

Azure Data Factory

Lab - Incrementally load data from multiple tables

Overview

Here are the important steps to create this solution:

1. Select the watermark column.

Select one column for each table in the source data store, which can be used to identify the new or updated records for every run. Normally, the data in this selected column (for example, last_modify_time or ID) keeps increasing when rows are created or updated. The maximum value in this column is used as a watermark.

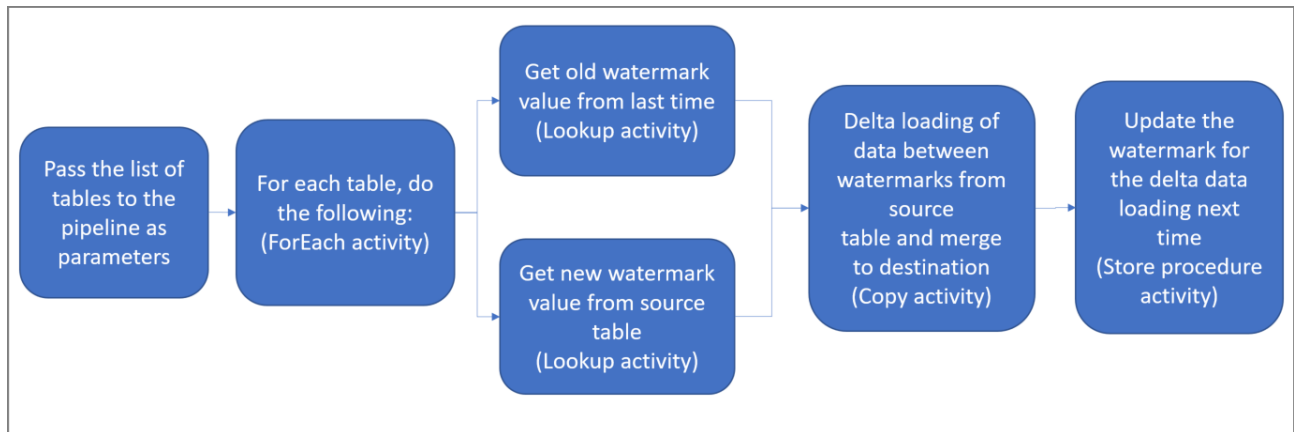
2. Prepare a data store to store the watermark value.

In this tutorial, you store the watermark value in a SQL database.

3. Create a pipeline with the following activities:

- a. Create a ForEach activity that iterates through a list of source table names that is passed as a parameter to the pipeline. For each source table, it invokes the following activities to perform delta loading for that table.
- b. Create two lookup activities. Use the first Lookup activity to retrieve the last watermark value. Use the second Lookup activity to retrieve the new watermark value. These watermark values are passed to the Copy activity.
- c. Create a Copy activity that copies rows from the source data store with the value of the watermark column greater than the old watermark value and less than the new watermark value. Then, it copies the delta data from the source data store to Azure Blob storage as a new file.
- d. Create a StoredProcedure activity that updates the watermark value for the pipeline that runs next time.

Here is the high-level solution diagram:



Reference: Microsoft Pages

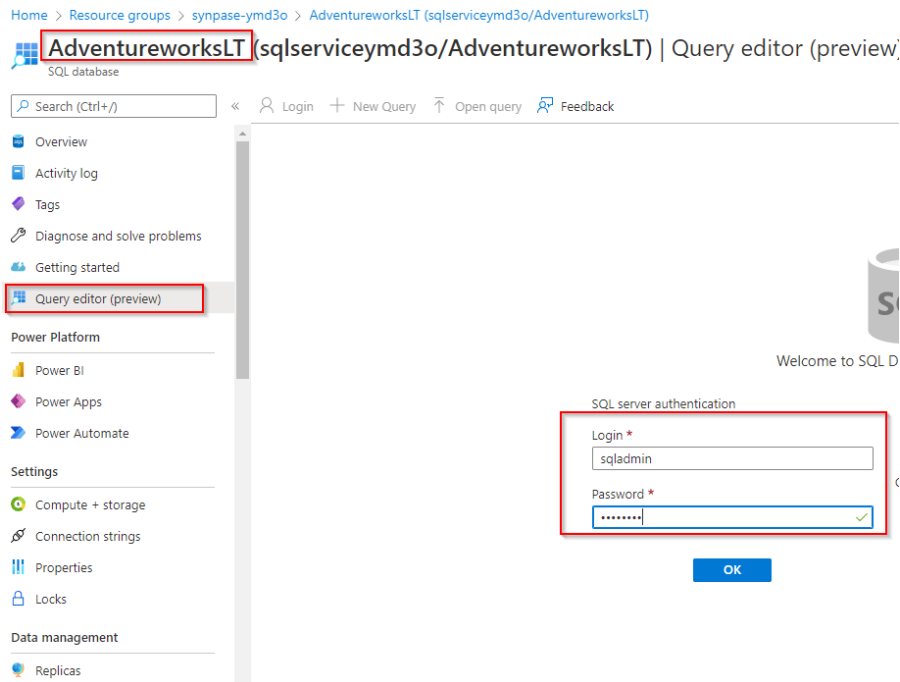
In this exercise you will be performing following task:

- Task 1: Create and configure a SQL Database instance.
- Task 2: Create tables and procedure
- Task 3: Create a pipeline

Task 1: Create and configure a Destination SQL Database in another region.

Create source tables in your SQL database (SQLDB)

1. Open SQL Database, and connect to your SQL Server database.



2. Run the following SQL command against your database to create tables named customer_table and project_table:

```
create table customer_table
```

```
(  
  PersonID int,  
  Name varchar(255),  
  LastModifytime datetime  
);
```

```
create table project_table
```

```
(  
  Project varchar(255),  
  Creationtime datetime  
);
```

```
INSERT INTO customer_table  
(PersonID, Name, LastModifytime)  
VALUES  
(1, 'John','9/1/2017 12:56:00 AM'),  
(2, 'Mike','9/2/2017 5:23:00 AM'),  
(3, 'Alice','9/3/2017 2:36:00 AM'),  
(4, 'Andy','9/4/2017 3:21:00 AM'),  
(5, 'Anny','9/5/2017 8:06:00 AM');
```

```

INSERT INTO project_table
(Project, Creationtime)
VALUES
('project1','1/1/2015 0:00:00 AM'),
('project2','2/2/2016 1:23:00 AM'),
('project3','3/4/2017 5:16:00 AM');

```

Create destination tables in destination SQL Database (SQLDB)

1. Open SQL Database, and connect to your SQL Server database.
2. Run the following SQL command against your database to create tables named customer_table_dest and project_table_dest:.

```

create table customer_table_dest
(
    PersonID int,
    Name varchar(255),
    LastModifytime datetime
);
GO

```

```

create table project_table_dest
(
    Project varchar(255),
    Creationtime datetime
);
GO

```

```

create table table_list
(
    TableName varchar(255),
    WaterMarkColumn varchar(255),
    UpsertColumn varchar(255)
);
GO

```

```

INSERT INTO table_list (TableName, WaterMarkColumn, UpsertColumn) VALUES ('customer_table',
'LastModifytime', 'PersonID')
GO

```

```

INSERT INTO table_list (TableName, WaterMarkColumn, UpsertColumn) VALUES
('project_table','Creationtime', 'Project')
GO

```

Create a stored procedure in your SQL Database (SQLDB)

1. Run the following command to create a stored procedure in your database. This stored procedure updates the watermark value after every pipeline run

```
create table watermarktable
(
    TableName varchar(255),
    WatermarkValue datetime
);
```

3. Insert initial watermark values for both source tables into the watermark table.

```
INSERT INTO watermarktable VALUES ('customer_table','1/1/2010 12:00:00 AM')
INSERT INTO watermarktable VALUES ('project_table','1/1/2010 12:00:00 AM')
```

Create a stored procedure in your destination database (SQLDB)

Run the following command to create a stored procedure in your database. This stored procedure updates the watermark value after every pipeline run.

```
CREATE PROCEDURE usp_write_watermark @LastModifiedtime datetime, @TableName varchar(50)
AS
```

```
BEGIN
```

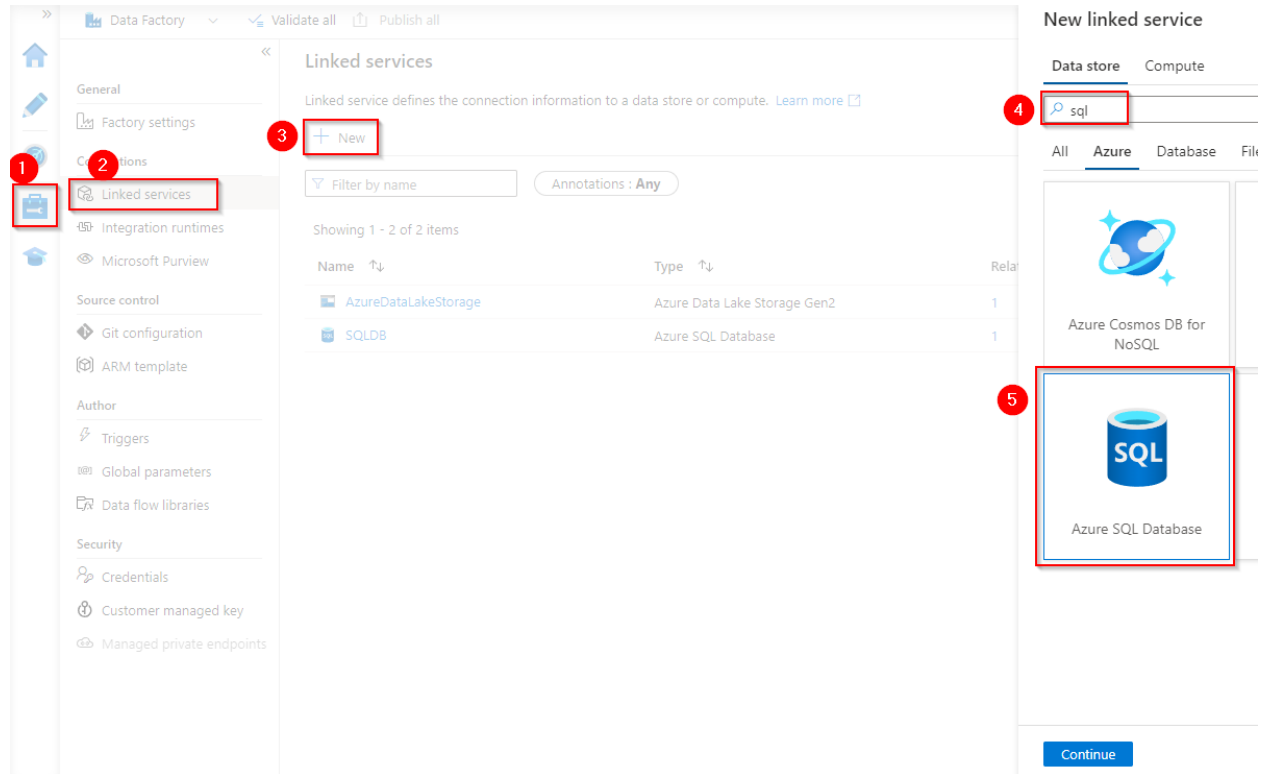
```
    UPDATE watermarktable
    SET [WatermarkValue] = @LastModifiedtime
    WHERE [TableName] = @TableName
```

```
END
```

Task 3 : Create and configure a pipeline.

Create SQL Database linked service for source and destination databases.

1. Under manage hub -> Linked Service -> + New -> Search for SQL Database -> Select SQL Database -> Click Continue



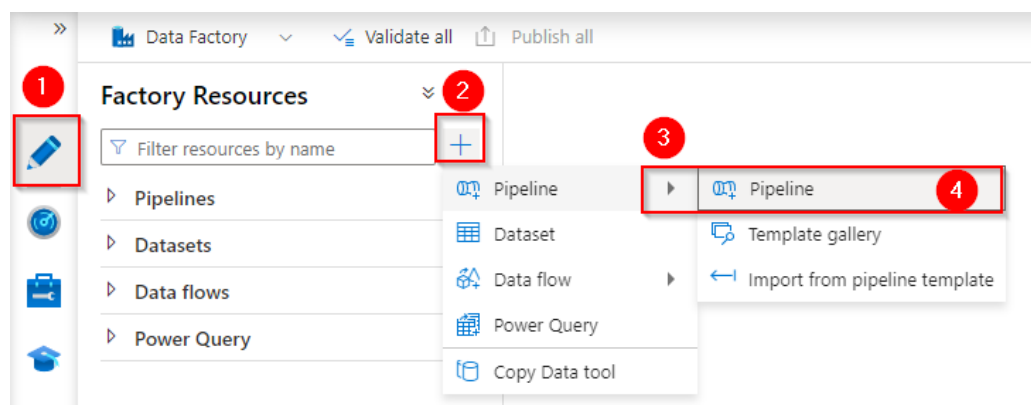
2. Provide the required configuration details for the Linked service.

Source Database Linked Service Name: AzureSQLDatabaseSource

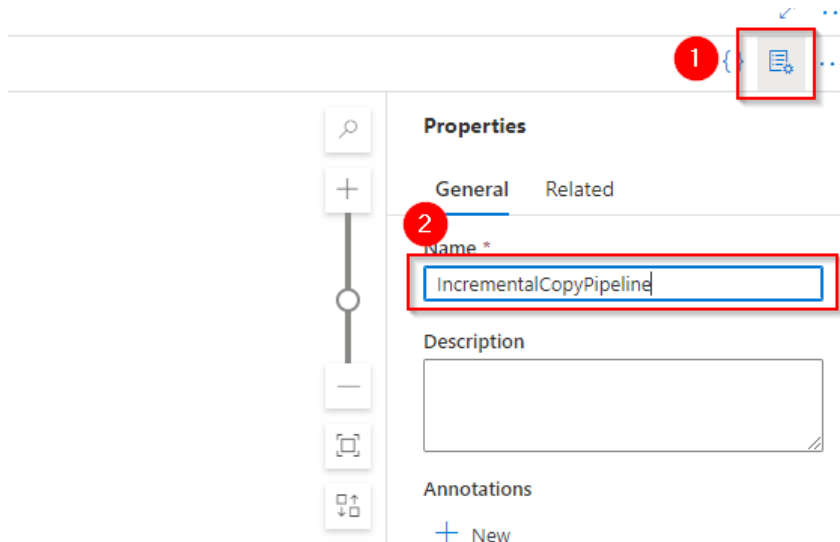
Destination Database Linked Service Name: AzureSQLDatabaseDest

Create a Pipeline

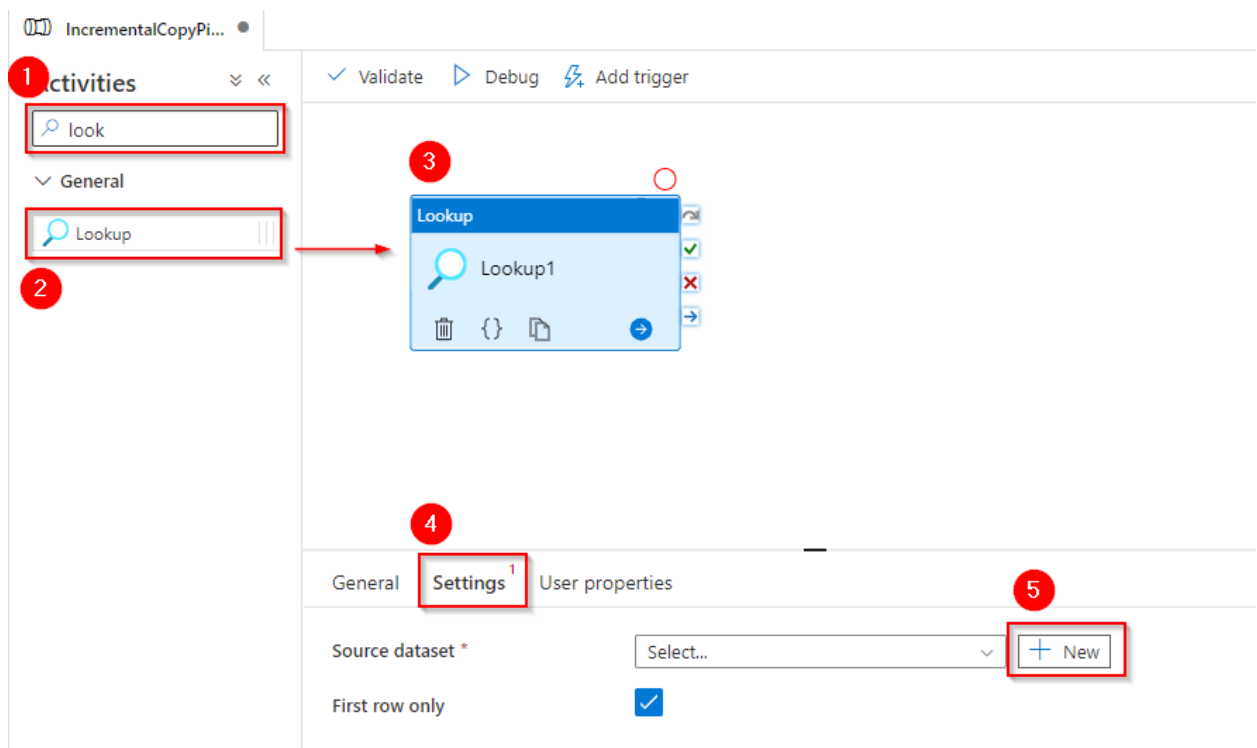
1. In the Author tab, click + (plus), and click Pipeline.



2. In the General panel under Properties, specify IncrementalCopyPipeline for Name.



3. Add lookup activity to pipeline and switch to the Settings tab, click + New to create a dataset as shown below:



- Select Azure SQL Database
- Set properties
 - Dataset Name: **AzureSynapseAnalyticsTableList**
 - Select Linked Service: **AzureSQLDatabaseDest**
 - Select Table Name: **table_list**
 - Click: **Ok**

Set properties

Name

AzureSynapseAnalyticsTableList

1

Linked service *

AzureSQLDatabaseDest

2

Table name

dbo.table_list

3

☐ Edit

Import schema



☒ From connection/store ☐ None

4

> Advanced

- Under setting tab of Lookup activity
 - IN General Tab enter the name: **LookupTableList**
 - Select Frist Row only: **Unchecked**
 - Use query: **Table**

General **Settings** User properties

Source dataset * AzureSynapseAnalyticsTableList  Open  New

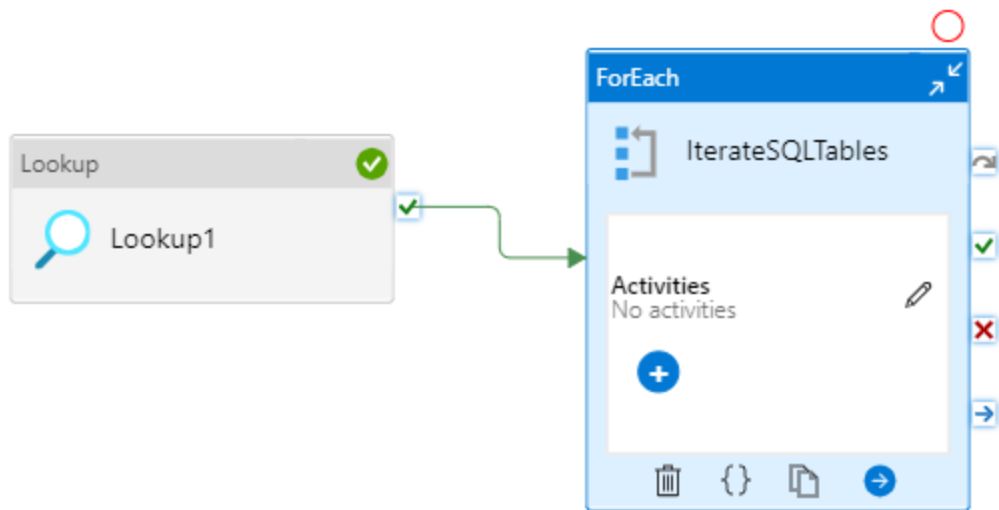
First row only ☐

Use query ☒ Table ☐ Query ☐ Stored procedure

Query timeout (minutes) ⓘ 120

Isolation level ⓘ None

4. In the Activities toolbox, expand Iteration & Conditionals, and drag-drop the ForEach activity to the pipeline designer surface. In the General tab of the Properties window, enter **IterateSQLTables**.



General Settings ¹ Activities (0) User properties

Name * [Learn more](#)

- Switch to the Settings tab, and enter **@activity('LookupTableList').output.value** for Items. The ForEach activity iterates through a list of tables and performs the incremental copy operation

1 General **Settings** Activities (0) User properties

Sequential ☐

Batch count ⓘ

Items * 2

- Select the ForEach activity in the pipeline if it isn't already selected. Click the Edit (Pencil icon) button.
- In the Activities toolbox, expand General, drag-drop the Lookup activity to the pipeline designer surface, and enter **LookupOldWaterMarkActivity** for Name.
- Switch to the Settings tab, click + New to create a dataset as mentioned below:
 - Search for SQL Database then select Azure SQL Database and click continue.
 - Set properties:

- Name: **WatermarkDataset**
- Linked Service Select: **AzureSQLDatabaseDest**
- Table Name: **watermarktable**
- Click" **Ok**

Set properties

1

Name

WatermarkDataset

2

Linked service *

AzureSQLDatabaseDest

3

Table name

dbo.watermarktable

4

Import schema

☒ From connection/store ☐ None

- First row only: **Checked**.
- Use Query: Select **Query**.
- Enter the following SQL query for Query.

select * from watermarktable where TableName = '{@item().TableName}'

1

Settings

User properties

Source dataset *

WatermarkDataset

Open + New

2

First row only

☒

3

Use query

☐ Table ☒ Query ☐ Stored procedure

4

Query

select * from watermarktable where
TableName = '{@item().TableName}'

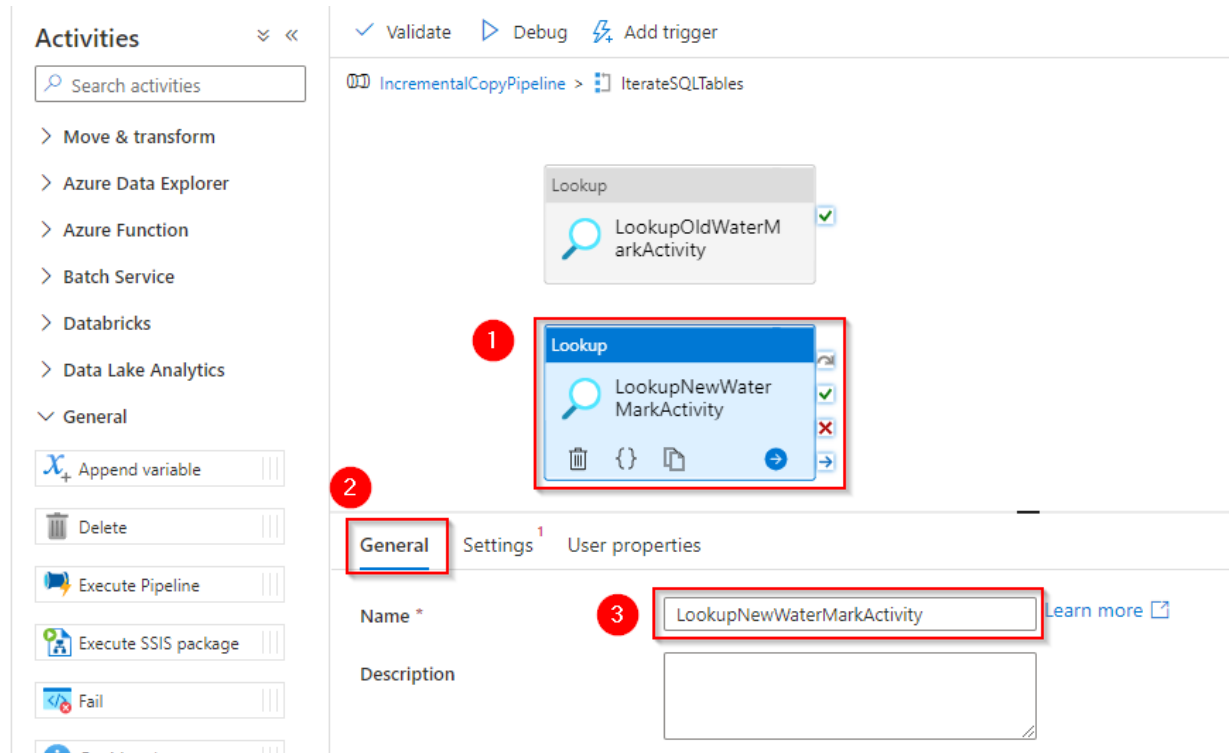
Query timeout (minutes) ⓘ

120

Isolation level ⓘ

None

9. Drag-drop the Lookup activity from the Activities toolbox, and enter **LookupNewWaterMarkActivity** for Name.



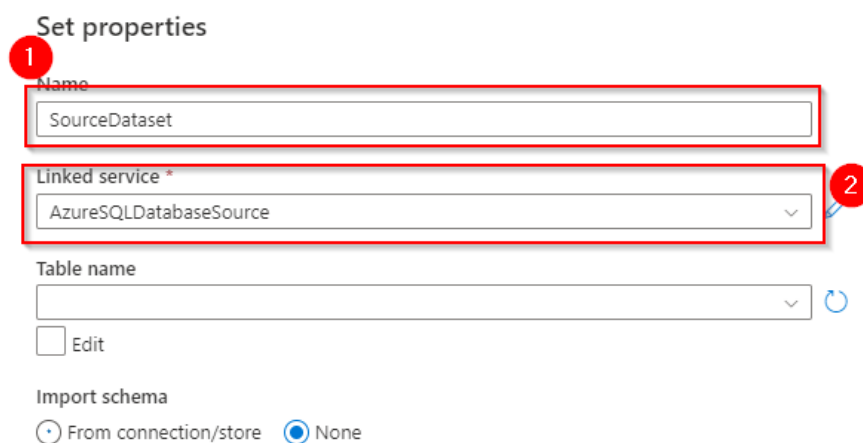
10. Switch to the Settings tab, click + New to create a dataset as mention below:.

- Search for SQL Database then select Azure SQL Database and click continue.
- Set properties:

- Name: **SourceDataset**
- Linked Service Select: **AzureSQLDatabaseSource**
- Table Name: **You do not select a table here.**

Note: The Copy activity in the pipeline uses a SQL query to load the data rather than load the entire table.

- Click" **Ok**



- First row only: **Checked**.
- Select **Query** for Use Query.
- Enter the following SQL query for Query.

```
select MAX(@{item().WaterMarkColumn}) as NewWatermarkvalue from @{item().TableName}
```

General **1 Settings** User properties

Source dataset * SourceDataset

First row only **2** ☒

Use query ☐ Table **3** ☒ Query ☐ Stored procedure

Query **4** `select MAX(@{item().WaterMarkColumn})
as NewWatermarkvalue from
@{item().TableName}`

11. Drag-drop the Copy activity from the Activities toolbox and enter **IncrementalCopyActivity** for Name.
12. Connect Lookup activities to the Copy activity one by one. To connect, start dragging at the green box attached to the Lookup activity and drop it on the Copy activity. Release the mouse button when the border color of the Copy activity changes to blue.

Activities

Search activities

1 Move & transform

Copy data

Data flow

Azure Data Explorer

Azure Function

Batch Service

Databricks

Data Lake Analytics

General

HDInsight

Iteration & conditionals

✓ Validate ✓ Validate copy runtime ▶ Debug ⚡ Add trigger

IncrementalCopyPipeline > IterateSQLTables

Lookup

LookupOldWaterMarkActivity **3**

Lookup

LookupNewWaterMarkActivity **4**

2 Copy data

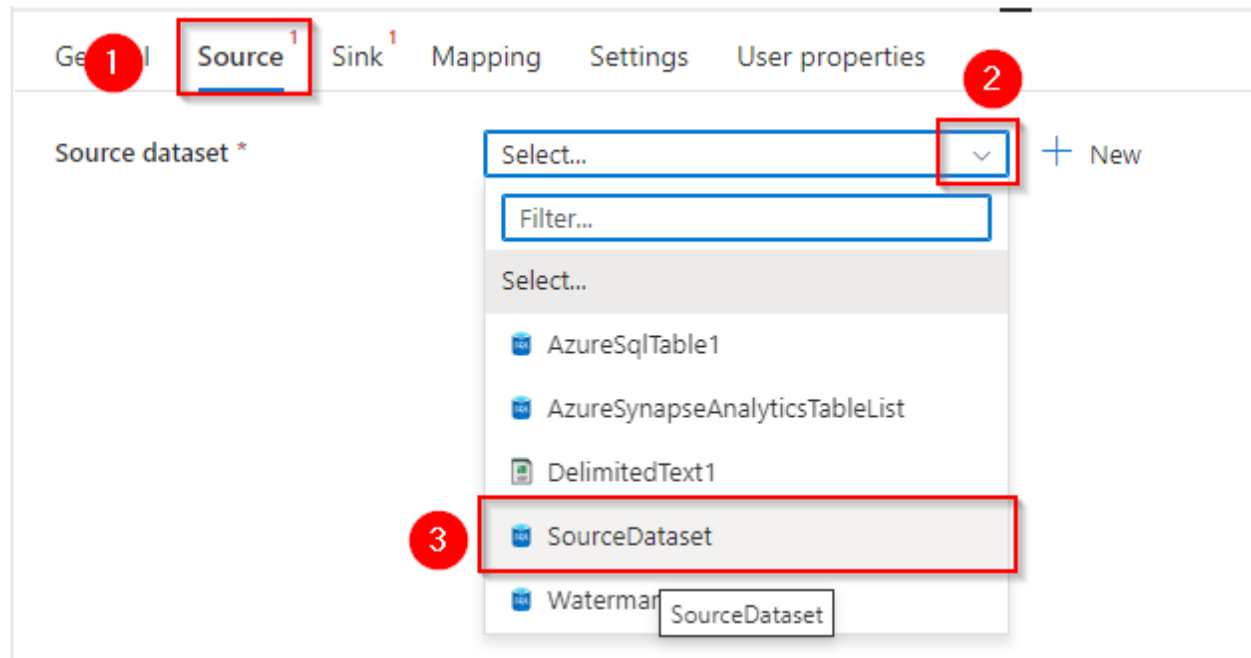
IncrementalCopyActivity

General **Source** **1 Sink** **1** Mapping Settings User properties

Name * **5** IncrementalCopyActivity [learn more](#)

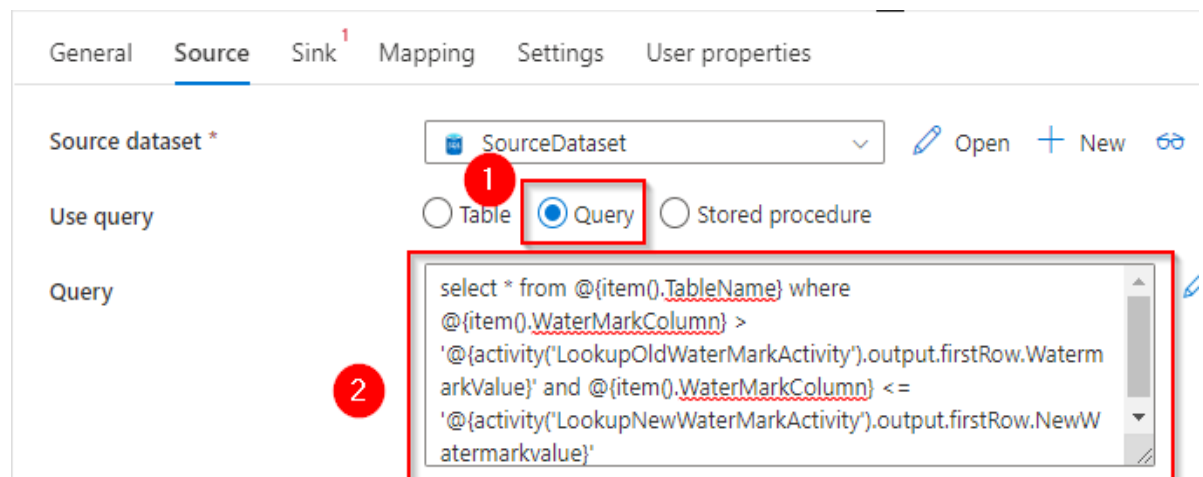
13. Select the Copy activity in the pipeline. Switch to the Source tab in the Properties window.

- Select **SourceDataset** for Source Dataset.



- Select **Query** for Use Query.
- Enter the following SQL query for Query.

```
select * from @item().TableName where @item().WaterMarkColumn >
'@{activity('LookupOldWaterMarkActivity').output.firstRow.WatermarkValue}' and
@item().WaterMarkColumn <=
'@{activity('LookupNewWaterMarkActivity').output.firstRow.NewWatermarkvalue}'
```



14. Switch to the Sink tab, and click + New for Sink Dataset as mention below:

- Search for SQL then select **Azure SQL Database** and click continue.

- Set properties:
 - Name: **SinkDataset**
 - Linked Service Select: **AzureSQLDatabaseDest**
 - Table Name: **You do not select a table here.**
 - Click" **Ok**

Set properties

1

Name

SinkDataset

Linked service *

AzureSQLDatabaseDest

2

Table name

Edit

Import schema

☐ From connection/store ☒ None

- In the sink tab select open

General Source **Sink** Mapping Settings User properties

Sink dataset *

SinkDataset

Open

Write behavior

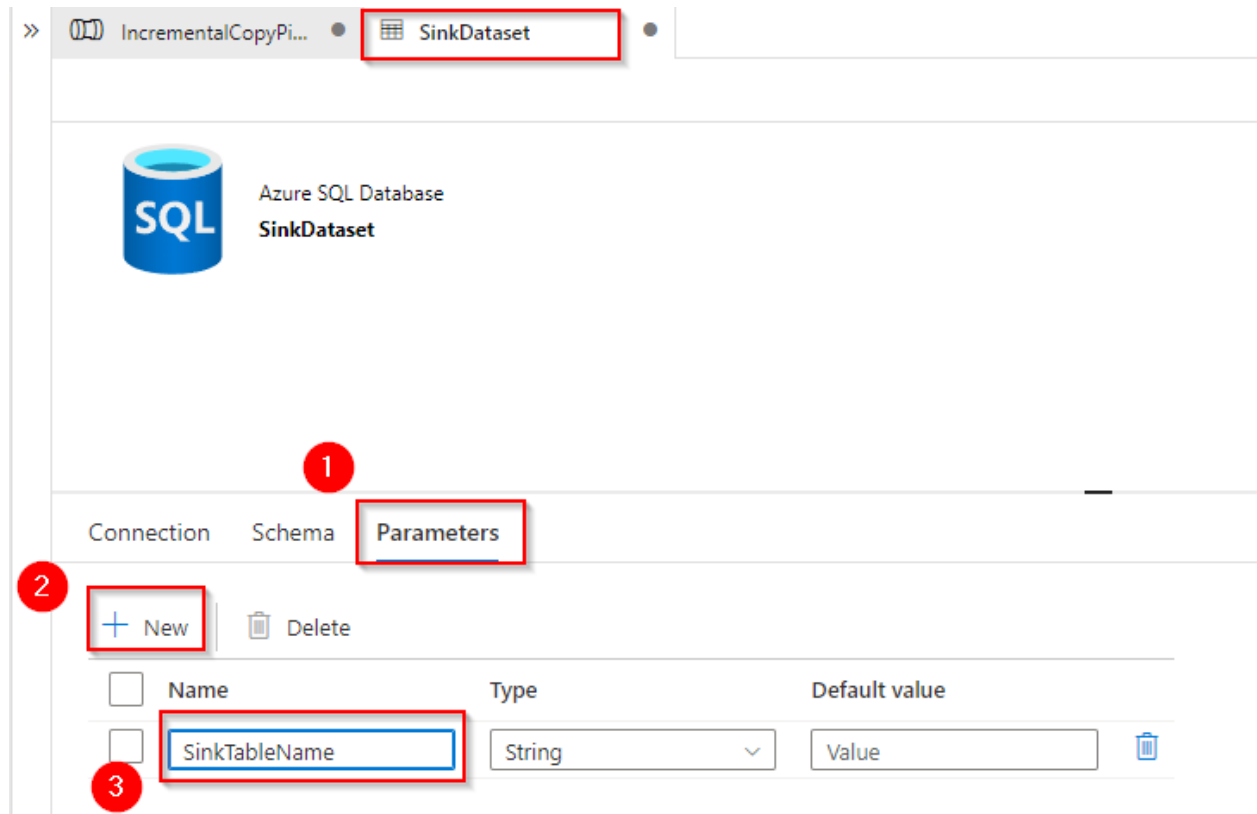
☒ Insert ☐ Upsert ☐ Stored procedure

Bulk insert table lock ⓘ

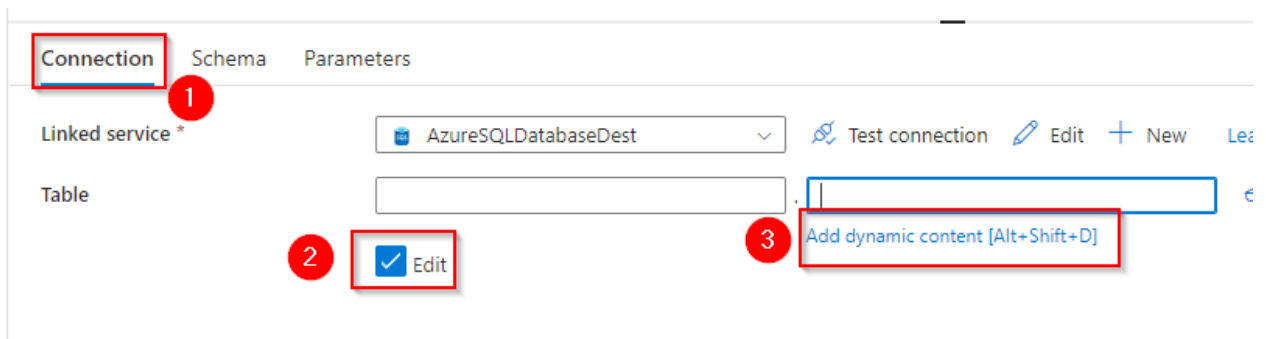
☐ Yes ☒ No

15. Switch to the Parameters tab in the Properties window of **SinkDataset**, and do the following steps:

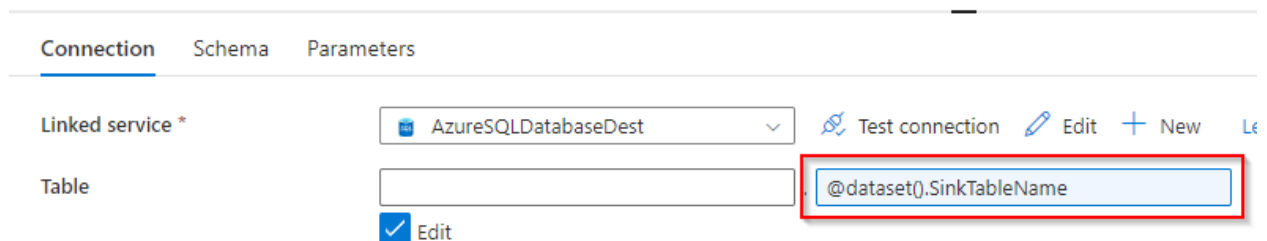
- Click **New** in the Create/update parameters section.
- Enter **SinkTableName** for the name, and String for the type. This dataset takes **SinkTableName** as a parameter. The SinkTableName parameter is set by the pipeline dynamically at runtime. The ForEach activity in the pipeline iterates through a list of table names and passes the table name to this dataset in each iteration.



16. Switch back to the **Connection** tab in the Properties window and For Table property, click **Add dynamic content**.



- In the Add Dynamic Content window, select **SinkTableName** in the Parameters section.
- After clicking Finish, you see "**@dataset().SinkTableName**" as the table name.



17. Switch back to the **Sink** tab of copy activity in the pipeline, and do the following steps:

- In the Dataset properties, for **SinkTableName** parameter, enter `@{concat(item().TableName,'_dest')}`
- Copy method, **upsert**
- In Key column select **+ New** and enter `@{item().UpsertColumn}`

The screenshot shows the configuration for the Sink tab of a Copy activity. The 'Sink dataset' is set to 'SinkDataset'. Under 'Dataset properties', the 'SinkTableName' is configured with the value '@{item().TableName}'. The 'Write behavior' is set to 'Upsert'. The 'Use TempDB' checkbox is checked. In the 'Key columns' section, a new key column is added with the value '@{item().UpsertColumn}'. The 'Bulk insert table lock' is set to 'No', and the 'Table option' is set to 'None'.

1 Sink tab selected

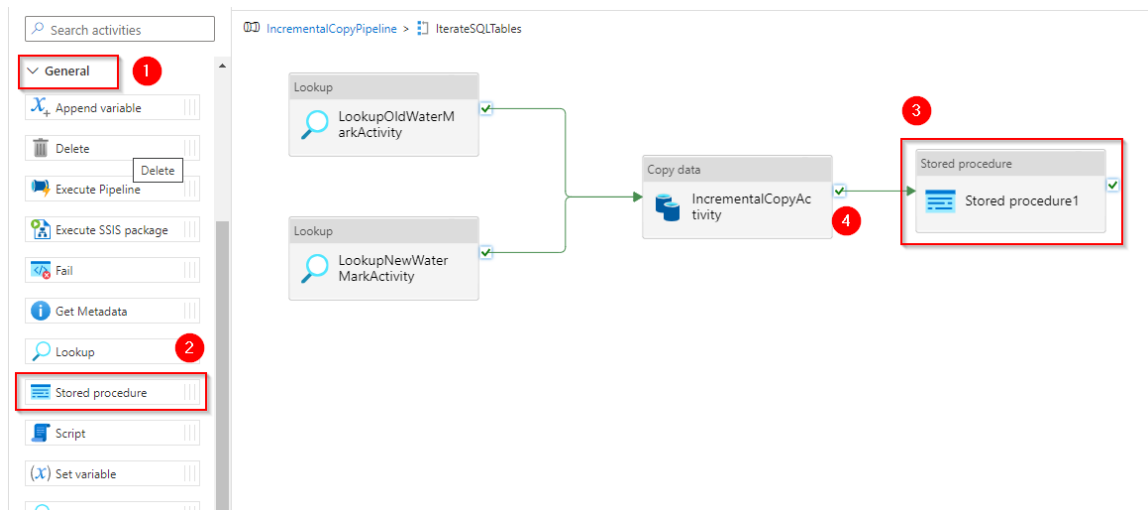
2 SinkTableName value: @{item().TableName}

3 Upsert selected

4 + New button

5 @{item().UpsertColumn} value

18. Drag-and-drop the **Stored Procedure** activity under **General** section from the Activities toolbox to the pipeline designer surface. Connect the Copy activity to the Stored Procedure activity.



19. Select the **Stored Procedure** activity in the pipeline, and enter **StoredProceduretoWriteWatermarkActivity** for Name in the **General** tab of the Properties window.

20. Switch to the Setting tab, and do the following steps:

- SQL pool: **AzureSQLDatabaseDest**
- Stored Procedure Parameters select: **[dbo].[usp_write_watermark]**
- Select **Import parameter** under Store procedure parameters.

General **Settings** User properties

1

Linked service * ⓘ 2 AzureSQLDatabaseDest Test connection

Stored procedure name * 3 [dbo].[usp_write_watermark] Refresh

4

Stored procedure parameters ⓘ

5 Import + New

Enter the following value:

Name	Type	Value
LastModified time	DateTime	@{activity('LookupNewWaterMarkActivity').output.firstRow.NewWatermarkvalue}
TableName	String	@{activity('LookupOldWaterMarkActivity').output.firstRow.TableName}

Edit ⓘ

Stored procedure parameters ⓘ

Import + New Delete






Name	Type	Value
LastModifiedtime	DateTime	@{activity('LookupNewWaterMarkActi...
TableName	String	@{activity('LookupOldWaterMarkActiv...

21. Select Publish All to publish the entities you created.

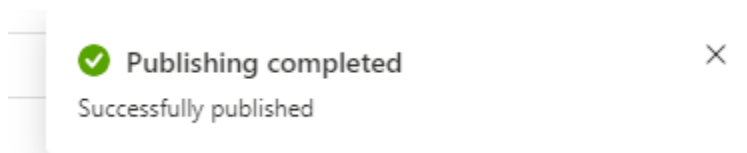
Publish all

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

Pending changes (5)

NAME	CHANGE	EXISTING
▼ Pipelines		
 IncrementalCopyPipeline	(New)	-
▼ Datasets		
 AzureSynapseAnalyticsTab...	(New)	-
 WatermarkDataset	(New)	-
 SourceDataset	(New)	-
 SinkDataset	(New)	-

22. Wait until you see the Successfully published message. To see the notifications, click the Show Notifications link. Close the notifications window by clicking X.



Run the pipeline

On the toolbar for the pipeline, click Add trigger, and click Trigger Now.

In Monitor tab you can monitor the pipeline run.

The screenshot displays the Azure Data Factory Monitor interface. On the left sidebar, the 'Runs' section is expanded, and 'Pipeline runs' is selected. The main area shows the 'IncrementalCopyPipeline' in the 'List' view. A red box highlights the pipeline's toolbar, which includes buttons for 'Rerun', 'Rerun from activity', 'Rerun from failed activity', 'Refresh', and 'Update pipeline'. Below the pipeline view, the 'Activity runs' section is shown for Pipeline run ID 8e5b2032-95ee-4e02-98e9-a2a739625163. A red box highlights the activity runs table, which shows three successful runs.

IncrementalCopyPipeline

Buttons: Rerun, Rerun from activity, Rerun from failed activity, Refresh, Update pipeline

Activity runs

Pipeline run ID 8e5b2032-95ee-4e02-98e9-a2a739625163

Activity name	Activity type	Run start	Duration	Status
StoredProceduretoWriteWate...	Stored procedure	12/14/2022, 5:34:05 PM	00:00:02	Succeeded
StoredProceduretoWriteWate...	Stored procedure	12/14/2022, 5:34:04 PM	00:00:09	Succeeded
IncrementalCopyActivity	Copy data	12/14/2022, 5:33:53 PM	00:00:10	Succeeded

Review the results

Create a SQL Script in SQL Database Query Editor, run the following queries against the target database to verify that the data was copied from source tables to destination tables:

Query: select * from customer_table_dest

Output:

PersonID	Name	LastModifytime
1	John	2017-09-01 00:56:00.000
2	Mike	2017-09-02 05:23:00.000
3	Alice	2017-09-03 02:36:00.000
4	Andy	2017-09-04 03:21:00.000
5	Anny	2017-09-05 08:06:00.000

Query: select * from project_table_dest

Output:

Project	Creationtime
project1	2015-01-01 00:00:00.000
project2	2016-02-02 01:23:00.000
project3	2017-03-04 05:16:00.000

Query: select * from watermarktable

Output:

TableName	WatermarkValue
customer_table	2017-09-05 08:06:00.000
project_table	2017-03-04 05:16:00.000

Notice that the watermark values for both tables were updated.

Add more data to the source tables

Run the following query against the source **SQL database** to update an existing row in **customer_table**. Insert a new row into **project_table**.

```
UPDATE customer_table  
SET [LastModifytime] = '2017-09-08T00:00:00Z', [name]='NewName' where [PersonID] = 3
```

```
INSERT INTO project_table  
(Project, Creationtime)  
VALUES  
('NewProject','10/1/2017 0:00:00 AM');
```

Rerun the pipeline

On the toolbar for the pipeline, click Add trigger, and click Trigger Now.

Review the final results

Create a SQL Script in Synapse under develop hub, run the following queries against the target SQL Pool to verify that the data was copied from source tables to destination tables:

Query: select * from customer_table_dest

Output:

PersonID	Name	LastModifytime
1	John	2017-09-01 00:56:00.000
2	Mike	2017-09-02 05:23:00.000
3	NewName	2017-09-08 00:00:00.000
4	Andy	2017-09-04 03:21:00.000
5	Anny	2017-09-05 08:06:00.000

Query: select * from project_table_dest

Output:

Project	Creationtime
project1	2015-01-01 00:00:00.000
project2	2016-02-02 01:23:00.000

Project	Creationtime
project3	2017-03-04 05:16:00.000
NewProject	2017-10-01 00:00:00.000

Query: select * from watermarktable

Output:

TableName	WatermarkValue
customer_table	2017-09-08 00:00:00.000
project_table	2017-10-01 00:00:00.000

Notice that the watermark values for both tables were updated.