Exercise 1 - Code-free transformation at scale with Azure Synapse Pipelines

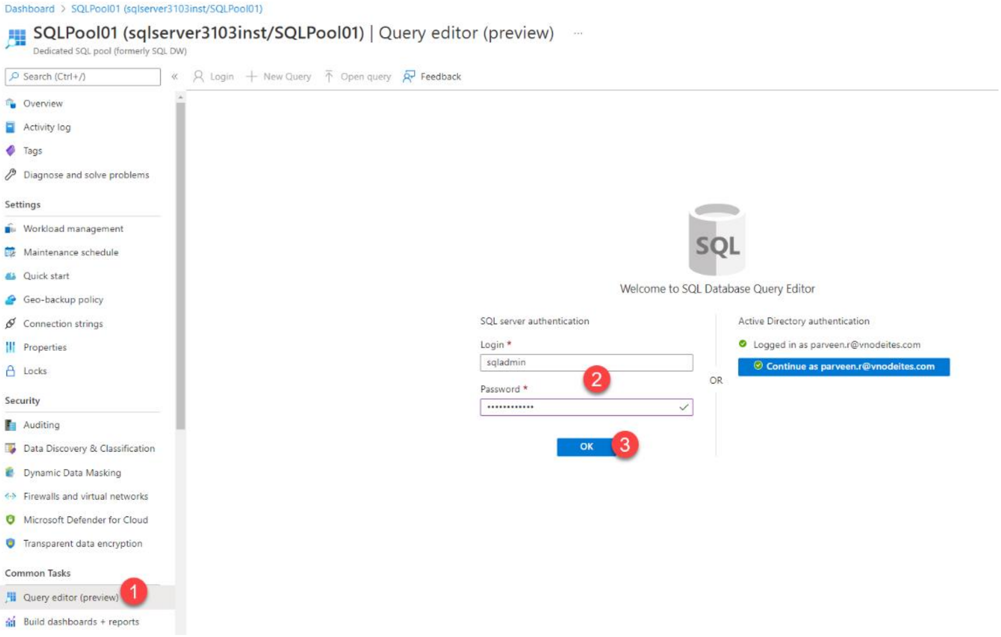
Tailwind Traders would like code-free options for data engineering tasks. Their motivation is driven by the desire to allow junior-level data engineers who understand the data but do not have a lot of development experience build and maintain data transformation operations. The other driver for this requirement is to reduce fragility caused by complex code with reliance on libraries pinned to specific versions, remove code testing requirements, and improve ease of long-term maintenance.

Their other requirement is to maintain transformed data in a data lake in addition to the dedicated SQL pool. This gives them the flexibility to retain more fields in their data sets than they otherwise store in fact and dimension tables, and doing this allows them to access the data when they have paused the dedicated SQL pool, as a cost optimization.

Create SQL table

The Mapping Data Flow we will build will write user purchase data to a dedicated SQL pool. Tailwind Traders does not yet have a table to store this data. We will execute a SQL script to create this table as a pre-requisite.

1. In Dedicated SQL pool (formerly SQL DW), navigate to the **Common Tasks** and then **Query Editor**



Dashboard > SQLPool01 (sqlserver3103inst/SQLPool01)

SQLPool01 (sqlserver3103inst/SQLPool01) | Query editor (preview)

Dedicated SQL pool (formerly SQL DW)

Welcome to SQL Database Query Editor

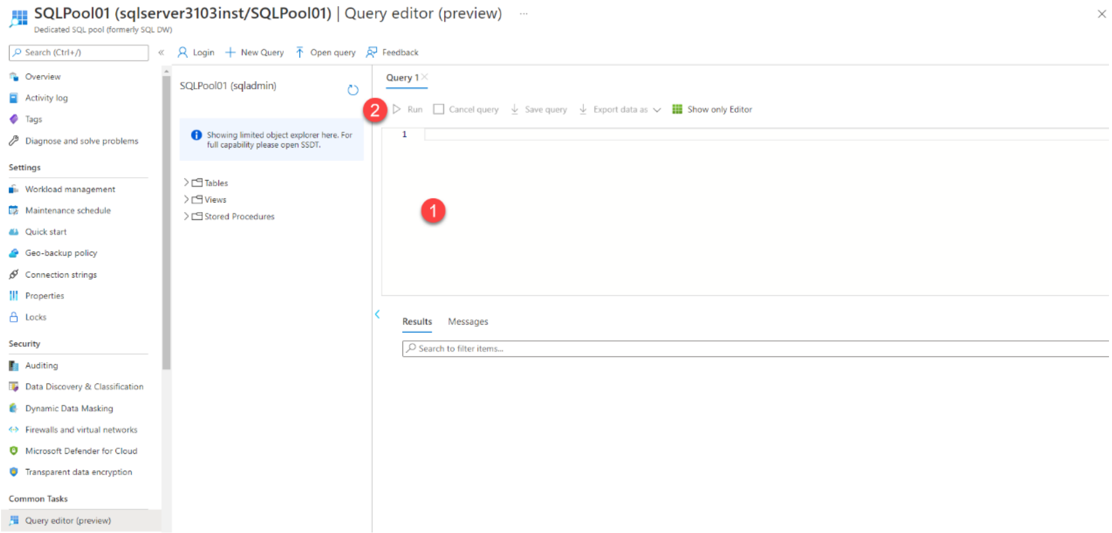
SQL server authentication

Login *: sqladmin

Password *: •••••••••••••

Active Directory authentication

Logged in as parveen.r@vnodeites.com



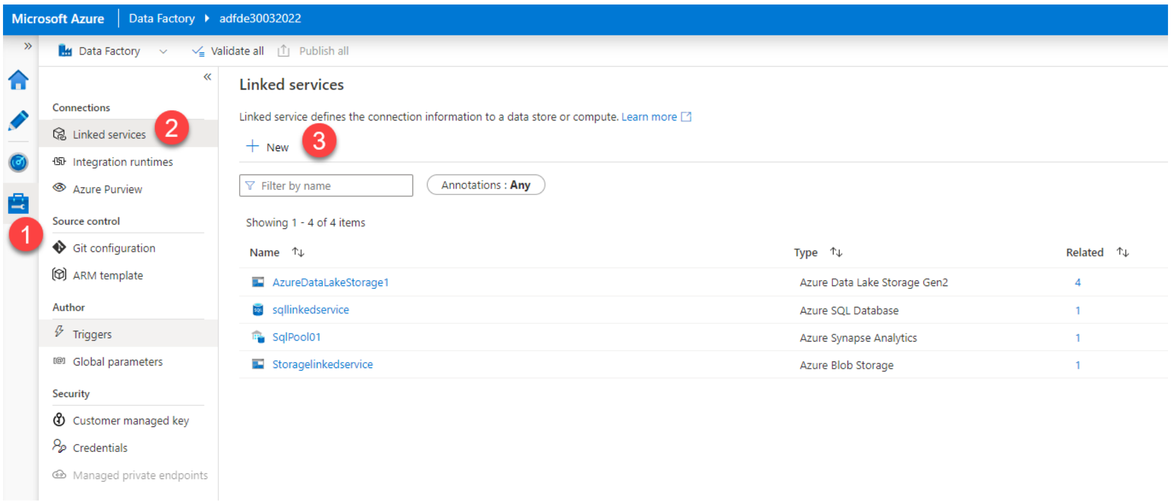
2. In the query window, create schema using the command and Run

```
create schem wwi

CREATE TABLE [wwi].[UserTopProductPurchases]
(
    [UserId] [int]  NOT NULL,
    [ProductId] [int]  NOT NULL,
    [ItemsPurchasedLast12Months] [int]  NULL,
    [IsTopProduct] [bit]  NOT NULL,
    [IsPreferredProduct] [bit]  NOT NULL
)
WITH
(
    DISTRIBUTION = HASH ( [UserId] ),
    CLUSTERED COLUMNSTORE INDEX
)

CREATE TABLE [wwi].[CampaignAnalytics]
(
    [Region] [nvarchar](50)  NOT NULL,
    [Country] [nvarchar](30)  NOT NULL,
    [ProductCategory] [nvarchar](50)  NOT NULL,
    [CampaignName] [nvarchar](500)  NOT NULL,
    [Revenue] [decimal](10,2)  NULL,
    [RevenueTarget] [decimal](10,2)  NULL,
    [City] [nvarchar](50)  NULL,
    [State] [nvarchar](25)  NULL
)
WITH
(
    DISTRIBUTION = HASH ( [Region] ),
    CLUSTERED COLUMNSTORE INDEX
)
```

Task 2: Create linked service

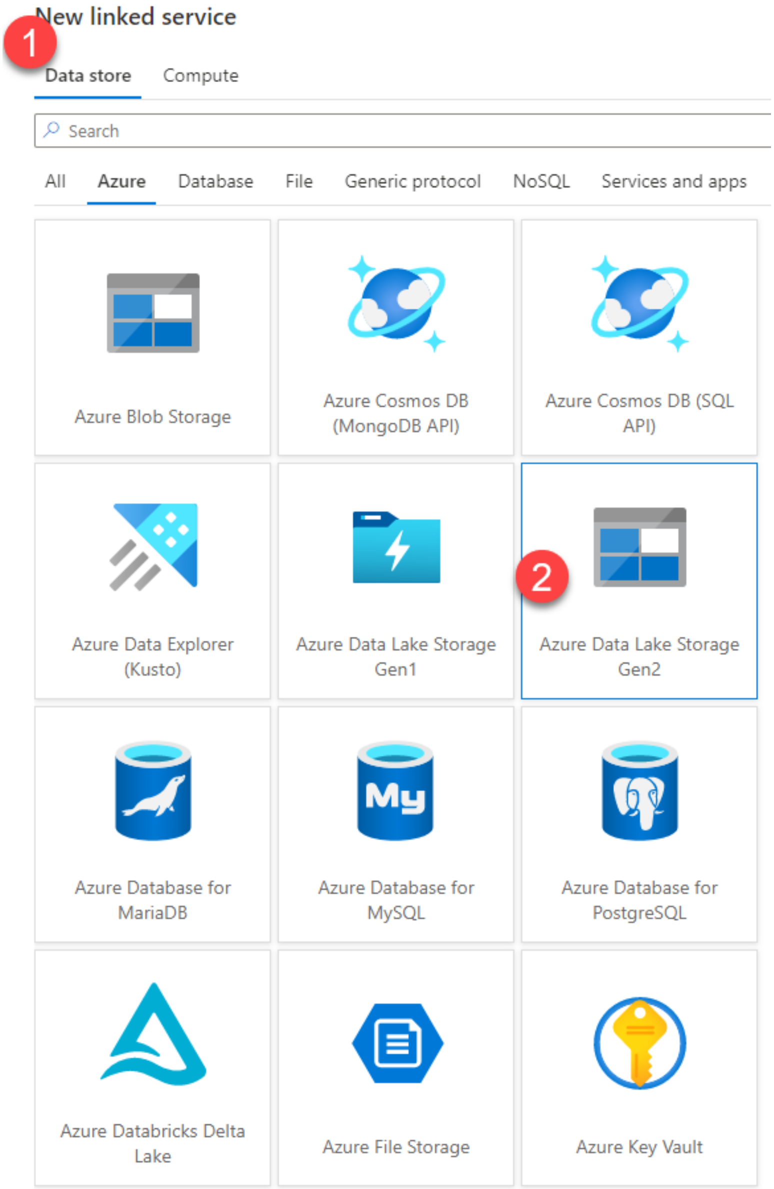


Linked services

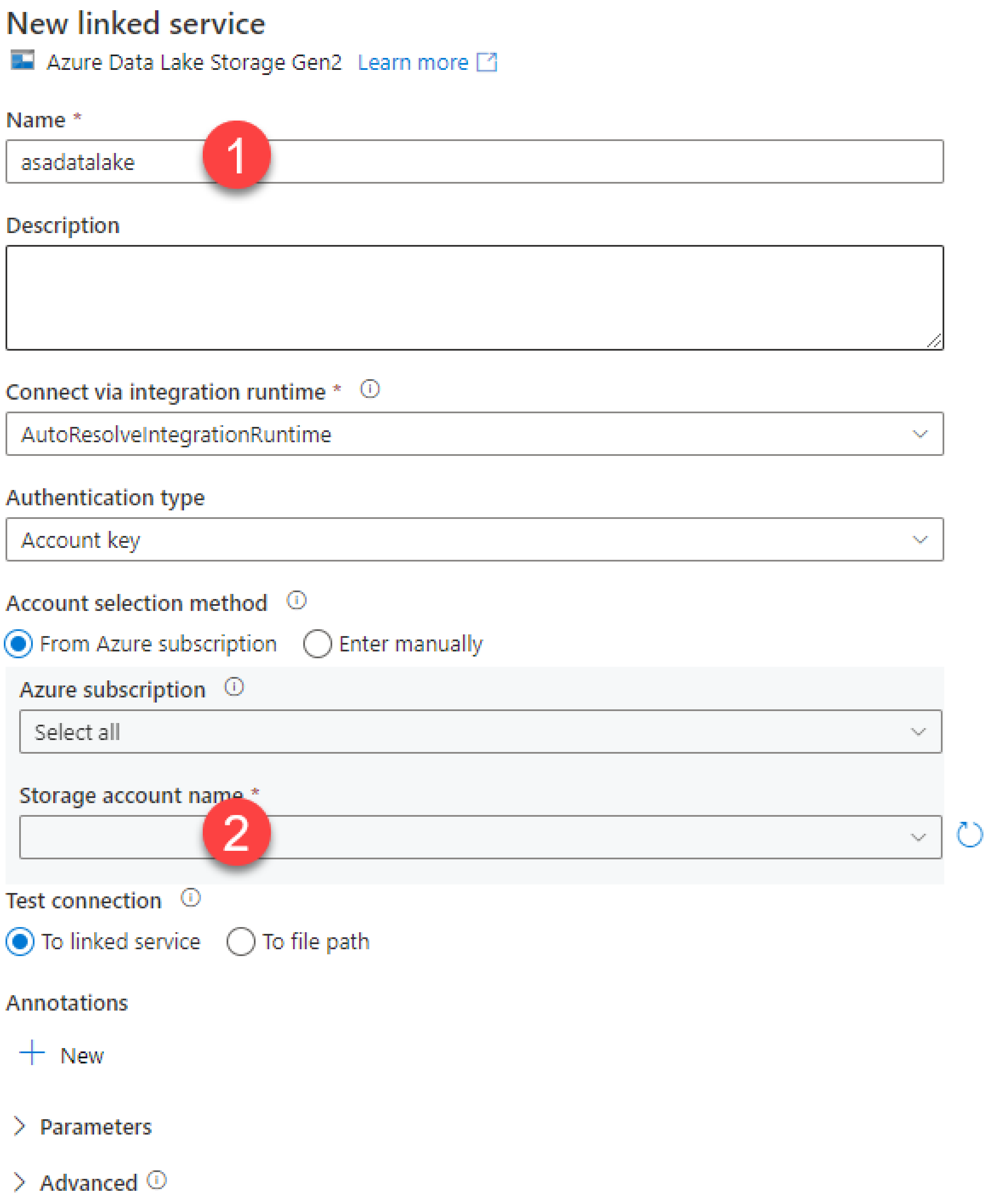
Linked service defines the connection information to a data store or compute. Learn more

Showing 1 - 4 of 4 items

Name	Type	Related
AzureDataLakeStorage1	Azure Data Lake Storage Gen2	4
sqllinkedservice	Azure SQL Database	1
SqlPool01	Azure Synapse Analytics	1
Storagelinkedservice	Azure Blob Storage	1


New linked service
1
Data store
Compute
Search
All
Azure
Database
File
Generic protocol
NoSQL
Services and apps
Azure Blob Storage
Azure Cosmos DB (MongoDB API)
Azure Cosmos DB (SQL API)
Azure Data Explorer (Kusto)
Azure Data Lake Storage Gen1
2
Azure Data Lake Storage Gen2
Azure Database for MariaDB
Azure Database for MySQL
Azure Database for PostgreSQL
Azure Databricks Delta Lake
Azure File Storage
Azure Key Vault

Create Linked service for Azure Data Lake Storage Gen2, and Synapse Analytics Workspace



New linked service

Azure Data Lake Storage Gen2 Learn more

Name *

asadatalake

Description

Connect via integration runtime *

AutoResolveIntegrationRuntime

Authentication type

Account key

Account selection method

● From Azure subscription ○ Enter manually

Azure subscription

Select all

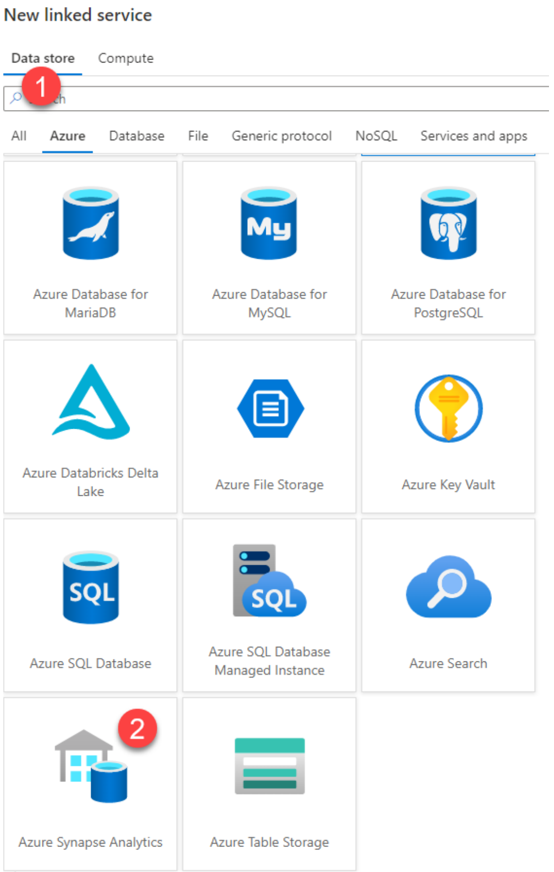
Storage account name *

Test connection

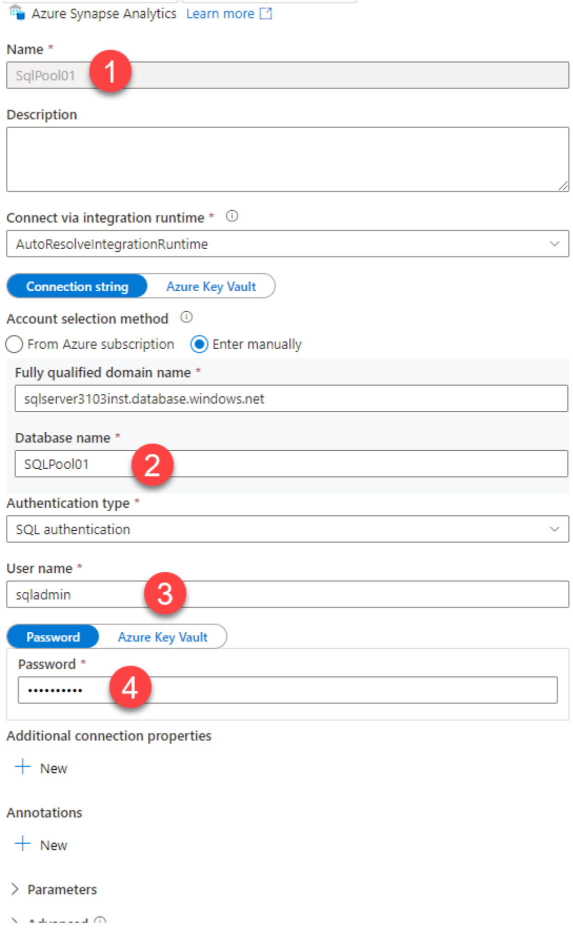
● To linked service ○ To file path

Annotations

- **Linked service Name:** asadatalake.



New linked service



Azure Synapse Analytics Learn more

Name * SqlPool01

Description

Connect via integration runtime * AutoResolveIntegrationRuntime

Connection string | Azure Key Vault

Account selection method ○ From Azure subscription ● Enter manually

Fully qualified domain name * sqlserver3103inst.database.windows.net

Database name * SQLPool01

Authentication type * SQL authentication

User name * sqladmin

Password | Azure Key Vault

Password * ••••••••••

Additional connection properties

+ New

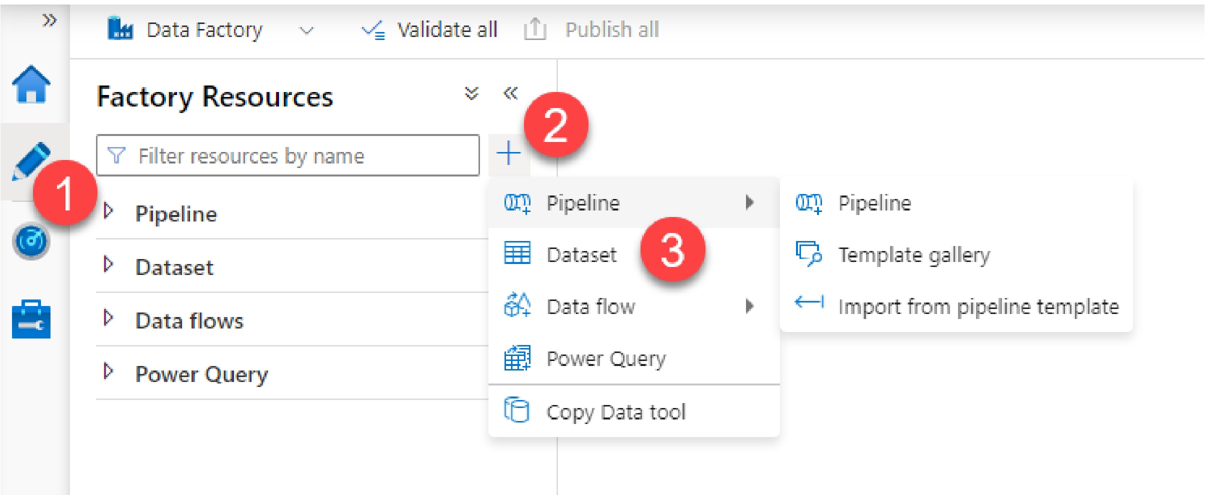
Annotations

+ New

> Parameters

Linked service Name: SqlPool01

Task 3: Create data sets



1. Configure the dataset as follows, then select **OK**:

- **Name**: Enter `asal400_ecommerce_userprofiles_source`.
- **Linked service**: Select the **asadatalake*xxxxxxx*** linked service.
- **File path**: Browse to the **wwi-02/online-user-profiles-02** path.
- **Import schema**: Select **From connection/store**.

2. Configure the dataset as follows, then select **OK**:

- **Name**: Enter `asal400_wwi_campaign_analytics_asa`.
- **Linked service**: Select the **SqlPool01** .
- **Table name**: Select **wwi.CampaignAnalytics**.
- **Import schema**: Select **From connection/store**.

3. Configure the dataset as follows, then select **OK**:

- **Name**: Enter `asal400_wwi_usertopproductpurchases_asa`.
- **Linked service**: Select the **SqlPool01**.
- **Table name**: Select **wwi.UserTopProductPurchases**.
- **Import schema**: Select **From connection/store**.

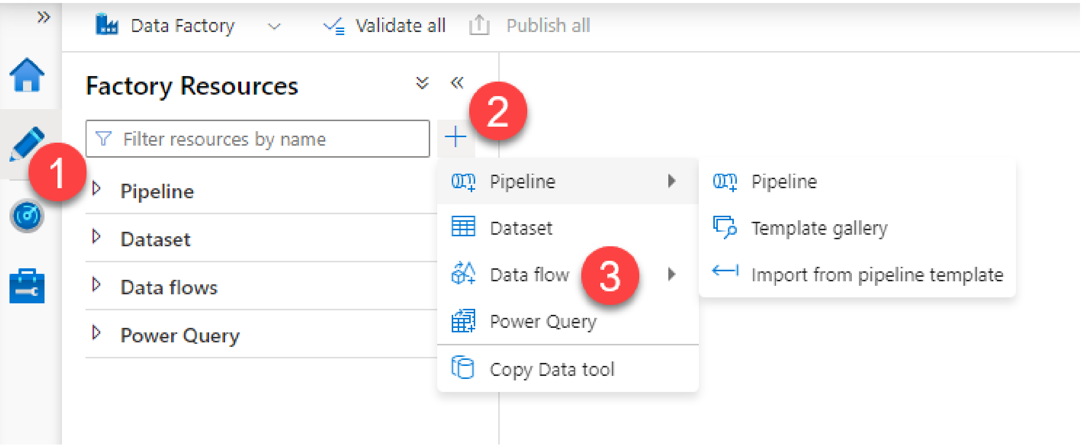
4. Configure the dataset as follows, then select **OK**:

- **Name**: Enter `asal400_campaign_analytics_source`.
- **Linked service**: Select the **asadatalake*xxxxxxx*** linked service.

- **File path**: Browse to **wwi-02/campaign-analytics/campaignanalytics.csv**.
- **First row as header**: Leave unchecked (we are skipping the header because there is a mismatch between the number of columns in the header and the number of columns in the data rows).
- **Import schema**: Select **From connection/store**.

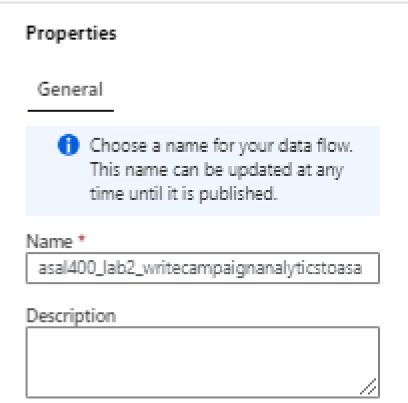
Task 3: Create campaign analytics data flow

1. Navigate to the **Author** hub.



2.

1. In the **General** settings of the **Properties** blade of the new data flow, change the **Name** to `asal400_lab2_writecampaignanalyticstoasa`.

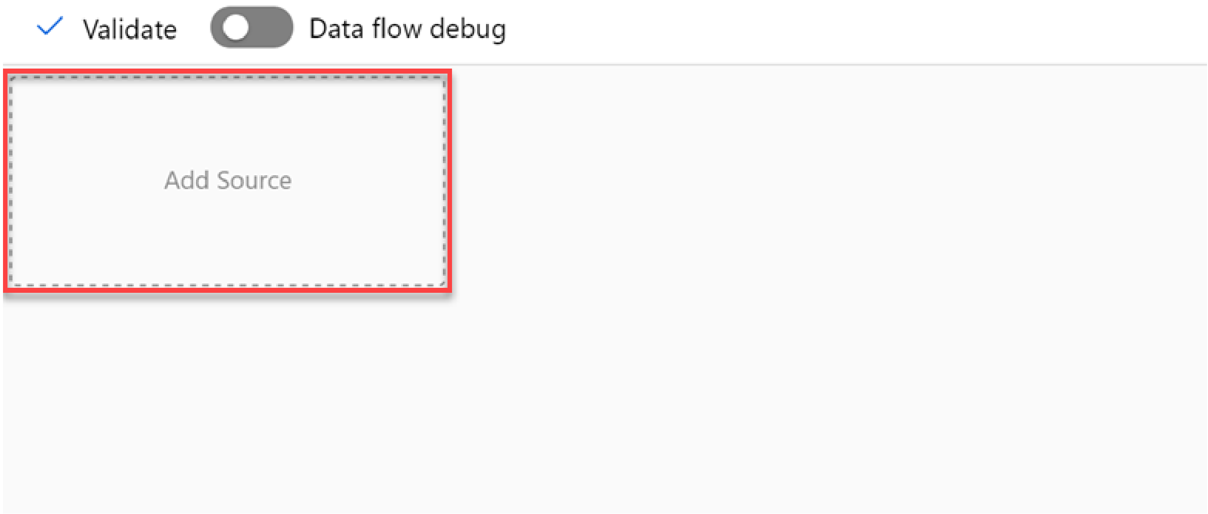


Choose a name for your data flow. This name can be updated at any time until it is published.

Name *: asal400_lab2_writecampaignanalyticstoasa

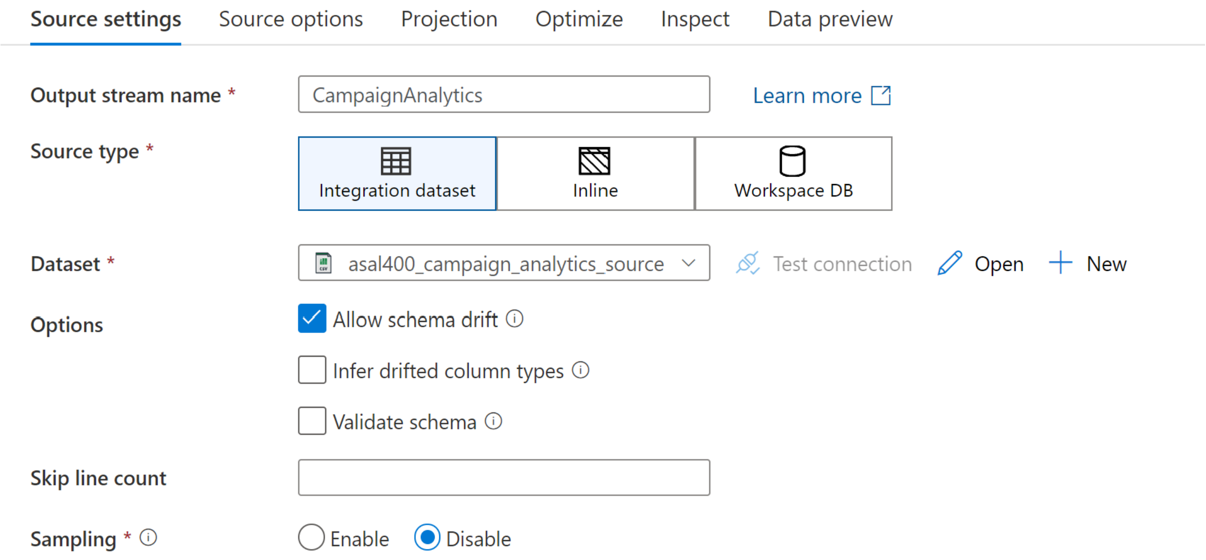
Description:

2. Select **Add Source** on the data flow canvas (again, if a tip is displayed, close it.)



3. Under **Source settings**, configure the following:

    - **Output stream name**: Enter `CampaignAnalytics`.
    - **Source type**: Select **Integration dataset**.
    - **Dataset**: Select **asal400_campaign_analytics_source**.
    - **Options**: Select **Allow schema drift** and leave the other options unchecked.
    - **Skip line count**: Enter `1`. This allows us to skip the header row which has two fewer columns than the rest of the rows in the CSV file, truncating the last two data columns.
    - **Sampling**: Select **Disable**.



Output stream name * CampaignAnalytics

Source type * Integration dataset

Dataset * asal400_campaign_analytics_source

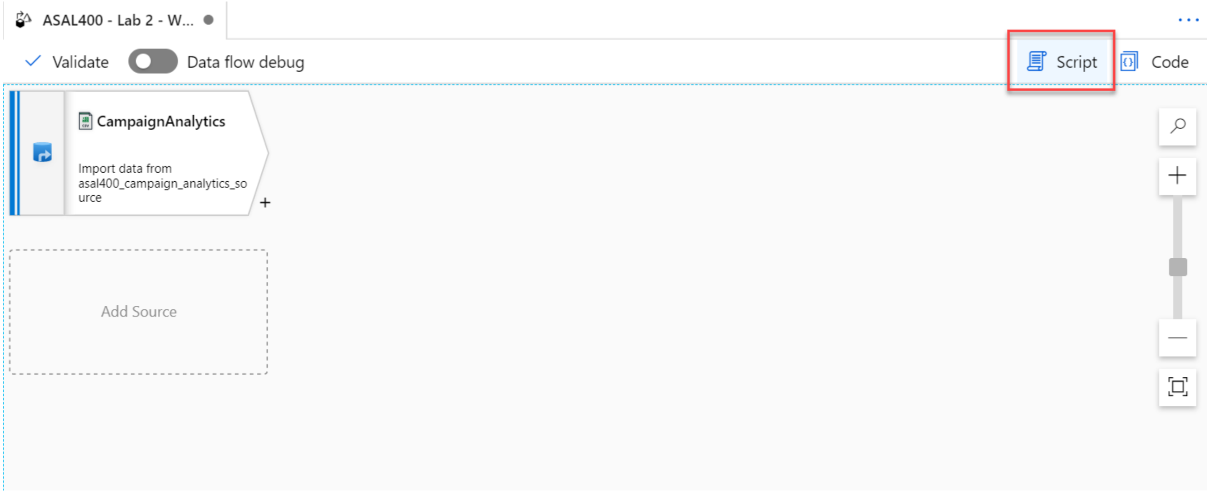
Options ☑ Allow schema drift ⓘ
☐ Infer drifted column types ⓘ
☐ Validate schema ⓘ

Skip line count

Sampling * ⓘ ○ Enable ● Disable

When you create data flows, certain features are enabled by turning on debug, such as previewing data and importing a schema (projection). Due to the amount of time it takes to enable this option, and to minimize resource consumption in the lab environment, we will bypass these features.

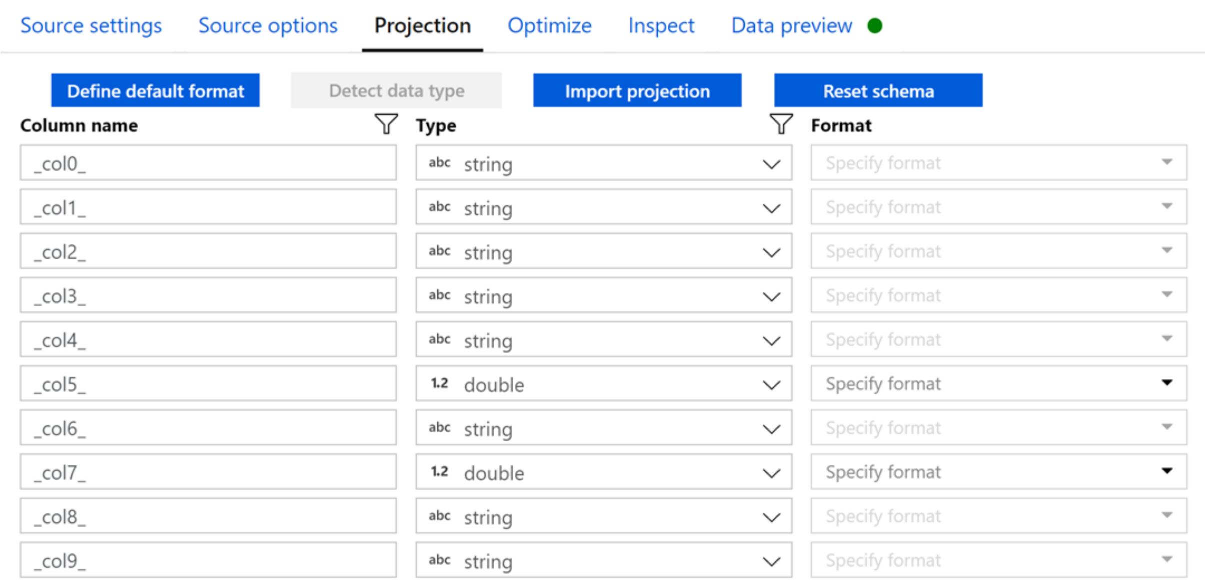
4. The data source has a schema we need to set. To do this, select **Script** above the design canvas.



5. Replace the script with the following to provide the column mappings, then select **OK**:

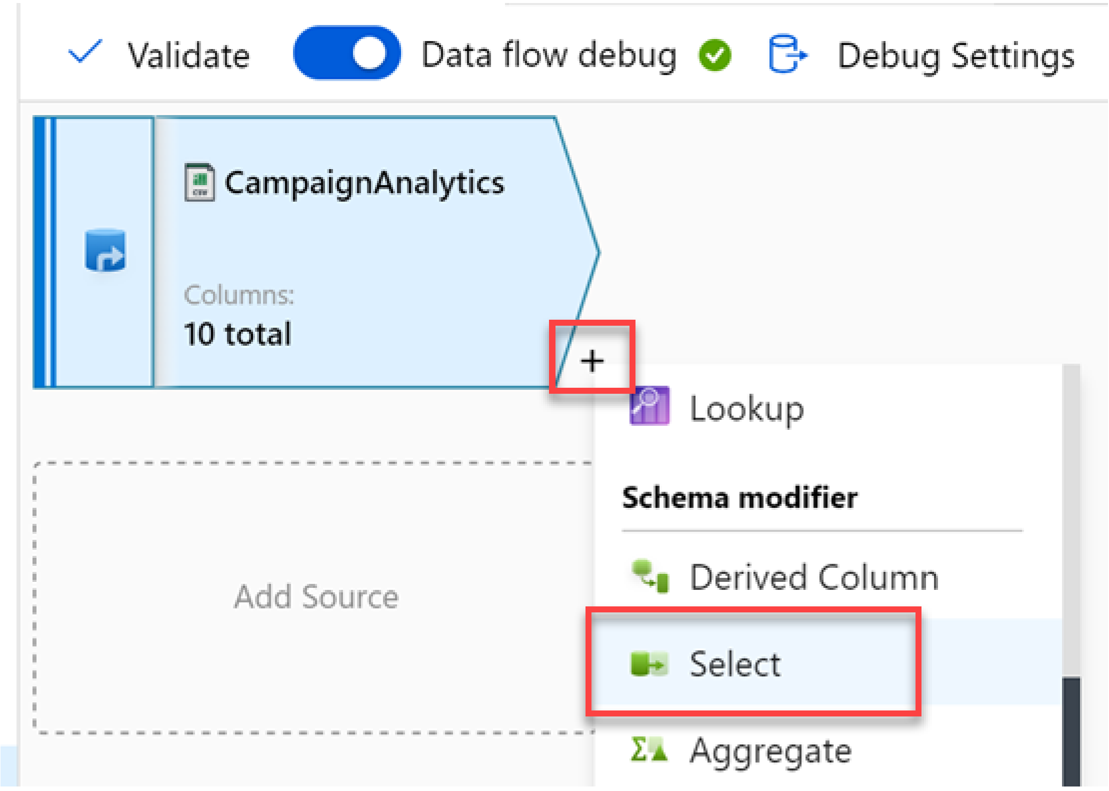
```
source(output(
        {_col0_} as string,
        {_col1_} as string,
        {_col2_} as string,
        {_col3_} as string,
        {_col4_} as string,
        {_col5_} as double,
        {_col6_} as string,
        {_col7_} as double,
        {_col8_} as string,
        {_col9_} as string
    ),
    allowSchemaDrift: true,
    validateSchema: false,
    ignoreNoFilesFound: false,
                  skipLines: 1) ~> CampaignAnalytics
```

6. Select the **CampaignAnalytics** data source, then select **Projection**. The projection should display the following schema:



Column name	Type	Format
col0	abc string	Specify format
col1	abc string	Specify format
col2	abc string	Specify format
col3	abc string	Specify format
col4	abc string	Specify format
col5	1.2 double	Specify format
col6	abc string	Specify format
col7	1.2 double	Specify format
col8	abc string	Specify format
col9	abc string	Specify format

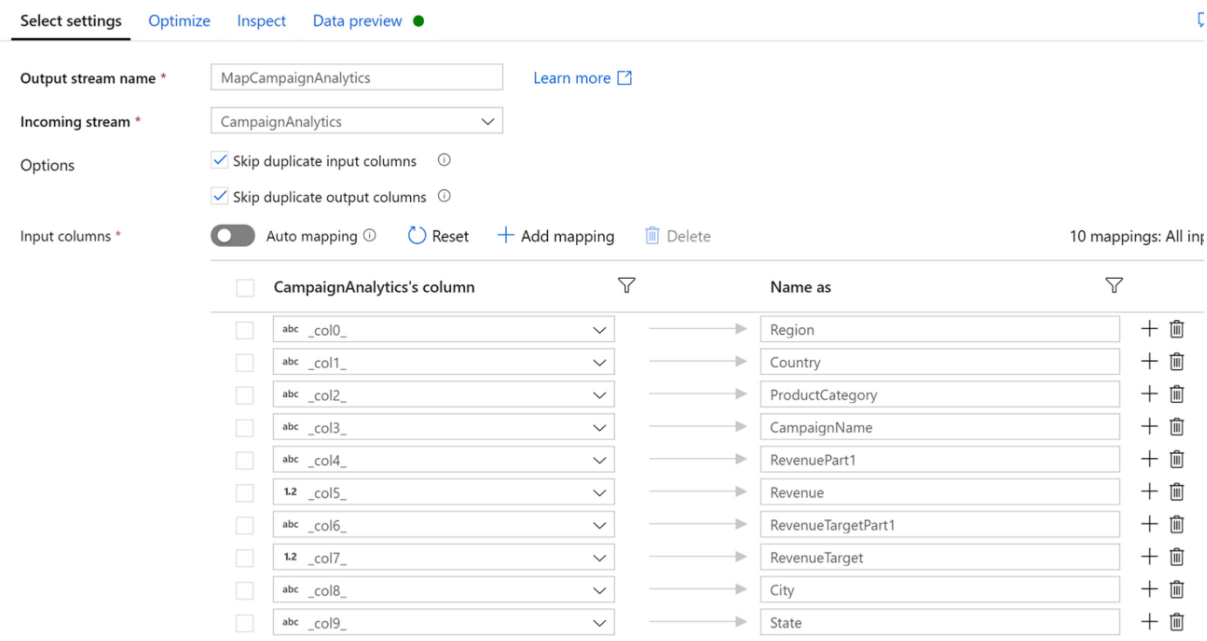
7. Select the **+** to the right of the **CampaignAnalytics** step, then select the **Select** schema modifier.



8. Under **Select settings**, configure the following:

   - **Output stream name**: Enter `MapCampaignAnalytics`.
   - **Incoming stream**: Select **CampaignAnalytics**.
   - **Options**: Check both options.
   - **Input columns**: make sure **Auto mapping** is unselected, then provide the following values in the **Name as** fields:

- Region
- Country
- ProductCategory
- CampaignName
- RevenuePart1
- Revenue
- RevenueTargetPart1
- RevenueTarget
- City
- State



Select settings | Optimize | Inspect | Data preview ●

Output stream name * MapCampaignAnalytics Learn more

Incoming stream * CampaignAnalytics

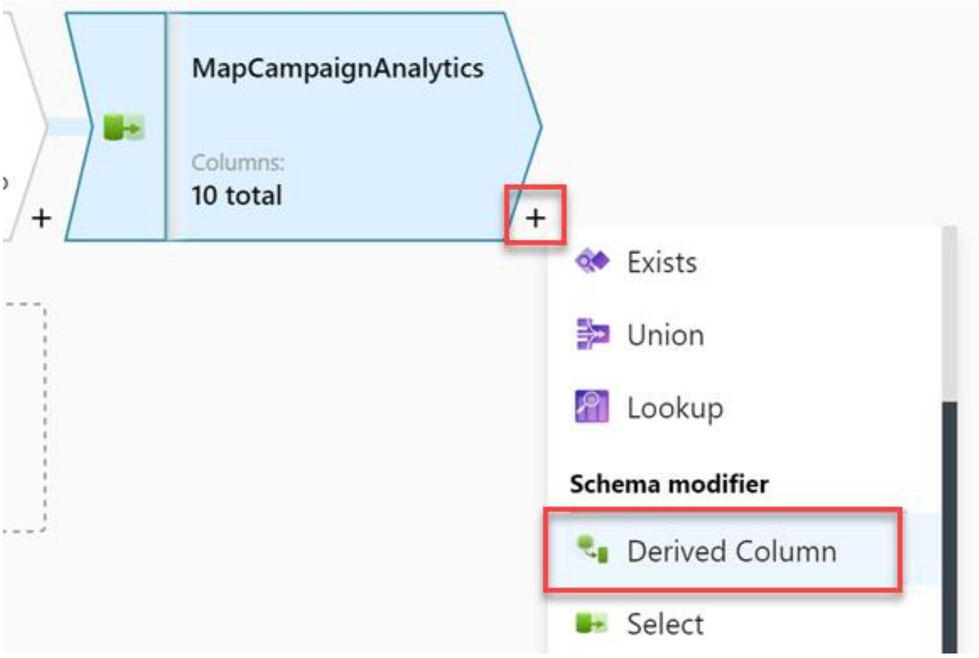
Options ☑ Skip duplicate input columns

☑ Skip duplicate output columns

Input columns * Auto mapping | Reset | Add mapping | Delete | 10 mappings: All inp

□ CampaignAnalytics's column	Name as
□ abc _col0_	Region
□ abc _col1_	Country
□ abc _col2_	ProductCategory
□ abc _col3_	CampaignName
□ abc _col4_	RevenuePart1
□ 1.2 _col5_	Revenue
□ abc _col6_	RevenueTargetPart1
□ 1.2 _col7_	RevenueTarget
□ abc _col8_	City
□ abc _col9_	State

9. Select the **+** to the right of the **MapCampaignAnalytics** step, then select the **Derived Column** schema modifier.



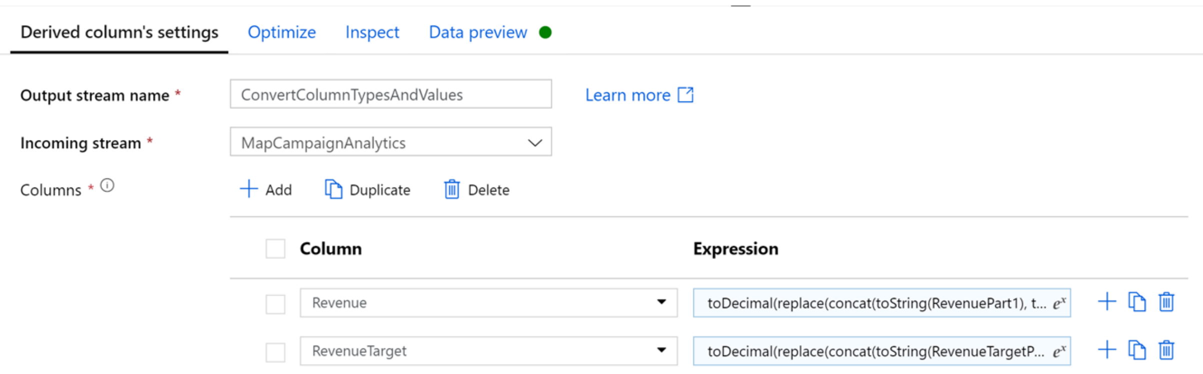
10. Under **Derived column's settings**, configure the following:

   - **Output stream name**: Enter `ConvertColumnTypesAndValues`.
   - **Incoming stream**: Select **MapCampaignAnalytics**.
   - **Columns**: Provide the following information:

Column	Expression
Revenue	`toDecimal(replace(concat(toString(RevenuePart1), toString(Revenue)), '\\', ''), 10, 2, '$###,###.##')`
RevenueTarget	`toDecimal(replace(concat(toString(RevenueTargetPart1), toString(RevenueTarget)), '\\', ''), 10, 2, '$###,###.##')`

11. **Note**: To insert the second column, select **+ Add** above the Columns list, then select **Add column**.

12.



Output stream name * ConvertColumnTypesAndValues

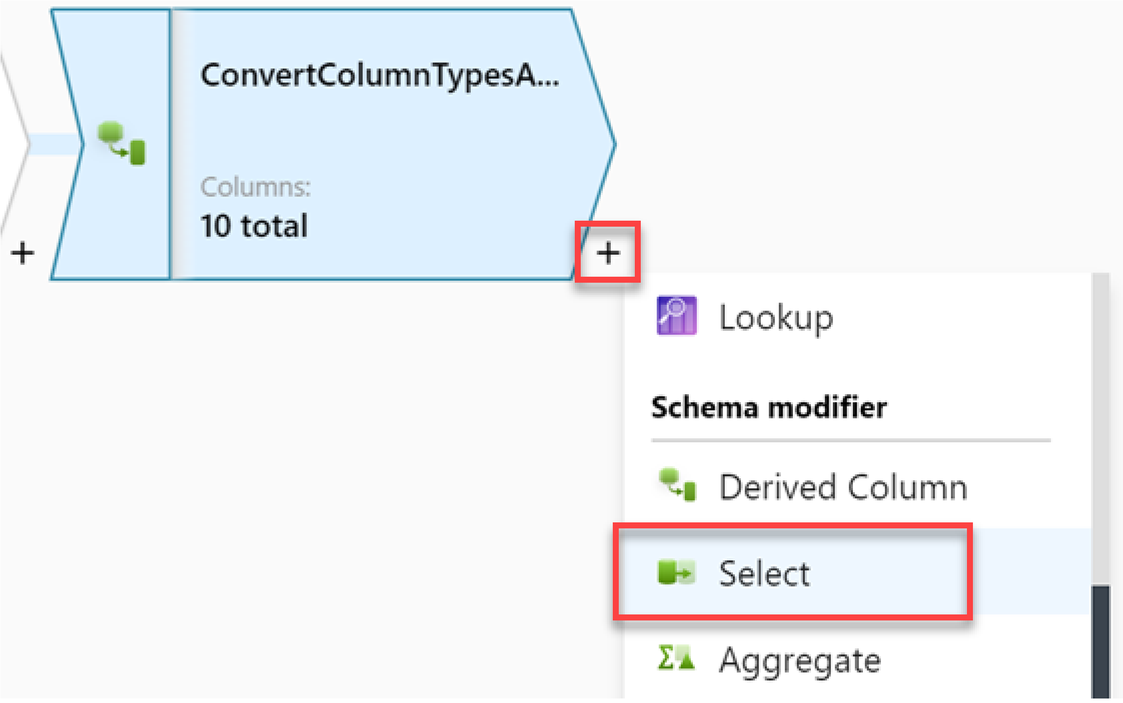
Incoming stream * MapCampaignAnalytics

Columns * ⓘ

	Column	Expression
□	Revenue	toDecimal(replace(concat(toString(RevenuePart1), t...
□	RevenueTarget	toDecimal(replace(concat(toString(RevenueTargetP...

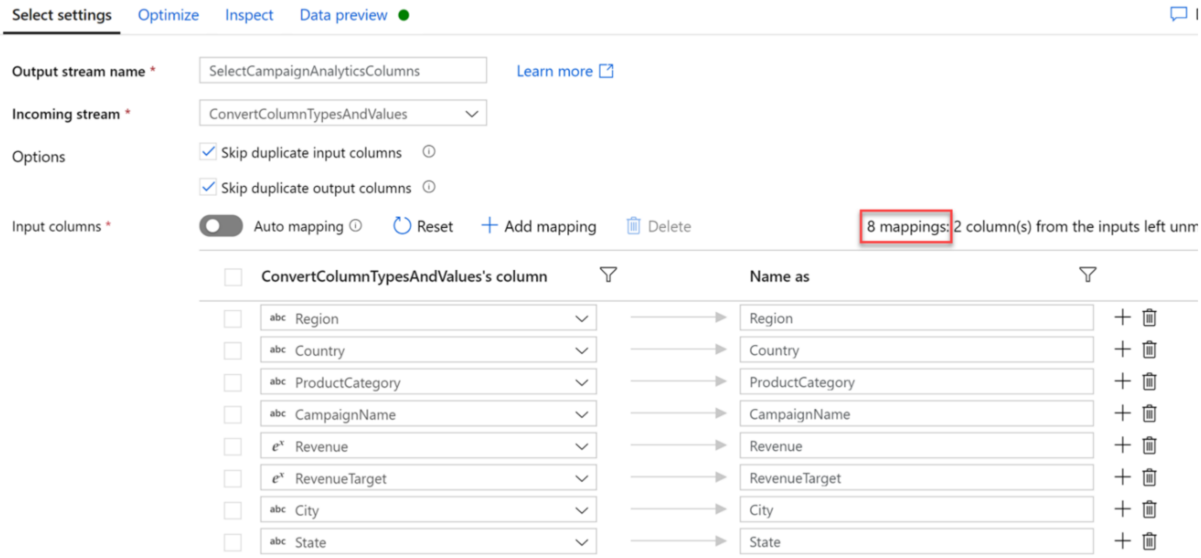
13. The expressions you defined will concatenate and clean-up the **RevenuePart1** and **Revenue** values and the **RevenueTargetPart1** and **RevenueTarget** values.

14. Select the **+** to the right of the **ConvertColumnTypesAndValues** step, then select the **Select** schema modifier from the context menu.



15. Under **Select settings**, configure the following:

   - **Output stream name**: Enter `SelectCampaignAnalyticsColumns`.
   - **Incoming stream**: Select **ConvertColumnTypesAndValues**.
   - **Options**: Check both options.
   - **Input columns**: make sure **Auto mapping** is unchecked, then **Delete RevenuePart1** and **RevenueTargetPart1**. We no longer need these fields.



Output stream name * SelectCampaignAnalyticsColumns

Incoming stream * ConvertColumnTypesAndValues

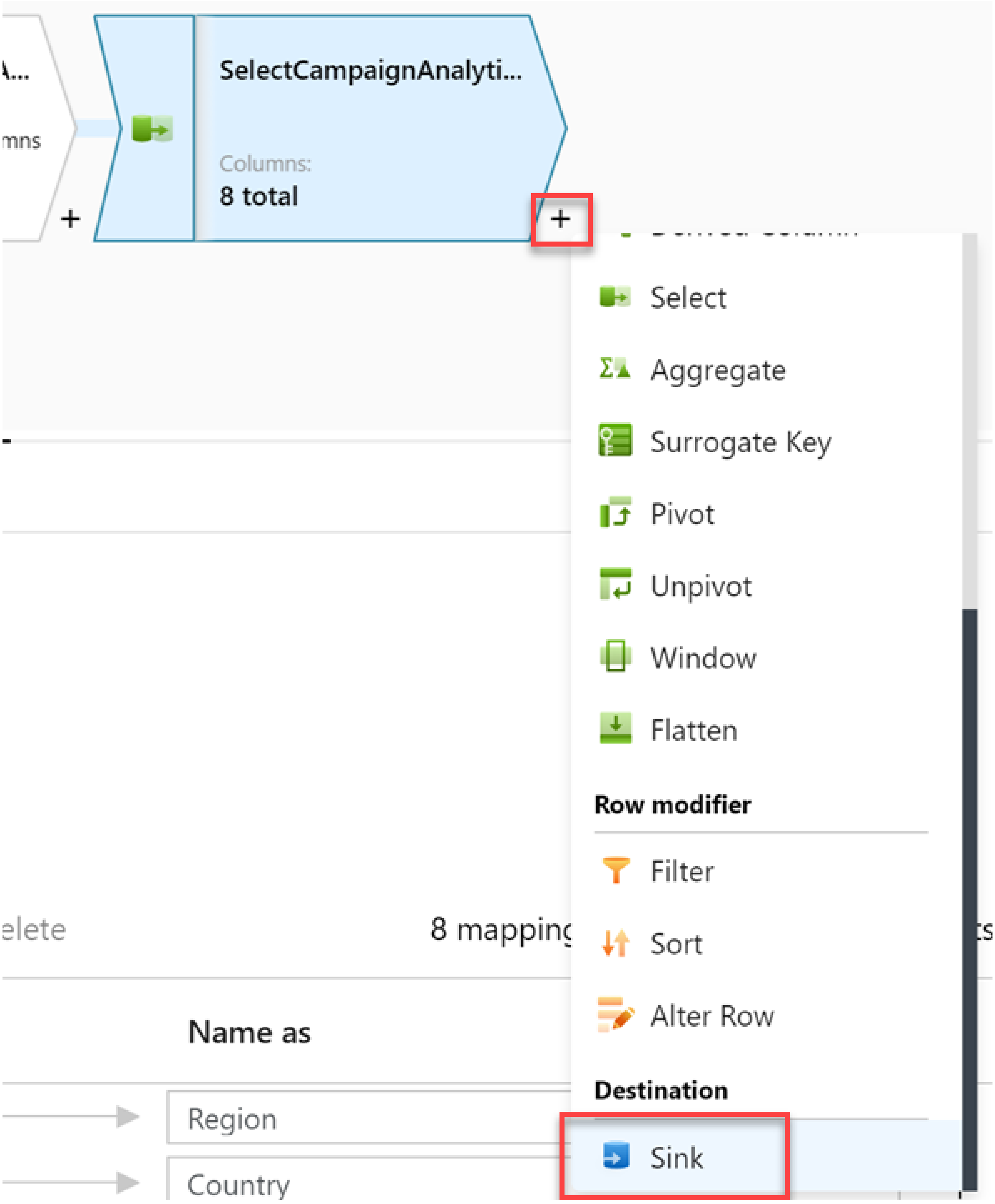
Options ☑ Skip duplicate input columns
☑ Skip duplicate output columns

Input columns * Auto mapping

8 mappings: 2 column(s) from the inputs left unm

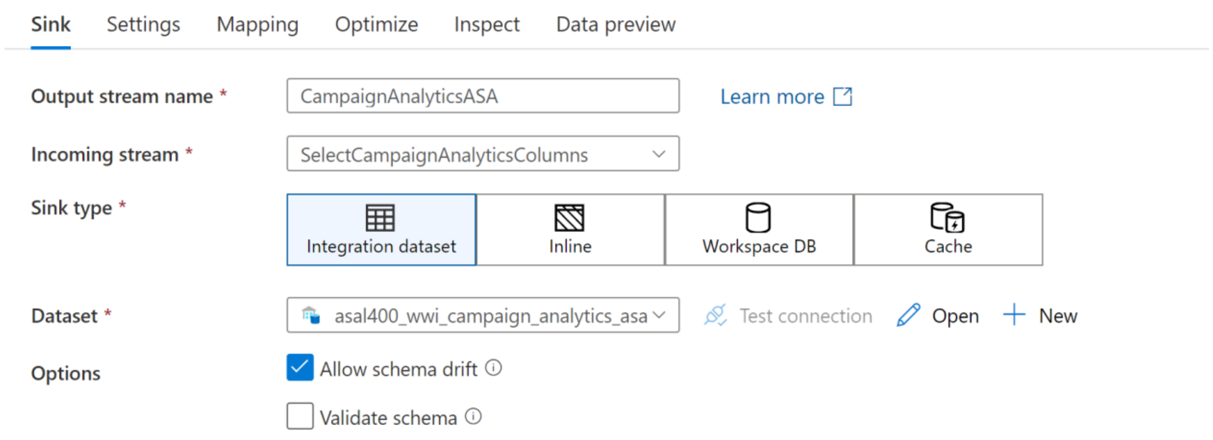
ConvertColumnTypesAndValues's column	Name as
abc Region	Region
abc Country	Country
abc ProductCategory	ProductCategory
abc CampaignName	CampaignName
eˣ Revenue	Revenue
eˣ RevenueTarget	RevenueTarget
abc City	City
abc State	State

16. Select the **+** to the right of the **SelectCampaignAnalyticsColumns** step, then select the **Sink** destination.



17. Under **Sink**, configure the following:

    - **Output stream name**: Enter `CampaignAnalyticsASA`.
    - **Incoming stream**: Select **SelectCampaignAnalyticsColumns**.
    - **Sink type**: Select **Integration dataset**.
    - **Dataset**: Select **asal400_wwi_campaign_analytics_asa**.
    - **Options**: Check **Allow schema drift** and uncheck **Validate schema**.



Output stream name * CampaignAnalyticsASA Learn more

Incoming stream * SelectCampaignAnalyticsColumns

Sink type * [✓ Integration dataset] Inline Workspace DB Cache

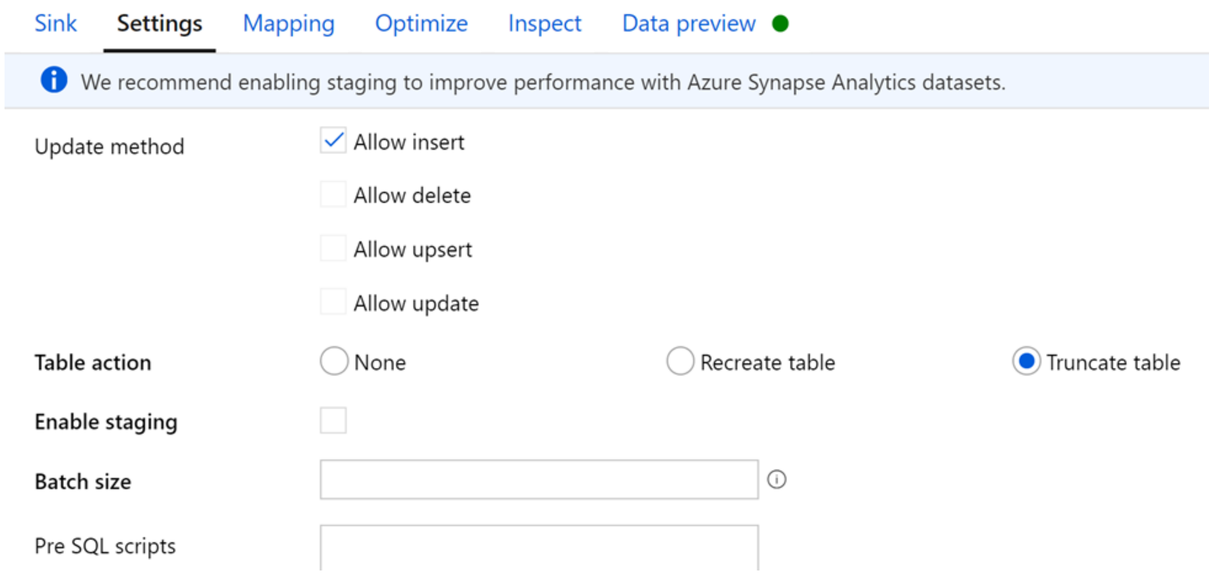
Dataset * asal400_wwi_campaign_analytics_asa

Options ☑ Allow schema drift

□ Validate schema

18. On the **Settings** tab, configure the following options:

    - **Update method**: Check **Allow insert** and leave the rest unchecked.
    - **Table action**: Select **Truncate table**.
    - **Enable staging**: Uncheck this option. The sample CSV file is small, making the staging option unnecessary.



We recommend enabling staging to improve performance with Azure Synapse Analytics datasets.

Update method ☑ Allow insert

□ Allow delete

□ Allow upsert

□ Allow update

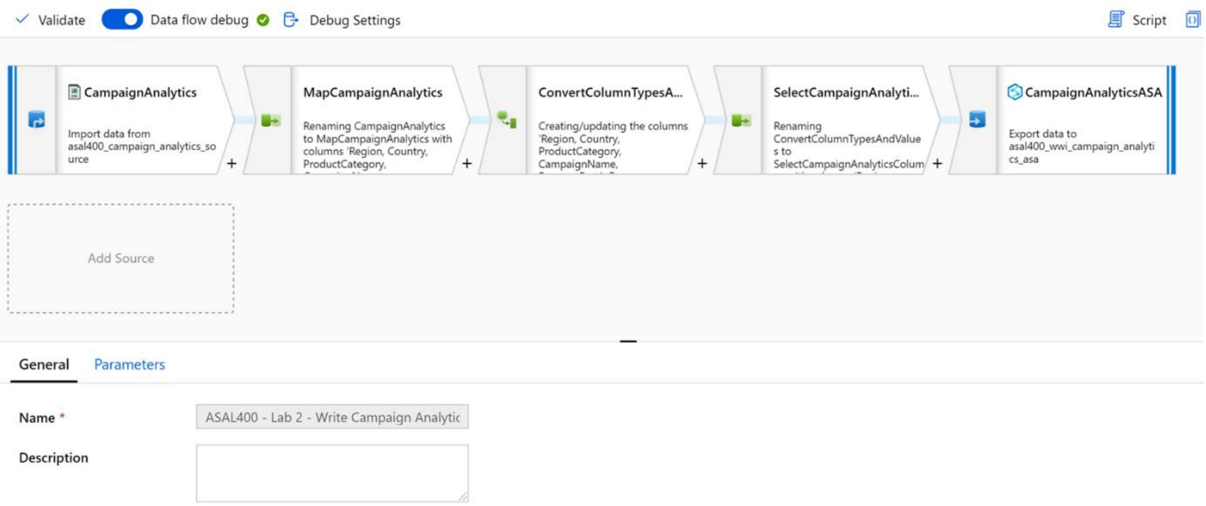
Table action ○ None ○ Recreate table ● Truncate table

Enable staging □

Batch size

Pre SQL scripts

19. Your completed data flow should look similar to the following:



Import data from asal400_campaign_analytics_source

Renaming CampaignAnalytics to MapCampaignAnalytics with columns 'Region, Country, ProductCategory,

Creating/updating the columns 'Region, Country, ProductCategory, CampaignName,

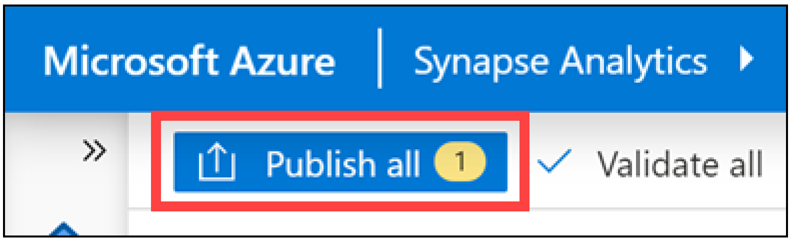
Renaming ConvertColumnTypesAndValues to SelectCampaignAnalyticsColum

Export data to asal400_wwi_campaign_analytics_asa

Name * ASAL400 - Lab 2 - Write Campaign Analytic

Description

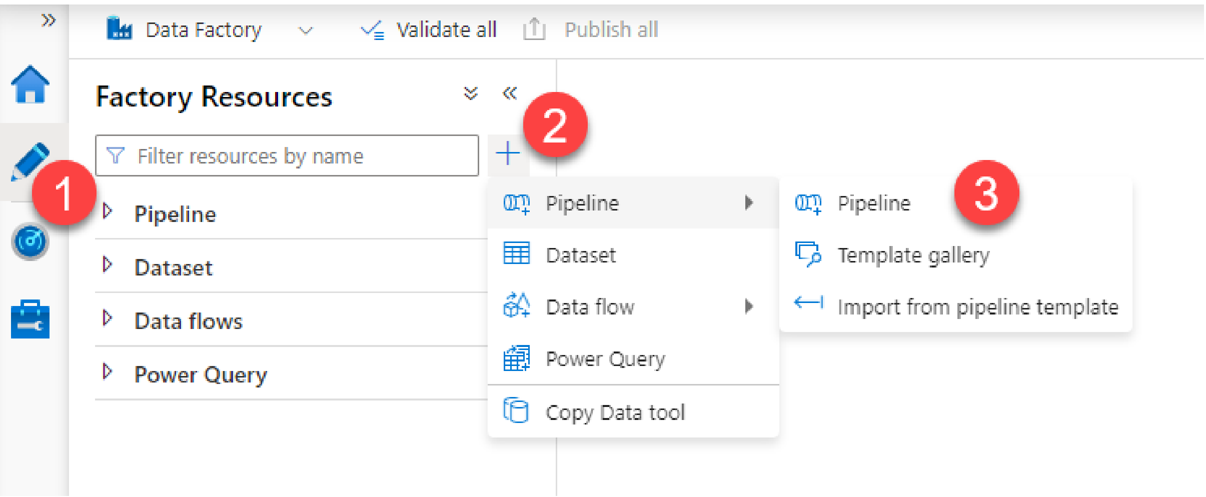
20. Select **Publish all** then **Publish** to save your new data flow.



Task 4: Create campaign analytics data pipeline

In order to run the new data flow, you need to create a new pipeline and add a data flow activity to it.

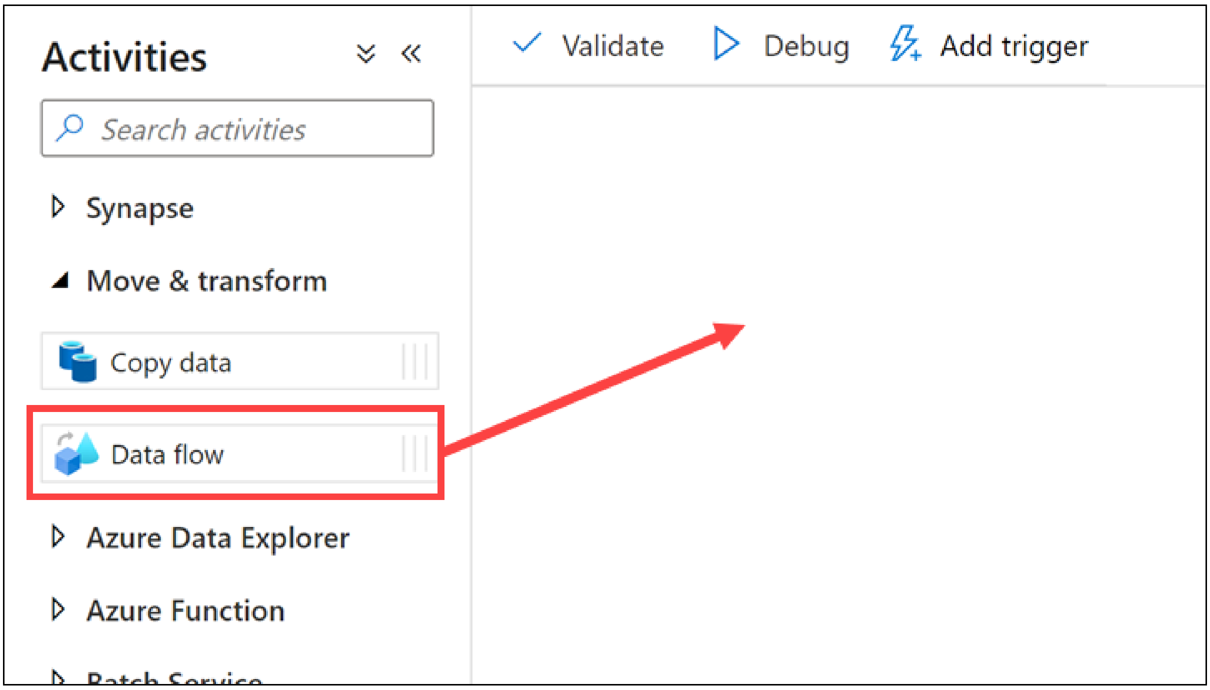
1. Navigate to the **Author** hub.



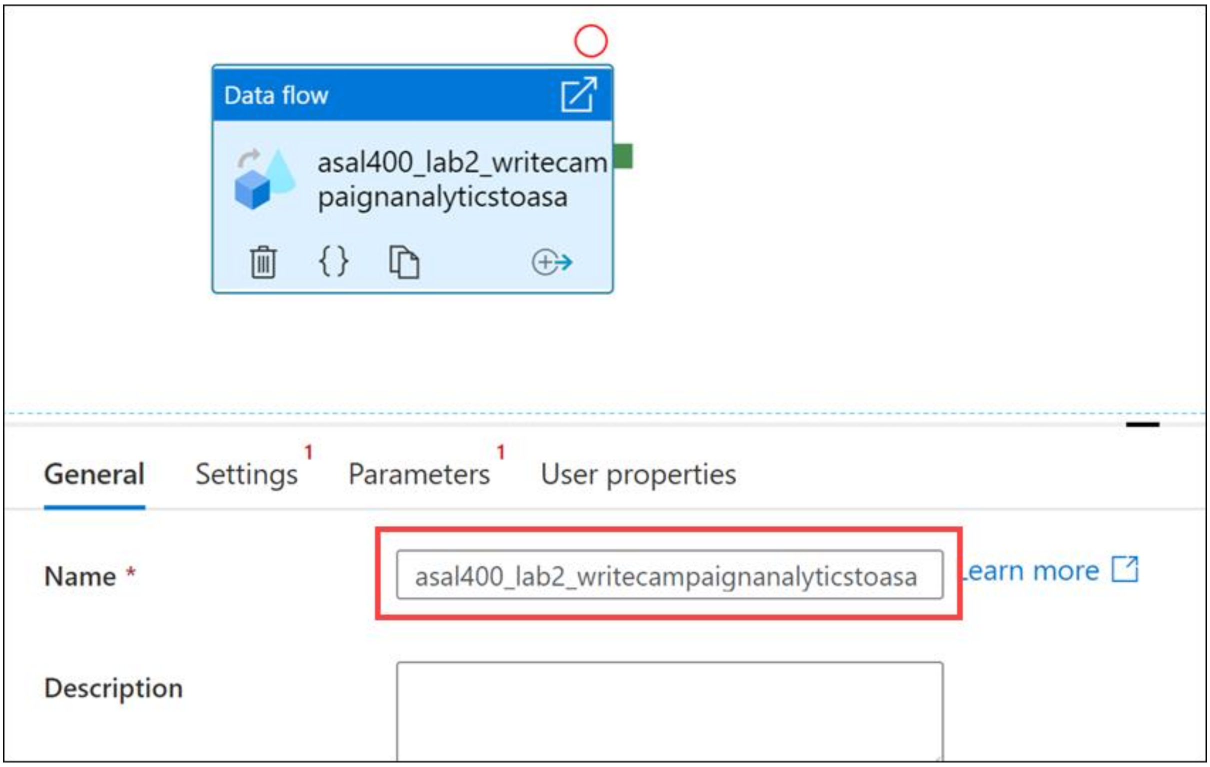
2. In the **+** menu, select **Pipeline** to create a new pipeline.

3. In the **General** section of the **Properties** blade for the new pipeline, enter the following **Name**: `Write Campaign Analytics to ASA`.

4. Expand **Move & transform** within the Activities list, then drag the **Data flow** activity onto the pipeline canvas.



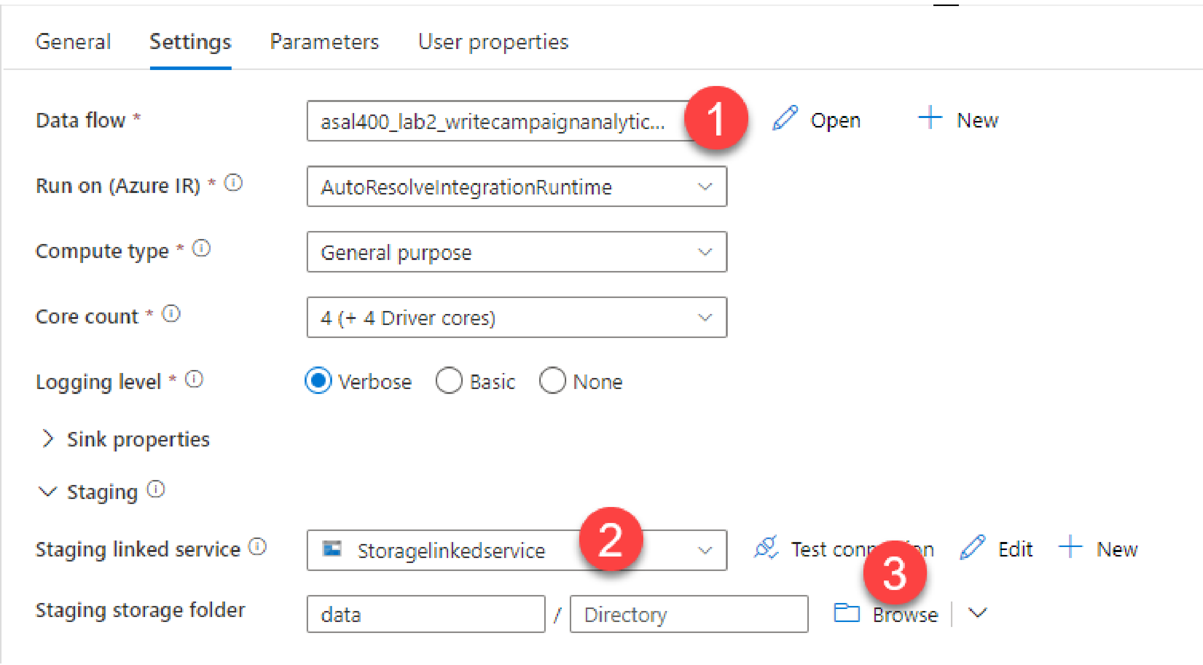
5. On the **General** tab for the data flow (beneath the pipeline canvas), set the **Name** to `asal400_lab2_writecampaignanalyticstoasa`.



Name * asal400_lab2_writecampaignanalyticstoasa

Description

6. Select the **Settings** tab; and then, in the **Data flow** list, select **asal400_lab2_writecampaignanalyticstoasa** . Under staging linked services select Storage account, select browse and select a container for staging



Data flow * asal400_lab2_writecampaignanalytic...

Run on (Azure IR) * AutoResolveIntegrationRuntime

Compute type * General purpose

Core count * 4 (+ 4 Driver cores)

Logging level * ● Verbose ○ Basic ○ None

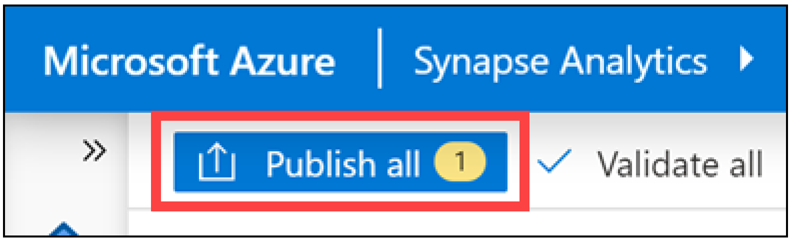
› Sink properties

⌄ Staging

Staging linked service Storagelinkedservice

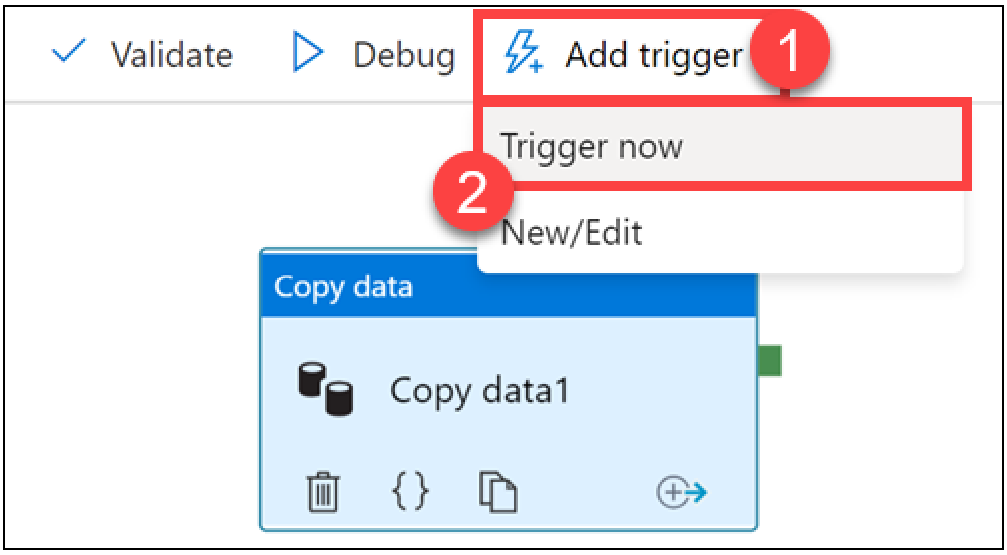
Staging storage folder data / Directory

7. Select **Publish all** to save your new pipeline, and then select **Publish**.

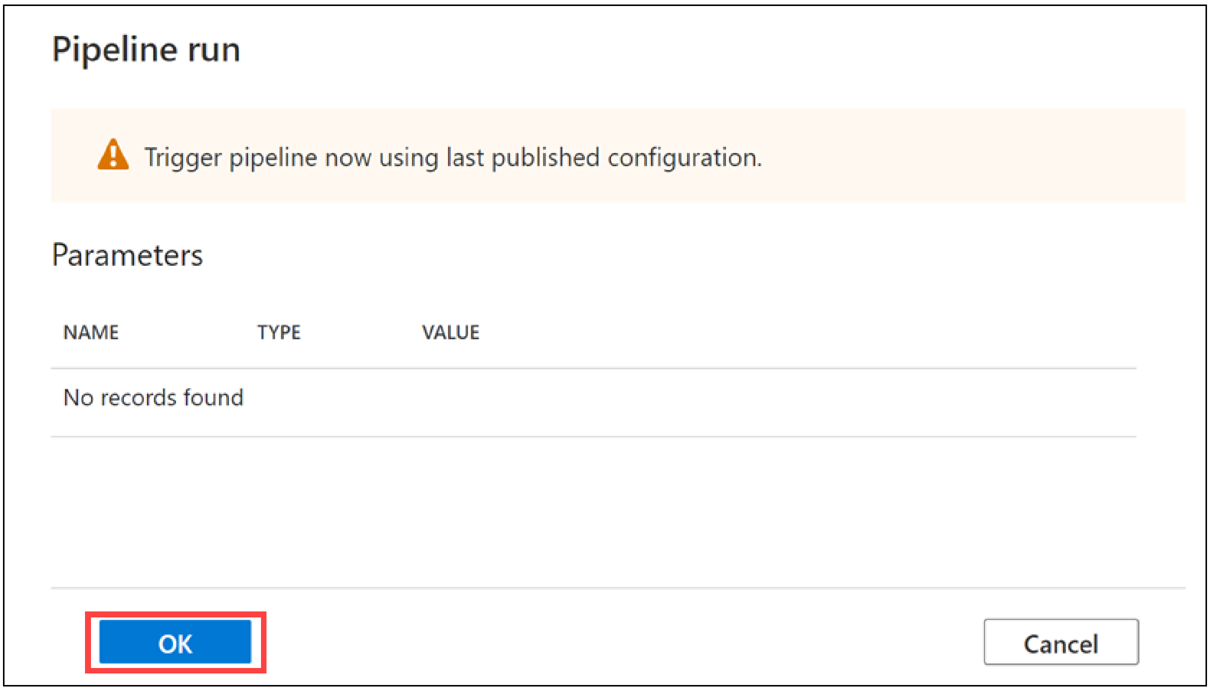


Task 5: Run the campaign analytics data pipeline

1. Select **Add trigger**, and then select **Trigger now** in the toolbar at the top of the pipeline canvas.



2. In the **Pipeline run** pane, select **OK** to start the pipeline run.



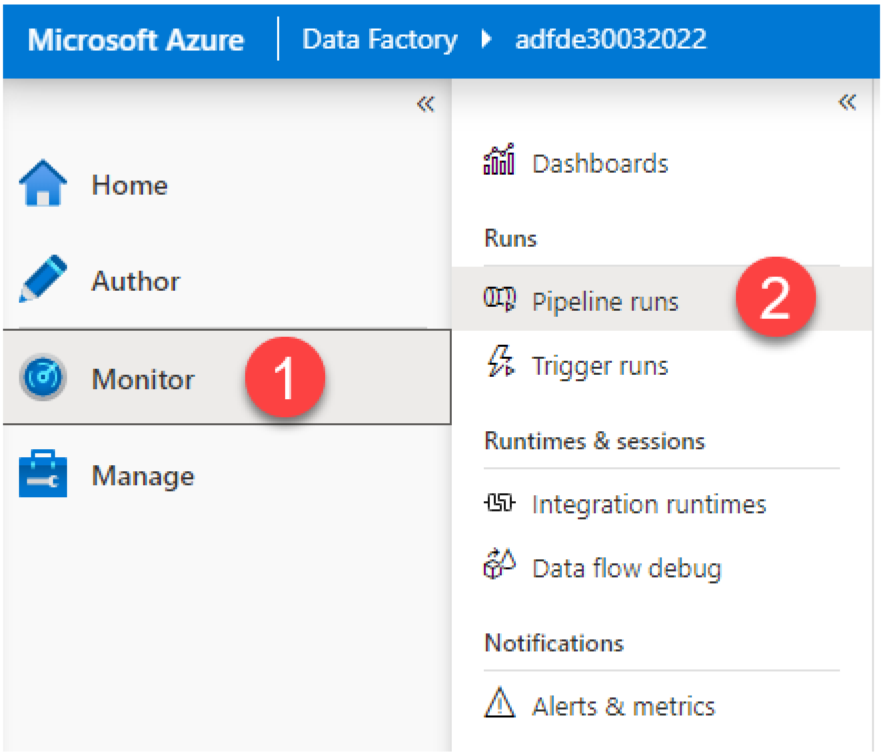
Pipeline run

⚠ Trigger pipeline now using last published configuration.

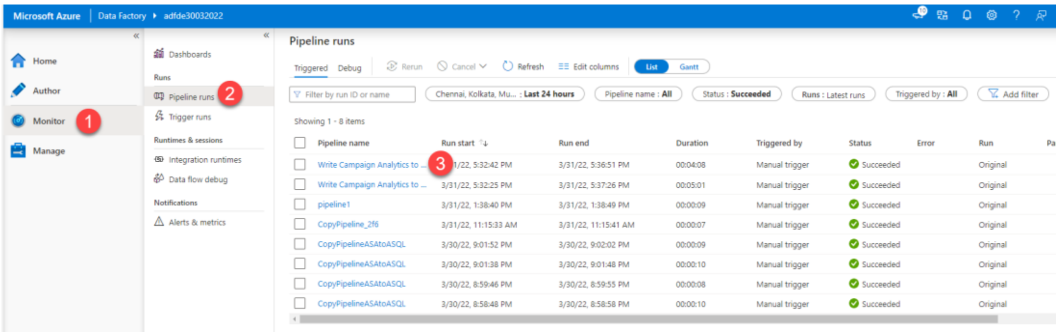
Parameters

NAME	TYPE	VALUE
No records found		

3. Navigate to the **Monitor** hub.



4. Wait for the pipeline run to successfully complete, which will take some time. You may need to refresh the view.



Pipeline runs

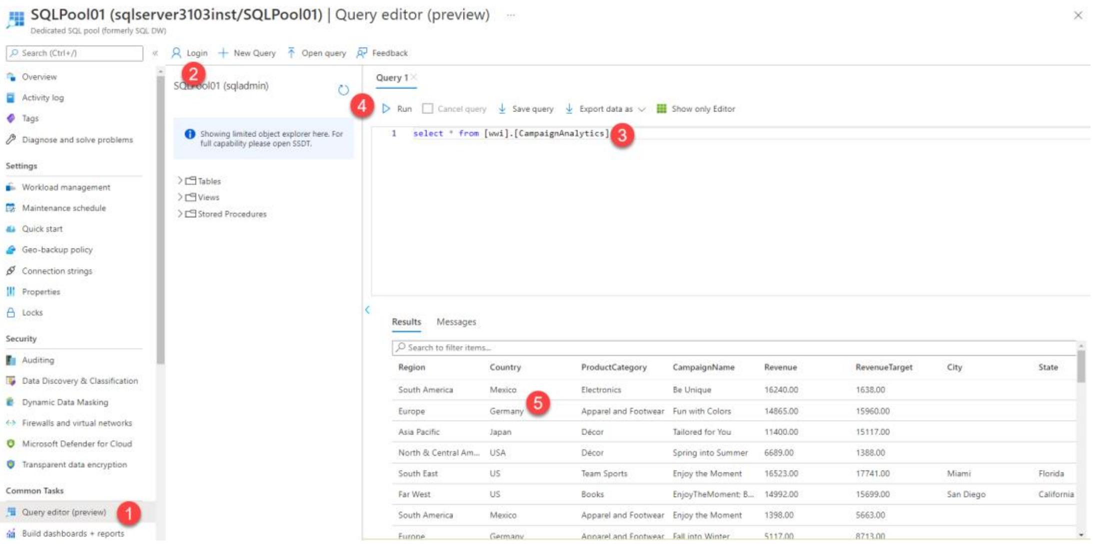
Chennai, Kolkata, Mu... : **Last 24 hours** | Pipeline name : **All** | Status : **Succeeded** | Runs : Latest runs | Triggered by : **All**

Showing 1 - 8 items

	Pipeline name	Run start	Run end	Duration	Triggered by	Status	Error	Run	Pa
□	Write Campaign Analytics to ...	1/22, 5:32:42 PM	3/31/22, 5:36:51 PM	00:04:08	Manual trigger	✓ Succeeded		Original	
□	Write Campaign Analytics to ...	3/31/22, 5:32:25 PM	3/31/22, 5:37:26 PM	00:05:01	Manual trigger	✓ Succeeded		Original	
□	pipeline1	3/31/22, 1:38:40 PM	3/31/22, 1:38:49 PM	00:00:09	Manual trigger	✓ Succeeded		Original	
□	CopyPipeline_2f6	3/31/22, 11:15:33 AM	3/31/22, 11:15:41 AM	00:00:07	Manual trigger	✓ Succeeded		Original	
□	CopyPipelineASAtoASQL	3/30/22, 9:01:52 PM	3/30/22, 9:02:02 PM	00:00:09	Manual trigger	✓ Succeeded		Original	
□	CopyPipelineASAtoASQL	3/30/22, 9:01:38 PM	3/30/22, 9:01:48 PM	00:00:10	Manual trigger	✓ Succeeded		Original	
□	CopyPipelineASAtoASQL	3/30/22, 8:59:46 PM	3/30/22, 8:59:55 PM	00:00:08	Manual trigger	✓ Succeeded		Original	
□	CopyPipelineASAtoASQL	3/30/22, 8:58:48 PM	3/30/22, 8:58:58 PM	00:00:10	Manual trigger	✓ Succeeded		Original	

Task 6: View campaign analytics table contents

Now that the pipeline run is complete, let's take a look at the SQL table to verify the data successfully copied.



```
select * from [wwi].[CampaignAnalytics]
```

Region	Country	ProductCategory	CampaignName	Revenue	RevenueTarget	City	State
South America	Mexico	Electronics	Be Unique	16240.00	1638.00		
Europe	Germany	Apparel and Footwear	Fun with Colors	14865.00	15960.00		
Asia Pacific	Japan	Décor	Tailored for You	11400.00	15117.00		
North & Central Am...	USA	Décor	Spring into Summer	6689.00	1388.00		
South East	US	Team Sports	Enjoy the Moment	16523.00	17741.00	Miami	Florida
Far West	US	Books	EnjoyTheMoment; B...	14992.00	15699.00	San Diego	California
South America	Mexico	Apparel and Footwear	Enjoy the Moment	1398.00	5663.00		
Europe	Germany	Apparel and Footwear	Fall into Winter	5117.00	8713.00		