

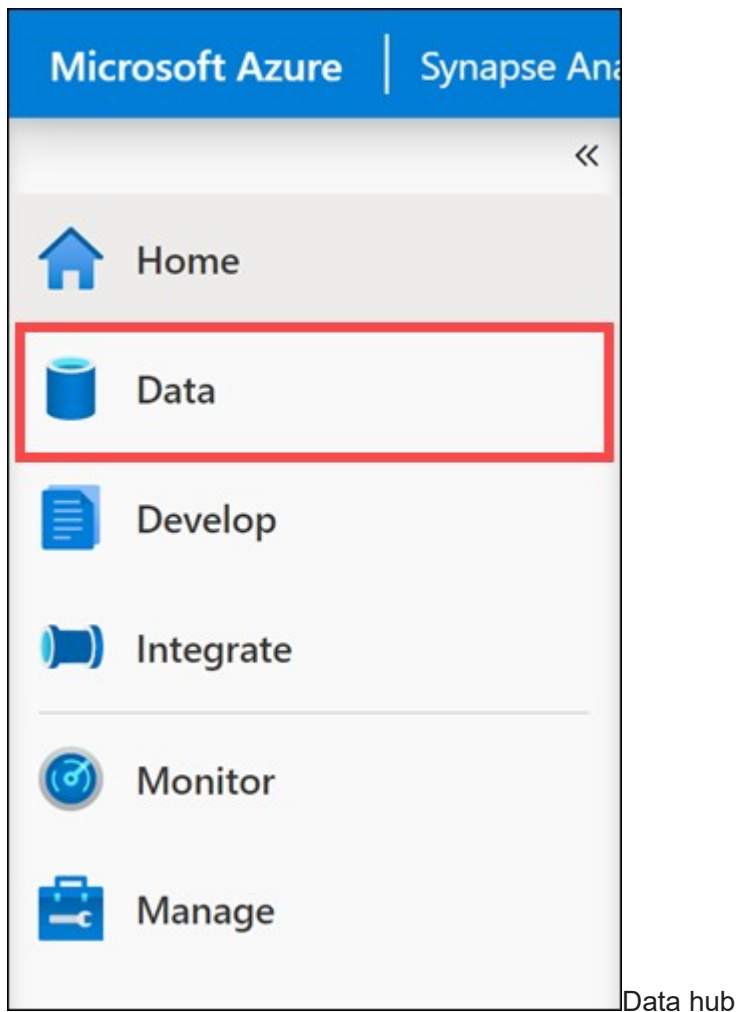
Exercise - Design and implement a Type 1 slowly changing dimension with mapping data flows

In this exercise, you create a Data flow for a Type 1 SCD using Azure Synapse dedicated SQL pool as the source and destination. This data flow could then be added to a Synapse Pipeline and run as part of the extract, transform, load (ETL) process.

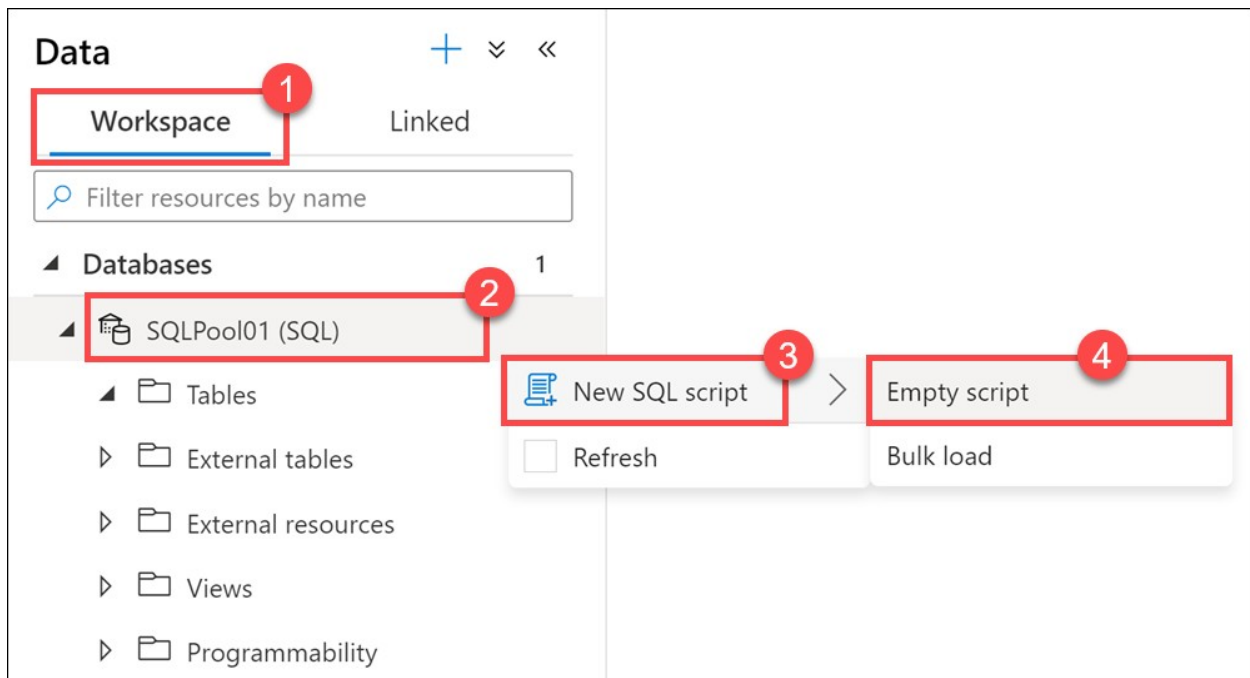
Setup source and dimension table

For this exercise you want to load a dimension table in Azure Synapse from source data that could be from many different system types, such as Azure SQL, Azure storage, etc. For this example you keep it simple by creating the source data in your Azure Synapse database.

1. From Synapse Studio, navigate to the **Data** hub.



2. Select the **Workspace** tab (1), expand **Databases**, then right-click on **SQLPool01** (2). Select **New SQL script** (3), then select **Empty script** (4).



The data hub is displayed with the context menus to create a new SQL script

3. Paste the following script into the empty script window, then select **Run or hit **F5** to execute the query:**

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

```
CREATE TABLE [dbo].[CustomerSource] (  
    [CustomerID] [int] NOT NULL,  
    [Title] [nvarchar](8),  
    [FirstName] [nvarchar](50),  
    [MiddleName] [nvarchar](50),  
    [LastName] [nvarchar](50),  
    [Suffix] [nvarchar](10),  
    [CompanyName] [nvarchar](128),  
    [SalesPerson] [nvarchar](256),  
    [EmailAddress] [nvarchar](50),  
    [Phone] [nvarchar](25)  
) WITH ( HEAP )
```

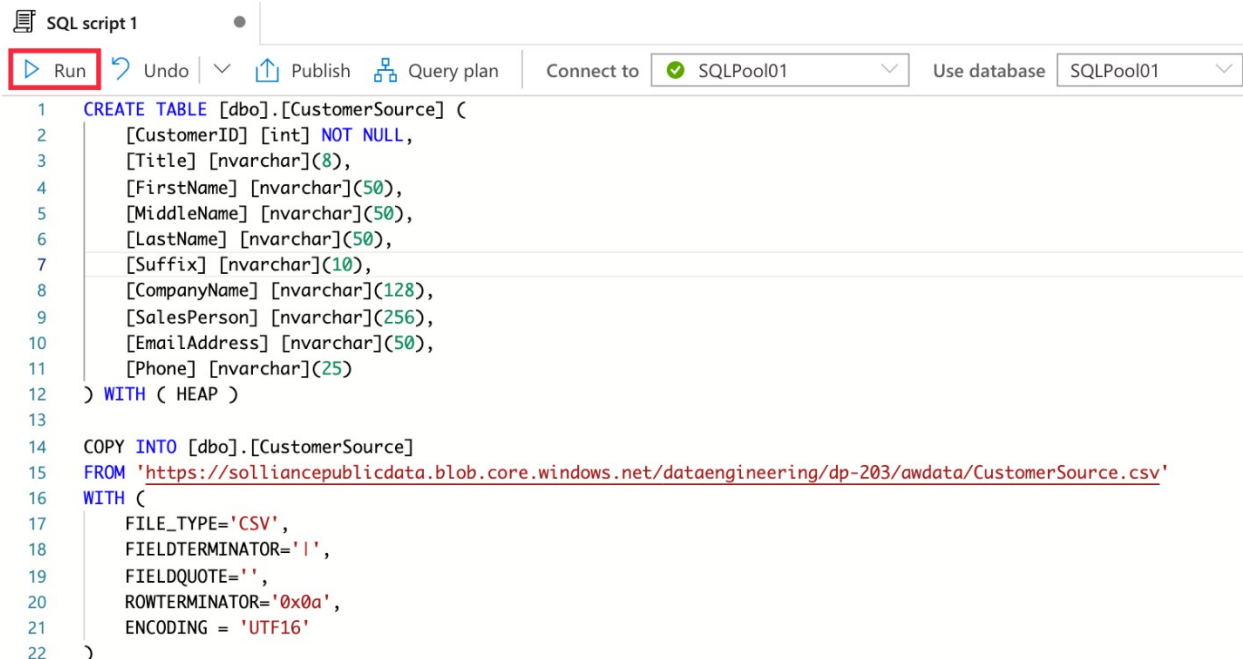
```
COPY INTO [dbo].[CustomerSource]  
FROM 'https://solliancepublicdata.blob.core.windows.net/dataengineering/dp-203/  
awdata/CustomerSource.csv'  
WITH (  
    FILE_TYPE='CSV',  
    FIELDTERMINATOR='|',  
    FIELDQUOTE='',  
    ROWTERMINATOR='0x0a',  
    ENCODING = 'UTF16'  
)
```

```
CREATE TABLE dbo.[DimCustomer](  
    [CustomerID] [int] NOT NULL,  
    [Title] [nvarchar](8) NULL,
```

```

[FirstName] [nvarchar](50) NOT NULL,
[MiddleName] [nvarchar](50) NULL,
[LastName] [nvarchar](50) NOT NULL,
[Suffix] [nvarchar](10) NULL,
[CompanyName] [nvarchar](128) NULL,
[SalesPerson] [nvarchar](256) NULL,
[EmailAddress] [nvarchar](50) NULL,
[Phone] [nvarchar](25) NULL,
[InsertedDate] [datetime] NOT NULL,
[ModifiedDate] [datetime] NOT NULL,
[HashKey] [char](66)
)
WITH
(

```



```

SQL script 1
▶ Run Undo ▾ ⬆ Publish ⚙ Query plan Connect to SQLPool01 Use database SQLPool01
1 CREATE TABLE [dbo].[CustomerSource] (
2     [CustomerID] [int] NOT NULL,
3     [Title] [nvarchar](8),
4     [FirstName] [nvarchar](50),
5     [MiddleName] [nvarchar](50),
6     [LastName] [nvarchar](50),
7     [Suffix] [nvarchar](10),
8     [CompanyName] [nvarchar](128),
9     [SalesPerson] [nvarchar](256),
10    [EmailAddress] [nvarchar](50),
11    [Phone] [nvarchar](25)
12 ) WITH ( HEAP )
13
14 COPY INTO [dbo].[CustomerSource]
15 FROM 'https://soiliencepublicdata.blob.core.windows.net/dataengineering/dp-203/awdata/CustomerSource.csv'
16 WITH (
17     FILE_TYPE='CSV',
18     FIELDTERMINATOR='|',
19     FIELDQUOTE='',
20     ROWTERMINATOR='0x0a',
21     ENCODING = 'UTF16'
22 )

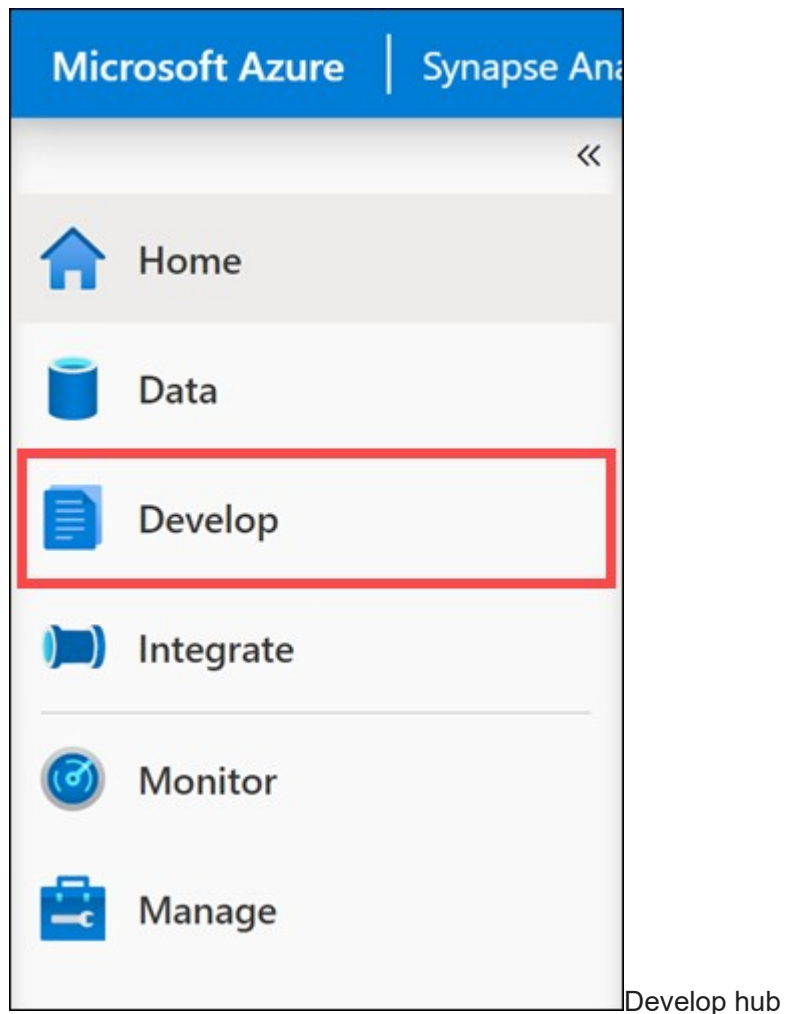
```

The script and Run button are both highlighted

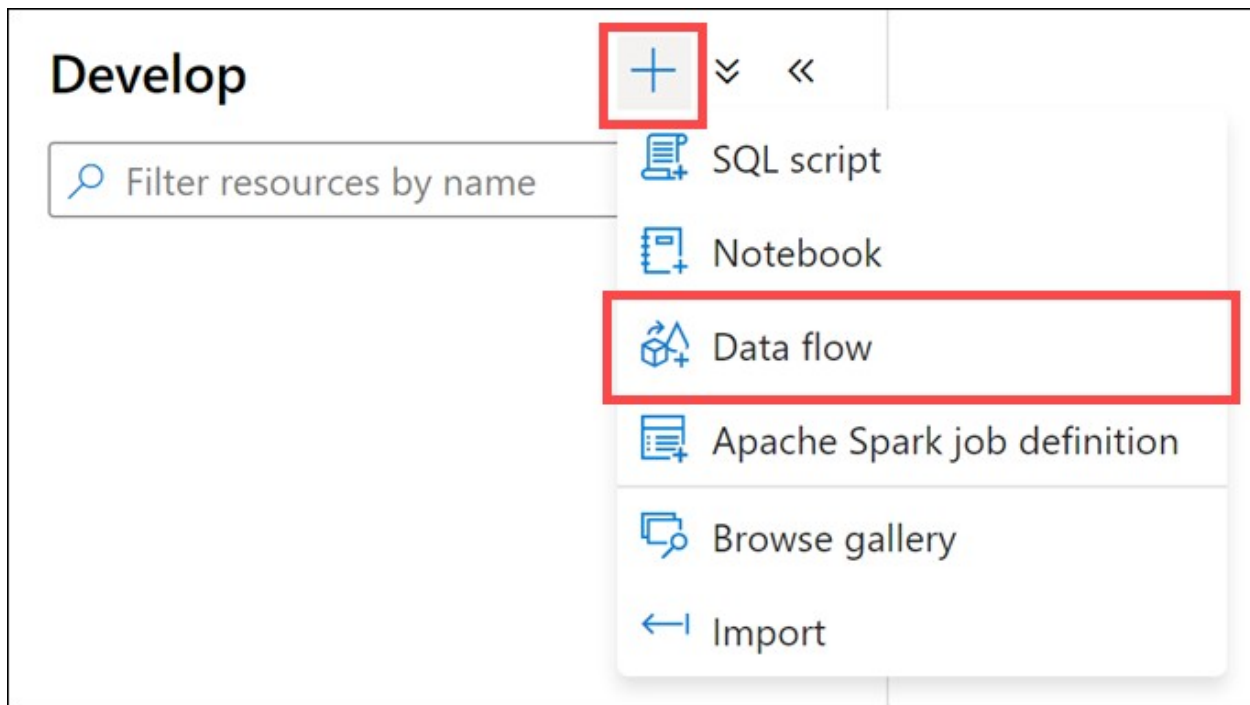
Create a mapping data flow

Mapping Data flows are pipeline activities that provide a visual way of specifying how to transform data, through a code-free experience. Next you will create a mapping data flow to create a Type 1 SCD.

1. Navigate to the Develop hub.

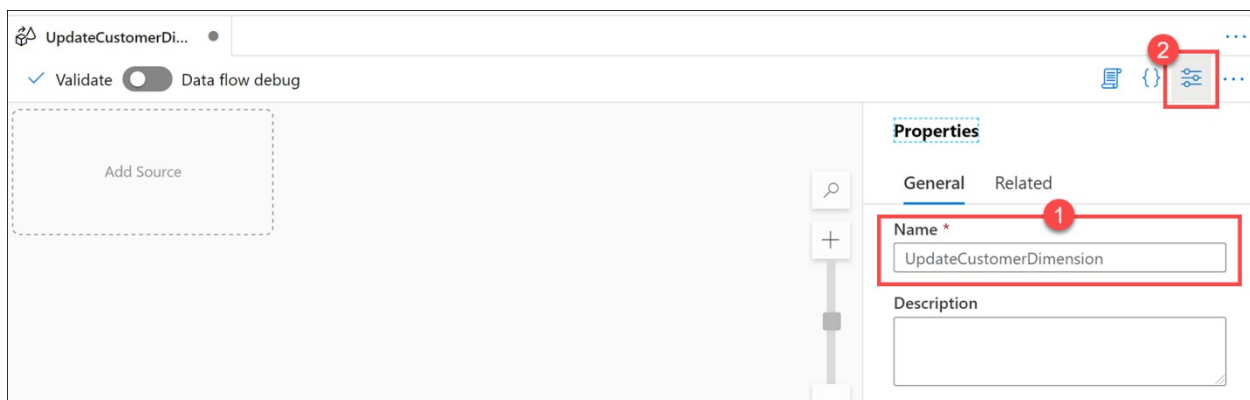


2. Select +, then select **Data flow**.



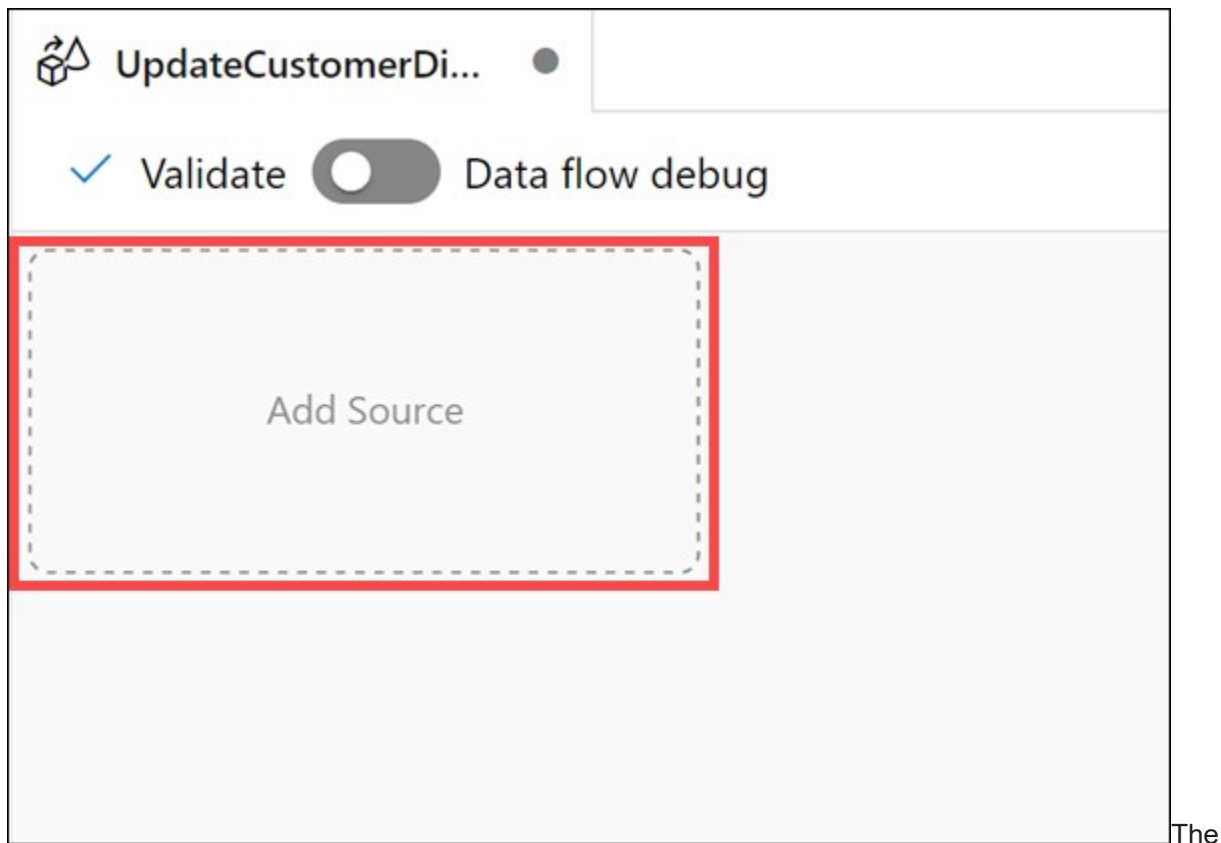
The plus button and data flow menu item are highlighted

3. In the properties pane of the new data flow, enter `UpdateCustomerDimension` in the Name field (1), then select the Properties button (2) to hide the properties pane.



The data flow properties pane is displayed

4. Select Add Source on the canvas.



Add Source button is highlighted on the data flow canvas

5. Under Source settings, configure the following properties:

- Output stream name: Enter **SourceDB**
- Source type: Select **Dataset**
- Options: Check **Allow schema drift** and leave the other options unchecked
- Sampling: Select **Disable**
- Dataset: Select **+ New** to create a new dataset

Source settings

Source options

Projection

Optimize

Inspect

Data preview

Output stream name *

SourceDB

[Learn more](#)

Source type *

Dataset

Dataset *

Select...

+ New

Options

☒ Allow schema drift ⓘ

☐ Infer drifted column types ⓘ

☐ Validate schema ⓘ

Sampling * ⓘ


☐ Enable

☒ Disable

The New button is highlighted next to Dataset










6. In the new integration dataset dialog, select Azure Synapse Analytics, then select Continue.

New integration dataset

In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. [Learn more](#) 

Select a data store

All Azure Database File Generic protocol NoSQL Services and apps

 Azure Data Lake Storage Gen2	 Azure Database for PostgreSQL	 Azure SQL Database
 Azure SQL Database Managed Instance	 Azure Synapse Analytics	 Snowflake
 Amazon Marketplace Web Service	 Amazon Redshift	 Amazon S3

Continue

Cancel

Azure Synapse Analytics and the Continue button are highlighted

7. In the dataset properties, configure the following:

- **Name:** Enter `CustomerSource`
- **Linked service:** Select the Synapse workspace linked service
- **Table name:** Select the **Refresh button** next to the dropdown

8. In the Value field, enter your SQL Pool name, then select OK.

Please provide actual value of the parameters to list tables

Parameters for linked service asagaworkspacedv031721-WorkspaceDefaultSqlServer

Name	Type	Value
DBName	String	SQLPool01

OK

Cancel

The SQLPool01 parameter is highlighted

9. Select `dbo.CustomerSource` under **Table name, select **From connection/store** under **Import schema**, then select **OK** to create the dataset.**

Set properties

Name

CustomerSource

Linked service *

asagaworkspacedv031721-WorkspaceDefaultSqlServer

Connect via integration runtime * ⓘ

AutoResolveIntegrationRuntime

Table name

dbo.CustomerSource

☐ Edit

Import schema

☒ From connection/store ☐ None

▶ Advanced

OK

Back

Cancel

The form is completed as described

10 Select Open next to the CustomerSource dataset that you added.

Source settings

Source options

Projection

Optimize

Inspect

Data preview

Output stream name *

SourceDB

[Learn more](#)

Source type *



Dataset



Inline

Dataset *

CustomerSource



Test connection



Open



New

Options

☒ Allow schema drift ⓘ

☐ Infer drifted column types ⓘ

☐ Validate schema ⓘ

Sampling * ⓘ

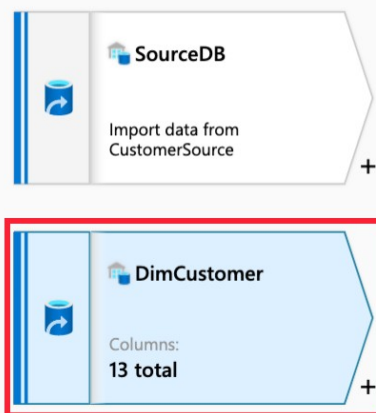
☐ Enable ☒ Disable

The open button is highlighted next to the new dataset

11. Enter your **SQL Pool** name in the **Value** field next to **DBName**.

12. In the data flow editor, select the **Add Source** box below the **SourceDB** activity. Configure this source as the **DimCustomer** table following the same steps used for **CustomerSource**.


- **Output stream name:** Enter DimCustomer
- **Source type:** Select **Dataset**
- **Options:** Check **Allow schema drift** and leave the other options unchecked
- **Sampling:** Select **Disable**
- **Dataset:** Select **+ New** to create a new dataset. Use the Azure Synapse linked service and choose DimCustomer table. Be sure to set the DBName to your SQL Pool name.




Source settings Source options Projection Optimize Inspect Data preview


Output stream name * [Learn more](#)


Source type *

 Dataset

 Inline

Dataset *

 DimCustomer



[Test connection](#) [Open](#) [+ New](#)

Options

☒ Allow schema drift ⓘ

☐ Infer drifted column types ⓘ

☐ Validate schema ⓘ

Sampling * ⓘ

☐ Enable

☒ Disable

The Add Source, Output stream name, and Dataset name are highlighted in the Source settings

Add transformations to data flow

1. Select + to the right of the SourceDB source on the canvas, then select Derived Column.

2. Under Derived column's settings, configure the following properties:


The screenshot shows a data integration tool interface. On the left, there are two source components: 'SourceDB' (Columns: 10 total) and 'DimCustomer' (Import data from DimCustomer). Below these is a settings panel with tabs for 'Source settings' and 'Source options'. The 'Source settings' tab is active, showing fields for 'Output stream name *', 'Source type *', 'Dataset *', 'Options', and 'Sampling *'. A plus button is visible next to the 'Source type' field. A context menu is open, displaying options under 'Multiple inputs/outputs' (Join, Conditional Split, Exists, Union, Lookup) and 'Schema modifier' (Derived Column, Select, Aggregate, Surrogate Key, Pivot, Unpivot, Window). The 'Derived Column' option is highlighted with a red rectangle.


SourceDB
Columns:
10 total

DimCustomer
Import data from
DimCustomer

Source settings Source options

Output stream name * Source

Source type *  Da

Dataset *  Cus

Options ☒ Allow ☐ Infer ☐ Valid ☐ Enab

Sampling * ⓘ

Search

Multiple inputs/outputs

- Join
- Conditional Split
- Exists
- Union
- Lookup

Schema modifier

- Derived Column**
- Select
- Aggregate
- Surrogate Key
- Pivot
- Unpivot
- Window

The plus button and derived column menu item are highlighted

- **Output stream name:** Enter `CreateCustomerHash`

- **Incoming stream:** Select **SourceDB**
- **Columns:** Enter the following:

Column	Expression	Description
Select InsertedDate	<code>iif(isNull(InsertedDate), currentTimestamp(), {InsertedDate})</code>	If the InsertedDate value is null, insert the current timestamp. Otherwise, use the InsertedDate value.
Select ModifiedDate	<code>currentTimestamp()</code>	Always update the ModifiedDate value with the current timestamp.

Derived column's settings Optimize Inspect Data preview ● [Description](#) ^

Output stream name * [Learn more](#)

Incoming stream * ▼

+ Add Clone Delete Open expression builder

Columns * ⓘ

<input type="checkbox"/>	Column	Expression
<input type="checkbox"/>	HashKey	sha2(256, Title+FirstName+MiddleName+LastNa... abc +

The Derived column's settings form is configured as described.

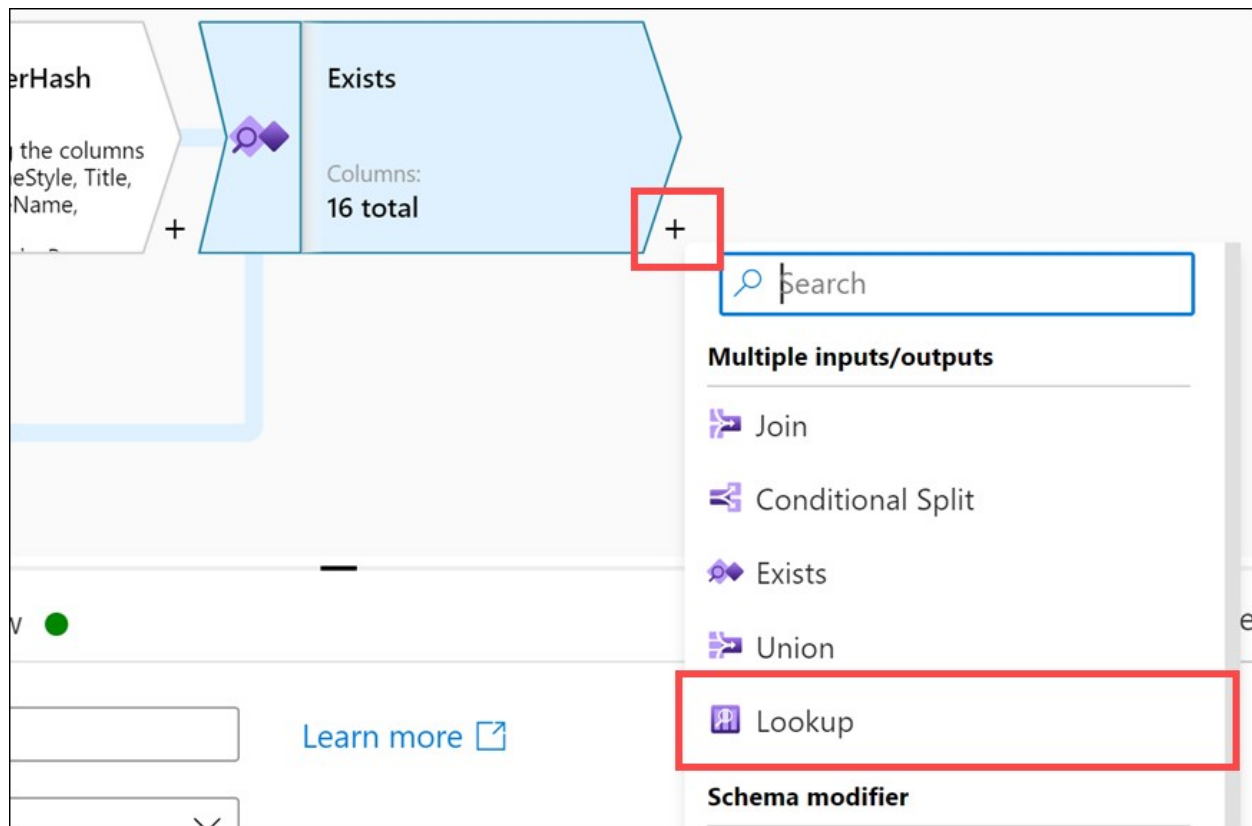
3. Select **+** to the right of the **CreateCustomerHash** derived column on the canvas, then select **Exists**.

4. Under **Exists** settings, configure the following properties:

- **Output stream name:** Enter **Exists**
- **Left stream:** Select **CreateCustomerHash**
- **Right stream:** Select **SynapseDimCustomer**
- **Exist type:** Select **Doesn't exist**
- **Exists conditions:** Set the following for Left and Right:

Left: CreateCustomerHash's column	Right: SynapseDimCustomer's column
HashKey	HashKey

5. Select **+** to the right of **Exists** on the canvas, then select **Lookup**.



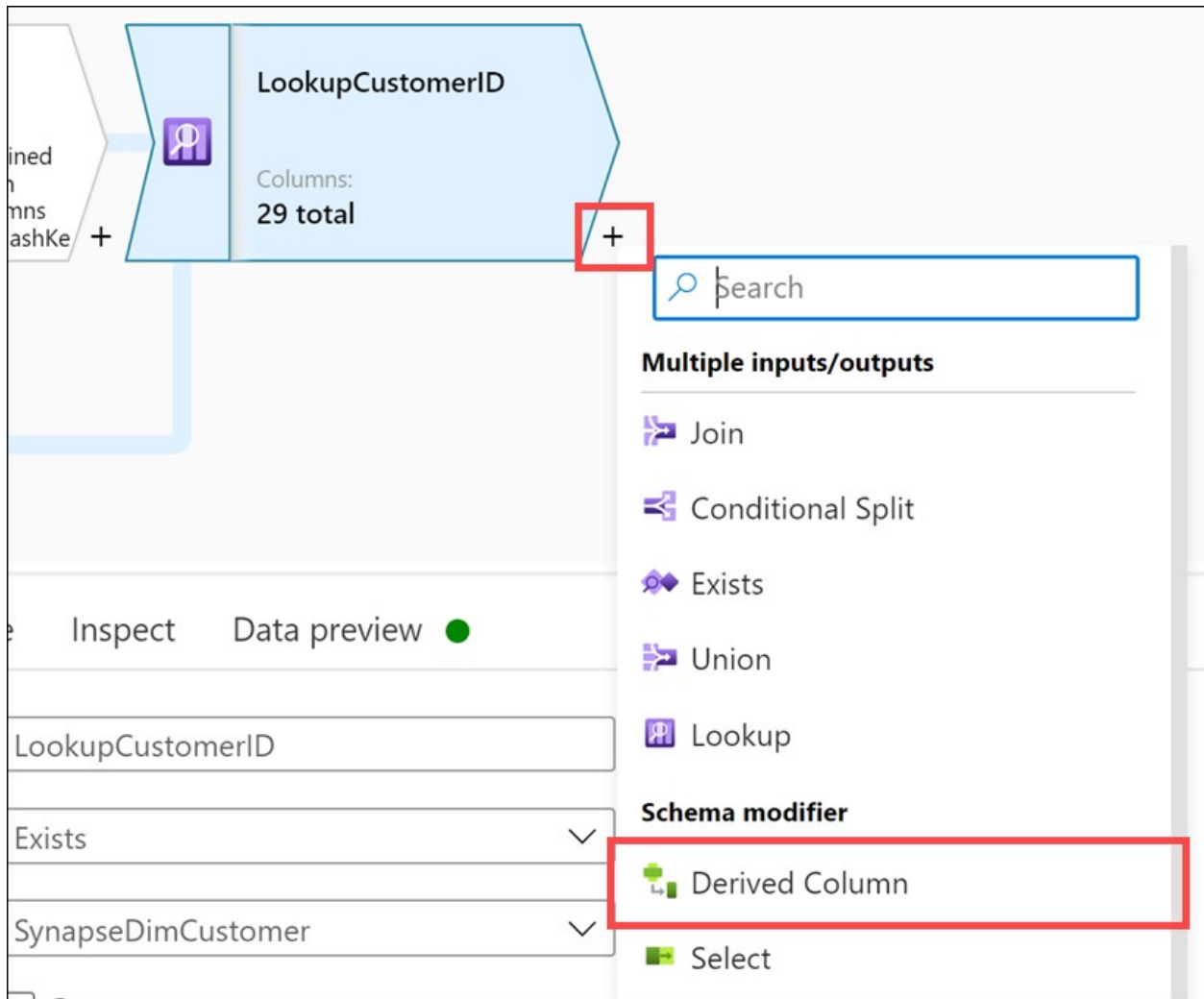
The plus button and lookup menu item are both highlighted

6. Under Lookup settings, configure the following properties:

- **Output stream name:** Enter `LookupCustomerID`
- **Primary stream:** Select **Exists**
- **Lookup stream:** Select **SynapseDimCustomer**
- **Match multiple rows:** Unchecked
- **Match on:** Select **Any row**
- **Lookup conditions:** Set the following for Left and Right:

Left: Exists's column	Right: SynapseDimCustomer's column
CustomerID	CustomerID

7. Select + to the right of `LookupCustomerID` on the canvas, then select **Derived Column**.



The plus button and derived column menu item are both highlighted.

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

- **Output stream name:** Enter **SetDates**
- **Incoming stream:** Select **LookupCustomerID**
- **Columns:** Enter the following:

Column	Expression	Description
Select InsertedDate	<code>iif(isNull(InsertedDate), currentTimestamp(), {InsertedDate})</code>	If the InsertedDate value is null, insert the current timestamp. Otherwise, use the InsertedDate value.
Select ModifiedDate	<code>currentTimestamp()</code>	Always update the ModifiedDate value with the current timestamp.

Derived column's settings | Optimize | Inspect | Data preview ● | Description

Output stream name * [Learn more](#)

Incoming stream *

+ Add Clone Delete Open expression builder

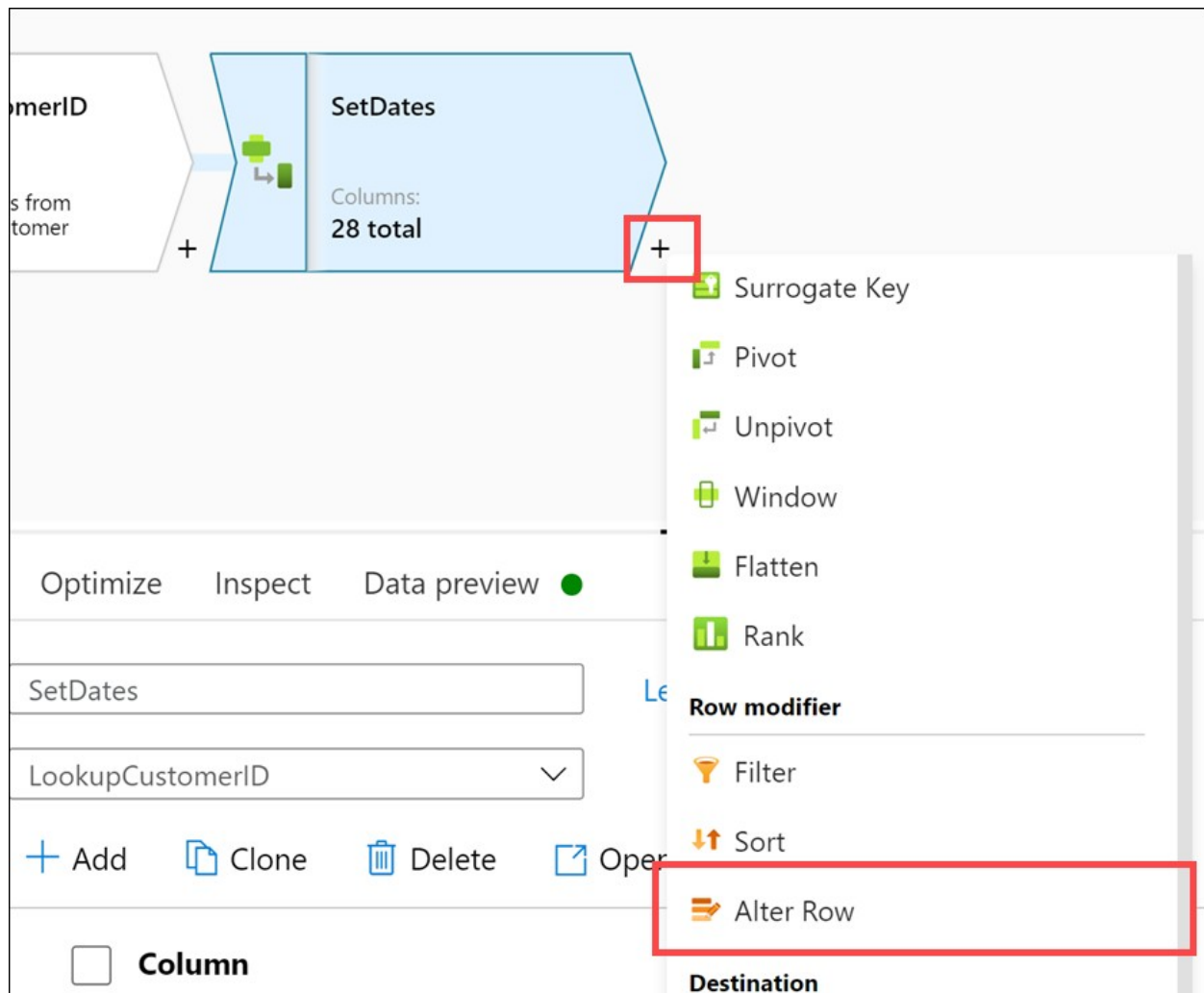
Columns * ⓘ

<input type="checkbox"/>	Column	Expression	
<input type="checkbox"/>	InsertedDate	<code>iif(isNull(InsertedDate), currentTimestamp(), {Insert...</code>	+
<input type="checkbox"/>	ModifiedDate	<code>currentTimestamp()</code>	+

Another Derived column's settings form is configured as described.

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

9. Select + to the right of the **SetDates** derived column step on the canvas, then select **Alter Row**.



The plus button and alter row menu item are both highlighted.

10 Under **Alter row settings**, configure the following properties:

- **Output stream name:** Enter **AllowUpserts**
- **Incoming stream:** Select **SetDates**
- **Alter row conditions:** Enter the following:

Condition	Expression	Description
Select Upsert if	<code>true()</code>	Set the condition to true() on the Upsert if condition to allow upserts. This ensures that all data that passes through the steps in the mapping data flow will be inserted or updated into the sink.

Alter row settings Optimize Inspect Data preview ● Description ^

Output stream name * [Learn more](#)

Incoming stream *

Alter row conditions * ⓘ

✦ Upsert if

The alter row settings form is configured as described.

11. Select + to the right of the `AllowUpserts` alter row step on the canvas, then select `Sink`.

The screenshot shows a data pipeline canvas. On the left, a stream labeled 'the columns Style, Title, ame,' is connected to a blue trapezoidal step named 'AllowUpserts'. The step has a small icon of three stacked books and text indicating 'Columns: 28 total'. To the right of the 'AllowUpserts' step is a red-outlined square containing a white plus sign '+'. A dropdown menu is open to the right of this plus sign, listing various data transformation steps. The 'Sink' option at the bottom of the menu is highlighted with a red rectangle. The menu items are: Surrogate Key, Pivot, Unpivot, Window, Flatten, Rank, Row modifier (a section header), Filter, Sort, Alter Row, and Destination (a section header). The 'Sink' option is located under the 'Destination' section.

The plus button and sink menu item are both highlighted

12. Under sink, configure the following properties:

- Output stream name: Enter **Sink**
- Incoming stream: Select **AllowUpserts**
- Sink type: Select **Dataset**
- Dataset: Select **DimCustomer**
- Options: Check **Allow schema drift** and uncheck **Validate schema**

Sink Settings Mapping Optimize Inspect Data preview ●

Output stream name * [Learn more](#)

Incoming stream *

Sink type *

Dataset * [Test connection](#) [Open](#) [New](#)

Options ☒ Allow schema drift [?](#) ☐ Validate schema [?](#)

The sink properties form is configured as described

13. Select the Settings tab and configure the following properties:

- **Update method:** Check **Allow upsert** and uncheck all other options
- **Key columns:** Select **List of columns**, then select **CustomerID** in the list
- **Table action:** Select **None**
- **Enable staging:** Unchecked

Sink **Settings** Mapping Optimize Inspect Data preview ●

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

Update method ☐ Allow insert [Add dynamic content \[Alt+P\]](#)

☐ Allow delete

☒ Allow upsert

☐ Allow update

Key columns * **i** ☒ List of columns ☐ Custom expression **i**

123 CustomerID **v** [Add dynamic content \[Alt+P\]](#) + **i**

Skip writing key columns ☐

Table action ☒ None ☐ Recreate table ☐ Truncate table

Enable staging ☐

Batch size **i**

The sink settings are configured as described

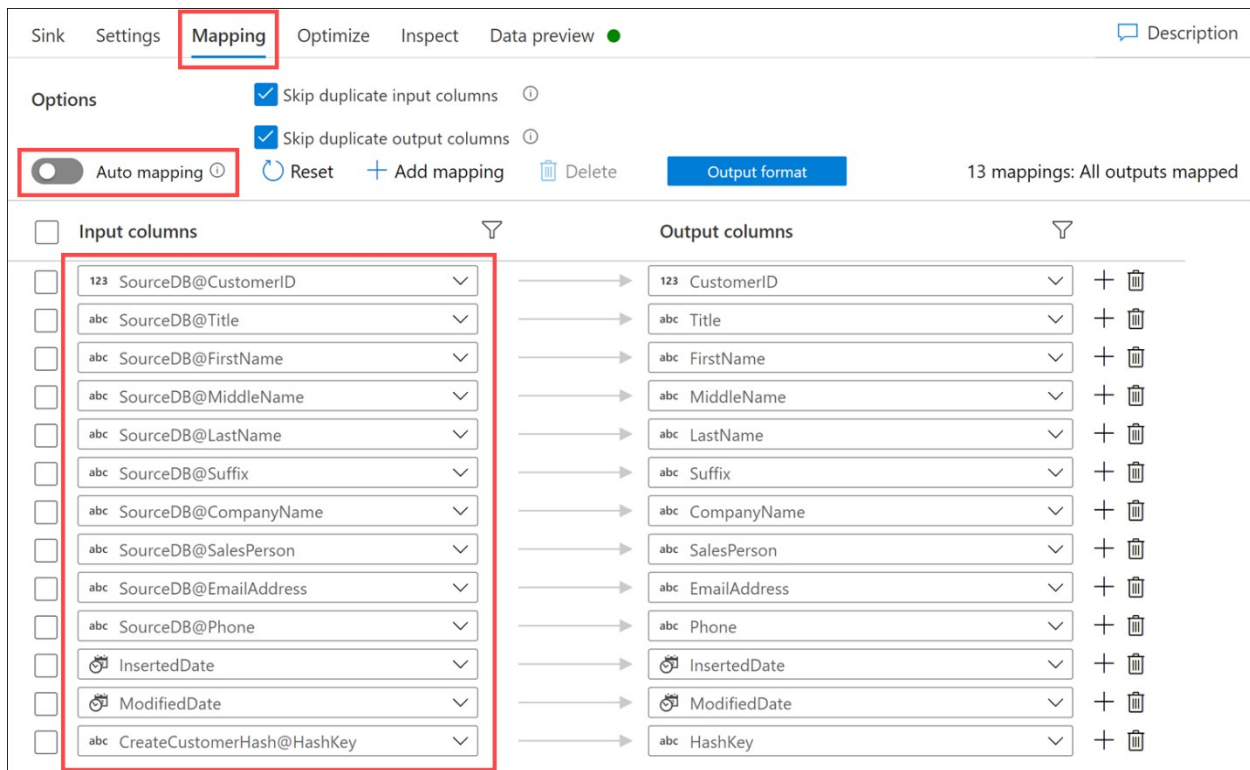
14. Select the Mapping tab, then uncheck Auto mapping. Configure the input columns mapping as outlined below:

Input columns

SourceDB@CustomerID
 SourceDB@Title
 SourceDB@FirstName
 SourceDB@MiddleName
 SourceDB@LastName
 SourceDB@Suffix
 SourceDB@CompanyName
 SourceDB@SalesPerson
 SourceDB@EmailAddress
 SourceDB@Phone
 InsertedDate
 ModifiedDate
 CreateCustomerHash@HashKey

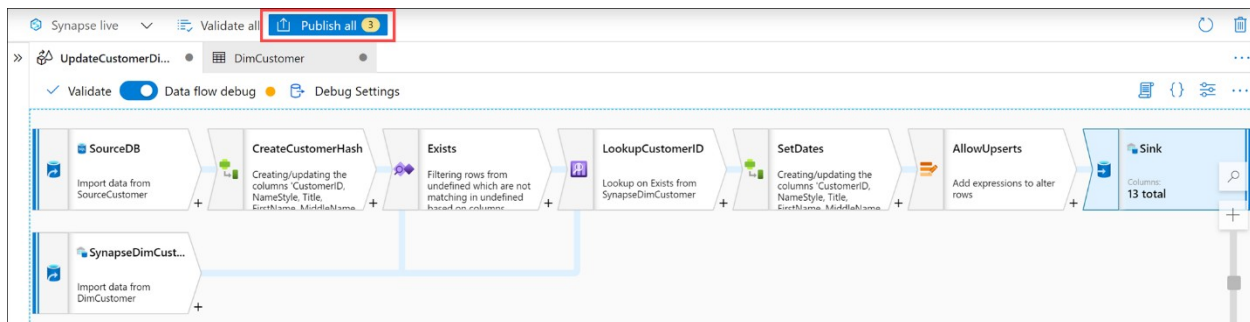
Output columns

CustomerID
 Title
 FirstName
 MiddleName
 LastName
 Suffix
 CompanyName
 SalesPerson
 EmailAddress
 Phone
 InsertedDate
 ModifiedDate
 HashKey



Mapping settings are configured as described

15. The completed mapping flow should look like the following. Select **Publish all to save your changes.**






The completed data flow is displayed and Publish all is highlighted.

16. Select **Publish.**

Publish all

You are about to publish all pending changes to the live environment. [Learn more](#)

Pending changes (3)

NAME	CHANGE	EXISTING
▲ Datasets		
 SourceCustomer	(New)	-
 DimCustomer	(New)	-
▲ Data flows		
 UpdateCustomerDimension	(New)	-

Publish

Cancel

The publish button is highlighted

How to test the data flow

You have completed a Type 1 SCD data flow. If you choose to test it out you could add this data flow to a Synapse integration pipeline. Then you could run the pipeline once to do the initial load of the customer source data to the DimCustomer destination.

Each additional run of the pipeline will compare the data in the source table to what is already in the dimension table (using the HashKey) and only update records that have changed. In order to test this, you could update a record in the source table then run the pipeline again and verify the record updates in the dimension table.

Take the customer Janet Gates as an example. The initial load shows the `LastName` is Gates and the `CustomerId` is 4.

Run Undo Publish Query plan Connect to SQLPool01 Use database SQLPool01

```
1 Select CustomerId, FirstName, LastName, ModifiedDate From DimCustomer Where CustomerId = 4
```

Results Messages

View Table Chart Export results

Search

CustomerId	FirstName	LastName	ModifiedDate
4	Janet	Gates	2021-03-27T05:01:17.9170000

The script is displayed with the initial customer record.

Here is an example statement that would update the customer last name in the source table.

```
1 UPDATE [dbo].[CustomerSource]
2 SET LastName = 'Lopez'
3 WHERE [CustomerId] = 4
```

After updating the record and running the pipeline again, DimCustomer would show this updated data.

Run Undo Publish Query plan Connect to SQLPool01 Use database SQLPool01

```
1 Select CustomerId, FirstName, LastName, ModifiedDate From DimCustomer Where CustomerId = 4
```

Results Messages

View Table Chart Export results

Search

CustomerId	FirstName	LastName	ModifiedDate
4	Janet	Lopez	2021-03-27T17:48:23.0270000

The script is displayed with the updated customer record.

The customer record successfully updated the `LastName` value to match the source record and updated the `ModifiedDate`, without keeping track of the old `LastName` value. That is the expected behavior for a Type 1 SCD. If history was required for the `LastName` field then you would modify the table and data flow to be one of the other SCD types you have learned.g

