

Azure Data Factory: Mapping Data Flow – first blood

Kamil Nowiński

About me

Kamil Nowinski



Microsoft
CERTIFIED
Solutions Associate
SQL Server 2012



Microsoft Data Platform **MVP**
Speaker, blogger, data enthusiast
Senior Data Engineer at ASOS (www.asos.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)

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

BLOG & Interviews

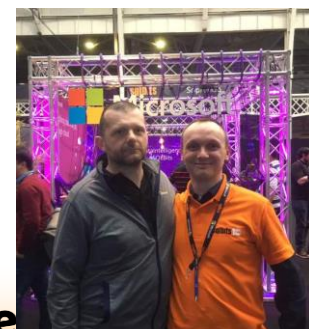
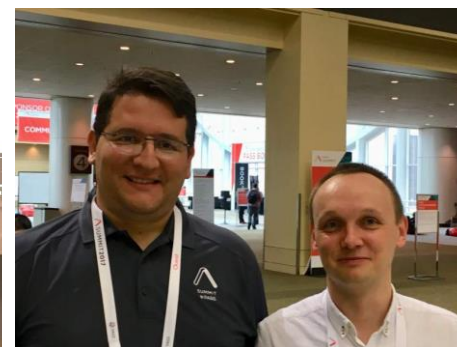
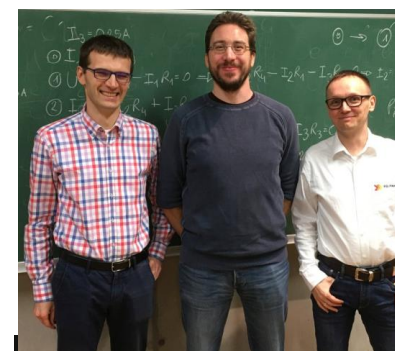
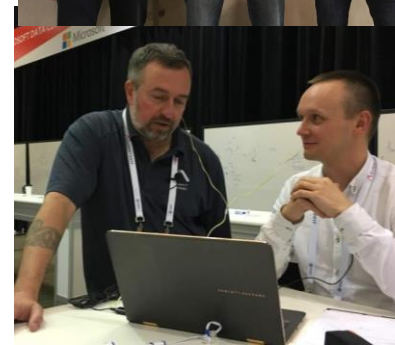


www.SQLPlayer.net

Ask SQL Family – episodes to date



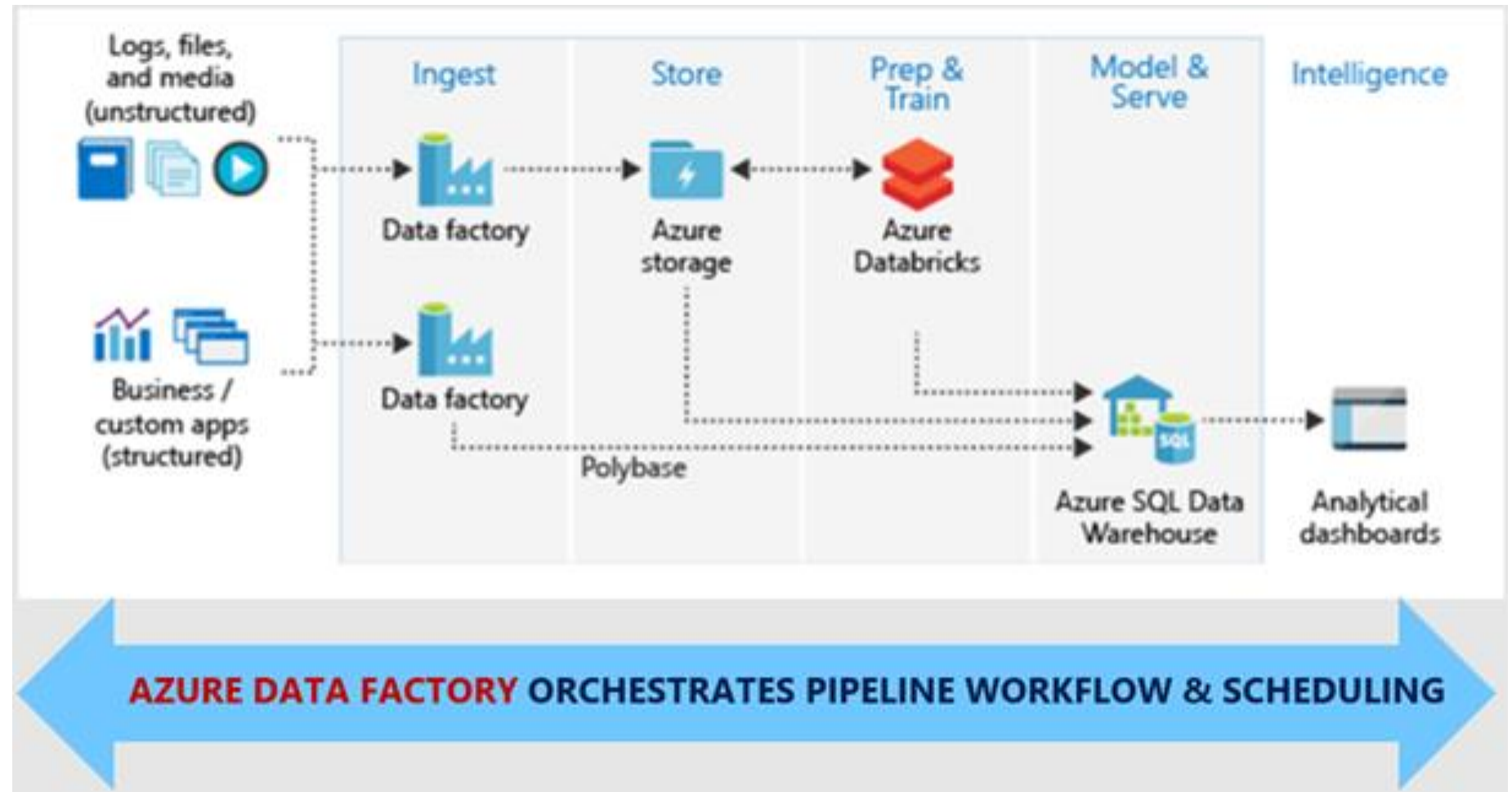
Scan me



Play with data & have fun!

Azure Data Factory. Mapping Data Flow. First blood

What the Azure Data Factory is?

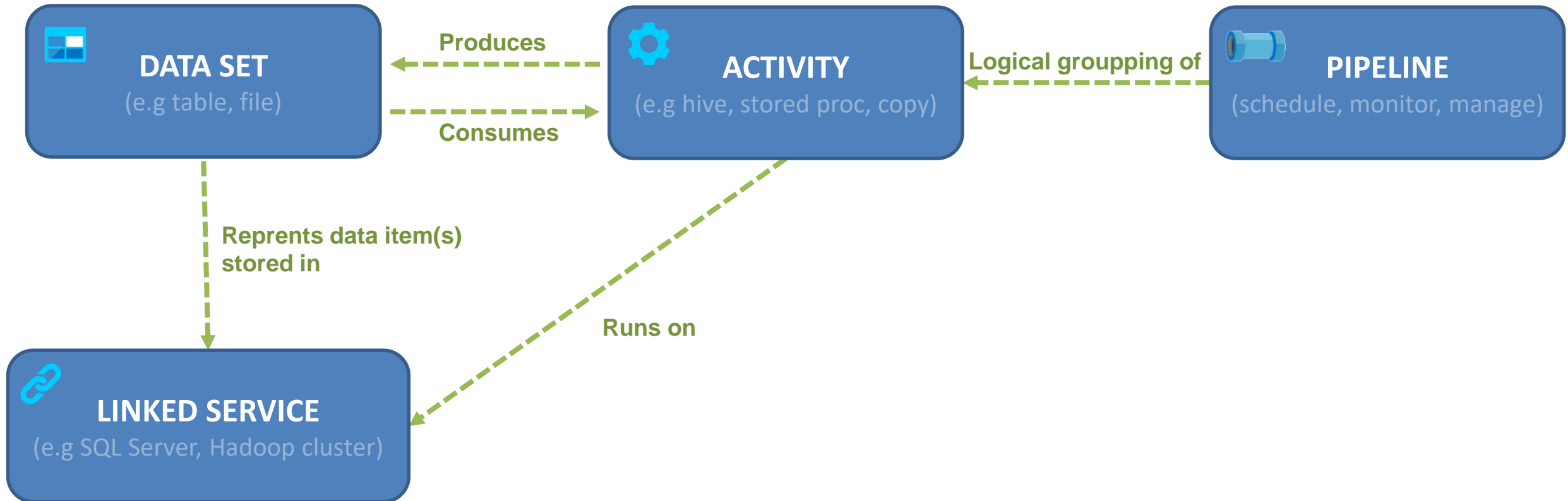


Access all your data

- 80+ connectors & growing
- Azure IR available in 20 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	
					Oracle Service Cloud	Zoho	
	Supported as Source and Sink						
	Supported as Source only						
	Supported as Sink only						

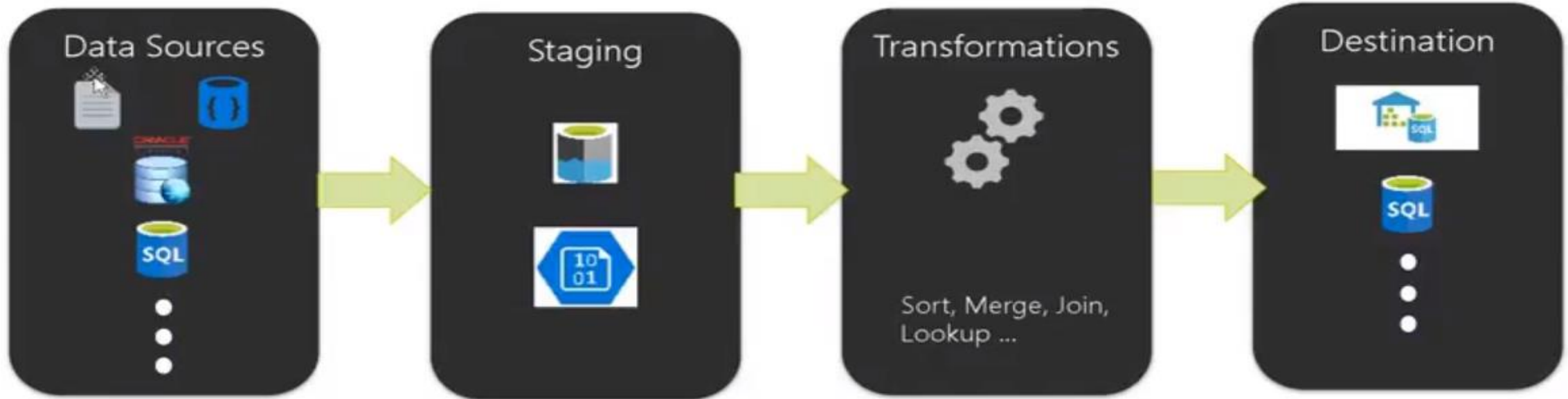
ADF Key Concepts



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

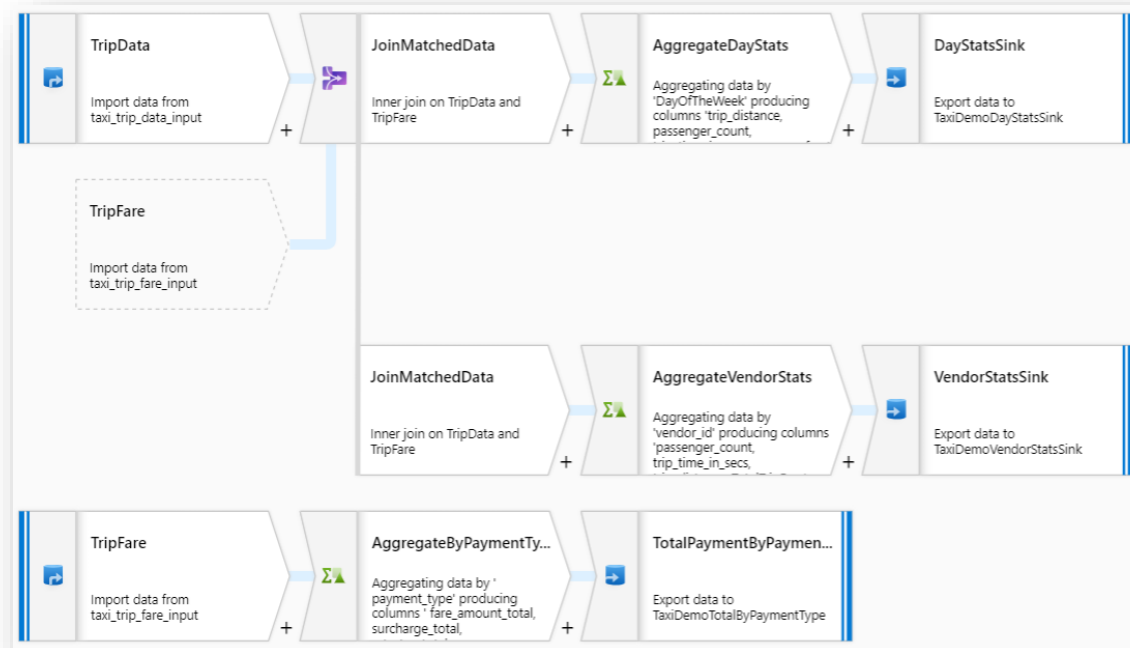
- Implicit/Explicit
- Data Lake staging area as default
- User does not need to configure this manually
- Advanced feature to set staging area options
- File formats/types (Parquet, JSON, txt, CSV, ...)

- 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

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



... not

```
1
2
3 HDI Cluster Details:
4 AdfHdi.azurehdinsight.net
5 Admin
6 Adf@123456
7
8 Storage:
9 adfhdistorage
10 /any?wp661j7f811BwmiSo/YGdJyG7Ad+S1JAr+SnS7b3g954706gK0K0ksZ19U0ut40z28x10WdMwQ==
11
12 Cluster Remote Login Details:
13 Adf
14 India@1234
15
16 HiveQuery:
17 DROP TABLE IF EXISTS MovieRatings;
18 CREATE EXTERNAL TABLE MovieRatings
19 (
20   UserID int,
21   MovieID int,
22   Rating int,
23   TimeStamp string
24 ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE LOCATION '${hiveconf:MovieRatings}';
25
26 DROP TABLE IF EXISTS MovieTitles;
27 CREATE EXTERNAL TABLE MovieTitles
28 (
29   MovieID int,
30   MovieName string
31 ) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE LOCATION '${hiveconf:MovieTitles}';
```

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

Microsoft Azure | Data Factory ► SQLPlayerDemo2

Search resources

» Data Factory ▼ Publish All ✓ Validate All Refresh Discard All

Factory Resources ≡ «

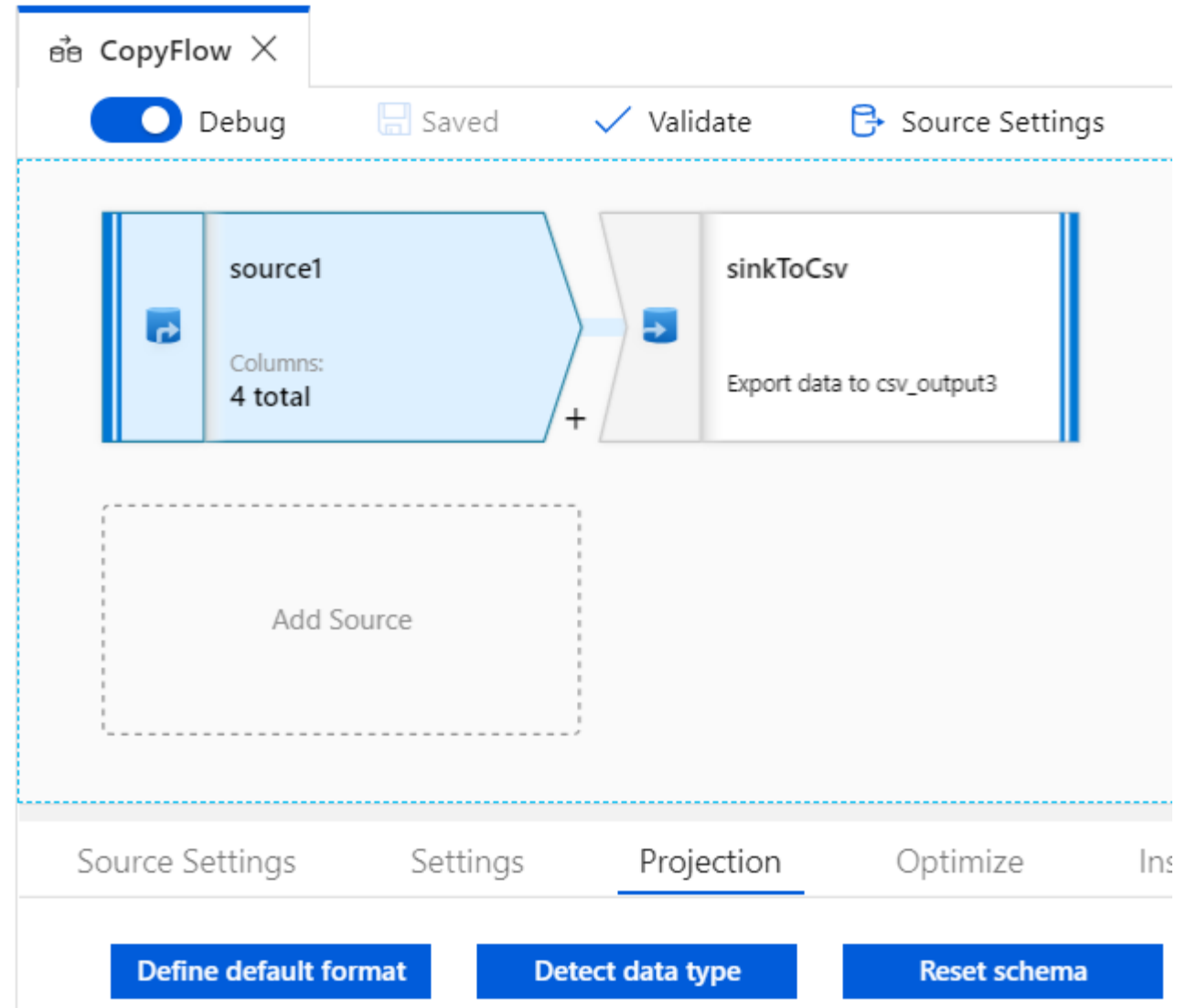
Filter resources by name +

Pipelines	2
Datasets	12
Data Flows (Preview)	5

CopyFlow × users × dstUsersBlob ×






Debug ✓ Validate Source Settings

Simple Copy Flow









Mapping Data Flow: Components = Actions *





Multiple inputs/outputs

-  New Branch
-  Join
-  Conditional Split
-  Union
-  Lookup


Schema modifier

-  Derived Column
-  Aggregate
-  Surrogate Key
-  Pivot
-  Unpivot
-  Window


Row modifier

-  Exists
-  Select
-  Filter
-  Sort

Custom

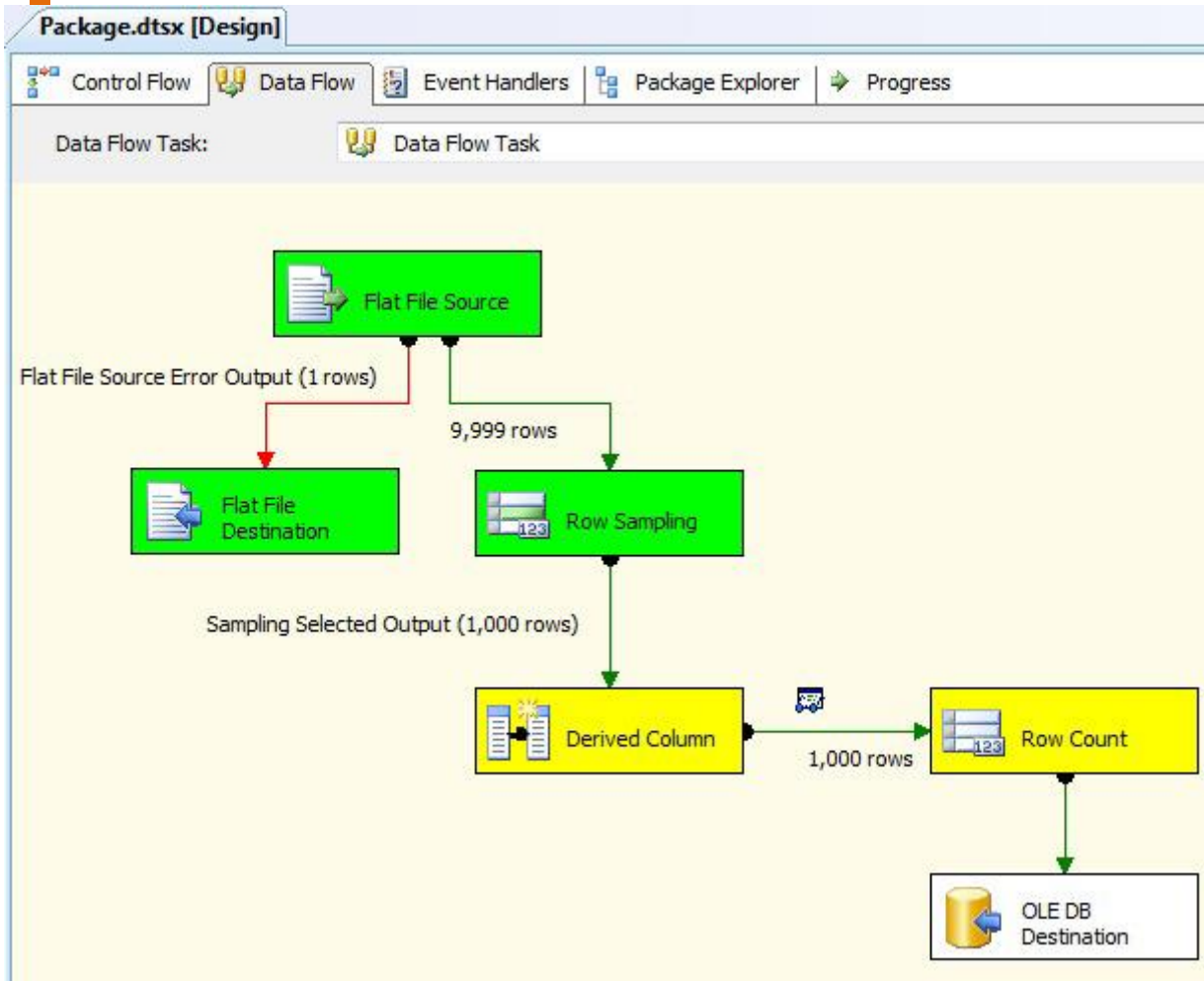
-  Extend

Destination

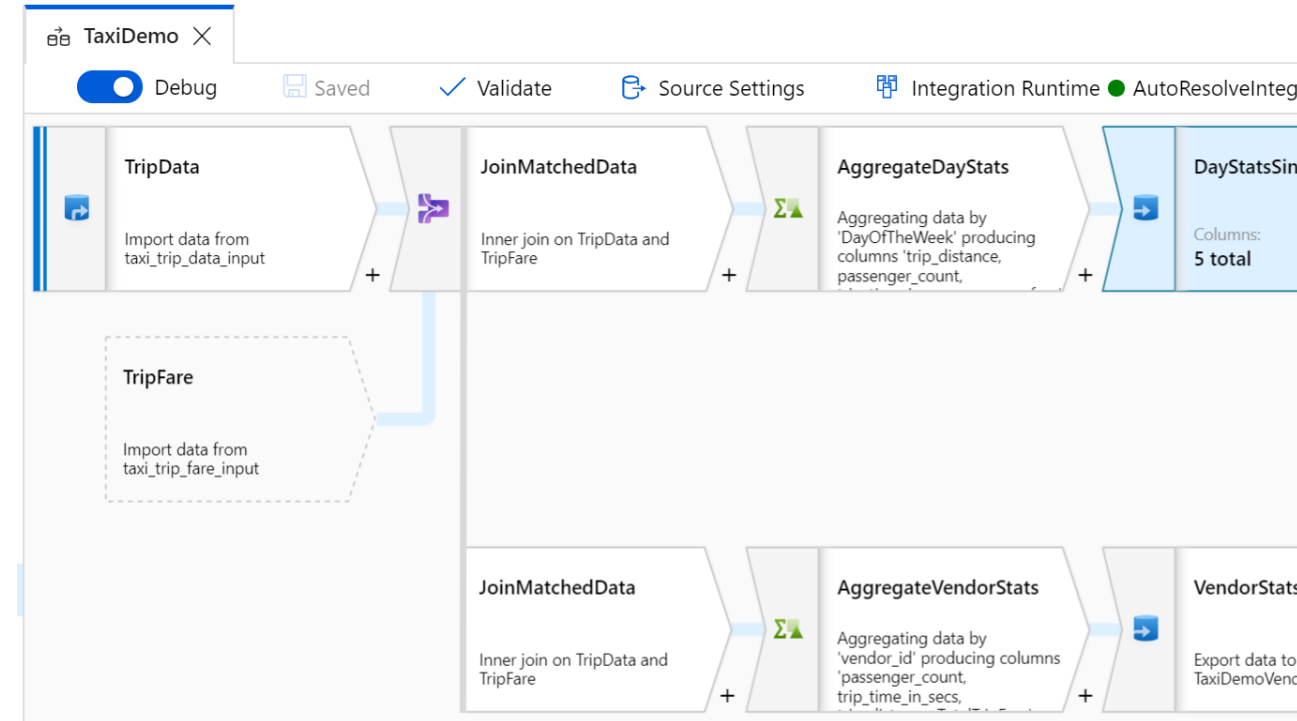
-  Sink

* 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

Search resources

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
 - badgesGroupByName
 - badgesGroupByName2
 - users

users

Debug Validate

sourceUsers

Import data from Users_BlobCsv

Select1

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

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 *

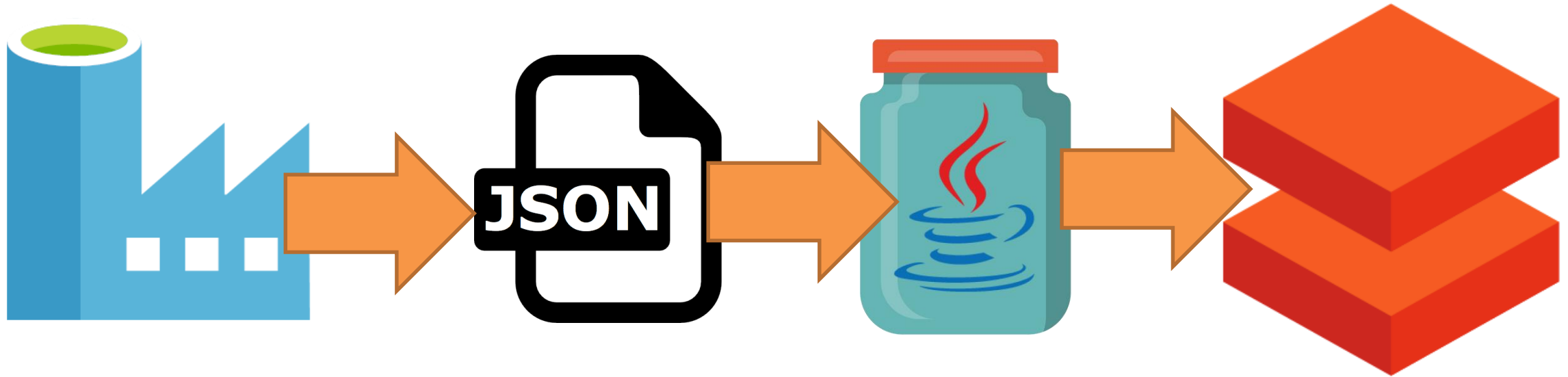
users

Description

Guided experience to build data flows

The screenshot displays the Microsoft Azure Data Factory interface for a workspace named 'SQLPlayerDemo'. The top navigation bar includes options like 'Publish All', 'Validate All', 'Refresh', 'Discard All', and 'ARM Template'. On the left, the 'Factory Resources' pane lists 'Pipelines' (5), 'Datasets' (16), and 'Data Flows (Preview)' (6). The 'usersql' data flow is selected, showing a pipeline with steps: source1, Select1, FilterByReputation, GroupByLocation, Sort1, Filter1, and sink1. A context menu is open over the 'source1' step, listing various actions under 'Multiple inputs/outputs' (New Branch, Join, Conditional Split, Union, Lookup) and 'Schema modifier' (Derived Column, Aggregate, Surrogate Key, Pivot, Unpivot, Window). The bottom pane shows the 'Source Settings' for 'source1', including 'Source Dataset' (stack_users), 'Options' (Allow schema drift checked), 'Sampling' (Enable selected), and 'Rows limit' (1000).

What is going on behind the scenes?



JAR

**Azure
Databricks**

DEMO TIME

Data Preview in Debug mode

Currency Conv... X Currency Conv... X MovieDemoPi... X TaxiDemo X MovieDemo X TaxiDemo X

Debug Saved Validate Source Settings Integration Runtime AutoResolveIntegrationRuntime

TripData
Import data from taxi_trip_data_input

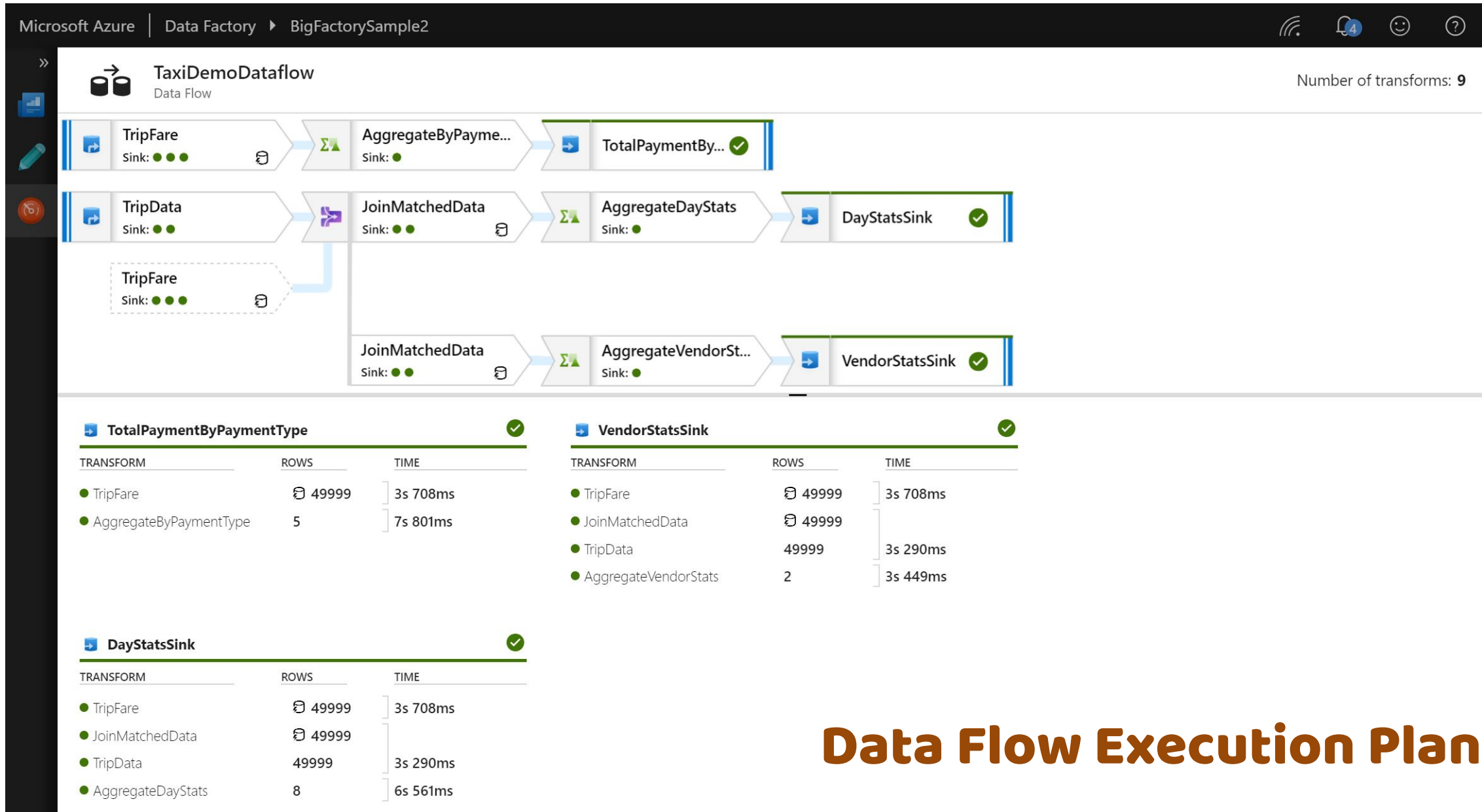
JoinMatchedData
Inner join on TripData and TripFare

AggregateDayStats
Aggregating data by 'DayOfTheWeek' producing columns 'trip_distance', 'passenger_count', 'trip_time_in_secs', 'average_fare'

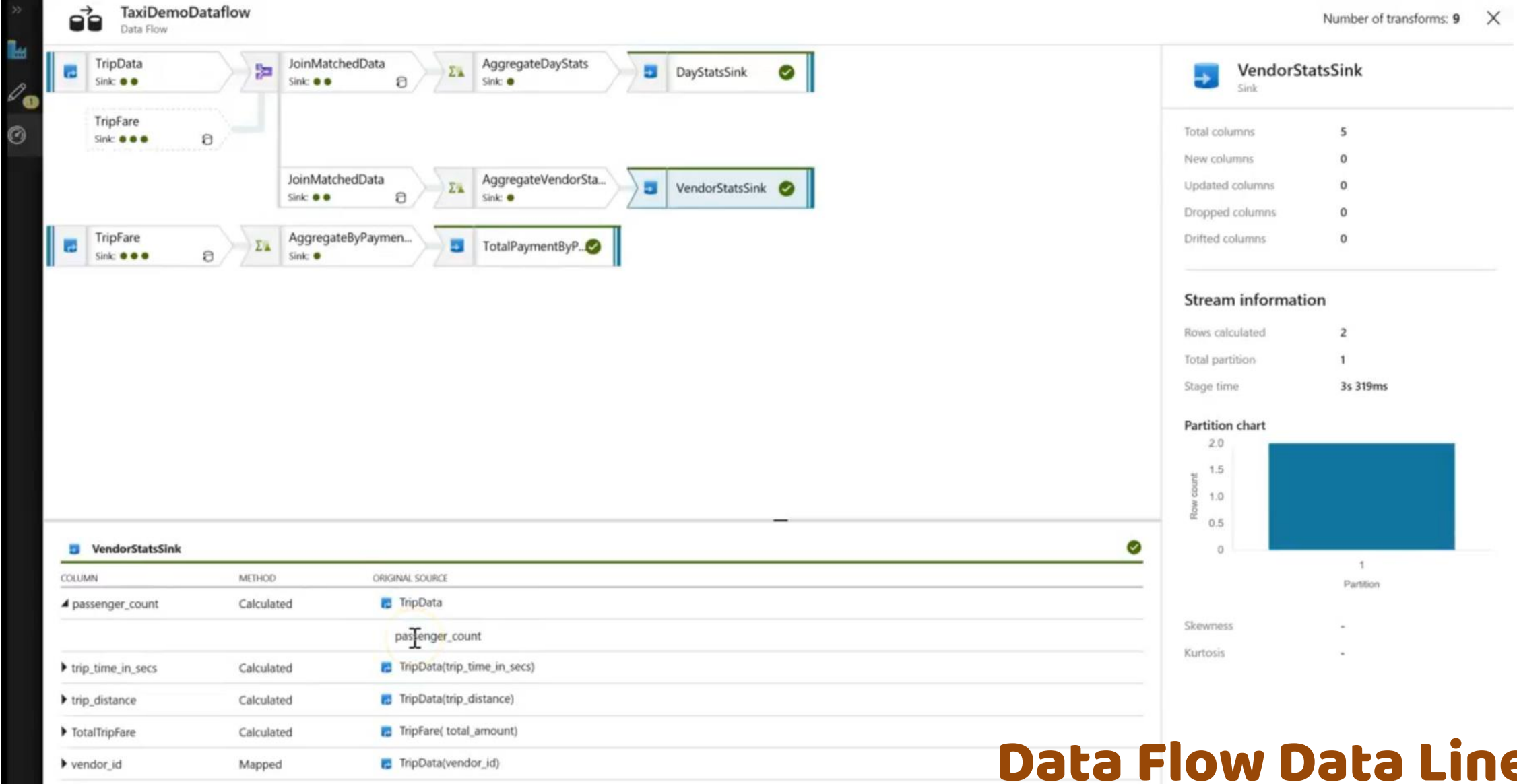
DayStatsSink
Columns: 5 total

Sink Settings Mapping Optimize Inspect Data Preview

	Updated*	New+	Unchanged	Total
Number of rows	N/A	N/A	N/A	8
DayOfTheWeek 123	trip_distance 1.2	passenger_count 1.2	trip_time_in_secs 1.2	average_fare 1.2
NULL	2.58	2.16	10.21	12.82
1	3.03	2.17	10.12	13.89
6	1.75	1.43	9.98	10.84
3	2.26	1.41	9.83	11.02
5	2.71	1.24	10.9	15.59



Data Flow Execution Plan

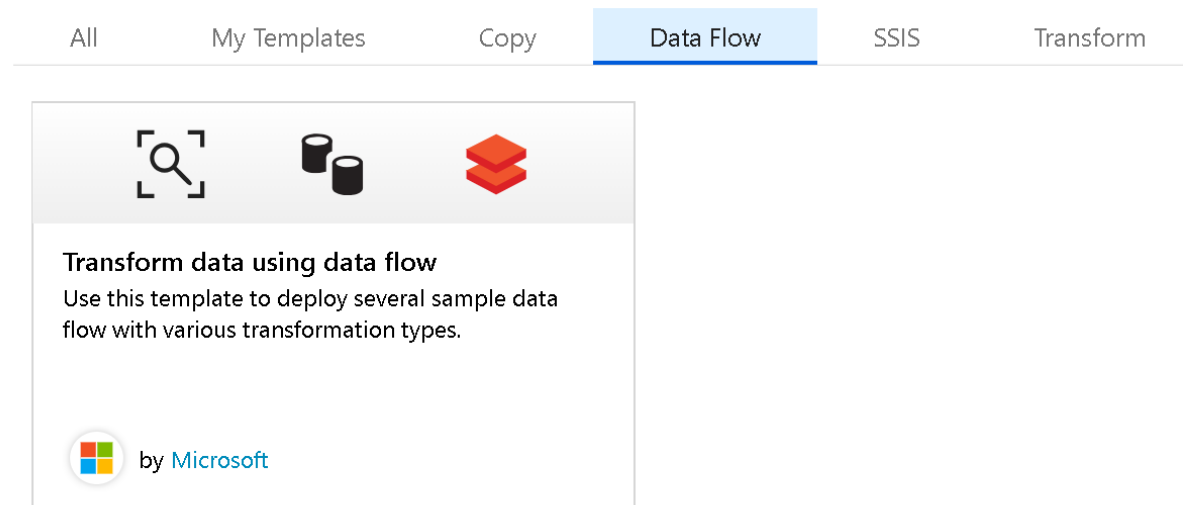


Data Flow Data Lineage

TAKEAWAYS

Mapping Data Flow – the latest update

- The Preview version of ADF with Data Flows has been deprecated (26 February)
- You will no longer need to stand-up Azure Databricks clusters. ADF will handle cluster management for you on-demand.
- Data Flow samples have been into the new ADF Template Gallery



Mapping Data Flow – the latest update for v2



- 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 – New Datasets



- New datasets for Data Flow (only):
 - Parquet
 - Delimited Text

General Settings User Properties

Data Flow * dataflow2 Edit + New

source1

NAME	VALUE
mypath	@pipeline0.parameters.mypath

Run on * AutoResolveIntegrationRuntime

Compute Type * General Purpose















Core count * 8

Staging linked service Select... + New

Staging storage folder container/folder Browse

- The Execute Data Flow transformation:
 - Now support **parameterized datasets**
 - Control size of cluster for specific data flow execution

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 SELECT OUTPUT</code>
 Join	Join data from two streams based on a condition	 Merge join	<code>INNER LEFT RIGHT JOIN, CROSS FULL OUTER JOIN</code>
 Conditional Split	Route data into different streams based on conditions	 Conditional Split	<code>SELECT INTO WHERE condition1 SELECT INTO WHERE condition2 CASE ... WHEN</code>
 Union	Collect data from multiple streams	 Union All	<code>SELECT col1a UNION (ALL) SELECT col1b</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, MIN(), MAX(), AVG() GROUP BY Year(DateOfBirth)</code>

<http://bit.ly/ADFDFvsSSIS>

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

Resources

- Microsoft Azure Data Factory – [Tutorials & API Reference](#)
- Azure Data Factory [Overview](#)
- Azure Data Factory – [Data integration service](#)
- ADF Mapping Data Flow's [documentation](#)
- ADF Mapping Data Flow's [videos](#)
- SQLPlayer blog:
 - [Azure Data Factory v2 and its available components in Data Flows](#)
 - Follow this tag on SQLPlayer blog: [#ADFDF](#)

Q&A



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