

# Azure Data Factory: Mapping Data Flow – first blood

Kamil Nowinski

Principal Microsoft Consultant

altius



Microsoft®  
Most Valuable  
Professional



#SQLSatMadrid



@NowinskiK

# BIG Thanks to SQLSatMadrid sponsors

Platinum



Gold



Silver



Venue

Azu

a Factor



Global



@NowinskiK

# Kamil Nowiński



Hola!

Principal Microsoft Consultant at Altius ([www.altiusdata.com](http://www.altiusdata.com))  
15+ yrs experience as DEV/BI/(DBA)  
Member of the Data Community PL  
Project member of „SCD Merge Wizard”  
Founder of blog SQLPlayer ([www.SQLplayer.net](http://www.SQLplayer.net))

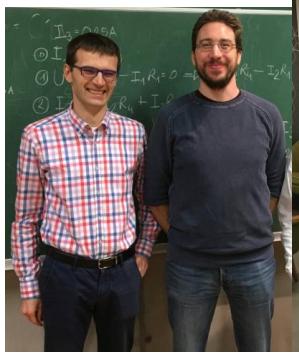
SQL Server Certificates:  
MCITP, MCP, MCTS, MCSA, MCSE Data Platform,  
MCSE Data Management & Analytics  
Moreover: Bicycle, Running, Digital photography  
@NowinskiK, @SQLPlayer

## BLOG & Interviews

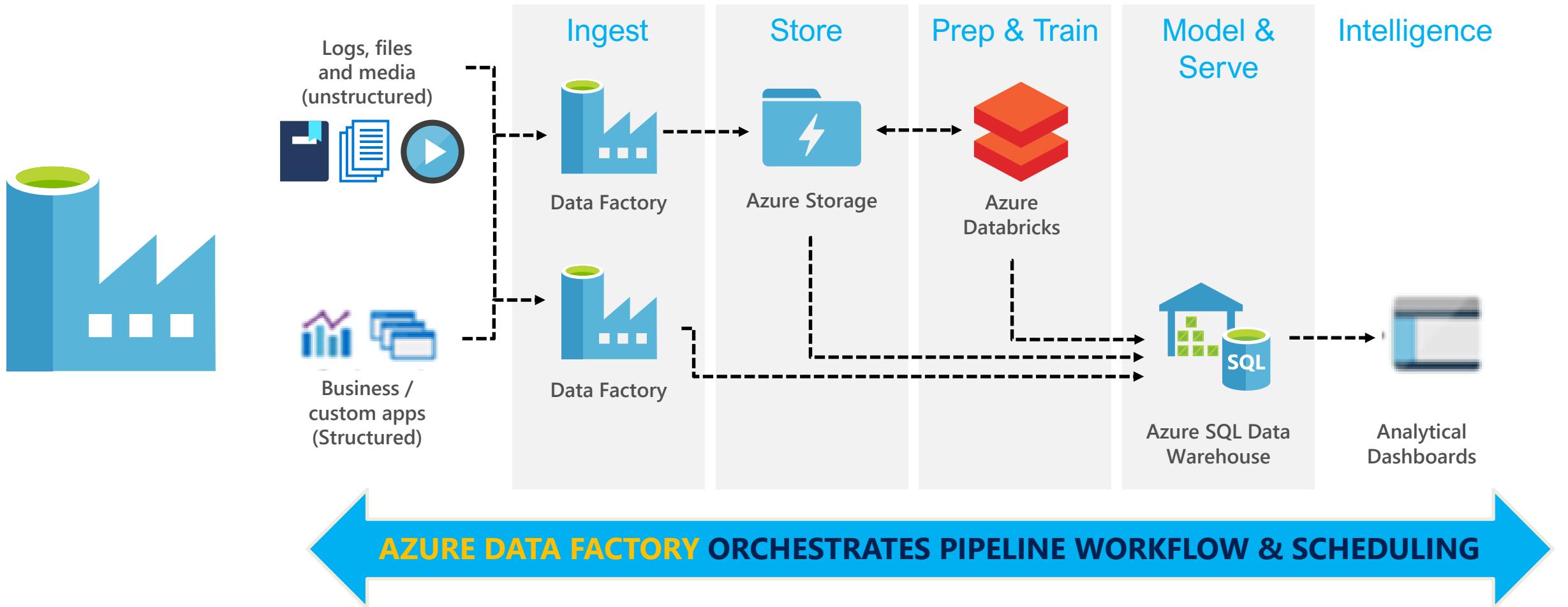


[www.SQLPlayer.net](http://www.SQLPlayer.net)

# PODCAST – interviews with...



# What the Azure Data Factory is?

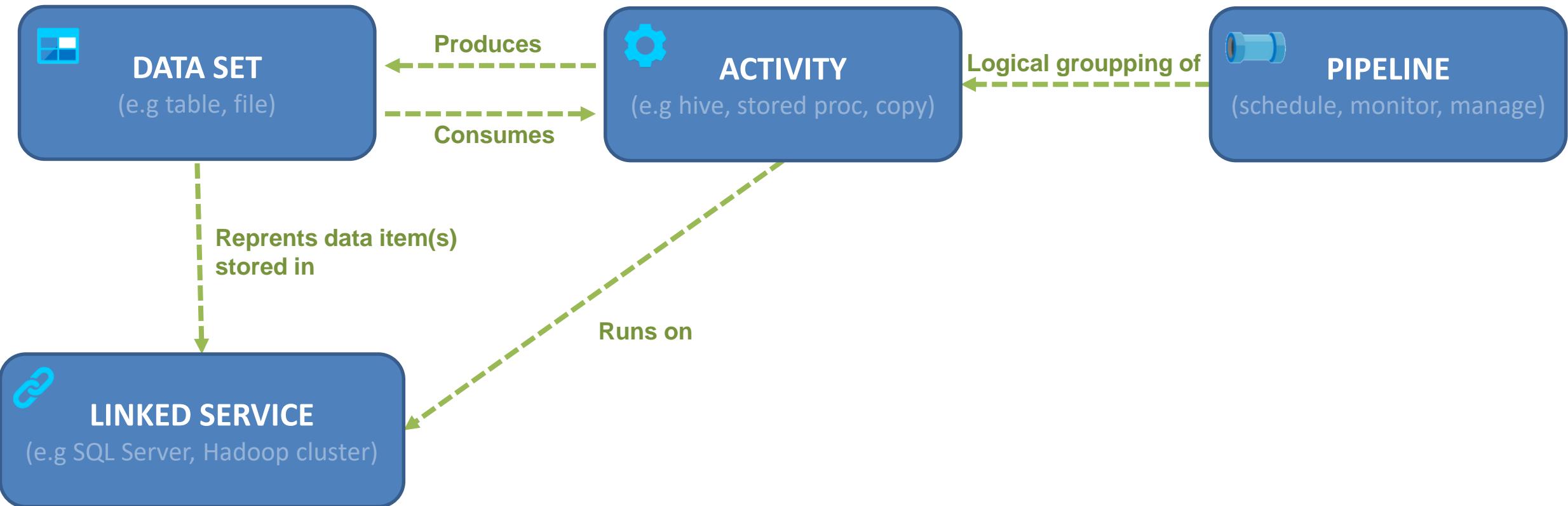


# Access all your data

- 80+ connectors & growing
- Azure IR available in 21 regions
- Hybrid connectivity using self-hosted IR: on-prem & VNet

| Azure (13)                | Database (24)                |            | File Storage (5) | NoSQL (3) | Services and Apps (28) |                            | Generic (4) |
|---------------------------|------------------------------|------------|------------------|-----------|------------------------|----------------------------|-------------|
| Blob Storage              | Amazon Redshift              | Netezza    | Amazon S3        | Cassandra | Amazon MWS             | Office 365 *               | HTTP        |
| Cosmos DB (MongoDB API) * | DB2                          | Oracle     | File System      | Couchbase | CDS for Apps           | Paypal                     | OData       |
| Cosmos DB (SQL API)       | Drill                        | Phoenix    | FTP              | MongoDB   | Concur                 | QuickBooks                 | ODBC        |
| Data Lake Storage Gen1    | Google BigQuery              | PostgreSQL | HDFS             |           | Dynamics 365           | Salesforce                 | REST *      |
| Data Lake Storage Gen2    | Greenplum                    | Presto     | SFTP             |           | Dynamics CRM           | Salesforce Marketing Cloud |             |
| DB for MySQL              | HBase                        | SAP BW     |                  |           | GE Historian           | Salesforce Service Cloud   |             |
| DB for PostgreSQL         | Hive                         | SAP HANA   |                  |           | Google AdWords         | SAP C4C                    |             |
| File Storage              | Impala                       | Spark      |                  |           | HubSpot                | SAP ECC                    |             |
| Kusto *                   | Informix                     | SQL Server |                  |           | Jira                   | ServiceNow                 |             |
| Search Index              | MariaDB                      | Sybase     |                  |           | Magento                | Shopify                    |             |
| SQL DB                    | Microsoft Access             | Teradata   |                  |           | Marketo                | Square                     |             |
| SQL DW                    | MySQL                        | Vertica    |                  |           | Oracle Eloqua          | Web table                  |             |
| Table Storage             |                              |            |                  |           | Oracle Responsys       | Xero                       |             |
|                           | Supported as Source and Sink |            |                  |           | Oracle Service Cloud   | Zoho                       |             |
|                           | Supported as Source only     |            |                  |           |                        |                            |             |
|                           | Supported as Sink only       |            |                  |           |                        |                            |             |

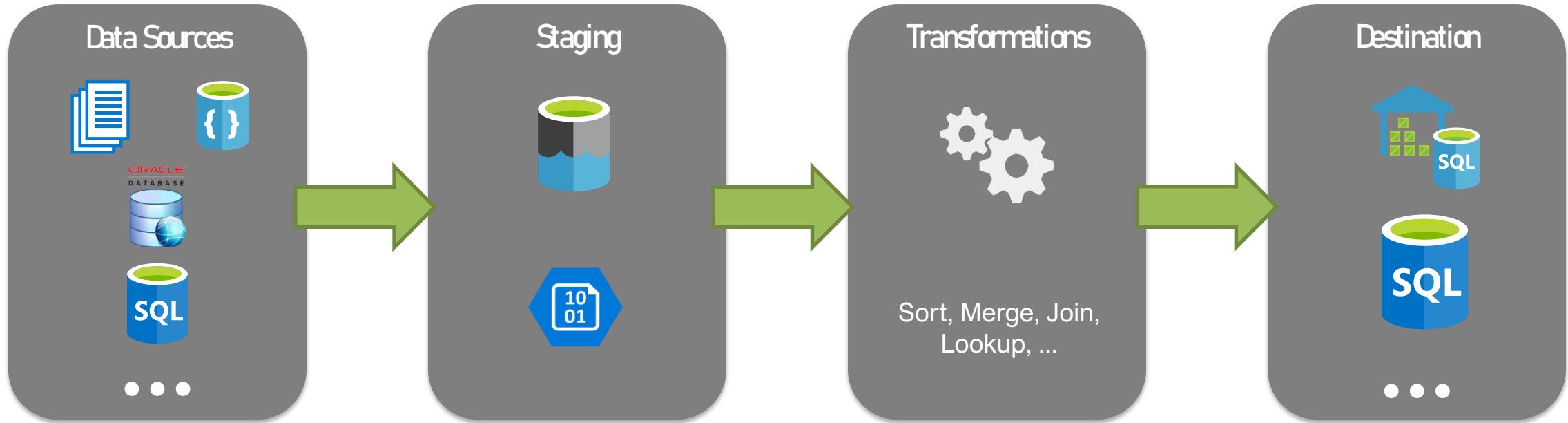
# ADF Key Concepts



Visual Data Transformations with

# MAPPING DATA FLOW

# What the hell (Mapping) Data Flows are?

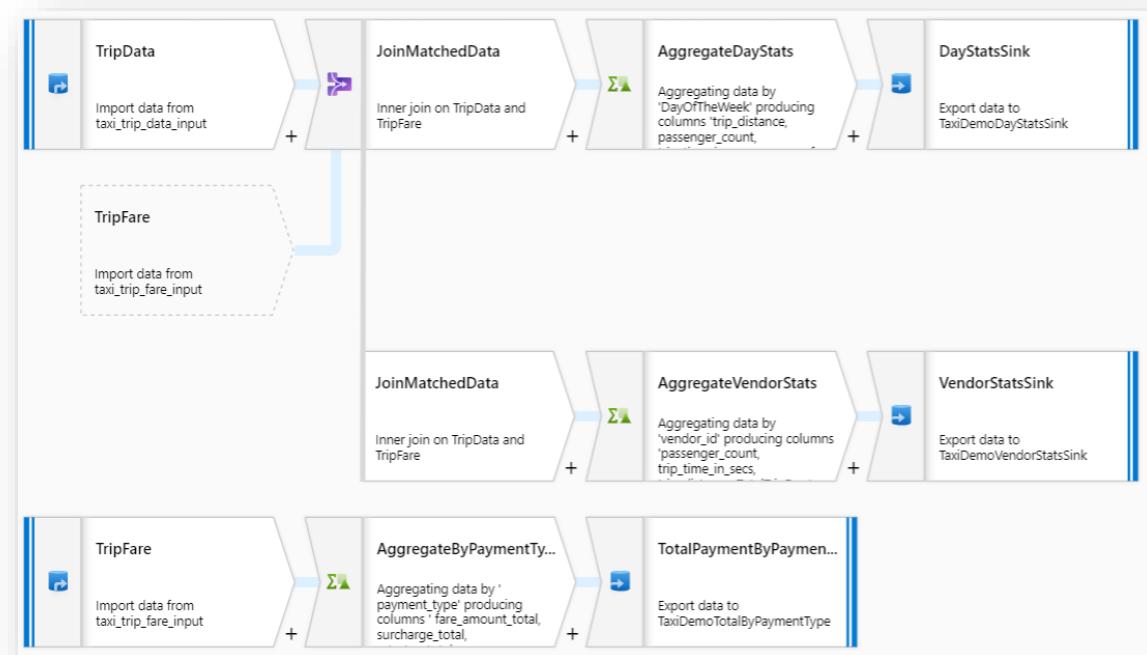


- Explicit user action
- User places data source(s) on design surface, from toolbox
- Select explicit sources
- Implicit/Explicit
- Data Lake staging area as default
- User does not need to configure this manually
- Advanced feature to set staging area options
- File formats/types:  
(Parquet, JSON, txt, CSV, ...)
- Explicit user action
- User places transformations on design surface, from toolbox
- User must set properties for transformation steps and step connectors
- Explicit user action
- User chooses destination connector(s)
- User sets connector property options

src: (Microsoft) ADF Data Flow Private Preview Overview

# Code-free Data Transformation at Scale

- Does not require understanding of Spark, Big Data Execution Engines, Clusters, Scala ...
- Focus on building business logic and data transformation
  - Data cleansing
  - Aggregation
  - Data conversions
  - Data prep
  - Data exploration
  - ETL Data Loading into DW



Azure Data Factory: Mapping Data Flow – first blood

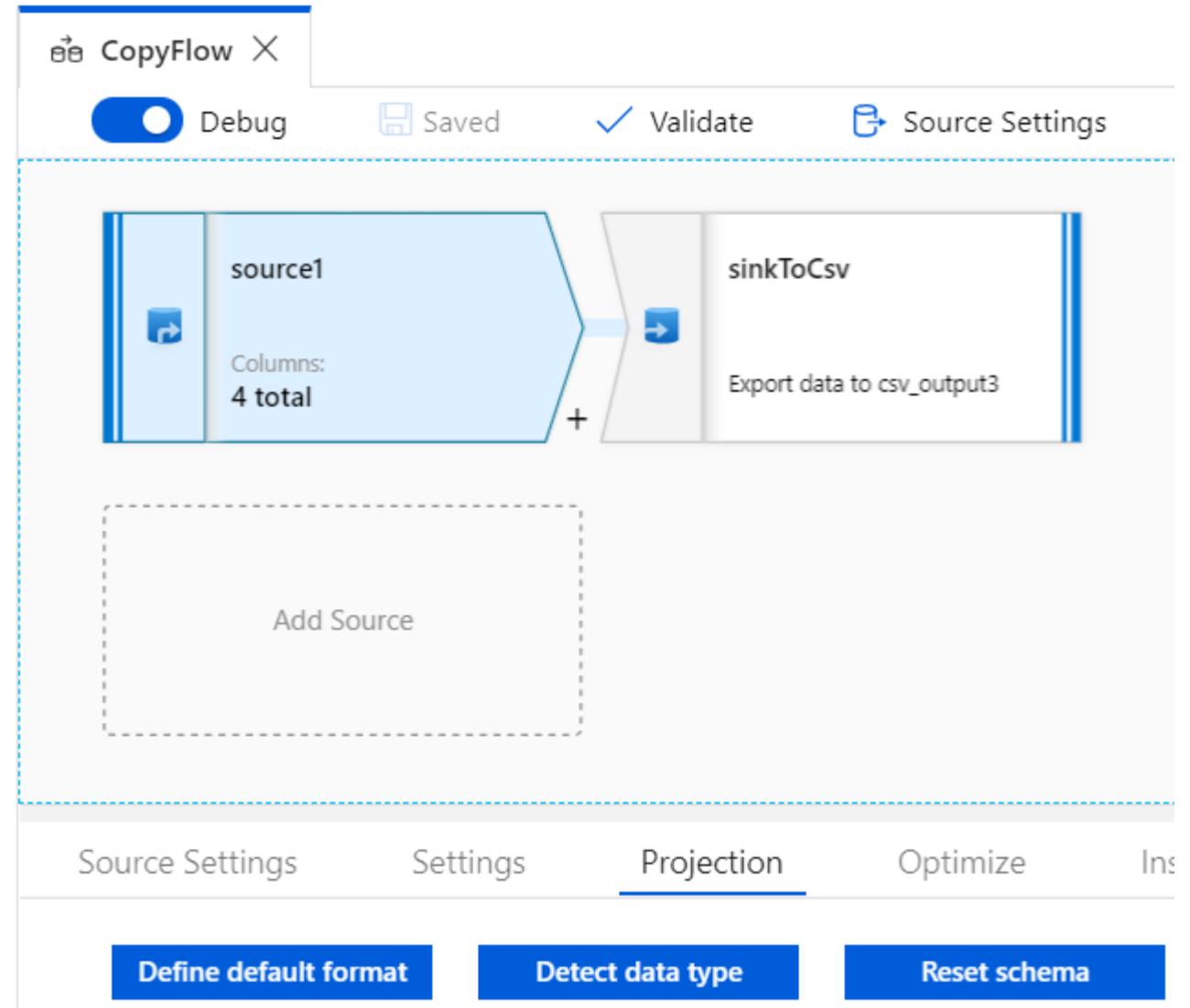
... not

```
MovieRecommendation4EDemo.txt
1 HDFS Cluster Details:
2 Adfhd1.azurehdinsight.net
3 Admin
4 Adfg123456
5
6 Storage:
7 adfhdstorage
8 /anyPw6G1J7Z81BWhm1So/YGdyG74d-S1JAr+sN7bJgb954705gUCMokzI9UXct4OxZo8xIKHdMKw==_
9
10 Cluster Remote Login Details:
11 Adf
12 India@1234
13
14 HiveQuery:
15 DROP TABLE IF EXISTS MovieRatings;
16 CREATE EXTERNAL TABLE MovieRatings
17 (
18     UserID int,
19     MovieID int,
20     Rating int,
21    TimeStamp string
22 ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE LOCATION '${hiveconf:MovieRatings}';
23
24 DROP TABLE IF EXISTS MovieTitles;
25 CREATE EXTERNAL TABLE MovieTitles
26 (
27     MovieID int,
28     MovieName string
29 ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE LOCATION '${hiveconf:MovieTitles}';
```

# Authoring of Azure Data Factory (v2) – what's new?

The screenshot shows the Microsoft Azure Data Factory v2 interface. The top navigation bar includes 'Microsoft Azure' and 'Data Factory' with a dropdown arrow, followed by the path 'SQLPlayerDemo2'. A search bar on the right says 'Search resources'. Below the navigation is a toolbar with icons for 'Data Factory' (dropdown), 'Publish All', 'Validate All' (with a checkmark), 'Refresh', and 'Discard All'. On the left, a sidebar lists 'Factory Resources' with a search bar and a '+' button. It also shows counts for 'Pipelines' (2), 'Datasets' (12), and 'Data Flows (Preview)' (5). The 'Data Flows (Preview)' item is highlighted with an orange rounded rectangle and has an orange arrow pointing to it from the bottom right. To the right of the sidebar, three resource cards are displayed: 'CopyFlow X' (with a 'Debug' toggle switch), 'users X', and 'dstUsersBlob X'. Each card has a 'Validate' checkbox and a 'Source Settings' link.

# Simple Copy Flow



# Mapping Data Flow: Components = Actions \*

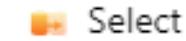
## Multiple inputs/outputs



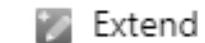
## Schema modifier



## Row modifier



## Custom

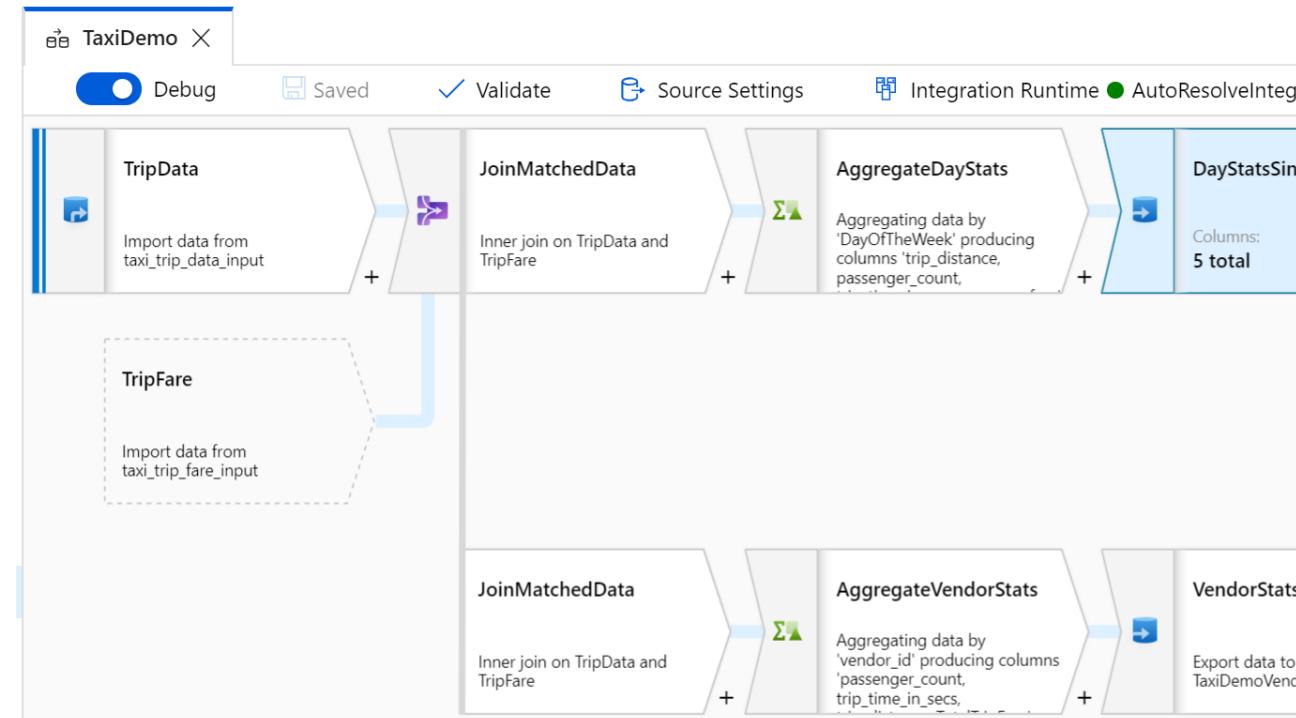
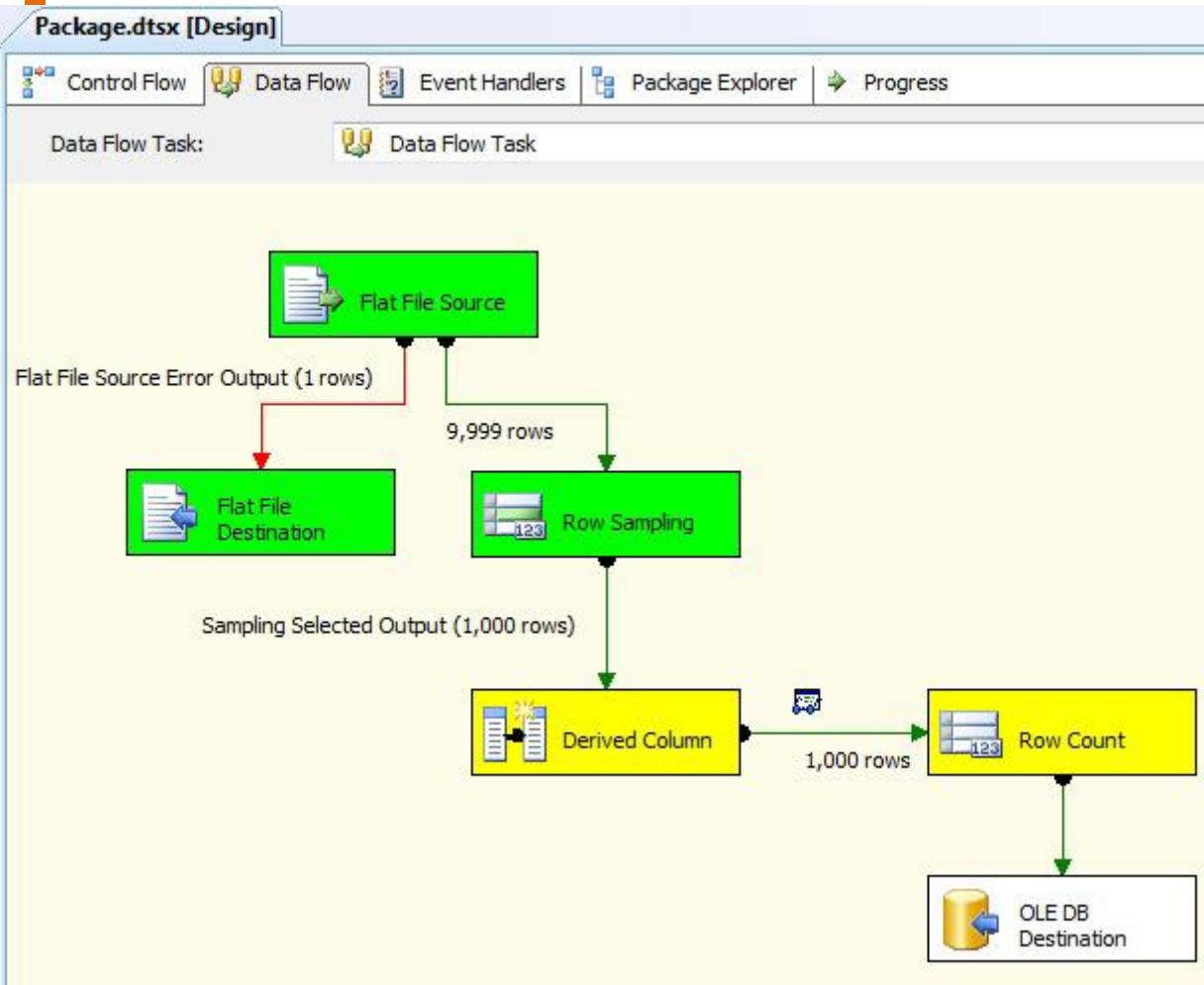


## Destination



\* With some small exceptions

# SSIS Data Flow VS ADF Mapping Data Flow



<https://www.red-gate.com/simple-talk/sql/ssis/debugging-data-flow-in-sql-server-integration-services/>

# Authoring of Azure Data Factory (v2)

Microsoft Azure

BigPlayer Data Factory ▾ Publish All ✓ Validate All Refresh Discard All ARM Template ▾

Factory Resources Filter Resources +

Pipelines ... 2 Datasets ... 9 Badges BadgesBlob BadgesBlobWithHeader BadgesStatsByName BadgesStatsByNameBlob Crimes\_BlobCsv Src\_Users Users\_BlobCsv UsersTest

Data Flows ... 3 StackOverflow 3 badgesGroupName badgesGroupName2 users

users X

Debug ✓ Validate

sourceUsers Import data from Users\_BlobCsv

Select1 Renaming sourceUsers to Select1 with columns 'DisplayName, DownVotes, LastAccessDate, Location'

FilterByReputation Filtering rows using expressions on columns 'Reputation'

GroupByLocation Aggregating data by 'Location' producing columns 'SumOfReputation, SumOfViews, Count'

SortByLocation Sorting rows on columns 'Location'

Wrong Conditionally distributing the data in 2 groups, based on columns 'Location, Location, Location, Location, Location'

AllRight Conditionally distributing the data in 2 groups, based on columns 'Location, Location, Location, Location, Location'

General External dependencies

Name \* users

Description

```
graph LR; subgraph users [ ]; direction LR; sourceUsers --> Select1; Select1 --> FilterByReputation; FilterByReputation --> GroupByLocation; GroupByLocation --> SortByLocation; SortByLocation --> Wrong; SortByLocation --> AllRight; end; sourceUsers["sourceUsers<br/>Import data from Users_BlobCsv"]; Select1["Select1<br/>Renaming sourceUsers to Select1 with columns 'DisplayName, DownVotes, LastAccessDate, Location'"]; FilterByReputation["FilterByReputation<br/>Filtering rows using expressions on columns 'Reputation'"]; GroupByLocation["GroupByLocation<br/>Aggregating data by 'Location' producing columns 'SumOfReputation, SumOfViews, Count'"]; SortByLocation["SortByLocation<br/>Sorting rows on columns 'Location'"]; Wrong["Wrong<br/>Conditionally distributing the data in 2 groups, based on columns 'Location, Location, Location, Location, Location'"]; AllRight["AllRight<br/>Conditionally distributing the data in 2 groups, based on columns 'Location, Location, Location, Location, Location'"];
```

# Guided experience to build data flows

The screenshot shows the Microsoft Azure Data Factory Data Flow designer interface. A pipeline named "usersql" is displayed, consisting of several components: source1, Select1, FilterByReputation, GroupByLocation, Sort1, Filter1, and sink1. The "source1" component is currently selected, and a context menu is open over it, highlighted with a red box. The menu includes options like "Multiple inputs/outputs", "New Branch", "Join", "Conditional Split", "Union", "Lookup", "Schema modifier" (with sub-options "Derived Column", "Aggregate", "Surrogate Key", "Pivot", "Unpivot", and "Window"), and "Row modifier". Below the menu, the "Source Settings" tab is active, showing the configuration for the "source1" stream. The "Output stream name" is set to "source1", and the "Source Dataset" is "stack\_users". Other settings include "Allow schema drift" checked, "Sampling" set to "Enable", and a "Rows limit" of 1000.

Microsoft Azure | Data Factory > SQLPlayerDemo

Factory Resources <>

usersql X

source1

Select1

FilterByReputation

GroupByLocation

Sort1

Filter1

sink1

Search resources

Pipelines: 5

Datasets: 16

Data Flows (Preview): 6

usersql

Beta: 2

StackOverflow: 3

badgesGroupName

badgesGroupName2

users

Add Source

Multiple inputs/outputs

- New Branch
- Join
- Conditional Split
- Union
- Lookup
- Schema modifier
- Derived Column
- Aggregate
- Surrogate Key
- Pivot
- Unpivot
- Window

Source Settings

Output stream name \* source1

Source Dataset \* stack\_users

Options  Allow schema drift

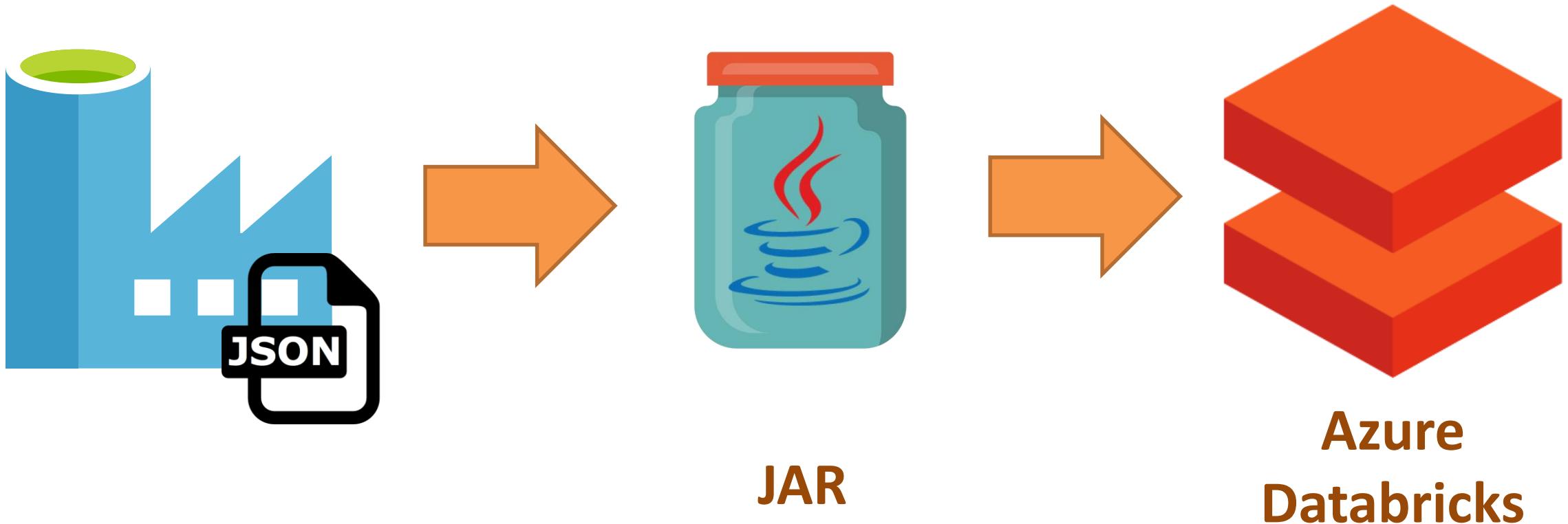
Sampling \*  Enable  Disable

Rows limit 1000

Connections

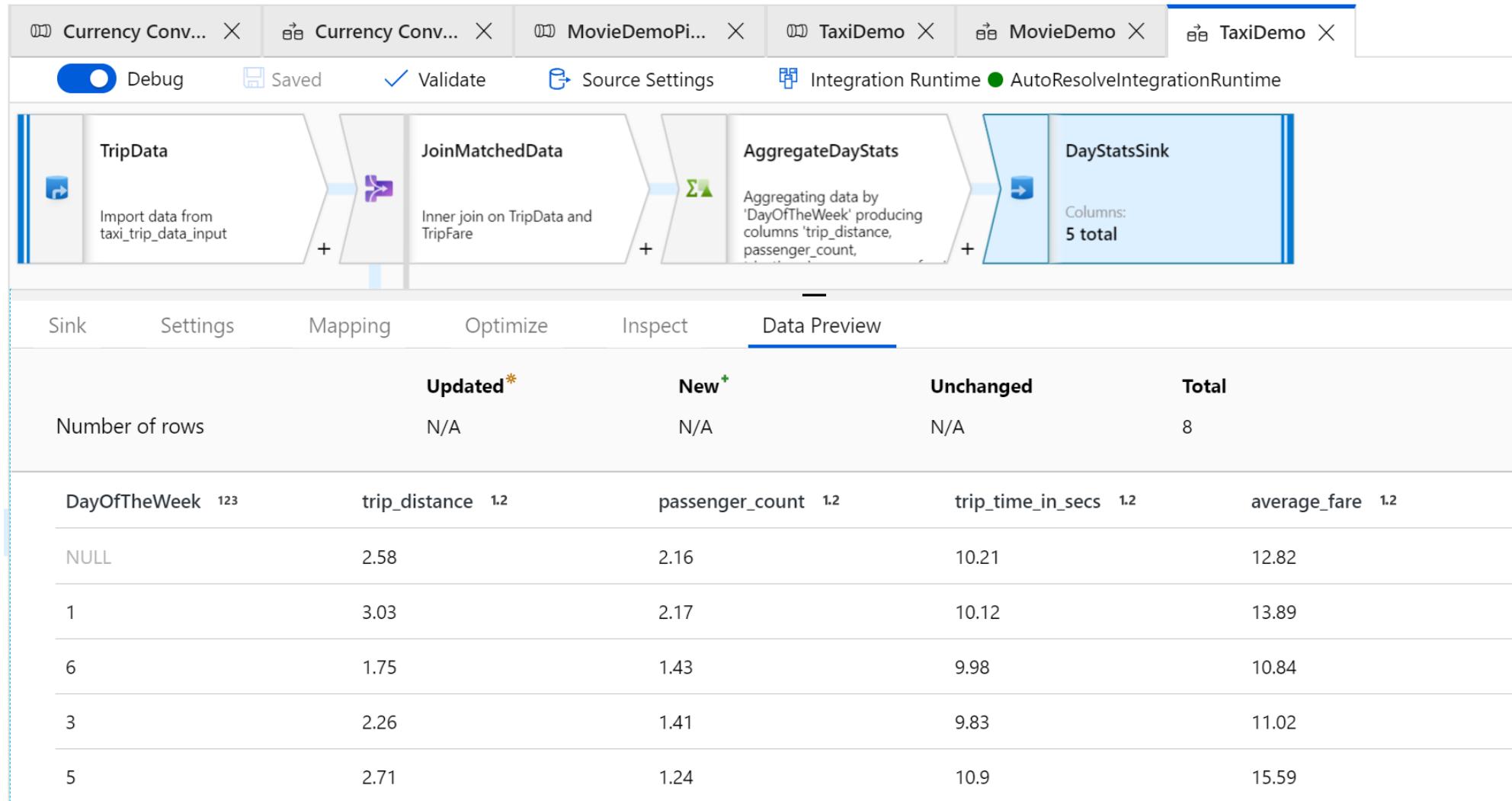
Triggers

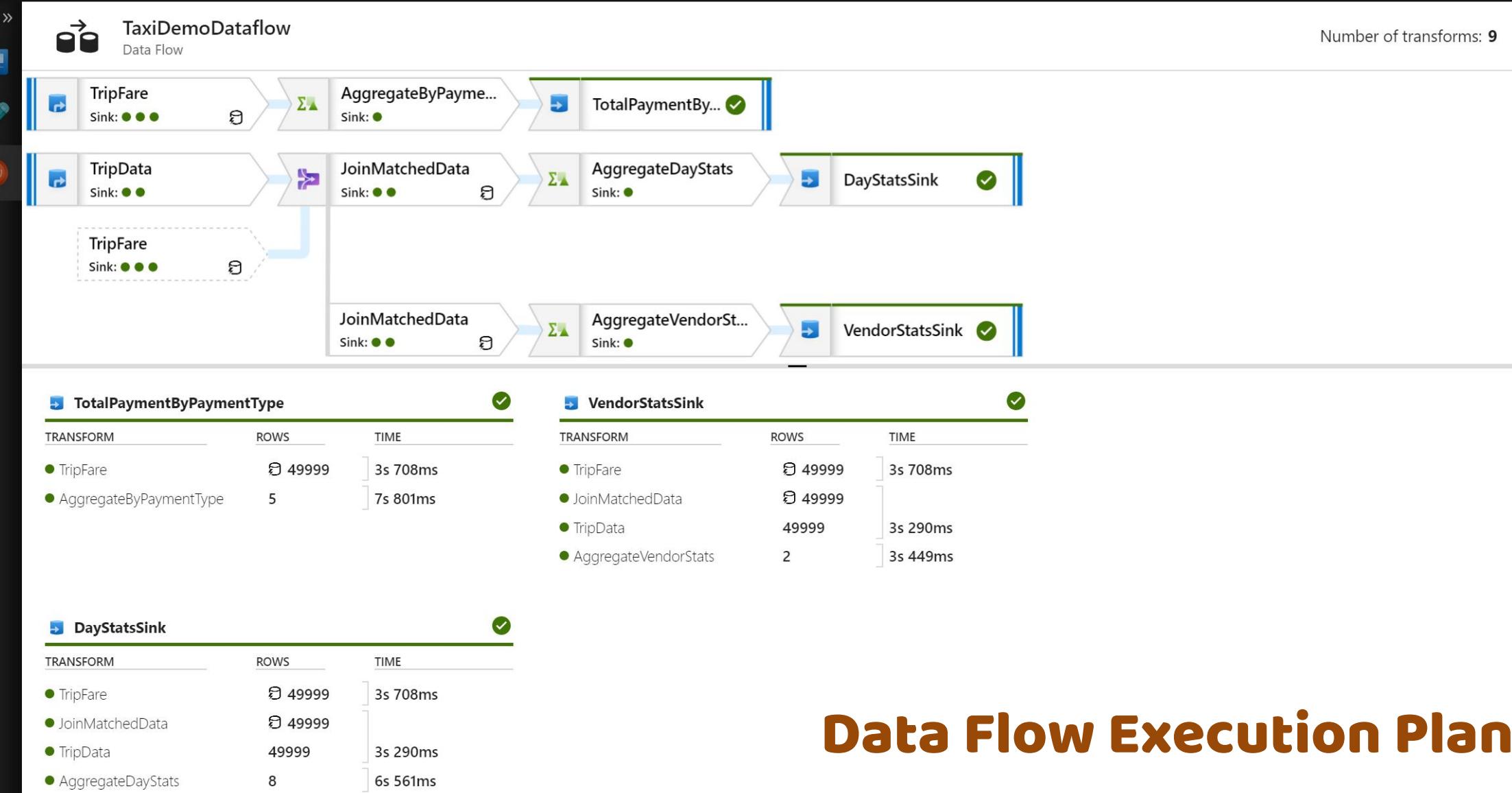
# What is going on behind the scenes?



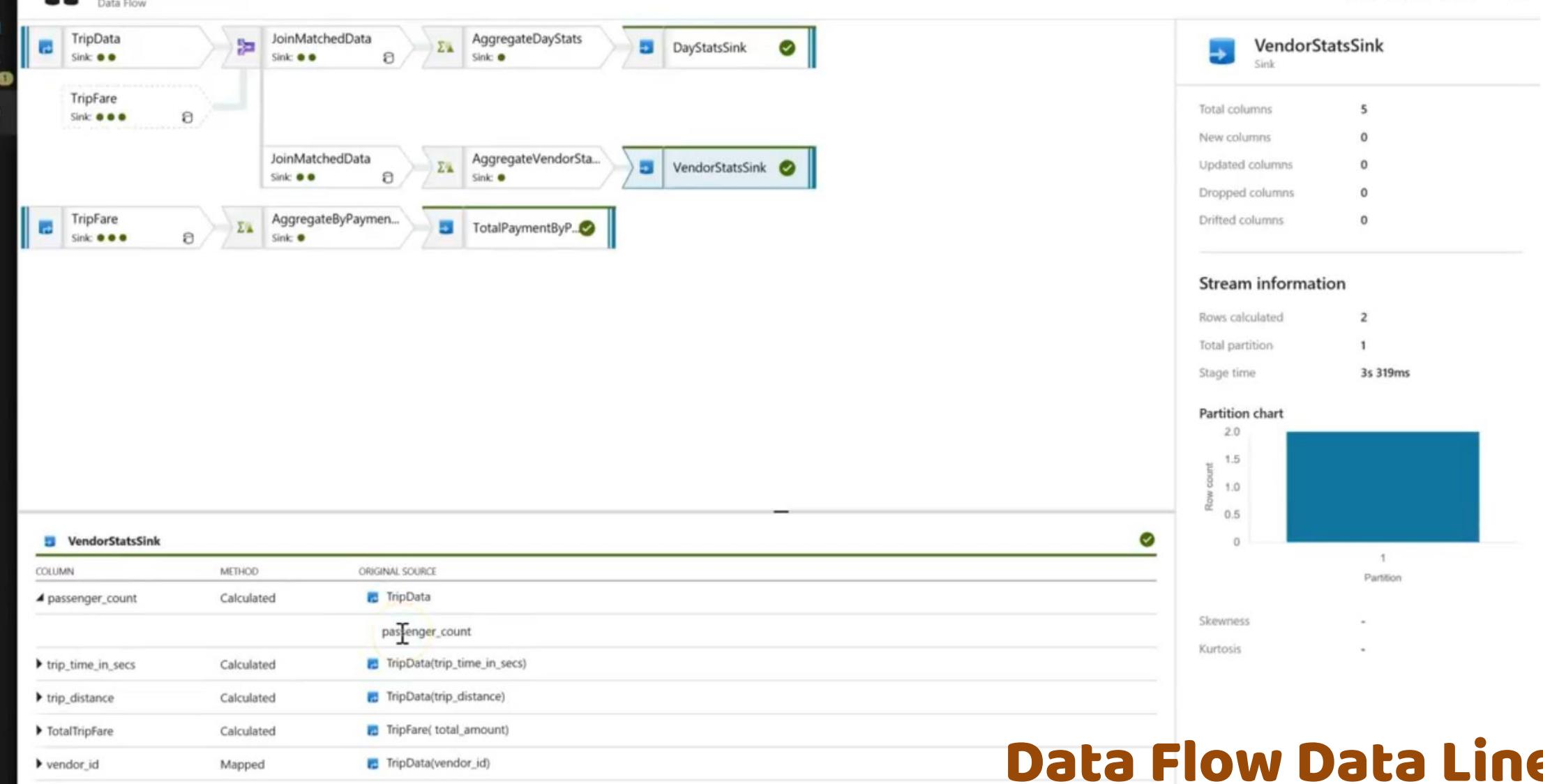
# DEMO TIME

# Data Preview in Debug mode





# Data Flow Execution Plan



# Data Flow Data Lineage

# ADF Template Gallery

The image shows two side-by-side screenshots of Azure Data Factory (ADF) interfaces. On the left is the 'Template gallery' interface, displaying a grid of 12 data transfer templates. On the right is the 'Factory Resources' interface, showing the 'Pipeline' blade with a 'Pipeline from template' option highlighted.

**Template gallery:**

- Bulk Copy from Database to Azure Data Explorer**: Copy large amounts of data from databases like SQL Server, Google BigQuery to Azure Data Explorer (ADX).
- Bulk Copy from Database**: Copy data in bulk from database using external control table to store partition list of source tables.
- Copy data from Google BigQuery to Azure Data Lake Store**: Copy data from Google BigQuery to Azure Data Lake Storage.
- Copy data from HDFS to Azure Data Lake Store**: Copy data from HDFS to Azure Data Lake Storage.
- Copy data from Netezza to Azure Data Lake Store**: Copy data from Netezza server to Azure Data Lake Storage.
- Copy data from on premise SQL Server to SQL Azure**: Copy data from on-premise SQL Server to SQL Azure.
- Copy data from on premise SQL Server to SQL Data Warehouse**: Copy data from on-premise SQL Server to SQL Data Warehouse.
- Copy data from Oracle to SQL Data Warehouse**: Copy data from Oracle server to SQL Data Warehouse.
- Copy delta data from AWS S3 to Azure Data Lake Storage Gen2**: Copy delta data from AWS S3 to Azure Data Lake Storage Gen2.
- Copy multiple files containers between File Stores**: Copy all files from AWS S3 to...
- Copy new files only by LastModifiedDate**: Copy new and changed files only by using LastModifiedDate.
- Data Flow Search Log Analytics**: Sample U-SQL SearchLog analytics example turned into an ADF Data Flow.

**Factory Resources - Pipeline:**

- Pipeline**
- Pipeline from template** (highlighted with a red circle and arrow)

# Mapping Data Flow – Source & Sink



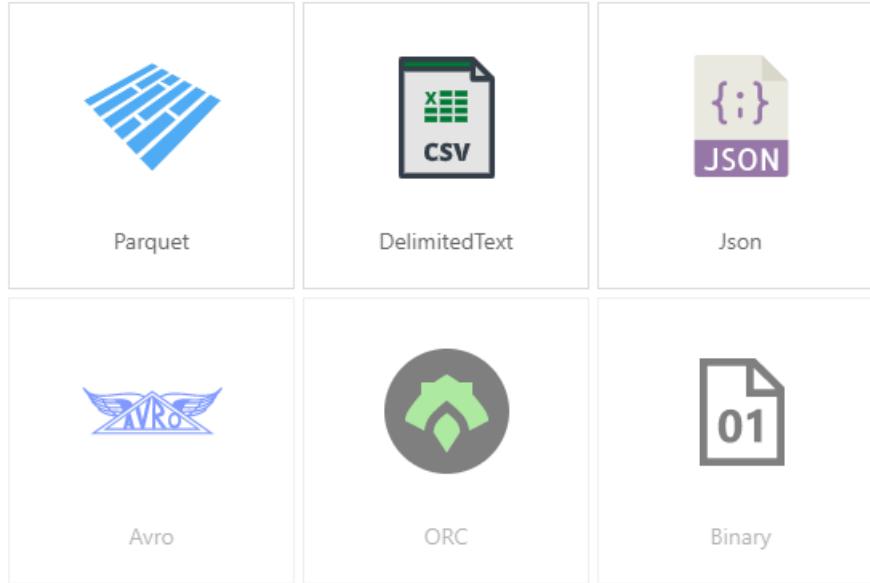
# Mapping Data Flow – Source & Sink capabilities



- New capabilities for Source transformations:
  - wildcards, file sets,
  - move file / Delete file,
  - auto-detect types,
  - schema validation
  - query statement
- New capabilities for Sink transformations:
  - output to single file,
  - clear folder,
  - truncate table / recreate table,
  - naming patterns



# Mapping Data Flow – DataSet File Formats



Available NOW

Available SOON

# Mapping Data Flow – Execution Settings

- The Execute Data Flow transformation:
  - Support **parameterized datasets**
  - Control **size of cluster** for specific Azure IR
  - Define **TTL (Time-To-Live)** to Azure IR to reduce data flow activity time



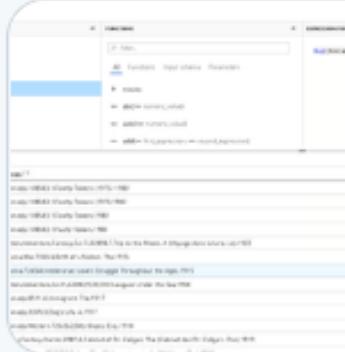
The screenshot shows the Azure Data Factory UI for configuring an Execute Data Flow transformation. On the left, under the 'Settings' tab, the 'Run on (Azure IR)' dropdown is set to 'AutoResolveIntegrationRuntime'. An orange arrow points from this dropdown to a detailed configuration pane on the right. This pane includes:

- Region \***: Auto Resolve
- Data Flow run time**
- Compute type \***: General Purpose
- Core count \***: 4 (+ 4 Driver cores)
- Time to live**: 0 minutes (selected, highlighted with a blue dashed border)
- Filter... button
- Other options: 10 minutes, 30 minutes, 1 hour, 4 hours

# Latest updates? Go Twitter!

 **Mark Kromer** @KromerBigData · 1h

#Azure #datafactory has released JSON and hierarchical transformation capabilities via #mappingdataflows. You can now read & write JSON datasets in Data Flow, build and manage complex data structures, and transform hierarchical data.



ADF Adds Hierarchical & JSON Data Transformations...  
The Azure Data Factory team has released JSON and hierarchical data transformations to Mapping Data ...  
[🔗 techcommunity.microsoft.com](https://techcommunity.microsoft.com/t5/azure-data-factory/announcing-hierarchical-and-json-data-transformations-in-mapping-data-flow/ba-p/1008492)

8 12 14

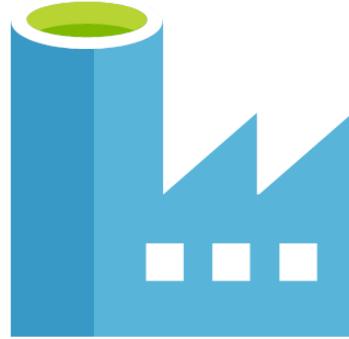
# SSIS vs ADF activities vs T-SQL

| Activity  | Description   | SSIS equivalent   | SQL Server equivalent   |
|---|---|---|---|
|  New branch        | Create a new flow branch with the same data           |  Multicast (+icon) | <code>SELECT INTO<br/>SELECT OUTPUT</code>  |
|  Join              | Join data from two streams based on a condition       |  Merge join        | <code>INNER   LEFT   RIGHT JOIN,<br/>CROSS   FULL OUTER JOIN</code>                                   |
|  Conditional Split | Route data into different streams based on conditions |  Conditional Split | <code>SELECT INTO WHERE condition1<br/>SELECT INTO WHERE condition2<br/>CASE ... WHEN</code>          |
|  Union             | Collect data from multiple streams                    |  Union All         | <code>SELECT colla UNION (ALL)<br/>SELECT collb</code>  |
|  Lookup            | Lookup additional data from another stream            |  Lookup            | <code>LEFT   RIGHT JOIN</code>  |
|  Derived Column    | Compute new columns based on the existing once        |  Derived Column    | <code>SELECT Column1 * 1.09 as NewColumn</code>   |
|  Aggregate       | Calculate aggregation on the stream                   |  Aggregate       | <code>SELECT Year(DateOfBirth) as Year,<br/>MIN(), MAX(), AVG()<br/>GROUP BY Year(DateOfBirth)</code> |

<http://bit.ly/ADFDFvsSSIS>

<http://bit.ly/ADFDF-CheatSheet>

## Resources



<http://sqlplayer.net/ADF>

# Q&A



**Muchas gracias!**

**Thank you!**



kamil@nowinski.net



@NowinskiK

@SQLPlayer



SQLPlayer.net



<https://github.com/NowinskiK/CommunityEvents>



## Kamil Nowinski

Microsoft Data Platform MVP

MCSE Data Platform & MCSE Data Management and Analytics