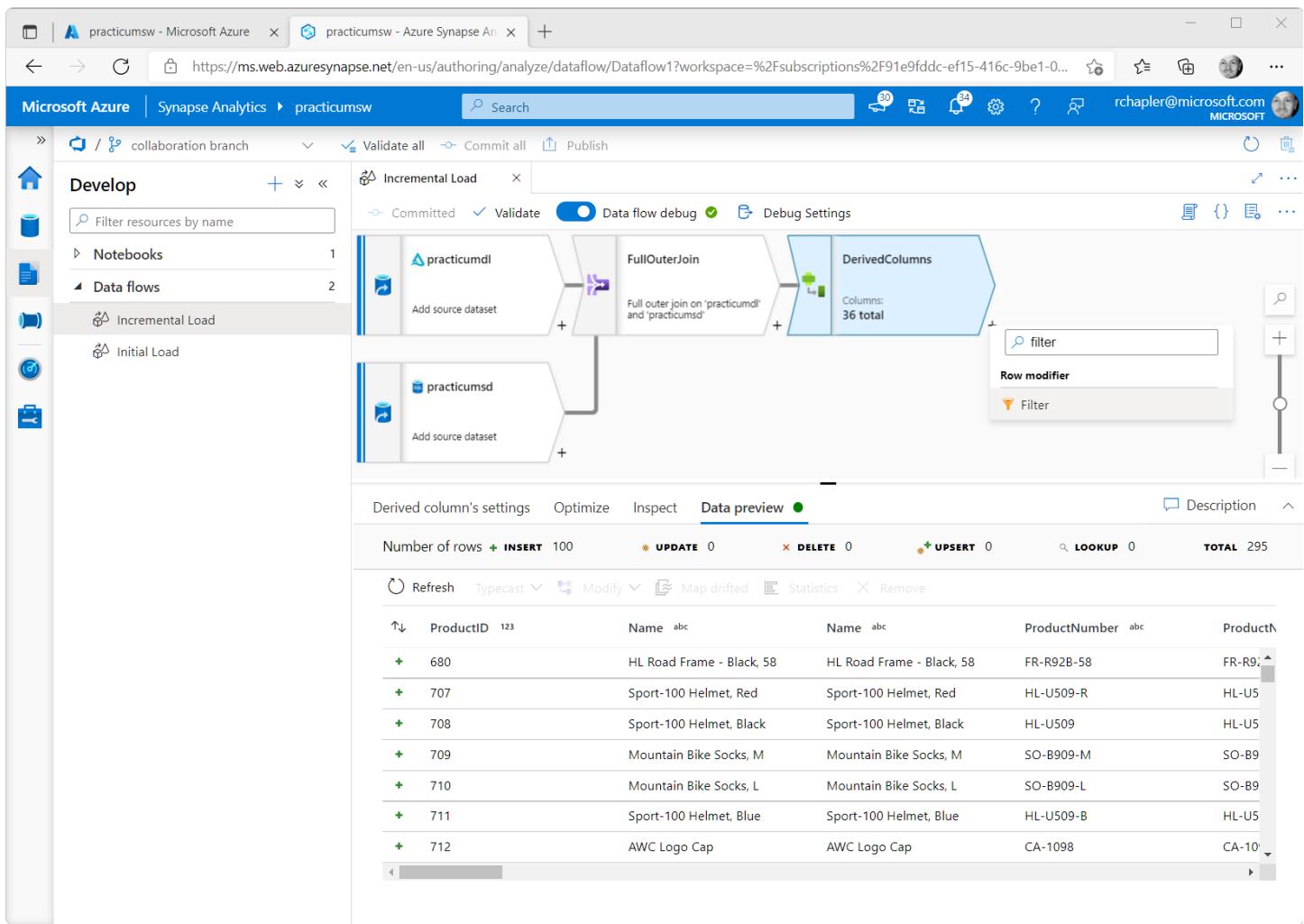


Click the **Commit** button.

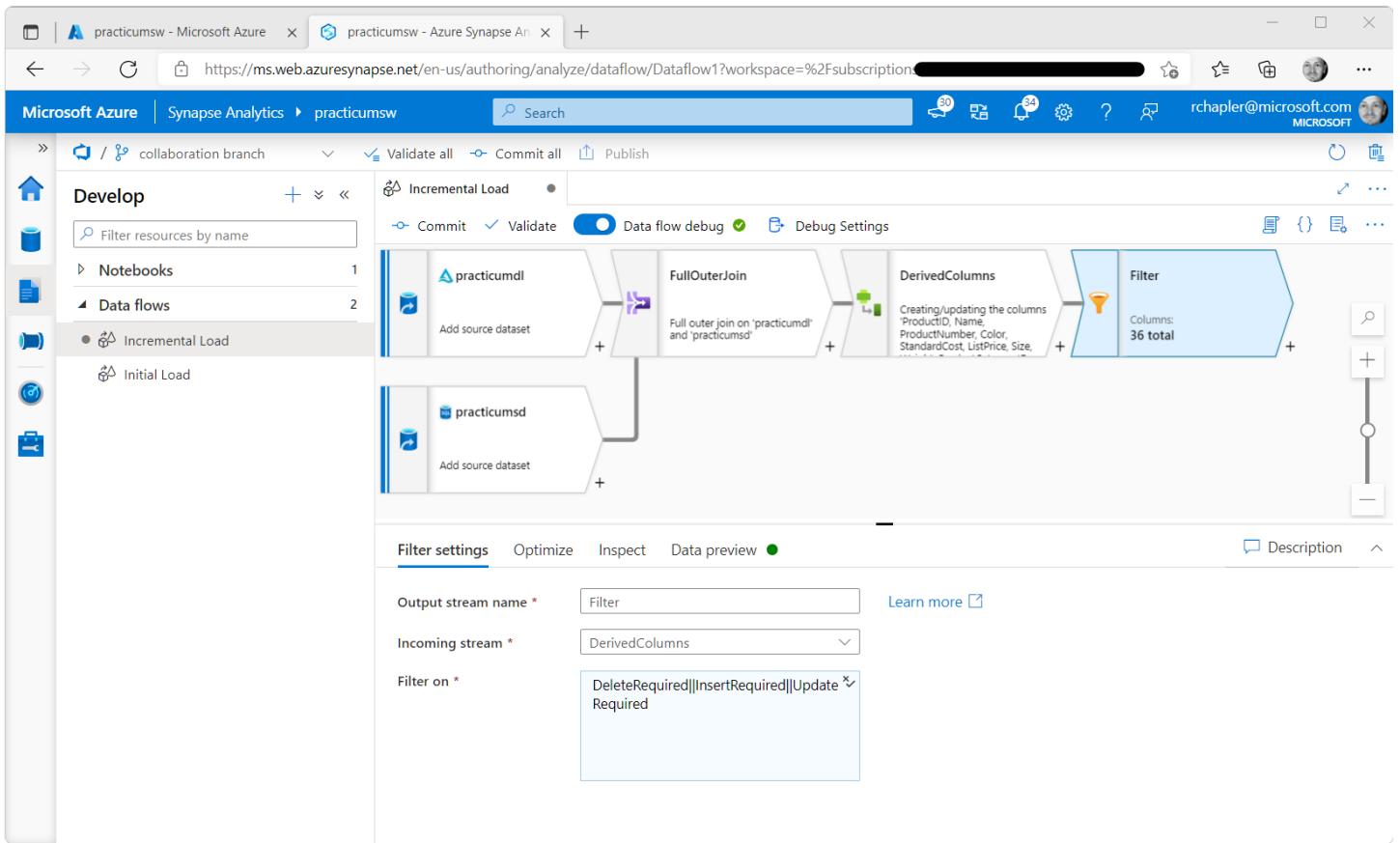
*Note: I included this step to provide an easy way to preview and provide for the delete, insert and update of data... arguably, this step could be consolidated into Filter or AlterRows.*

### Filter Non-Actionable

In this step, we will remove rows that have no required action.



Click the **+** in the bottom right of the SQL source, and then select **Filter** from the resulting pop-up list.



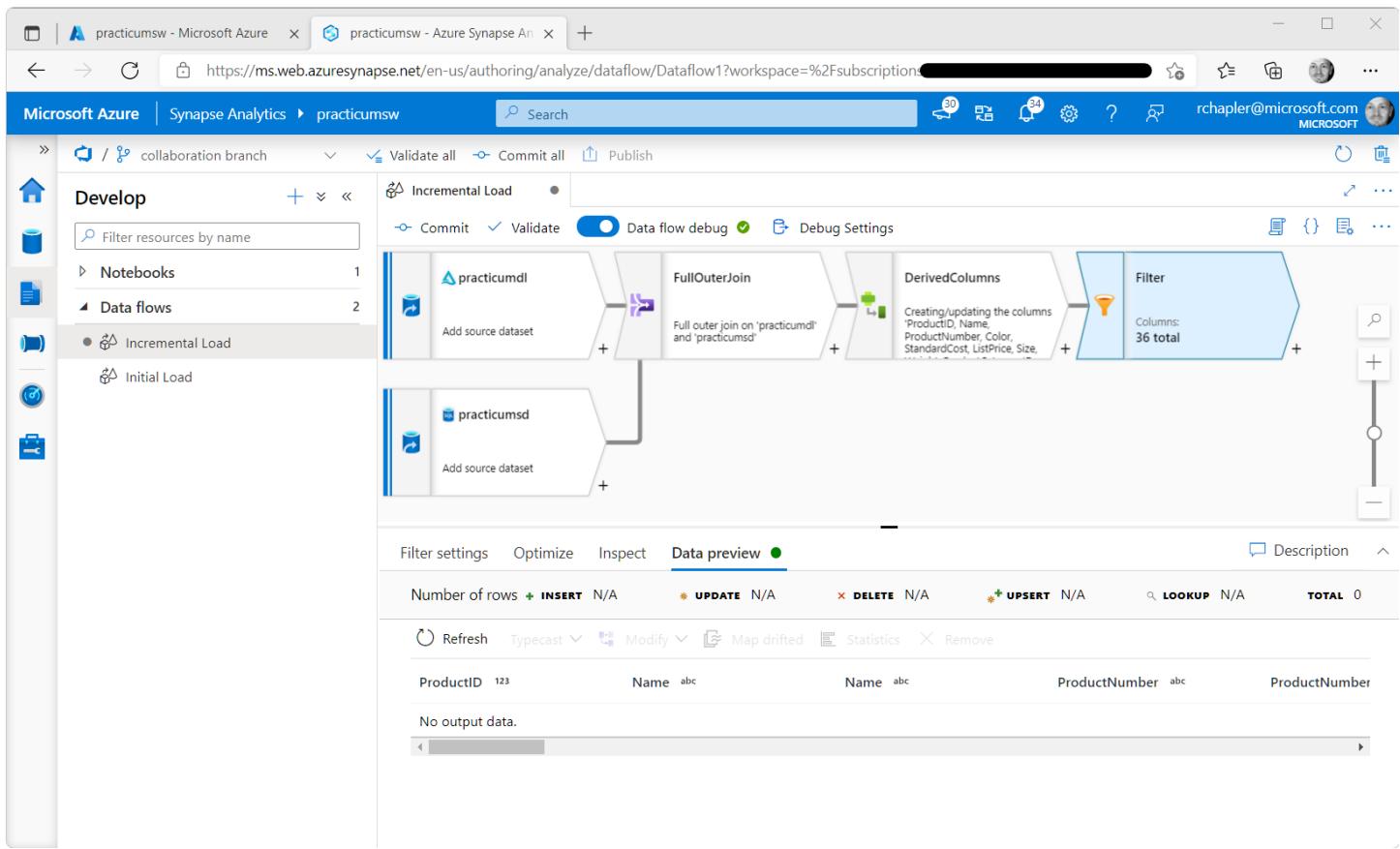
On the “**Filter settings**” tab, enter values for the following items:

---

<b>Output Stream Name</b>	Self-Explanatory
<b>Incoming Stream</b>	Confirm default selection
<b>Filter On</b>	<code>DeleteRequired    InsertRequired    UpdateRequired</code>

---

Navigate to the “**Data preview**” tab and then click **Refresh**.



Click the **Commit** button.

*Note: Because we are doing an incremental load immediately after an initial load, on a sample database which doesn't have unexpected change, we see no current results in the Data Preview.*

## Change Examples

To make the remaining steps a bit more interesting, we will force changes to the sample database.

Navigate to your Azure SQL Database, click the “**Query editor (preview)**” link in the left-hand navigation, and login.

## Delete Example

The screenshot shows the Microsoft Azure Query editor (preview) interface. The left sidebar lists various services: Overview, Activity log, Tags, Diagnose and solve problems, Quick start, and Query editor (preview). The Query editor (preview) option is selected. The main area displays a query window titled "Query 1". The query is:

```
1 DELETE FROM [SalesLT].[Product] WHERE [ProductID]=730
```

The "Messages" tab shows the result: "Query succeeded: Affected rows: 1". A yellow banner at the bottom also indicates "Query succeeded | 0s".

Execute the following T-SQL:

```
DELETE FROM [SalesLT].[Product] WHERE [ProductID]=730
```

*Note: I chose 730 because it did not trigger constraints; any row deletion will work.*

## Insert Example

The screenshot shows the Microsoft Azure Query editor (preview) interface. The left sidebar lists various services: Overview, Activity log, Tags, Diagnose and solve problems, Quick start, and Query editor (preview). The Query editor (preview) option is selected. The main area displays a query window titled "Query 1". The query code is:

```
1 INSERT INTO [SalesLT].[Product]
2     ([Name],[ProductNumber],[Color],[StandardCost],[ListPrice],[SellStartDate],[rowguid],[ModifiedDate])
3 VALUES
4     ('New Product ABC','ABC-123','Bronze',1000,1500,getdate(),newid(),getdate())
```

The "Messages" tab shows the message: "Query succeeded: Affected rows: 1". A yellow banner at the bottom also indicates "Query succeeded | 0s".

Execute the following T-SQL:

```
INSERT INTO [SalesLT].[Product]
    ([Name],[ProductNumber],[Color],[StandardCost],[ListPrice],[SellStartDate],[rowguid],[ModifiedDate])
VALUES
    ('New Product ABC','ABC-123','Bronze',1000,1500,getdate(),newid(),getdate())
```

## Update Example

The screenshot shows the Microsoft Azure Query editor (preview) interface. The left sidebar lists various services: Overview, Activity log, Tags, Diagnose and solve problems, Quick start, and Query editor (preview). The Query editor (preview) option is selected. The main area displays a query window titled "Query 1". The query code is:

```
1 UPDATE [SalesLT].[Product]
2 SET [ListPrice] = 999.99, [ModifiedDate] = getdate()
3 WHERE ProductID=680
4
```

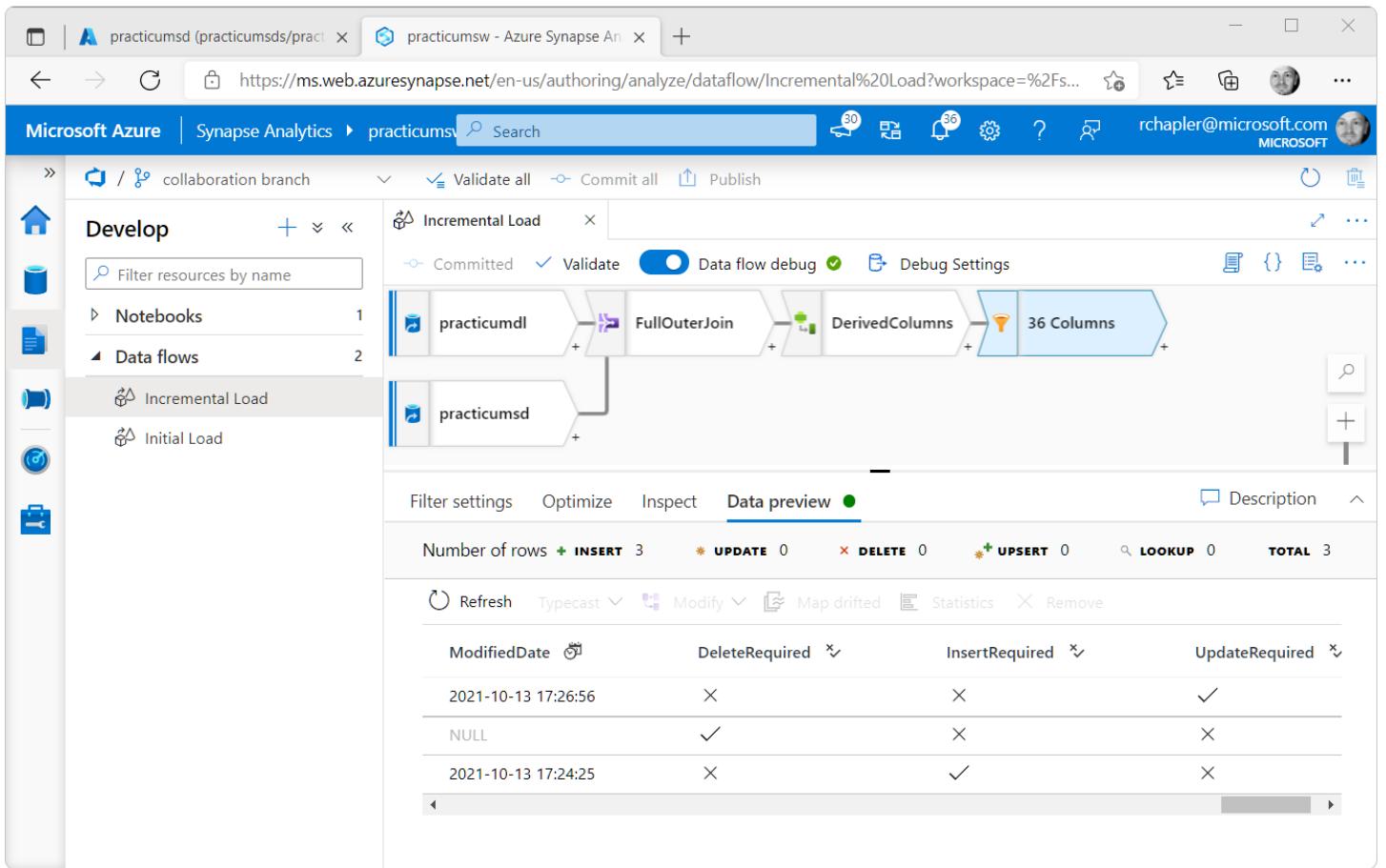
Below the query window, the "Messages" tab is active, showing the message: "Query succeeded: Affected rows: 1". At the bottom of the screen, a yellow bar indicates "Query succeeded | 0s".

Execute the following T-SQL:

```
UPDATE [SalesLT].[Product]
SET [ListPrice] = 999.99, [ModifiedDate] = getdate()
WHERE ProductID=680
```

Confirm Changes

Return to the “**Incremental Load**” data flow and sequentially update with “**Data preview**” > **Refresh**.

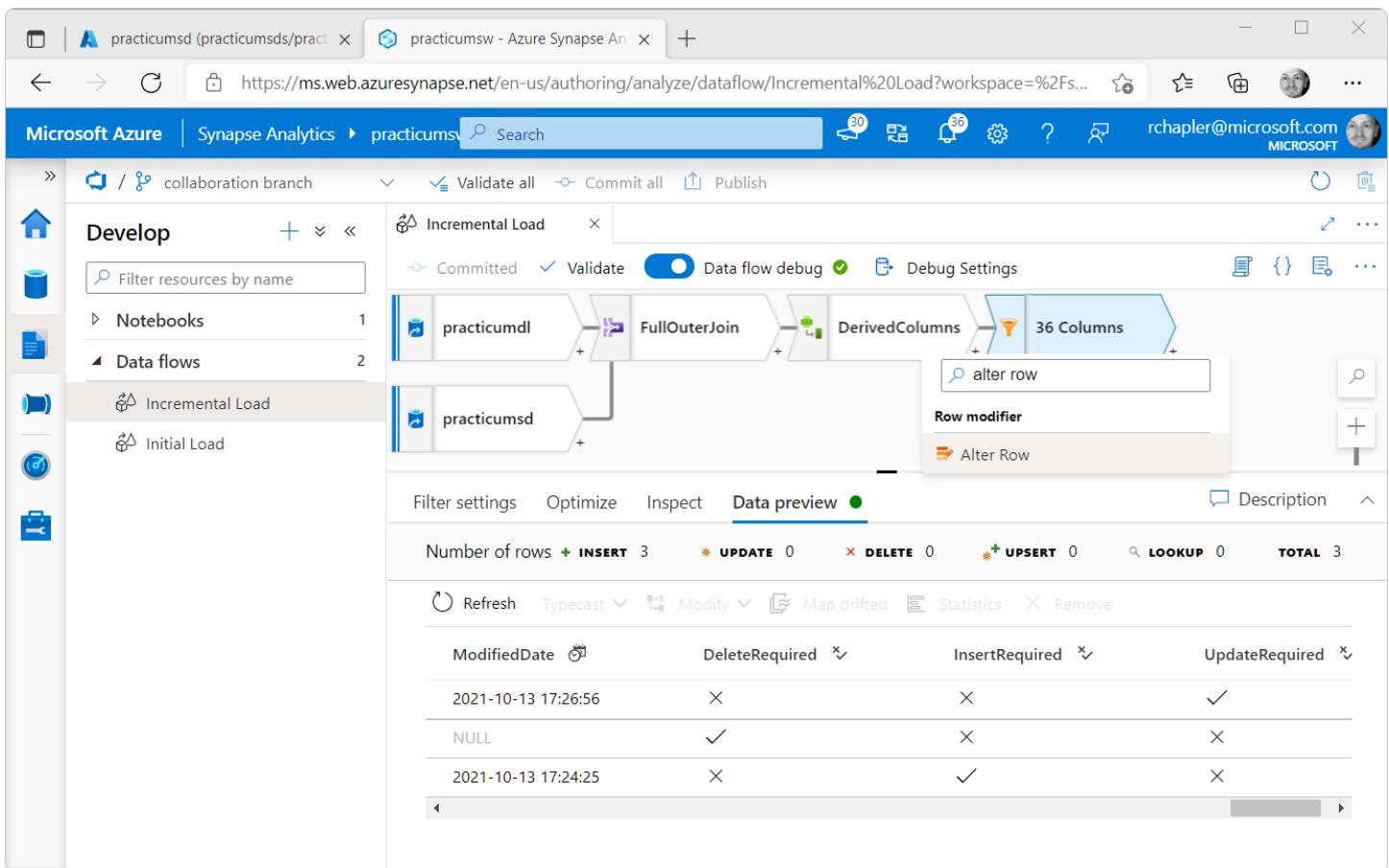


After update, you should see the three actions listed in **Filter > “Data preview”**.

## Finalize Data Flow

### Alter Row

In this step, we will apply delete, insert, and update policies to the rows.



Click the + in the bottom right of the SQL source, and then select “Alter Row” from the resulting pop-up list.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade with the 'Alter row settings' tab selected. The pipeline 'Incremental Load' is shown. The 'Alter row settings' tab contains the following configuration:

- Output stream name \***: AlterRow
- Incoming stream \***: Filter
- Alter row conditions \*** (with a help icon):
  - Delete if: DeleteRequired
  - Insert if: InsertRequired
  - Update if: UpdateRequired

On the “Alter row settings” tab, enter values for the following items:

<b>Output Stream Name</b>	Self-Explanatory
<b>Incoming Stream</b>	Confirm default selection
<b>Alter Row Conditions</b>	"Delete if" DeleteRequired
	"Insert if" InsertRequired
	"Update if" UpdateRequired

Navigate to the “Data preview” tab and then click **Refresh**.

ProductID	Name	Name	ProductNumber	ProductNumber
680	HL Road Frame - Black, 58	HL Road Frame - Black, 58	FR-R92B-58	FR-R92B-58
730	LL Road Frame - Red, 62	NULL	FR-R38R-62	NULL
1006	NULL	New Product ABC	NULL	ABC-123

Click the **Commit** button.

Sink (delta lake)

In this step, we finalize the data flow.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade. On the left, the 'Develop' sidebar lists 'Incremental Load' as the active pipeline. The main area displays a data flow diagram titled 'Incremental Load'. The flow starts with a 'practicumdl' source, followed by a 'FullOuterJoin' operation, then a 'DerivedColumns' operation, a 'Filter' operation, and finally a '36 Columns' sink. A second source, 'practicumsd', is connected to the 'FullOuterJoin' operation. On the right, there's a 'sink' search bar and a 'Destination' section with a 'Sink' button. Below the diagram, the 'Data preview' tab is selected, showing a table with three rows. The table has columns for ProductID, Name, and ProductNumber.

	ProductID	Name	ProductNumber
*	680	HL Road Frame - Black, 58	FR-R92B-58
x	730	LL Road Frame - Red, 62	FR-R38R-62
+	1006	NULL	ABC-123

Click the + in the bottom right of the SQL source, and then select **Sink** from the resulting pop-up list.

The screenshot shows the Microsoft Azure Synapse Analytics Data Flow blade with the 'Sink' tab selected. The 'AlterRow' component is highlighted with a dashed blue border. The 'Sink' tab configuration includes:

- Output stream name:** Sink
- Incoming stream:** AlterRow
- Sink type:** Inline (selected)
- Inline dataset type:** Delta
- Linked service:** practicums-WorkspaceDefaultStorage (selected)
- Options:** Allow schema drift (unchecked), Validate schema (unchecked)

On the **Sink** tab, enter values for the following items:

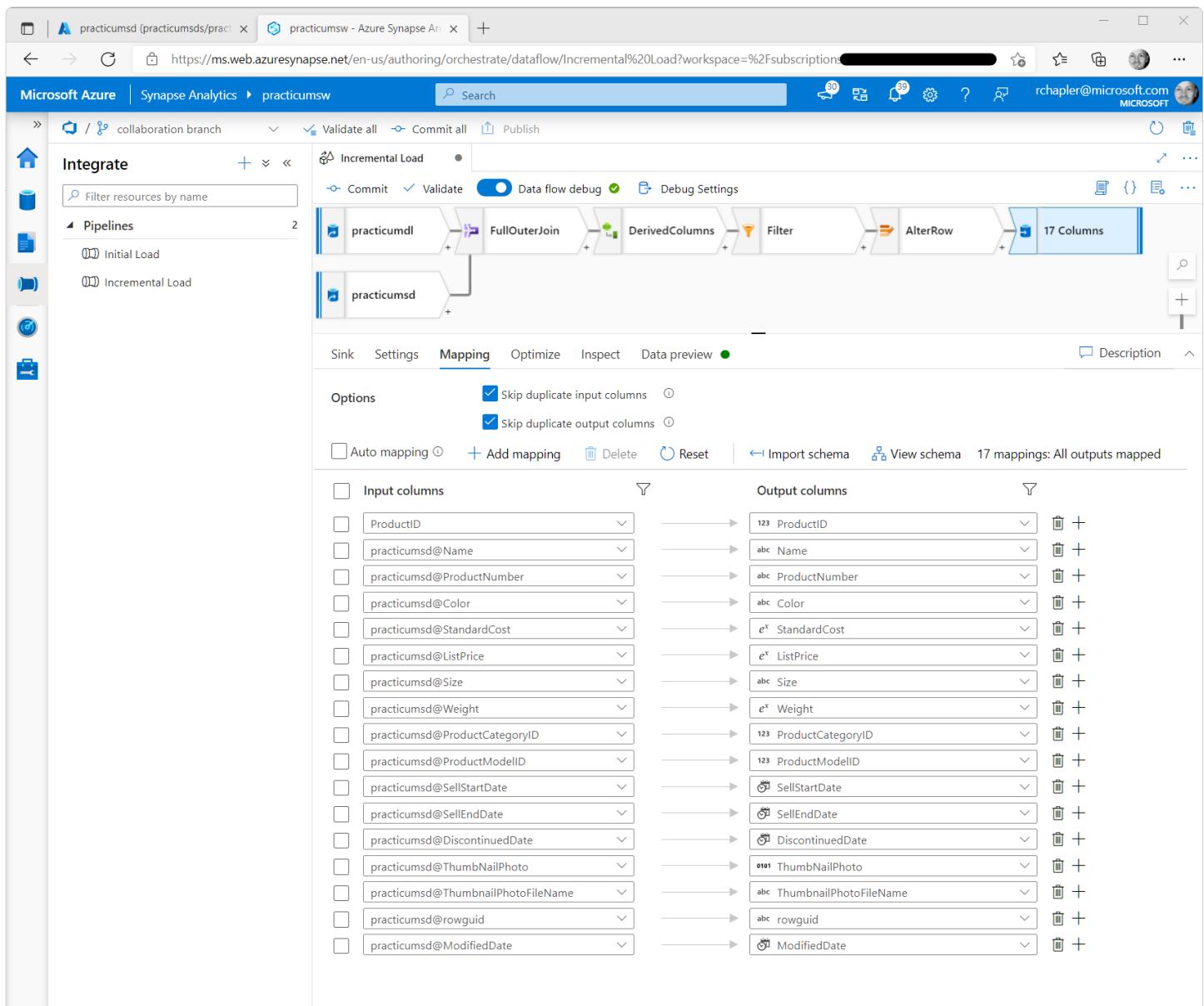
<b>Output Stream Name</b>	Self-Explanatory
<b>Sink Type</b>	Select <b>Inline</b>
<b>Inline Dataset Type</b>	Select <b>Delta</b>

The screenshot shows the Microsoft Azure Synapse Analytics studio interface. The left sidebar shows 'Microsoft Azure | Synapse Analytics > practicumsw'. The main area displays a pipeline named 'Incremental Load'. The pipeline consists of several stages: 'practicumdl' (source), 'FullOuterJoin', 'DerivedColumns', 'Filter', 'AlterRow', and a final stage labeled '20 Columns' (sink). The 'Settings' tab is currently selected for the sink stage. The configuration includes:

- File System:** Enter **practicumdlc**
- Folder Path:** Enter "**bronze/northwind/product**"
- Compression Type:** Select **snappy**
- Compression Level:** Select **Fastest**
- Update Method:** "Allow insert", "Allow delete", and "Allow update" checked
- Key Columns:** Confirm default selection, "List of columns" and select **ProductID**
- Auto Compact:** Checked
- Optimize Write:** Checked

On the **Settings** tab, enter values for the following items:

<b>File System</b>	Enter <b>practicumdlc</b>
<b>Folder Path</b>	Enter " <b>bronze/northwind/product</b> "
<b>Compression Type</b>	Select <b>snappy</b>
<b>Compression Level</b>	Select <b>Fastest</b>
<b>Update Method</b>	"Allow insert", "Allow delete", and "Allow update" checked
<b>Key Columns</b>	Confirm default selection, "List of columns" and select <b>ProductID</b>
<b>Auto Compact</b>	Checked
<b>Optimize Write</b>	Checked

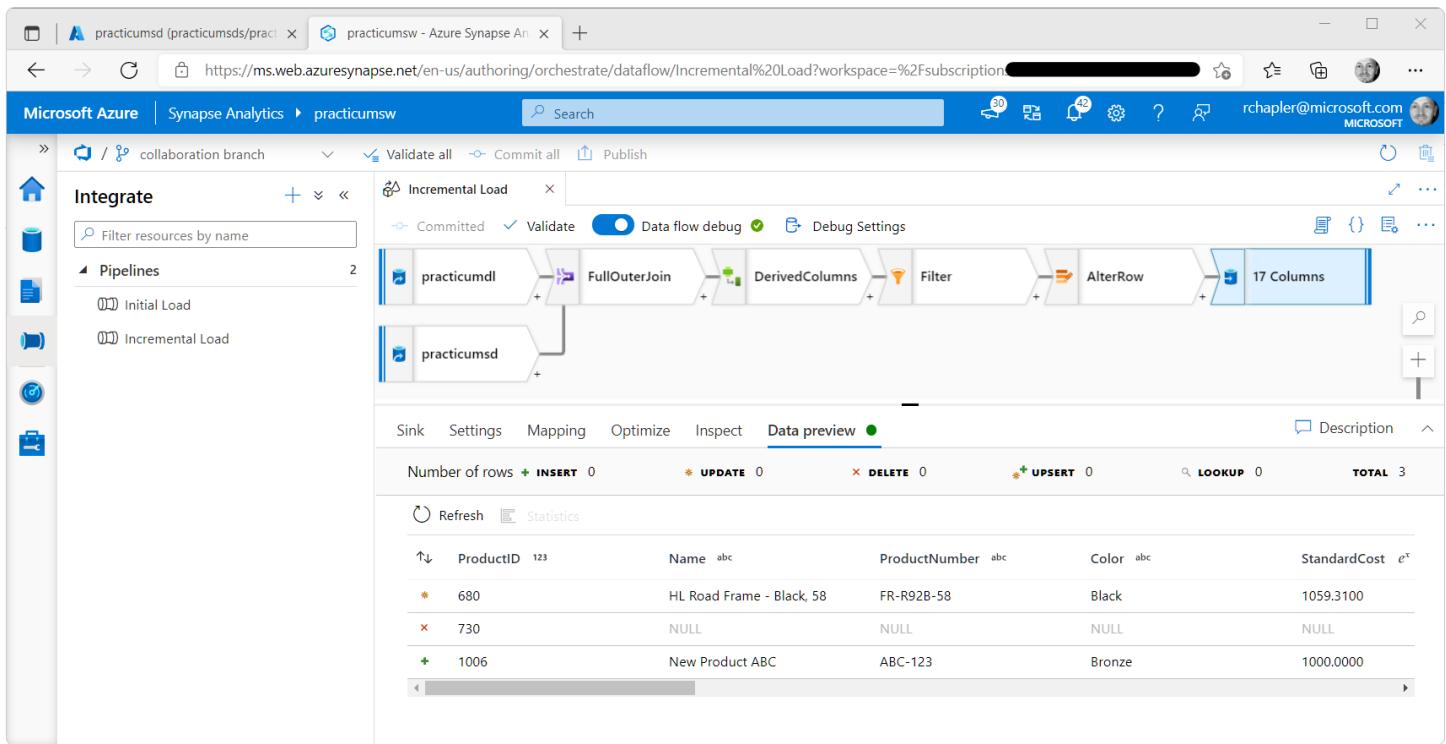


On the **Mapping** tab, un-check “Auto mapping”, click “Import schema” and then **Import** on the resulting pop-out.

Review the auto-generated list of columns and remove:

- Rows where “**Input columns**” begin with “**practicumdl...**” (need data changes from the source, not the waterline dataset)
- Derived Columns **DeleteRequired**, **InsertRequired**, and **UpdateRequired**

Navigate to the “**Data preview**” tab and then click **Refresh**.



Click the **Commit** button.

## Create Pipeline

Click the **Integrate** icon in the left-hand navigation.

Click the **+** icon just above and to the right of the “**Filter resources by name**” input.

Select “**Pipeline**” from the resulting drop-down.

On the **Properties** pop-out, enter Name, “**Incremental Load**”.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline editor. On the left, the sidebar displays 'Integrate' and 'Pipelines' under 'Activities'. A 'Data flow' component is selected. The main area shows the 'Incremental Load' activity settings for this component. The 'Settings' tab is active, showing the following configuration:

- Data flow: Incremental Load
- Run on (Azure IR): AutoResolveIntegrationRuntime
- Compute type: Basic (General Purpose)
- Core count: 4 (+ 4 Driver cores)
- Logging level: Verbose (selected)

At the top right, there are buttons for Commit, Save as template, Validate, Debug, and Publish.

Search for, and then drag-and-drop a “**Data flow**” component into the activity window.

On the **Settings** tab of the “**Data flow...**” component, select your data flow.

No additional changes are required.

Click the **Commit** button.

Confirm Success

Debug Pipeline / Data Flow

Click **Debug** and confirm successful execution.

The screenshot shows the Microsoft Azure Synapse Analytics pipeline interface. On the left, the navigation bar includes 'Microsoft Azure', 'Synapse Analytics', and 'practicumsw'. The main area displays an 'Integrate' pipeline titled 'Incremental Load'. Under 'Activities', there is a single 'Data flow' activity named 'Data flow1'. The 'Output' tab is selected, showing a table with one row: 'Data flow1' (Type: Data flow, Run start: 2021-10-13T20:30:16.012, Duration: 00:00:34, Status: Succeeded, Integ: Auto). Pipeline run ID is 8524fdc2-bb62-4066-848b-49aac733c988.

## Review Produced Files

Click the **Data** icon in the left-hand navigation, and then the **Linked** tab.

Expand “**Azure Data Lake Storage Gen2**”, then **practicumsw** to **practicumdlc**.

Click to navigate to “**practicumdlc > bronze > northwind > product**”

The screenshot shows the Microsoft Azure Synapse Analytics Data blade. The left navigation bar has 'Data' selected, with 'Linked' tab highlighted. The main area shows a list of cached items from the 'practicumdlc' storage account, specifically from the 'bronze' container under 'northwind' and 'product'. The table includes columns for Name, Last Modified, Content Type, and Size. The table lists 158 items, with the first few rows shown:

Name	Last Modified	Content Type	Size
_delta_log	10/13/2021, 6:51:37 AM	Folder	
part-00000-6d4ac397-beee-4b62-9dfc-33458fb11fd6-c000.snappy.parquet	10/13/2021, 6:51:38 AM		147.3 KB
part-00000-82bf94ed-432a-4b12-958e-92309788c2b7-c000.snappy.parquet	10/13/2021, 1:30:25 PM		14.2 KB
part-00001-28ac0035-b7d0-48d7-b5ae-2cfb0e133fb0-c000.snappy.parquet	10/13/2021, 1:30:25 PM		13.9 KB
part-00003-55c643f4-1b5b-4b1d-a69b-f7f557e6f843-c000.snappy.parquet	10/13/2021, 1:30:25 PM		7.5 KB
part-00004-3817f4e7-f32f-45b9-a797-5f94050729e5-c000.snappy.parquet	10/13/2021, 1:30:25 PM		13.8 KB
part-00005-c61bb8c7-177b-4a4d-8b3e-0f5a9f019fc4-c000.snappy.parquet	10/13/2021, 1:30:25 PM		14.9 KB

Review the files produced by delta lake.

## Query Record Count

Navigate to your instance of Synapse, and then click the **Develop** icon in the left-hand navigation.

Click the + icon just above and to the right of the “**Filter resources by name**” input.

Select “**Notebook**” from the resulting drop-down.

In your new notebook, select your Apache Spark pool (if required) from the “**Attach to**” drop-down.

The screenshot shows the Microsoft Azure Synapse Analytics Notebook interface. The top navigation bar includes tabs for 'practicumsds/pract' and 'practicumsw - Azure Synapse An'. The URL is https://ms.web.azure-synapse.net/en-us/authoring/analyze/notebooks/Notebook%201?workspace=%2Fsubscriptions%2F... The main area is titled 'Develop' under 'Notebooks'. A search bar says 'Filter resources by name'. Below it, 'Notebook 1' is selected. The notebook content pane displays the following PySpark code:

```
1 %%pyspark
2 df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
3 print(df.count())
4
```

Cell [2] has a status message: "✓ 2 min 13 sec - Apache Spark session started in 1 min 28 sec 353 ms. Command executed in 43 sec 732 ms by rc...". It also shows "Job execution Succeeded" with "Spark 1 executors 4 cores". There are buttons for "View in monitoring" and "Open Spark UI". The output of the cell is "295". At the bottom of the notebook content are buttons for "+ Code" and "+ Markdown".

Run the following code to produce a count of records in the new Delta Lake:

```
%%pyspark
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
print(df.count())
```

*Note: Expected value is 295... we started with 295, deleted one and inserted one.*

## Confirm Delete Example

The screenshot shows the Azure Synapse Analytics interface. On the left, there's a navigation sidebar with icons for Home, Notebooks, Data flows, and others. The main area is titled 'Notebook 1' and shows a code cell with the following Python code:

```
1 %%pyspark
2 df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
3 display(df.where("ProductID == 730"))
4
```

Below the code, a message indicates a successful execution: [5] ✓ 11 sec - Command executed in 11 sec 23 ms by rchaper on 2:01:43 PM, 10/13/21. It also shows 'Job execution Succeeded' with 'Spark 1 executors 4 cores'. There are buttons for 'View in monitoring' and 'Open Spark UI'.

Run the following code to confirm our update example:

```
%%pyspark
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')
display(df.where("ProductID == 730"))
```

## Confirm Insert Example

The screenshot shows the same Azure Synapse Analytics interface. The notebook content remains the same as the previous example. After running the code, the interface displays a table view of the results. The table has columns: ProductID, Name, ProductNumber, and Color. One row is visible:

ProductID	Name	ProductNumber	Color
1006	New Product ABC	ABC-123	Bronze

Run the following code to confirm our update example:

```
%pyspark  
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')  
display(df.where("ProductID == 1006"))
```

## Confirm Update Example

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar is titled 'Develop' and lists 'Notebooks' (1) and 'Data flows' (2). The main area shows 'Notebook 1' with the following code:

```
1 %%pyspark  
2 df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')  
3 display(df.where("ProductID == 680"))  
4
```

[3] [✓] 11 sec - Command executed in 10 sec 642 ms by rchapler on 1:54:40 PM, 10/13/21

Job execution Succeeded Spark 1 executors 4 cores

View Table Chart Export results

ProductID	Name	ProductNumber	Color	StandardCost	ListPrice
680	HL Road Frame - Black, 58	FR-R92B-58	Black	1059.3100	999.9900

Run the following code to confirm our update example:

```
%pyspark  
df = spark.read.format('delta').load('abfss://practicumdlc@practicumdl.dfs.core.windows.net/bronze/northwind/product')  
display(df.where("ProductID == 680"))
```

## Objective: Databricks + Data Lake | Mount Data Lake

Follow the instructions in this section to establish connection to data in Azure Data Lake Storage.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Lake (with “Storage Blob Data Reader” permissions set for your Application Registration)
- Databricks (with cluster, notebook, and secret scope)

### The Logic

Navigate to Databricks and the **practicumdbn** notebook.

Add a new cell and paste the following code:

```
configs = {"fs.azure.account.auth.type": "OAuth",
           "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
           "fs.azure.account.oauth2.client.id": dbutils.secrets.get( scope="practicumdbss", key=" praticumar-clientid"),
           "fs.azure.account.oauth2.client.secret": dbutils.secrets.get( scope="practicumdbss", key="practicumar-clientsecret" ),
           "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/" + dbutils.secrets.get(
scope="practicumdbss", key=" praticumar-tenantid" ) + "/oauth2/token"}

adlsAccount = "practicumdl"
adlsContainer = "practicumdlc"
adlsFolder = ""
mountPoint = "/mnt/practicum"

if not any(mount.mountPoint == mountPoint for mount in dbutils.fs.mounts()):
    dbutils.fs.mount( source = "abfss://" + adlsContainer + "@" + adlsAccount + ".dfs.core.windows.net/" + adlsFolder,
mount_point = mountPoint, extra_configs = configs )
```

Some callouts...

- **practicumdbss** ... refers to the Secret Scope
- **myClientId** ... refers to the Key Vault secret containing the “Application (client) ID”
- **myClientSecret** ... refers to the Key Vault secret containing the “Client Secret”
- **myTenantId** ... refers to the Key Vault secret containing the “Directory (tenant) ID”
- **practicumdl** ... refers to your Data Lake
- **practicumdlc** ... refers to the Data Lake Container
- **adlsFolder** ... placeholder / syntax for inclusion of a folder (null because it is not applicable in this instance)

Run “**Cmd 1**”.

The screenshot shows a Microsoft Azure Databricks notebook interface. The top navigation bar includes tabs for 'practicumdb - Microsoft Azure' and 'practicumdbn - Databricks'. The main area is titled 'practicumdbn (Python)' and contains a single command cell labeled 'Cmd 1'. The code in the cell is as follows:

```
1 configs = {"fs.azure.account.auth.type": "OAuth",
2             "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
3             "fs.azure.account.oauth2.client.id": dbutils.secrets.get(scope="practicumdbss", key="practicumar-clientid"),
4             "fs.azure.account.oauth2.client.secret": dbutils.secrets.get(scope="practicumdbss", key="practicumar-clientsecret"),
5             "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/" +
6             dbutils.secrets.get(scope="practicumdbss", key="practicumar-tenantid") + "/oauth2/token"}
7
8 adlsAccount = "practicumdl"
9 adlsContainer = "practicumdlc"
10 adlsFolder = ""
11 mountPoint = "/mnt/practicum"
12
13 if not any(mount.mountPoint == mountPoint for mount in dbutils.fs.mounts()):
14     dbutils.fs.mount( source = "abfss://" + adlsContainer + "@" + adlsAccount + ".dfs.core.windows.net/" + adlsFolder,
15     mount_point = mountPoint, extra_configs = configs )
```

Below the code, there is a note '(1) Spark Jobs' and a timestamp 'Command took 26.98 seconds -- by rchapler@microsoft.com at 10/4/2021, 10:27:15 AM on practicumdbc'. A tooltip 'Shift+Enter to run' is visible at the bottom of the cell.

## Quick Idea...

Consider organizing mount locations by associating them with a Databricks database entity; code example:

```
%sql
CREATE DATABASE myDatabase LOCATION "/mnt/practicum/myDatabase"
```

## Confirm Success

### Proof #1 ... Sample Files at Mount Point

*Note: Running this logic requires "Storage Blob Data Reader" permissions on your data lake.*

Add a cell below "**Cmd 1**". In cell "**Cmd 2**", paste and run the following code:

```
%fs
ls "/mnt/practicum"
```

practicumdb - Microsoft Azure

practicumdbn - Databricks

Microsoft Azure | Databricks

practicumdbn (Python)

practicumdbc

Cmd 2

```
1 %fs
2 ls "mnt/practicum"
```

	path	name	size
1	dbfs:/mnt/practicum/sample1.csv	sample1.csv	502
2	dbfs:/mnt/practicum/sample2.csv	sample2.csv	56890
3	dbfs:/mnt/practicum/sample3.csv	sample3.csv	723

Showing all 3 rows.

You should see your sample files at the mount point.

## Proof #2 ... Schema and Data

In your Databricks notebook, add a new cell, then paste and run the following code:

```
df = spark.read.csv("dbfs:/mnt/practicum/sample1.csv")
df.printSchema()
df.show()
```

practicumdbn - Microsoft Azure    prakticumdbn - Databricks

**Microsoft Azure | Databricks**

practicumdbn (Python)

practicumdbc

Cmd 3

```

1 df = spark.read.csv("dbfs:/mnt/practicum/sample1.csv")
2 df.printSchema()
3 df.show()

```

▶ (2) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [c0: string, c1: string ... 11 more fields]

Month	"Average"	"2005"	"2006"	"2007"	"2008"	"2009"	"2010"	"2011"	"2012"	"2013"	"2014"	"2015"
May	0.1	0	0	1	1	0	0	0	2	0	0	0
Jun	0.5	2	1	1	0	0	1	1	2	2	0	1
Jul	0.7	5	1	1	2	0	1	3	0	2	2	1
Aug	2.3	6	3	2	4	4	4	7	8	2	2	3
Sep	3.5	6	4	7	4	2	8	5	2	5	2	5
Oct	2.0	8	0	1	3	2	5	1	5	2	3	0
Nov	0.5	3	0	0	1	1	0	1	0	1	0	1
Dec	0.0	1	0	1	0	0	0	0	0	0	0	1

Command took 5.57 seconds -- by rchapler@microsoft.com at 1/26/2021, 9:48:04 AM on practicumdbc

Shift+Enter to run [shortcuts](#)

Some observations:

- Schema ... the schema is not defined, so the resulting interpretation calls out each field as data type string.
- Resultset ... data formatting {e.g., first row as headers, double quotes, etc.} hasn't been applied.

## Objective: Databricks | Localize Mounted Data

Follow the instructions in this section to **localize data from a Data Lake mount**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Lake
- Databricks

### Why Localize?

*Databricks File System (DBFS) is a distributed file system mounted into a Databricks workspace and available on Databricks clusters. DBFS is an abstraction on top of scalable object storage and offers the following **benefits**:*

- Allows you to mount storage objects so that you can **seamlessly access data without requiring credentials**.
- Allows you to **interact with object storage using directory and file semantics instead of storage URLs**.
- Persists files to object storage, so you **won't lose data after you terminate a cluster**.

[Databricks File System \(DBFS\) — Databricks Documentation](#) | December 16, 2020

Though it isn't listed here, many customers refer to the following benefits when justifying the decision to localize data from Azure Data Lake Storage to Azure Databricks:

- Performance Gains
- Decreased Compute Cost
- Leverage Benefits of Partitioning

### The Logic

Navigate to Databricks and the **practicumdbn** notebook. Add a new cell and paste the following code:

```
import uuid
myUUID = uuid.uuid4().hex

spark.read.csv( "/mnt/practicum/sample1.csv" ).write.format( "delta" ).save( "/mnt/" + myUUID )
spark.sql( "CREATE TABLE IF NOT EXISTS myDeltaTable_fromADLS USING DELTA LOCATION '/mnt/" + myUUID + "/" )
```

The screenshot shows the Microsoft Azure Databricks interface. On the left, there's a navigation sidebar with icons for Home, Workspace, Recents, and Clusters. The main area is titled 'practicumdbn (Python)' and shows a single cell labeled 'Cmd 4' containing the Python code. The code imports the 'uuid' module, generates a UUID, reads a CSV file from a local mount, and creates a Delta table in the same location. The cell has been run, and the output shows the command took 13.51 seconds. The status bar at the bottom indicates the command was run by rchabler@microsoft.com at 1/26/2021, 10:26:28 AM on practicumdbc.

Click on **Data** in the left-hand navigation. Click on the **default** database, then the **mydeltatable\_fromadls** table.

practicumdb - Microsoft Azure    mydeltatable\_fromadls - Databricks

https://adb-2154823451042175.15.azuredatabricks.net/?o=2154823451042175#table/default/mydeltatable\_fromadls

Microsoft Azure | Databricks

Portal rchapler@microsoft.com

Data

Databases ✓

Tables

Create Table

mydeltatable\_fromadls

Table: mydeltatable\_fromadls

practicumdbc

Details History Refresh

Description:  
Created at: 2021-01-26 18:26:29  
Last modified: 2021-01-26 18:26:32  
Partition columns:  
Number of files: 1  
Size: 3.54 kB

Schema:

	col_name	data_type	comment
1	_c0	string	
2	_c1	string	
3	_c2	string	
4	_c3	string	
5	_c4	string	
6	_c5	string	
7	_c6	string	
8	c7	string	

Showing all 16 rows.

Sample Data:

	_c0	_c1	_c2	_c3	_c4	_c5
1	Month	"Average"	"2005"	"2006"	"2007"	"2008"
2	May	0.1	0	0	1	1
3	Jun	0.5	2	1	1	0
4	Jul	0.7	5	1	1	2
5	Aug	2.3	6	3	2	4
6	Sep	3.5	6	4	7	4
7	Oct	2.0	8	0	1	3
8	Nov	0.5	3	0	0	1

Showing all 9 rows.

Observation: Previously noted schema and resultset challenges remain present in the newly created table.

## Objective: Databricks | Source from APIs

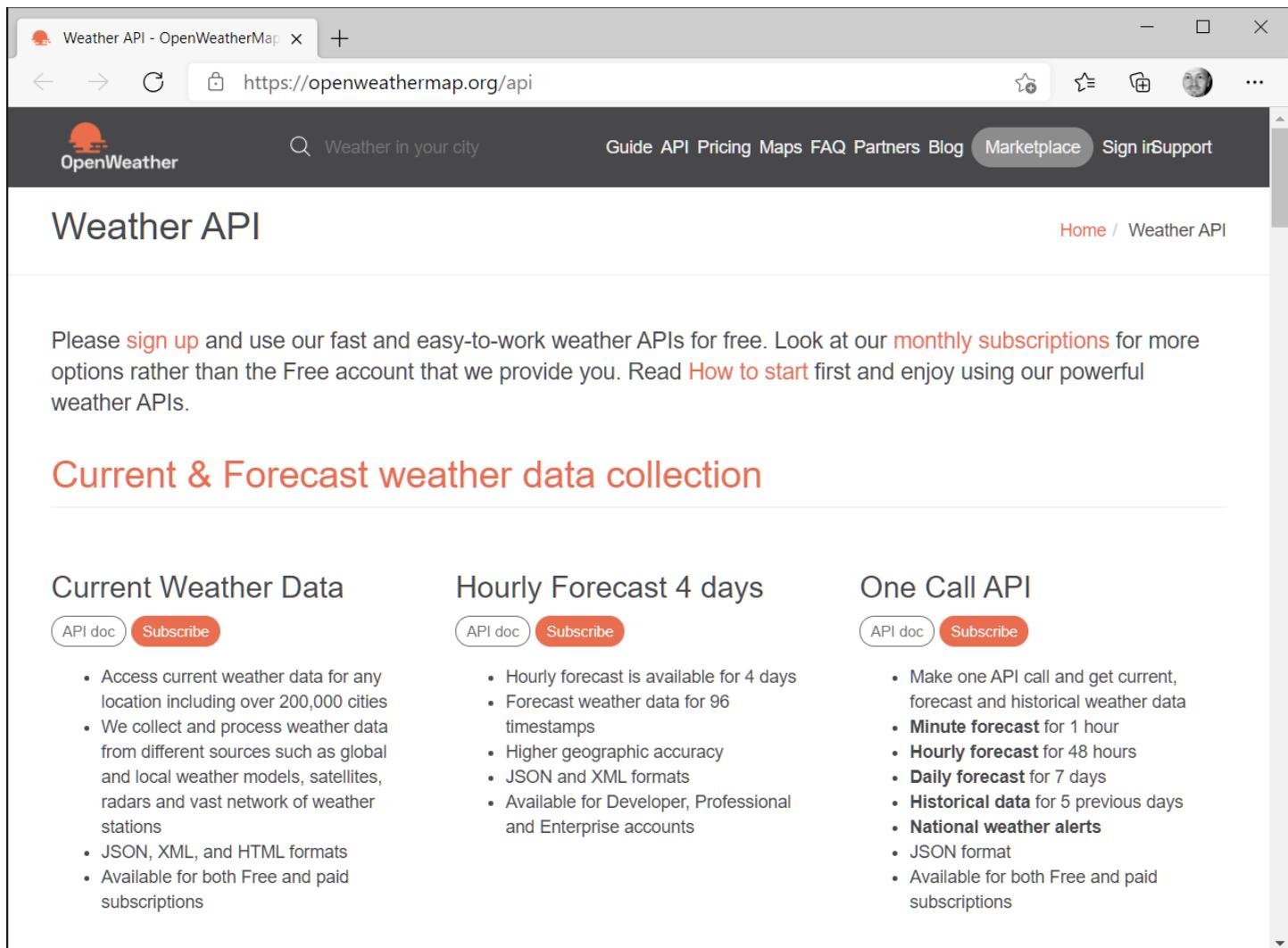
Follow the instructions in this section to **source data from a REST API**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Databricks
- Key Vault

### Stage Resources

For this exercise, we will use “**Current Weather**” data from <https://openweathermap.org/api>.



The screenshot shows a web browser window titled "Weather API - OpenWeatherMap". The URL in the address bar is <https://openweathermap.org/api>. The page header includes the OpenWeather logo, a search bar with placeholder "Weather in your city", and navigation links for Guide, API, Pricing, Maps, FAQ, Partners, Blog, Marketplace, Sign in, and Support.

## Weather API

Home / Weather API

Please [sign up](#) and use our fast and easy-to-work weather APIs for free. Look at our [monthly subscriptions](#) for more options rather than the Free account that we provide you. Read [How to start](#) first and enjoy using our powerful weather APIs.

### Current & Forecast weather data collection

**Current Weather Data**

[API doc](#) [Subscribe](#)

- Access current weather data for any location including over 200,000 cities
- We collect and process weather data from different sources such as global and local weather models, satellites, radars and vast network of weather stations
- JSON, XML, and HTML formats
- Available for both Free and paid subscriptions

**Hourly Forecast 4 days**

[API doc](#) [Subscribe](#)

- Hourly forecast is available for 4 days
- Forecast weather data for 96 timestamps
- Higher geographic accuracy
- JSON and XML formats
- Available for Developer, Professional and Enterprise accounts

**One Call API**

[API doc](#) [Subscribe](#)

- Make one API call and get current, forecast and historical weather data
- **Minute forecast** for 1 hour
- **Hourly forecast** for 48 hours
- **Daily forecast** for 7 days
- **Historical data** for 5 previous days
- **National weather alerts**
- JSON format
- Available for both Free and paid subscriptions

I chose this as my sample source because they have a Free subscription, their data is relatable, and it is easy to use.

Navigate to <https://openweathermap.org/price>.

The screenshot shows the OpenWeatherMap Pricing page. At the top, there's a navigation bar with links for Guide, API, Pricing, Maps, FAQ, Partners, Blog, Marketplace, Sign In, and Support. Below the navigation is a search bar with the placeholder "Weather in your city". The main content area has a heading "Pricing" and a sub-heading "Current weather and forecasts collection". A table displays five subscription plans:

Free	Startup 40 USD / month	Developer 180 USD / month	Professional 470 USD / month	Enterprise 2,000 USD / month
<a href="#">Get API key</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>	<a href="#">Subscribe</a>
60 calls/minute <b>1,000,000 calls/month</b>	600 calls/minute 10,000,000 calls/month	3,000 calls/minute 100,000,000 calls/month	30,000 calls/minute 1,000,000,000 calls/month	200,000 calls/minute 5,000,000,000 calls/month
<b>Current Weather</b>	Current Weather	Current Weather	Current Weather	Current Weather

Click the “Get API key” button in the **Free** section and complete the “Create a New Account” process.

When you’re successfully created an account, making an API call for “**Current Weather**” data is as simple as:

```
api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}
```

The screenshot shows a browser window with the URL `api.openweathermap.org/data/2.5/weather?q=redmond&appid=...`. The page title is "Not secure". The content of the page is a JSON response:

```
{"coord":{"lon":-122.1215,"lat":47.674}, "weather":[{"id":804,"main":"Clouds","description":"overcast clouds","icon":"04d"}], "base":"stations", "main": {"temp":279.53,"feels_like":277.4,"temp_min":278.15,"temp_max":280.93,"pressure":1007,"humidity":65,"sea_level":1007,"grnd_level":1006}, "visibility":10000,"wind": {"speed":0.27,"deg":241}, "clouds": {"all":100}, "dt":1611870000, "sys": {"type":3,"id":2010401,"country":"US","sunrise":1611848385,"sunset":1611882185}, "timezone":-28800, "id":5808079, "name": "Redmond", "cod":200 }
```

## Key Vault

Add your API Key to your instance of Key Vault.

The screenshot shows the Microsoft Azure portal interface for creating a secret. The page title is "Create a secret - Microsoft Azure". The URL in the address bar is <https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/91e9fddc-...>. The top navigation bar includes "Microsoft Azure (Preview)", a search bar, and user information for "rchapler@microsoft.com MICROSOFT (MICROSOFT.ONM...)".

The main content area is titled "Create a secret". It contains the following form fields:

- Upload options: Manual
- Name: myOpenWeatherAPIKey
- Value: (redacted)
- Content type (optional): (empty)
- Set activation date?: (unchecked)
- Set expiration date?: (unchecked)
- Enabled?: Yes (selected)

At the bottom left is a blue "Create" button.

## The Logic

Navigate to Databricks and the **practicumdbn** notebook. Add a new cell and paste the following code:

```
import requests
response = requests.get( 'http://api.openweathermap.org/data/2.5/weather?q=redmond&appid=' + dbutils.secrets.get(
scope="practicumdbss", key="myOpenWeatherAPIKey" ) )

df = spark.read.option( 'multiline', "true" ).json( sc.parallelize( [response.text] ) )
```

Run the cell.

The screenshot shows a Microsoft Azure Databricks notebook titled "practicumdbn (Python)". The notebook interface includes a left sidebar with icons for Home, Workspace, Recents, Data, Clusters, Jobs, Models, and Search. The main area displays a code editor with the following Python script:

```
1 import requests
2 response = requests.get( 'http://api.openweathermap.org/data/2.5/weather?q=redmond&appid=' + dbutils.secrets.get(
3     scope="practicumdbss", key="myOpenWeatherAPIKey" ) )
4 df = spark.read.option( 'multiline', "true" ).json( sc.parallelize( [response.text] ) )
```

Below the code editor, a schema browser shows the structure of the DataFrame `df`:

- (1) Spark Jobs
- df: pyspark.sql.dataframe.DataFrame
  - base: string
  - clouds: struct
    - all: long
    - cod: long
  - coord: struct
    - lat: double
    - lon: double
  - dt: long
  - id: long
  - main: struct
    - feels\_like: double
    - grnd\_level: long
    - humidity: long
    - pressure: long
    - sea\_level: long

Another Python example (including POST, headers, and body):

```
import json
stringJSON = '{"data": [{"1212":0,"1227":0,..."ZZ9":0}]}'
theJSON = json.loads(stringJSON)
response = requests.post(
    'http://eedfb2cf-deb1-4260-971d-d7a6c308d9b2.eastus.azurecontainer.io/score',
    headers={'CONTENT-TYPE': 'application/json'},
    json = theJSON
)
response.json()
```

## Objective: Databricks | Batch Upsert Data

Follow the instructions in this section to **provide for recurring upsert of data**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Lake
- Databricks

### CREATE TABLE

Navigate to Databricks and the **practicumdbn** notebook.

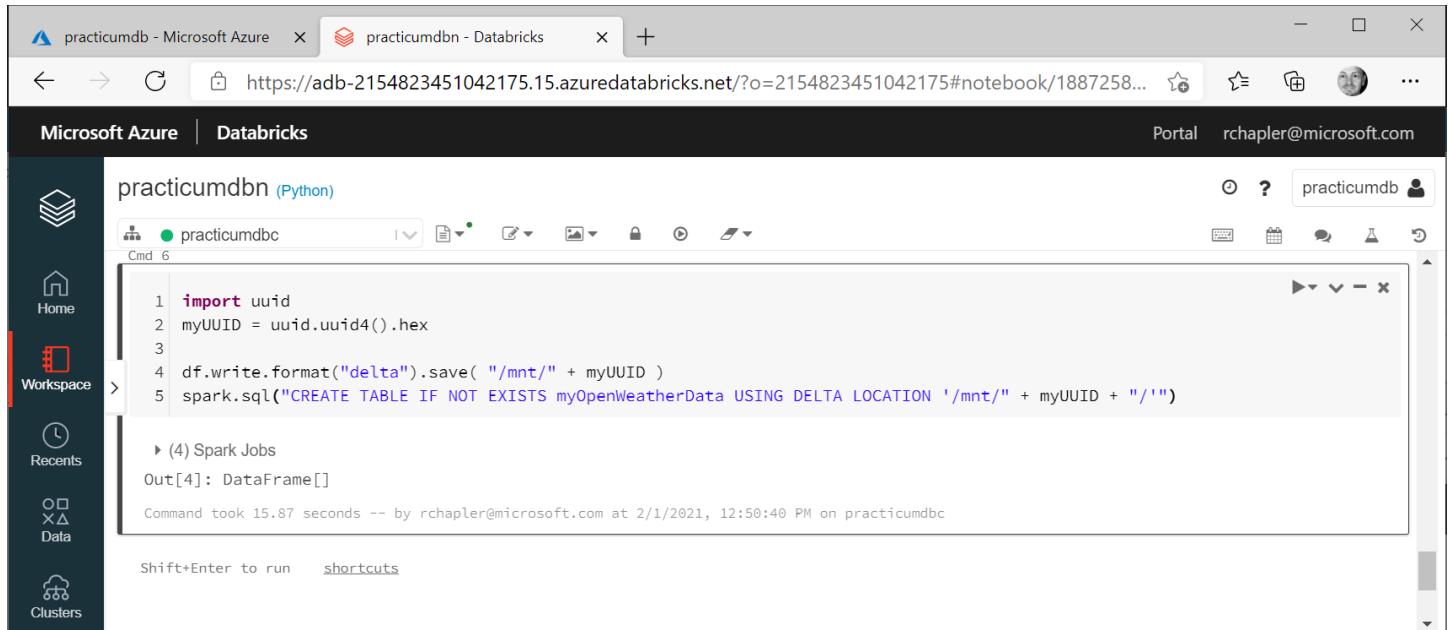
Add a new cell and paste the following code:

```
import uuid
myUUID = uuid.uuid4().hex

df.write.format("delta").save( "/mnt/" + myUUID )
spark.sql("CREATE TABLE IF NOT EXISTS myOpenWeatherData USING DELTA LOCATION '/mnt/" + myUUID + "'")
```

Notes:

- We are using dataframe **df** created in [Objective #3 ... The Logic](#)
- We are using format **delta** to provide for future UPSERT operations.



The screenshot shows the Microsoft Azure Databricks interface. On the left is a sidebar with icons for Home, Workspace (which is selected), Recents, Data, and Clusters. The main area is titled 'practicumdbn (Python)'. It shows a code editor with the following Python code:

```
import uuid
myUUID = uuid.uuid4().hex
df.write.format("delta").save( "/mnt/" + myUUID )
spark.sql("CREATE TABLE IF NOT EXISTS myOpenWeatherData USING DELTA LOCATION '/mnt/" + myUUID + "'")
```

Below the code editor is a results pane showing:

- (4) Spark Jobs
- Out[4]: DataFrame[]

A status message at the bottom says: "Command took 15.87 seconds -- by rchapler@microsoft.com at 2/1/2021, 12:50:40 PM on practicumdbc". At the bottom of the code editor, it says "Shift+Enter to run shortcuts".

### MERGE INTO

Navigate to Databricks and the **practicumdbn** notebook. Add a new cell and paste the following code:

```
import requests
import uuid

response = requests.get( 'http://api.openweathermap.org/data/2.5/weather?q=redmond&appid=' + dbutils.secrets.get(
scope="practicumdbss", key="myOpenWeatherAPIKey" ) )
df = spark.read.option( 'multiline', "true" ).json( sc.parallelize( [response.text] ) )

myUUID = uuid.uuid4().hex
```

```

df.write.format("delta").save( "/mnt/" + myUUID )
spark.sql("DROP TABLE myOpenWeatherData_updates")
spark.sql("CREATE TABLE myOpenWeatherData_updates USING DELTA LOCATION '/mnt/" + myUUID + "'")

mySQL = "MERGE INTO myOpenWeatherData existing USING myOpenWeatherData_updates updates ON existing.id = updates.id AND
existing.dt = updates.dt"
mySQL += " WHEN MATCHED THEN UPDATE SET existing.main = updates.main"
mySQL += " WHEN NOT MATCHED THEN INSERT (id, dt, main) VALUES (id, dt, main)"

print( mySQL )

spark.sql( mySQL )

```

Run the cell.

The screenshot shows the Azure Databricks interface. On the left is the sidebar with icons for Home, Workspace, Recents, Data, Clusters, Jobs, Models, and Search. The main area has tabs for 'practicumdb - Microsoft Azure' and 'practicumdbn - Databricks'. Below the tabs is a URL bar with the address https://adb-2154823451042175.15.azure.databricks.net/?o=2154823451042175#notebook/1887258... and a user profile for rchapler@microsoft.com. The central workspace contains a code editor window titled 'practicumdbn (Python)' with a tab 'practicumdbc'. The code in the editor is identical to the one provided at the top of the page. Below the code editor is a results pane showing the output of the executed code. The output includes 14 Spark Jobs, a DataFrame named 'df' with 12 fields, and a SQL command for merging data into the 'myOpenWeatherData' table. The command is:

```

MERGE INTO myOpenWeatherData existing USING myOpenWeatherData_updates updates ON existing.id = updates.id AND
existing.dt = updates.dt
WHEN MATCHED THEN UPDATE SET existing.main = updates.main
WHEN NOT MATCHED THEN INSERT (id, dt, main) VALUES (id, dt, main)

```

At the bottom of the results pane, it says 'Command took 16.50 seconds -- by rchapler@microsoft.com at 2/1/2021, 1:00:44 PM on practicumdbc'. A note at the bottom of the code editor says 'Shift+Enter to run [shortcuts](#)'.

## Confirm Success

Navigate to Databricks and the **practicumdbn** notebook. Add a new cell and paste the following code:

```
%sql
SELECT * FROM myOpenWeatherData
```

Run the cell.

The screenshot shows the Microsoft Azure Databricks workspace interface. On the left is a sidebar with icons for Home, Workspace, Recents, Data, Clusters, Jobs, Models, and Search. The main area has two tabs: "practicumdb - Microsoft Azure" and "practicumdb - Databricks". The "practicumdb - Databricks" tab is active, displaying a Python notebook titled "practicumdbn (Python)". The notebook contains the following code:

```
MERGE INTO myOpenWeatherData existing USING myOpenWeatherData_updates updates ON existing.id = updates.id AND existing.dt = updates.dt WHEN MATCHED THEN UPDATE SET existing.main = updates.main WHEN NOT MATCHED THEN INSERT (id, dt, main) VALUES (id, dt, main)
Out[8]: DataFrame[]

Command took 16.50 seconds -- by rchapler@microsoft.com at 2/1/2021, 1:00:44 PM on practicumdbc
```

Below the notebook is a command editor window titled "Cmd 8" containing:

```
1 %sql
2 SELECT * FROM myOpenWeatherData
```

Underneath the command editor is a table titled "(3) Spark Jobs" showing the results of the SQL query:

	base	clouds	cod	coord	dt	id	main
1	stations	{"all": 100}	200	{"lat": 47.674, "lon": -122.1215}	1612211690	5808079	{"feels_like": 280.49, "grnd_level": 1004, "h": 282.59, "temp_min": 281.48}
2	null	null	null	null	1612212650	5808079	{"feels_like": 279.62, "grnd_level": 1005, "h": 282.59, "temp_min": 281.48}

Text below the table indicates "Showing all 2 rows." and a command history entry: "Command took 1.18 seconds -- by rchapler@microsoft.com at 2/1/2021, 1:22:13 PM on practicumdbc". At the bottom of the command editor is a note: "Shift+Enter to run [shortcuts](#)".

Note that the row created using MERGE INTO has a limited number of populated cells since only some of the possible fields were included in WHEN NOT MATCHED THEN INSERT (id, dt, main)... we can, of course, expand this to include all fields about which we care.

## Objective: Data Explorer | Query from On-Prem

Follow the instructions in this section to **query from an on-prem client**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Azure CLI
- Data Explorer
- Postman

### Bearer Token

In a Command Prompt window (with [Azure CLI](#) installed), execute the following command: `az login`

Use the resulting browser window to provide credentials.

In a Command Prompt window (with [Azure CLI](#) installed), execute the following command: `az account get-access-token --resource https://practicumdec.westus2.kusto.windows.net --query accessToken --output tsv`

Capture the returned token for use later in this section.

### API Request

Open a new request in Postman.

The screenshot shows the Postman application window. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Help', 'Home', 'Workspaces', 'Reports', 'Explore', and a search bar. On the right side of the header are 'Sign In' and 'Create Account' buttons. Below the header is a yellow banner with the text 'Working locally in Scratch Pad. Switch to a Workspace'. The main area shows a request card with 'POST https://practicumdec.westus2.kusto.windows.net/v2/rest/query'. The 'Authorization' tab is selected, showing a dropdown set to 'Bearer Token' with a token value 'eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1Ni...'. Other tabs like 'Params', 'Headers (13)', 'Body', 'Pre-request Script', 'Tests', and 'Settings' are visible. At the bottom, there are buttons for 'Save', 'Send', 'Cookies', and status information: 'Status: 200 OK', 'Time: 288 ms', 'Size: 8.31 KB', and 'Save Response'.

Select **POST** from the drop-down. Enter a hostname (with the name you used for your cluster); example:

`https://practicumdec.westus2.kusto.windows.net/v2/rest/query`

On the **Authorization** tab, select “**Bearer Token**” from the drop-down and paste your previously copied token in the resulting interface.

The screenshot shows the Postman application window with a similar layout to the first one. The 'Headers' tab is selected in the request card, displaying the following header values:  
Accept: application/json  
Accept-Encoding: deflate  
Content-Type: application/json; charset=utf-8  
Host: help.kusto.windows.net

On the **Headers** tab, enter the following items:

Accept application/json

Accept-Encoding	deflate
Content-Type	application/json; charset=utf-8
Host	help.kusto.windows.net

The screenshot shows the Postman application window. At the top, there are tabs for Home, Workspaces, Reports, and Explore, along with a search bar and user authentication links. Below the header, a yellow banner says "Working locally in Scratch Pad. Switch to a Workspace". The main area shows a list of recent requests and the current request details. The request method is set to POST, and the URL is https://practicumdec.westus2.kusto.windows.net/v2/rest/query. The "Body" tab is selected, showing a JSON payload:

```
1 {
2     "db": "practicumded",
3     "csl": "StormEvents | take 5"
4 }
```

At the bottom right of the interface, status information is displayed: Status: 200 OK, Time: 288 ms, Size: 8.31 KB, and a "Save Response" button.

On the **Body** tab, paste the following:

```
{
  "db": "practicumded",
  "csl": "StormEvents | take 5"
}
```

Click the **Send** button.

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Help', 'Home', 'Workspaces', 'Reports', 'Explore', and a search bar. On the right of the top bar are 'Sign In' and 'Create Account' buttons. Below the navigation is a yellow banner with the text 'Working locally in Scratch Pad. Switch to a Workspace'. The main area shows a request card for a POST method to 'https://practicumdec.westus2.kusto.windows.net/v2/rest/query'. The response status is 200 OK, time 244 ms, size 8.29 KB. The response body is displayed in a JSON editor with the 'Pretty' tab selected, showing the following schema:

```
1  {
2      "FrameType": "DataSetHeader",
3      "IsProgressive": false,
4      "Version": "v2.0"
5  },
6  {
7      "FrameType": "DataTable",
8      "TableId": 0,
9      "TableKind": "QueryProperties",
10     "TableName": "@ExtendedProperties",
11     "Columns": [
12         {
13             "ColumnName": "TableId",
14             "ColumnType": "int"
15         },
16         {
17             "ColumnName": "Key",
18             "ColumnType": "string"
19         },
20         {
21             "ColumnName": "Value",
22             "ColumnType": "dynamic"
23         }
24     ]
}
```

You should see JSON with both schema and data included.

# Analysis

Objective(s) in this section describe methods for analyzing data.

## Objective: Metrics Advisor + Data Explorer | Onboard and Investigate

**Use Case:** Company XYZ wants to detect data anomalies for Metric 123. When significant anomalies are detected, the solution should support investigation.

Follow the instructions in this section to **on-board data feeds and investigate findings**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Explorer
- Metrics Advisor

Navigate to your instance of Metrics Advisor.

The screenshot shows the Microsoft Azure portal interface. The left sidebar contains navigation links for Microsoft Azure (Preview), Dashboard, Microsoft.CognitiveServicesMetricsAdvisor-20210917090741, and practicumma. The main content area is titled "practicumma | Quick start" and displays three numbered steps: 1. Monitor your first metrics using the web-based workspace, 2. Use APIs to customize your own solution, and 3. Learn more about the service. Each step includes a brief description and a "Go to your workspace" link. The URL in the browser bar is https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions/.../Microsoft.CognitiveServicesMetricsAdvisor-20210917090741/practicumma.

Click the “Go to your workspace” link.

The screenshot shows a web browser window with the title bar "practicumma - Microsoft Azure" and the tab "Metrics Advisor". The URL in the address bar is "https://metricsadvisor.azurewebsites.net". The main content area has a background graphic of blue and white lines radiating from a central point. On the left, there is a large text block: "Protect your organization's growth". Below this, a smaller text block describes Metrics Advisor's function: "Metrics Advisor monitors the performance of your organization's growth engines, from sales revenue to manufacturing operations. It helps you quickly identify and fix problems through a powerful combination of monitoring in near-real time, adapting models to your scenario, offering granular analysis with diagnostics, and alerting." To the right, there is a section titled "Welcome to Metrics Advisor!" with the sub-instruction: "Select a subscription and a workspace to get started or go to the Azure Portal to change subscription and workspace. You can switch subscriptions and workspaces at any time. When switching directory, you will be requested to sign-in again." Below this are three dropdown menus: "Directory" set to "Microsoft", "Subscription" set to "rchapler", and "Workspace" set to "practicumma". There is also a link "Create a Metrics Advisor resource". At the bottom are two buttons: "Get started" (in blue) and "Learn more".

On the “Welcome to Metrics Advisor!” page, enter values for the following items:

Directory	Select your directory
Subscription	Select your subscription
Workspace	Select your instance of Metrics Advisor

Click the “Get started” button.

Screenshot of the Microsoft Azure Metrics Advisor Data feeds Onboarding page.

The page title is "Metrics Advisor" and the URL is "https://metricsadvisor.azurewebsites.net/data-feed".

The left sidebar shows the following navigation items:

- Collapse menu
- Onboarding
  - Add data feed
- Monitor & Diagnostic
- Data feeds **(selected)**
- Incident hub
- Metrics graph
- Settings
- API keys
- Hooks
- Credential entity
- Help
  - Documentation
  - Public community

The main content area is titled "Data feeds" and displays three steps:

- Step 1: Onboard time-series data**
  - Icon: A stack of three boxes with a gear and a checkmark.
  - Description: "Ingest your time-series data from various data sources supported".
- Step 2: Tune configuration & subscribe anomaly alerts**
  - Icon: A laptop displaying a chart with a gear and a warning sign.
  - Description: "Fine tune Detection configurations to better serve real-world scenarios" and "Create a hook and subscribe real-time anomaly alerts".
- Step 3: Diagnose incidents**
  - Icon: A smartphone with a magnifying glass over a dashboard.
  - Description: "Identify key contributors with dimension tree" and "Chase down correlations with metrics graph".

A blue button at the bottom center says "Onboard my first data feed".

Click the “Onboard my first data feed” button.

The screenshot shows the 'Add data feed' page in the Microsoft Azure Metrics Advisor. The left sidebar has a blue header with the Metrics Advisor logo and a 'Collapse menu' button. Below the header, the sidebar lists various sections: Onboarding, Add data feed (which is selected and highlighted in blue), Monitor & Diagnostic, Data feeds, Incident hub, Metrics graph, Settings, API keys, Hooks, Credential entity, Help, Documentation, and Public community. The main content area has a white background. At the top, there's a 'Connection settings' section with a note about granting IP access. Below it are dropdowns for 'Source type' (set to 'Azure Data Explorer (Kusto)'), 'Granularity' (set to 'Daily'), and 'Ingest data since (UTC)' (set to '2007-01-01'). There's also a 'Import' and 'Start a tour' link. The next section is 'Authentication type', with a dropdown set to 'Managed Identity'. Under 'Connection string', there's a text input field containing the URL 'Data Source=https://practicumdec.westus2.kusto.windows.net;Initial Catalog=practicumdec'. Below this is a 'Tips' section with a note about specifying the connection string to access Azure Data Explorer. A 'Sample' section provides a template: 'Data Source=<URI Server>;Initial Catalog=<Database>'. There's a 'Learn more' link. The final section is 'Query', which includes an 'Important' note about running queries continuously and aggregating data by dimensions. It also includes a 'Sample' Kusto query for 'StormEvents':  

```
StormEvents
| summarize sum(DamageProperty) by startofday(StartTime), State
| where StartTime >= todatetime("@IntervalStart") and StartTime < todatetime("@IntervalEnd")
```

At the bottom right of the main content area is a blue 'Load data' button.

On the “Add data feed” page, enter values for the following items:

<b>Source Type</b>	Select “ <b>Azure Data Explorer (Kusto)</b> ”
<b>Granularity</b>	Confirm default selection, <b>Daily</b>
<b>Ingest Data Since (UTC)</b>	Select a date appropriate for your data set
<b>Authentication Type</b>	Select “ <b>Managed Identity</b> ”
<b>Connection String</b>	Enter in form “Data Source=https://{{ADX Cluster}}.{region}.kusto.windows.net;Initial Catalog={{ADX Database}}
<b>Query</b>	Create and test a query in your database, then parameterize; example below:  <pre>StormEvents   summarize sum(DamageProperty) by startofday(StartTime), State   where StartTime &gt;= todatetime("@IntervalStart") and StartTime &lt; todatetime("@IntervalEnd")</pre>

Click the “Load Data” button.

The screenshot shows the Microsoft Azure Metrics Advisor interface for adding a data feed. On the left, a sidebar includes links for 'Credential entity', 'Help', 'Documentation', and 'Public community'. The main area has tabs for 'Query' and 'Schema configuration'. The 'Query' tab displays a DAX-like query:

```
StormEvents  
| summarize sum(DamageProperty) by startofday(StartTime), State  
| where StartTime >= todatetime("@IntervalStart") and StartTime < todatetime("@IntervalEnd")
```

The 'Schema configuration' tab shows sample data and schema mapping. The sample data table has columns: State, StartTime, and sum\_DamageProperty. The data rows are:

State	StartTime	sum_DamageProperty
NORTH CAROLINA	2007-01-01T00:00:00Z	0
WISCONSIN	2007-01-01T00:00:00Z	0
NEW YORK	2007-01-01T00:00:00Z	20000
ALASKA	2007-01-01T00:00:00Z	0
DELAWARE	2007-01-01T00:00:00Z	0
OKLAHOMA	2007-01-01T00:00:00Z	775000
INDIANA	2007-01-01T00:00:00Z	110000

The schema configuration table maps columns from the sample data to dimensions and measures. The columns are:

Column name	Display name	Column type	Select:
State	State	String	<input type="radio"/> Timestamp <input checked="" type="radio"/> Dimensions <input type="radio"/> Measure
StartTime	Event Date	String	<input checked="" type="radio"/> Timestamp <input type="radio"/> Dimensions <input type="radio"/> Measure
sum_DamageProperty	Count of Events	String	<input type="radio"/> Timestamp <input type="radio"/> Dimensions <input checked="" type="radio"/> Measure

A blue 'Verify schema' button is located at the bottom right of the schema configuration section.

Click the “Verify Schema” button.

practicumma - Microsoft Azure Metrics Advisor https://metricsadvisor.azurewebsites.net/add-data-feed

**Advanced settings** Help

**Automatic roll-up settings** Help

My data has already rolled up and the dimension value is represented by  NULL or Empty (Default)

I need the service to roll up my data by calculating  None and represent it by  Set roll-up columns

I do not need to perform root cause analysis into dimensions for my metrics.

**Ingestion options** Help

Ingestion time offset  hours Enter Max concurrency... Stop retrying after  hours Minimum retrying interval  hours

**Data feed not available alert** Help

Alert hooks  Select Grace period  hours Snooze  hours consecutive "data feed not available" alerts

By applying hooks, you can receive alerts. You can [create and manage hooks](#).

**Misc**

Missing points filling for anomaly detection model Help

Smart filling  Fill previous  Fill custom value:  0  No filling

Action link template Help

Tips: You can use these placeholders in the URL template %datafeed, %metric, %timestamp, %detect\_config, %tagset.

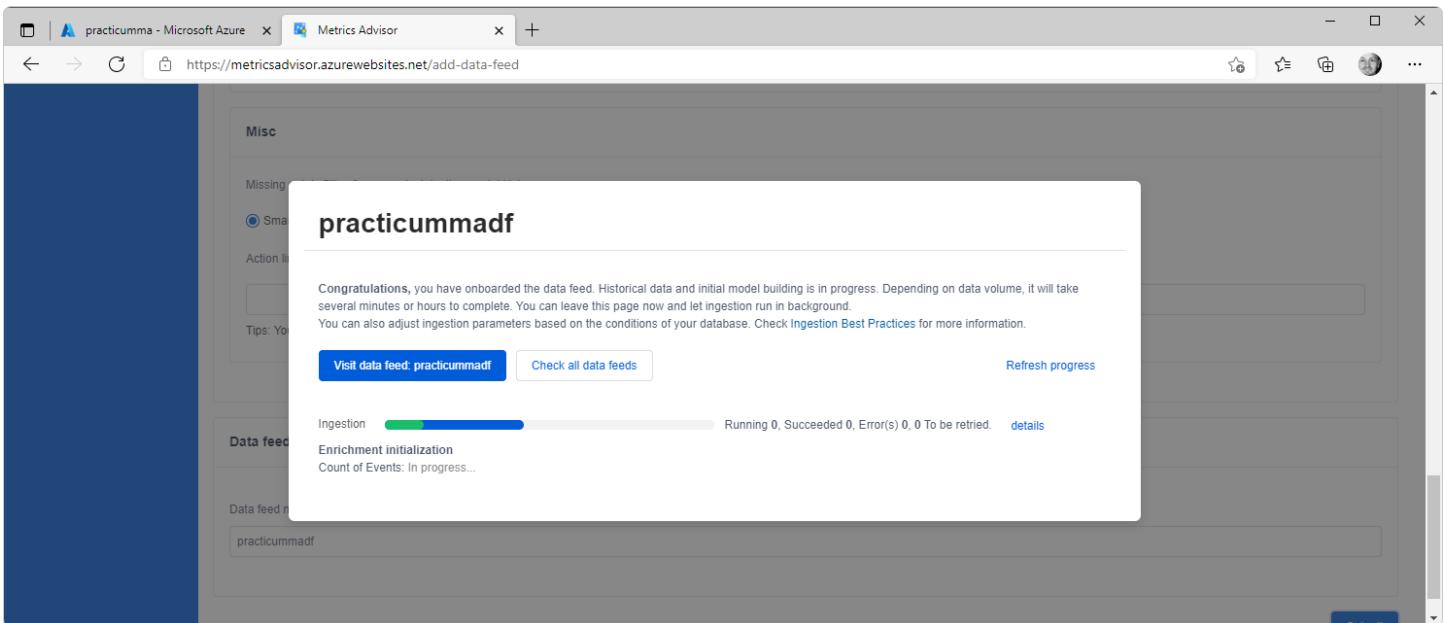
**Data feed name**

Data feed name  practicummadf

**Submit**

No changes are required to default values in “**Advanced Settings**”.

Enter a “**Data Feed Name**” and then click the Submit button.



You will receive a “**Congratulations...**” message.

Click the “**Visit data feed...**” button.

The screenshot shows the Microsoft Azure Metrics Advisor interface for the 'practicummadf' data feed. The left sidebar includes 'Collapse menu', 'Onboarding', 'Add data feed', 'Monitor & Diagnostic' (selected), 'Data feeds' (selected), 'Incident hub', 'Metrics graph', 'Settings', and 'API keys'. The main content area shows the 'practicummadf' feed with a progress bar indicating 'Running 0, Succeeded 50, Error(s) 0, 0 To be retried'. It lists a single metric: 'Count of Events' with ID 'b2172aec-5fd0-4a0a-bb8d-5da630c4e217', description 'Metric calculated by current metrics.', series count '0', and start time '2007-01-01T00:00:00Z'. A button '+ Add a new metric' is available. On the right, the 'Data feed Information' section shows details: Name 'practicummadf', ID 'dbf98b41-f6f3-4fcc-8fdd-328b6be1a311', Created time '2021-09-21T15:06:28Z', and Dimensions 'State'. Buttons for 'Backfill', 'Refresh Progress' (highlighted in blue), 'Delete', 'Export', and 'Start a tour' are at the top right.

Monitor progress on the data feed page.

Click into the Metric Name link to see analysis.

practicumma - Microsoft Azure Metrics Advisor https://metricsadvisor.azurewebsites.net/metric/b2172aec-5fd0-4a0a-bb8d-5da630c4e217 Metric ID: b2172aec-5fd0-4a0a-bb8d-5da630c4e217

practicumadaf / Count of Events Configure Preset Event ?

**Detection configuration** (1)

Default

**Basic information**

ID: ae582f95-0854-490e-84e9-1ac673f07a43  
Name: Default

**Metric-level configuration**

Smart detection (✓)

Sensitivity: 79

Value: Out of boundary is an anomaly.

Do not report anomaly until 100 % of latest 1 points are detected as anomalies. (✓)

+ Series group configs: 0 Specific series configs: 0 More details in Advanced Configuration

Save

2007-01-01 00:00:00 - 2007-04-06 00:00:00 Layout 4x4 Choose series

Browse series Incidents Data view Please wait for your data ingestion and detection process to complete, click refresh button to check latest status.

[ sum\_DamageProperty ] State= TENNESSEE [ sum\_DamageProperty ] State= GEORGIA

[ sum\_DamageProperty ] State= TEXAS [ sum\_DamageProperty ] State= VERMONT

# Application

Objective(s) in this section describe methods for using data assets in user-facing applications, workflows, and artificial intelligence.

## Objective: SQL + Power Apps + AI Builder | Capture Data

**Use Case:** Company XYZ wants to produce a customer-facing app that can be used to capture images and metadata for an insurance claim {e.g., a vehicle with damage to a headlight}.

Follow the instructions in this section to **build an application that employs artificial intelligence with minimal code**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Azure SQL
- PowerApps

### Create Target Table

Navigate to your instance of SQL.

practicumsd (practicumsds/practicumsd) | Query editor (preview)

Welcome to SQL Database Query Editor

SQL server authentication

Login \* rchapler

Password \*

Active Directory authentication

Continue as rchapler@microsoft.com

OR

Cannot open server 'practicumsds' requested by the login. Client with IP address [REDACTED] is not allowed to access the server. To enable access, use the Azure Portal or run sp\_set\_firewall\_rule on the master database to create a firewall rule for this IP address or address range. It may take up to five minutes for this change to take effect.

Whitelist IP [REDACTED] on server practicumsds

OK

Navigate to “Query editor...” and login. Whitelist your IP address as appropriate.

The screenshot shows the Microsoft Azure Query editor (preview) interface. The left sidebar lists various services: Overview, Activity log, Tags, Diagnose and solve problems, Quick start, and Query editor (preview). The Query editor (preview) option is selected. The main area displays a query window titled "Query 1". The query code is:

```
1 CREATE TABLE dbo.myTable( Id INT NOT NULL IDENTITY(1,1) PRIMARY KEY, Name VARCHAR(64), Picture IMAGE )
```

The results pane shows the message: "Query succeeded: Affected rows: 0". A status bar at the bottom indicates "Query succeeded | 0s".

Execute the following T-SQL:

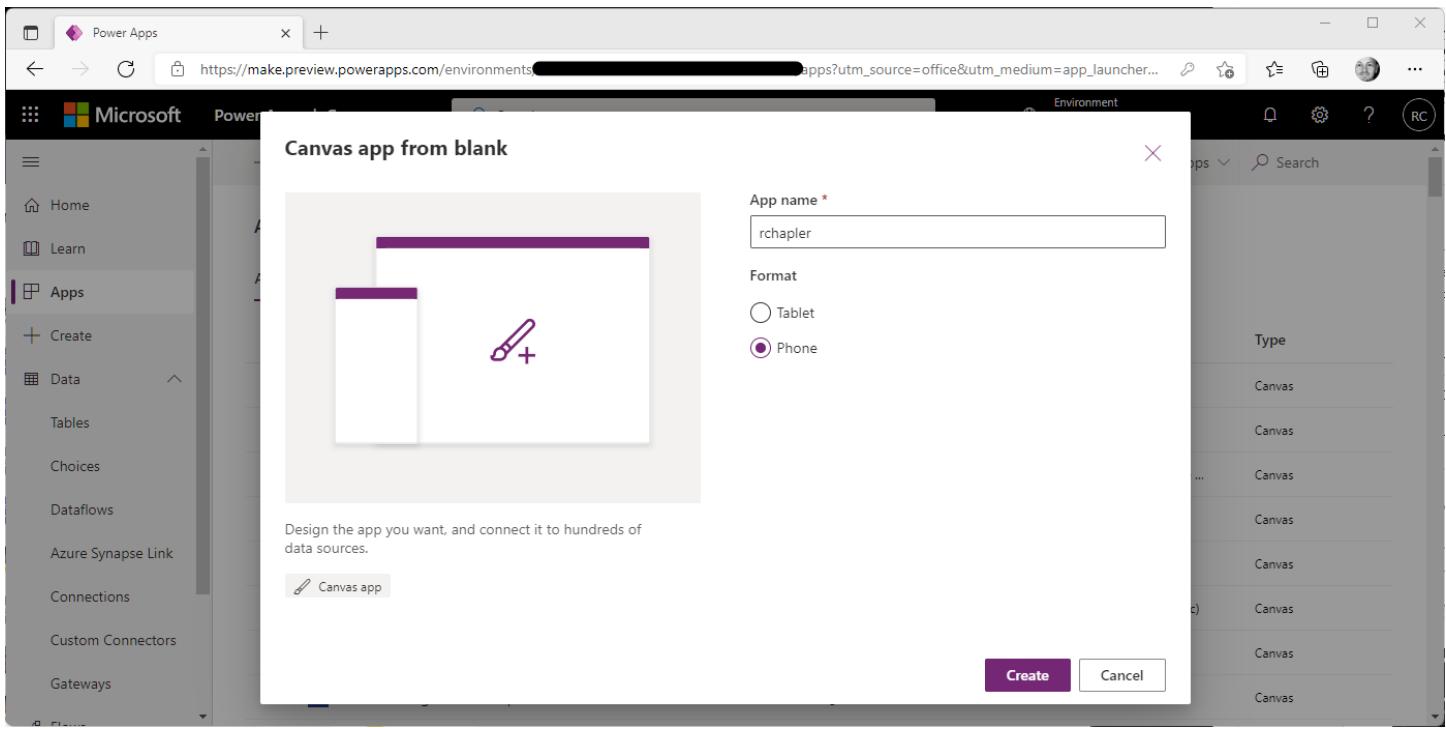
```
CREATE TABLE dbo.myTable( Id INT NOT NULL IDENTITY(1,1) PRIMARY KEY, Name VARCHAR(64), Picture IMAGE )
```

## Create Canvas App

Navigate to Power Apps (<https://make.preview.Power Apps.com/>).

Click on **Apps** in the left-hand navigation.

At the top of the “**Apps**” page, click “+ New app” and select **Canvas** from the drop-down.



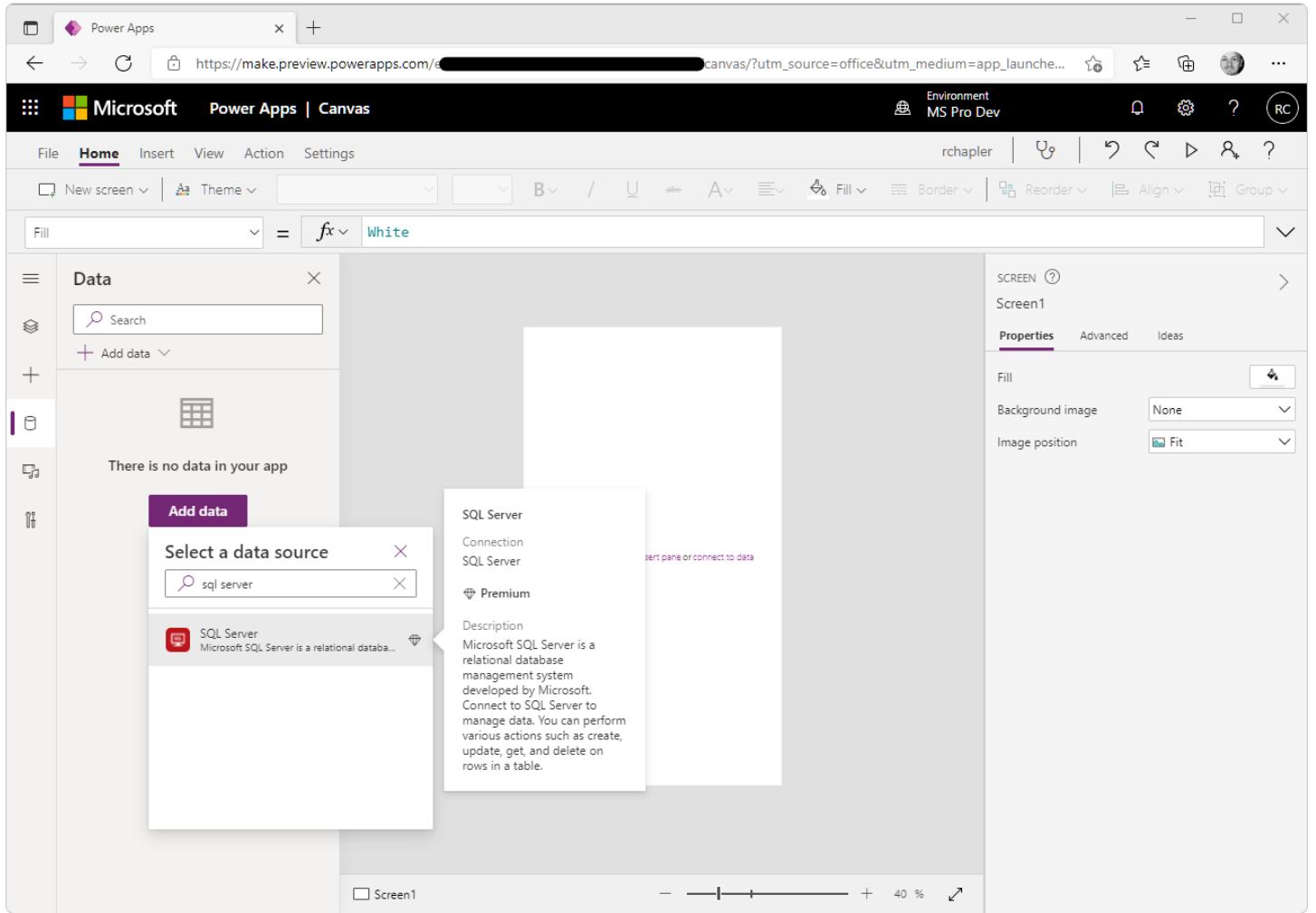
On the “**Canvas app from blank**” pop-up, enter the following items:

<b>App Name</b>	Self-Explanatory
<b>Format</b>	Select the <b>Phone</b> radio button

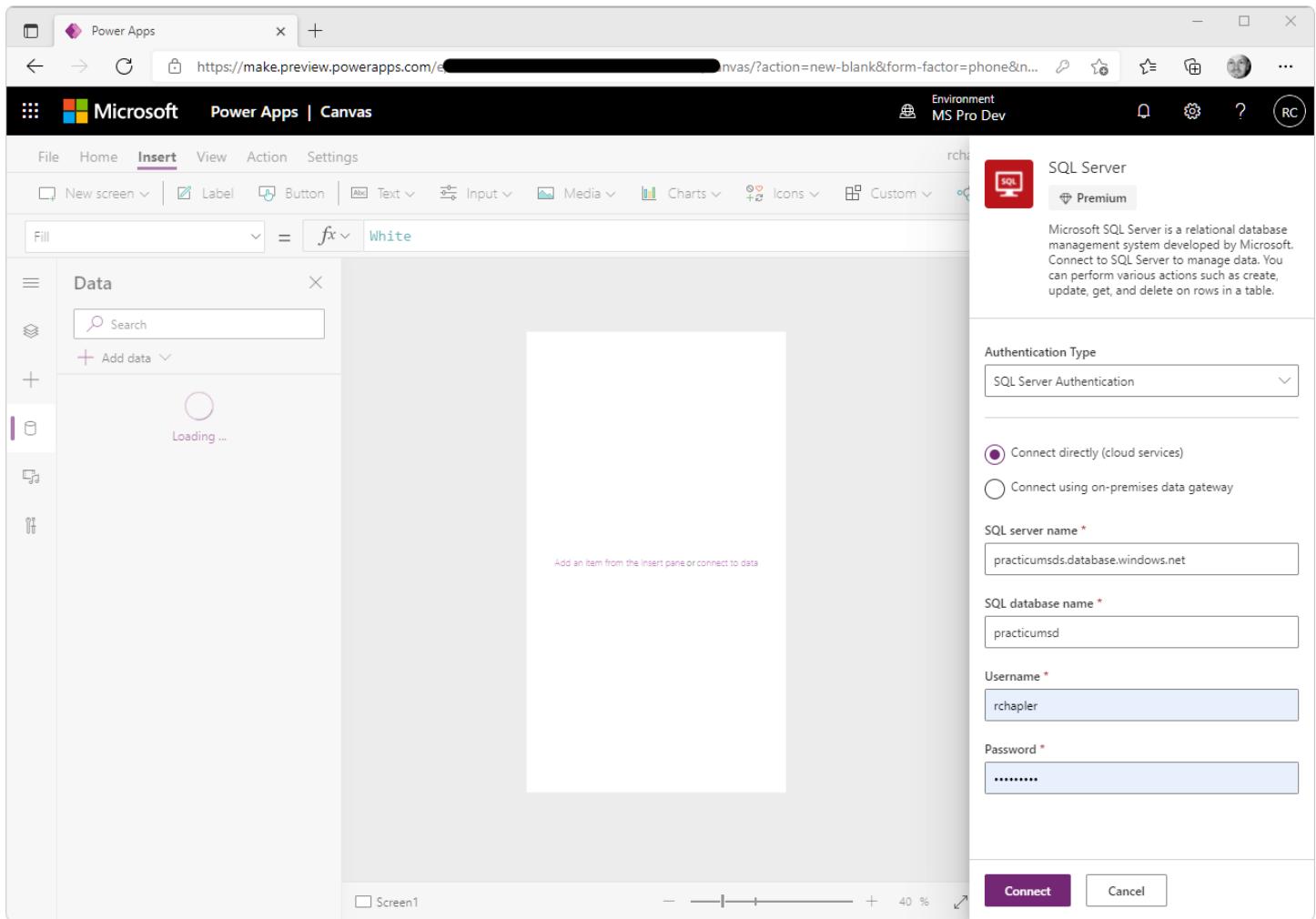
Click the **Create** button.

## Add Data

Click on **Data** in the left-hand navigation. Click on the “**Add data**” button.



In “Select a data source”, enter “sql server” in the search input and select “SQL Server” from the results.  
Click “+ Add a connection” on the resulting drop-down.



On the “SQL Server” pop-out, enter the following items:

**Authentication Type** SQL Server Authentication

**SQL Server Name** Enter the values employed during database server creation ([Appendix | SQL](#))

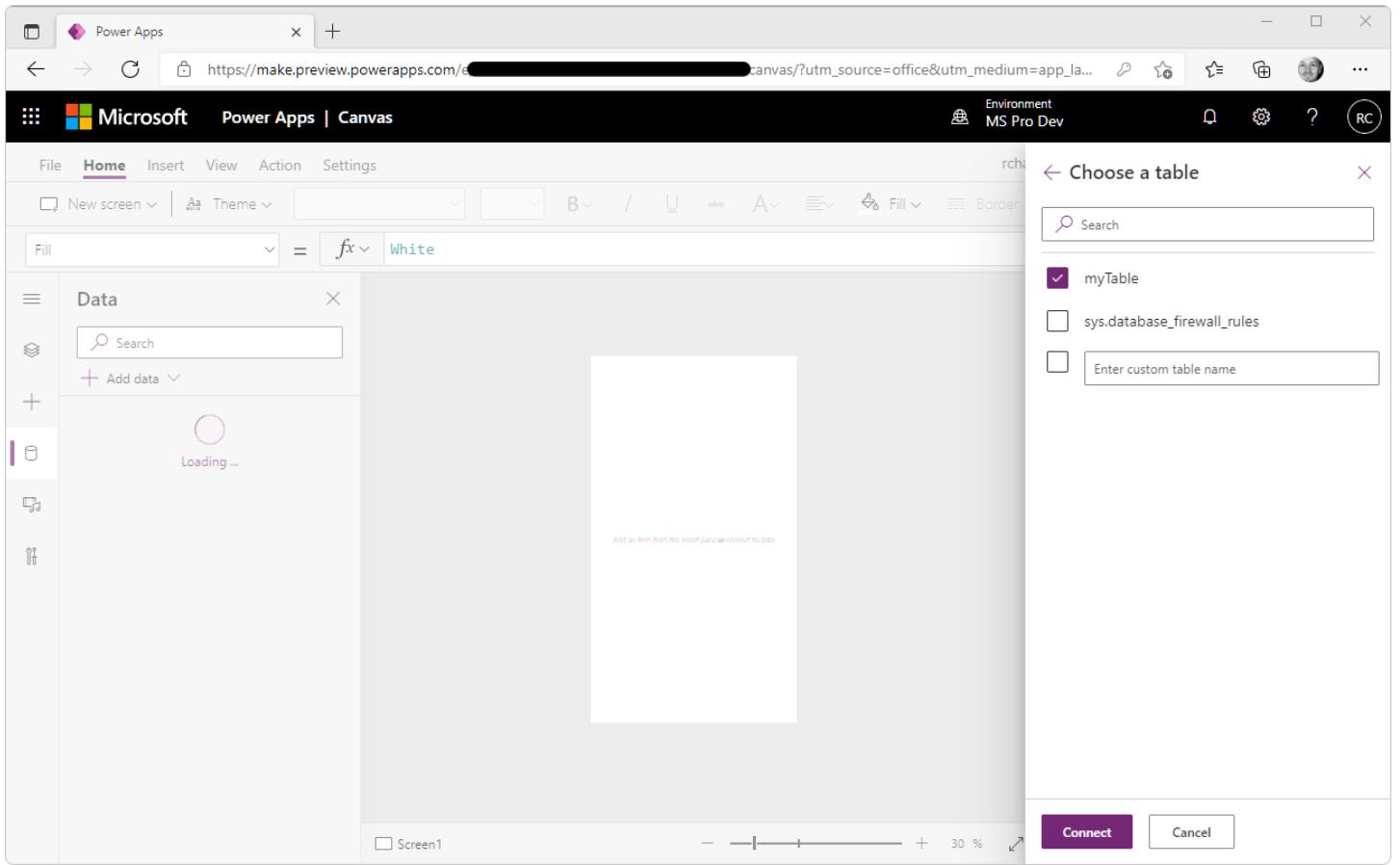
**SQL Database Name**

**Username**

**Password**

Click the **Connect** button.

*Note: Resolve error messages “We weren’t able to add this connection...” and “Client with IP address #.#.#.# is not allowed...”, by adding your IP address to the Azure SQL Server firewall.*



On the “Choose a table” pop-out, check the table created in [Create Target Table](#).

Click the **Connect** button.

## Insert Object Detector

Click the “Add an item from the Insert pane” link.

On the resulting, left-hand **Insert** menu, expand “AI Builder” and select “Object detector”.

On the resulting, right-hand pop-out, click “+ New model”.

## Create AI Model

The screenshot shows the Microsoft Power Apps | AI Builder interface. On the left, there's a navigation sidebar with options like Home, Learn, Apps, Create, Data, Flows, Chatbots, AI Builder (which is selected and highlighted in purple), Models, and Solutions. The main content area has a heading "Enhance your business with AI" and a sub-section "Refine a model for your business needs". It displays five cards: "Category Classification" (document icon), "Entity Extraction" (document icon), "Form Processing" (document icon), "Object Detection" (camera icon), and "Prediction" (checkbox icon). Below the "Object Detection" card, there's a note: "Recognize and count things in images." At the top right, there are environment settings (Environment MS Pro Dev) and a help icon.

On the new “Power Apps | AI Builder”, “Enhance your business with AI” page, click the “Object Detection” button.

Name the AI model and then click the **Create** button.

The screenshot shows the "Select your model's domain" page. On the left, a vertical list of steps is shown: "Select domain" (checked), "Choose objects", "Add images", "Tag images", and "Model summary". The main content area has a heading "Select your model's domain" and a note: "When models focus on specific types of objects, they can be more accurate. If you don't see the right option, select **Common objects**. [Learn more](#)". Below this is a section with three buttons: "Common objects" (highlighted in pink), "Objects on retail shelves", and "Brand logos". To the right is a large image of a field with several combine harvesters. Three of the harvesters have purple bounding boxes around them, each labeled "Combine". At the bottom left is a "Next" button. On the right side, there's a sidebar with "Quick tips" and "Get help or send feedback" sections.

On the “Select your model's domain” page, click the “Common objects” and then the “Next” button.

Select from database

### Choose objects for your model to detect

You can add them manually or select from your database. [Learn more](#)

Object names

Headlights

+ Add new object

?

Quick tips

Select from database instead  
Save time and use objects in Dataverse, rather than creating them manually.

Switch to database

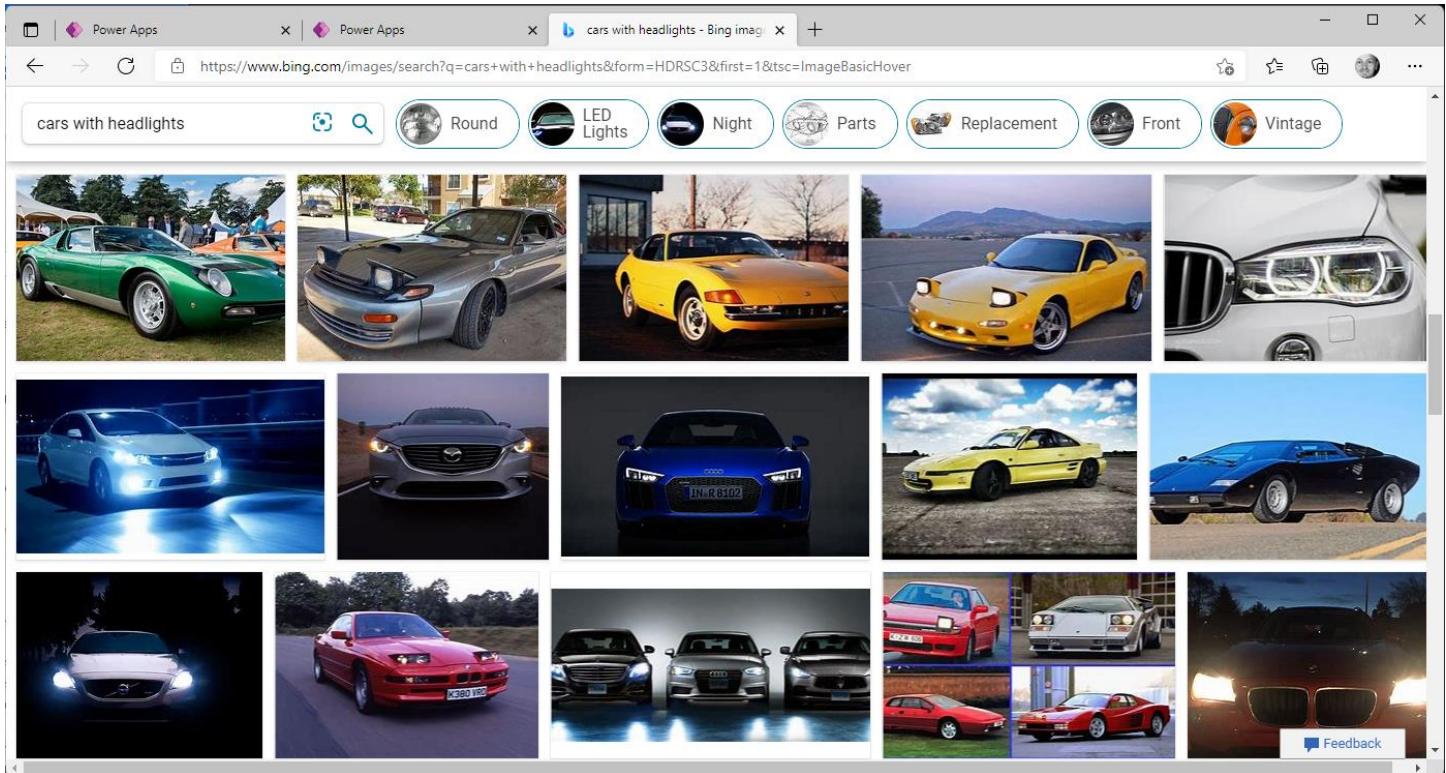
Get help or send feedback  
Get the answers you need, or tell us about your experiences.

Get help

Back Next 1 object name selected

On the “Choose objects for your model to detect” page, click “+ Add new object”, enter an object name and then click the “Next” button.

On the “Add example images for your objects” page, we will add image files that will be used to train the AI model. For this exercise, we can pull images of cars (with headlights) from a search engine.



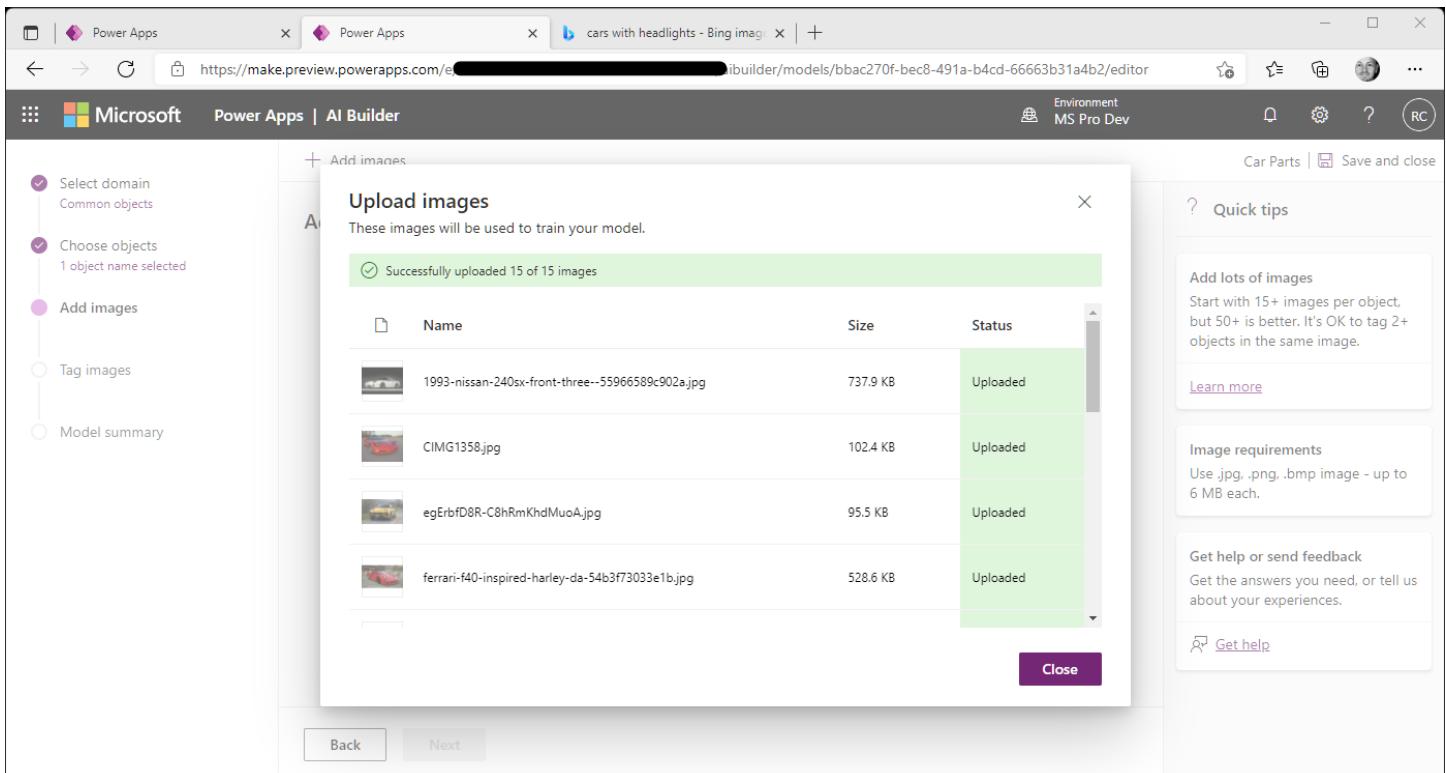
For each of the required 15 images, download an image file to a temporary folder on the Windows desktop.

The screenshot shows the Microsoft Power Apps AI Builder interface. On the left, a sidebar lists steps: 'Select domain' (checked), 'Common objects', 'Choose objects' (checked, showing '1 object name selected'), 'Add images' (highlighted in pink), 'Tag images', and 'Model summary'. The main area has a title 'Add example images for your objects' with a sub-instruction 'Find images containing your objects'. It says 'You need to add at least 15 images for each object you want to detect.' A large 'Add images' button is centered. To the right, there's a 'Quick tips' section with 'Add lots of images' (start with 15+ images per object, up to 50+), 'Image requirements' (use .jpg, .png, .bmp images up to 6 MB each), and 'Get help or send feedback' (link to Get help). Navigation buttons 'Back' and 'Next' are at the bottom.

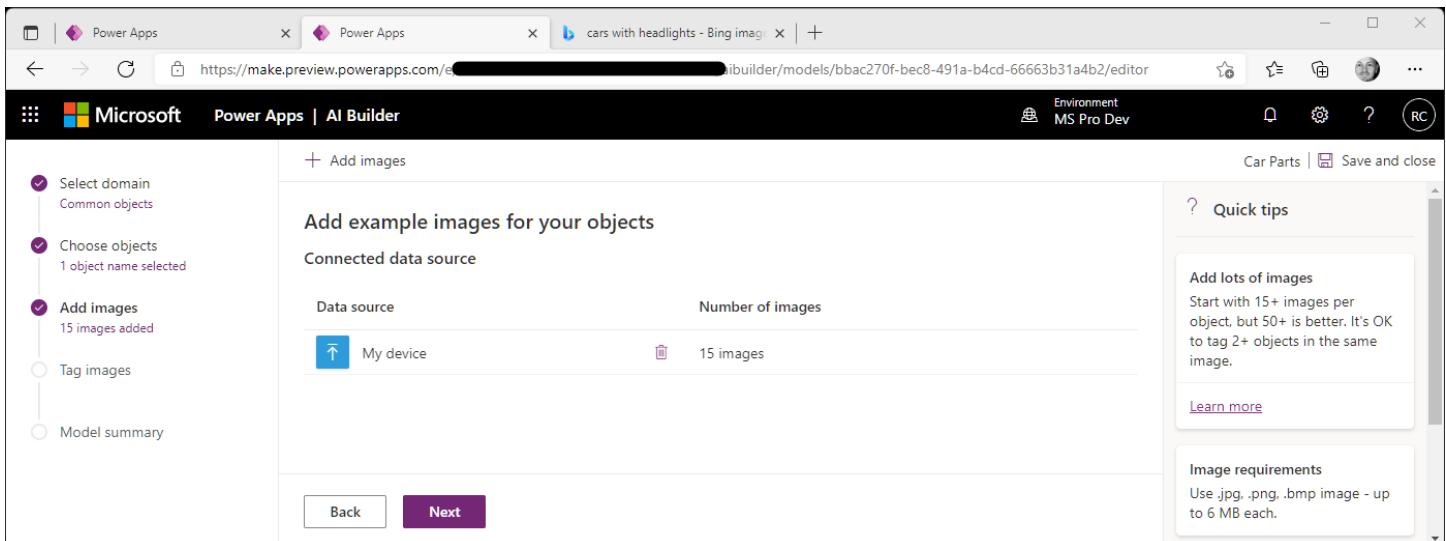
On the “Add example images for your objects” page, click the “Add images” button.

This screenshot shows the 'Add images' pop-out dialog from the previous step. It has a title 'Add images' and a sub-section 'Select a data source'. It lists three options: 'Upload from local storage' (with an upward arrow icon), 'Azure Blob Storage' (with a cloud icon and a 'Premium' badge), and 'SharePoint' (with a SharePoint icon). The 'Upload from local storage' option is the active choice. The background of the dialog is semi-transparent, showing the 'Add example images for your objects' page from the previous screenshot.

On the resulting “Add images” pop-out, click “Upload from local storage”. Select the Example Images downloaded to your desktop.



Review the items in the “Upload images” pop-up, then click the “Upload 15 images” button. Click **Close**.



Back on the “Add example images for your desktop” page, confirm “Number of images” shows 15 or more images, then click **Next**.

Power Apps | Microsoft Power Apps | AI Builder

Environment: MS Pro Dev

Car Parts | Save and close

You will need to tag at least 15 images per object to continue.

Tag the objects in your images

All  

Model summary

Tagging requirements  
You must tag at least 15 images for each object. Tagging more than 50 images for each object could yield better results. You can have multiple tags per image.  
0 Headlights

Want to add more images?  
Adding more images may increase model performance.  
+ Add images

Get help or send feedback  
Get the answers you need, or tell us about your experiences.  
Get help

Select the first image on the “Tag the objects in your images” page.

Power Apps | Microsoft Power Apps | AI Builder

https://make.preview.powerapps.com/e...

Don't use image Clear Discard changes Details Done tagging

Choose object

Search

Headlights

Tagging progress  
Image 1 of 15  
You must tag at least 15 images for each object. Tagging more than 50 images for each object could yield better results. You can have multiple tags per image.

Tags applied  
0 Headlights

Drag a rectangle around headlights in the image, then click the Headlights button on the “Choose object” pop-up.  
Repeat for all taggable items, click > to move to through images, and then click the “Done tagging” button in the upper-right.

Power Apps | AI Builder

cars with headlights - Bing image

Environment: MS Pro Dev

Car Parts | Save and close

Tag the objects in your images

All

15 Headlights

Tagging requirements  
You must tag at least 15 images for each object. Tagging more than 50 images for each object could yield better results. You can have multiple tags per image.

Want to add more images?  
Adding more images may increase model performance.

+ Add images

Get help or send feedback  
Get the answers you need, or tell us about your experiences.

Get help

Select domain  
Common objects

Choose objects  
1 object name selected

Add images  
15 images added

Tag images  
15 images tagged

Model summary

Back Next

Click the **Next** button.

Screenshot of the Microsoft Power Apps AI Builder "Model summary" page.

The left sidebar shows completed steps:

- Select domain: Common objects
- Choose objects: 1 object name selected
- Add images: 15 images added
- Tag images: 15 images tagged
- Model summary

The main area displays the Model summary details:

**Overview**

Model type	Object Detection	Owner	Rich Chapler	Object type	Common objects
------------	------------------	-------	--------------	-------------	----------------

**Image sources**

Data source	Number of images
My device	15 images

**Information to extract**

Object	Tags
Headlights	15

Buttons at the bottom: Back, Train

Right sidebar:

- Car Parts | Save and close
- Quick tips
- What is training? Training prepares your model to be used, based on the information you provided.
- What's next? Once trained, you will be able to evaluate your model performance results and publish your model for use in Power Apps. At any time, you can add and tag more images.
- Get help or send feedback Get the answers you need, or tell us about your experiences.
- Get help

Review the “Model Summary” and then click the **Train** button.

Screenshot of the Microsoft Power Apps AI Builder "Model summary" page during training.

The left sidebar shows completed steps:

- Select domain: Common objects
- Choose objects: 1 object name selected
- Add images: 15 images added
- Tag images: 15 images tagged
- Model summary

The main area displays the Model summary details and a training status message:

**Model summary**

Review your model's details

**Overview**

Model type: Object Detection

**Image sources**

Data source: My device, Number of images: 15 images

**Information to extract**

Object	Tags
Headlights	15

Your model is training

This may take a while. You can close this window and come back later.

Buttons at the bottom: Back, Training, Go to models

Right sidebar:

- Car Parts | Save and close
- Quick tips
- What is training? Training prepares your model to be used, based on the information you provided.
- What's next? Once trained, you will be able to evaluate your model performance results and publish your model for use in Power Apps. At any time, you can add and tag more images.
- Get help or send feedback Get the answers you need, or tell us about your experiences.
- Get help

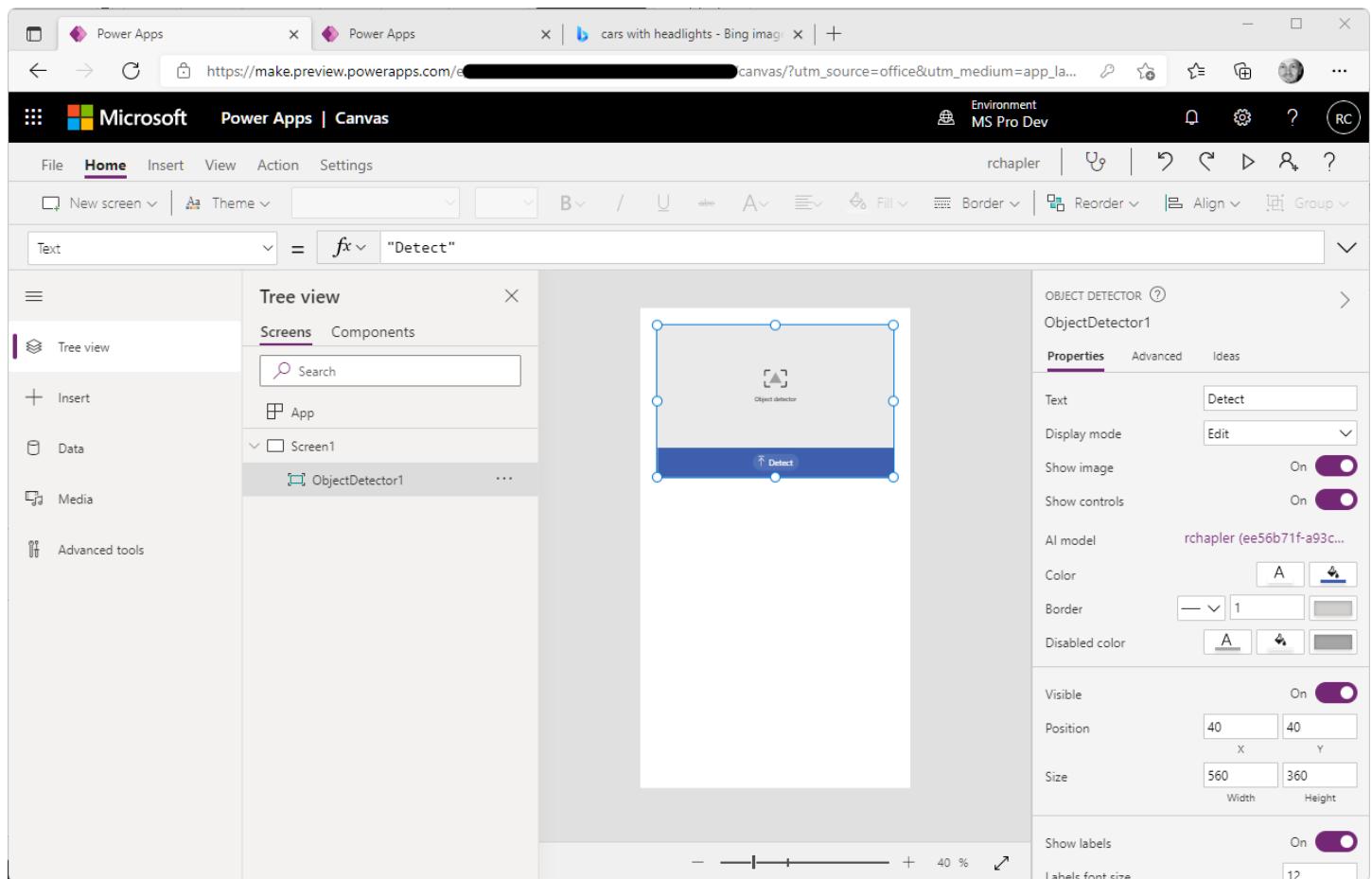
Click the “Go to models” button. After Status changes to **Trained**, navigate to the model.

The screenshot shows the Microsoft Power Apps AI Builder interface. On the left, a sidebar menu includes Home, Learn, Apps, Create, Data, Flows, Chatbots, AI Builder, Build, Models (which is selected), and Solutions. The main content area displays a 'Models > Car Parts' page for 'Object Detection'. A large circular progress bar indicates a 'Performance' of 46%. Below it, a callout box suggests improving the model by providing more training data. Two buttons at the bottom are 'Publish' (in purple) and 'Quick test'. To the right, there's a 'Details' section with 'Training date' (10:29:01 AM), 'Object type' (Common objects), and 'Number of objects' (1). A 'See more' link is also present. On the far right, a panel titled 'How your model is used' lists 'Power Automate' and 'Power Apps', both noting that the model isn't published.

In the snip above, we see that **Performance** {i.e., model performance expectation %} of 46 for the example images included in training.  
In the future, you can improve model quality by including more images and re-training the model.

Click the **Publish** button.

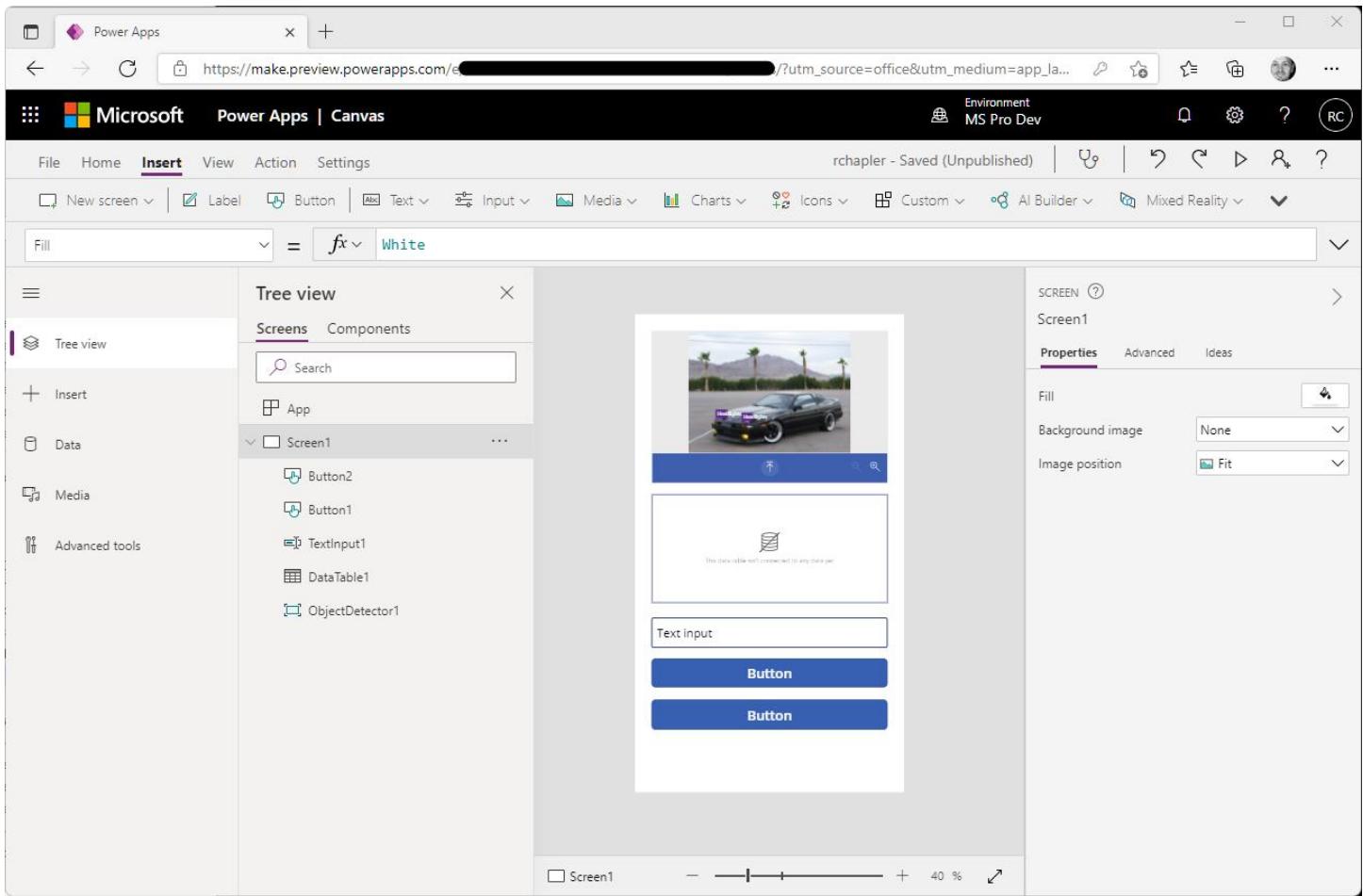
Return to Power Apps and complete the ObjectDetector1 component by selecting your newly created model.



*Note: Consider resizing the component to improve visibility.*

## Insert / Configure Controls

Return by closing Preview Mode.

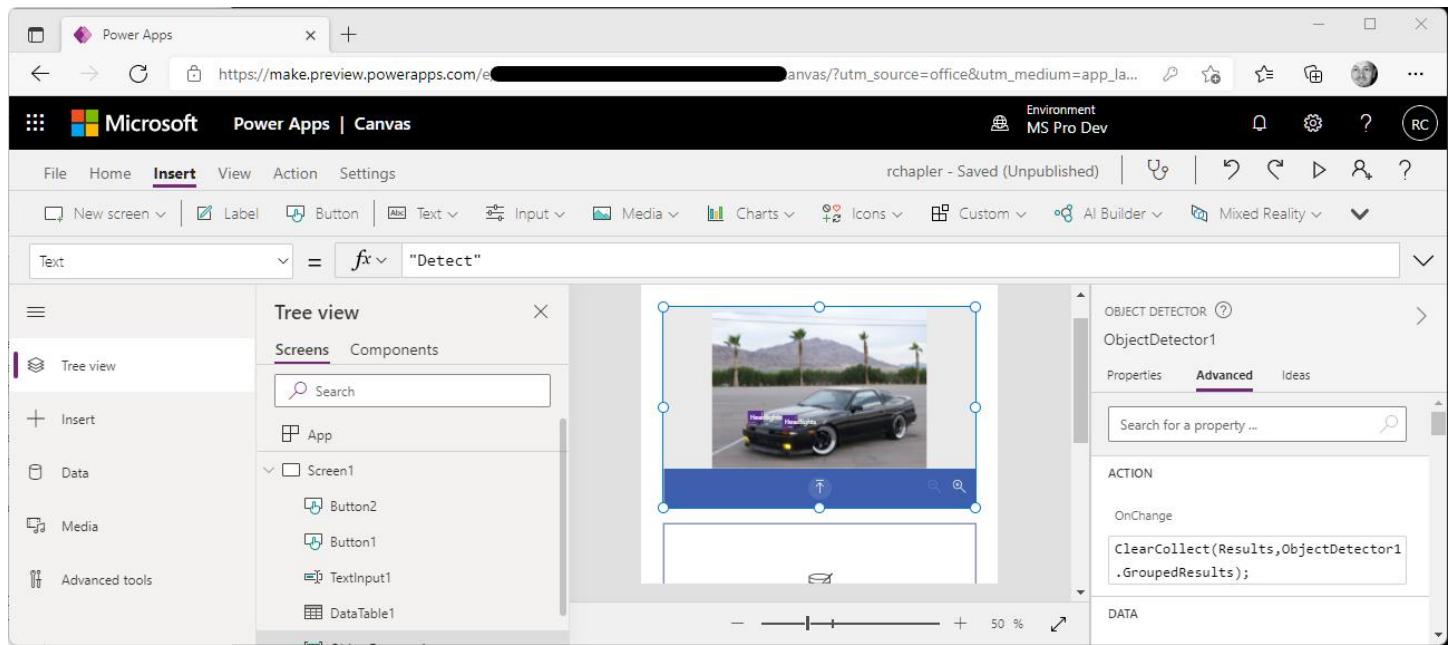


Insert the following controls:

- **Data Table** ... users will use this control to view the results generated by AI Builder
- **Text Input** ... users will use this control to enter comments about the analyzed image
- “Submit” **Button** ... users will use this control to save the image and comment to Azure SQL
- “Reset” **Button** ... users will use this control to clear previously entered values

You might note that I employed formatting {e.g., control width, border size, color, etc.} to enhance usability.

## ObjectDetector1

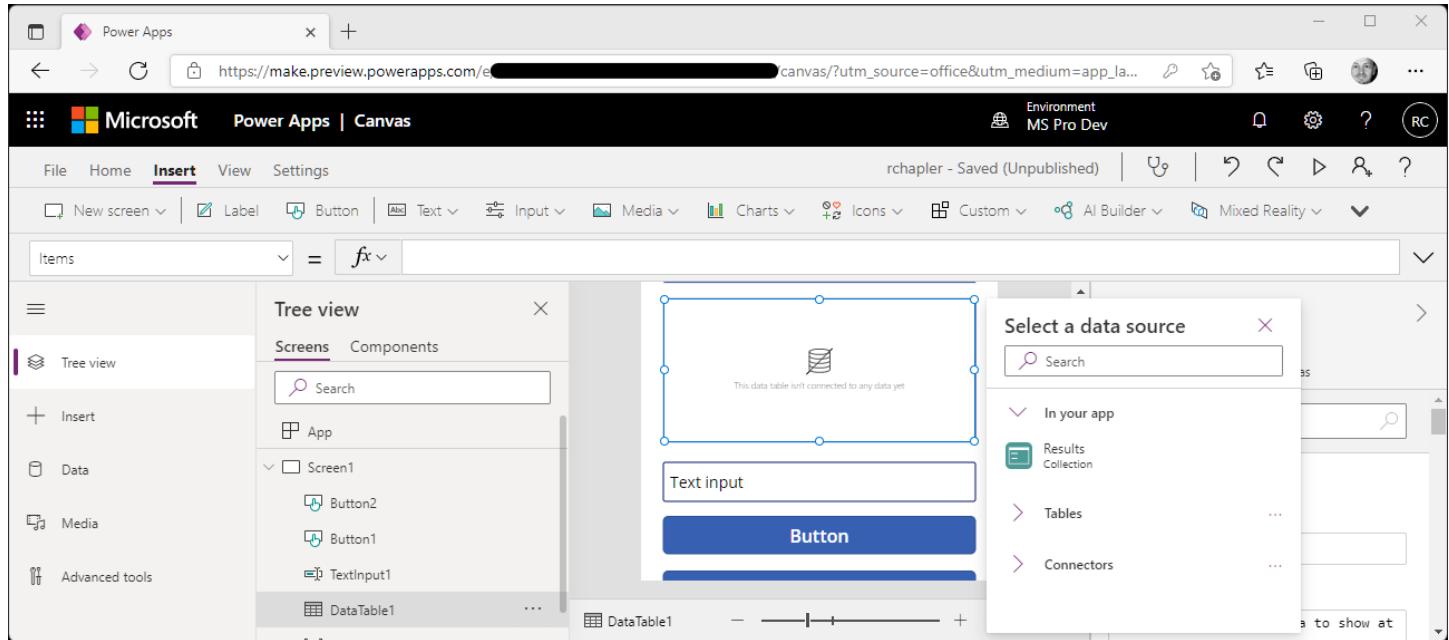


Click on **ObjectDetector1** in the left-hand **Tree View**. On the resulting right-hand pop-out, click the **Advanced** tab.

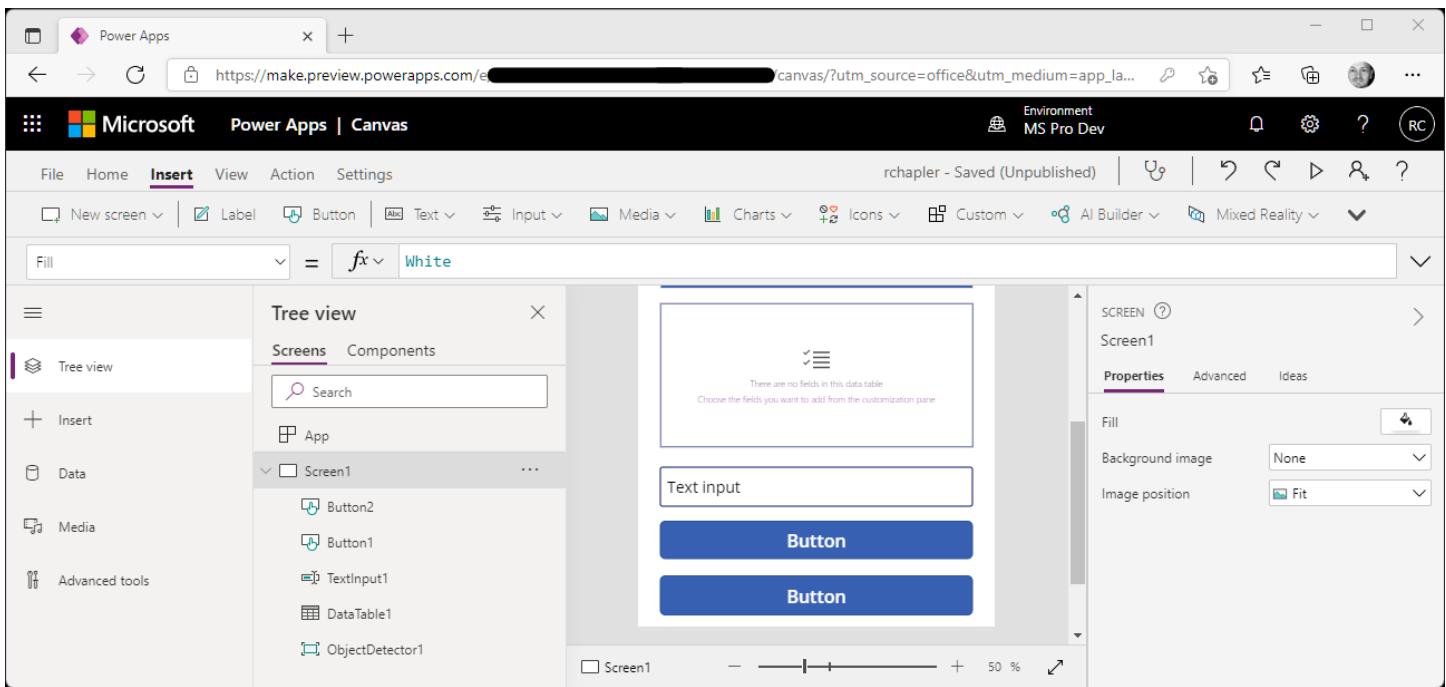
Paste the following logic into **Action | OnChange**:

```
ClearCollect(Results, ObjectDetector1.GroupedResults);
```

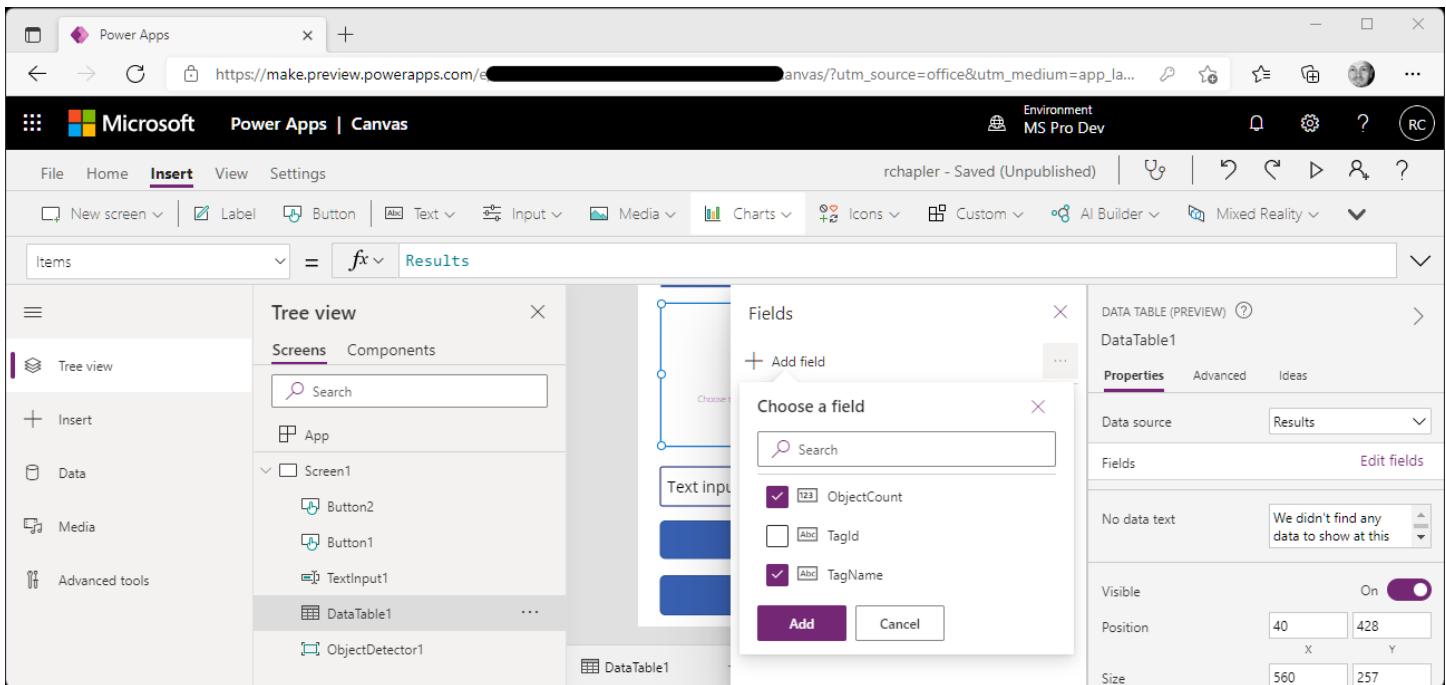
## DataTable1



Click on **DataTable1** in the left-hand **Tree View**. On the resulting pop-up, select “**In your app**” | **Results**.



On the updated control, click the “Choose the fields you want to add...” link.



On the resulting **Fields** pop-out, click “+ Add field”.

In the resulting “Choose a field” pop-up, check **TagName** and then **ObjectCount**.

Click the **Add** button.

## TextInput1

The screenshot shows the Microsoft Power Apps Canvas editor interface. On the left, there's a tree view of components under 'Screen1'. In the center, a table is displayed with two rows: 'Headlights' and 'ObjectCount'. Below the table, there are three buttons labeled 'Text input', 'Button1', and 'Button2'. On the right, a detailed properties pane is open for the 'Text input' component. The 'Properties' tab is selected, showing the following settings:

Default	Text input
Format	Text
Hint text	"Enter Comments"
Font	Open Sans
Font size	21
Font weight	B Normal
Font style	/ U +
Text alignment	Left Center

Click on **TextInput1** in the left-hand **Tree View**.

On the resulting “**TEXT INPUT**” pop-out, **Advanced** tab, enter the following items:

Default	""
Hint Text	"Enter Comments"

## Button1

The screenshot shows the Microsoft Power Apps Canvas editor interface. On the left, there's a tree view of components under 'Screen1'. In the center, a table is displayed with two rows: 'Enter Comments' and 'Submit'. Below the table, there are two buttons labeled 'Submit' and 'Button'. On the right, a detailed properties pane is open for the 'Submit' button. The 'Properties' tab is selected, showing the following settings:

Text	Submit
Display mode	Edit
Visible	On
Position	X: 40 Y: 819
Size	Width: 560 Height: 67
Padding	Top: 5 Bottom: 5

Click on **Button1** in the left-hand **Tree View**.

On the resulting right-hand pop-out, click the **Properties** tab, and enter the word “Submit” in the **Text** input.

On the resulting right-hand pop-out, click the **Advanced** tab,

Paste the following logic into **Action | OnSelect**:

```
Patch(myTable, Defaults(myTable), {Name:TextInput1.Text, Picture:ObjectDetector1.OriginalImage})
```

## Button2

The screenshot shows the Microsoft Power Apps Canvas editor interface. On the left, there is a tree view of the app components, including a screen named "Screen1" which contains "Button2", "Button1", and "TextInput1". The main canvas area displays a "Submit" button and a "Reset" button. The "Reset" button is currently selected, as indicated by its blue color and the fact that it is highlighted in the properties panel on the right. The properties panel shows the "Advanced" tab selected, with the "Text" input field containing the value "Reset".

Click on **Button2** in the left-hand **Tree View**.

On the resulting right-hand pop-out, click the **Properties** tab, and enter the word “Reset” in the **Text** input.

On the resulting right-hand pop-out, click the **Advanced** tab,

Paste the following logic into **Action | OnSelect**:

```
Reset(TextInput1);
```

## Confirm Success

Click **File** in the menu bar, then “**Save As**”, confirm the app name, and finally, click the **Save** button in the bottom-right.

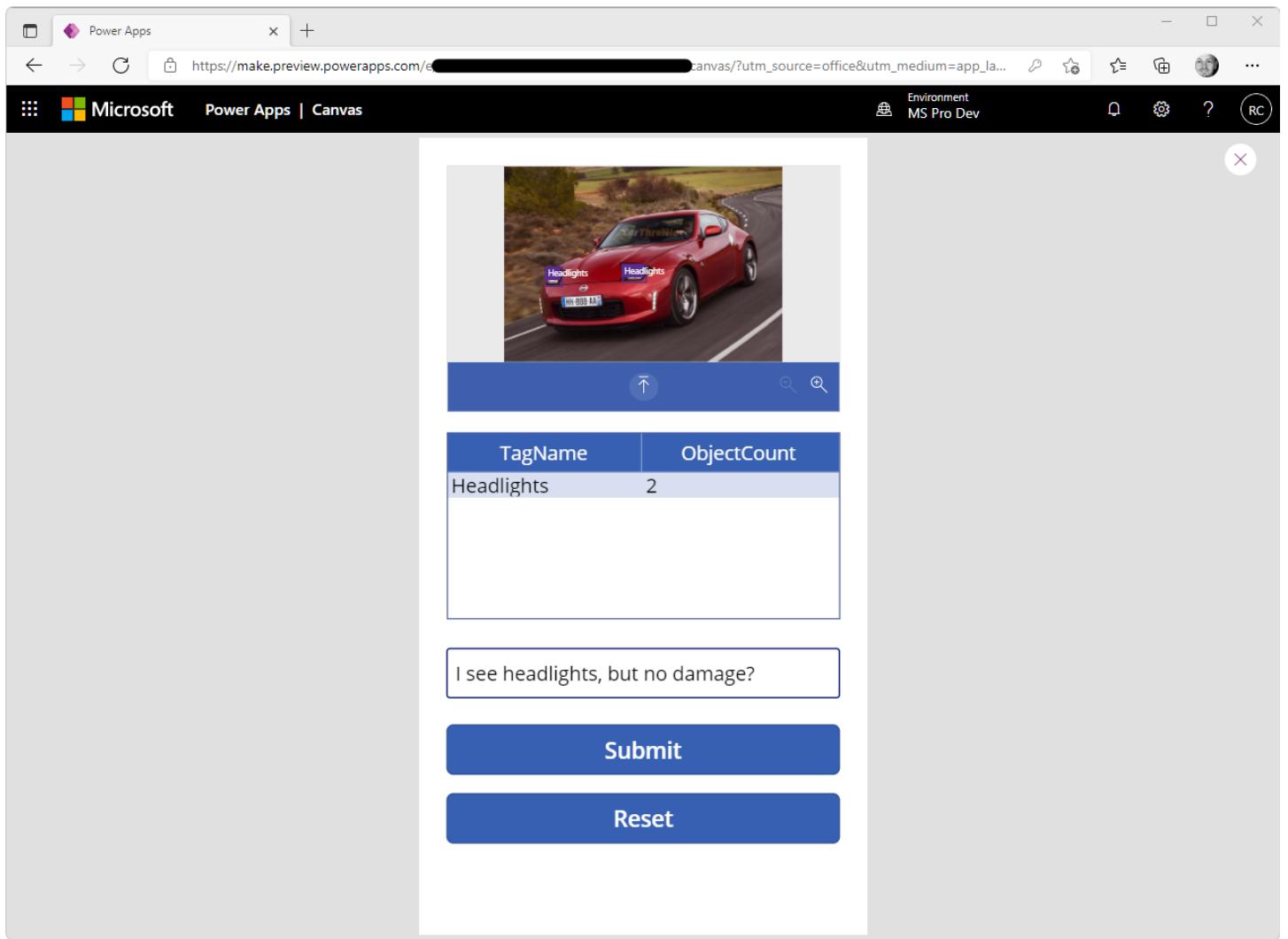
Return to the main page, click the “**Preview the app**” button {i.e., Play icon} in the upper-right.

The screenshot shows the Microsoft Power Apps interface in Canvas mode. The left sidebar displays a tree view of the app's components, including screens and components like Button2, Button1, TextInput1, DataTable1, and ObjectDetector1. The main canvas area shows a red sports car with bounding boxes labeled "Headlights". Below the image is a table with one row:

TagName	ObjectCount
Headlights	2

Below the table is a text input field labeled "Enter Comments" and two blue buttons labeled "Submit" and "Reset". The right side of the screen shows the "Screen" properties panel for "Screen1", which includes tabs for Properties (selected), Advanced, and Ideas. The "Properties" tab contains fields for ACTION (OnVisible, OnHidden), DATA (BackgroundImage, ContentLanguage), and DESIGN (Fill, ImagePosition). The "Fill" field is set to "White" and the "ImagePosition" field is set to "ImagePosition.Fit".

Your app will be presented in a phone simulation.



Click the **Detect** button and select an image file to test objection detection.

Review the resulting values in the data table.

Enter a comment in the text input and click the **Submit** button.

The screenshot shows the Microsoft Azure Query editor (preview) interface. The left sidebar contains navigation links for Power Apps, practicumsd (practicumsds/practicumsd), Dashboard, practicumsds, practicumsd (practicumsds/practicumsd), SQL database, Overview, Activity log, Tags, Diagnose and solve problems, Quick start, Query editor (preview), Power Platform, Power BI (preview), Power Apps (preview), Power Automate (preview), Settings, Compute + storage, Connection strings, Maintenance, Properties, Locks, Data management, Replicas, Sync to other databases, and Integrations. The main area shows a login screen for 'practicumsd (rchapler)'. Below it, the object explorer shows Tables (dbo.myTable selected), Views, and Stored Procedures. The query editor has two tabs: Query 1 and Query 2. Query 2 is active, showing the query: 'SELECT TOP (1000) \* FROM [dbo].[myTable]'. The results pane displays a table with two rows:

ID	Name	Picture
1	Nice car, but no apparent quality d...	/9j/4AAQSkZJRgABAQEAAAAAAA...
2	I see headlights, but no damage?	/9j/2wBDAAYEBQYFBAYGBQYHBwY...

At the bottom, a message says 'Query succeeded | 0s'.

Navigate to **Query Editor** in the Azure SQL Database and confirm that your comment has been added as a new row.

Consider publishing your app and loading it on your phone as one additional confirmation of success.

# Governance

Objective(s) in this section describe methods for cataloging and classifying data.

## Objective: Purview | Discover Data

Data discovery is “an organic process based on communal knowledge”, characterized by the following questions:

- **Search** ... “does the data that I need exist?”
- **Location** ... “how do I connect to Data Asset X?”
- **Related** ... “are there related data assets that I might find valuable?”
- **Contacts** ... “who are the owners and experts for Data Asset X?”

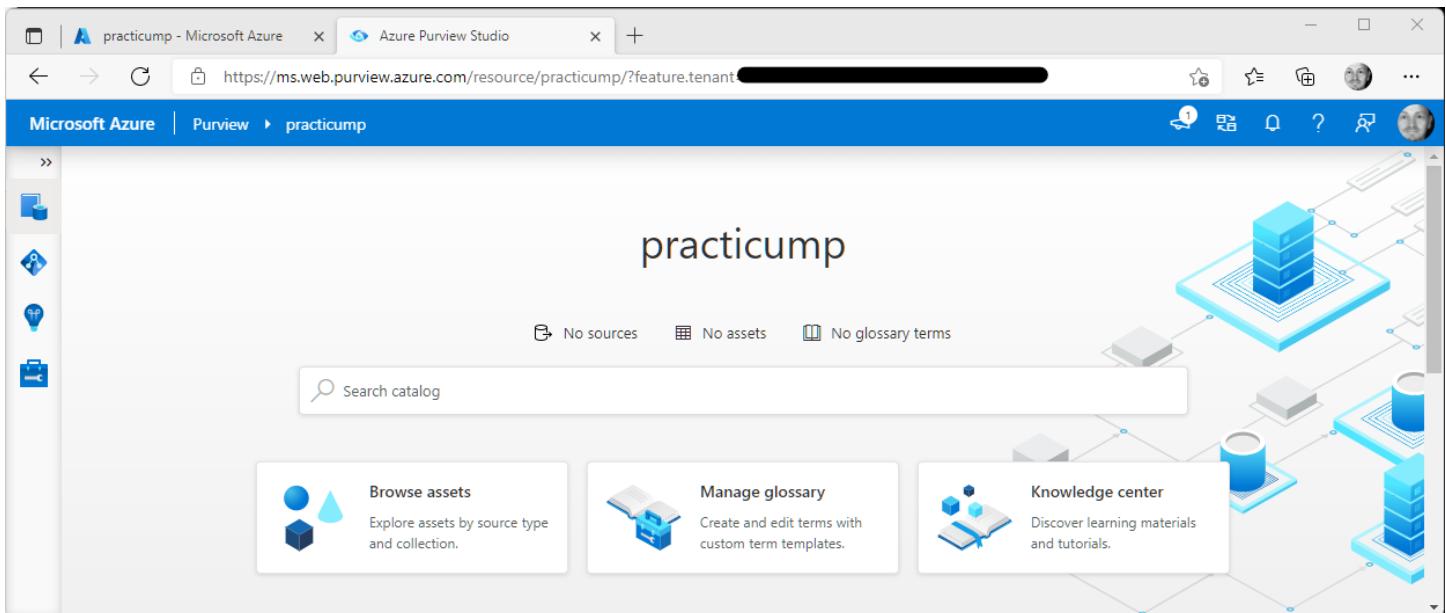
In this section, we will configure Purview and successfully respond to these questions.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Purview
- SQL

Navigate to your instance of Purview in the Azure portal.

Click the “Open Purview Studio” button.



### Register Source

Click the **Data Map** icon on the left-hand navigation.

Click the **Register** button on the resulting page.

practicump - Microsoft Azure    Azure Purview Studio

https://ms.web.purview.azure.com/resource/practicump/main/datasource/registeredSources?feature.tenant=██████████

Microsoft Azure | Purview > practicump

Search assets

- Sources
- Collections
- Source management
- Scan rule sets
- Integration runtimes
- Annotation management
- Classifications
- Classification rules

## Sources

Register Refresh Map view

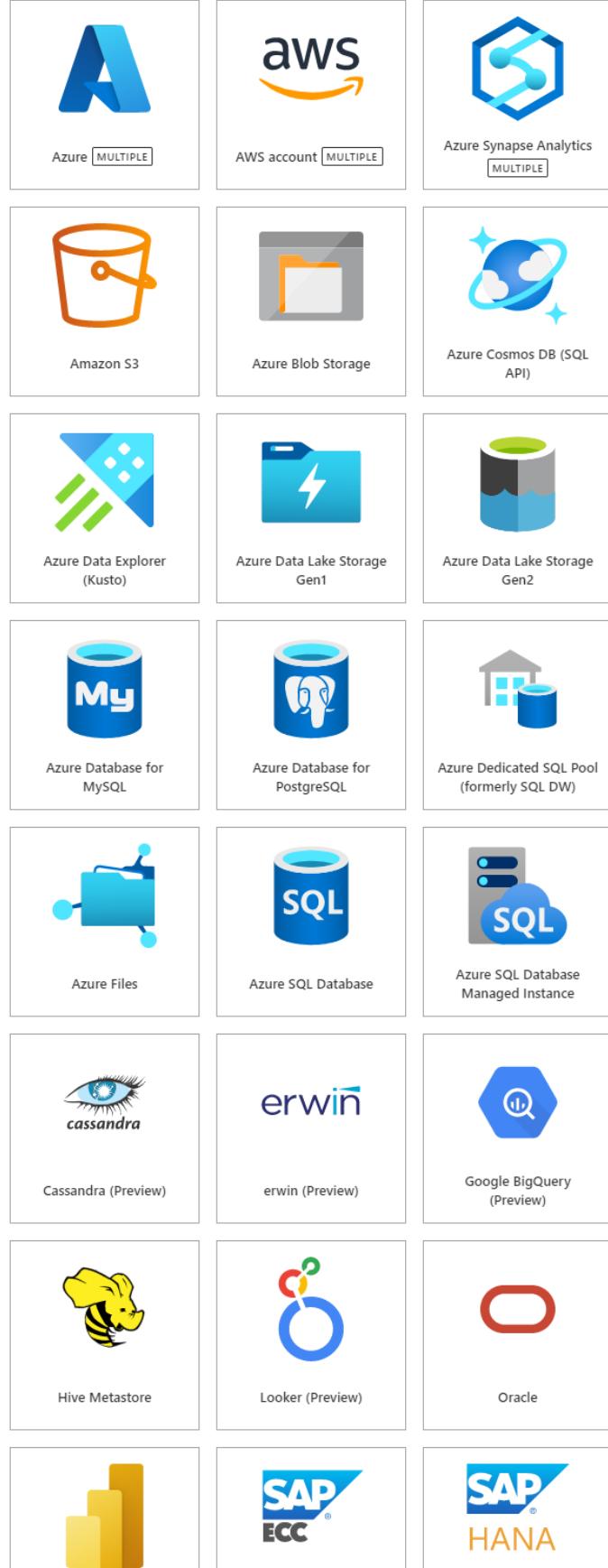
Filter by keyword

Showing 1 collection, 0 sources

## Register sources

Filter by keyword

All Azure Database Power BI Services and apps



Click the **Azure** tab on the “Register sources” pop-out.

Click to check the “**Azure SQL Database**” option.

Click the **Continue** button.

The screenshot shows the Azure Purview Studio interface with the URL [https://ms.web.purview.azure.com/resource/practicump/main/datasource/registeredSources?feature.tenant=\[REDACTED\]](https://ms.web.purview.azure.com/resource/practicump/main/datasource/registeredSources?feature.tenant=[REDACTED]). On the left, there's a sidebar with various management options like Sources, Collections, and Scan rule sets. The main area is titled 'Sources' and has a 'Register' button. A 'Register sources (Azure SQL Database)' dialog is open on the right. It contains fields for 'Name' (practicumsd), 'Azure subscription' (rchapler (91e9fddc-ef15-416c-9be1-085f8b1b46ed)), 'Server name' (practicumsd), 'Endpoint' (practicumsd.database.windows.net), and 'Select a collection' (practicump). A note below says 'All assets under this source will belong to the collection you select.' At the bottom of the dialog are 'Register', 'Back', and 'Cancel' buttons.

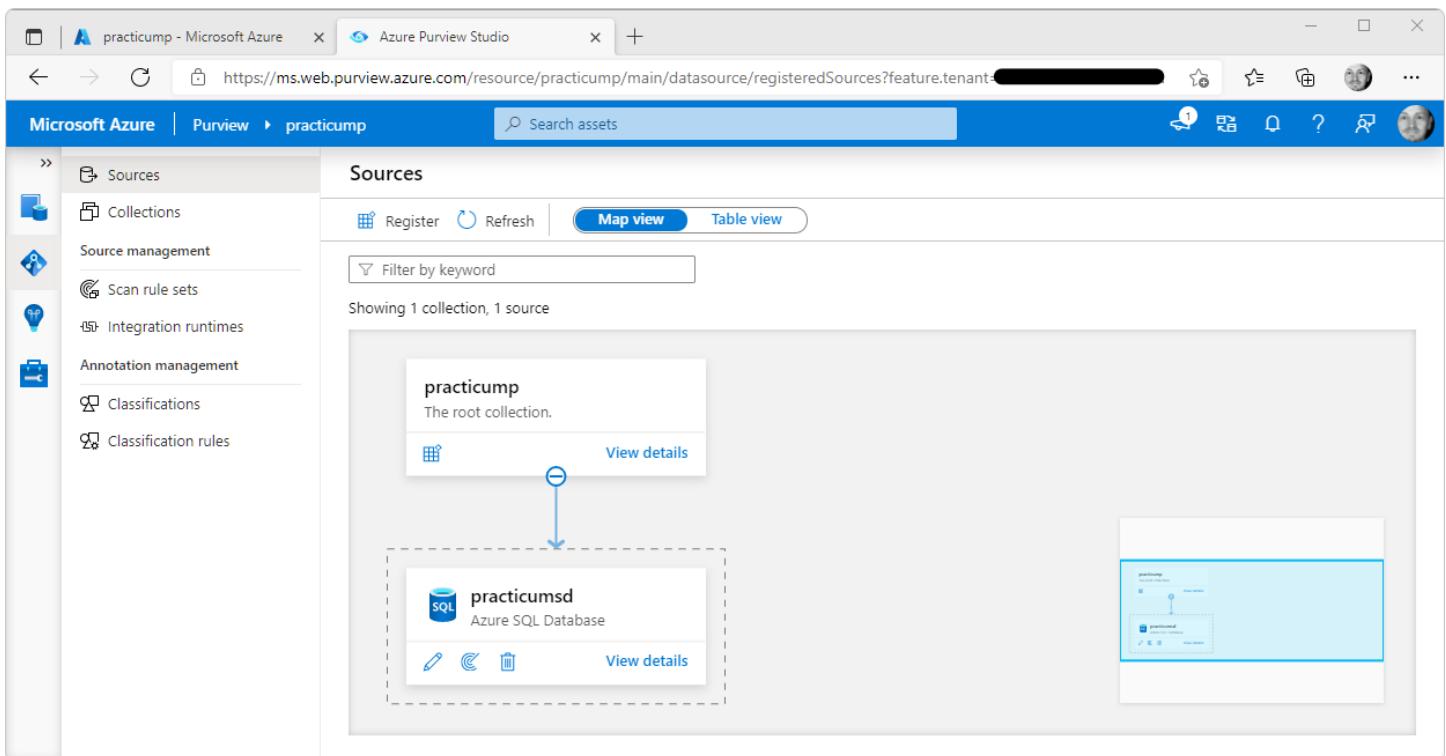
On the “Register sources (Azure SQL Database)” pop-out, enter values for the following items:

Name	Self-Explanatory
Subscription	Select your subscription
Server Name	Enter the name of your SQL Database Server
Endpoint	Confirm the populated value
Select a Collection	Leave default value

Click the **Register** button.

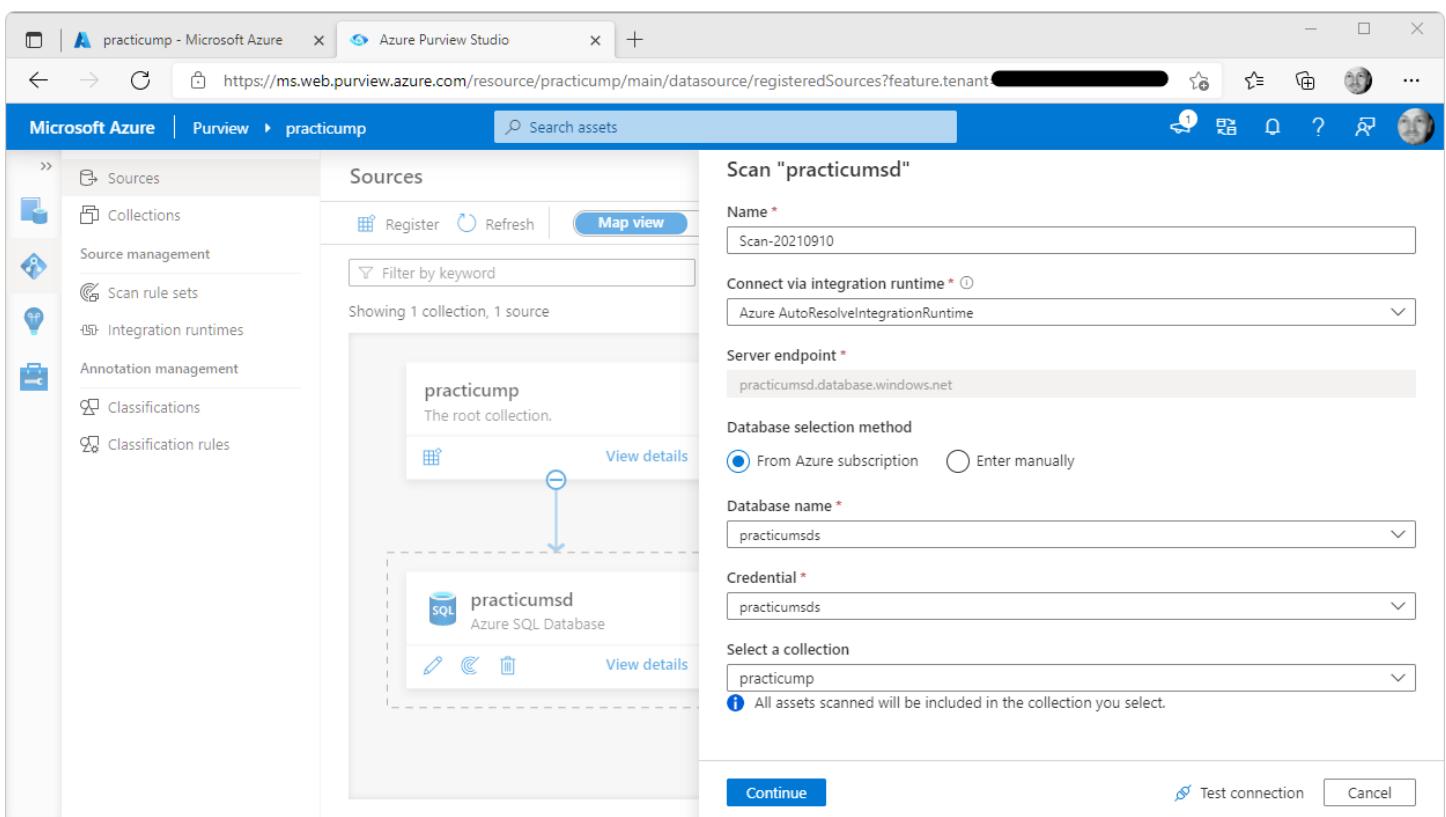
Scan Data

Navigate to Sources, “Map view”.



The screenshot shows the Microsoft Azure Purview Studio interface. The left sidebar has a 'Sources' icon selected. The main area displays a tree view under the 'Sources' tab. A collection named 'practicump' is expanded, showing its root status and a 'View details' button. Below it, a sub-collection 'practicumsd' is shown, also with a 'View details' button. To the right, there's a preview pane showing a simplified version of the collection structure.

Click the “New scan” icon on the newly created item.



The screenshot shows the 'Scan "practicumsd"' configuration dialog. It includes fields for Name (set to 'Scan-20210910'), Connect via integration runtime (set to 'Azure AutoResolveIntegrationRuntime'), Server endpoint (set to 'practicumsd.database.windows.net'), Database selection method (radio button selected for 'From Azure subscription'), Database name (set to 'practicumsds'), Credential (set to 'practicumsds'), and Select a collection (set to 'practicump'). At the bottom are 'Continue', 'Test connection', and 'Cancel' buttons.

On the “Scan...” pop-out, enter values for the following items:

Name	Self-Explanatory
Database Selection Method	Select the “From Azure subscription” radio button
Database Name	Select your Azure SQL database
Credential	Select your “SQL Authentication” credential

Click “Test connection” to confirm successful configuration and then click the **Continue** button.

Sources

practicump

practicumsd

Scan rule sets

Scope your scan

Search

practicumsd

SalesLT.Customer

SalesLT.ProductModel

SalesLT.ProductDescription

SalesLT.Product

SalesLT.ProductModelProductDescription

SalesLT.ProductCategory

dbo.BuildVersion

dbo.ErrorLog

SalesLT.Address

SalesLT.CustomerAddress

SalesLT.SalesOrderDetail

SalesLT.SalesOrderHeader

SalesLT.vProductModelCatalogDescription

SalesLT.vProductAndDescription

SalesLT.vGetAllCategories

Continue Back Cancel

Review and select desired data assets from those listed in the “Scope your scan” pop-out and then click the **Continue** button.

Sources

practicump

practicumsd

Select a scan rule set

+ New scan rule set Refresh

Select one scan rule set to be used by your scan.

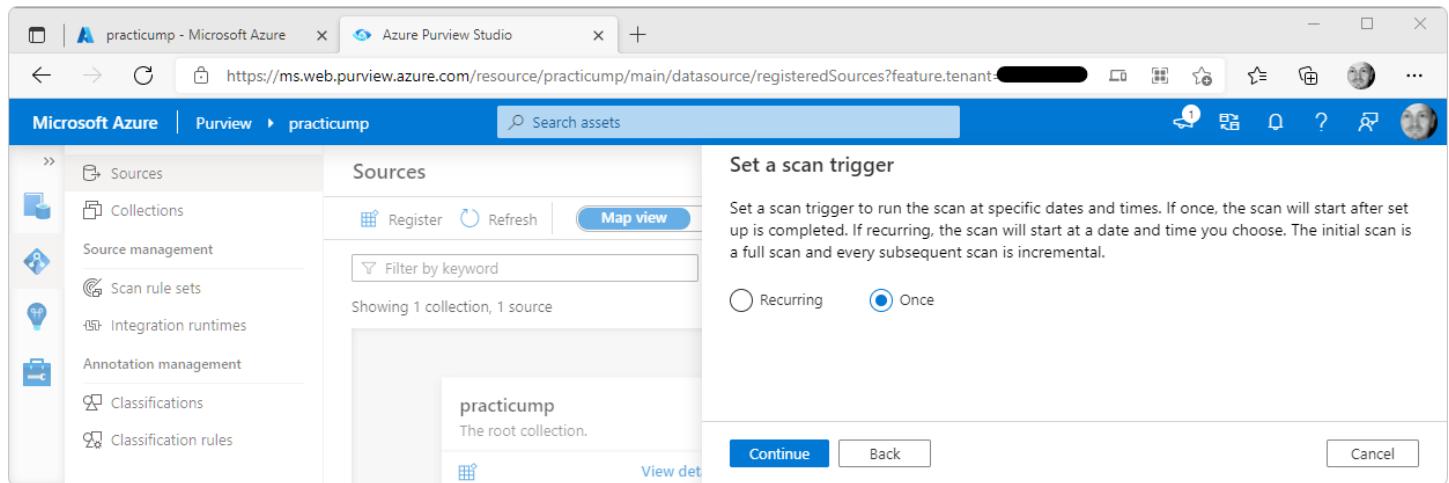
AzureSqlDatabase [SYSTEM DEFAULT]

Microsoft default scan rule set that includes all supported system classification rules

View detail

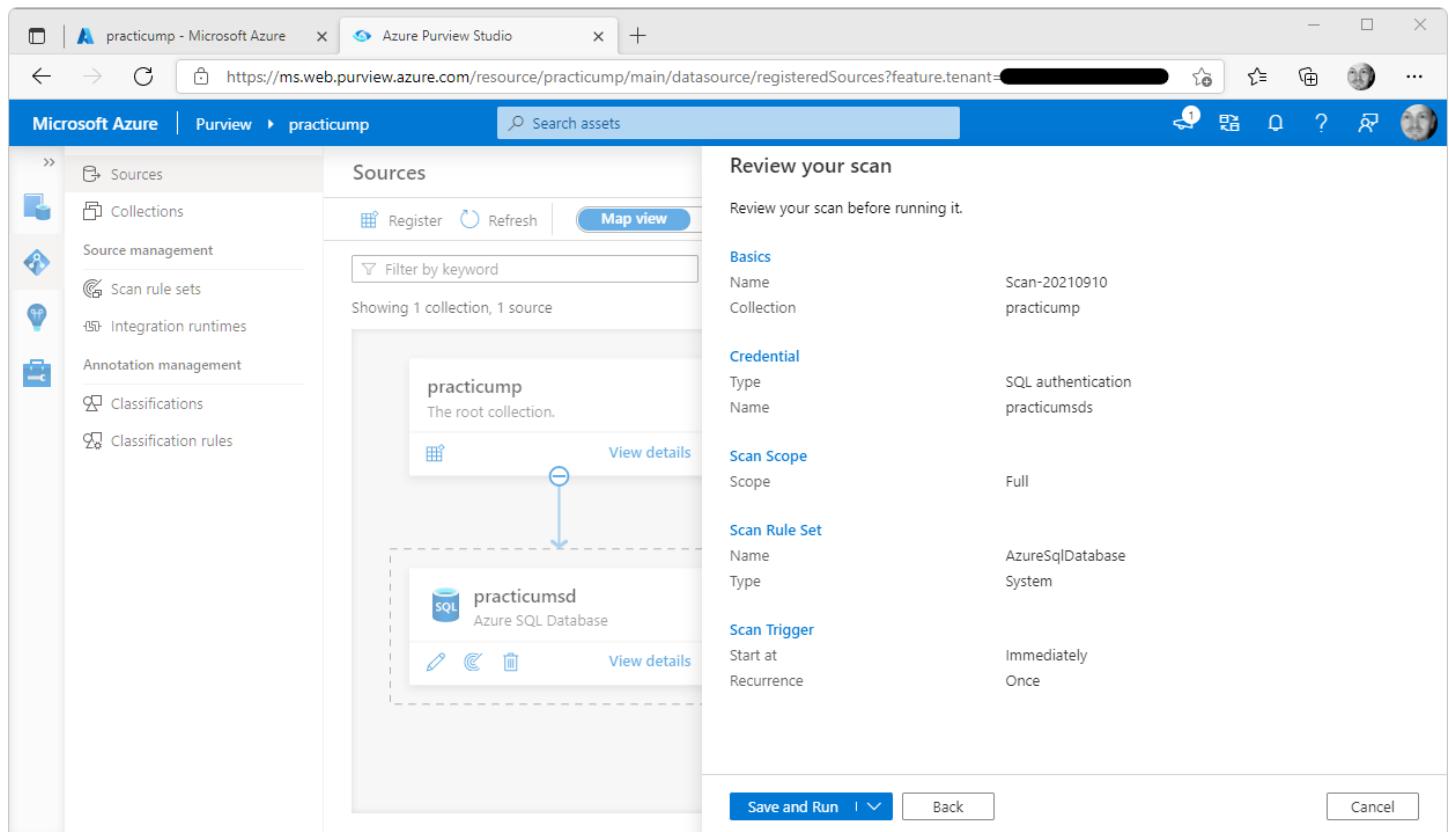
Continue Back Cancel

Confirm selection of the default Scan Rule Set on the “Select a scan rule set” pop-out and then click the **Continue** button.



The screenshot shows the Azure Purview Studio interface. On the left, there's a sidebar with icons for Sources, Collections, Source management, Scan rule sets, Integration runtimes, Annotation management, Classifications, and Classification rules. The main area is titled "Sources" and shows a single collection named "practicump". A modal window titled "Set a scan trigger" is open over the main content. Inside the modal, there's a description: "Set a scan trigger to run the scan at specific dates and times. If once, the scan will start after setup is completed. If recurring, the scan will start at a date and time you choose. The initial scan is a full scan and every subsequent scan is incremental." Below the description are two radio buttons: "Recurring" and "Once". The "Once" button is selected and highlighted in blue. At the bottom of the modal are three buttons: "Continue", "Back", and "Cancel".

Click the **Once** radio button on the “Set a scan trigger” pop-out and then click the **Continue** button.



The screenshot shows the Azure Purview Studio interface. The left sidebar is identical to the previous screenshot. The main area shows the "practicump" collection. A modal window titled "Review your scan" is open. It contains a summary of the scan settings: Name: Scan-20210910, Collection: practicump; Type: SQL authentication, Name: practicumsts; Scope: Full; Scan Rule Set: AzureSqlDatabase System; Scan Trigger: Start at: Immediately, Recurrence: Once. At the bottom of the modal are three buttons: "Save and Run", "Back", and "Cancel".

Review scan settings and then click the **Save and Run** button.

Confirm Scan

Navigate to Sources, “Table view”.

**Sources**

Showing 1 source

Name	Source type	Collection	Source id	Scans	Registered on
AzureSqlDatabase-P3k	Azure SQL Data...	practicump	practicumsds.d...	1	09/10/21 01:59 PM

Note that the **Scans** column now shows 1.

Click on the **Name** link to drill through to details. Allow time for processing.

**practicumsd**

Azure SQL Database

New scan Edit source Delete source Refresh

**Overview** Scans

Source ID: practicumsd.database.windows.net Registered on 09/11/2021 03:18:28 AM

Scans 1 Scanned assets 17

Classified assets 15

Recent scans

Scan name	Last run status	Scan rule set	Last scan time
Scan-20210910	In progress	AzureSqlData...	09/11/21 03:25 AM

→ See all applied scans

Recent failed scans

Scan name	Status	Suggestions
No recent failed scan runs		

→ See all failed scan runs

Collection path  
practicump

Source hierarchy  
rchapter  
Subscription  
practicumrg  
Resource group

## Browse Assets

After scanning, it takes time for **Asset Insights** to reflect new assets.

We do not have to wait to browse assets, however.

Click the **Data Catalog** icon on the left-hand navigation.

The screenshot shows the Azure Purview Studio interface. The title bar reads "practicump - Microsoft Azure" and "Azure Purview Studio". The URL in the address bar is "https://ms.web.purview.azure.com/resource/practicump/?feature.tenant=[REDACTED]". The top navigation bar includes "Microsoft Azure", "Purview", and "practicump". On the left, there's a vertical sidebar with icons for Data Catalog, Data Governance, Data Quality, and Data Integration. The main area has a title "practicump" and a summary: "1 source", "No assets", and "No glossary terms". A search bar says "Search catalog". Below are three buttons: "Browse assets" (Explore assets by source type and collection), "Manage glossary" (Create and edit terms with custom term templates), and "Knowledge center" (Discover learning materials and tutorials). To the right is a network diagram showing various data assets connected.

Click on the “**Browse assets**” button.

The screenshot shows the "Browse assets" page under the "Data catalog" section. The title bar and URL are identical to the previous screenshot. The left sidebar shows "Data catalog >". The main content area has a title "Browse assets" with a "Refresh" button. It offers two filtering options: "By collection" (selected) and "By source type". There's also a "Filter by keyword" input field. Below, it says "Showing 1 collection". A table lists one collection:

Name	Description	Assets	Collection admin
practicump	The root collection.	4	Rich Chapler

Click on the Root Collection link {i.e., name}.

**Browse assets**

By collection **practicump** By source type

Source types : all Instances : all Clear all filters

Showing 1-23 out of 23 results Sort by: Name

Asset Type	Name	Description
Azure SQL Table	Address	mssql://practicumsd.database.windows.net/practicumsds/SalesLT/Address
Azure SQL Table	BuildVersion	mssql://practicumsd.database.windows.net/practicumsds/dbo/BuildVersion
Azure SQL Table	Customer	mssql://practicumsd.database.windows.net/practicumsds/SalesLT/Customer
Azure SQL Table	CustomerAddress	mssql://practicumsd.database.windows.net/practicumsds/SalesLT/CustomerAddress
Azure SQL Schema	dbo	mssql://practicumsd.database.windows.net/practicumsds/dbo

Page 1 of 1

Explore results.

Click on the first item, **Address**.

The screenshot shows the Azure Purview Studio interface. At the top, there are two tabs: 'practicump - Microsoft Azure' and 'Azure Purview Studio'. The URL in the address bar is <https://ms.web.purview.azure.com/resource/practicump/main/catalog/entity?guid=81afae99-9619-4559-aa95-b5f6f6f60000&secti...>. The main content area displays an asset named 'Address' (type: Azure SQL Table). The 'Overview' tab is selected. The asset has no description, classifications, or glossary terms. It is part of a collection path 'practicump' and is located in a hierarchy under 'SalesLT' schema. The last update was on September 11, 2021, at 10:29 AM UTC by an automated scan. A blue button at the top right says 'Open in Power BI Desktop'.

Notice the automatic “**World Cities**” classification.

Drill-through the tabs {e.g., Properties, Schema, Lineage, etc.}.

## Confirm Success

Now that we have implemented the basics in Purview, can we answer the questions previously posed?

### Search

Question: “**Does the data that I need exist?**”

Enter a phrase into the search box at the top center of the window {e.g., “address”}.

The screenshot shows the Microsoft Azure Purview Studio interface. The top navigation bar includes tabs for 'practicump - Microsoft Azure' and 'Azure Purview Studio'. The main title bar says 'Microsoft Azure | Purview > practicump'. The address bar contains the URL 'https://ms.web.purview.azure.com/resource/practicump/main/catalog/search?searchQueryKeyword=address&searchQueryLimit=2...'. The left sidebar has a 'Data catalog' section with a 'Search results for address' heading. It includes filters for 'Source types : all', 'Instances : all', and a 'Clear all filters' button. Below these are sections for 'Collection', 'Classification', 'Contact', 'Label', and 'Glossary term', each with a checkbox and a count of results (e.g., 2 for Collection). The main content area displays two search results: 'Address' (an Azure SQL Table located at mssql://practicumsd.database.windows.net/practicumsds/SalesLT/Address) and 'CustomerAddress' (an Azure SQL Table located at mssql://practicumsd.database.windows.net/practicumsds/SalesLT/CustomerAddress). The results are sorted by Relevance. At the bottom, there are navigation buttons for 'Previous', 'Page 1 of 1', and 'Next'.

Returned search results will include all assets that include the requested phrase.

## Location

Question: "how do I connect to Data Asset X?"

Drill into the first search result, Address.

The screenshot shows the Azure Purview Studio interface with the URL <https://ms.web.purview.azure.com/resource/practicump/main/catalog/entity?guid=81afae99-9619-4559-aa95-b5f6f6f60000&secti...>. The main content area displays the 'Address' asset, which is an Azure SQL Table. The 'Overview' tab is selected, showing the following details:

- Collection path:** practicump
- Hierarchy:** A tree view showing the database structure:
  - practicumsd.database.windows.net (Azure SQL Server)
  - practicumsds (Azure SQL Database)
    - SalesLT (Azure SQL Schema)
      - Address (Azure SQL Table)

Other tabs visible include Properties, Schema, Lineage, Contacts, and Related. A note indicates the asset was updated on September 11, 2021, at 10:29 AM UTC by automated scan.

On the **Overview** tab, you will see **Hierarchy** information which includes the Fully Qualified Domain Name of the Azure SQL Server.

## Related

Question: “**are there related data assets that I might find valuable?**”

Continue in the search result for phrase “Address”.

The screenshot shows the Azure Purview Studio interface. The left sidebar has icons for Data Catalog, Purview, and other services. The main area shows a search result for "Address". The "Related" tab is selected, displaying a list of 13 related assets under the heading "Showing 1 to 13 of 13 items". The assets listed are: Address (Azure SQL Table), Customer (Azure SQL Table), CustomerAddress (Azure SQL Table), Product (Azure SQL Table), ProductCategory (Azure SQL Table), ProductDescription (Azure SQL Table), ProductModel (Azure SQL Table), ProductModelProductDescription (Azure SQL Table), SalesOrderDetail (Azure SQL Table), and SalesOrderHeader (Azure SQL Table). The "Overview" tab is also visible on the left.

On the **Related** tab, you will see a list of related assets (as identified by Purview).

## Contacts

Question: “who are the owners and experts for Data Asset X?”

Continue in the search result for phrase “Address”.

The screenshot shows the Azure Purview Studio interface. The left sidebar has icons for Data Catalog, Purview, and other services. The main area shows a search result for "Address". The "Contacts" tab is selected, displaying sections for "Experts" and "Owners". Under "Experts", it says "No experts for this asset.". Under "Owners", it says "No owners for this asset.". The "Overview" tab is also visible on the left.

On the **Contacts** tab, you will see lists of **Experts** and **Owners** you can contact about the asset (as identified by Purview).

If, as in the snip above, there are no Contacts and you can add information not already captured, click the **Edit** button.

The screenshot shows the Azure Purview Studio interface for editing an asset named "Address". The top navigation bar includes tabs for "Overview", "Schema", and "Contacts", with "Contacts" being the active tab. On the left, there are sections for "Experts" and "Owners", each with a search bar and a list of users. The "Experts" section shows one user: Rich Chapter (Curator, richard.chapter@microsoft.com). The "Owners" section is currently empty. At the bottom of the page are "Save" and "Cancel" buttons.

In either the **Experts** or the **Owners** drop-down, search for the user or group in your directory. Click **Save**.

## Objective: Purview | Classify Data

Data classification is a process of annotating data assets to help organizations answer the following questions:

- **Definition** ... “how does data map to a business context?” and “what is the intended use of Data Asset X?”
- **Documentation** ... “where is the documentation of Data Asset X?”
- **Vocabulary** ... “what else {e.g., synonyms, acronyms, etc.} might help me understand Data Asset X?”
- **Standardization** ... “how do I classify all data assets in a standard way?”

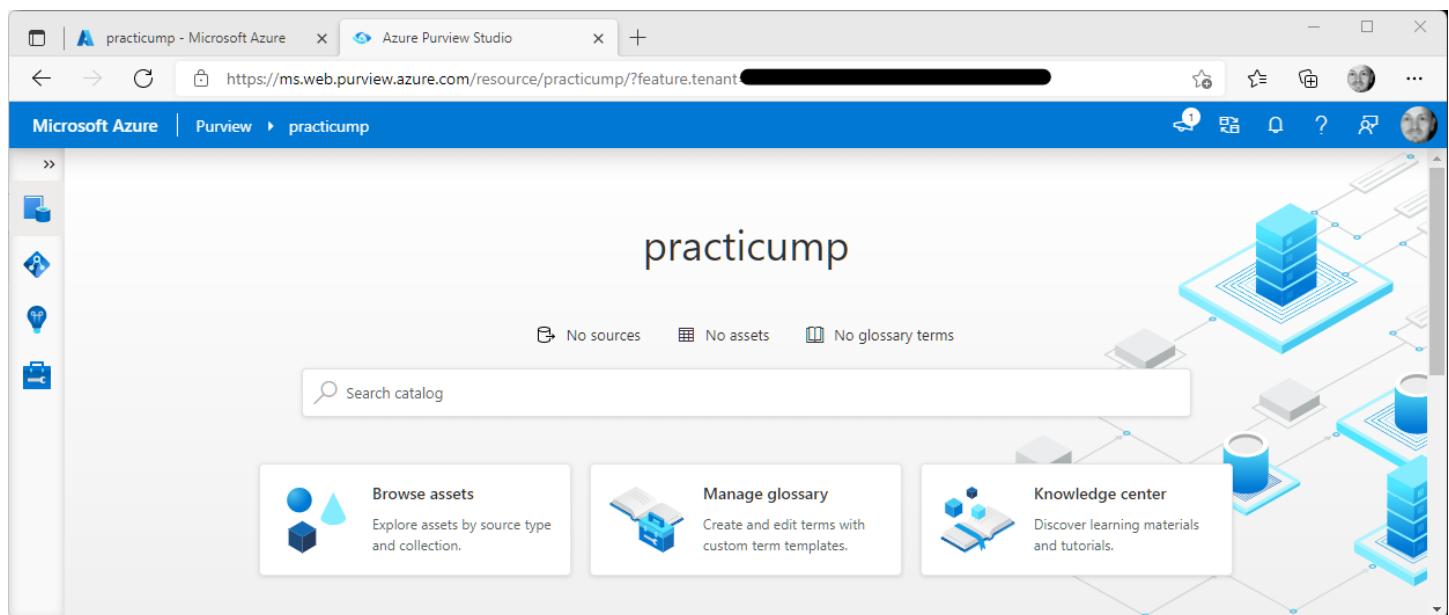
In this section, we will configure Purview and successfully respond to these questions.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

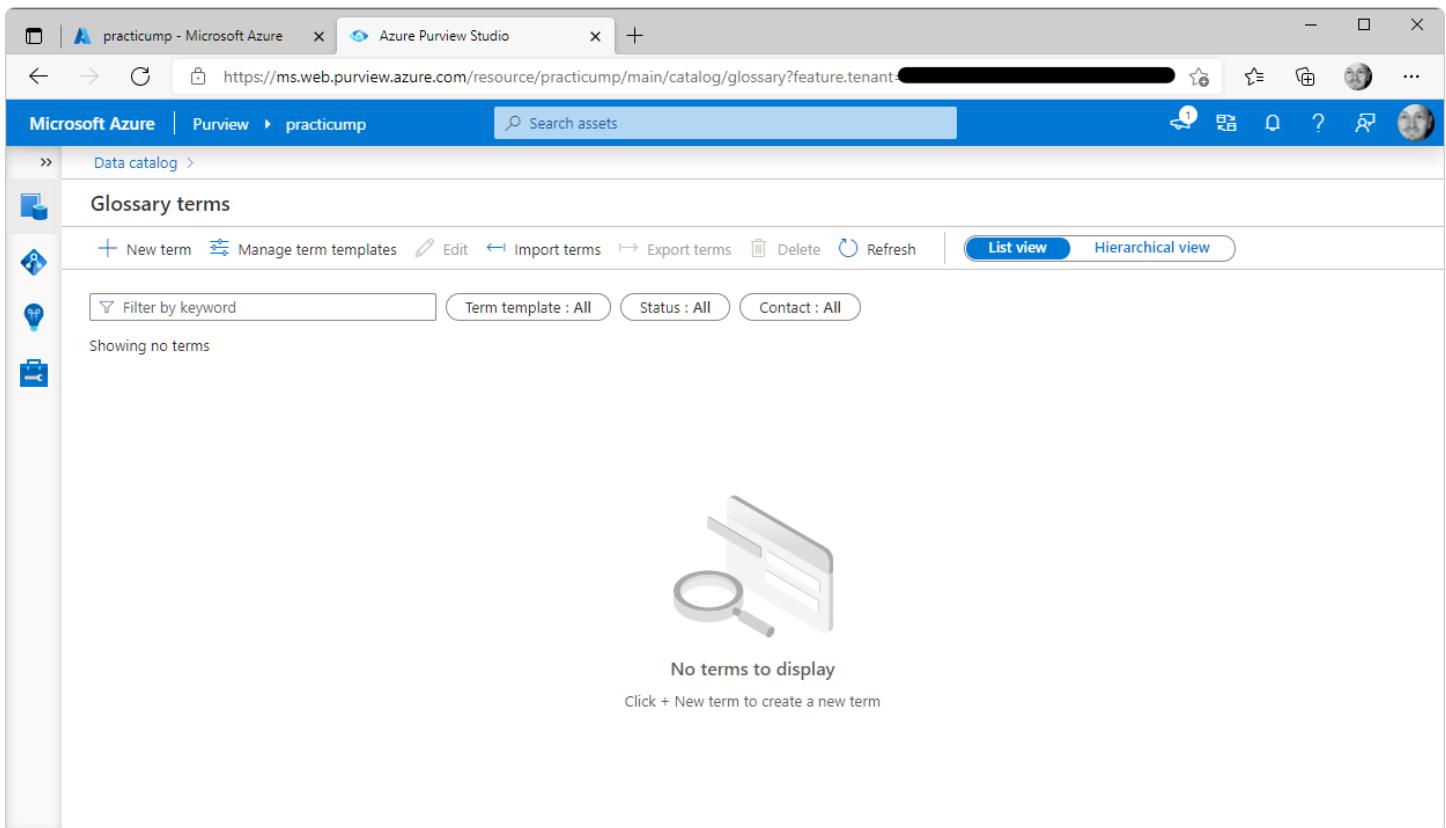
- Purview
- SQL

Navigate to your instance of Purview in the Azure portal.

Click the “Open Purview Studio” button.



Click the “Manage Glossary” button.

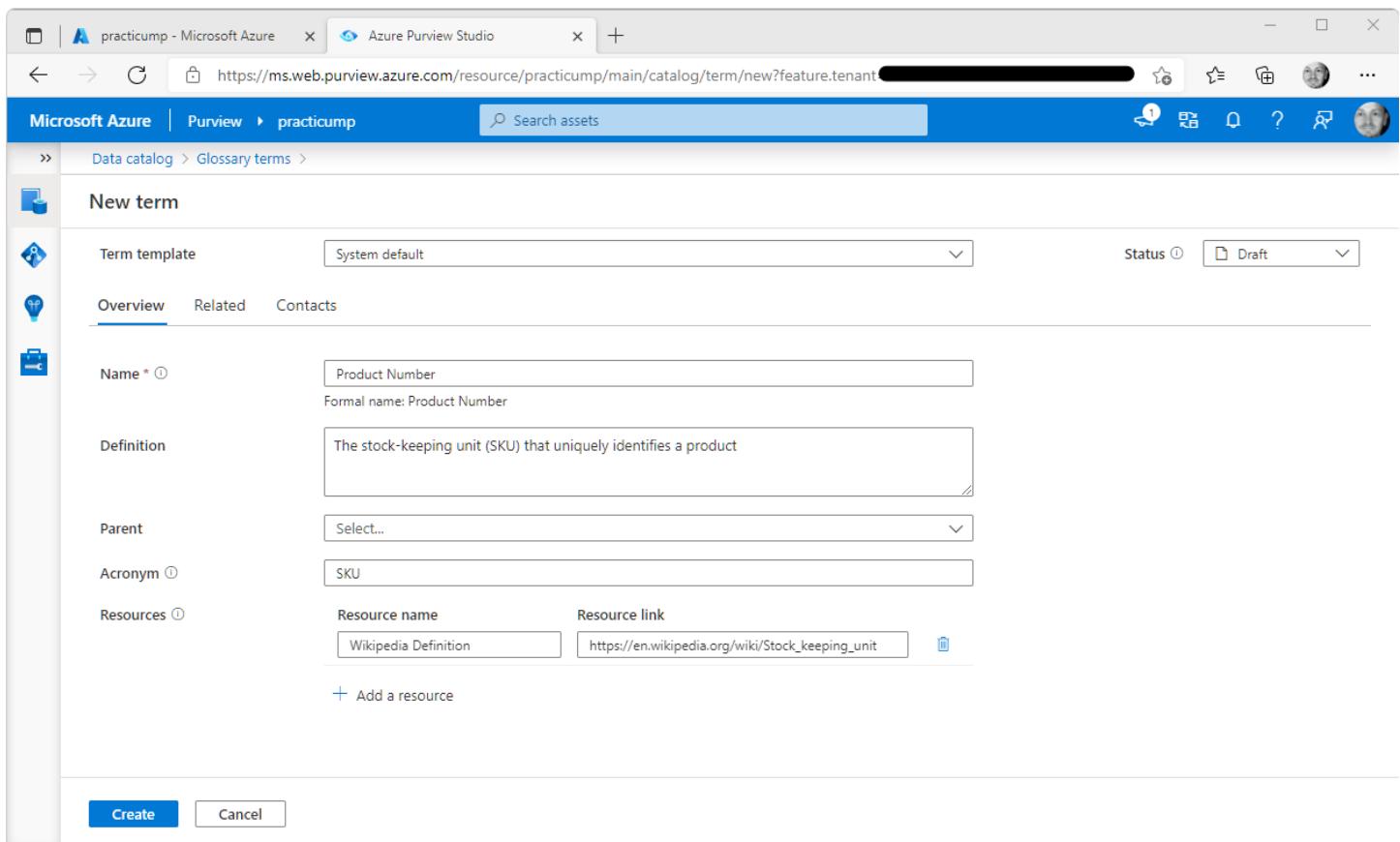


No terms to display  
Click + New term to create a new term

## Create Term

Click the “+ New term” button on the resulting page.

On the resulting pop-out, confirm selection of “System default” and then click the **Continue** button.



New term

Term template: System default

Status: Draft

Overview   Related   Contacts

Name \* (Product Number)  
Formal name: Product Number

Definition  
The stock-keeping unit (SKU) that uniquely identifies a product

Parent  
Select...

Acronym (SKU)

Resources (Wikipedia Definition: [https://en.wikipedia.org/wiki/Stock\\_keeping\\_unit](https://en.wikipedia.org/wiki/Stock_keeping_unit))

+ Add a resource

Create   Cancel

On the “New term” page, enter values for the following items:

Name	Self-Explanatory
Definition	Select your subscription
Parent	Select a parent term, if applicable
Acronym	Capture abbreviation, if applicable
Resources	Click “+ Add a resource” to capture hyperlinks with descriptive content

Click the **Create** button.

## Use Term

Search for “product” to identify assets that use terms like Product Number.

The screenshot shows the Azure Purview Studio interface with a search query "product". The results list four entries, all of which are Azure SQL Tables under the "practicump" collection:

- Product**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/Product
- ProductModelProductDescription**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/ProductModelProductDescription
- ProductCategory**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/ProductCategory
- ProductDescription**: Azure SQL Table, mssql://practicumsd.database.windows.net/practicumsds/SalesLT/ProductDescription

Click **Product**.

The screenshot shows the Azure Purview Studio interface. The top navigation bar includes 'practicump - Microsoft Azure' and 'Azure Purview Studio'. The search bar contains 'product'. The main content area displays the 'Data catalog > Search results "product" > Product' page. A sidebar on the left has icons for Data catalog, Data flow, Data quality, Data governance, and Data protection. The 'Product' card shows it's an 'Azure SQL Table'. Below the card are buttons for 'Edit', 'Refresh', 'Delete', and 'Edit columns', along with a 'Open in Power BI Desktop' button. The 'Schema' tab is selected, showing a table of columns with their details:

Column name	Classifications	Sensitivity label	Glossary terms	Data type	Asset description	Description
Color				nvarchar		
DiscontinuedDate				datetime		
ListPrice				money		
ModifiedDate				datetime		
Name				nvarchar		
ProductCategoryID				int		
ProductID				int		
ProductModelID				int		
ProductNumber				nvarchar		
rowguid				uniqueidentifier		

Click the **Schema** tab.

Click on the **ProductNumber** link.

The screenshot shows the Azure Purview Studio interface, similar to the previous one but focused on the 'ProductNumber' column. The top navigation bar and search bar are identical. The main content area displays the 'Data catalog > Search results "product" > ProductNumber' page. The 'ProductNumber' card shows it's an 'Azure SQL Column'. Below the card are buttons for 'Edit', 'Refresh', 'Delete', and 'Edit columns'. The 'Overview' tab is selected, showing the following details:

Asset description	Collection path
No description for this asset.	practicump
Description	Hierarchy
-	
Classifications	Glossary terms
No classifications for this asset.	No glossary terms for this asset.
Fully qualified name	mssql://practicumsd.database.windows.net/practicumsds/SalesLT/Product#ProductNumber

Click the **Edit** button.

The screenshot shows the Azure Purview Studio interface. The top navigation bar includes 'practicump - Microsoft Azure', 'Azure Purview Studio', and a search bar with the query 'product'. Below the navigation is a breadcrumb trail: 'Microsoft Azure | Purview > practicump > Data catalog > Search results "product" > ProductNumber > Edit "ProductNumber"'. On the left, there's a sidebar with icons for Overview, Contacts, and other data management options. The main content area is titled 'Edit "ProductNumber"'. It has tabs for 'Overview' (which is selected) and 'Contacts'. Under 'Overview', there are fields for 'Name' (set to 'ProductNumber'), 'Asset description' (empty), 'Classifications' (dropdown menu 'Select...'), and 'Glossary terms' (dropdown menu showing 'Product Number' with a checked checkbox). At the bottom are 'Save' and 'Cancel' buttons.

Select the new term, “**Product Number**” from the “**Glossary terms**” drop-down.

Click the **Save** button.

## Confirm Success

Now that we have implemented the basics in Purview, can we answer the questions previously posed?

### Definition

Questions: “**how does data map to a business context?**” and “**what is the intended use of Data Asset X?**”

The screenshot shows the Azure Purview Studio interface. The URL is <https://ms.web.purview.azure.com/resource/practicump/main/catalog/term?section=overview&termGuid=52836041-66f9-43d5-8...>. The page displays a glossary term named "Product Number".

**Product Number**

- Term | System default
- [Edit](#) [Delete](#) [Refresh](#)

[Overview](#) [Related](#) [Contacts](#)

**Formal name**: Product Number

**Parent**: No parent term for this term

**Definition**: The stock-keeping unit (SKU) that uniquely identifies a product.

**Last updated**: 09/13/2021 07:37 by Rich Chapter

**Created on**: 09/13/2021 07:37 by Rich Chapter

**Acronym**: [SKU](#)

**Catalog assets**: 1 asset associated with "Product Number" [View asset](#)

**Resources**: [Wikipedia Definition](#)

This also demonstrates response to questions:

- **Documentation** ... “where is the documentation of Data Asset X?”
  - Links to documentation can be included in **Resources**
- **Vocabulary** ... “what else {e.g., synonyms, acronyms, etc.} might help me understand Data Asset X?”

## Standardization

Question: “**how do I classify all data assets in a standard way?**”

The screenshot shows the Azure Purview Studio interface. The URL is <https://ms.web.purview.azure.com/resource/practicump/main/catalog/entity/edit?guid=81afae99-9619-4559-aa95-b5f6f6f60000&...>. The page displays the schema for an "Address" asset.

**Edit "Address"**

[Overview](#) [Schema](#) [Contacts](#)

**Warning**: Making a manual update to the schema will prevent future scans on this asset from updating it.

Column name	Common Passwords	Glossary terms	Data type	Asset description
AddressID	<input type="checkbox"/>	Select...	int	
AddressLine1	<input type="checkbox"/>	Select...	nvarchar	
AddressLine2	<input type="checkbox"/>	Select...	nvarchar	
City	<input type="checkbox"/>	Select...	nvarchar	
CountryRegion	<input checked="" type="checkbox"/> Country/Region	Select...	nvarchar	
ModifiedDate	<input type="checkbox"/>	Select...	datetime	

[Save](#) [Cancel](#)

Use “**Column level classification**” options to characterize all country- / region-type schema elements as a standard classification.

## Objective: Purview | Understand Lineage

Data lineage describes provenance {i.e., where data comes from, where it ends up and what happens to it along the way}.

Understanding data lineage enables stakeholders to answer the following question:

- **Accountability** (aka Impact Analysis) ... “who and what would be affected by a change to or problem with Data Asset X?”

In this section, we will configure Purview and successfully respond to this question.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Factory
- Data Share
- Purview
- SQL

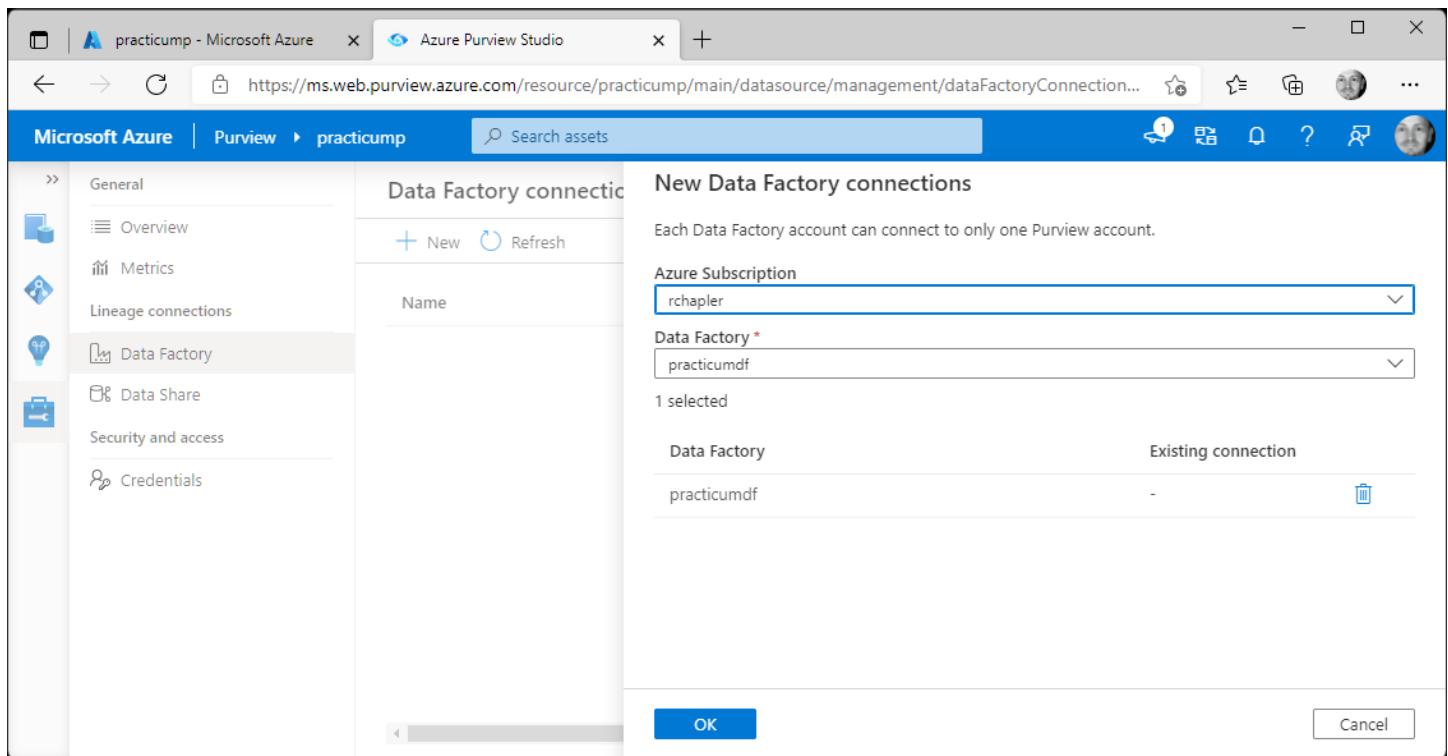
### Configure Data Factory

Navigate to your instance of Purview in the Azure portal.

Click the “Open Purview Studio” button.

Click the **Management** icon on the left-hand navigation and then “**Data Factory**” from the resulting menu.

Click the “**+ New**” button on the resulting page.



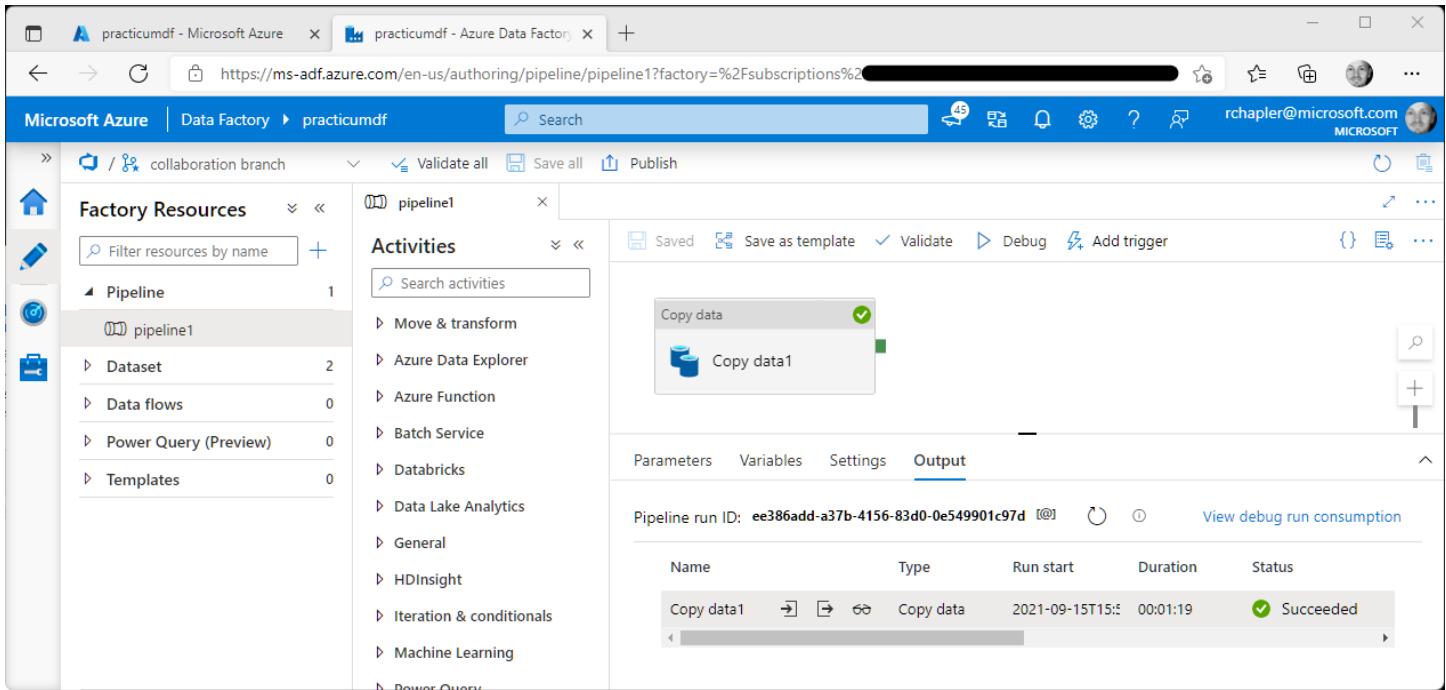
On the “New Data Factory connections” pop-out, enter values for the following items:

Azure Subscription	Select your subscription
Data Factory	Select your instance of data factory

Click the **OK** button. Confirm connection.

## Confirm Success

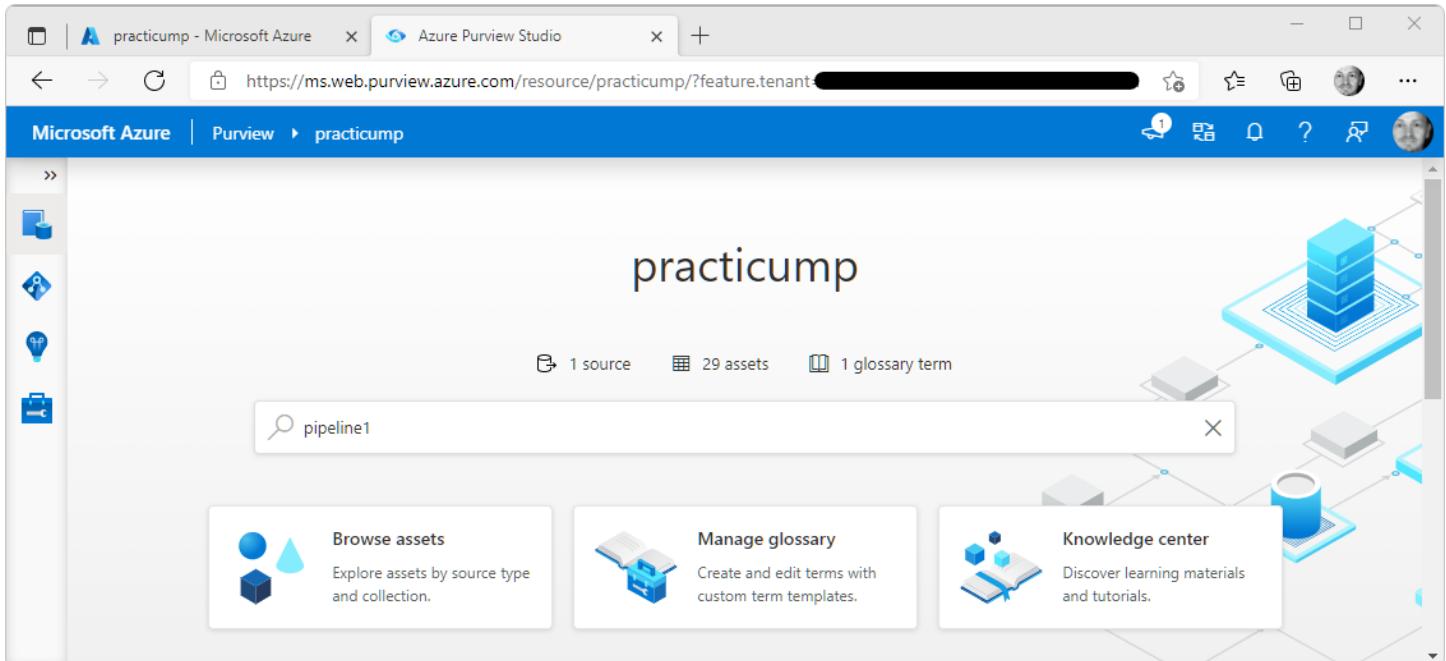
Purview will automatically detect data movement and capture data about a pipeline when it is executed in a connected data factory. Navigate to your instance of Data Factory and execute your sample pipeline.



The screenshot shows the Azure Data Factory pipeline execution results. The pipeline named 'pipeline1' contains one activity, 'Copy data1'. The activity has a green checkmark indicating success. Below the activities, the 'Output' tab is selected, showing a table of the run details:

Name	Type	Run start	Duration	Status
Copy data1	Copy data	2021-09-15T15:50:00Z	00:01:19	Succeeded

After successful execution, navigate to your instance of Purview.



The screenshot shows the Azure Purview Studio interface. The search bar at the top contains the text 'pipeline1'. The main area displays a network diagram of data assets. A callout box highlights the 'Browse assets' button, which is described as 'Explore assets by source type and collection.'

Search for "pipeline1".

The screenshot shows the Azure Purview Studio interface. The top navigation bar includes 'practicump - Microsoft Azure', 'Azure Purview Studio', and a search bar with the query 'pipeline1'. The main content area is titled 'Search results for pipeline1' and displays two results:

- Copy data1**  
Azure Data Factory Copy Activity  
/subscriptions/[REDACTED]/resourceGroups/practicumrg/providers/Microsoft.Dat...
- pipeline1**  
Azure Data Factory Pipeline  
/subscriptions/[REDACTED]/resourceGroups/practicumrg/providers/Microsoft.Dat...

On the left, there is a sidebar with filters for 'Collection' (selected 'practicump'), 'Classification', 'Contact', and 'Content type'. The bottom of the page shows pagination controls: '< Previous', 'Page 1 of 1', and 'Next >'.

Click on the “Copy data1” result.

Click on the **Lineage** tab.

The screenshot shows the Microsoft Purview Studio interface. The top navigation bar includes 'practicump - Microsoft Azure' and 'Azure Purview Studio'. The main title bar says 'pipeline1'. The left sidebar has icons for Data catalog, Purview, and practicump. The search bar also shows 'pipeline1'. The main content area displays a 'Copy data1' activity under 'Azure Data Factory Copy Activity'. It shows two 'Product' datasets connected by a curved arrow. The activity card includes details: Factory: practicumpdf, Pipeline: pipeline1. Below the card are 'See details' and 'Open in Azure Data Factory' buttons. To the left is a sidebar for 'Input datasets' and 'Columns', showing a tree view of columns: Product > ProductNumber > ListPrice. The top right of the page indicates it was updated on September 15, 2021, at 4:00 PM UTC by Azure Data Factory pipeline.

Continue to drill-through elements to better understand the capture of Lineage data.

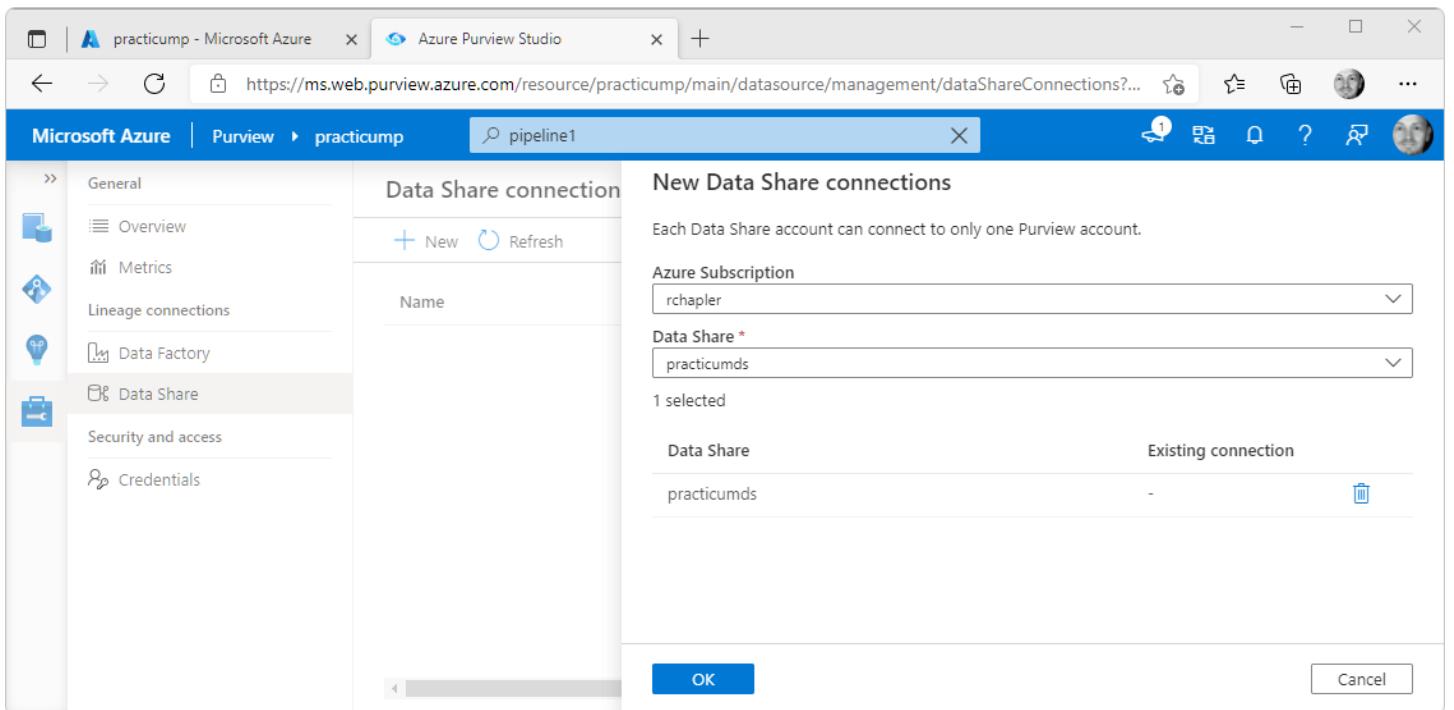
## Configure Data Share

Navigate to your instance of Purview in the Azure portal.

Click the “Open Purview Studio” button.

Click the **Management** icon on the left-hand navigation and then “Data Share” from the resulting menu.

Click the “+ New” button on the resulting page.



On the “**New Data Factory connections**” pop-out, enter values for the following items:

Azure Subscription	Select your subscription	Self-expl
Data Share	Select your instance of data share	Select th

Click the **OK** button.

Article: [Connect to Azure Data Share - Azure Purview | Microsoft Docs](#)

Note from this article... “For Data Share assets to show in Purview, a snapshot job must be run after you connect your Data Share to Purview”

# Monitoring

## Objective: Synapse | Audit Data Usage

Follow the instructions in this section to **support regular audit of and alerting on key data assets**.

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Synapse

### Synapse | Auditing

Navigate to the **Overview** page in Synapse, Dedicated SQL Pool.

The screenshot shows the Microsoft Azure portal interface for a Dedicated SQL pool named "practicumdsp" within the "practicumsaw/practicumdsp" resource group. The main content area displays the "Essentials" section, which includes details such as the workspace name ("practicumsaw"), performance level ("DW100c"), connection strings, and maintenance schedule. Below this, there are sections for "Notifications (0)", "Features (4)", and "Tasks (2)". Under "Features", two cards are visible: "Transparent data encryption" (NOT CONFIGURED) and "Auditing" (NOT CONFIGURED). The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Settings, Workload management, Maintenance schedule, Geo-backup policy, Connection strings, Properties, Locks, Security, Auditing, Data Discovery & Classification, Dynamic Data Masking, and Security Center.

Click the **Auditing** button.

practicumdsp (practicumsaw/practicumdsp) | Auditing

Search (Ctrl+ /)

Save Discard View audit logs Feedback

Overview Activity log Access control (IAM) Tags

Settings Workload management Maintenance schedule Geo-backup policy Connection strings Properties Locks

If Azure SQL Auditing is enabled on the workspace, it will always apply to the SQL Pool, regardless of the SQL Pool settings.

View workspace settings

Workspace-level Auditing: **Disabled**

Azure SQL Auditing

Azure SQL Auditing tracks SQL Pool events and writes them to an audit log in your Azure storage account. [Learn more about Azure SQL Auditing](#)

Enable Azure SQL Auditing

Storage details >

practicumsa

Flip “Enable Azure SQL Auditing” switch to **ON** position. Click to configure “Storage details”.

Storage settings

Subscription: rchapler

\*Storage account: practicumsa

Retention (Days): 0

Storage access key: Primary

OK

Save changes.

Click “View audit logs”.

Audit records - Microsoft Azure

Microsoft Azure (Preview)

Dashboard > practicumdsp (practicumsaw/practicumdsp) >

## Audit records

Refresh Filter Log Analytics View dashboard

Click here to learn more about methods for viewing & analyzing audit records.

Audit source: Workspace audit

Event time (UTC)	Principal name	Event type	Action status
2/3/2021 8:58:30 PM	##MS_InstanceCertificate##	BATCH COMPLETED	Succeeded
2/3/2021 8:56:26 PM	##MS_InstanceCertificate##	BATCH COMPLETED	Succeeded
2/3/2021 8:56:26 PM	##MS_InstanceCertificate##	DATABASE AUTHENTICATION SUCCEEDED	Succeeded

Click into one of “**BATCH COMPLETED**” audit records.

Audit record - Microsoft Azure

Microsoft Azure (Preview)

Search resources, services, and docs (G+)

Dashboard > practicumdsp (practicumsaw/practicumdsp) > Audit records > Audit record

Event time (UTC)  
2/8/2021 7:32:50 PM

Event type  
BATCH COMPLETED

Server name  
practicumsaw

Database name  
practicumdsp

Application name  
SynapseSqlEditor

Principal name  
rchapler@microsoft.com

Client IP  
76.121.194.132

Status  
Succeeded

---

STATEMENT

```
SELECT
    s.name AS [schema_name],
    o.name AS [object_name], o.type AS [object_type], o.type_desc AS
    [object_type_desc],
    c.name AS [column_name], TYPE_NAME(c.system_type_id) AS
    [column_type]
FROM sys.schemas s
LEFT OUTER JOIN sys.all_objects o
    ON s.schema_id = o.schema_id
LEFT OUTER JOIN sys.all_columns c
    ON c.object_id = o.object_id
```

Expand statement

# Appendix

Appendix topics describe instantiation of basic resources necessary to complete other exercises.

Sections are not sequential.

Instantiate only those resources required by a specific exercise.

## Application Registration (aka Service Principal)

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Lake (with container)

Navigate to “Azure Active Directory” > “App Registrations”.

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft logo, a search bar, and various navigation icons. The main title is "Microsoft - Microsoft Azure". Below the title, the URL is https://ms.portal.azure.com/#blade/Microsoft\_AAD\_IAM/ActiveDirectoryMenuBlade/Register... The user's email, rchapler@microsoft.com, is visible in the top right corner.

The left sidebar has a dark theme and lists several categories under "Manage": Overview, Preview features, Diagnose and solve problems, Manage (with sub-options: Users, Groups, External Identities, Roles and administrators, Administrative units, Enterprise applications, Devices), App registrations (which is currently selected and highlighted in grey), and Identity Governance.

The main content area is titled "Microsoft | App registrations" and "Azure Active Directory". It features a "New registration" button, "Endpoints", "Troubleshooting", "Refresh", "Download", "Preview features", and a "..." menu. A warning message states: "If you are building an application for external users that will be distributed by Microsoft, you must register as a first party application to meet all security, privacy, and compliance policies. Read our decision guide".

Below the warning, there are tabs for "All applications", "Owned applications" (which is underlined in blue to indicate it is selected), and "Deleted applications (Preview)". A search bar says "Start typing a name or Application ID to filter these results". A message below the tabs says "This account isn't listed as an owner of any applications in this directory." A blue button labeled "View all applications in the directory" is present.

Click the “+ New Registration” button.

The screenshot shows the 'Register an application' page in the Microsoft Azure portal. The URL is https://ms.portal.azure.com/#blade/Microsoft\_AAD\_IAM/ActiveDirectoryMenuBlade/RegisterApplication/0. The page has a dark theme. At the top, there's a navigation bar with 'Microsoft Azure (Preview)', a search bar, and a user profile for rchapler@microsoft.com.

**Name:** The input field contains 'practicum'. A green checkmark icon is visible to the right of the input field.

**Supported account types:**

Who can use this application or access this API?

- Accounts in this organizational directory only (Microsoft only - Single tenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant)
- Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- Personal Microsoft accounts only

[Help me choose...](#)

**Redirect URI (optional):**

We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web  e.g. https://example.com/auth

Register an app you're working on here. Integrate gallery apps and other apps from outside your organization by adding from [Enterprise applications](#).

By proceeding, you agree to the Microsoft Platform Policies [\[link\]](#)

**Register**

On the “Register an application” page, enter values for the following items:

Name	Self-Explanatory
------	------------------

No additional changes are required.

Click the **Register** button.

The screenshot shows the Microsoft Azure portal interface. The left sidebar has a dark theme with the following navigation items under 'Manage': Overview, Quickstart, Integration assistant, Branding, Authentication, Certificates & secrets, Token configuration, API permissions, and Expose an API. The main content area displays the 'practicumar' application registration. It includes a search bar, a toolbar with 'Delete', 'Endpoints', and 'Preview features' buttons, and a feedback banner. The 'Essentials' section contains the following details:

Setting	Value	Action
Display name	practicumar	Client credentials Add a certificate or secret
Application (client) ID	[REDACTED]	Redirect URIs Add a Redirect URI
Object ID	062d3984-d6b4-4f83-abcd-35d68d49a955	Application ID URI Add an Application ID URI
Directory (tenant) ID	[REDACTED]	Managed application in local directory practicumar
Supported account types	My organization only	

Make note of the “Application (client) ID” and “Directory (tenant) ID” values for use in the next section.

## Client Identifier

Navigate to the App Registration, **Overview** page.

The screenshot shows the Microsoft Azure portal interface, identical to the previous one but with different application registration details. The left sidebar and main content area are the same, but the 'Essentials' section now shows:

Setting	Value	Action
Display name	practicumar	Client credentials 0 certificate, 1 secret
Application (client) ID	93de551e-b998-4b1e-8930-3d9d8ac4757e	Redirect URLs Add a Redirect URI
Object ID	062d3984-d6b4-4f83-abcd-35d68d49a955	Application ID URI Add an Application ID URI
Directory (tenant) ID	[REDACTED]	Managed application in local directory practicumar
Supported account types	My organization only	

Copy the “Directory (tenant) ID” and “Application (client) ID” values.

## Key Vault Secrets, Tenant and Client Identifiers

Open a new tab and navigate to your instance of Key Vault.

Click **Secrets** in the **Settings** group of the left-hand navigation.

Click the “**+ Generate/Import**” button.

The screenshot shows the 'Create a secret' page in the Microsoft Azure (Preview) portal. The 'Name' field is set to 'practicumar-clientid'. The 'Value' field contains several redacted dots. The 'Enabled' switch is set to 'Yes'. There are also fields for 'Content type (optional)', 'Set activation date', 'Set expiration date', and 'Tags' (0 tags). At the bottom is a prominent blue 'Create' button.

On the “**Create a secret**” page, enter values for the following items:

Name	Self-Explanatory
Value	Paste the previously copied value

No additional changes are required.

Click the **Create** button.

Repeat for Tenant Id.

## Client Secret

In the **Manage** grouping, click “**Certificates & secrets**”.

Navigate to “**Certificates & secrets**” in the **Manage** group in the left-hand navigation.

The screenshot shows the Microsoft Azure portal interface. The URL in the address bar is [https://ms.portal.azure.com/#blade/Microsoft\\_AAD\\_RegisteredApps/ApplicationMenuBlade/Credentials/q...](https://ms.portal.azure.com/#blade/Microsoft_AAD_RegisteredApps/ApplicationMenuBlade/Credentials/q...). The top navigation bar includes the Microsoft Azure logo, a search bar, and a user profile for rchapler@microsoft.com.

The main content area is titled "practicumar | Certificates & secrets". On the left, there's a sidebar under "Manage" with the following items:

- Overview
- Quickstart
- Integration assistant
- Branding
- Authentication
- Certificates & secrets** (selected)
- Token configuration
- API permissions
- Expose an API
- App roles
- Owners
- Roles and administrators | Preview
- Manifest

**Certificates**

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

[Upload certificate](#)

Thumbprint	Start date	Expires	Certificate ID
No certificates have been added for this application.			

**Client secrets**

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

[New client secret](#)

Description	Expires	Value	Secret ID
No client secrets have been created for this application.			

Click the “+ New client secret” button, and then in the resulting “Add a client secret” popout, click the **Add** button.

**Certificates**

Certificates can be used as secrets to prove the application's identity when requesting a token. Also can be referred to as public keys.

**Client secrets**

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

Description	Expires	Value	Secret ID
Password uploaded on Fri Sep ...	3/24/2022	[Copied]	[Copied]

Copy Value.

## Key Vault Secret, Client Secret

Open a new tab and navigate to your instance of Key Vault.  
Click **Secrets** in the **Settings** group of the left-hand navigation.  
Click the “+ Generate/Import” button.

The screenshot shows the Microsoft Azure portal's "Create a secret" interface. The "Upload options" dropdown is set to "Manual". The "Name" field contains "practicumar-clientsecret". The "Value" field is redacted with dots. The "Content type (optional)" field is empty. There are two unchecked checkboxes for "Set activation date" and "Set expiration date". The "Enabled" switch is set to "Yes". The "Tags" section shows "0 tags". At the bottom is a blue "Create" button.

On the “Create a secret” page, enter values for the following items:

Name	Self-Explanatory
Value	Paste the previously copied value

No additional changes are required.

Click the **Create** button.

## Permissions, Data Lake

Navigate to your instance of data lake, your container and then “Access Control (IAM)” in left-hand navigation.

Click the “+ Add” button and then “Add role assignment” in the resulting drop-down.

The screenshot shows the Microsoft Azure (Preview) portal with the URL [https://ms.portal.azure.com/#blade/Microsoft\\_Azure\\_Storage/ContainerMenuBlade/accesscontrol/storage...](https://ms.portal.azure.com/#blade/Microsoft_Azure_Storage/ContainerMenuBlade/accesscontrol/storage...). The main page displays the 'practicumdlc | Access Control (IAM)' blade for a storage container. On the left, there's a sidebar with 'Overview', 'Diagnose and solve problems', 'Access Control (IAM)', 'Settings' (with options like 'Shared access tokens', 'Manage ACL', 'Access policy', 'Properties', 'Metadata', and 'Editor (preview)'), and a search bar. The 'Access Control (IAM)' tab is selected. In the center, there's a table showing 'Number of role assignments for this subscription' (13 total, 2000 limit). The 'Role assignments' tab is active, showing a list of roles: Contributor, Owner, Reader, Storage Blob Data Contributor, Storage Blob Data Reader, and User Access Administrator. A modal window titled 'Add role assignment' is open, allowing the user to assign a 'Storage Blob Data Reader' role to a 'User, group, or service principal' named 'practicumar'. The 'Selected members' list contains this single entry. At the bottom of the modal are 'Save' and 'Discard' buttons.

On the “Add role assignment” pop-out, enter values for the following items:

Role	Select “Storage Blob Data Reader”
Assign access to	Confirm default selection, “User, group, or service principal”
Select	Search for and select your application registration

Click the **Save** Button.

## CLI

Browse to [Install the Azure CLI for Windows | Microsoft Docs](#)

The screenshot shows a web browser displaying the Microsoft Azure documentation page for installing the Azure CLI on Windows. The URL is <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?tabs=azure-cli>. The page title is "Install Azure CLI on Windows". On the left, there's a navigation sidebar with links for Overview, Install, Update, and Run, as well as specific sections for Windows, macOS, Linux, and Bash. The "Install - Windows" link is highlighted. The main content area starts with a heading "Install or update" and a note about using the MSI distributable for Windows. It includes a screenshot of the Microsoft Installer (MSI) interface with two tabs: "Microsoft Installer (MSI)" and "Microsoft Installer (MSI) with Command". Below the screenshot, there's a note about clicking "Yes" if prompted to make changes. A section titled "Azure CLI current version" provides a link to download the "Current release of the Azure CLI". On the right side, there are "Is this page helpful?" buttons for "Yes" and "No", and a "In this article" sidebar with links for Install or update, Run the Azure CLI, Troubleshooting, Uninstall, and Next Steps.

Download and install the current release of the Azure CLI.

## Data Explorer

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Resource Group

The screenshot shows the 'Create an Azure Data Explorer Cluster' page in the Microsoft Azure (Preview) portal. The page is titled 'Create an Azure Data Explorer Cluster' and includes tabs for Basics, Scale, Configurations, Security, Network, Tags, and Review + Create. The Basics tab is selected.

**PROJECT DETAILS**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*: rchabler

Resource group \*: practicumrg

**CLUSTER DETAILS**

Cluster name \*: practicumdec

Region \*: West US 2

Enable performance update (EngineV3):

**COMPUTE SPECIFICATION**

Workload \*: Compute optimized

Size \*: Extra Small (2 cores)

Compute specifications \*: Standard\_D11\_v2

Availability zones: (none)

**Buttons:**

- Review + Create
- Next : Scale >

On the “Create an Azure Data Explorer Cluster” page, enter values for the following form items:

Subscription	Select your subscription
Resource Group	Select your Resource Group
Cluster Name	Self-Explanatory
Region	Select the value used during Resource Group creation
Enable Performance...	Leave checked
Workload	Select “Compute optimized” from the drop-down
Size	Select “Extra Small (2 cores)” from the drop-down
Compute Specifications	This should be auto populated with “Standard_D11_v2” based on the Workload and Size selections
Availability Zones	Confirm default selection, “(none)”

Review settings on remaining tabs {e.g., Tags}.

No additional changes are required.

Click the “Review + create” button, validate, and then click the **Create** button.

## Add Database

The screenshot shows the Microsoft Azure (Preview) portal. On the left, there's a sidebar for the 'practicumdec' cluster, which includes options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Permissions, Query, Settings, Scale up, Scale out, and Configurations. In the center, there's a 'Create new database' dialog titled 'Azure Data Explorer Database'. It has fields for 'Database name' (set to 'practicumdec'), 'Retention period (in days)' (set to '3650'), and 'Cache period (in days)' (set to '31'). There are also checkboxes for 'Unlimited days for retention period' and 'Unlimited days for cache period'. At the bottom right of the dialog is a blue 'Create' button.

In the newly created Data Explorer Cluster, click the “+ Add database” button.

On the “Create an Azure Data Explorer Database” pop-out, enter values for the following form items:

<b>Database Name</b>	Self-Explanatory
<b>Retention Period (in days)</b>	Confirm the default value, 3650
<b>Cache Period (in days)</b>	Confirm the default value, 31

Click the **Create** button.

## Sample #1, Product Table

Navigate to the Data Explorer database.

Click **Query** in the left-hand navigation.

The screenshot shows the Microsoft Azure Data Explorer Cluster interface. On the left, there's a sidebar with options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Permissions, and Query (which is selected). The main area shows a query editor with the following KQL code:

```
.create table Product ( ProductNumber: string, ListPrice: decimal )
```

Below the code, a table named "Table 1" is displayed with one record:

TableName	Schema	DatabaseName	Folder	DocString
Product	{"Name": "Product", "OrderedColumns": [{"Name": "Pr..."}]}	practicumdec		

Execute the following KQL query to create a new table named Product:

```
.create table Product ( ProductId:int, ProductNumber: string, Name:string, ListPrice: decimal, ModifiedDate:datetime )
```

## Sample #2, StormEvents Data

Follow the instructions in the “Quickstart: Ingest sample data into Azure Data Explorer” article (<https://docs.microsoft.com/en-us/azure/data-explorer/ingest-sample-data>) to populate sample data that we can surface in Power Apps.

The screenshot shows the Azure Data Explorer interface. The left sidebar has Data, Query (selected), and Dashboards (Preview) options. The main area shows a query editor with the following KQL code:

```
.create table StormEvents (StartTime: datetime, EndTime: datetime, EpisodeId: int, EventId: int, State: string, EventType: string)
.ingest into table StormEvents 'https://kustosamplefiles.blob.core.windows.net/samplef...
StormEvents | sort by StartTime desc | take 10
```

Below the code, a table named "Table 1" is displayed with 10 records:

StartTime	EndTime	EpisodeId	EventId	State	EventType
2007-12-31 23:53:00.0000	2007-12-31 23:53:00.0000	12,037	65,839	CALIFORNIA	High Wind
2007-12-31 23:53:00.0000	2007-12-31 23:53:00.0000	12,037	65,838	CALIFORNIA	High Wind
2007-12-31 22:30:00.0000	2007-12-31 23:59:00.0000	12,950	71,590	MICHIGAN	Winter Storm
2007-12-31 22:30:00.0000	2007-12-31 23:59:00.0000	12,950	71,588	MICHIGAN	Winter Storm
2007-12-31 22:30:00.0000	2007-12-31 23:59:00.0000	12,950	71,589	MICHIGAN	Winter Storm

Familiarize yourself with the resulting data for use in later sections.

## Data Factory

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Key Vault
- Resource Group

The screenshot shows the Microsoft Azure portal with the URL <https://ms.portal.azure.com/#create/Microsoft.DataFactory>. The page title is "Create Data Factory". The navigation bar includes "Microsoft Azure (Preview)" and the user's email "rchapler@microsoft.com". The main content area is titled "Create Data Factory" and has a sub-header "Project details". It asks to select a subscription and resource group. The "Subscription" dropdown is set to "rchapler" and the "Resource group" dropdown is set to "practicumrg". Below this, under "Instance details", the "Region" is set to "West US 2", the "Name" is "practicumdf", and the "Version" is "V2". At the bottom, there are buttons for "Review + create", "< Previous", and "Next : Git configuration >".

On the “Create Data Factory” page, **Basics** tab, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your Resource Group
Region	Select the value used during Resource Group creation
Name	Self-Explanatory
Version	Confirm default selection, “V2”

Click the “Next: Git configuration >” button.

The screenshot shows the Microsoft Azure (Preview) portal with the URL <https://ms.portal.azure.com/#create/Microsoft.DataFactory>. The page title is "Create Data Factory". The "Git configuration" tab is active. A note states: "Azure Data Factory allows you to configure a Git repository with either Azure DevOps or GitHub. Git is a version control system that allows for easier change tracking and collaboration." Below this is a link to "Learn more about Git integration in Azure Data Factory". A checkbox labeled "Configure Git later" is checked. At the bottom, there are buttons for "Review + create", "< Previous", and "Next : Networking >".

On the “Create Data Factory” page, “Git configuration” tab, enter values for the following items:

---

Configure Git later	Check the box
---------------------	---------------

---

Review settings on remaining tabs {e.g., Networking, Advanced, Tags}.

No additional changes are required.

Click the “Review + create” button, review configuration, and then click the **Create** button.

## Add Access Policy to Key Vault

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Key Vault

Navigate to your instance of Key Vault.

The screenshot shows the Microsoft Azure Key Vault settings page for the 'practicumkv' vault. The left sidebar shows navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, and Settings (Keys, Secrets, Certificates, Access policies). The main content area is titled 'Essentials' and displays resource group ('practicumrg'), location ('West US 2'), subscription ('rchapler'), and directory information ('Vault URI: https://practicumkv.vault.azure.net/'). The 'Access control (IAM)' section is visible but not currently selected.

Click “Access Policies” in the **Settings** group of the left-hand navigation.

The screenshot shows the Microsoft Azure Key Vault Access policies page for the 'practicumkv' vault. The left sidebar shows the 'Access policies' option selected under 'Settings'. The main content area includes sections for enabling access to Azure VM deployment, Resource Manager deployment, and Disk Encryption, and a permission model selector between 'Vault access policy' (selected) and 'Azure role-based access control'. A prominent blue link '+ Add Access Policy' is located below these settings.

Click the “+ Add Access Policy” link.

**Add access policy**

Configure from template (optional)

Key permissions: 0 selected

Secret permissions: 2 selected

Certificate permissions: 0 selected

Select principal \*: None selected

Authorized application: None selected

Add

Principal

Select a principal

practicumdf  
a383d929-ecd9-47cf-812c-3053a7dbff87  
Selected

Selected items

practicumdf  
a383d929-ecd9-47cf-812c-3053a7dbff87

Select

On the “Add access policy” page, enter values for the following items:

**Secret Permissions** Select **Get** and **List** in the drop-down

**Select Principal** Search for, and then **Select** the managed identity for your Data Factory

Click the **Add** button.

practicumkv | Access policies

Key vault

Search (Ctrl+ /)

Save Discard Refresh

Please click the 'Save' button to commit your changes.

Enable Access to:

Azure Virtual Machines for deployment

Azure Resource Manager for template deployment

Azure Disk Encryption for volume encryption

Permission model:  Vault access policy

Azure role-based access control

+ Add Access Policy

Current Access Policies

Name	Email	Key Permissions	Secret Permissions	Certificate Permissions	Action
practicumdf		0 selected	2 selected	0 selected	Delete

Click the **Save** button.

## Add Permissions to Data Explorer

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Explorer

Navigate to your instance of Data Explorer.

Click **Permissions** in the left-hand navigation.

Click “**+ Add**” on the resulting page.

Search for and then click to **Select** your instance of Data Factory.

## Data Lake

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Resource Group
- Storage Account

The screenshot shows the Microsoft Azure portal interface with a dark theme. The title bar reads "Create storage account - Microsoft Azure". The URL in the address bar is "https://ms.portal.azure.com/#create/Microsoft.StorageAccount-ARM". The top navigation bar includes "Microsoft Azure (Preview)", a search bar, and user information for "rchapler@microsoft.com". Below the header, the breadcrumb navigation shows "Dashboard > practicumrg > New > Create storage account".

The main content area is titled "Create storage account". It features a tab navigation bar with "Basics" selected, followed by "Networking", "Data protection", "Advanced", "Tags", and "Review + create".

**Basics** tab content:

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below.  
[Learn more about Azure storage accounts](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription: rchapler  
Resource group: practicumrg  
Create new

**Instance details**

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

Storage account name: practicuml (highlighted with a green checkmark)

Location: (US) West US 2

Performance: Standard (radio button selected)

Account kind: StorageV2 (general purpose v2)

Replication: Read-access geo-redundant storage (RA-GRS)

At the bottom, there are buttons for "Review + create", "< Previous", and "Next : Networking >".

On the “Create Storage Account” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your Resource Group
Storage Account Name	Self-Explanatory
Location	Select the value used during Resource Group creation
Performance	Confirm default radio button selection, “Standard”
Account Kind	Confirm default selection, “StorageV2 (general purpose v2)”
Replication	Confirm default selection, “Read-access geo-redundant storage (RA-GRS)”

Navigate to the **Advanced** tab.

The screenshot shows the 'Create storage account' wizard in Microsoft Azure. The 'Advanced' tab is selected. The configuration includes:

- Security**: Secure transfer required is set to Enabled, and Allow shared key access is set to Enabled. Minimum TLS version is set to Version 1.2, and Infrastructure encryption is set to Disabled. A note indicates that sign up is required for infrastructure encryption on a per-subscription basis.
- Blob storage**: Allow Blob public access is set to Enabled, Blob access tier (default) is set to Hot, and NFS v3 is set to Disabled. A note indicates that sign up is required for NFS v3.
- Data Lake Storage Gen2**: Hierarchical namespace is set to Enabled.

At the bottom, there are buttons for 'Review + create' (highlighted in blue), '< Previous', and 'Next : Tags >'.

Click on the **Enabled** radio button in the “Data Lake Storage Gen2” grouping.

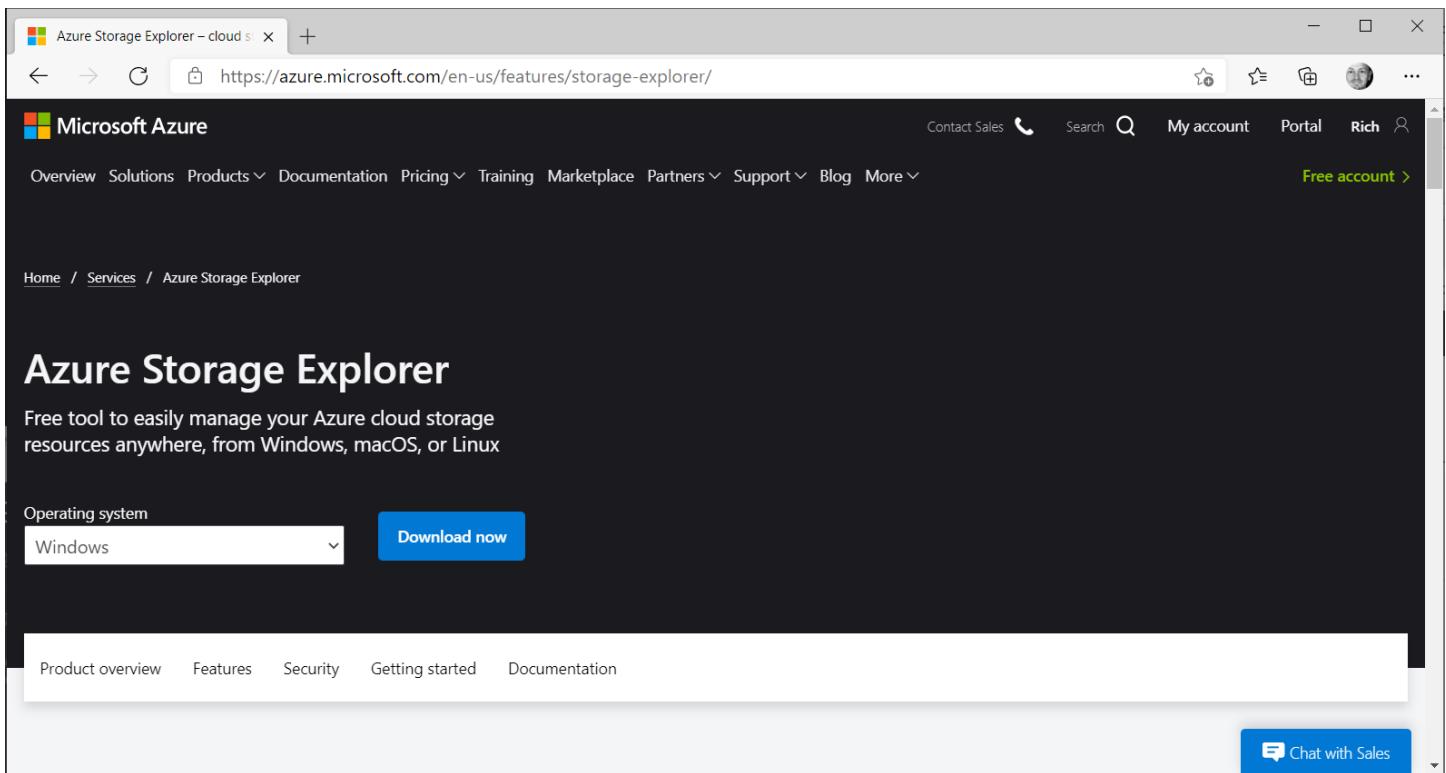
Review settings on remaining tabs {e.g., Networking, Data Protection, Tags}.

No additional changes are required.

Click the “Review + create” button, review configuration, and then click the **Create** button.

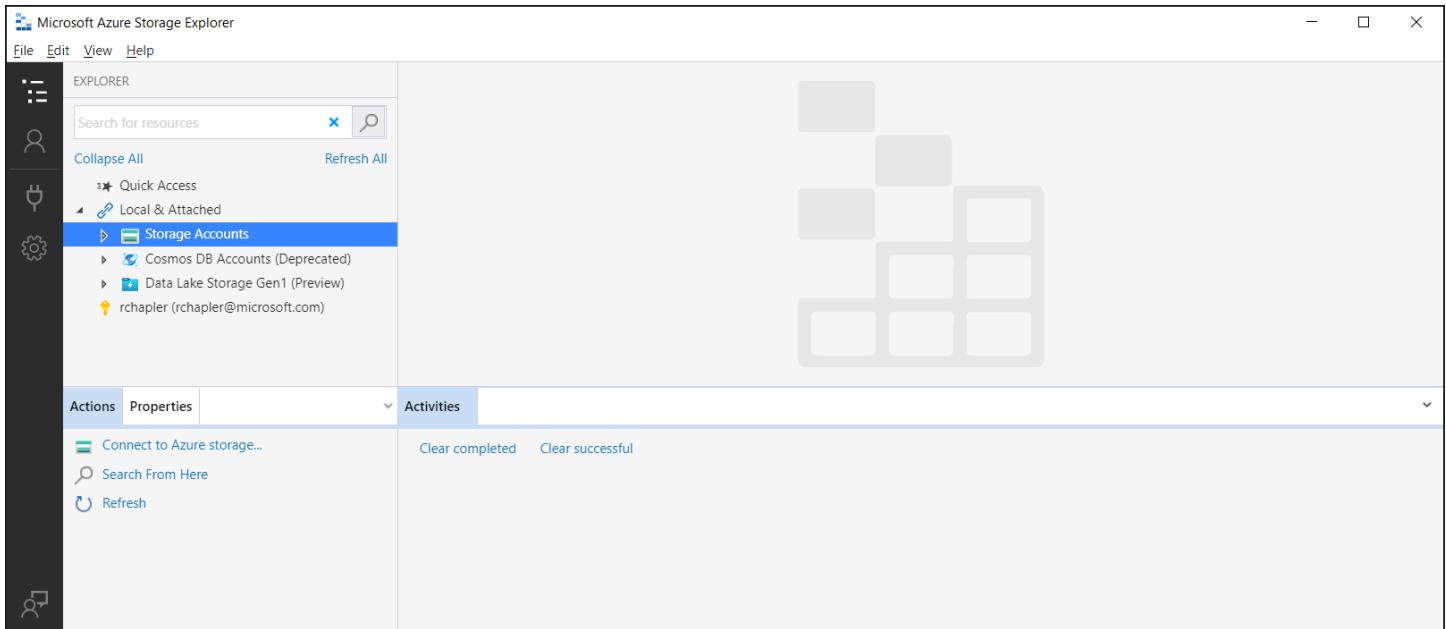
## Storage Explorer

Download and install the Microsoft Azure Storage Explorer app ([Azure Storage Explorer – cloud storage management | Microsoft Azure](#)).



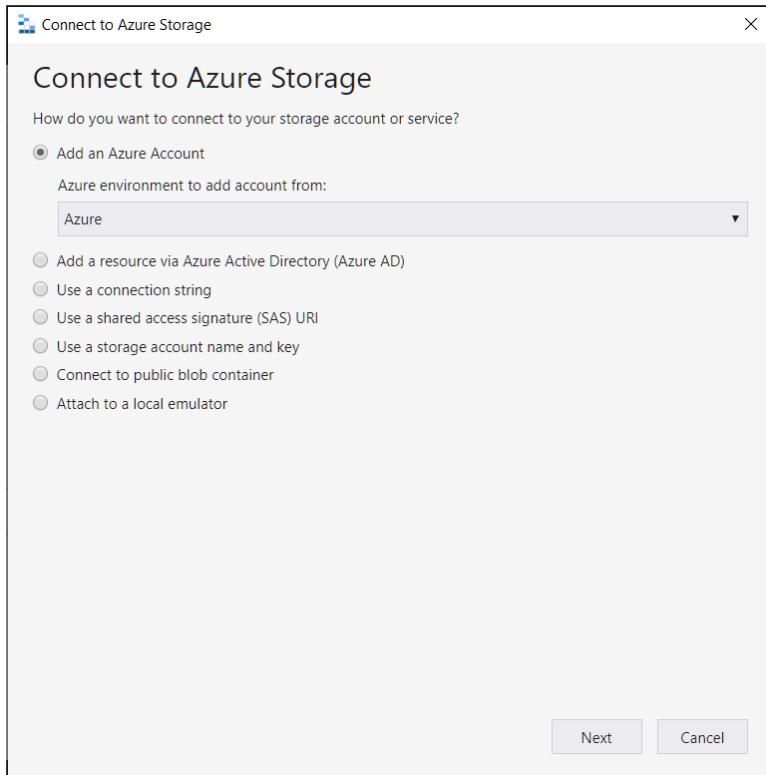
The screenshot shows the Azure Storage Explorer landing page. At the top, there's a navigation bar with links like Overview, Solutions, Products, Documentation, Pricing, Training, Marketplace, Partners, Support, Blog, More, and a prominent 'Free account' button. Below the navigation is a breadcrumb trail: Home / Services / Azure Storage Explorer. The main title is 'Azure Storage Explorer' with the subtitle 'Free tool to easily manage your Azure cloud storage resources anywhere, from Windows, macOS, or Linux'. There's a dropdown for 'Operating system' set to 'Windows' with a 'Download now' button. Below this is a navigation bar with links: Product overview, Features, Security, Getting started, Documentation, and a 'Chat with Sales' button.

Launch the app and provide Azure credentials.



The screenshot shows the Microsoft Azure Storage Explorer application window. The left sidebar has icons for Home, Quick Access, Local & Attached, Storage Accounts, and a user profile. The 'Storage Accounts' section is expanded, showing 'Cosmos DB Accounts (Deprecated)', 'Data Lake Storage Gen1 (Preview)', and an email entry for 'rchapler (rchapler@microsoft.com)'. The main pane is titled 'EXPLORER' with a search bar and a refresh button. It displays a grid of placeholder icons representing storage accounts. At the bottom, there are 'Actions' and 'Properties' tabs, and an 'Activities' section with links to 'Connect to Azure storage...', 'Search From Here', and 'Refresh'.

Click the "Connect to Azure storage..." link.



On the resulting pop-up, click the “**Add an Azure Account**” radio button. Click the **Next** button. Authenticate with Azure credentials.

## Container

Continue with the “**Microsoft Azure Storage Explorer**” app. Use the search bar to focus on your Storage Account.

Right-click on “**Blob Containers**” and click “**Create Blob Container**” in the resulting pop-up menu. Enter a meaningful name.

## Sample Data

Download sample files from a site like [Sample CSV Files Download - Get Examples Instantly \(filesamples.com\)](http://filesamples.com)

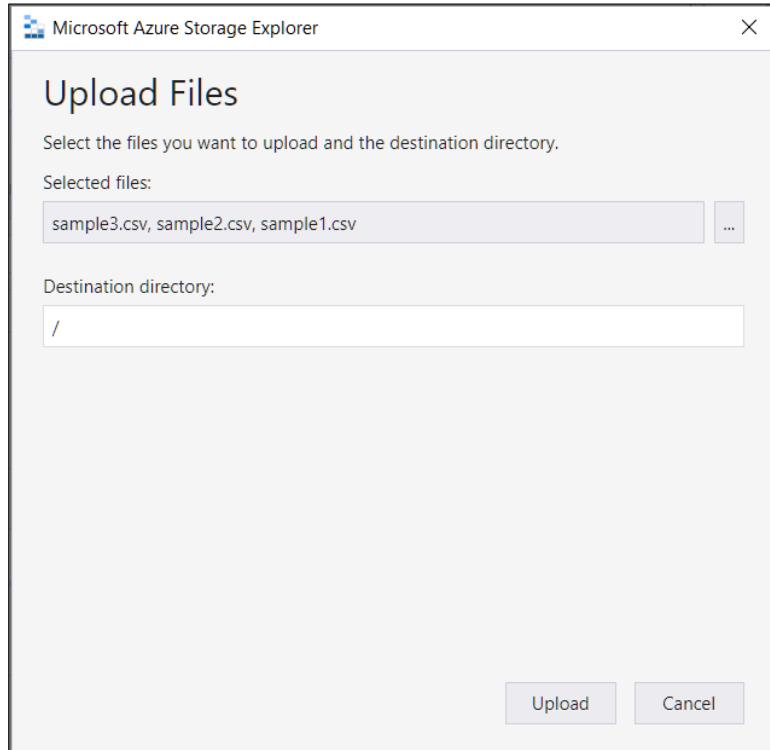
The screenshot shows a Microsoft Edge browser window with two tabs open: "practicumsa - Microsoft Azure" and "Sample CSV Files Download - Ge...". The current page is "https://filesamples.com/formats/csv". The main content is titled "Sample CSV Files Download" and features a "CSV" icon and the text "Comma Separated Values". It describes the purpose of the page: "Below you will find a selection of sample .csv document files for you to download. On the right there are some details about the file such as its size so you can best decide which one will fit your needs." Three CSV files are listed:

- sample1.csv: CSV / 502.00 B, Download button
- sample2.csv: CSV / 55.56 KB, Download button
- sample3.csv: CSV / 723.00 B, Download button

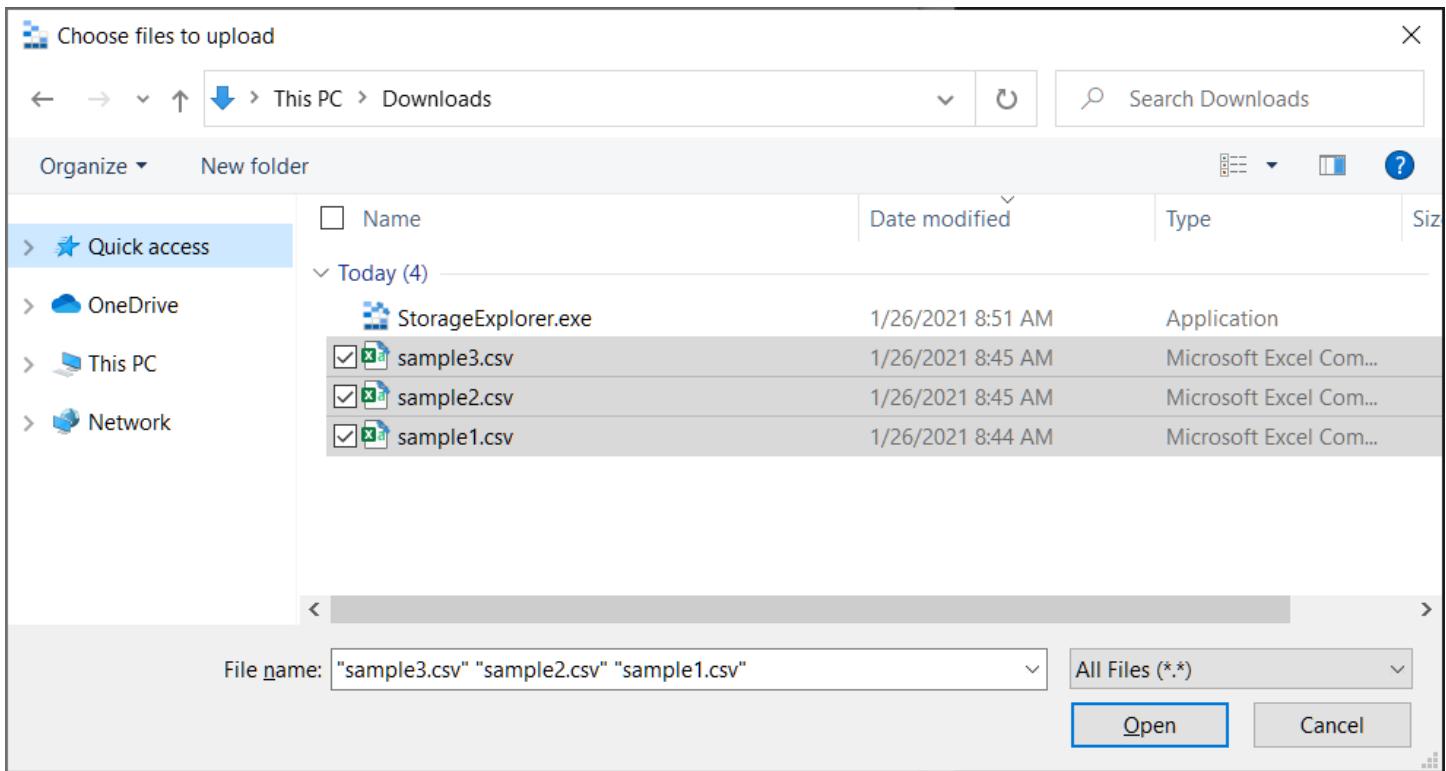
At the bottom, there are navigation buttons for "sample3.csv", "sample2.csv", "sample1.csv", and a "Show all" link.

Return to the “Microsoft Azure Storage Explorer” app and navigate to your container. Click the **Upload** button and select “Upload Files” in the resulting pop-up menu.

Click the ellipses button to the right of the “Selected Files” box.



Navigate to **Downloads** and select your sample data files.



Click the **Open** button.

On the “Upload Files” pop-up, click the **Upload** button. Allow time for file transfer.

A screenshot of the Microsoft Azure Storage Explorer application. The left sidebar shows a file structure under "practicumdl": "rchapler (rchapler@microsoft.com)" &gt; "practicumdl (ADLS Gen2)" &gt; "Blob Containers" &gt; "practicumdlc". The main pane is titled "practicumdlc" and shows a list of files in a table format. The columns are: Name, Access Tier, Access Tier Last Modified, Last Modified, Blob Type, Content Type, Size, and Lease State. The table contains three rows: "sample1.csv" (Hot (inferred)), "sample2.csv" (Hot (inferred)), and "sample3.csv" (Hot (inferred)). All files are of type "Block Blob" and content type "application/vnd.ms-excel". The "Size" column shows values of 502 B, 55.6 KB, and 723 B respectively. Below the table, it says "Showing 1 to 3 of 3 cached items". At the bottom, there is an "Activities" section with a message: "Transfer from 'C:\Users\rchapler\Downloads\' to 'practicumdlc/' complete: 3 items transferred (used SAS, discovery completed) Started at 2/3/2021, 12:11:00 PM Duration: 2 seconds". There are also "Clear completed" and "Copy AzCopy Command to Clipboard" buttons.

## Data Share

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Explorer (2 instances)
- Resource Group

The screenshot shows the 'Create Data Share' wizard in Microsoft Azure (Preview). The 'Basics' tab is selected. The 'Project details' section asks to select a subscription and resource group. The 'Subscription' dropdown is set to 'rchapler'. The 'Resource group' dropdown is set to 'practicumrg', with a 'Create new' link below it. The 'Instance details' section asks for a location and name. The 'Location' dropdown is set to 'West US 2'. The 'Name' input field contains 'practicumds', which has a green checkmark next to it. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next : Tags >'.

On the “Create Data Share” page, enter values for the following items:

<b>Subscription</b>	Select your subscription
<b>Resource Group</b>	Select your Resource Group
<b>Location</b>	Select the value used during Resource Group creation
<b>Name</b>	Self-Explanatory

Review settings on remaining tabs.

No additional changes are required.

Click the “Review + create” button, validate, and then click the **Create** button.

After deployment is complete, click the “Go to resource” button.

The screenshot shows the Microsoft Azure Data Share configuration page for a resource named 'practicumds'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, and Diagnose and solve problems under the 'Overview' section; Properties, Locks, Sent Shares, and Received Shares under 'Data Share'; and Settings. The main content area is titled 'Essentials' and displays the following information: Resource group (practicumrg), Location (West US 2), Subscription (rchapler), and Subscription ID. It also includes a 'Tags (change)' section with a link to 'Click here to add tags'. Below this are two large buttons: 'Start sharing your data' (with an upward arrow icon) and 'View received shares' (with a downward arrow icon).

## Sample Share

*Note: To demonstrate sharing data to a target, you must instantiate a second ADX cluster.*

Click the “Start sharing your data” button.

On the resulting “Sent Shares” page, click the “+ Create” button.

The screenshot shows the Microsoft Azure (Preview) portal with the URL <https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions>. The user is logged in as rchapler@microsoft.com. The main navigation bar includes 'Dashboard', 'Microsoft DataShare-20210915092919', and 'practicumds'. On the left, a sidebar for 'practicumds | Sent Shares' lists options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Properties, Locks, and several monitoring and diagnostic settings. The 'Sent Shares' option is currently selected. The main content area displays a four-step wizard: '1. Details', '2. Datasets', '3. Recipients', and '4. Review + Create'. The '1. Details' step is active, showing fields for Share name (set to 'practicumdec'), Share type (set to 'In-place'), Description (placeholder 'Enter description for the share'), and Terms of use (placeholder 'Enter terms of use for the share'). At the bottom are 'Cancel' and 'Continue' buttons.

On the “Sent Shares” page, “1. Details” tab, enter values for the following items:

Share Name                      Self-Explanatory

Share Type                      Select “In-place” from the drop-down menu

Click the **Continue** button.

practicumds | Sent Shares

Dashboard > Microsoft.DataShare-20210915092919 > practicumds

1. Details   2. Datasets   3. Recipients   4. Review + Create

Search (Ctrl+ /)

Overview  
Activity log  
Access control (IAM)  
Tags  
Diagnose and solve problems

Settings  
Properties  
Locks

Data Share  
Sent Shares

Add datasets

Select datasets to be shared. You must have permission to add role assignment to the data store. This permission exists in the Owner role. See [Share Your Data tutorial](#) for details.

Previous Continue

On the “Sent Shares” page, “2. Datasets” tab, click the “Add datasets” button.

practicumds | Sent Shares

1. Details   2. Datasets   3. Recipients

Select dataset type

Azure Data Explorer

Cancel Next

On the resulting “Select dataset type” pop-out, click the “Azure Data Explorer” button, then click the **Next** button.

The screenshot shows the Microsoft Azure (Preview) portal interface. The left sidebar is dark-themed and includes sections for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Properties, Locks), Data Share (Sent Shares, Received Shares, Feedback), Monitoring (Alerts, Metrics, Diagnostic settings, Logs), and a search bar. The main content area shows a 'practicumds | Sent Shares' Data Share. A modal window titled 'Azure Data Explorer' is open, displaying a list of datasets. It includes a 'Refresh' button, dropdown menus for Subscriptions (selected 'rchapler'), Resource groups (selected 'practicumrg'), and Azure data explorer clusters (selected 'practicumdec2'). Navigation buttons 'Previous' and 'Next' are at the bottom of the modal.

On the resulting “**Azure Data Explorer**” pop-out, enter values for the following items:

<b>Subscriptions</b>	Select your subscription
<b>Resource Groups</b>	Select your Resource Group
<b>Azure Data Explorer Clusters</b>	Select your Data Explorer cluster
<b>Name</b>	Self-Explanatory

Click the **Next** button.

practicumds | Sent Shares

1. Details    2. Datasets    3. Recipients

Azure Data Explorer  
Select datasets

Refresh    Search by prefix

practicumdec2    1 dataset(s)

Select role  
Owner

Showing 1 to 1 of 1 items

Previous    Next

On the second "...Select datasets" pop-out, check the box next to your cluster and then click the **Next** button.

practicumds | Sent Shares

1. Details    2. Datasets    3. Recipients

Azure Data Explorer  
Rename datasets

Provide names for your datasets. This is the name that your data consumer will see when they accept the share. Dataset names must be unique.

Dataset name	Path
practicumdec2	practicumdec2/

Previous    Add datasets

On the "...Rename datasets" pop-out, confirm the automatically generated "Dataset name" value.  
Click the "Add datasets" button.

practicumds | Sent Shares

2. Datasets

Datasets ↑	Type	Path
practicumdec2	Azure Data Explorer Cluster	practicumdec2/

Previous Continue

Back on the “Sent Shares” page, “2. Datasets” tab, click the **Continue** button.

practicumds | Sent Shares

3. Recipients

Enter email address of the recipients for the share. Please ensure you are using recipient's Azure login email.

Add recipient Delete All Update expiration for all

Email	Share expiration
rchapler@microsoft.com	Fri Oct 15 2021 11:59 PM

Previous Continue

On the “Sent Shares” page, “3. Recipients” tab, click the “Add recipient” button.

In the resulting interface, enter an **Email** value.

Click the **Continue** button.

practicumds - Microsoft Azure

Microsoft Azure (Preview)

Dashboard > Microsoft.DataShare-20210915092919 > practicumds

practicumds | Sent Shares

Search (Ctrl+ /)

1. Details    2. Datasets    3. Recipients    4. Review + Create

Share Contents  
Number of datasets: 1

Settings  
Name of data share: practicumdec  
Description: -  
Terms of use: -

Recipients  
Number of recipients: 1

Previous    Create

On the “Sent Shares” page, “4. Review + Create” tab, click the **Create** button.

The designated email recipient can expect to receive an email like the one below:

Azure Data Share invitation from Rich Chapler - Message (HTML) Search

File **Message** Help

Delete Respond Share to Teams Quick Steps Move Tags Editing Immersive Translate Zoom Dynamics 365 Insights Report Message Protection Reply with Meeting Poll FindTime

Teams Quick Steps Language Add-in Add-in Add-in Protection FindTime

Previous Item Next Item Quick Print

## Azure Data Share invitation from Rich Chapler

 Microsoft Azure  
To: Rich Chapler

If there are problems with how this message is displayed, click here to view it in a web browser.

 Microsoft Azure

### You're invited to access data from Microsoft

You're receiving this email because Rich Chapler from Microsoft wants to share the following data with you.

Share name: practicumdec

Description: NA

[View invitation >](#)

See detailed instructions on how to accept and configure your data share.  
If you're new to Azure, [create a free Azure subscription](#).

[f](#) [t](#) [y](#) [in](#)  
[Privacy Statement](#)  
Microsoft Corporation, One Microsoft Way, Redmond, WA 98052  
 Microsoft

Click the “View invitation >” link.

A Data Share Invitations - Microsoft

https://ms.portal.azure.com/#blade/Microsoft\_Azure\_DataShare/InvitationsBrowseBlade

Microsoft Azure (Preview)

Search resources, services, and docs (G+/)

rchapler@microsoft.com MICROSOFT

Dashboard >

## Data Share Invitations

Refresh

Pending invitations sent to your Azure login email are listed. See [accept and receive data tutorial](#) for details.

Invitation	Sender	Company	Status	Received On
practicumdec	Rich Chapler	Microsoft	Pending	9/15/2021 11:02:56 AM

Click the **Invitation** link.

practicumdec - Microsoft Azure

https://ms.portal.azure.com/#blade/Microsoft\_Azure\_DataShare/InvitationsBrowseBlade

Microsoft Azure (Preview)

Search resources, services, and docs (G+/)

rchapler@microsoft.com MICROSOFT

Dashboard > Data Share Invitations >

### practicumdec

Invitation

From	Number of datasets
Rich Chapter	1
Company	Expires on
Microsoft	-
Description	-
Terms of use	-

**TARGET DATA SHARE ACCOUNT**

Subscription *	rchapler
Resource group *	practicumrg
Data share account *	practicumds
Received share name * ⓘ	practicumdec

Accept and configure

Reject

On the resulting “...Invitation” page, enter values for the following items:

Subscriptions	Select your subscription
Resource Groups	Select your Resource Group
Data Share Account	Select your Data Share account
Received Share Name	Confirm default value

Click the “Accept and configure” button.

The screenshot shows the Microsoft Azure portal interface. The left sidebar has a dark theme with white icons and text. The main content area has a light background. The URL in the browser bar is <https://ms.portal.azure.com/#@microsoft.onmicrosoft.com/resource/subscriptions>. The top navigation bar includes the Microsoft Azure logo, a search bar, and user profile information for rchapler@microsoft.com.

**Dashboard > Data Share Invitations > practicumds**

**practicumds | Received Shares**

**Details**   **Datasets**

Configure [Datasets](#) to receive data into your target data store.

Refresh

Source share	Provider
practicumdec	Rich Chapler

Provider company	Shared on
Microsoft	11:14:31 AM, 9/15/2021

Number of source datasets	Accepted by
1	Rich Chapler

Received share status
Active

Click the **Datasets** link.

On the resulting “Received Shares...” page, **Datasets** tab, check the box next to your Data Explorer cluster.

Click the “+ Map to target” button.

The screenshot shows the Microsoft Azure portal interface. In the top left, there's a navigation bar with 'practicums - Microsoft Azure' and a search bar. The main area shows 'practicums | Received Shares' under 'Data Share Invitations'. On the left, a sidebar has sections like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings', 'Properties', 'Locks', 'Data Share' (with 'Sent Shares' and 'Received Shares' selected), 'Feedback', 'Monitoring' (with 'Alerts' and 'Metrics'), and 'Diagnostic settings'. A central modal window titled 'Map datasets to target' is open. It has tabs for 'Details' and 'Datasets', with 'Datasets' selected. It shows a list of datasets: 'practicumdec' and 'practicumdec2'. Below this is a table with columns 'Datasets', 'Source Type', and 'Source Path'. The first row shows 'practicumdec' as an 'Azure Data Explorer Cluster' from 'practicumdec'. At the bottom of the modal are 'Cancel' and 'Map to target' buttons.

On the resulting “Map datasets to target” pop-out, enter values for the following items:

<b>Subscriptions</b>	Select your subscription
<b>Resource Groups</b>	Select your Resource Group
<b>Select Kusto clusters</b>	Select your Data Explorer cluster

Click the “Map to target” button.

The screenshot shows the Microsoft Azure (Preview) interface. The left sidebar has sections for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Properties, Locks), Data Share (Sent Shares), and Received Shares. The main area shows 'Received Shares > practicumdec'. The 'Datasets' tab is selected, displaying a table with one row:

Datasets	Source Type	Source Path	Status
practicumdec	Azure Data Explorer Cluster	practicumdec	Mapped

## Data Studio

Browse to [Download and install Azure Data Studio - Azure Data Studio | Microsoft Docs](https://docs.microsoft.com/en-us/sql/azure-data-studio/download-azure-data-studio?view=sql-server-ver15)

The screenshot shows the Microsoft Docs page for 'Download and install Azure Data Studio'. The left sidebar has sections for Version (SQL Server 2019), Filter by title, Azure Data Studio documentation, Download Azure Data Studio, Release notes, Overview, Quickstarts, Tutorials, Concepts, How-to guides, References, and Resources. The main content area has a heading 'Download and install Azure Data Studio' with a sub-section 'Download Azure Data Studio'. It includes a table for Windows:

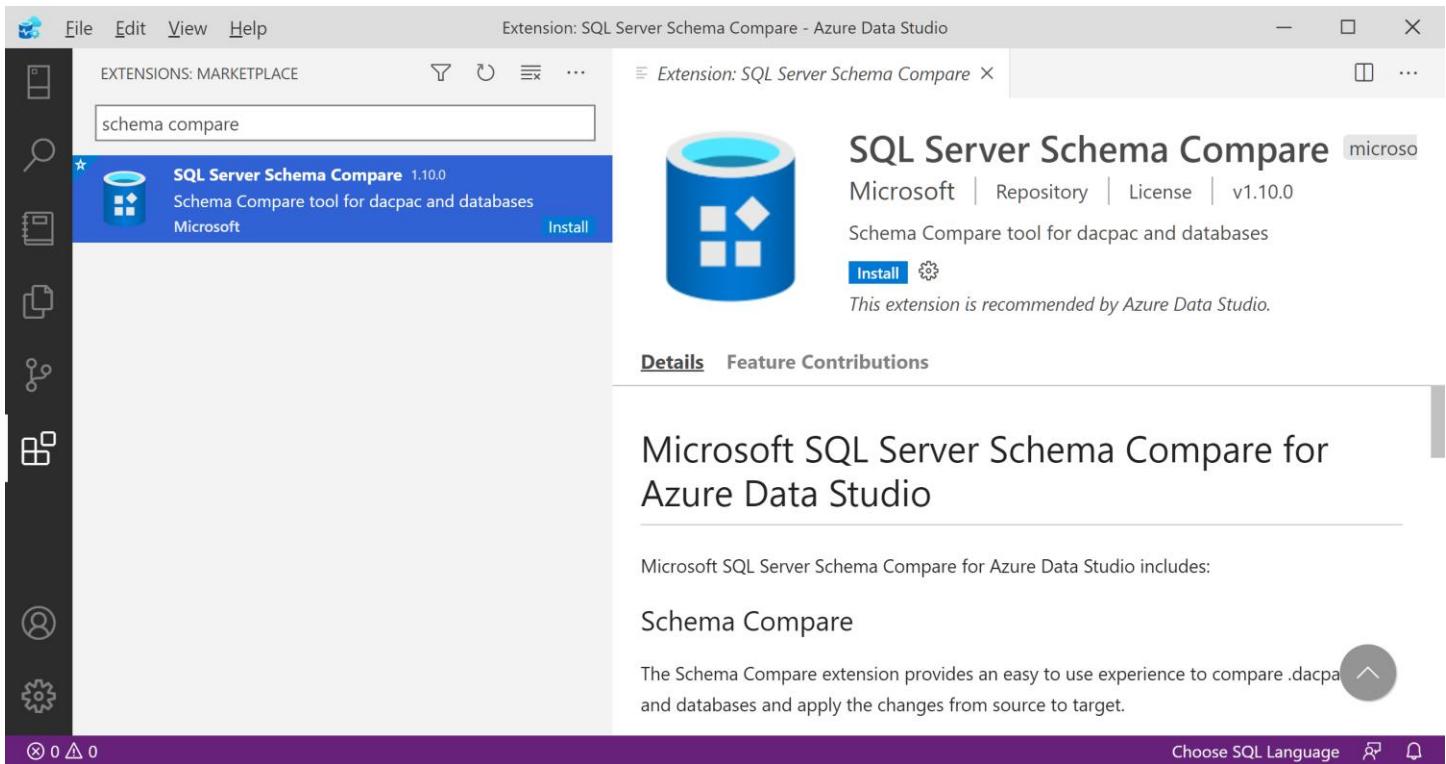
Platform	Download
Windows	User Installer (recommended) System Installer .zip

On the right, there are links for 'Is this page helpful?' (Yes, No), 'In this article' (Download Azure Data Studio, Install Azure Data Studio, What's new, Download Insiders build of Azure Data Studio, Supported operating systems, Recommended System Requirements, Check for updates, Move user settings, Uninstall Azure Data Studio from Windows, Uninstall Azure Data Studio from macOS, Uninstall Azure Data Studio from Linux, Next Steps, Get help for SQL tools, Contribute to SQL documentation), and a 'Download PDF' button.

Download, open, and install the “User Installer” for your platform. Once installed, launch Data Studio.

## Add Extensions

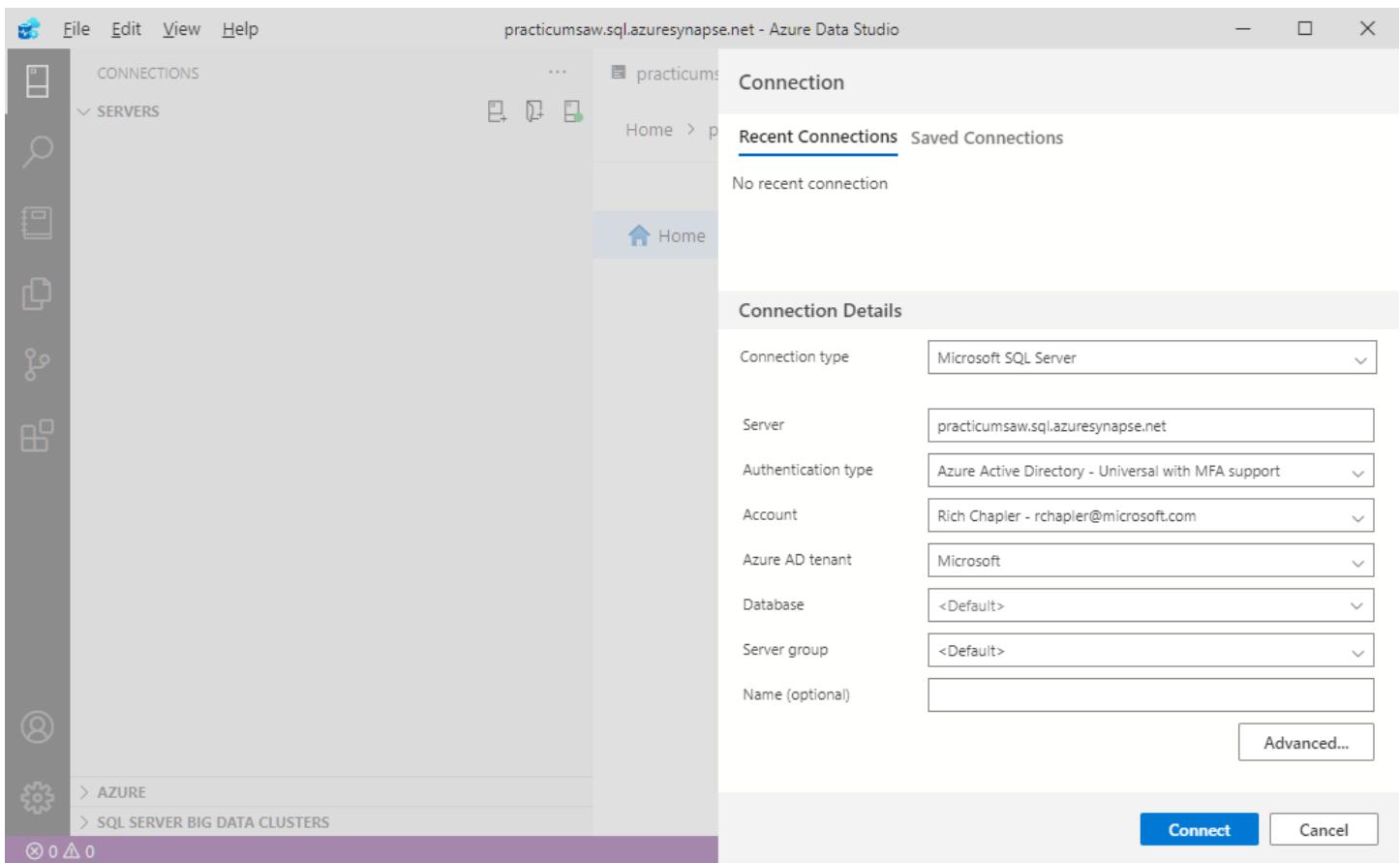
Navigate to **Extensions** in the left-hand navigation.



Search for “schema compare” and install the “**SQL Server Schema Compare**” extension.

## Create Connection

Navigate to **Connections** in the left-hand navigation.



Click the “Create Connection” icon (first of three) to the right of the SERVERS section header.

On the “Connection Details” pop-out, enter values for the following items:

Connection Type	Confirm default value, “Microsoft SQL Server”
Server	Paste the “Dedicated SQL endpoint” value copied in <a href="#">Instantiate Resources   Synapse</a>
Authentication Type	Select “Azure Active Directory – Universal with MFA support”
Account	Confirm credentials
Azure AD Tenant	Confirm tenant

Click the **Connect** button.

## Databricks

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Resource Group

The screenshot shows the 'Create an Azure Databricks workspace' page in the Microsoft Azure portal. The 'Basics' tab is active. In the 'Project Details' section, the subscription is set to 'rchapler' and the resource group is 'practicumrg'. In the 'Instance Details' section, the workspace name is 'practicumdb', the region is 'West US 2', and the pricing tier is 'Premium (+ Role-based access controls)'. At the bottom, there are navigation buttons for 'Review + create', '< Previous', and 'Next : Networking >'.

On the “Create an Azure Databricks workspace” page, enter values for the following items:

<b>Subscription</b>	Select your subscription
<b>Resource Group</b>	Select your Resource Group
<b>Workspace Name</b>	Self-Explanatory
<b>Region</b>	Select the value used during Resource Group creation
<b>Pricing Tier</b>	Select “ <b>Premium...</b> ” to enable features that we will need for this exercise

Review settings on remaining tabs {e.g., Networking, Tags}.

No additional changes are required.

Click the “**Review + create**” button, review configuration, and then click the **Create** button.

## New Cluster

Navigate to the **Overview** page in Databricks, click the “**Launch Workspace**” button, and login if required.

The screenshot shows the Azure Databricks landing page. On the left, there is a vertical sidebar with icons for Home, Workspace, Recents, Data, Clusters, Jobs, Models, and Search. The main content area features the Azure Databricks logo at the top. Below it are three main sections:

- Explore the Quickstart Tutorial**: An icon of a document with curly braces and a lightbulb. Description: Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.
- Import & Explore Data**: An icon of a dashed box with a cloud and a plus sign. Description: Quickly import data, preview its schema, create a table, and query it in a notebook.
- Create a Blank Notebook**: An icon of a document with curly braces and a plus sign. Description: Create a notebook to start querying, visualizing, and modeling your data.

Below these sections are three tabs: **Common Tasks**, **Recents**, and **Documentation**. The **Common Tasks** tab is active, showing a list of tasks with icons:

- New Notebook
- Create Table
- New Cluster
- New Job
- New MLflow Experiment
- Import Library
- Read Documentation

In the “Common Tasks” grouping, click “**New Cluster**”.

Create Cluster

New Cluster

Cancel Create Cluster

Cluster Name: practicumdbc

Cluster Mode: Standard

Pool: None

Databricks Runtime Version: Runtime: 7.4 (Scala 2.12, Spark 3.0.1)

Autopilot Options:

- Enable autoscaling
- Terminate after 120 minutes of inactivity

Worker Type: Standard\_DS3\_v2 (14.0 GB Memory, 4 Cores, 0.75 DBU)

Driver Type: Same as worker (14.0 GB Memory, 4 Cores, 0.75 DBU)

Min Workers: 2

Max Workers: 8

UI | JSON

On the “Create Cluster” page, enter values for the following items:

**Cluster Name** Self-Explanatory

No additional changes are required.

Click the “Create Cluster” button.

## New Notebook

Return to the start page.

In the “Common Tasks” grouping, click “New Notebook”.

The screenshot shows the 'Create Notebook' dialog box over a blurred background of the Azure Databricks interface. The dialog box has fields for 'Name' (set to 'practicumdbn'), 'Default Language' (set to 'Python'), and 'Cluster' (set to 'practicumdbc'). It includes 'Cancel' and 'Create' buttons. The background shows a 'Quickstart' section with a 'Create a Blank Notebook' button and documentation links for 'Documentation', 'Release Notes', and 'Getting Started'.

Enter a meaningful name and click the **Create** button.

The screenshot shows the Azure Databricks workspace with a new notebook titled 'practicumdbn (Python)'. The notebook is currently detached. The left sidebar shows the 'Workspace' tab is selected. The notebook area displays a single command cell labeled 'Cmd 1' containing the number '1'. A tooltip indicates 'Shift+Enter to run' and lists 'shortcuts'.

Make note of the URL {i.e., <http://adb-21458...>} for use in the next section.

## Secret Scope

Navigate to your instance of Key Vault.

Click **Properties** in the **Settings** group of the left-hand navigation.

practicumkv | Properties

Name: practicumkv

Sku (Pricing tier): Standard

Location: westus2

Vault URI: <https://practicumkv.vault.azure.net/>

Resource ID: /subscriptions/.../resourceGroups/practicumrg...

Subscription ID: [REDACTED]

Subscription Name: rchapler

Directory ID: [REDACTED]

Directory Name: Microsoft

**Soft-delete**

Soft delete has been enabled on this key vault

Days to retain deleted vaults: 90

Purge protection:

Disable purge protection (allow key vault and objects to be purged during retention period)

Enable purge protection (enforce a mandatory retention period for deleted vaults and vault objects)

Make note of the values in “Vault URI” and “Resource ID”.

On a new tab, navigate to [https://\[databricksInstance\]#secrets/createScope](https://[databricksInstance]#secrets/createScope).

You will replace {databricksInstance} with the start of the URL in your workspace, from my exercise, example:

<https://adb-2154823451042175.15.azuredatabricks.net/#secrets/createScope>

The screenshot shows the Microsoft Azure Databricks portal with the URL <https://adb-2154823451042175.15.azuredatabricks.net/?o=2154823451042175#secrets/createScope>. The left sidebar includes icons for Home, Workspace, Recents, Data, Clusters, Jobs, and Models. The main content area is titled "Create Secret Scope" with a "Cancel" and "Create" button. It contains fields for "Scope Name" (practicumdbss), "Manage Principal" (Creator), and "Azure Key Vault" (DNS Name: https://practicumkv.vault.azure.net/ and Resource ID: /46ed/resourceGroups/practicumrg/providers/Microsoft.KeyVault/vaults/practicumkv).

On the “Create Secret Scope” page, enter values for the following items:

Scope Name	Self-Explanatory
DNS Name	Paste the copied “Vault UI” value
Resource ID	Paste the copied “Resource ID” value

Click the **Create** button.

## DevOps

Browse to <https://azure.microsoft.com/en-us/services/devops/>

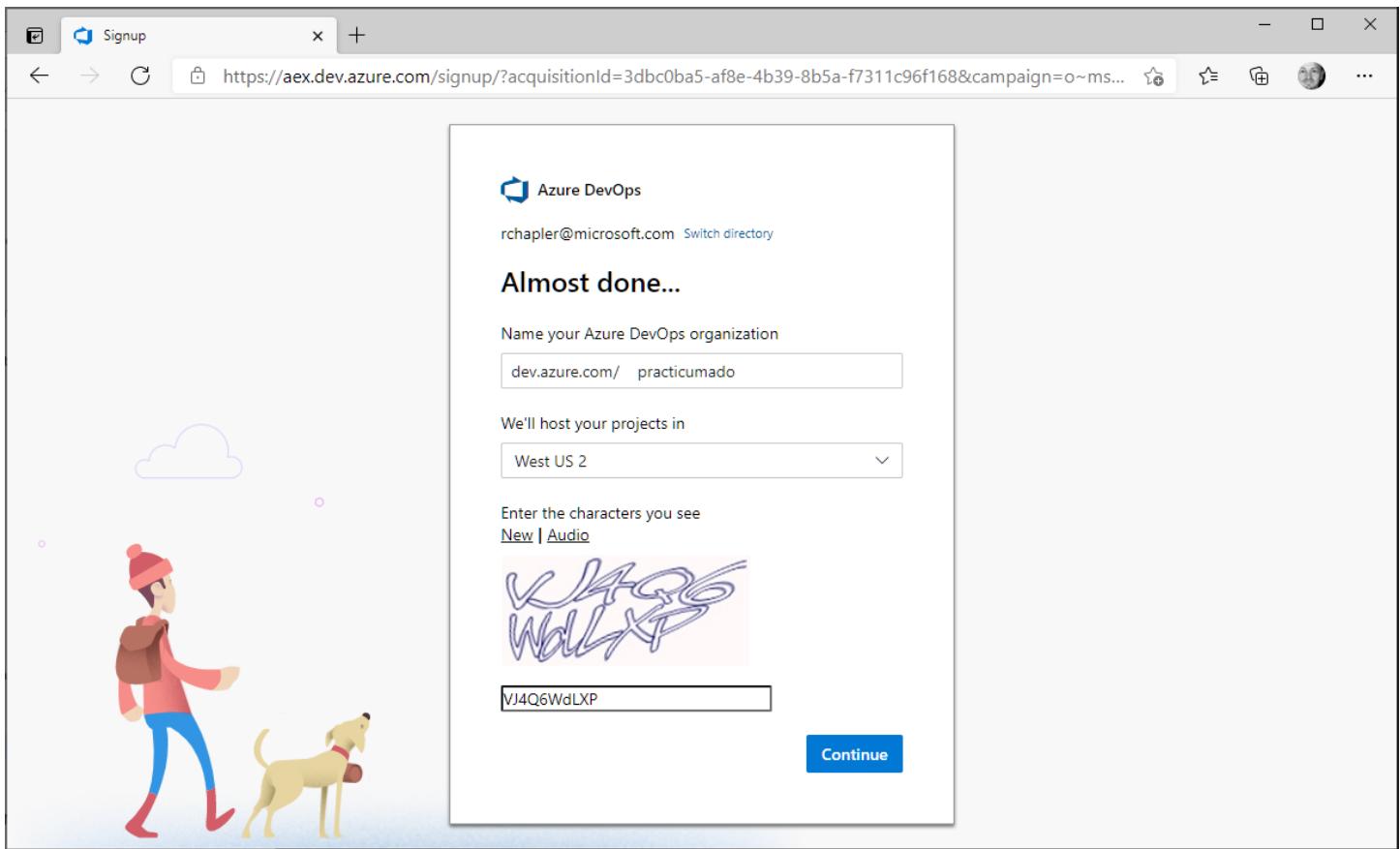
The screenshot shows the Microsoft Azure DevOps Services landing page. At the top, there's a navigation bar with links for Overview, Solutions, Products (which is currently selected), Documentation, Pricing, Training, Marketplace, Partners, Support, Blog, and More. Below the navigation is a breadcrumb trail: Home / Services / Azure DevOps. The main heading is "Azure DevOps" with the subtext "Plan smarter, collaborate better, and ship faster with a set of modern dev services." Two buttons are visible: "Start free" and "Start free with GitHub". To the right is a colorful illustration of people working on a rocket launching from a stack of servers. Below the illustration, there's a link for users who already have an account and a "Sign in to Azure DevOps" button.

Click the “Start free with GitHub” button.

On the resulting Azure DevOps page, click the “New organization” link on the left-hand navigation.

The screenshot shows the Azure DevOps Signup page. On the left, there's a cartoon illustration of a person walking a dog. The main content area has a "Get started with Azure DevOps" heading and a "Continue" button. It includes a note about agreeing to Terms of Service, Privacy Statement, and Code of Conduct, and a checkbox for staying informed about Azure DevOps and other Microsoft products.

Click the **Continue** button.



On the “Almost done...” page, enter values for the following form items:

Name your Azure DevOps organization      Self-Explanatory

We will host your projects in      Select the value used during Resource Group creation

Click the **Continue** button.

## Create Project

The screenshot shows the Azure DevOps 'Create a project to get started' page. On the left, there's a sidebar with organization names (practicumado, [REDACTED], [REDACTED], [REDACTED]), a '46 more organizations' link, and a 'New organization' button. A 'What's new' section mentions 'Sprint 184 release notes' about Azure Pipelines grants. Below that is an 'Organization settings' link. The main area has a title 'Create a project to get started'. It includes a 'Project name \*' field with 'practicumadop' entered, a 'Description' field (empty), and a 'Visibility' section. The 'Private' option is selected, highlighted with a blue border. At the bottom is a blue '+ Create project' button.

On the “Create a project to get started” page, enter values for the following form items:

Project Name	Self-Explanatory
Visibility	Select the value that best aligns with your requirements

Click the “+ Create project” button.

When processing is complete, you will be navigated to a screen like the one snipped below.

The screenshot shows the Azure DevOps interface for the 'practicumadop' project. The left sidebar contains navigation links: Overview, Summary (selected), Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. Below the sidebar is a 'Project settings' link. The main content area features a green header bar with the project name 'practicumadop'. A central illustration depicts a person working at a desk with a laptop, accompanied by a dog. The text 'Welcome to the project!' is displayed above a row of buttons labeled 'Boards', 'Repos', 'Pipelines', 'Test Plans', and 'Artifacts'. Below these buttons is the text 'or manage your services'. At the top right of the main area are 'Private' and 'Invite' buttons, along with a search bar and other user interface elements.

## Key Vault

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Resource Group

The screenshot shows the 'Create key vault' wizard in the Microsoft Azure portal. The current step is 'Basics'. The page includes a brief description of Azure Key Vault, project details (subscription and resource group selection), instance details (key vault name, region, and pricing tier), and navigation buttons for 'Review + create' and 'Next : Access policy >'.

Azure Key Vault is a cloud service used to manage keys, secrets, and certificates. Key Vault eliminates the need for developers to store security information in their code. It allows you to centralize the storage of your application secrets which greatly reduces the chances that secrets may be leaked. Key Vault also allows you to securely store secrets and keys backed by Hardware Security Modules or HSMs. The HSMs used are Federal Information Processing Standards (FIPS) 140-2 Level 2 validated. In addition, key vault provides logs of all access and usage attempts of your secrets so you have a complete audit trail for compliance.

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* rchabler

Resource group \* practicumrg

Create new

**Instance details**

Key vault name \* practicumkv

Region \* West US 2

Pricing tier \* Standard

**Buttons:**

- Review + create
- < Previous
- Next : Access policy >

On the “Create Key Vault” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your Resource Group
Key Vault Name	Self-Explanatory
Region	Select the value used during Resource Group creation
Pricing Tier	Confirm default selection, “Standard”

Review settings on remaining tabs {e.g., Access Policy, Networking, Tags}.

No additional changes are required.

Click the “Review + create” button, review configuration, and then click the **Create** button.

## Log Analytics

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Resource Group

A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. [Learn more](#)

With Azure Monitor Logs you can easily store, retain, and query data collected from your monitored resources in Azure and other environments for valuable insights. A Log Analytics workspace is the logical storage unit where your log data is collected and stored.

**Project details**  
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* rchapler

Resource group \* practicumrg  
[Create new](#)

**Instance details**  
Name \* praticumlaw

Region \* West US 2

[Review + Create](#) [<< Previous](#) [Next : Pricing tier >](#)

On the “Create Log Analytics workspace” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your Resource Group
Name	Self-Explanatory
Region	Select the value used during Resource Group creation

Review settings on remaining tabs {e.g., Pricing Tier, Tags}.

No additional changes are required.

Click the **Review + create** button, review configuration, and then click the **Create** button.

# Metrics Advisor

The screenshot shows the Microsoft Azure portal in dark mode, specifically the 'Create Metrics Advisor' wizard. At the top, the URL is https://ms.portal.azure.com/#create/Microsoft.CognitiveServicesMetricsAdvisor. The page title is 'Create Metrics Advisor'. The breadcrumb navigation shows: Dashboard > Resource groups > practicumrg > Create a resource > Marketplace > Metrics Advisor > Create Metrics Advisor.

**Basics**   Virtual network   Tags   Review + create

Embed AI-powered monitoring features to stay one step ahead of incidents no machine-learning expertise required. Metrics Advisor monitors the performance of your organization's growth engines, from sales revenue to manufacturing operations. It helps you quickly identify and fix problems through a powerful combination of monitoring in near-real time, adapting models to your scenario, offering granular analysis with diagnostics, and alerting. [Learn more](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*    Resource group \*  [Create new](#)

**Instance details**

Region \*    Name \*    Pricing tier \*

[View full pricing details](#)

**Storage**

Bring your own storage \*  Yes  No

The deployment could take up to 60 minutes to complete, although it normally finishes in less than 10 minutes.

I confirm I have read and understood the  notice below.

[Service Agreement & Terms](#)

[Review + create](#)   < Previous   [Next : Virtual network >](#)

On the “Create Metrics Advisor” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your Resource Group
Region	Select the value used during Resource Group creation
Name	Self-Explanatory
Pricing Tier	Select S0
Bring Your Own Storage	Confirm default selection, “No”
I confirm...	Check the box

Review settings on remaining tabs {e.g., Virtual Network, Tags}.

No additional changes are required.

Click the “**Review + create**” button, review configuration, and then click the **Create** button.

## Add Permissions to Data Explorer

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Explorer

Navigate to your instance of Data Explorer.

The screenshot shows the Azure portal interface for managing permissions. On the left, the navigation menu is visible with 'Permissions' selected. The main area displays a table of permissions with one item listed: 'practicumdf' (Type: App). To the right, a modal window titled 'New Principals' is open, showing a search result for 'practicumma' with one item selected: 'practicumma' (GUID: 1b12f297-9025-4ac9-bfc3-98832a38dc38). A 'Selected' button is shown below the search results.

Click **Permissions** in the left-hand navigation.

Click “+ Add” on the resulting page.

Select **AllDatabaseViewer** from the resulting drop-down.

Search for and then click to **Select** your instance of Metrics Advisor.

## Postman

Browse to [Download Postman | Try Postman for Free](#)

Select, download, and install the version of the Postman app appropriate for your system.

## Power Apps

If you don't have a working instance of Power Apps, you can get started at <https://Power Apps.microsoft.com/en-us/>

If you are already set up, click “**Sign In**”.

## Purview

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Key Vault
- Resource Group

**Create Purview account**

Provide Purview account info

**\*Basics**   **\*Networking**   Tags   Review + Create

Create a Purview account to develop a data governance solution in just a few clicks. A storage account and eventhub will be created in a managed resource group in your subscription for catalog ingestion scenarios. [Learn more](#)

**Project details**

Subscription \*

Resource group \*  [Create new](#)

**Instance details**

Purview account name \*  ✓

Location \*

**Managed resources**

A resource group, a storage account, and an Eventhub will be created in the selected subscription for catalog ingestion scenarios. The Microsoft.Storage and Microsoft.EventHub resource providers will get registered. [Learn more](#)

Managed resource group name \*  ✓

Storage account name *Name will be auto-generated during account creation.*

Event Hubs namespace name *Name will be auto-generated during account creation.*

[Review + Create](#)   [Previous](#)   [Next: Networking >](#)

On the “Create Purview account” page, enter values for the following items:

<b>Subscription</b>	Select your subscription
<b>Resource Group</b>	Select your Resource Group
<b>Purview Account Name</b>	Self-Explanatory
<b>Location</b>	Select the value used during Resource Group creation
<b>Managed Resource Group...</b>	Self-Explanatory

Review settings on remaining tabs {e.g., Networking, Tags}.

No additional changes are required.

Click the “Review + Create” button, review configuration, and then click the **Create** button.

## Add Access Policy to Key Vault

Navigate to the Key Vault.

The screenshot shows the Microsoft Azure Key Vault settings page for the 'practicumkv' resource. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Keys, Secrets, Certificates, and Access policies. The 'Access control (IAM)' link is underlined, indicating it is selected. The main content area displays the 'Essentials' section with details like Resource group (practicumrg), Location (West US 2), Subscription (rchapler), and Directory ID (redacted). It also shows Vault URI (https://practicumkv.vault.azure.net/), Sku (Standard), and various access controls like Soft-delete (Enabled) and Purge protection (Disabled).

Click “Access Policies” in the **Settings** group of the left-hand navigation.

The screenshot shows the Microsoft Azure Key Vault Access policies page for the 'practicumkv' resource. The left sidebar includes links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Keys, Secrets, Certificates, and Access policies. The 'Access policies' link is underlined, indicating it is selected. The main content area shows the 'Enable Access to:' section with checkboxes for Azure Virtual Machines for deployment, Azure Resource Manager for template deployment, and Azure Disk Encryption for volume encryption. Below this, the 'Permission model' is set to 'Vault access policy' (radio button selected). A link '+ Add Access Policy' is visible. A table titled 'Current Access Policies' lists one entry: 'APPLICATION'.

Click the “+ Add Access Policy” link.

Principal

Select a principal

practicump  
d0206ad4-fa5d-4923-982a-4cf48bcf3559  
Selected

Selected items

practicump  
d0206ad4-fa5d-4923-982a-4cf48bcf3559

Add

Configure from template (optional)

Key permissions

Secret permissions

Certificate permissions

Select principal \*

Authorized application ⓘ

Secret Management Operations

Get

List

Set

Delete

Recover

Backup

Restore

Select

On the “Add access policy” page, enter values for the following items:

**Secret Permissions** Select **Get** and **List** in the drop-down

**Select Principal** Search for, and then **Select** the managed identity for your instance of Purview

Click the **Add** button.

practicumkv | Access policies

Key vault

Search (Ctrl+ /)

Save Discard Refresh

Please click the 'Save' button to commit your changes.

Enable Access to:

Azure Virtual Machines for deployment ⓘ

Azure Resource Manager for template deployment ⓘ

Azure Disk Encryption for volume encryption ⓘ

Permission model

Vault access policy (selected)

Azure role-based access control

+ Add Access Policy

Current Access Policies

Name	Email	Key Permissions	Secret Permissions	Certificate Permissions	Action
APPLICATION	practicump	0 selected	2 selected	0 selected	Delete

Click the **Save** button.

## Connect Key Vault

Navigate to Purview and click on the “**Open Purview Studio**” button.

Select the **Management** icon on the left-hand navigation.

Click **Credentials** in the “**Security and access**” group of the resulting menu.

Click the “**Manage Key Vault connections**” button on the resulting page.

Click the “**+ New**” button the resulting pop-out.

The screenshot shows the 'New Key Vault' creation dialog in the Azure Purview Studio. The dialog has the following fields:

- Name \***: practicumkv
- Description**: Enter description
- Azure subscription**: rchapler [dropdown]
- Key Vault name \***: practicumkv
- Note**: You must grant the Purview managed identity access to your Azure Key Vault. [See more](#)

At the bottom are 'Create', 'Back', and 'Cancel' buttons.

On the “**New Key Vault**” pop-out, enter values for the following items:

<b>Name</b>	Self-Explanatory
<b>Subscription</b>	Select your subscription
<b>Key Vault Name</b>	Select your instance of Key Vault

Click the **Create** button.

On the resulting “**Confirm granting access**” pop-up, review and then click the **Confirm** button.

Confirm the addition of your instance of Key Vault on the “**Manage Key Vault connections**” pop-out.

Click the **Close** button.

## Add New Credential

On the **Credentials** page, click the “**+ New**” button.

The screenshot shows the Azure Purview Studio interface. In the top navigation bar, there are tabs for 'practicump - Microsoft Azure' and 'Azure Purview Studio'. The URL in the address bar is [https://ms.web.purview.azure.com/resource/practicump/main/datasource/management/credentials?feature.tenant=\[REDACTED\]](https://ms.web.purview.azure.com/resource/practicump/main/datasource/management/credentials?feature.tenant=[REDACTED]). The main content area has a blue header bar with icons for notifications, search, and user profile. On the left, a sidebar menu includes 'General', 'Overview', 'Metrics', 'Lineage connections', 'Data Factory', 'Data Share', 'Security and access', and 'Credentials' (which is selected). The central panel is titled 'Credentials' and shows a sub-section for 'New credential'. The 'Name' field is populated with 'practicumsds'. The 'Description' field is empty. Under 'Authentication method', 'SQL authentication' is selected. The 'User name' field contains 'rchapler'. The 'Password' field is empty. Under 'Key Vault connection', 'practicumkv' is selected. The 'Secret name' field contains 'practicumsds-adminpassword'. The 'Secret version' field contains the placeholder 'Use the latest version if left blank'. At the bottom right of the dialog are 'Create' and 'Cancel' buttons.

On the resulting “**New Credential**” pop-out, enter values for the following items:

<b>Name</b>	Self-Explanatory
<b>Authentication</b>	Select “SQL authentication”
<b>User Name</b>	Enter the “server admin login” value used during creation of the SQL Database Server
<b>Key Vault Connection</b>	Select the Key Vault created in <a href="#">Appendix   Key Vault</a>
<b>Secret Name</b>	Enter the value used in <a href="#">Appendix   SQL   Add Key Vault Secret</a>

No additional changes are required.

Click the **Create** button.

## Resource Group

The screenshot shows the Microsoft Azure (Preview) portal with the URL <https://ms.portal.azure.com/#create/Microsoft.ResourceGroup>. The page is titled "Create a resource group". The "Basics" tab is selected. The "Project details" section contains fields for "Subscription" (set to "rchapler") and "Resource group" (set to "practicumrg"). The "Resource details" section contains a field for "Region" (set to "(US) West US 2"). At the bottom, there are buttons for "Review + create", "< Previous", and "Next : Tags >".

On the “Create a Resource Group” page, enter values for the following items:

<b>Subscription</b>	Select your subscription
<b>Resource Group</b>	Self-Explanatory
<b>Region</b>	Select a region appropriate for your situation; take into consideration that: <ul style="list-style-type: none"><li>• Some regions {e.g., West US and East US} see higher demand than others</li><li>• Creation of resources in the same region offers best performance and lowest cost</li></ul>

Review settings on remaining tabs {e.g., Tags}.

No additional changes are required.

Click the “Review + create” button, review configuration, and then click the **Create** button.

## SQL

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Resource Group

**Select SQL deployment option**

**How do you plan to use the service?**

**SQL databases**  
Best for modern cloud applications. Hyperscale and serverless options are available.  
Resource type: Single database  
**Create** **Show details**

**SQL managed instances**  
Best for most migrations to the cloud. Lift-and-shift ready.  
Resource type: Single instance  
**Create** **Show details**

**SQL virtual machines**  
Best for migrations and applications requiring OS-level access. Lift-and-shift ready.  
Image  
**Create** **Show details**

On the “Select SQL deployment option” page, confirm default selection “Single database” in the “SQL databases” > “Resource type” drop-down.

Click the **Create** button.

**New server**

Microsoft

Server name \* **practicumsds** .database.windows.net

Server admin login \* **rchabler**

Password \* **\*\*\*\*\***

Confirm password \* **\*\*\*\*\***

Location \* **(US) West US 2**

**Create SQL Database**

**Basics** • Networking Security Additional settings Tags Review + create

Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* **rchabler**

Resource group \* **practicumrg** [Create new](#)

**Database details**

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name \* **practicumsd**

Server \* **Select a server** [Create new](#)

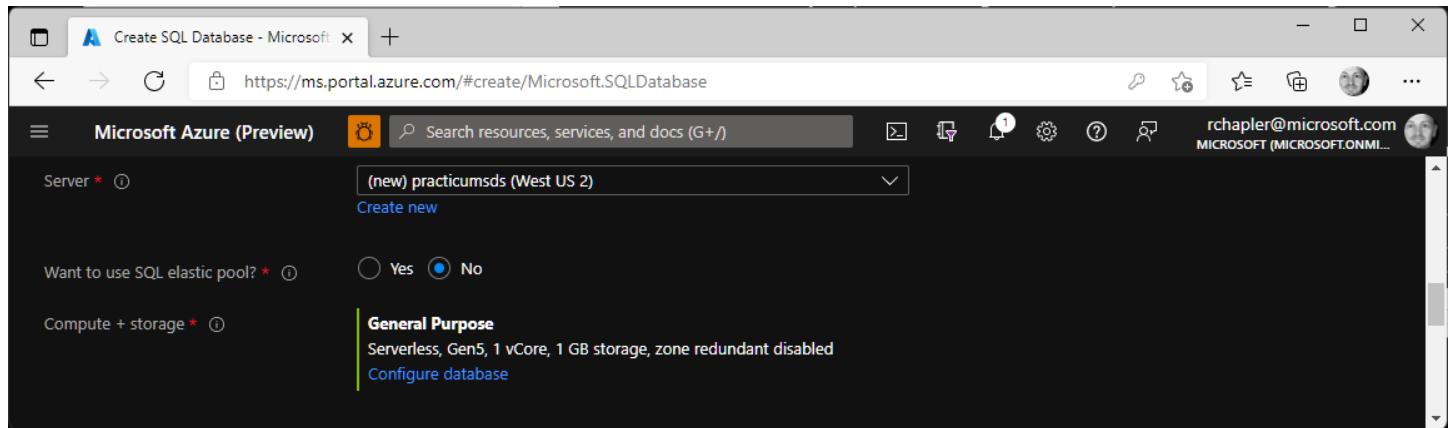
**Review + create** **Next : Networking >** **OK**

On the “Create SQL Database” page, **Basics** tab, enter values for the following items:

---

Subscription	Select your subscription
Resource Group	Select your Resource Group
Database Name	Self-Explanatory
Server	Click the “Create new” link, enter values in the resulting “New server” pop-out and then click the <b>OK</b> button

---



Further down on the “Create SQL Database” page, **Basics** tab, enter values for the following items:

---

Want to use SQL elastic pool?	Confirm default value, <b>No</b>
Compute + Storage	Click the “Configure database” link

---

The screenshot shows the Microsoft Azure (Preview) portal with the URL <https://ms.portal.azure.com/#create/Microsoft.SQLDatabase>. The page is titled "Configure" and is part of the "Create SQL Database" process. It includes sections for "Service and compute tier", "Compute Hardware", and configuration sliders for "Max vCores" and "Min vCores". The "Compute tier" section shows "Serverless" selected. The "Hardware Configuration" section shows "Gen5" selected. The "Compute Hardware" section shows "Max vCores" set to 1 vCore and "Min vCores" set to 0.5 vCore. An "Apply" button is visible at the bottom.

On the resulting **Configure** page, enter values for the following items:

Service Tier	Confirm default value, “General Purpose...”
Compute Tier	Click the <b>Serverless</b> radio button
Max vCores	Confirm minimum value
Min vCores	Confirm minimum value

The screenshot shows the Microsoft Azure (Preview) portal with the URL <https://ms.portal.azure.com/#create/Microsoft.SQLDatabase>. The user is on the 'Create SQL Database' configuration page. Key settings shown include:

- Auto-pause delay:** Enabled, set to 0 Days, 1 Hour, 0 Minutes.
- Data max size (GB):** Set to 1 GB, resulting in 307.2 MB LOG SPACE ALLOCATED.
- Cost summary:** Details the cost for a Gen5 - General Purpose (GP\_S\_Gen5\_1) database:
  - Cost per GB (in USD): 0.12
  - Max storage selected (in GB): x 1.3
  - ESTIMATED STORAGE COST / MONTH: 0.15 USD
  - COMPUTE COST / VCORE / SECOND<sup>1</sup>: 0.000145 USD

At the bottom, there is an 'Apply' button.

Further down on the resulting **Configure** page, enter values for the following items:

<b>Enable Auto-Pause</b>	Confirm default value, checked
<b>Days   Hours   Minutes</b>	Confirm default values, 0   1   0
<b>Data Max Size (GB)</b>	Set minimum appropriate value
<b>“...database zone redundant”</b>	Confirm default value, No

Review the “Cost Summary” and then click the **Apply** button.

**Backup storage redundancy**

Choose how your PITR and LTR backups are replicated. Geo restore or ability to recover from regional outage is only available when geo-redundant storage is selected.

Backup storage redundancy (1)

Locally-redundant backup storage - Preview

Zone-redundant backup storage - Preview

Geo-redundant backup storage

i Your use of either of the Preview backup storage redundancy options (ZRS and LRS) is governed by the agreement under which you obtained Microsoft Azure Services. By selecting a Preview redundancy option, you confirm that you agree to the preview terms in such agreement.  
Microsoft Azure Legal Information: [Learn more](#)

Review + create Next : Networking >

Back on and further down on the “Create SQL Database” page, **Basics** tab, enter values for the following items:

**Backup Storage Redundancy** Confirm default value, “Locally-redundant backup storage”

## Key Vault Secret, Admin Password

Navigate to your instance of Key Vault.

Click **Secrets** in the **Settings** group of the left-hand navigation.

Click the “+ Generate/Import” button.

**Create a secret**

Upload options Manual

Name (1) practicumsds-adminpassword

Value (1) .....

Content type (optional)

Set activation date

Set expiration date

Enabled Yes No

Tags 0 tags

Create

On the “Create a secret” page, enter values for the following items:

Name	Self-Explanatory
Value	Enter the value used when creating the SQL Database Server

No additional changes are required.

Click the **Create** button.

## Sample Database

Navigate to the “Additional settings” tab.

The screenshot shows the Microsoft Azure portal interface for creating a new SQL database. The URL in the browser is <https://ms.portal.azure.com/#create/Microsoft.SQLDatabase>. The top navigation bar includes links for Dashboard, practicumrg, Create a resource, Marketplace, Azure SQL, Select SQL deployment option, and a user profile for rchaler@microsoft.com. The main title is "Create SQL Database". Below it, there are tabs for Basics, Networking, Security, Additional settings (which is underlined to indicate it's selected), Tags, and Review + create. The "Additional settings" section contains the following configuration:

- Data source:** Set to "Sample". Other options are "None" and "Backup". A note states: "AdventureWorksLT will be created as the sample database."
- Database collation:** Set to "SQL\_Latin1\_General\_CI\_AS". A note states: "Database collation defines the rules that sort and compare data, and cannot be changed after database creation. The default database collation is SQL\_Latin1\_General\_CI\_AS." A "Learn more" link is provided.

At the bottom of the screen, there are navigation buttons: "Review + create" (highlighted in blue), "< Previous", and "Next : Tags >".

On the “Create SQL Database” page, “Additional Settings” tab, enter values for the following items:

Use Existing Data	Select the “Sample” option
-------------------	----------------------------

No additional changes are required.

Click the “Review + create” button, review configuration, and then click the **Create** button.

## Configure Firewall

Navigate to your Azure SQL Server and click the “Show firewall settings” link on the **Overview** page.

The screenshot shows the Azure portal interface for managing a SQL server. On the left, there's a sidebar with various navigation options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Quick start, and Settings. Under Settings, there are links for Azure Active Directory, SQL databases, and SQL elastic pools. The main content area is titled 'practicumsds | Firewalls and virtual networks'. It contains several configuration sections: 'Deny public network access' (set to Yes), 'Click here to create a new private endpoint. Create Private Endpoint', 'Minimum TLS Version' (set to 1.2), 'Connection Policy' (set to Default), 'Allow Azure services and resources to access this server' (set to Yes), and a 'Client IP address' input field. At the top right, there are Save, Discard, and Add client IP buttons.

Set “Allow Azure services and resources to access this server” to “Yes”.

Click the **Save** button.

## Storage Account

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Resource Group

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below.

[Learn more about Azure storage accounts](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* rchapler

Resource group \* praticumrg

Create new

**Instance details**

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. [Choose classic deployment model](#)

Storage account name \* praticumsa

Location \* (US) West US 2

Performance Standard

Account kind StorageV2 (general purpose v2)

Replication Read-access geo-redundant storage (RA-GRS)

Review + create < Previous Next : Networking >

On the “Create Storage Account” page, enter values for the following items:

Subscription	Select your subscription
Resource Group	Select your Resource Group
Storage Account Name	Self-Explanatory
Location	Select the value used during Resource Group creation
Performance	Confirm default radio button selection, “Standard”
Account Kind	Confirm default selection, “StorageV2 (general purpose v2)”
Replication	Confirm default selection, “Read-access geo-redundant storage (RA-GRS)”

Review settings on remaining tabs {e.g., Networking, Data Protection, Tags}.

No additional changes are required.

Click the “Review + create” button, review configuration, and then click the **Create** button.

## Synapse

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Lake

- Key Vault
- Resource Group
- SQL

**Create Synapse workspace**

**\* Basics**   **Security**   **Networking**   **Tags**   **Review + create**

Create a Synapse workspace to develop an enterprise analytics solution in just a few clicks.

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all of your resources.

Subscription \*    Resource group \*    Managed resource group

**Workspace details**

Name your workspace, select a location, and choose a primary Data Lake Storage Gen2 file system to serve as the default location for logs and job output.

Workspace name \*    Region \*    Select Data Lake Storage Gen2 \*  From subscription    Manually via URL

Account name \*    File system name \*

**Info** We will automatically grant the workspace identity data access to the specified Data Lake Storage Gen2 account, using the [Storage Blob Data Contributor role](#). To enable other users to use this storage account after you create your workspace, perform these tasks:
 

- Assign other users to the **Contributor** role on workspace
- Assign other users the appropriate [Synapse RBAC roles](#) using Synapse Studio
- Assign yourself and other users to the **Storage Blob Data Contributor** role on the storage account

[Learn more](#)

**Review + create**   [< Previous](#)   [Next: Security >](#)

On the “Create Synapse workspace” page, enter values for the following items:

<b>Subscription</b>	Select your subscription
<b>Resource Group</b>	Select your Resource Group
<b>Managed Resource Group</b>	Self-Explanatory This additional Resource Group will hold ancillary resources created specifically for Synapse
<b>Workspace Name</b>	Self-Explanatory
<b>Region</b>	Select the value used during Resource Group creation

Select Data Lake Storage Gen2	Confirm default, “From subscription”
Account Name	Select the Storage Account created previously
File System Name	Select the Storage Account container created previously
Assign myself...	Check to assign necessary permissions

Click the “Next: Security >” button.

\* Basics \* **Security** Networking Tags Review + create

Configure security options for your workspace.

**SQL administrator credentials**

Provide credentials that can be used for administrator access to the workspace's SQL pools. If you don't provide a password, one will be automatically generated. You can change the password later.

SQL Server admin login \*  rchapler

SQL Password

Confirm password

**System assigned managed identity permission**

Choose the permissions that you would like to assign to the workspace's system-assigned identity. [Learn more](#)

Allow pipelines (running as workspace's system assigned identity) to access SQL pools. (i)

Allow network access to Data Lake Storage Gen2 account. (i)

i The selected Data Lake Storage Gen2 account does not restrict network access using any network access rules, or you selected a storage account manually via URL under Basics tab. [Learn more](#)

**Workspace encryption**

⚠ Double encryption configuration cannot be changed after opting into using a customer-managed key at the time of workspace creation.

Choose to encrypt all data at rest in the workspace with a key managed by you (customer-managed key). This will provide double encryption with encryption at the infrastructure layer that uses platform-managed keys. [Learn more](#)

Double encryption using a customer-managed key  Enable  Disable

[Review + create](#) [< Previous](#) [Next: Networking >](#)

Enter values for “SQL Server Admin Login” and “SQL Password”.

No other setting modifications are required.

Click the “Review + create” button, review configuration, and then click the **Create** button.

## Key Vault Secret, Admin Password

Navigate to your instance of Key Vault.

Click **Secrets** in the **Settings** group of the left-hand navigation.

Click the “+ Generate/Import” button.

The screenshot shows the 'Create a secret' page in the Microsoft Azure portal. The 'Name' field is set to 'practicumsw-adminpassword' and the 'Value' field contains a redacted password. The 'Enabled' switch is set to 'Yes'. The 'Create' button is visible at the bottom left.

On the “Create a secret” page, enter values for the following items:

Name	Self-Explanatory
Value	Enter the value used when creating the Synapse Workspace

No additional changes are required.

Click the **Create** button.

## Add Access Policy to Key Vault

Navigate to your instance of Key Vault.

Click “Access Policies” in the **Settings** group of the left-hand navigation.

practicumkv | Access policies

Key vault

Search (Ctrl+ /)

Save Discard Refresh

Enable Access to:

- Azure Virtual Machines for deployment
- Azure Resource Manager for template deployment
- Azure Disk Encryption for volume encryption

Permission model

- Vault access policy (selected)
- Azure role-based access control

+ Add Access Policy

Name	Email	Key Permissions	Secret Permissions	Certificate Permissions	Action
APPLICATION					

Click the “+ Add Access Policy” link.

Principal

Select a principal

practicums

practicumsw  
e5ddc6b3-a858-4bcf-a9f0-26b5df8a27d7  
Selected

Selected items

practicumsw  
e5ddc6b3-a858-4bcf-a9f0-26b5df8a27d7  
Remove

Configure from template (optional)

Key permissions: 0 selected

Secret permissions: 2 selected

Certificate permissions: 0 selected

Select principal \*

None selected

Authorized application

None selected

Add

Select

On the “Add access policy” page, enter values for the following items:

**Secret Permissions** Select **Get** and **List** in the drop-down

**Select Principal** Search for, and then **Select** the managed identity for your Synapse Workspace

Click the **Add** button.

Dashboard > practicumkv

practicumkv | Access policies

Key vault

Search (Ctrl+ /)

Save Discard Refresh

Please click the 'Save' button to commit your changes.

Enable Access to:

Azure Virtual Machines for deployment

Azure Resource Manager for template deployment

Azure Disk Encryption for volume encryption

Permission model

Vault access policy (selected)

Azure role-based access control

+ Add Access Policy

Current Access Policies

Name	Email	Key Permissions	Secret Permissions	Certificate Permissions	Action
APPLICATION					
practicumsw		0 selected	2 selected	0 selected	De

https://ms.portal.azure.com/#

Click the **Save** button.

## Add Permissions to Data Explorer

To complete this objective, use the instructions in the Appendix to instantiate the following resources:

- Data Explorer

Navigate to your instance of Data Explorer.

Click **Permissions** in the left-hand navigation.

Click “**+ Add**” on the resulting page.

Select **AllDatabasesViewer** for read permissions and **AllDatabasesAdmin** for write, etc. permissions from the resulting drop-down.

Search for and then click to **Select** your instance of Synapse.

## Databases

### Dedicated SQL Pool

Navigate to your instance of Synapse.

Click the “**+ New dedicated SQL pool**” button.

Enter a meaningful name in “**Dedicated SQL pool name**” and choose a pricing tier.

Click the “**Review + create**” button, review configuration, and then click the **Create** button.

## Serverless SQL Database

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Data** icon in the left-hand navigation.

Click the **Workspace** tab. Click the **+** button and “**SQL database**” in the resulting drop-down.

On the “**Create SQL database**” pop-out, enter values for the following items:

Select SQL Pool Type	Confirm default, <b>Serverless</b>
<b>Database</b>	Self-Explanatory

Click the **Create** button.

## Data Explorer Database

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Data** icon in the left-hand navigation.

Click the **+** icon just above and to the right of the “**Filter resources by name**” input.

Select “**Data Explorer Database**” from the **Workspace** group in the resulting drop-down.

On the “**Data Explorer database...**” pop-out, enter values for the following items:

Pool Name	Self-Explanatory
<b>Name</b>	Self-Explanatory
<b>Default Retention Period...</b>	Confirm default, <b>365</b>
<b>Default Cache Period...</b>	Confirm default, <b>31</b>

Click the **Create** button.

## Linked Services

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Manage** icon and then “**Linked Services**” in the left-hand navigation.

Click the “**Create linked service**” (or “**+ New**”) button.

Search for and then select the resource for which you are creating a linked service; examples:

- Azure Key Vault
- Azure Data Explorer (Kusto)
- Azure Data Lake Storage Gen2
- Azure SQL Database

Click the **Continue** button.

On the “**New linked service...**” pop-out, enter values for the following common items:

<b>Name</b>	Self-Explanatory
<b>Connect via...</b>	Confirm default selection, “ <b>AutoResolveIntegrationRuntime</b> ”
<b>Selection Method</b>	Confirm default selection, “ <b>From Azure subscription</b> ”
<b>Subscription</b>	Select your subscription

And enter values for resource specific items...

### Key Vault

<b>Azure Key Vault Name</b>	Select your instance of Key Vault
-----------------------------	-----------------------------------

## Data Explorer

<b>Authentication Method</b>	Select “ <b>Managed Identity</b> ”
<b>Cluster</b>	Select your Data Explorer Cluster
<b>Database</b>	Select your Data Explorer Database

## Data Lake

<b>Authentication Method</b>	Confirm default selection, “ <b>Account key</b> ”
<b>Storage Account Name</b>	Select your Data Lake

## SQL

<b>Server Name</b>	Select your Azure SQL Server
<b>Database Name</b>	Select your Azure SQL Database
<b>Authentication Type</b>	Select “ <b>SQL authentication</b> ”
<b>User Name</b>	Enter the “ <b>server admin login</b> ” value used during instantiation of the SQL Database Server
<b>AKV Linked Service</b>	Select the name of the Linked Service created for the Key Vault
<b>Secret Name</b>	Enter the Secret Name used to capture the Azure SQL Server administrator password

Click “**Test connection**” to confirm successful connection and then click the **Create** (or **Commit**) button.

## Apache Spark Pool

Navigate to your Synapse Analytics workspace and then click the **Open** link on the “**Open Synapse Studio**” rectangle.

Click the **Manage** icon and then “**Apache Spark pools**” in the left-hand navigation.

Click the “**New Apache Spark pool**” button.

On the “**New Apache Spark pool**” pop-out, enter values for the following items:

<b>Apache Spark Pool Name</b>	Self-Explanatory
<b>Isolated Compute</b>	Confirm default, <b>Disabled</b>
<b>Node Size Family</b>	Confirm default, “ <b>Memory Optimized</b> ”
<b>Node Size</b>	Select “ <b>Small (4 vCores / 32 GB)</b> ”
<b>Autoscale</b>	Select <b>Disabled</b>
<b>Number of Nodes</b>	Slide to lowest possible value (to minimize demonstration cost)
<b>Estimated Price</b>	Review final “Est. cost per hour” and view pricing details, as desired

Click the “**Review + create**” button, validate, and then click the **Create** button.