Exercise: Join and transform shared and current dataset with Mapping Flow in Azure Data Factory

Transform data using mapping data flow

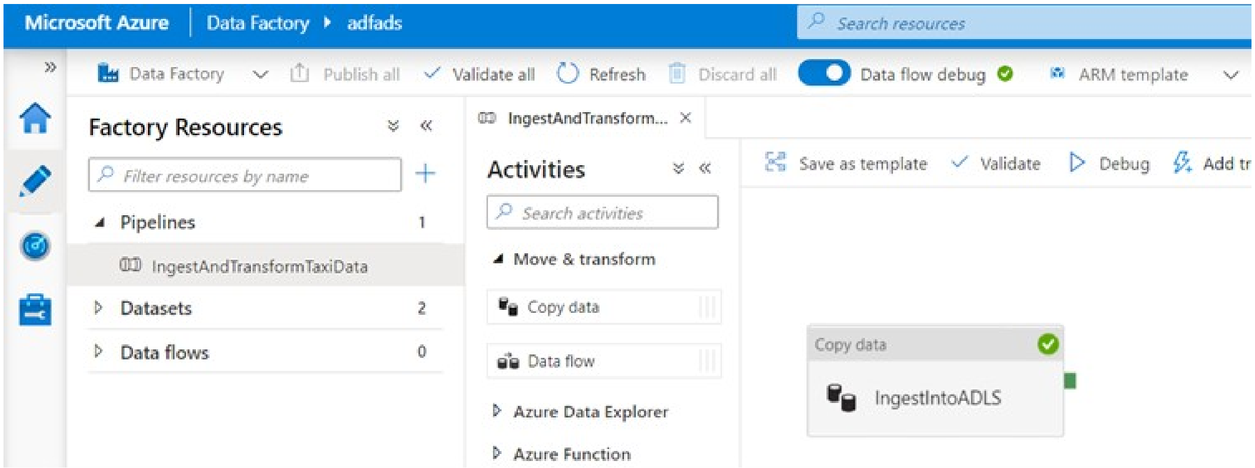
Now that you have successfully copied data into Azure Data Lake Storage, it is time to join and aggregate that data into a data warehouse.

We will use mapping data flow, Azure Data Factory's visually designed transformation service. Mapping data flows allow users to develop transformation logic code-free and execute them on spark clusters managed by the ADF service.

The data flow created in this step inner joins the 'TripDataCSV' dataset created in the previous section with the 'TripFares.csv' file shared via Azure Data Share. based on four key columns. Then the data gets aggregated based upon column payment_type to calculate the average of certain fields and written in a Azure Synapse Analytics table.

Add a data flow activity to your pipeline

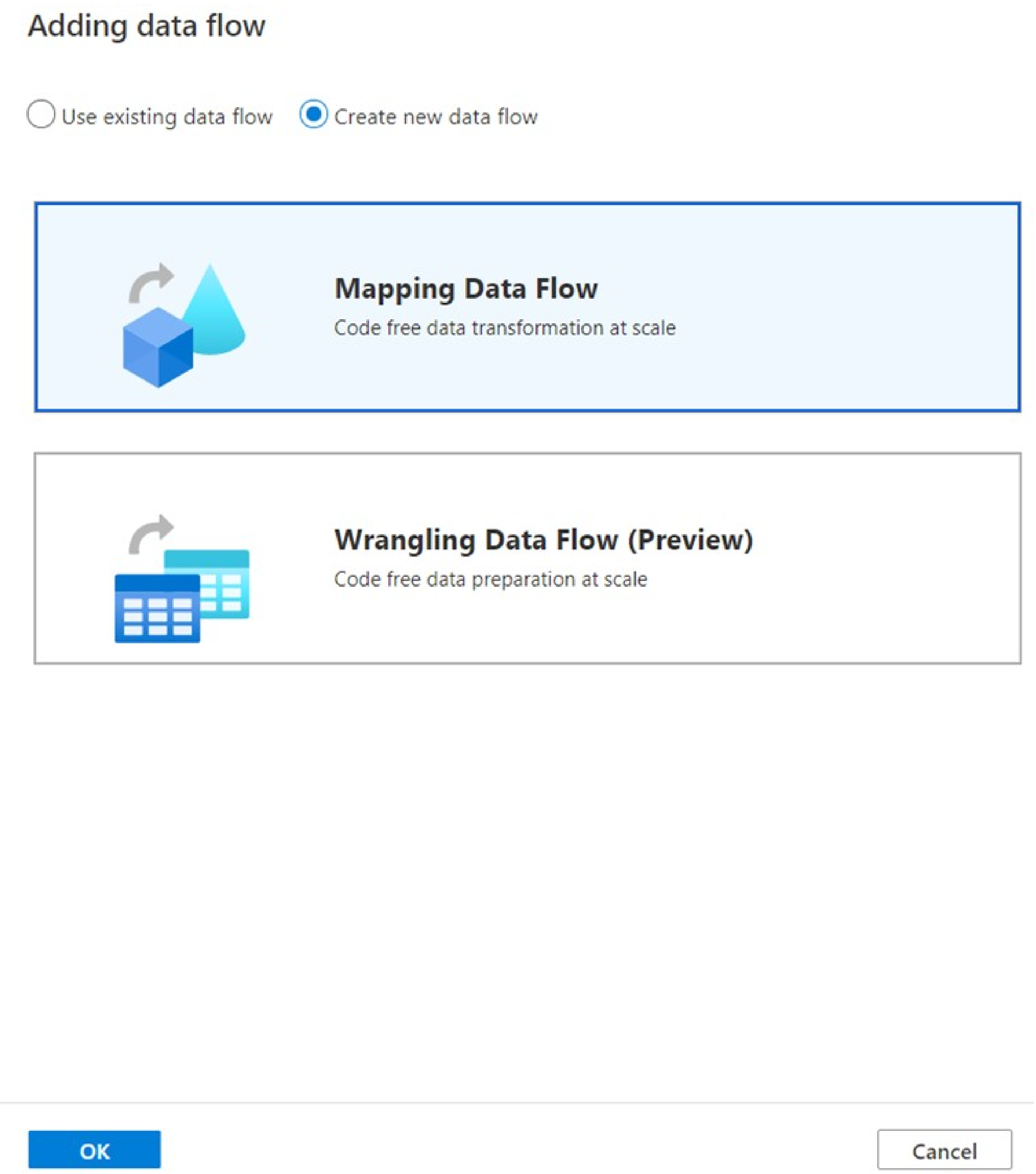
In the activities pane of the pipeline canvas, open the Move and Transform accordion and drag the Data flow activity onto the canvas.



Move and Transform

In the side pane that opens, select Create new data flow and choose Mapping data flow.

Click OK.



Adding data flow

○ Use existing data flow ● Create new data flow

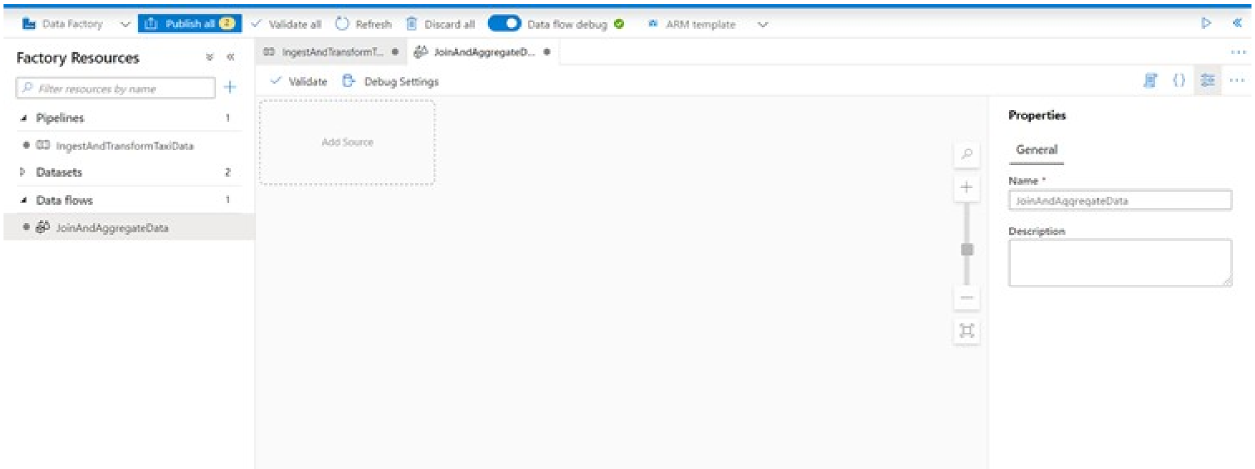
Mapping Data Flow
Code free data transformation at scale

Wrangling Data Flow (Preview)
Code free data preparation at scale

Create new DataFlow

You'll be directed to the data flow canvas where you will be building your transformation logic.

In the general tab, name your data flow 'JoinAndAggregateData'.

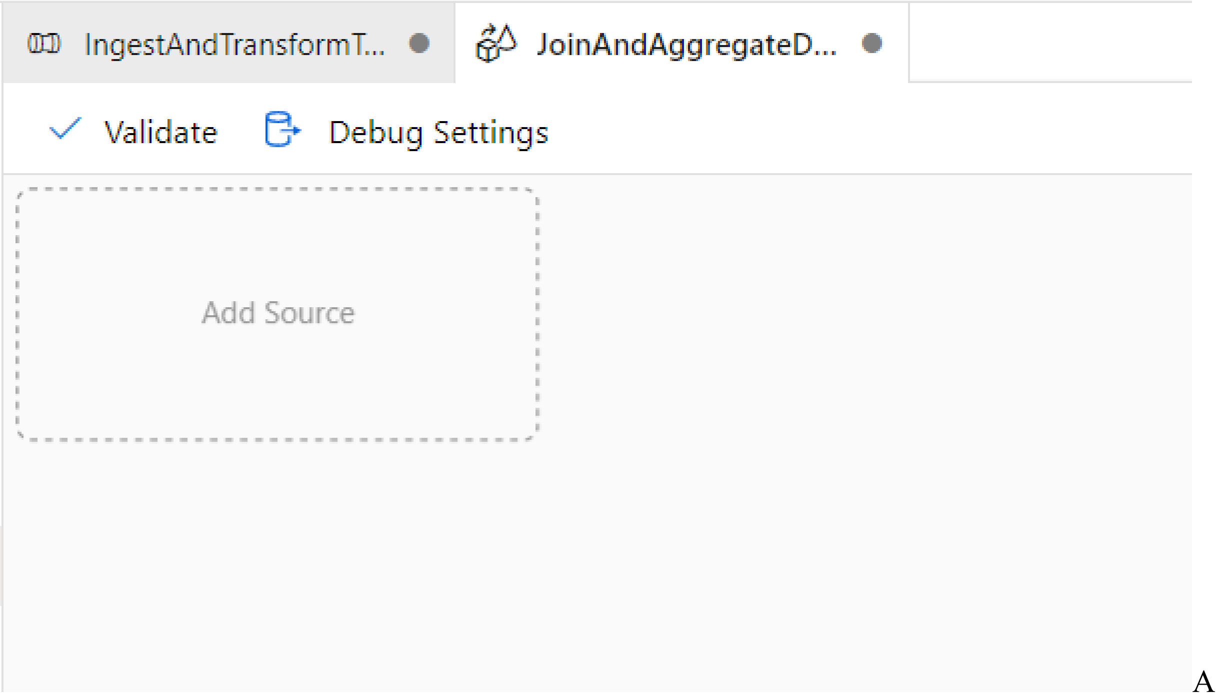


Name *
JoinAndAggregateData

Properties new DataFlow

Configure your trip data csv source

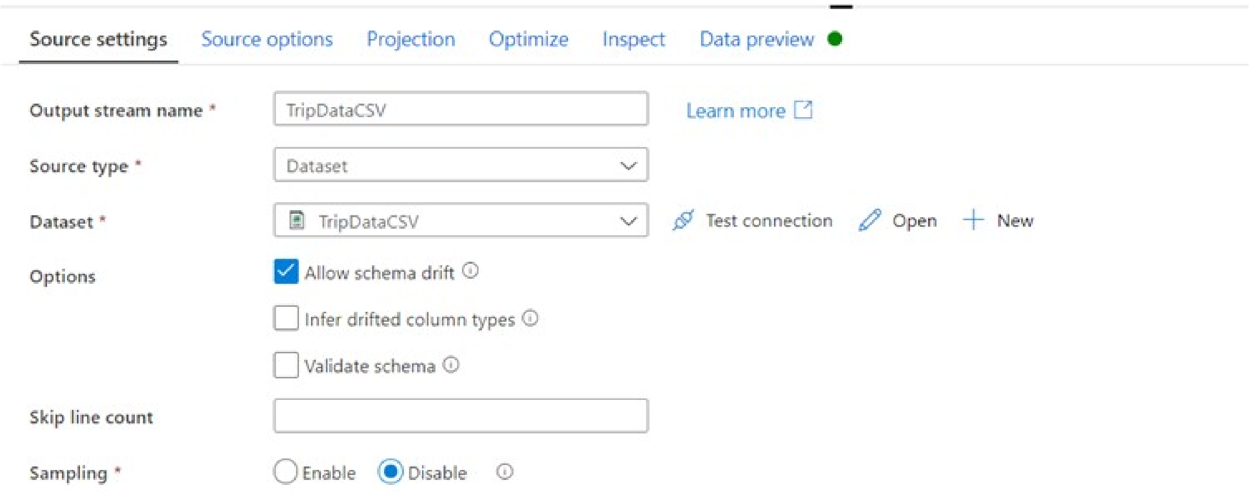
The first thing you want to do is configure your two source transformations. The first source will point to the 'TripDataCSV' DelimitedText dataset. To add a source transformation, click on the Add Source box in the canvas.



Add Source to Dataflow

Name your source 'TripDataCSV' and select the 'TripDataCSV' dataset from the source drop-down.

Since trip-data.csv exists now, click Open to go to the dataset settings tab.



Output stream name * TripDataCSV

Source type * Dataset

Dataset * TripDataCSV

Options ☑ Allow schema drift ⓘ

□ Infer drifted column types ⓘ

□ Validate schema ⓘ

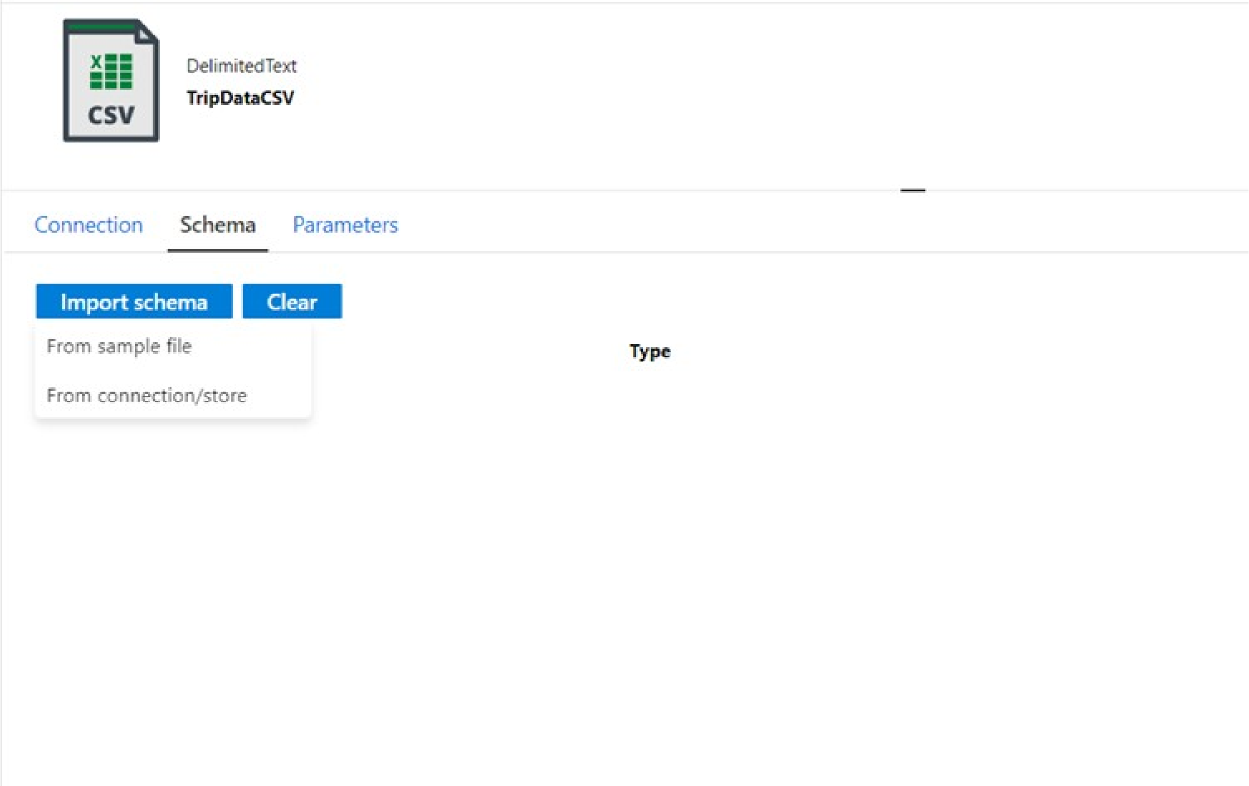
Skip line count

Sampling * ○ Enable ● Disable ⓘ

Source Settings of Dataflow

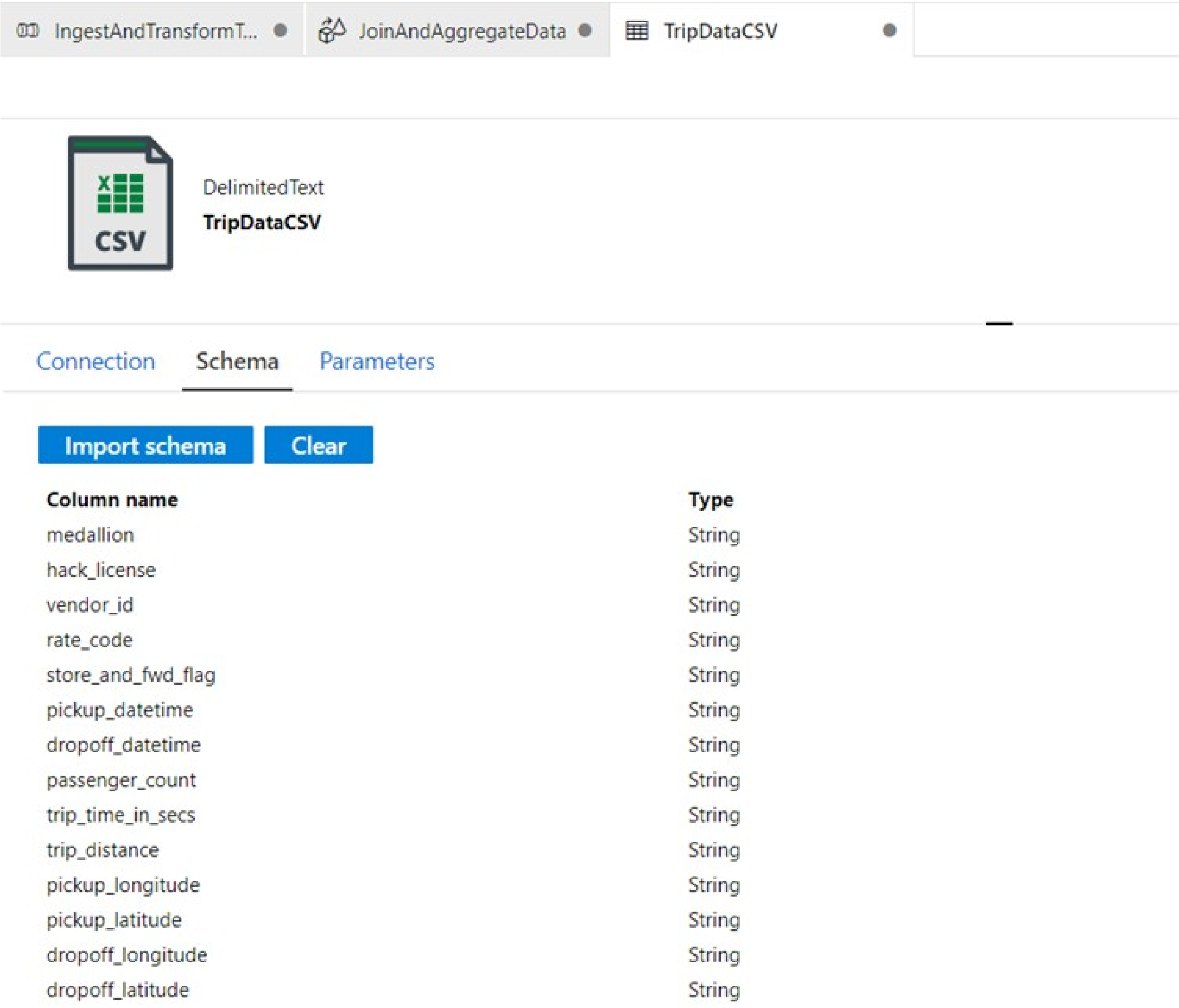
Go to tab Schema and click Import schema.

Select From connection/store to import directly from the file store.



Schema definition source Dataflow

If you select From connection/store the columns should appear as following:



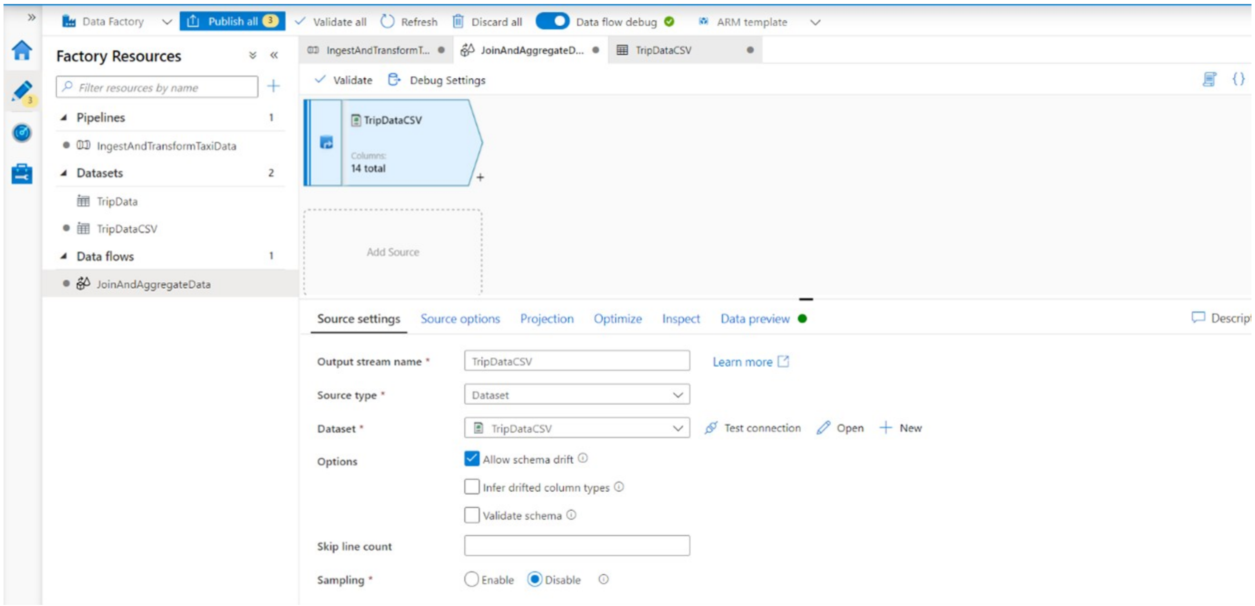
Column name	Type
medallion	String
hack_license	String
vendor_id	String
rate_code	String
store_and_fwd_flag	String
pickup_datetime	String
dropoff_datetime	String
passenger_count	String
trip_time_in_secs	String
trip_distance	String
pickup_longitude	String
pickup_latitude	String
dropoff_longitude	String
dropoff_latitude	String

image-07 Schema definition from Connection

Go back to data flow 'JoinAndAggregateData'.

If your debug cluster has started (indicated by a green circle next to the debug slider), you can get a snapshot of the data in the Data Preview tab.

Click Refresh to fetch a data preview.



Output stream name * TripDataCSV Learn more

Source type * Dataset

Dataset * TripDataCSV Test connection Open + New

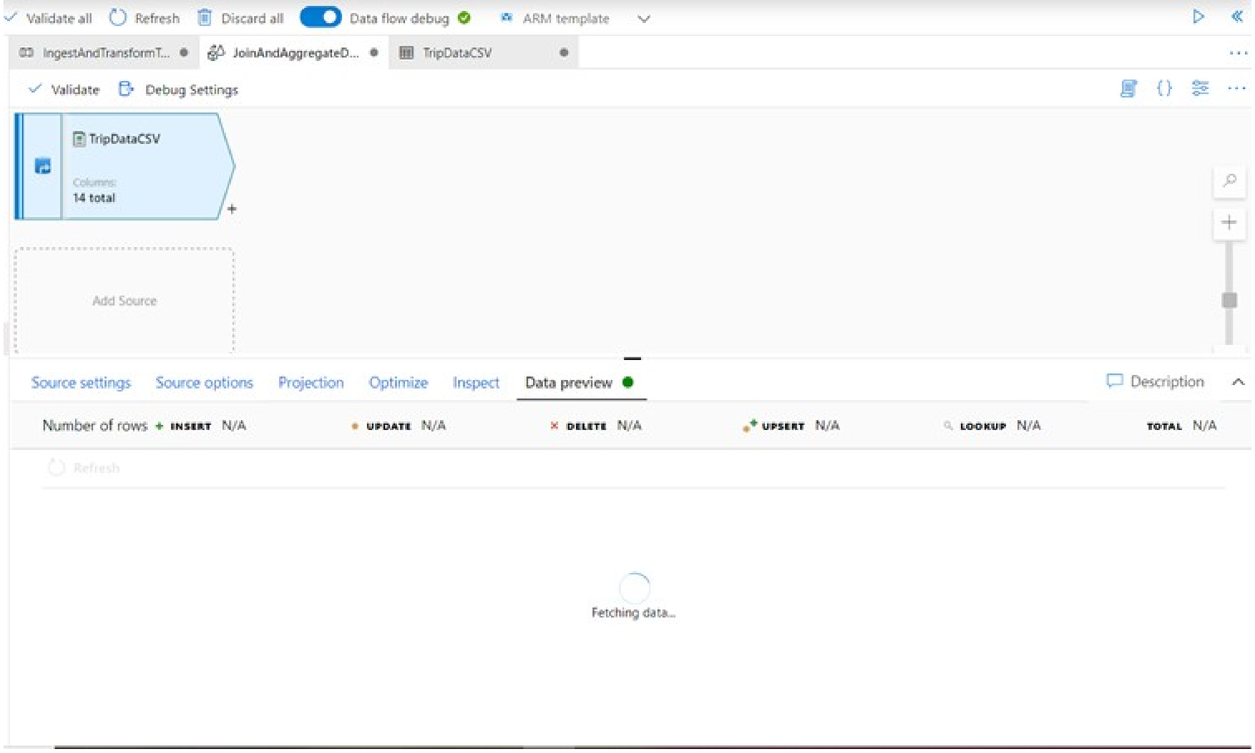
Options
☑ Allow schema drift
☐ Infer drifted column types
☐ Validate schema

Skip line count

Sampling * ○ Enable ● Disable

Source Settings after Schema Mapping Dataflow

When you select the Data Preview tab and click refresh the data will be fetched as long as your debug cluster is green, however Data preview does not write data.



Number of rows	INSERT	UPDATE	DELETE	UPSERT	LOOKUP	TOTAL
	N/A	N/A	N/A	N/A	N/A	N/A

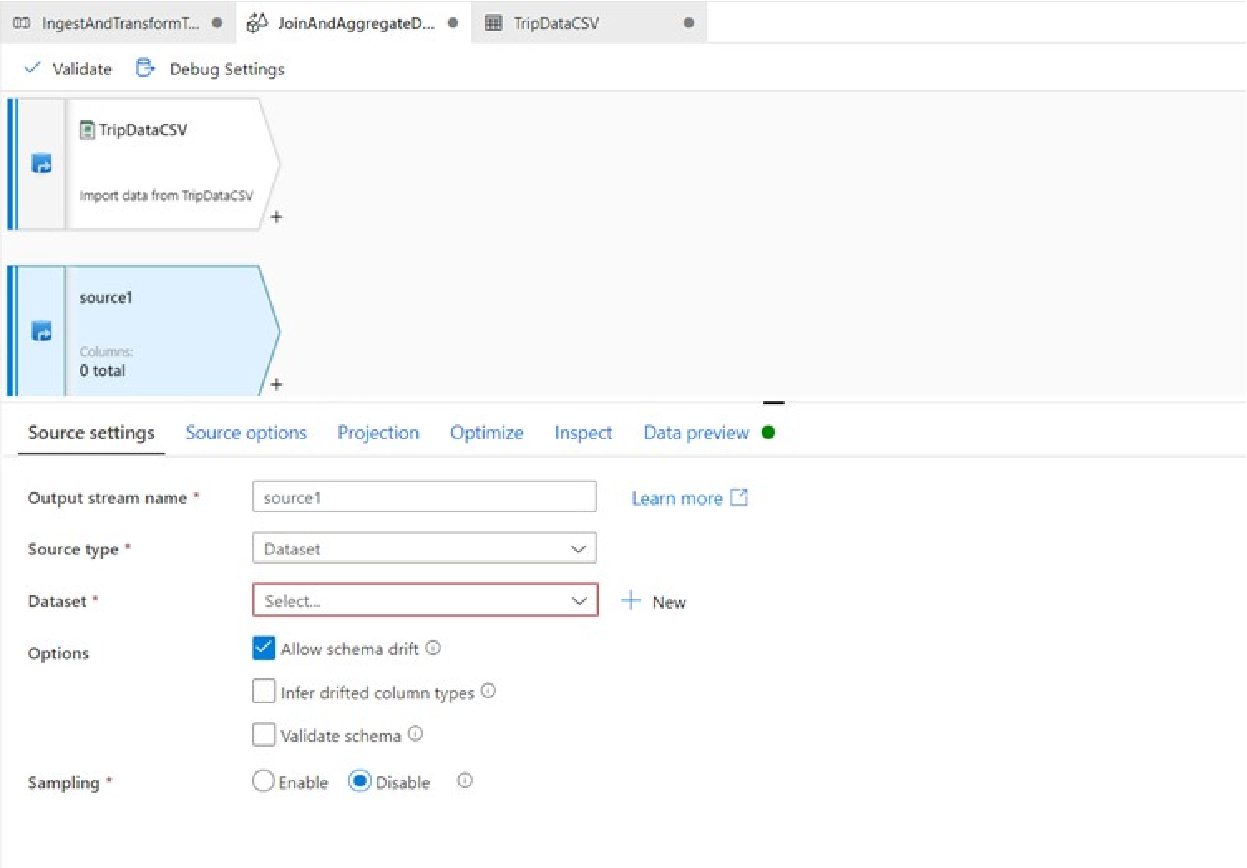
Data Preview Fetching Data

Configure your trip fares csv source:

The second source you're adding will point at the csv file 'TripFares.csv' that was shared during the Azure Data Share exercise.

Under your 'TripDataCSV' source, there will be another Add Source box.

Click it to add a new source transformation.



Output stream name * source1

Source type * Dataset

Dataset * Select...

Options ☑ Allow schema drift

□ Infer drifted column types

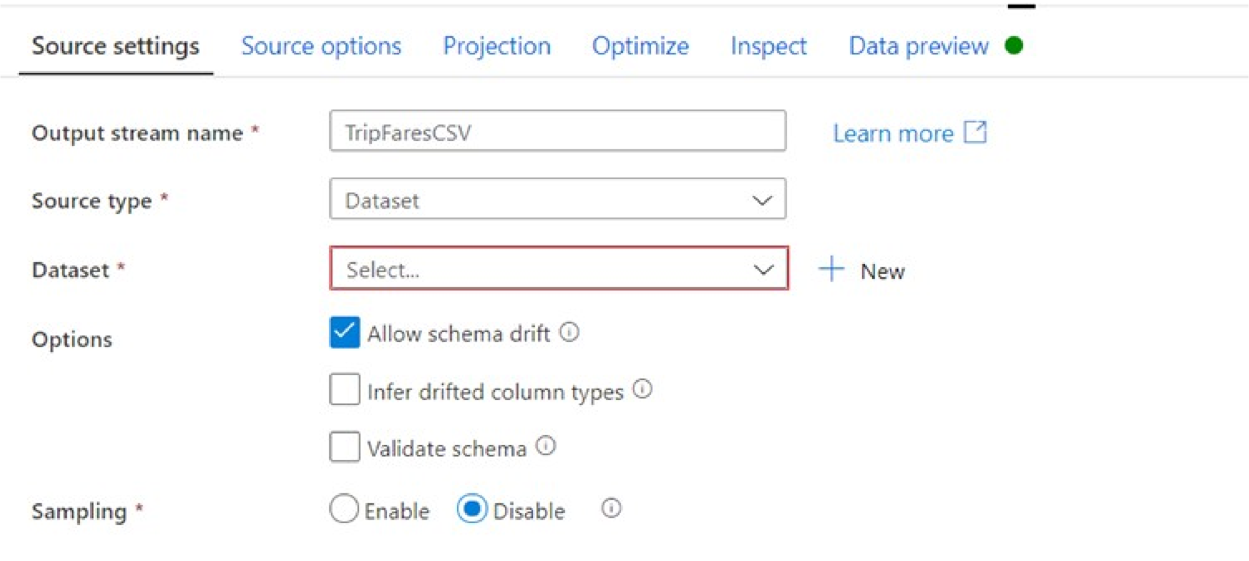
□ Validate schema

Sampling * ○ Enable ● Disable

Add Source Transformation

Name this source 'TripFaresCSV'.

Click New next to the source dataset field to create a new ADLS gen2 dataset.



Output stream name * TripFaresCSV

Source type * Dataset

Dataset * Select...

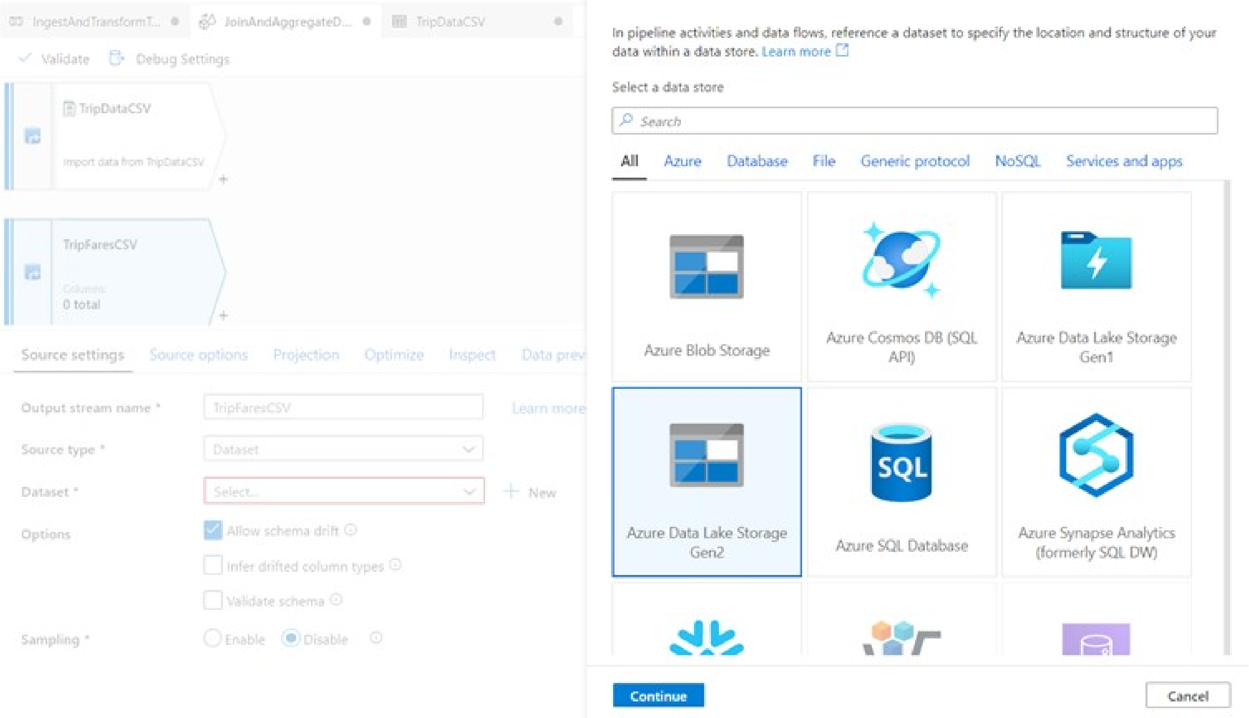
Options
☑ Allow schema drift
□ Infer drifted column types
□ Validate schema

Sampling * ○ Enable ● Disable

New Source creation for Transformation

Select the Azure Data Lake Storage gen2 tile and click continue.

You may notice many of the connectors in data factory are not supported in mapping data flow. To transform data from one of these sources, ingest it into a supported source using the copy activity.

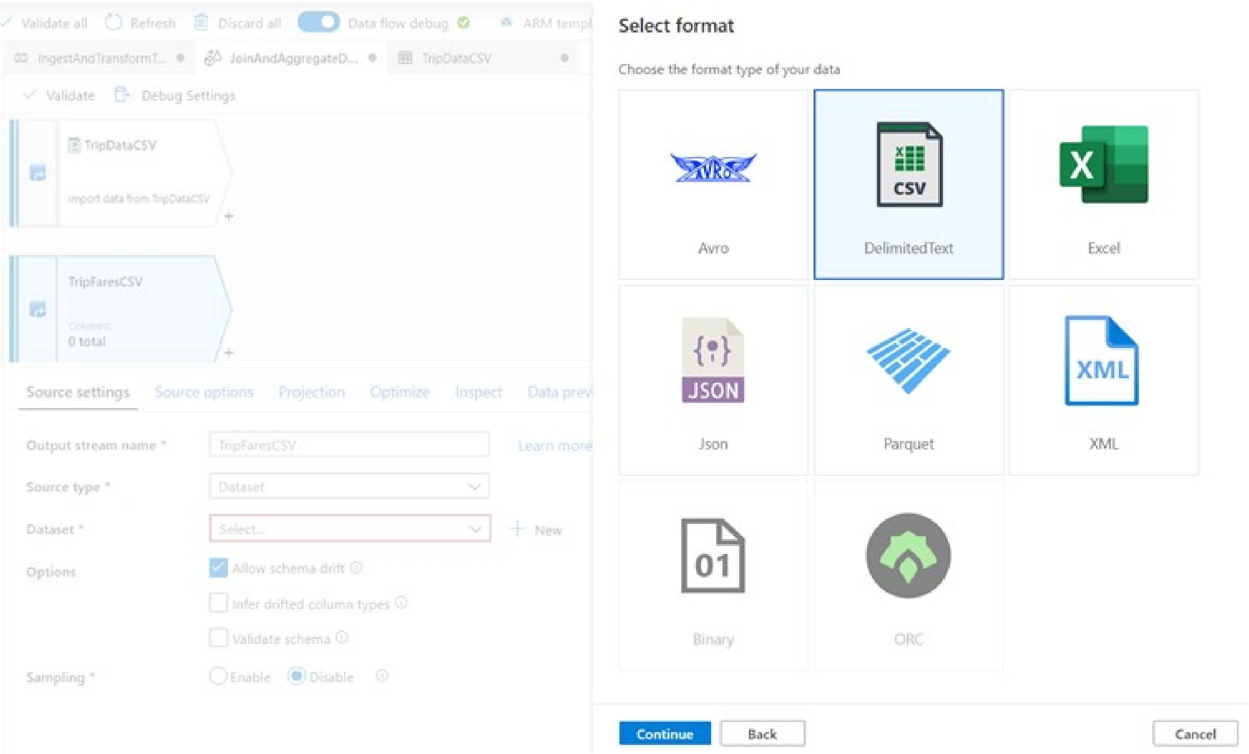


In pipeline activities and data flows, reference a dataset to specify the location and structure of your data within a data store. Learn more

Select a data store

Select Azure Data Lake Storage Gen 2 for new Source Transformation

In the select format pane, select DelimitedText as you are reading from a csv file. Click continue.



Output stream name * TripFaresCSV

Source type * Dataset

Dataset * Select...

Options ☑ Allow schema drift

□ Infer drifted column types

□ Validate schema

Sampling * ○ Enable ● Disable

Select Delimited Text

Name your dataset 'TripFaresCSV'.

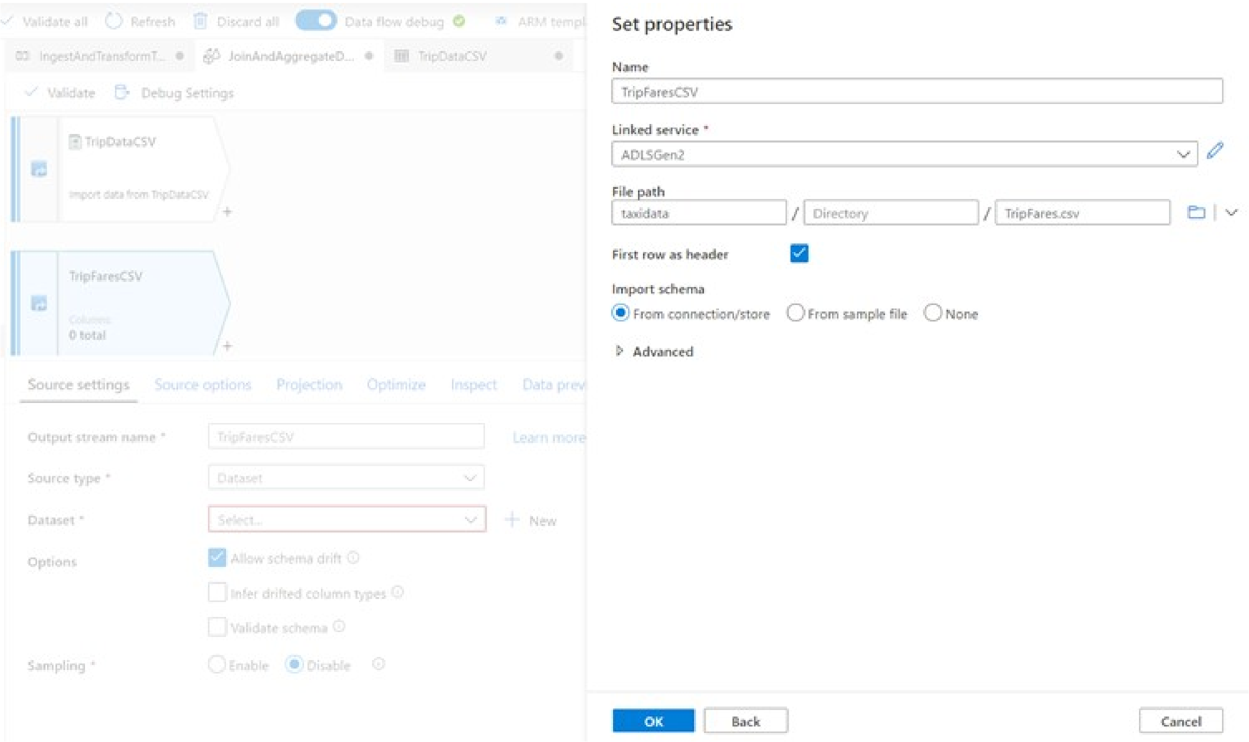
Select 'ADLSGen2' as your linked service.

Set the file path to taxidata/TripFares.csv. In the Azure Data Share exercise you have called it TripFares.csv in container 'taxidata'.

Set First row as header to true as the input data has headers.

Import the schema From connection/store.

Click OK when finished.



Output stream name * TripFaresCSV

Source type * Dataset

Dataset * Select...

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

Sampling * ○ Enable ● Disable

Set properties

Name: TripFaresCSV

Linked service *: ADLSGen2

File path: taxidata / Directory / TripFares.csv

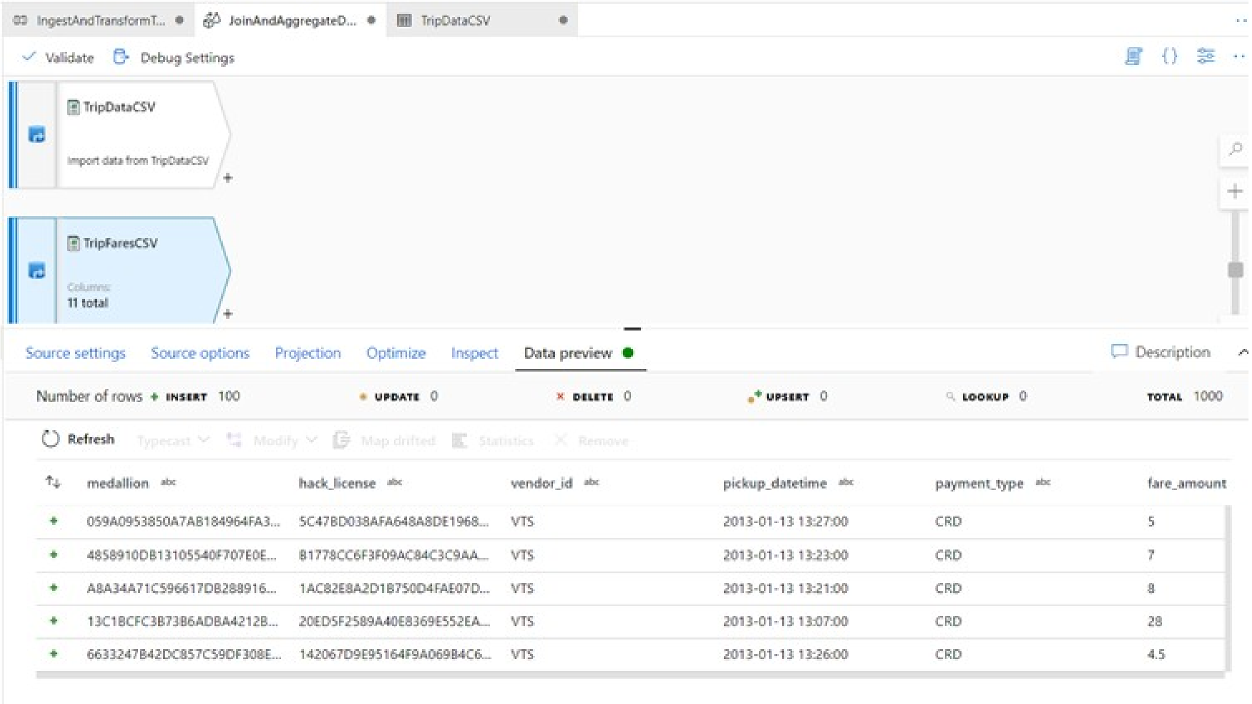
First row as header ☑

Import schema: ● From connection/store ○ From sample file ○ None

▷ Advanced

Set properties of File

To verify your source is configured correctly, fetch a data preview in the Data Preview tab.



Number of rows + INSERT 100 UPDATE 0 × DELETE 0 UPSERT 0 LOOKUP 0 TOTAL 1000

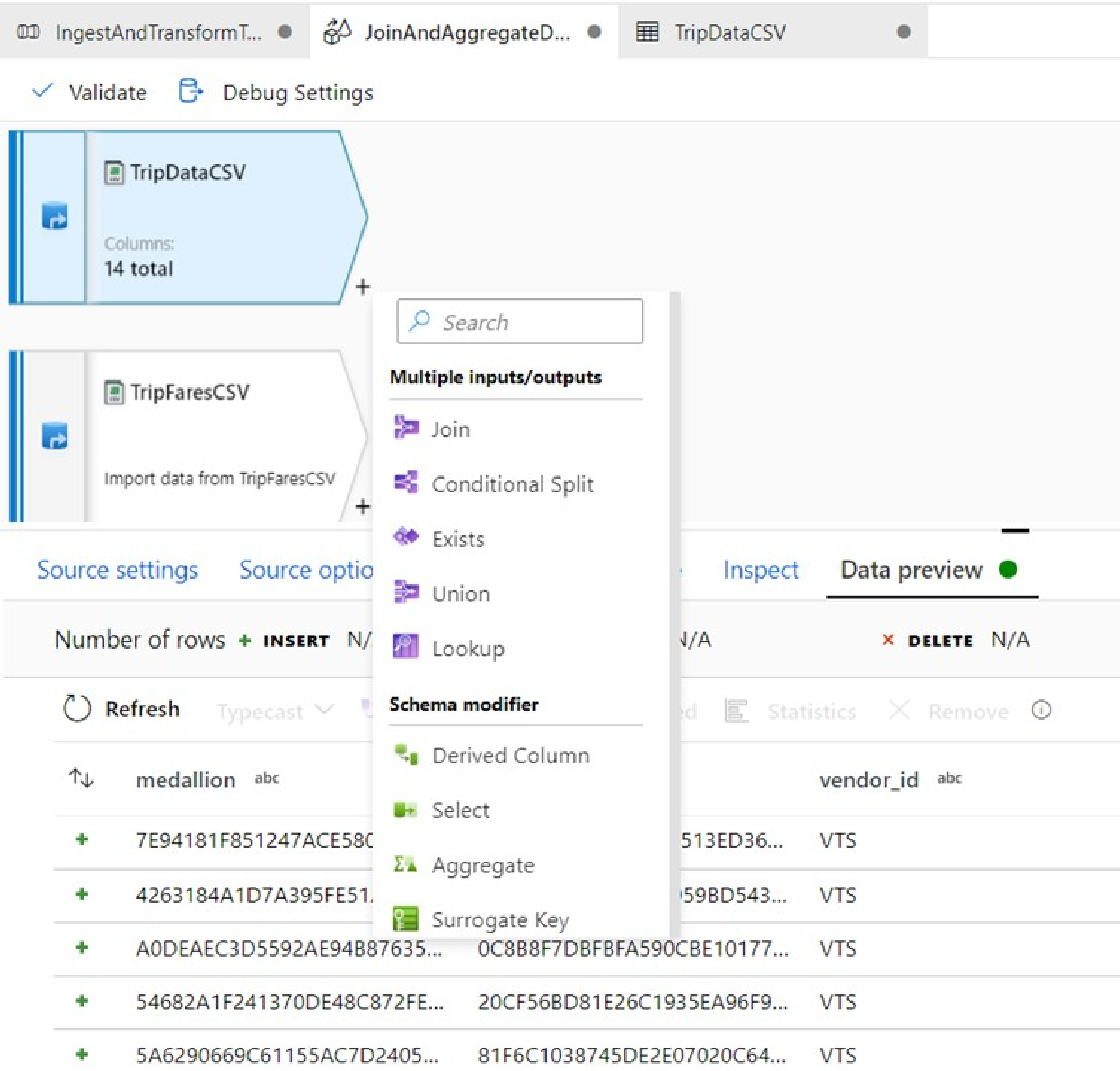
	medallion	hack_license	vendor_id	pickup_datetime	payment_type	fare_amount
+	059A0953850A7AB184964FA3...	5C47BD038AFA648A8DE1968...	VTS	2013-01-13 13:27:00	CRD	5
+	4858910DB13105540F707E0E...	81778CC6F3F09AC84C3C9AA...	VTS	2013-01-13 13:23:00	CRD	7
+	A8A34A71C596617DB288916...	1AC82E8A2D1B750D4FAE07D...	VTS	2013-01-13 13:21:00	CRD	8
+	13C1BCFC3B73B6ADBA4212B...	20ED5F2589A40E8369E552EA...	VTS	2013-01-13 13:07:00	CRD	28
+	6633247B42DC857C59DF308E...	142067D9E95164F9A069B4C6...	VTS	2013-01-13 13:26:00	CRD	4.5

Data Preview of created dataset

Inner join TripDataCSV and TripFaresSQL

To add a new transformation, click the plus icon in the bottom-right corner of 'TripDataCSV'.

Under Multiple inputs/outputs, select Join.



medallion abc		vendor_id abc
7E94181F851247ACE58C	513ED36...	VTS
4263184A1D7A395FE51.	59BD543...	VTS
A0DEAEC3D5592AE94B87635...	0C8B8F7DBFBFA590CBE10177...	VTS
54682A1F241370DE48C872FE...	20CF56BD81E26C1935EA96F9...	VTS
5A6290669C61155AC7D2405...	81F6C1038745DE2E07020C64...	VTS

Add Join Transformation

Name your join transformation 'InnerJoinWithTripFares'.

Select 'TripFaresCSV' from the right stream dropdown.

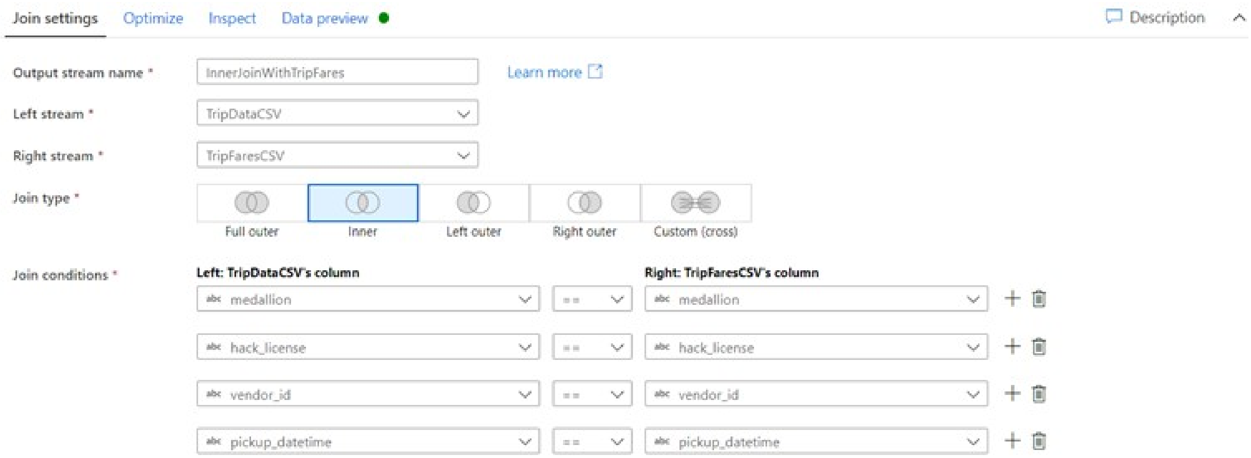
Select Inner as the join type.

Select which columns you wish to match on from each stream via the Join conditions dropdown.

To add an additional join condition, click on the plus icon next to an existing condition.

By default, all join conditions are combined with an AND operator which means all conditions must be met for a match.

We want to match on columns medallion, hack_license, vendor_id, and pickup_datetime



Output stream name * InnerJoinWithTripFares

Left stream * TripDataCSV

Right stream * TripFaresCSV

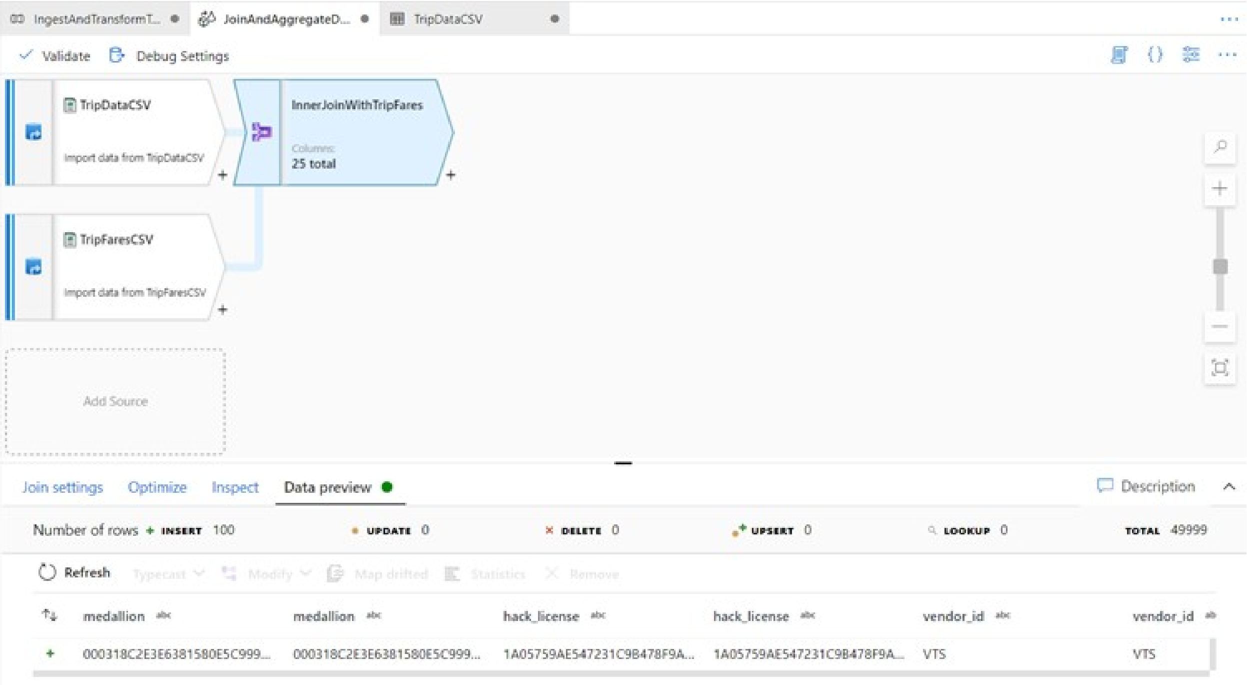
Join type * Inner

Join conditions *

Left: TripDataCSV's column		Right: TripFaresCSV's column
medallion	==	medallion
hack_license	==	hack_license
vendor_id	==	vendor_id
pickup_datetime	==	pickup_datetime

Match Columns for Join Transformation

Verify you successfully joined 25 columns together with Data Preview.



Number of rows + INSERT 100 UPDATE 0 DELETE 0 UPSERT 0 LOOKUP 0 TOTAL 49999

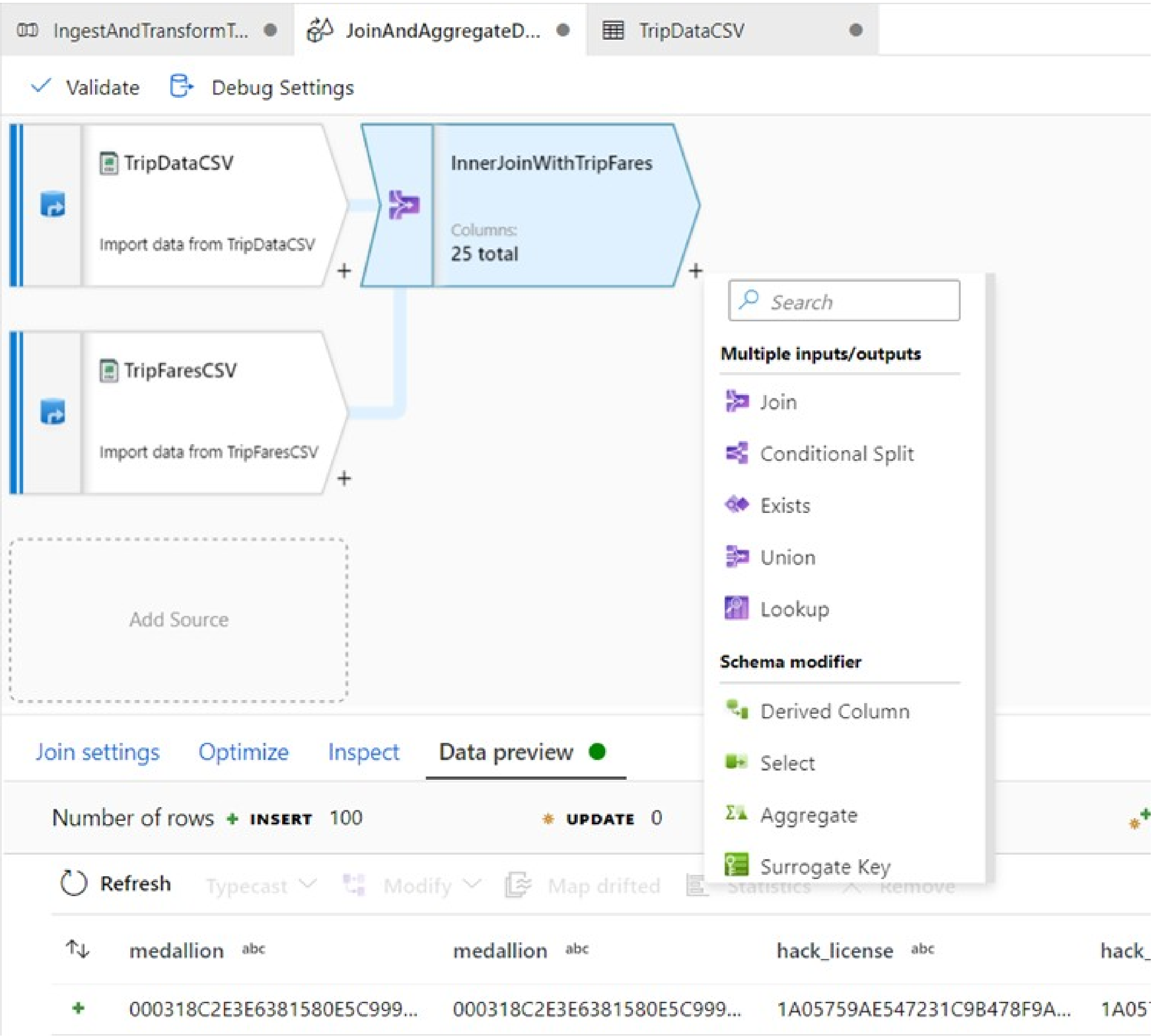
	medallion	medallion	hack_license	hack_license	vendor_id	vendor_id
+	000318C2E3E6381580E5C999...	000318C2E3E6381580E5C999...	1A05759AE547231C9B478F9A...	1A05759AE547231C9B478F9A...	VTS	VTS

Data Preview for Join Transformation

Aggregate by payment_type

After you complete a join transformation, add an aggregate transformation by clicking the plus icon next to 'InnerJoinWithTripFares.

Choose Aggregate under Schema modifier and call the Aggregate: ‘AggregateByPaymentType’

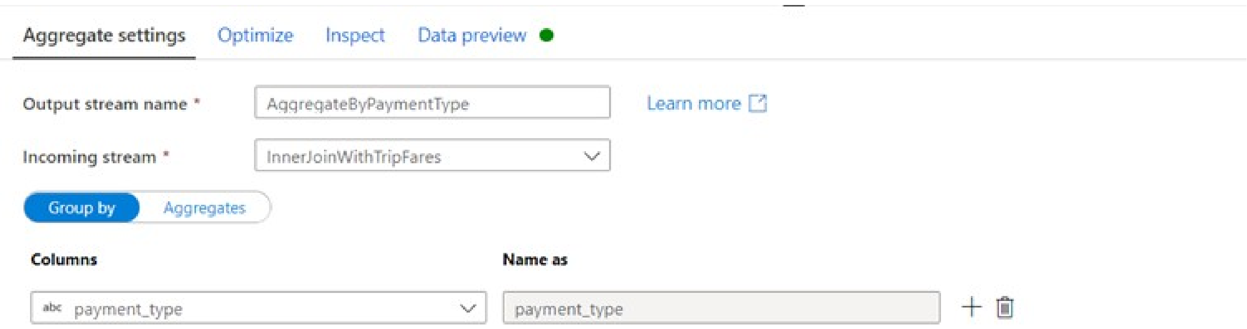


Number of rows + INSERT 100 * UPDATE 0

	medallion abc	medallion abc	hack_license abc	hack_
+	000318C2E3E6381580E5C999...	000318C2E3E6381580E5C999...	1A05759AE547231C9B478F9A...	1A05

Add Aggregation for Transformation

Name your aggregate transformation 'AggregateByPaymentType'. Select payment_type as the group by column.



Output stream name * AggregateByPaymentType

Incoming stream * InnerJoinWithTripFares

Columns	Name as
abc payment_type	payment_type

Group by setting for Aggregation

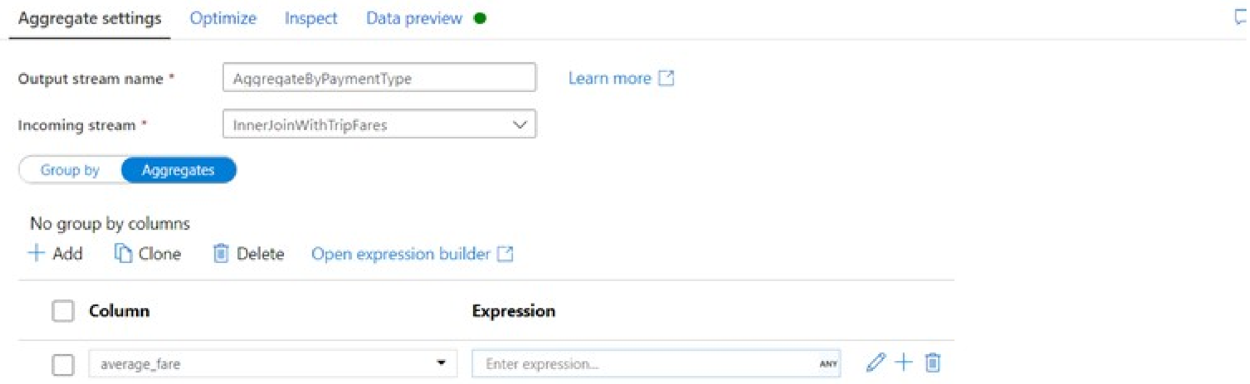
Go to the Aggregates tab. Here, you will specify two aggregations:

```
* The average fare grouped by payment type

* The total trip distance grouped by payment type
```

Create the average fare expression.

In the text box labeled Add or select a column, enter 'average_fare'.



Output stream name *: AggregateByPaymentType

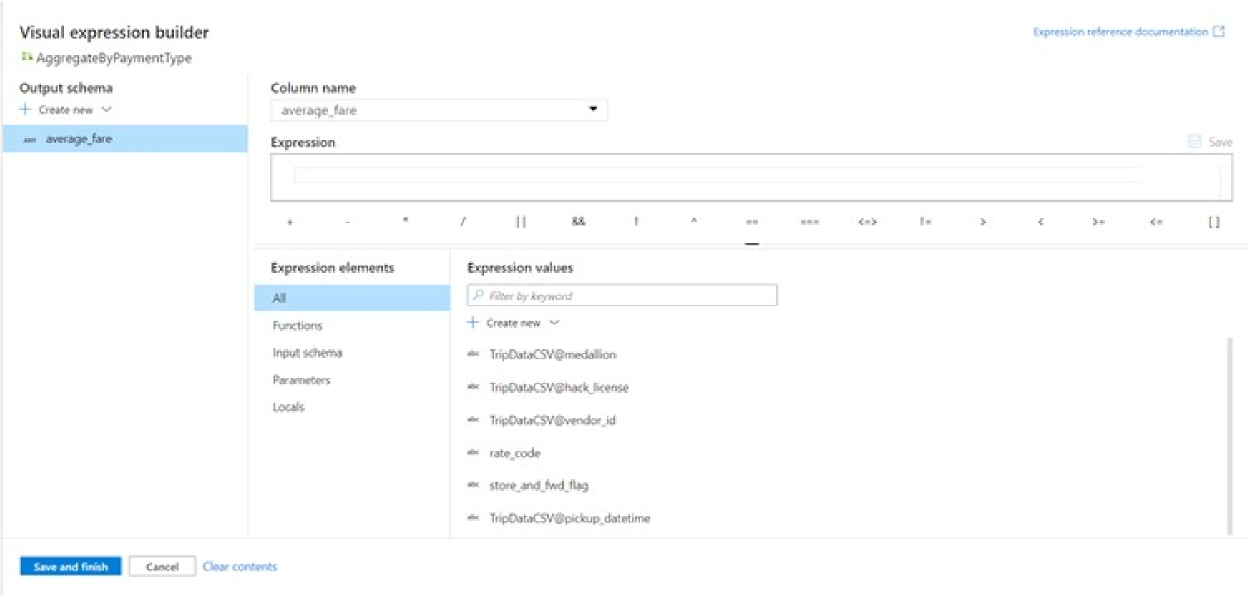
Incoming stream *: InnerJoinWithTripFares

No group by columns

Column	Expression
average_fare	Enter expression...

Aggregates settings

To enter an aggregation expression, click on the marker next to the the blue box labeled Enter expression.



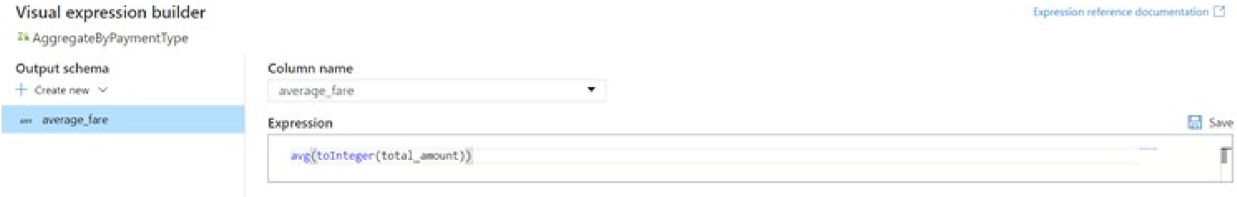
Column name: average_fare

Expression builder for Aggregations

This will open up the data flow expression builder, a tool used to visually create data flow expressions using input schema, built-in functions and operations, and user-defined parameters. For more information on the capabilities of the expression builder, see the expression builder documentation.

To get the average fare, use the avg() aggregation function to aggregate the total_amount column cast to an integer with toInteger(). In the data flow expression language, this is defined as avg(toInteger(total_amount)).

Click Save and finish when you are done.



Column name: average_fare

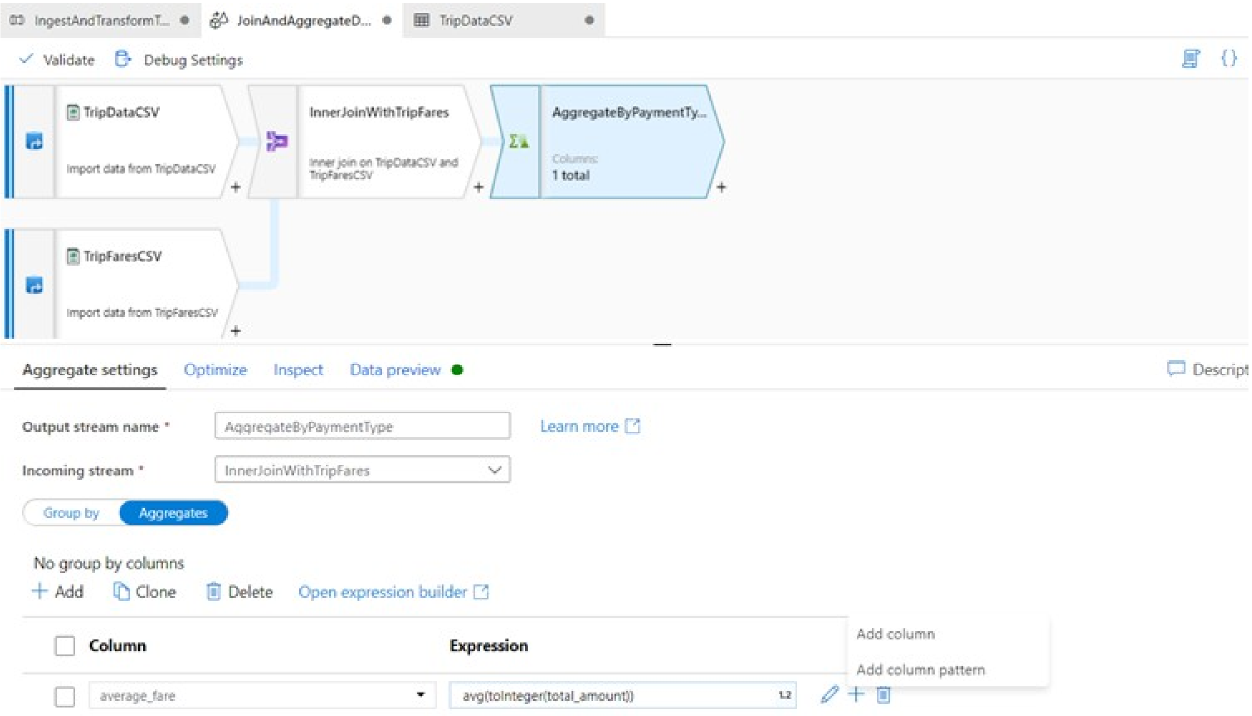
Expression:

```
avg(toInteger(total_amount))
```

Visual Expression builder for Aggregations

To add an additional aggregation expression, click on the plus icon next to average_fare.

Select Add column.



Output stream name * AggregateByPaymentType

Incoming stream * InnerJoinWithTripFares

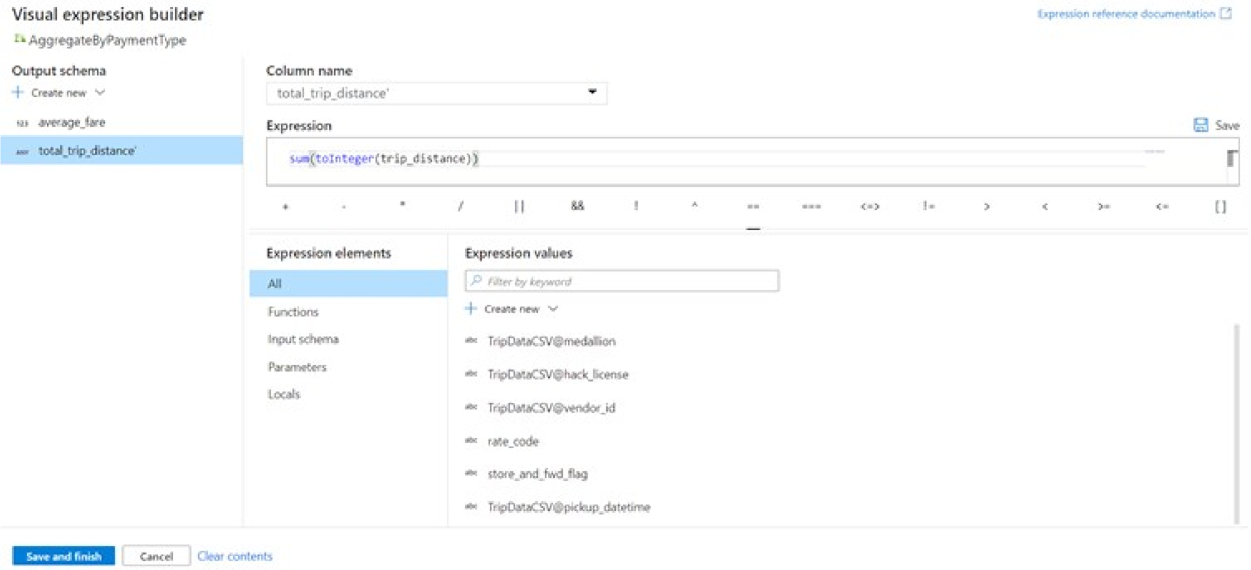
Column	Expression
average_fare	avg(toInteger(total_amount))

Add a column for another expression

In the text box labeled Add or select a column, enter 'total_trip_distance'. As in the last step, open the expression builder to enter in the expression.

To get the total trip distance, use the sum() aggregation function to aggregate the trip_distance column cast to an integer with toInteger().

In the data flow expression language, this is defined as sum(toInteger(trip_distance)).



Visual expression builder

AggregateByPaymentType

Column name: total_trip_distance'

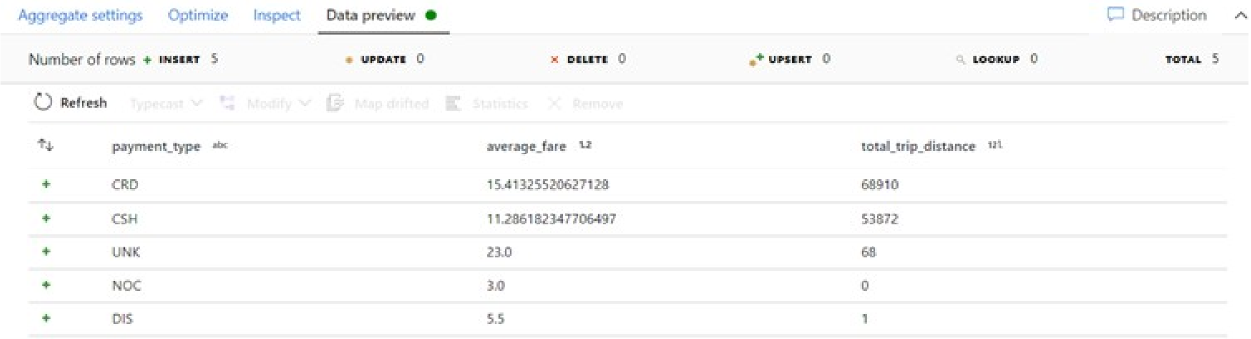
Expression:

```
sum(toInteger(trip_distance))
```

Add expression to new column

Click Save and finish.

Test your transformation logic in the Data Preview tab.



Number of rows + INSERT 5 · UPDATE 0 × DELETE 0 UPSERT 0 LOOKUP 0 TOTAL 5

	payment_type	average_fare	total_trip_distance
+	CRD	15.41325520627128	68910
+	CSH	11.286182347706497	53872
+	UNK	23.0	68
+	NOC	3.0	0
+	DIS	5.5	1