

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

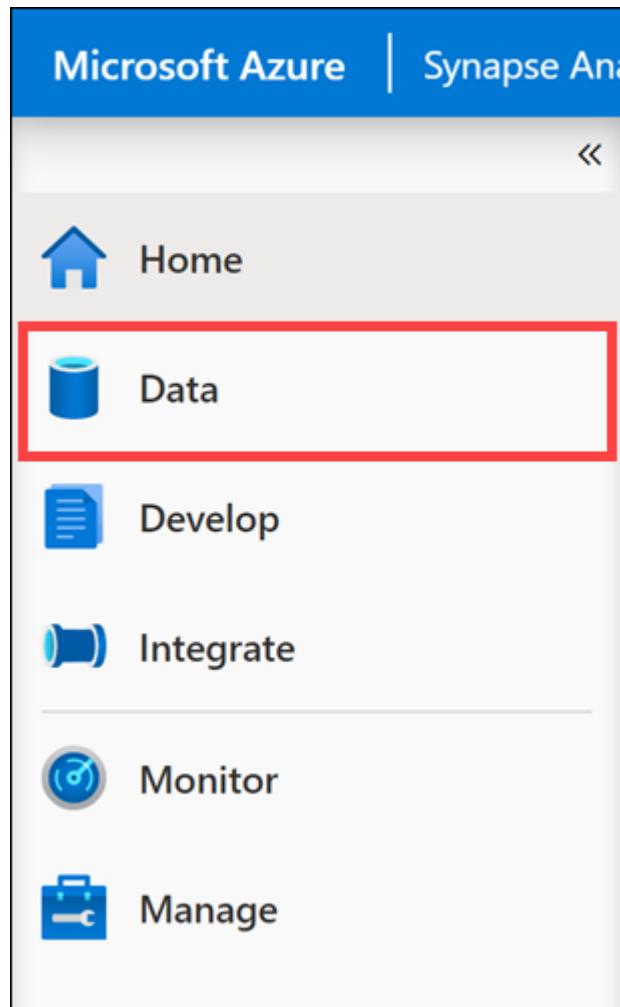
10 minutes

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.

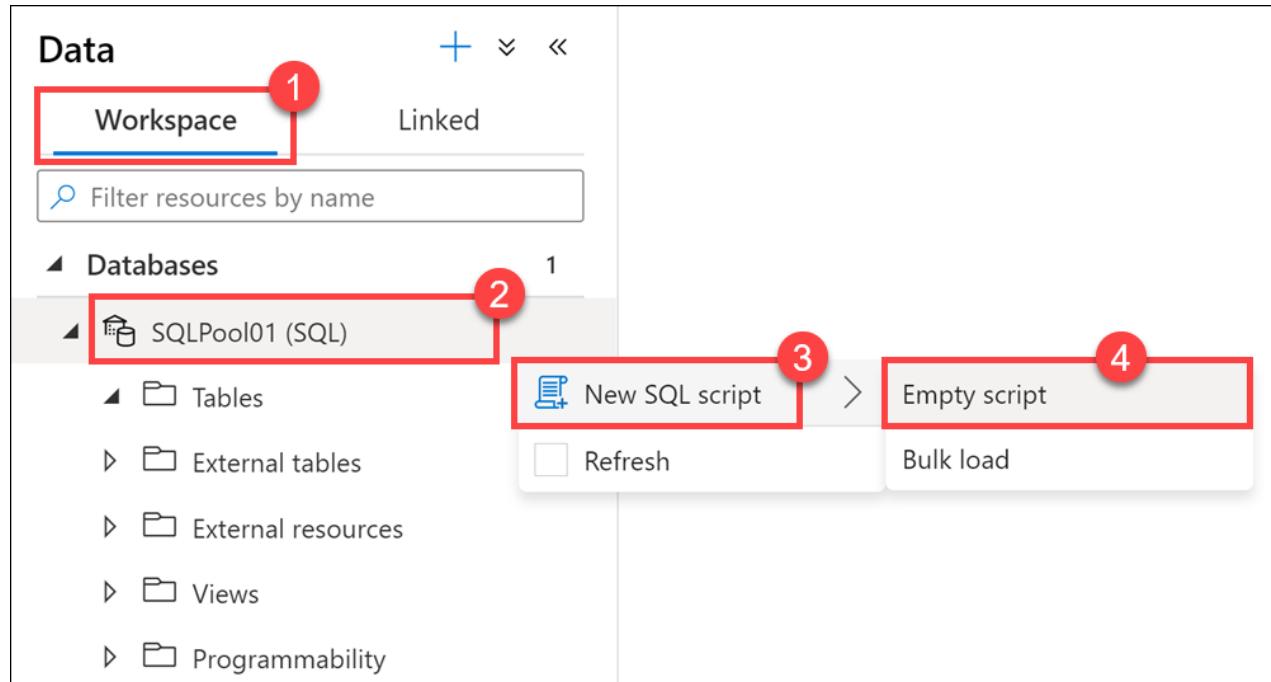
Setup source and dimension table

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).



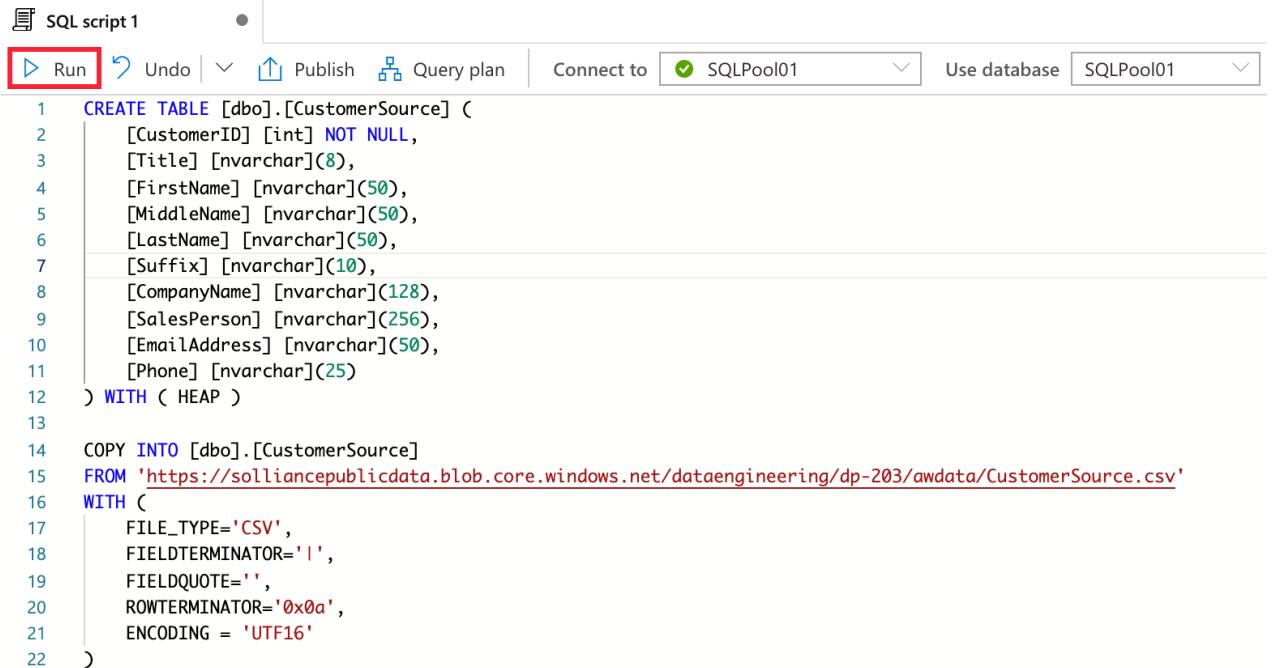
3. Paste the following script into the empty script window, then select **Run** or hit F5 to execute the query:

SQL

```
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](64)
)
WITH
(
    DISTRIBUTION = REPLICATE,
    CLUSTERED COLUMNSTORE INDEX
)
```

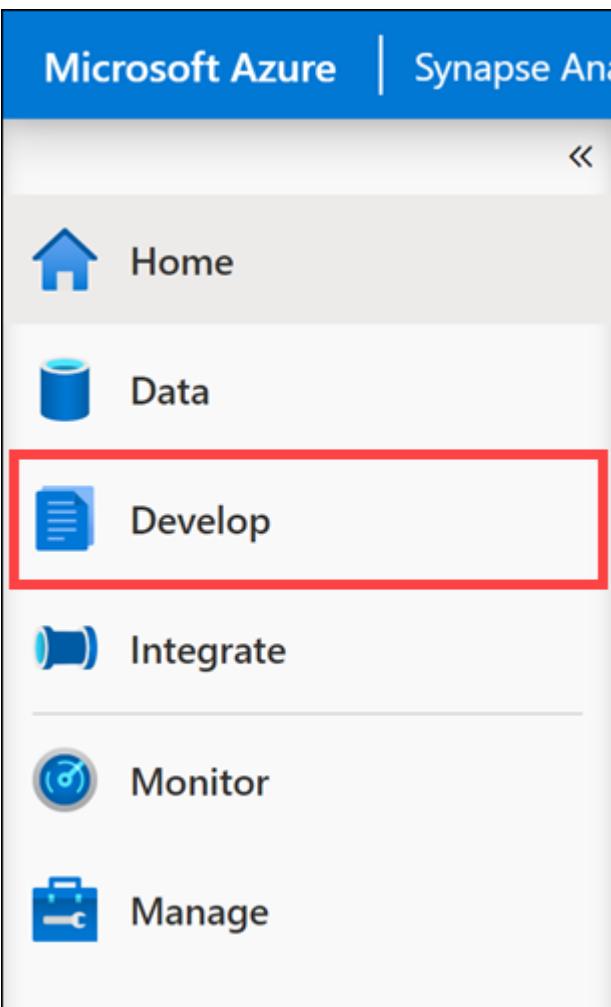


```
SQL script 1
Run Undo Publish Query plan Connect to SQLPool01 Use database SQLPool01
1 CREATE TABLE [dbo].[CustomerSource] (
2     [CustomerID] [int] NOT NULL,
3     [Title] [nvarchar](8),
4     [FirstName] [nvarchar](50),
5     [MiddleName] [nvarchar](50),
6     [LastName] [nvarchar](50),
7     [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://solliancepublicdata.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 )
```

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.

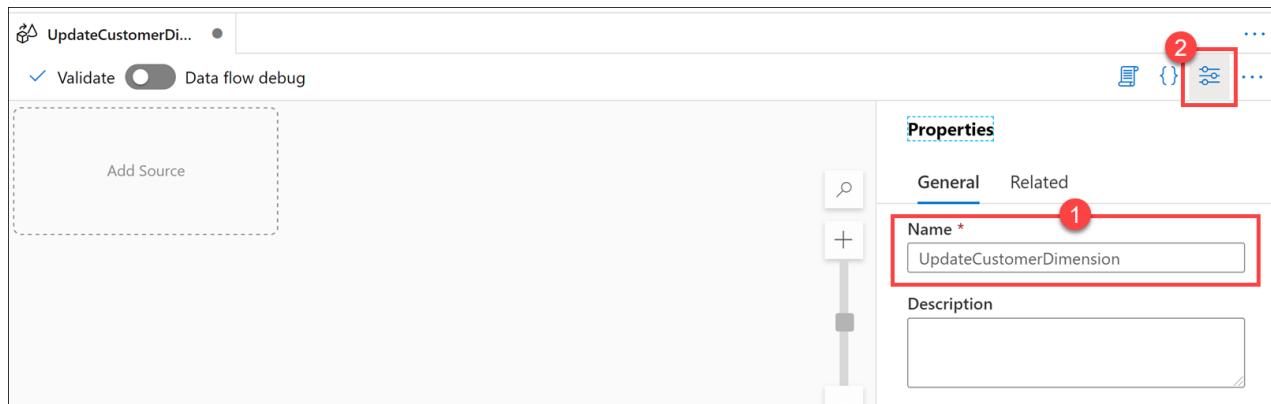


2. Select +, then select Data flow.

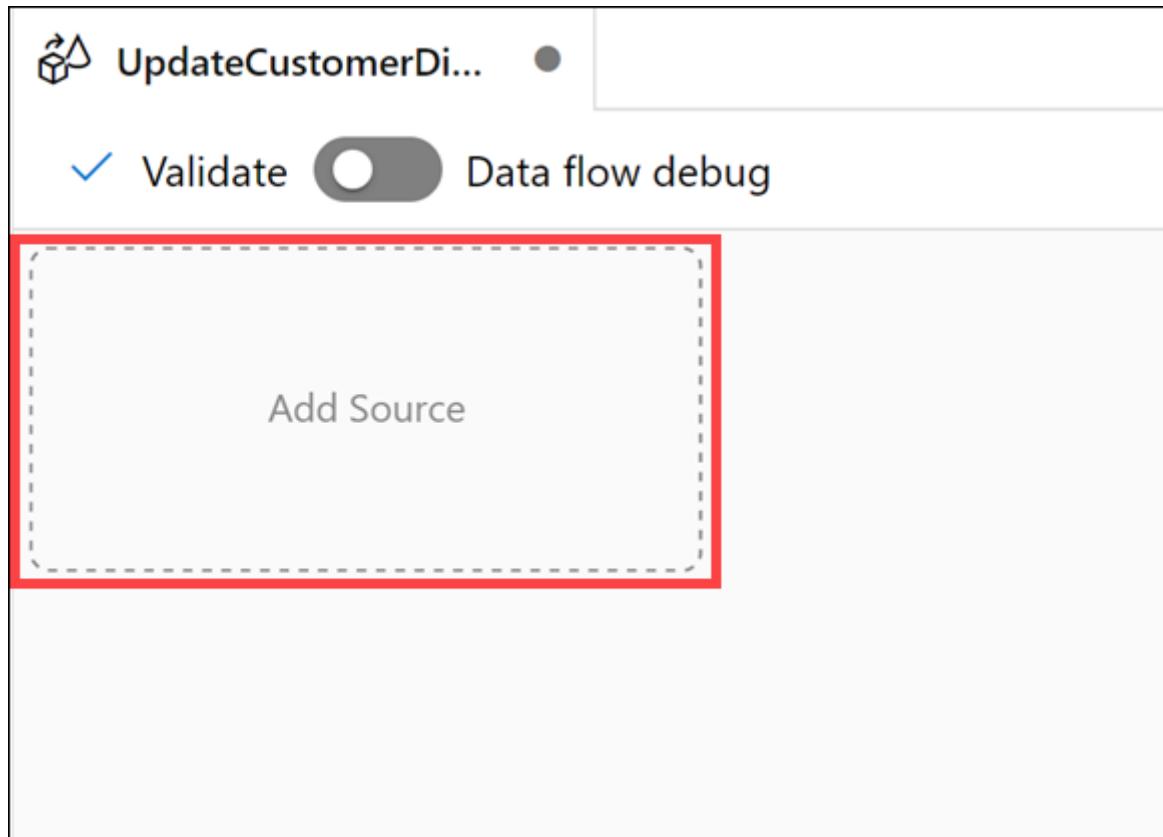
The screenshot shows the "Develop" page in the Microsoft Azure Synapse Analytics portal. At the top left is the title "Develop". To the right is a button with a blue plus sign (+) which is highlighted with a red box. Below the button is a search bar labeled "Filter resources by name". To the right of the search bar are several options:

- SQL script
- Notebook
- Data flow (highlighted with a red box)
- Apache Spark job definition
- Browse gallery
- Import

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.



4. Select Add Source on the 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 * Learn more 

Source type *

Dataset * 

Options

Allow schema drift 

Infer drifted column types 

Validate schema 

Sampling *  Enable Disable

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

 Search

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

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

Set properties

Name
CustomerSource

Linked service *
asagaworkspacedv031721-WorkspaceDefaultSqlServer 

Connect via integration runtime * 
AutoResolveIntegrationRuntime 

Table name
 

Edit

Import schema
 From connection/store None

 Advanced

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	<input type="text" value="SQLPool01"/> 

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

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 *

CustomerSource

Test connection



+ New

Options

Allow schema drift ⓘ

Infer drifted column types ⓘ

Validate schema ⓘ

Sampling * ⓘ

Enable Disable

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.

The screenshot shows the 'Source settings' page for a dataset named 'DimCustomer'. The 'Output stream name' is set to 'DimCustomer'. The 'Source type' is set to 'Dataset'. The 'Dataset' dropdown is set to 'DimCustomer'. Under 'Options', 'Allow schema drift' is checked. Under 'Sampling', 'Disable' is selected.

Setting	Value
Output stream name *	DimCustomer
Source type *	Dataset
Dataset *	DimCustomer
Options	<input checked="" type="checkbox"/> Allow schema drift <small> ⓘ </small>
	<input type="checkbox"/> Infer drifted column types <small> ⓘ </small>
	<input type="checkbox"/> Validate schema <small> ⓘ </small>
Sampling *	<input type="radio"/> Enable <input checked="" type="radio"/> Disable

Add transformations to data flow

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

The screenshot shows the 'Source settings' tab of the SSIS Derived Column properties dialog. At the top, there are two source components: 'SourceDB' (with 10 total columns) and 'DimCustomer'. Below them are tabs for 'Source settings' (selected) and 'Source options'. The 'Source type' dropdown is set to 'Derived Column', which is highlighted with a red box. Other options include 'Select', 'Aggregate', 'Surrogate Key', 'Pivot', 'Unpivot', 'Window', and 'None'. The 'Options' section contains checked boxes for 'Allow inferences' and 'Infer column types'. The 'Sampling' section has a radio button for 'Enabled'.

Source settings Source options

Output stream name *

Source type *

Derived Column

Select

Aggregate

Surrogate Key

Pivot

Unpivot

Window

None

Dataset *

Options

Sampling *

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

- **Output stream name:** Enter CreateCustomerHash
- **Incoming stream:** Select SourceDB
- **Columns:** Enter the following:

Column	Expression	Description
Type in HashKey	sha2(256, iifNull>Title,'') +FirstName +iifNull(MiddleName,'') +LastName +iifNull(Suffix,'') +iifNull(CompanyName,'') +iifNull(SalesPerson,'') +iifNull(EmailAddress,'') +iifNull(Phone,''))	Creates a SHA256 hash of the table values. We use this to detect row changes by comparing the hash of the incoming records to the hash value of the destination records, matching on the CustomerID value. The iifNull function replaces null values with empty strings. Otherwise, the hash values tend to duplicate when null entries are present.

Derived column's settings Optimize Inspect Data preview Description ^

Output stream name * Learn more [🔗](#)

Incoming stream *

+ Add [Clone](#) [Delete](#) Open expression builder

Columns * ⓘ

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

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

CreateCustomerHash
Columns: 16 total

+

Search

Multiple inputs/outputs

Join

Conditional Split

Exists

Union

Optimize Inspect Data preview

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

Exists settings Optimize Inspect Data preview ● Description ^

Output stream name * Learn more [🔗](#)

Left stream *

Right stream *

Exist type * Exists Doesn't exist

Custom expression ⓘ

Exists conditions * **Left: CreateCustomerHash's column** abc HashKey **Right: SynapseDimCustomer's column** abc HashKey == + [Delete]

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

Multiple inputs/outputs

- Join
- Conditional Split
- Exists
- Union
- Lookup**

6. Under **Lookup settings**, configure the following properties:

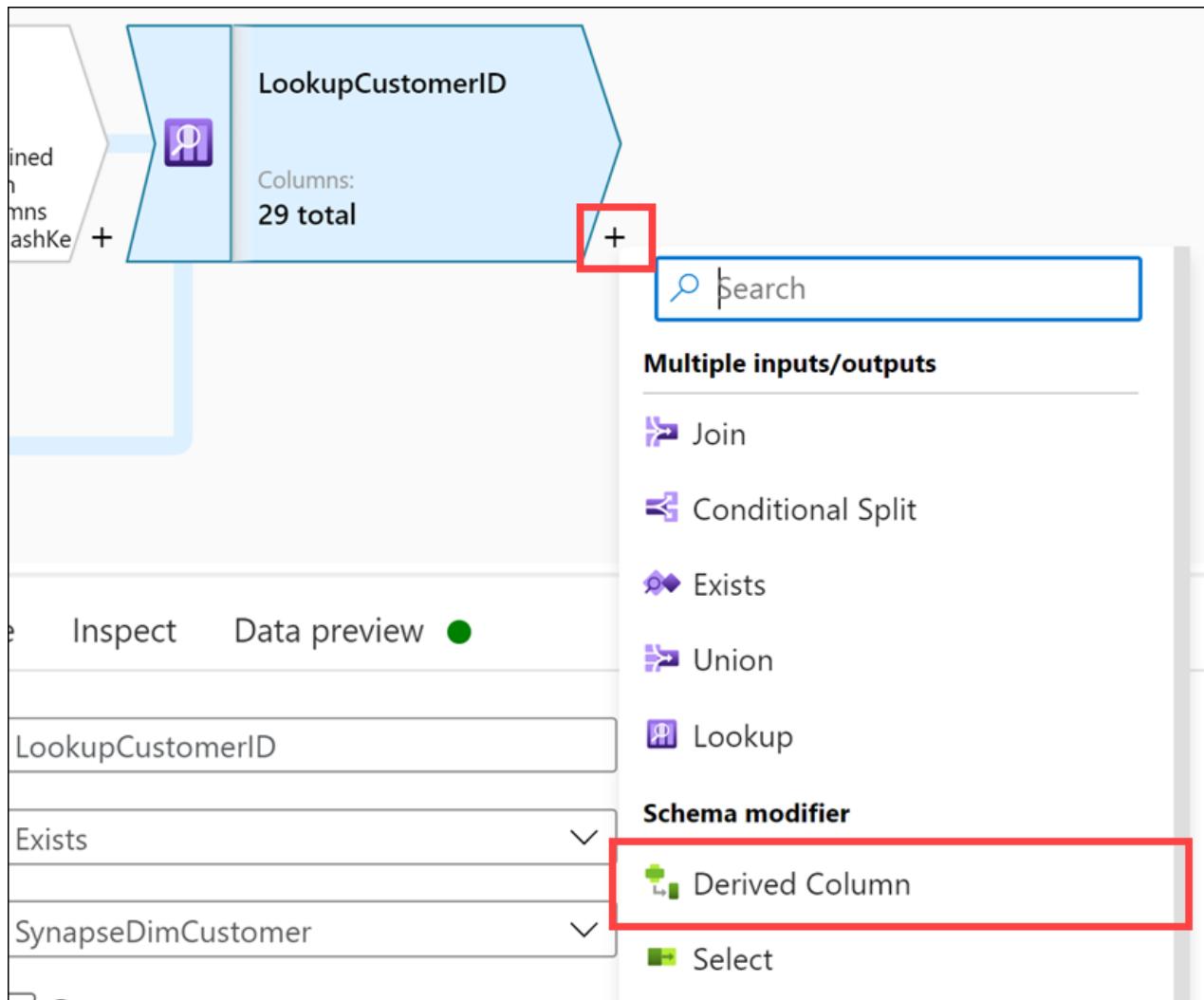
- **Output stream name:** Enter `LookupCustomerID`
- **Primary stream:** Select `Exists`
- **Lookup stream:** Select `SynapseDimCustomer`
- **Match multiple rows:** Unchecked
- **Match on:** Select `Any row`
- **Lookup conditions:** Set the following for Left and Right:

Left: Exists's column	Right: SynapseDimCustomer's column
<code>CustomerID</code>	<code>CustomerID</code>

The screenshot shows the 'Lookup settings' page with the following configuration:

- Output stream name ***: `LookupCustomerID`
- Primary stream ***: `Exists`
- Lookup stream ***: `SynapseDimCustomer`
- Match multiple rows**: Unchecked
- Match on ***: `Any row`
- Lookup conditions *** (highlighted with a red box):
 - Left: Exists's column**: `123 CustomerID`
 - Right: SynapseDimCustomer's column**: `123 CustomerID`
 - Operator: `=`

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



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	iif(isNull(InsertedDate), currentTimestamp(), {InsertedDate})	If the InsertedDate value is null, insert the current timestamp. Otherwise, use the InsertedDate value.
Select ModifiedDate	currentTimestamp()	Always update the ModifiedDate value with the current timestamp.

Derived column's settings Optimize Inspect Data preview ● Description

Output stream name * SetDates Learn more ↗

Incoming stream * LookupCustomerID

+ Add Clone Delete Open expression builder

Columns * ⓘ

Column	Expression
InsertedDate	iif(isNull(InsertedDate), currentTimestamp(), {Insert...})
ModifiedDate	currentTimestamp()

ⓘ 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**.

CustomerID
s from tomer

+

SetDates
Columns: 28 total

+

Surrogate Key

Pivot

Unpivot

Window

Flatten

Rank

Row modifier

Filter

Sort

Alter Row

Optimize Inspect Data preview ●

SetDates

LookupCustomerID

+ Add Clone Delete Open expression builder

Column

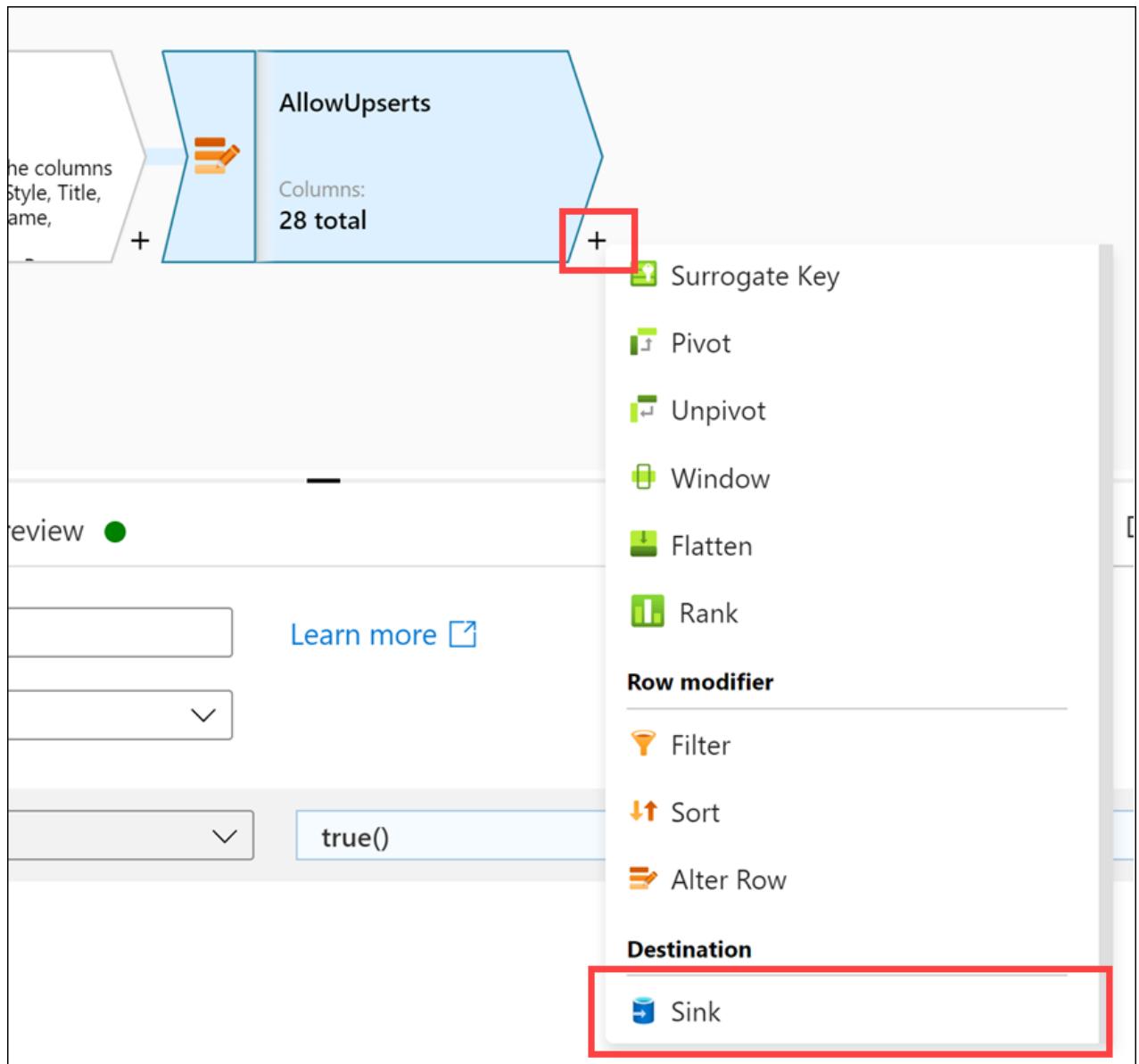
10. Under Alter row settings, configure the following properties:

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

Condition	Expression	Description
Select Up insert if	true()	Set the condition to true() on the Up insert 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.

The screenshot shows the 'Alter row settings' configuration page. At the top, there are tabs for 'Alter row settings' (which is selected), 'Optimize', 'Inspect', and 'Data preview'. Below these are three main input fields: 'Output stream name' set to 'AllowUpserts', 'Incoming stream' set to 'SetDates', and 'Alter row conditions' set to 'Up insert if true()'. The 'Alter row conditions' field is highlighted with a red box.

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



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 ⓘ

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 ●

💡 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 * ⓘ List of columns Custom expression ⓘ
 [🔗 Add dynamic content \[Alt+P\]](#) [➕](#) [trash](#)

Skip writing key columns

Table action **None** Recreate table Truncate table

Enable staging

Batch size ⓘ

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

Input columns

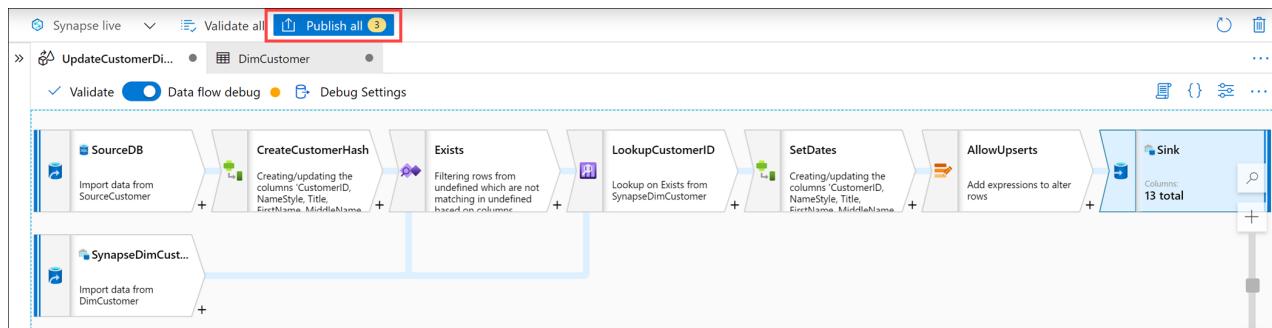
Output columns

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

The screenshot shows the 'Mapping' tab of the Azure Data Flow interface. At the top, there are several tabs: Sink, Settings, **Mapping**, Optimize, Inspect, and Data preview. The 'Mapping' tab is selected and highlighted with a red box. Below the tabs, there are sections for Options, Auto mapping, and Output format. The 'Auto mapping' section is also highlighted with a red box. The main area displays the mapping between 'Input columns' from SourceDB and 'Output columns' to Sink. A large red box highlights the entire list of input columns on the left.

Input columns	Output columns
123 CustomerID	123 CustomerID
abc SourceDB@Title	abc Title
abc SourceDB@FirstName	abc FirstName
abc SourceDB@MiddleName	abc MiddleName
abc SourceDB@LastName	abc LastName
abc SourceDB@Suffix	abc Suffix
abc SourceDB@CompanyName	abc CompanyName
abc SourceDB@SalesPerson	abc SalesPerson
abc SourceDB@EmailAddress	abc EmailAddress
abc SourceDB@Phone	abc Phone
InsertedDate	InsertedDate
ModifiedDate	ModifiedDate
abc CreateCustomerHash@HashKey	abc HashKey

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



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

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.

The screenshot shows the Azure Data Studio interface with a SQL query results tab. The query is:

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

The results table has columns: CustomerId, FirstName, LastName, and ModifiedDate. One row is shown for CustomerId 4, FirstName Janet, LastName Gates, and ModifiedDate 2021-03-27T05:01:17.9170000. The 'LastName' cell is highlighted with a red box.

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

SQL

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

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

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

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.