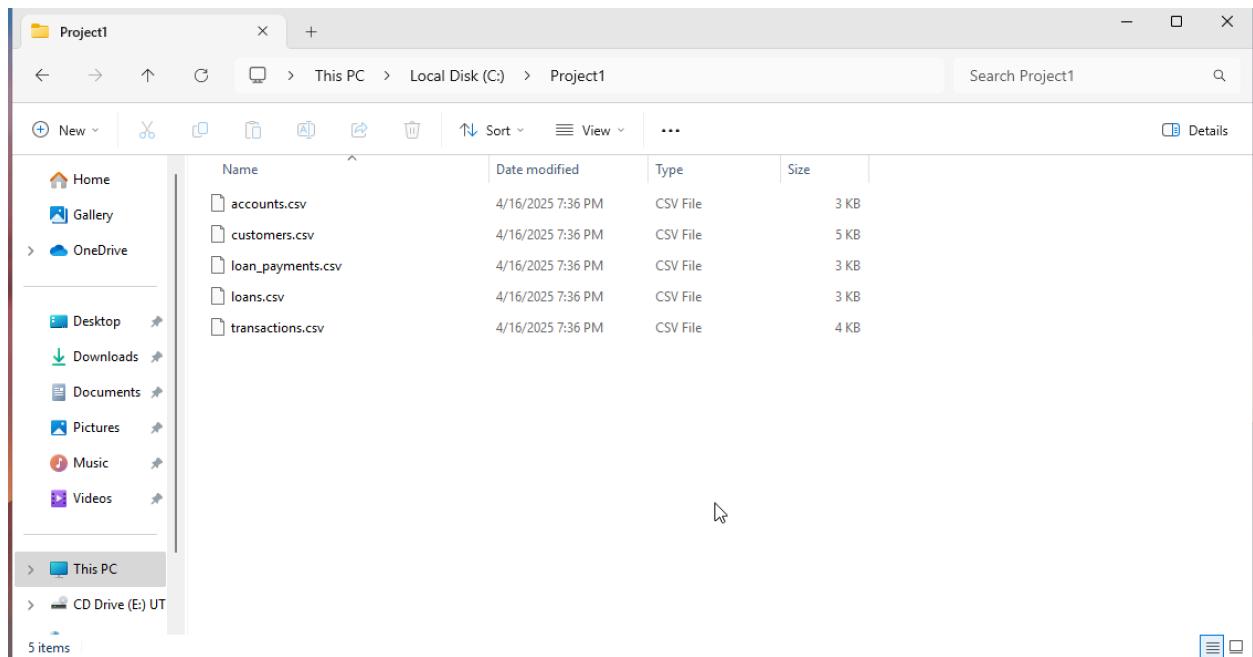


Architecture diagram for the project



Files downloaded from kaggle

The screenshot shows the 'Integration runtime setup' page in the Azure Data Factory interface. On the left, a sidebar lists various settings like General, Connections, Integration runtimes, and Source control. The main area is titled 'Integration runtimes' and shows a single item named 'AutoResolveIntegrationRuntime'. To the right, there's a section for 'Network environment' with two options: 'Azure' (selected) and 'Self-Hosted'. Below that is a section for 'External Resources' with 'Linked Self-Hosted' listed. At the bottom are 'Continue', 'Back', and 'Cancel' buttons.

SelfHosted Integration runtime for the local machine

The screenshot shows the 'Secrets' page in the Azure Key Vault interface. The left sidebar includes 'Overview', 'Activity log', 'Access control (IAM)', and 'Tags'. The main table displays a single secret named 'lappy password' with type 'String', status 'Enabled', and no expiration date. Action buttons at the top include 'Generate/Import', 'Refresh', 'Restore Backup', 'Manage deleted secrets', and 'View sample code'.

Secret key in keyvault for the laptop password

Edit linked service

File system [Learn more](#)

Name *

FileServer1

Description

Connect via integration runtime * ⓘ

selfHostedIR



Host * ⓘ

C:\Project1

User name *

Harsha Vardhan

Password

Azure Key Vault

AKV linked service * ⓘ

AzureKeyVault1



Secret name * ⓘ

lappypassword



Edit

Secret version ⓘ

96c7c0eb3155425f8f9e1415f52f5397 (Current version)



Edit

Connection successful

Test connection

Save

Cancel

Linked service to copy the backend data

The screenshot displays three stacked configuration panels for a Data Factory pipeline activity.

Top Panel: Pipeline Activity Configuration

```

graph TD
    GetMetadata[Get Metadata] --> ForEach1[ForEach1]
    ForEach1 --> CopyData1[Copy data1]
    
```

This panel shows a "Get Metadata" activity followed by a "ForEach" loop. Inside the loop, there is a single "Copy data1" activity.

Middle Panel: Source Dataset Configuration

General Tab:

- Dataset: DelimitedText3
- Field list: Argument, Child items
- Filter by last modified: Start time (UTC), End time (UTC)
- Skip line count: [empty input]

Source Tab (selected):

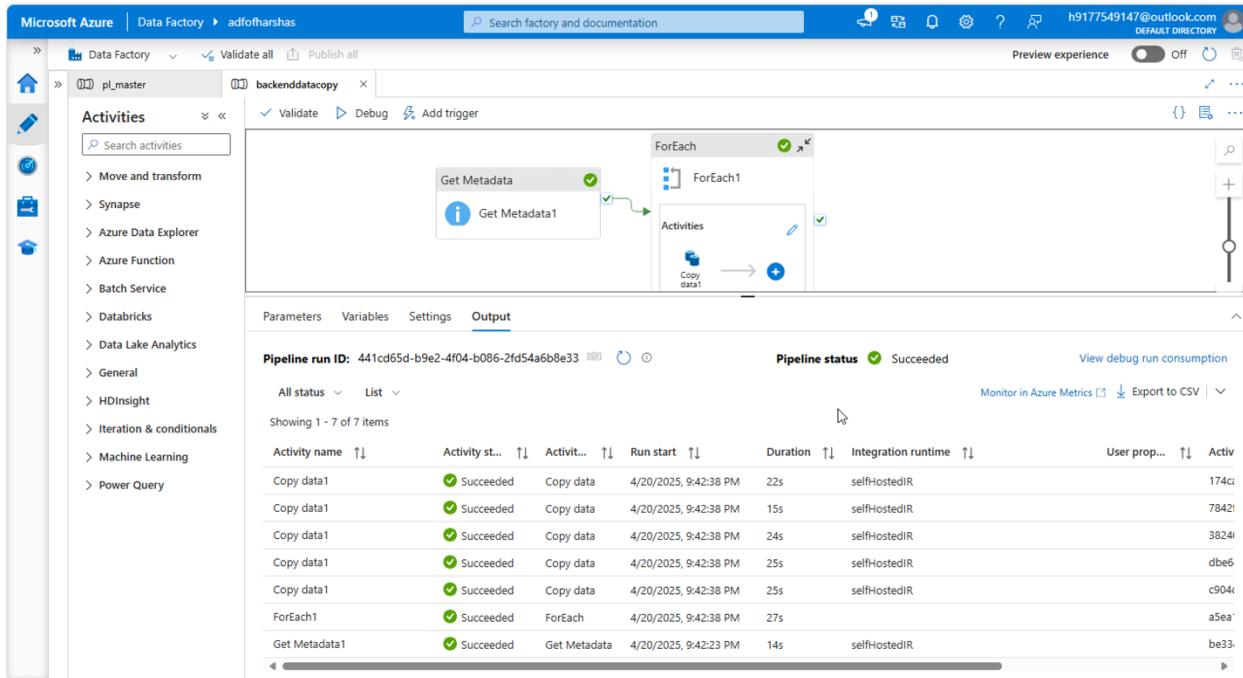
- Source dataset: DelimitedText1
- Dataset properties:**

| Name | Value | Type |
|----------|--------------|--------|
| filename | @item().name | string |
- File path type: File path in dataset (selected)
- Start time (UTC) and End time (UTC) inputs
- Filter by last modified: [empty input]
- Recursively: checked

Bottom Panel: Connection Configuration

Connection Tab (selected):

- Linked service: FileServer1
- Integration runtime: selfHostedIR
- File path: C:\Project1\ / Directory / @dataset().filename
- Compression type: No compression
- Column delimiter: Comma (,)
- Row delimiter: Default (\r,\n, or \r\n)
- Encoding: Default(UTF-8)
- Quote character: Double quote ("")
- Escape character: Backslash (\)



Pipeline to get the data from laptop to ADLS gen 2

The screenshot shows the Azure Storage Explorer interface. It displays a container named 'project1' under the 'adlsgen2h' storage account. The 'Overview' tab is selected. The container has two blobs: 'bronze' and 'silver'. The blob details table shows the following information:

| Name | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|--------|-----------------------|-------------|----------------|------------|------|-------------|
| bronze | 4/18/2025, 4:32:22 PM | | | Block blob | - | --- |
| silver | 4/18/2025, 4:32:28 PM | | | Block blob | - | --- |

Containers created inside the ADLS to store the files

The screenshot shows the Azure Storage Explorer interface, similar to the previous one but at a deeper level. It displays a container named 'project1/bronze' under the 'adlsgen2h' storage account. The 'Overview' tab is selected. The container contains several CSV files: 'accounts.csv', 'customers.csv', 'loan_payments.csv', 'loans.csv', and 'transactions.csv'. The blob details table shows the following information:

| Name | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|-------------------|-----------------------|----------------|----------------|------------|----------|-------------|
| accounts.csv | 4/20/2025, 9:42:58 PM | Hot (Inferred) | | Block blob | 2.28 KiB | Available |
| customers.csv | 4/20/2025, 9:43:00 PM | Hot (Inferred) | | Block blob | 4.5 KiB | Available |
| loan_payments.csv | 4/20/2025, 9:42:51 PM | Hot (Inferred) | | Block blob | 2.55 KiB | Available |
| loans.csv | 4/20/2025, 9:43:00 PM | Hot (Inferred) | | Block blob | 2.29 KiB | Available |
| transactions.csv | 4/20/2025, 9:43:00 PM | Hot (Inferred) | | Block blob | 3.43 KiB | Available |

These are files loaded by successful run of the pipeline.



Here is the dataflow design for step_2

The screenshot shows the Azure Data Factory pipeline run details for step_2. The pipeline run ID is 37a74090-960b-4e45-ac72-34bb07b408cf. The pipeline status is Succeeded. The table below lists the activity logs for Data flow1:

| Activity name | Activity st... | Activit... | Run start | Duration | Integration runtime | User prop... | Acti |
|---------------|----------------|------------|-----------------------|----------|---|--------------|------|
| Data flow1 | Succeeded | Data flow | 4/19/2025, 7:03:21 PM | 1m 23s | AutoResolveIntegrationRuntime (East US) | | 54e |

The pipeline run for step 2

Source settings Source options Projection Optimize Inspect Data preview

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

Description Import data from AzureDataLakeStorage1 [Reset](#)

Source type * Dataset Inline

Inline dataset type * DelimitedText

Linked service * AzureDataLakeStorage1 [Test connection](#) [Edit](#) [New](#)

Skip line count

Sampling * Enable Disable

We are giving the source from our ADLS gen 2 from bronze container

Source settings **Source options** Projection Optimize Inspect Data preview

File settings

File mode File Wildcard

File path * project1 / bronze / accounts.csv [Browse](#)

Allow no files found

Change data capture

Compression type No compression

Encoding Default(UTF-8)

Column delimiter Comma (,)

Row delimiter Default (\r,\n, or \v\n)

Quote character Double quote ("")

Escape character Backslash (\)

Selecting the accounts.csv file from bronze layer.

Source settings Source options **Projection** Optimize Inspect Data preview

[Import schema](#) [Clear schema](#) [Schema options](#)

| Column name | Type | Format |
|--------------|------------|----------------|
| account_id | 12s short | Specify format |
| customer_id | 12s short | Specify format |
| account_type | abc string | Specify format |
| balance | 1.2 double | Specify format |

Then import schema for the file

Filter settings Optimize Inspect Data preview

Output stream name * [Learn more](#)

Description

Incoming stream *

Filter on *

Now use filter option to check main field like account_id & customer_id if they are null we remove the entire row

Derived column's settings Optimize Inspect Data preview

Output stream name * [Learn more](#)

Description

Incoming stream *

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

Columns [①](#)

| Column | Expression |
|---------------------------------------|--|
| <input type="checkbox"/> account_type | <input type="text" value="iif(isNull(account_type), 'NA', account_type)"/> abc + ✖ |

Then we give some values to the missing values in the row we are using derived column function for that with expression “`iif(isNull(account_type), 'NA', account_type)`”.

Window settings Optimize Inspect Data preview

Output stream name * removeDuplicate1 ? Help Learn more [🔗](#)

Description Aggregates data based on a window and joins with original data [Reset](#)

Incoming stream * derivedColumn1

1. Over 2. Sort 3. Range by 4. Window columns

derivedColumn1's column Name as

| | | |
|------------------|--------------|--------------------------|
| 12s account_id | account_id | + Delete |
| 12s customer_id | customer_id | + Delete |
| abc account_type | account_type | + Delete |
| 1.2 balance | balance | + Delete |

Window settings Optimize Inspect Data preview

Output stream name * removeDuplicate1 ? Help Learn more [🔗](#)

Description Aggregates data based on a window and joins with original data [Reset](#)

Incoming stream * derivedColumn1

1. Over 2. Sort 3. Range by 4. Window columns

derivedColumn1's column Order Nulls first

| | | |
|------------------|-----------|--|
| 12s account_id | Ascending | <input checked="" type="checkbox"/> + Delete |
| 12s customer_id | Ascending | <input checked="" type="checkbox"/> + Delete |
| abc account_type | Ascending | <input checked="" type="checkbox"/> + Delete |
| 1.2 balance | Ascending | <input checked="" type="checkbox"/> + Delete |

Sort in window function

Window settings Optimize Inspect Data preview

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

Description ⏪ Reset

Incoming stream * ▾

1. Over 2. Sort 3. Range by 4. Window columns 4. Window columns

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

| <input type="checkbox"/> Column | Expression |
|---------------------------------|---|
| <input type="checkbox"/> rowcol | <input type="text" value="rowNumber()"/> 123 + ↗ Delete ↗ |

Window column expression of window function

Filter settings Optimize Inspect Data preview

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

Description ⏪ Reset

Incoming stream * ▾

Filter on * [X](#)

Now to remove the duplicate values we use filtering function if select `rowcol==1` and only let the fields that pass-through filter.

Select settings Optimize Inspect Data preview

Output stream name * removingcol1 [Learn more](#)

Description Renaming filter1 to removingcol1 with columns 'account_id, customer_id, account_type, balance'

Incoming stream * filter1

Options

- Skip duplicate input columns
- Skip duplicate output columns

Input columns *

| <input type="checkbox"/> | filter1's column | ▼ | Name as |
|--------------------------|------------------|-------------------|--------------|
| <input type="checkbox"/> | 12s account_id | ▼ | account_id |
| <input type="checkbox"/> | 12s customer_id | ▼ | customer_id |
| <input type="checkbox"/> | abc account_type | ▼ | account_type |
| <input type="checkbox"/> | 12 balance | ▼ | balance |

4 mappings: 1 column(s) fr

[Auto mapping](#) [Reset](#) [+ Add mapping](#) [Delete](#)

We use select function and remove rowcol and let only original rows of the table to pass

Alter row settings Optimize Inspect Data preview

Output stream name * alterRow1 [Learn more](#)

Description Add expressions to alter rows

Incoming stream * removingcol1

Alter row conditions * [①](#)

| | | | |
|-----------------------------------|------|-------------------|-------------------|
| Upsert if | 1==1 | + | - |
|-----------------------------------|------|-------------------|-------------------|

We give alter row function and give condition upsert to create delta format file.

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

Folder path * project1 / concat('silver','accounts') [abc](#) [Browse](#)

Compression type No compression

Vacuum 0

Table action None Overwrite Truncate

Update method [①](#)

- Allow insert
- Allow delete
- Allow upsert
- Allow update

We give the path to set the file to be saved.

The screenshot shows the Azure Storage Explorer interface. At the top, there's a navigation bar with 'Home > adlsgen2h | Containers >'. Below it, the 'project1' container is selected. The 'Overview' tab is active, displaying blob details. A table lists blobs with columns: Name, Modified, Access tier, Archive status, Blob type, Size, and Lease state. The table contains several entries, including '_delta_log', '_temporary', and several part files. Below the table is a search bar and a 'Show deleted objects' toggle. On the left, there's a sidebar with 'Diagnose and solve problems', 'Access Control (IAM)', and 'Settings' options.

The bottom half of the screenshot shows a Power BI data flow diagram. It starts with a 'source1' icon, followed by a 'renamingcolumn...' step, a 'hashkeygenerat...' step, and a 'join1' step. A 'target' icon is connected to the 'join1' step. From 'join1', the flow splits into two parallel paths. The top path goes through an 'insert' step, an 'auditcolumns' step, and an 'insertsink' step. The bottom path goes through an 'update' step, an 'updateauditcol...' step, an 'alterRow1' step, and an 'updatesink' step. A 'target' icon is also connected to the bottom path. A 'dotted line' labeled 'target' connects the 'join1' step to the target icon. At the bottom left, there's a 'Add Source' button with a dropdown arrow.

SCD type 1 transformation

Source settings Source options Projection Optimize Inspect Data preview

Output stream name * [Learn more](#)

Description [Reset](#)

Source type * Dataset **Inline**

Inline dataset type * Delta [Edit](#)

Linked service * [Test connection](#) [Edit](#) [New](#)

Sampling * Enable Disable

The source details are here we are selecting the delta file format from ADLS gen 2

Source settings **Source options** Projection Optimize Inspect Data preview

Folder path * / [Browse](#)

Allow no files found

Compression type

Time travel * Disable Query by timestamp Query by version

Setting the folder path

Select settings Optimize Inspect Data preview

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

Description Renaming source1 to renamingcolumns with columns 'src_account_id', 'src_customer_id', 'src_account_type', [Reset](#)

Incoming stream * source1

Options Skip duplicate input columns [①](#)
 Skip duplicate output columns [①](#)

Input columns *

Auto mapping [①](#) [Reset](#) [+ Add mapping](#) [Delete](#) 1 mappings: All inputs mapped

| | source1's column | Name as |
|--|------------------|---------------------|
| | 1==1 | concat("src_",\$\$) |

Derived column's settings Optimize Inspect Data preview

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

Description Creating/updating the columns 'src_account_id', 'src_customer_id', 'src_account_type', 'src_balance', [Reset](#)

Incoming stream * renamingcolumns [+ Add](#) [Clone](#) [Delete](#) [Open expression builder](#)

Columns * [①](#)

| Column | Expression |
|-------------|--|
| src_hashkey | crc32(concat(toString(src_account_id),toString(sr... 121)) |

Source settings Source options Projection Optimize Inspect Data preview

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

Description Import data from AzureSqlDatabase1 [Reset](#)

Source type *

| | |
|--|--|
| | |
|--|--|

Inline dataset type * [Azure SQL Database](#)

Linked service * AzureSqlDatabase1 [Test connection](#) [Edit](#) [New](#)

Sampling * [①](#) Enable Disable

Input

Table Query Stored procedure

Query * ①

```
select account_id , hashkey from
dbo.accountsccdtype1
```



Incremental column ①



Isolation level ①

Read uncommitted



We give this query to select the account_id & haskey from the table.

Output stream name *

join1

[Learn more](#)

Description

Left outer join on 'haskkeygeneration' and
'target'

Reset

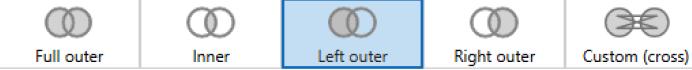
Left stream *

haskkeygeneration

Right stream *

target

Join type *



Use fuzzy matching ①



Join conditions *

Left: haskkeygeneration's column

Right: target's column

12s src_account_id

==

12s account_id



We are using left join to join the source and target

Conditional split settings Optimize Inspect Data preview

Output stream name * [Learn more](#)

Description

Incoming stream *

Split on First matching condition All matching conditions

Split condition

| Stream names | Condition |
|--------------|--|
| insert | isNull(account_id) |
| update | src_account_id==account_id && src_hashkey!=hashkey |

We are giving the conditions to set the insert and update functions

For insert we are giving : isnull(account_id)

For update we are giving: src_account_id==account_id && src_hashkey!=hashkey

Derived column's settings Optimize Inspect Data preview

Output stream name * [Learn more](#)

Description

Incoming stream *

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

Columns * ①

| Column | Expression |
|------------------|--------------------|
| src_created_by | "dataflow" |
| src_created_date | currentTimestamp() |
| src_updated_by | "dataflow" |
| src_updated_date | currentTimestamp() |

Now we are seeing the derived columns functions on insert side

Derived column's settings Optimize Inspect Data preview

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

Description Creating/updating the columns
'src_account_id, src_customer_id,
src_account_type, src_balance,

Incoming stream * split1@update

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

Columns * [①](#)

| <input type="checkbox"/> Column | Expression |
|---|--------------------|
| <input type="checkbox"/> src_updated_by | "dataflow-updated" |
| <input type="checkbox"/> src_updated_date | currentTimestamp() |

This is derived columns on the update side

Sink Settings Errors Mapping Optimize Inspect Data preview

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

Description Add sink dataset

Incoming stream * auditcolumns

Sink type *

| | | |
|---------|--------|-------|
| | | |
| Dataset | Inline | Cache |

Inline dataset type * Azure SQL Database

Linked service * AzureSqlDatabase1 [Test connection](#) [Edit](#) [New](#)

Options

Allow schema drift [①](#)

Validate schema [①](#)

Here is the sink settings we are going to save the scd type 1 transformed data to sql database.

Sink Settings Errors Mapping Optimize Inspect Data preview

Schema name * Refresh

Table name *

Table action None Recreate table Truncate table

Update method Allow insert
 Allow delete
 Allow upsert
 Allow update

Use tempdb

Interim table schema Refresh

Pre SQL scripts List of scripts Custom expression
 +
 [Empty box]

Post SQL scripts List of scripts Custom expression
 +
 [Empty box]

Select the dbo schema and select the table required and click on allow insert for insertion

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

Options Skip duplicate input columns
 Skip duplicate output columns

Auto mapping
 + Add mapping
 Delete
 Reset
 Import schema
 View schema
 9 mappings: All outputs mapped

| Input columns | Output columns |
|----------------------|------------------|
| 12s src_account_id | 12s account_id |
| 12s src_customer_id | 12s customer_id |
| abc src_account_type | abc account_type |
| 1.2 src_balance | 1.2 balance |
| 12l src_hashkey | 12l hashkey |
| abc src_created_by | abc created_by |
| ⌚ src_created_date | ⌚ created_date |
| abc src_updated_by | abc updated_by |
| ⌚ src_updated_date | ⌚ updated_date |

And then set the mapping for the sql table to allow data to be inserted properly

Alter row settings Optimize Inspect Data preview

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

Alter row conditions * [①](#)

* Update if [+](#)

Update condition for updates in scd type 1

Sink Settings Errors Mapping Optimize Inspect Data preview

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

Sink type * Dataset Cache

Inline dataset type * [Azure SQL Database](#)

Linked service * [Test connection](#) [Edit](#) [New](#)

Options Allow schema drift [①](#)
 Validate schema [①](#)

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

Schema name * Refresh

Table name *

Table action None Recreate table Truncate table

Update method Allow insert
 Allow delete
 Allow upsert
 Allow update

Skip writing key columns

Key columns * List of columns Custom expression

Here is the update side sink details

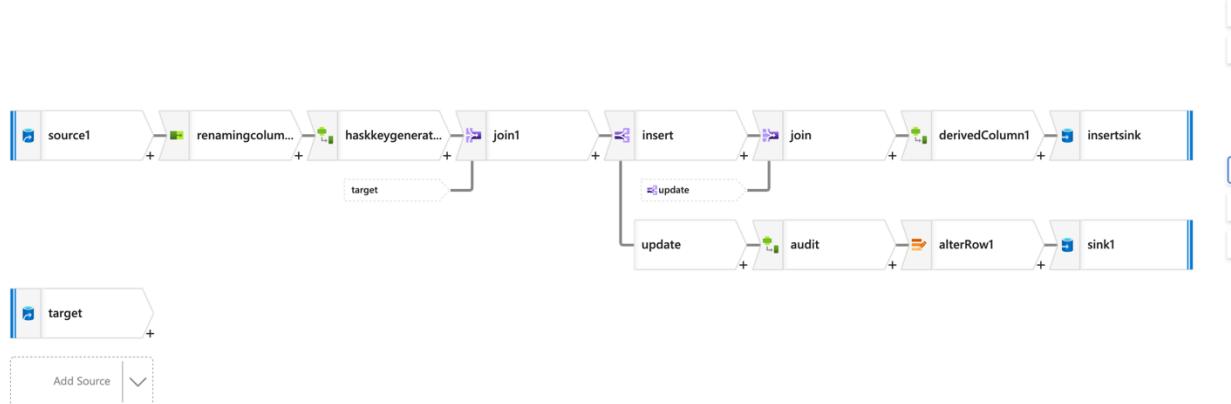
Sink Settings Errors **Mapping** Optimize Inspect Data preview

Options Skip duplicate input columns
 Skip duplicate output columns

Auto mapping Add mapping Delete Reset Import schema View schema 7 mappings: 2 column(s) from the output schema left unmapped

| Input columns | Output columns |
|---|---|
| <input type="text" value="12s src_account_id"/> | <input type="text" value="12s account_id"/> |
| <input type="text" value="12s src_customer_id"/> | <input type="text" value="12s customer_id"/> |
| <input type="text" value="abc src_account_type"/> | <input type="text" value="abc account_type"/> |
| <input type="text" value="1.2 src_balance"/> | <input type="text" value="1.2 balance"/> |
| <input type="text" value="12l src_hashkey"/> | <input type="text" value="12l hashkey"/> |
| <input type="text" value="abc src_updated_by"/> | <input type="text" value="abc updated_by"/> |
| <input type="text" value="⌚ src_updated_date"/> | <input type="text" value="⌚ updated_date"/> |

Update side mapping



SCD type 2 transformation

Union settings Optimize Inspect Data preview

Output stream name * [Learn more](#)

Description [Reset](#)

Incoming stream *

Union by * Name Position

Union with * [+](#) [-](#)

Derived column's settings Optimize Inspect Data preview

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

Description Creating/updating the columns
'src_customer_id, src_first_name,
src_last_name, src_address, src_city,'

Incoming stream * split1@update

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

Columns * [①](#)

| <input type="checkbox"/> Column | Expression | + | Delete |
|---|--------------------|-------------------|------------------------|
| <input type="checkbox"/> src_updated_by | "dataflow-updated" | abc | + |
| <input type="checkbox"/> src_updated_date | currentTimestamp() | ⌚ | + |
| <input type="checkbox"/> src_isActive | 0 | 123 | + |



Here is the update side derived column

Derived column's settings Optimize Inspect Data preview

Output stream name * derivedColumn1 [Learn more](#)

Description Creating/updating the columns
'src_customer_id, src_first_name,
src_last_name, src_address, src_city,'

Incoming stream * join

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

Columns * [①](#)

| <input type="checkbox"/> Column | Expression | + | Delete |
|--|--------------------|-------------------|------------------------|
| <input type="checkbox"/> src_createdby | "dataflow" | abc | + |
| <input type="checkbox"/> src_createddate | currentTimestamp() | ⌚ | + |
| <input type="checkbox"/> src_updatedby | "dataflow" | abc | + |
| <input type="checkbox"/> src_updateddate | currentTimestamp() | ⌚ | + |
| <input type="checkbox"/> src-isactive | 1 | 123 | + |



This is the insert side derived columns currenttimestamp() gives todays date and time at the time of insertion

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

Skip duplicate input columns Skip duplicate output columns

Auto mapping 13 mappings: All outputs mapped

| Input columns | | Output columns | |
|---------------------|---|------------------|---|
| 12s src_customer_id | → | 12s customer_id | <input type="button" value="Delete"/> + |
| abc src_first_name | → | abc first_name | <input type="button" value="Delete"/> + |
| abc src_last_name | → | abc last_name | <input type="button" value="Delete"/> + |
| abc src_address | → | abc address | <input type="button" value="Delete"/> + |
| abc src_city | → | abc city | <input type="button" value="Delete"/> + |
| abc src_state | → | abc state | <input type="button" value="Delete"/> + |
| abc src_zip | → | abc zip | <input type="button" value="Delete"/> + |
| 12l src_hashkey | → | 12l hashkey | <input type="button" value="Delete"/> + |
| abc src_createdby | → | abc created_by | <input type="button" value="Delete"/> + |
| src_createddate | → | src_created_date | <input type="button" value="Delete"/> + |
| abc src_updatedby | → | abc updated_by | <input type="button" value="Delete"/> + |
| src_updateddate | → | src_updated_date | <input type="button" value="Delete"/> + |
| 123 src-isactive | → | 123 isActive | <input type="button" value="Delete"/> + |

This is the mapping on the insert side of the pipeline

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

Schema name *

Table name *

Table action None Recreate table Truncate table

Update method Allow insert Allow delete Allow upsert Allow update

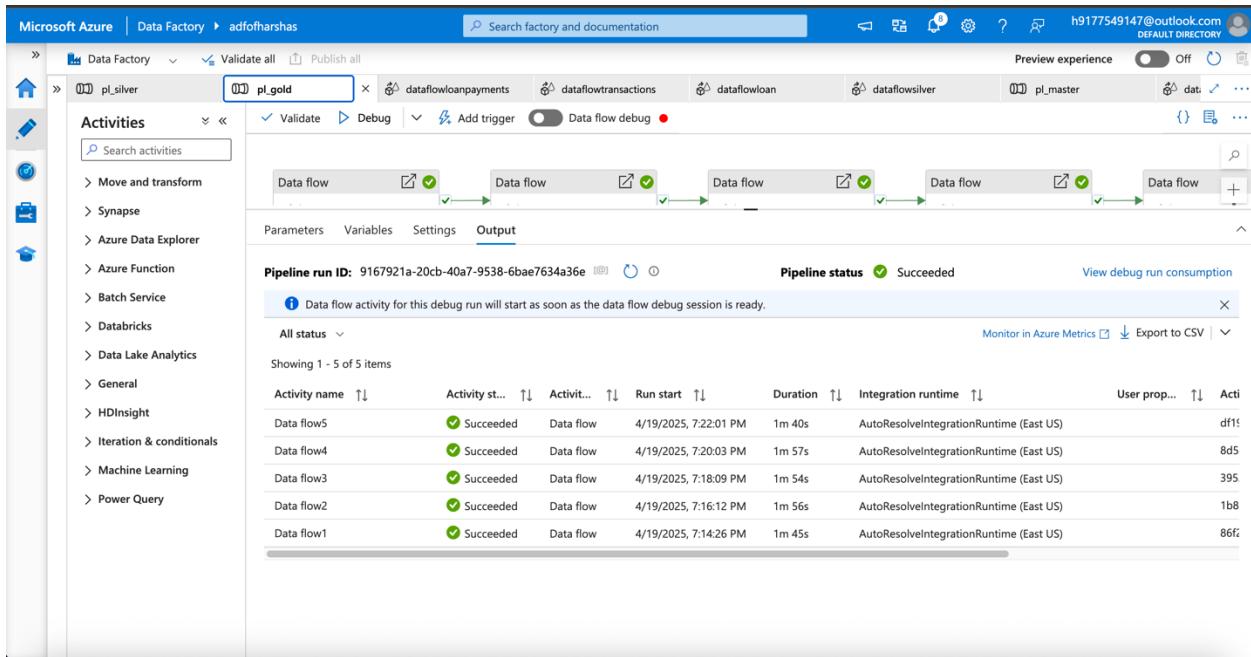
We are allowing the insert function for the insert side of the pipeline

The screenshot shows the 'Mapping' tab of a data integration interface. At the top, there are several options: 'Auto mapping' (unchecked), 'Add mapping' (+), 'Delete' (-), 'Reset' (refresh), 'Import schema' (import icon), and 'View schema' (view icon). A status message indicates '5 mappings: 8 column(s) from the output schema left unmapped'. Below this, there are two columns: 'Input columns' and 'Output columns'. The 'Input columns' list contains five items: 'src_customer_id', 'src_hashkey', 'src_updated_by', 'src_updated_date', and 'src_isActive'. The 'Output columns' list also contains five items: 'customer_id', 'hashkey', 'updated_by', 'updated_date', and 'isActive'. Each input item has a dropdown arrow pointing to its corresponding output item. To the right of each output item are three icons: a trash can, a plus sign, and a refresh symbol.

We only update haskey updated_by updated_date and isActive fields only in the update side

The screenshot shows the 'Settings' tab of a data integration interface. It includes fields for 'Schema name' (set to 'dbo'), 'Table name' (set to 'customersscdtype2'), and 'Table action' (set to 'None'). Under 'Update method', the 'Allow update' checkbox is checked. In the 'Skip writing key columns' section, the checkbox is unchecked. The 'Key columns' section is set to 'List of columns' and lists 'customer_id' and 'hashkey' as key columns, each with a '+' and a trash can icon to the right.

We are allowing update and are giving customer_id and hashkey for the key columns



Pipelines successfully run

Run Cancel query Save query Export data as Show only Editor

```
1  SELECT TOP (1000) * FROM [dbo].[transactionsscdtype2]
```

Results Messages

Search to filter items...

| transaction_id | account_id | transaction_date | transaction_amount | transaction_type | hash |
|----------------|------------|-----------------------------|--------------------|------------------|--------|
| 65 | 69 | 2024-03-05T00:00:00.0000000 | 250.00 | Deposit | 3733\$ |
| 53 | 86 | 2024-02-22T00:00:00.0000000 | 150.00 | Deposit | 3828€ |
| 85 | 65 | 2024-03-25T00:00:00.0000000 | 250.00 | Deposit | 1442: |
| 31 | 71 | 2024-01-31T00:00:00.0000000 | 100.50 | Deposit | 4779: |
| 78 | 4 | 2024-03-18T00:00:00.0000000 | 275.75 | Withdrawal | 2544\$ |

SCD type 2 data stored successfully in SQL database

Query 1 × Query 4 ×

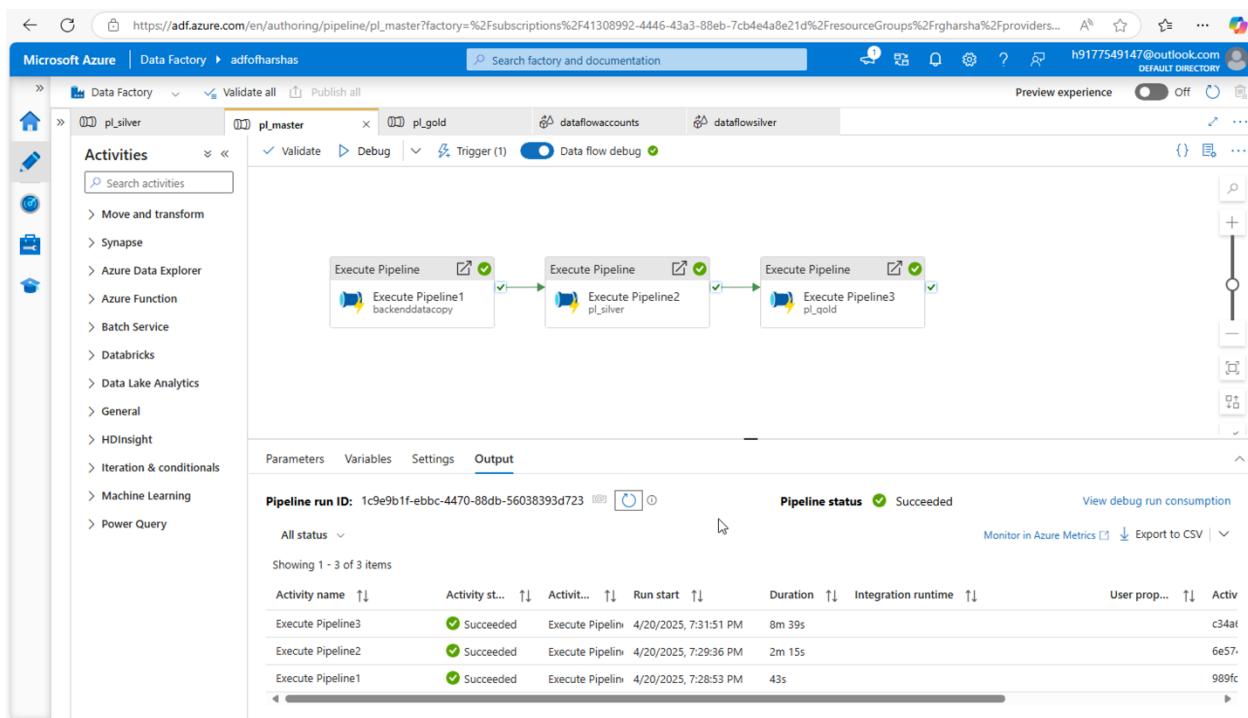
Cancel query Show only Editor

```
1  SELECT TOP (1000) * FROM [dbo].[accountsscdtype1]
```

Results Messages

| hashkey | created_by | created_date | updated_by | updated_date |
|------------|------------|-----------------------------|------------------|-----------------------------|
| 2662522274 | dataflow | 2025-04-20T02:12:37.6510000 | dataflow-updated | 2025-04-21T02:32:33.1570000 |
| 2464207871 | dataflow | 2025-04-20T02:12:37.6510000 | dataflow-updated | 2025-04-21T05:04:01.8550000 |
| 1122301726 | dataflow | 2025-04-20T02:12:37.6510000 | dataflow-updated | 2025-04-21T02:32:33.1570000 |
| 630820887 | dataflow | 2025-04-20T02:12:37.6510000 | dataflow-updated | 2025-04-21T02:32:33.1570000 |

SCD type 1 data successfully stored in SQL database.



Triggers

To execute a pipeline set the trigger. Triggers represent a unit of processing that determines when a pipeline execution needs to be kicked off.

[+ New](#) [⟳ Refresh](#)

Filter by name

Annotations : Any

Showing 1 - 1 of 1 items

| Name ↑↓ | Type ↑↓ | Status ↑↓ | Related ↑↓ | Annotations ↑↓ |
|-------------------|----------|-----------|------------|----------------|
| pl_master_trigger | Schedule | ✓ Started | 1 | |