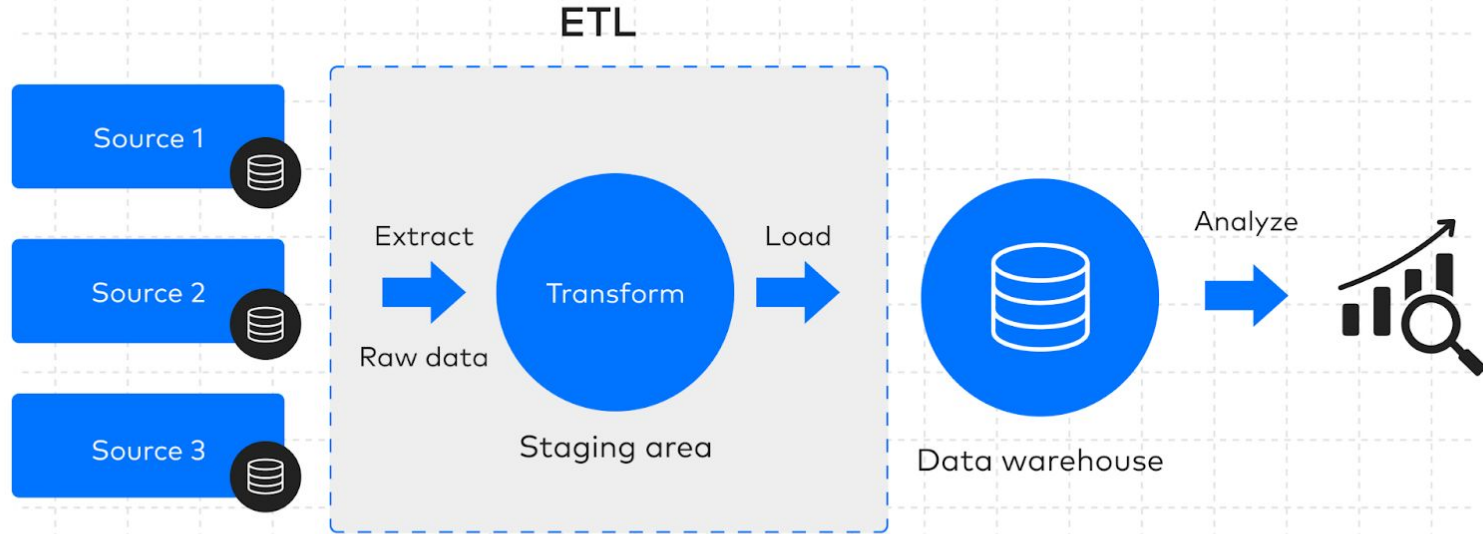


# Visual ETL with Mapping Data Flows in Azure Data Factory

**Antony Prince J**

# ETL in data pipeline

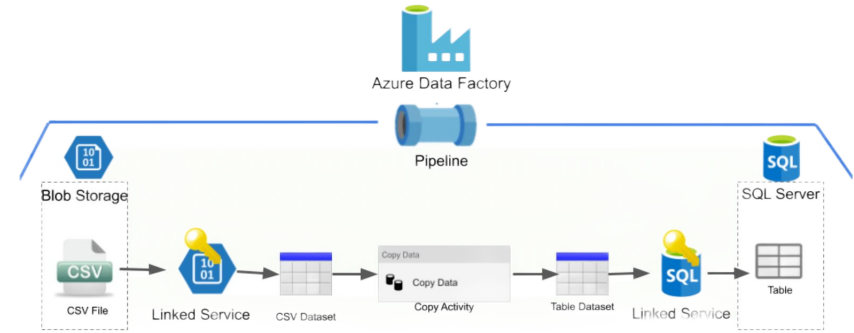


Reference - <https://www.fivetran.com/learn/data-pipeline-vs-etl>

# Azure Data Factory

Azure Data Factory (ADF) is a cloud-based ETL and data integration service data integration service provided by Microsoft as part of its Azure cloud platform.

Create, schedule, and orchestrate data workflows and pipelines for moving, transforming, and integrating data from various sources to desired destinations.



Reference

<https://www.productiveedge.com/blog/azure-data-factory-capabilities>

# CodeFree ETL as a service

## Ingest



- Multi-cloud and on premise hybrid copy data
- 100+ native connectors
- Serverless and auto scale
- Use wizard for quick copy jobs

## Control Flow



- Design codefree data pipelines
- Generate pipelines via SDK
- Utilize workflow constructs: loops, branches, conditional execution, variables, parameters, ...

## Data Flow



- Codefree data transformations that execute in Spark
- Scaleout with Azure Integration Runtimes
- Generate data flows via SDK
- Designers for data engineers and data analysts

## Schedule



- Build and maintain operational schedules for your data pipelines
- Wall clock, eventbased, tumbling windows, chained

## Monitor



- View active executions and pipeline history
- Detail activity and data flow executions
- Establish alerts and notifications



Home



Author



Monitor



Manage



Learning Center

Data factory

## adf-geeknight-demo

New ▾



## Ingest

Copy data at scale once or on a schedule.



## Orchestrate

Code-free data pipelines.



## Transform data

Transform your data using data flows.



## Configure SSIS

Manage &amp; run your SSIS packages in the cloud.

## Recent resources

Name	Type	Last opened by you
 <a href="#">df_restaurant_reviews</a>	Data flow	10 minutes ago
 <a href="#">ds_sink_sqldb</a>	Dataset	34 minutes ago
 <a href="#">pl_process_restaurant_reviews</a>	Pipeline	an hour ago
 <a href="#">ds_source_csv</a>	Dataset	11 hours ago

Show more ▾

## Discover more

Browse partners  
(preview)

Pipeline templates

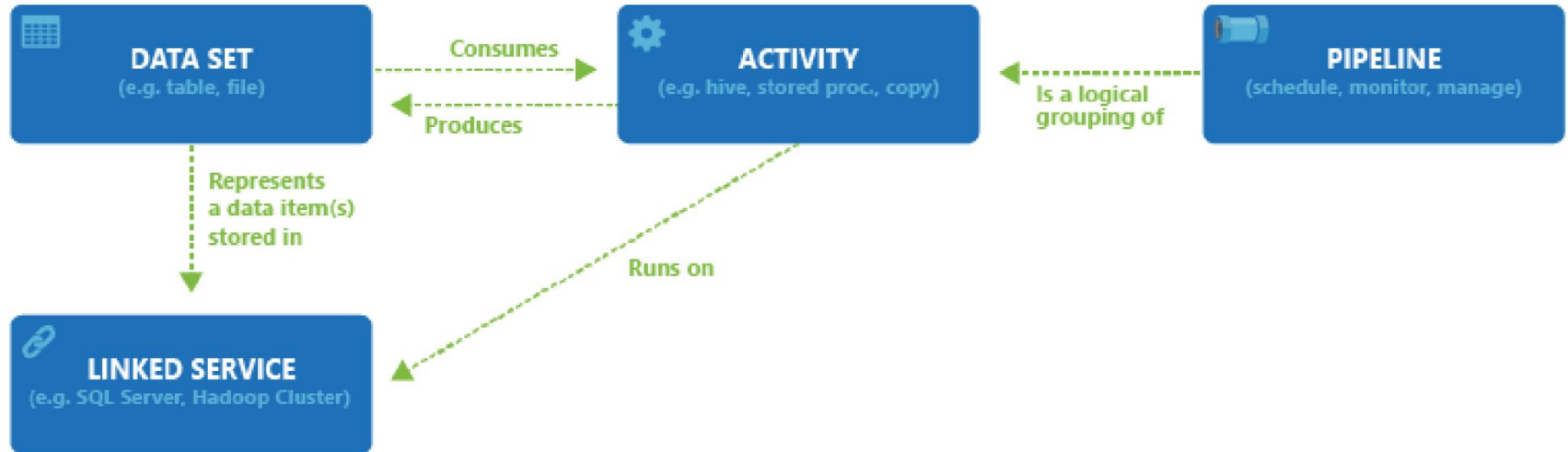


SAP pipeline templates

# USE CASES

Banking & Finance	Consolidate transaction, credit score, and loan data for risk assessment and regulatory compliance.
Retail & Ecommerce	Merge online and in-store sales data to analyze customer buying patterns and optimize inventory.
Healthcare	Aggregate insurance claims, and clinical trial data to support research, and comply with health data regulations.

# AZURE DATA FACTORY COMPONENTS



# LINKED SERVICES

Linked services are connections to external data stores or compute resources. They define the connection information needed to connect to external sources.

Would you like to see Data Factory inside of Microsoft Fabric, Microsoft's newest cloud-first data analytics SaaS platform? Click [here](#) to get started with Fabric Data Factory!

adf\_dev branch Validate all Save all Publish

General

Factory settings

Connector upgrade advisor

Connections

**Linked services**

Integration runtimes

Microsoft Purview

Source control

Git configuration

ARM template

Author

Triggers

Global parameters

Data flow libraries

Security

Credentials

Customer managed key

Outbound rules

Managed private endpoints

Workflow orchestration manager

Apache Airflow

**Linked services**

Linked service defines the connection information to a data store or compute. [Learn more](#)

+ New

Filter by name Annotations : Any

Showing 1 - 2 of 2 items

Name ↑↓	Type ↑↓
ls_adls_storage	Azure Data Lake Storage Gen2
ls_az_sqldb	Azure SQL Database

**Edit linked service**

Azure SQL Database [Learn more](#)

To avoid publishing immediately to Data Factory, please use Azure Key Vault to retrieve secrets securely. [Learn more here](#)

Name \*

ls\_az\_sqldb

Description

Connect via integration runtime \* ⓘ

AutoResolveIntegrationRuntime

Version

☒ 2.0 ☐ 1.0

Account selection method ⓘ

☐ From Azure subscription ☒ Enter manually

Fully qualified domain name \*

geeknight.database.windows.net

Database name \*

sqldb-geeknight

Authentication type \*

SQL authentication

User name \*

admin\_sa

Password

Azure Key Vault

Password \*

Apply

Cancel

[Test connection](#)



# DATASET

Dataset is a representation of the data you intend to work with in your data pipelines.

- Acts as a bridge between the source or destination data store and the ADF pipeline activities.
- Doesn't contain the data itself but references the data stored in various locations (e.g., databases, file systems, cloud storage).

The screenshot displays the Azure Data Factory (ADF) user interface. On the left, the 'Factory Resources' pane shows a hierarchical view of resources. Under 'Databases', the 'ds\_sink\_sqldb' resource is selected and highlighted. The main pane on the right shows the configuration for this dataset. It features a 'Saved' status indicator and a visual representation of an Azure SQL Database. Below this, the 'Connection' tab is active, showing the 'Linked service' as 'ls\_az\_sqldb'. The 'Table' field is configured with the path '@dataset().schema\_name' and '@dataset().table\_name'. A checkbox for 'Enter manually' is checked. Other options like 'Test connection', 'Edit', 'New', and 'Learn more' are visible.

**Factory Resources**

- Filter resources by name
- Pipelines: 1
  - demo: 1
- Change Data Capture (preview): 0
- Datasets: 2
  - demo: 2
  - ds\_sink\_sqldb**
  - ds\_source\_csv
- Data flows: 1
  - demo: 1
    - df\_restaurant\_reviews
- Power Query: 0
- Templates: 0

**ds\_sink\_sqldb**

Azure SQL Database  
**ds\_sink\_sqldb**

**Connection** Schema Parameters

Linked service \* ls\_az\_sqldb Test connection Edit + New Learn more

Table @dataset().schema\_name . @dataset().table\_name Preview data

☒ Enter manually

# ACTIVITY

Activities are the individual tasks that get executed within a pipeline.

The screenshot displays the Azure Data Factory (ADF) interface for a pipeline named 'pipeline1'. The left sidebar shows the 'Activities' section with a search bar and a list of activity types: Databricks, Data Lake Analytics, and General. The 'General' section is expanded, showing activities like 'Append variable', 'Delete', 'Execute Pipeline', 'Execute SSIS package', 'Fail', 'Get Metadata', 'Lookup', 'Stored procedure', 'Script', 'Set variable', 'Validation', 'Web', 'WebHook', and 'Wait'. The main canvas shows a pipeline diagram with two activities: 'Copy data' (labeled 'Copy data1') and 'Data flow' (labeled 'Data flow1'). A green arrow indicates the flow from 'Copy data' to 'Data flow'. The 'Copy data' activity is highlighted with a blue border. The bottom panel shows the 'General' tab for the selected 'Copy data' activity, with fields for Name, Description, Activity state (Activated/Deactivated), Timeout, Retry, Retry interval, and Secure output.

**Activities**

- > Databricks
- > Data Lake Analytics
- ▼ General
  - Append variable
  - Delete
  - Execute Pipeline
  - Execute SSIS package
  - Fail
  - Get Metadata
  - Lookup
  - Stored procedure
  - Script
  - Set variable
  - Validation
  - Web
  - WebHook
  - Wait

**pipeline1**

Save Save as template Validate Validate copy runtime Debug Add trigger Data flow debug

**Copy data**

Copy data1

**Data flow**

Data flow1

**General** Source Sink Mapping Settings User properties

Name \* Copy data1 [Learn more](#)

Description

Activity state ☒ Activated ☐ Deactivated

Timeout 0.12:00:00

Retry 0

Retry interval (sec) 30

Secure output

## PIPELINE

A pipeline is a logical grouping of activities that together perform a task. A pipeline allows you to manage and schedule workflows of data movement and transformation

## TRIGGER

Triggers allow you to automatically initiate the execution of a pipeline.

## INTEGRATION RUNTIME

The Integration Runtime is the compute infrastructure used by ADF to perform data movement, data transformation, and other activities.

# MAPPING DATA FLOWS IN AZURE DATA FACTORY

Mapping Data Flows in Azure Data Factory (ADF) allows you to transform and move data in a no-code environment. Data flows support a variety of transformation activities to achieve tasks like aggregations, filtering, joins, and more. It can support - Data cleaning, aggregation, transformation and enrichment.

## WHY USE MAPPING DATA FLOWS IN AZURE DATA FACTORY

**Drag and Drop Transformations:** Data Flows allow for transformations like joins, filters, aggregations, and more, making it easy to prepare data exactly as needed.

**Scalability:** With ADF's Spark-based engine, Data Flows can process large volumes of data efficiently.

**Reusability:** Create reusable transformation logic for various datasets, reducing redundancy.

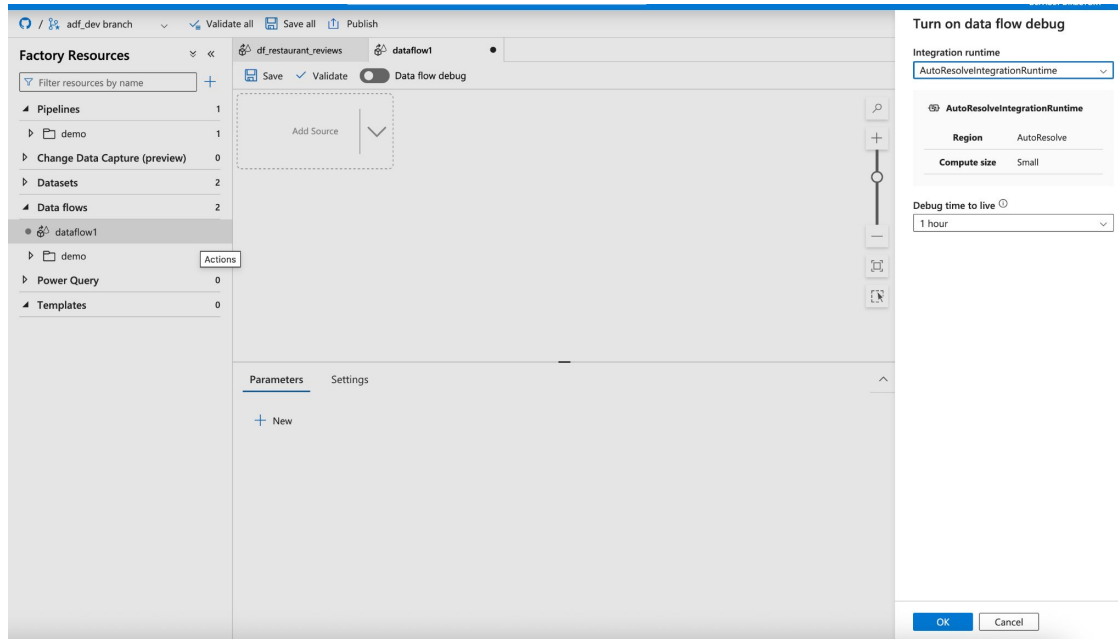
**Ease of Use:** The intuitive visual interface simplifies complex transformations, making them accessible even to non-developers.

**Integration:** Seamlessly integrates with other ADF activities, allowing for end-to-end automation within data pipelines.

# 1. PREPARING THE ENVIRONMENT

Turn the Data Flow Debug slider located at the top of the authoring module on  
Add a Data Flow activity.

Create the Required Linked Services and Datasets for source and destination sink

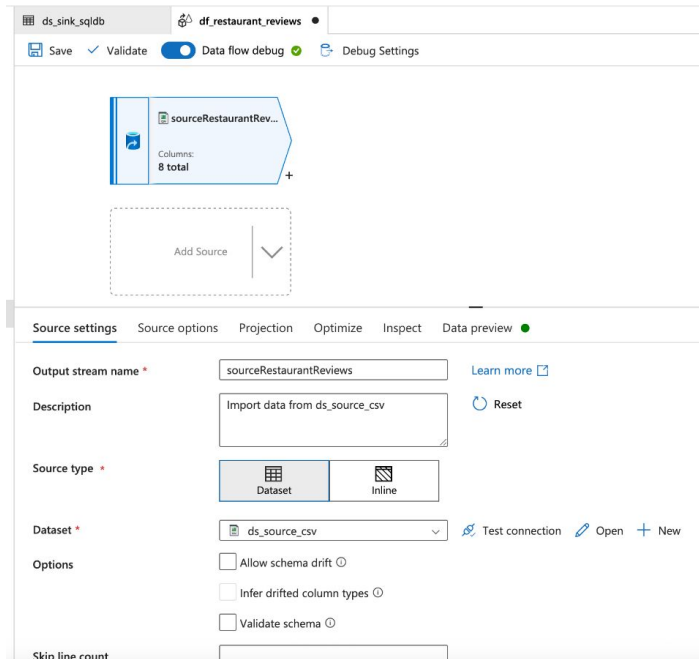


## 2. ADD SOURCE DATA TO THE MAPPING DATA FLOW

When you design data flows, your first step is always configuring a source.

Every data flow requires at least one source transformation, but you can add as many sources as necessary to complete your data transformations.

The sources can be combined using a join, lookup, or a union transformation.



### 3. TRANSFORMATIONS IN THE MAPPING DATA FLOW

Mapping Data Flows provides a number of different transformations types that enable you to modify data.

- Schema modifier transformations

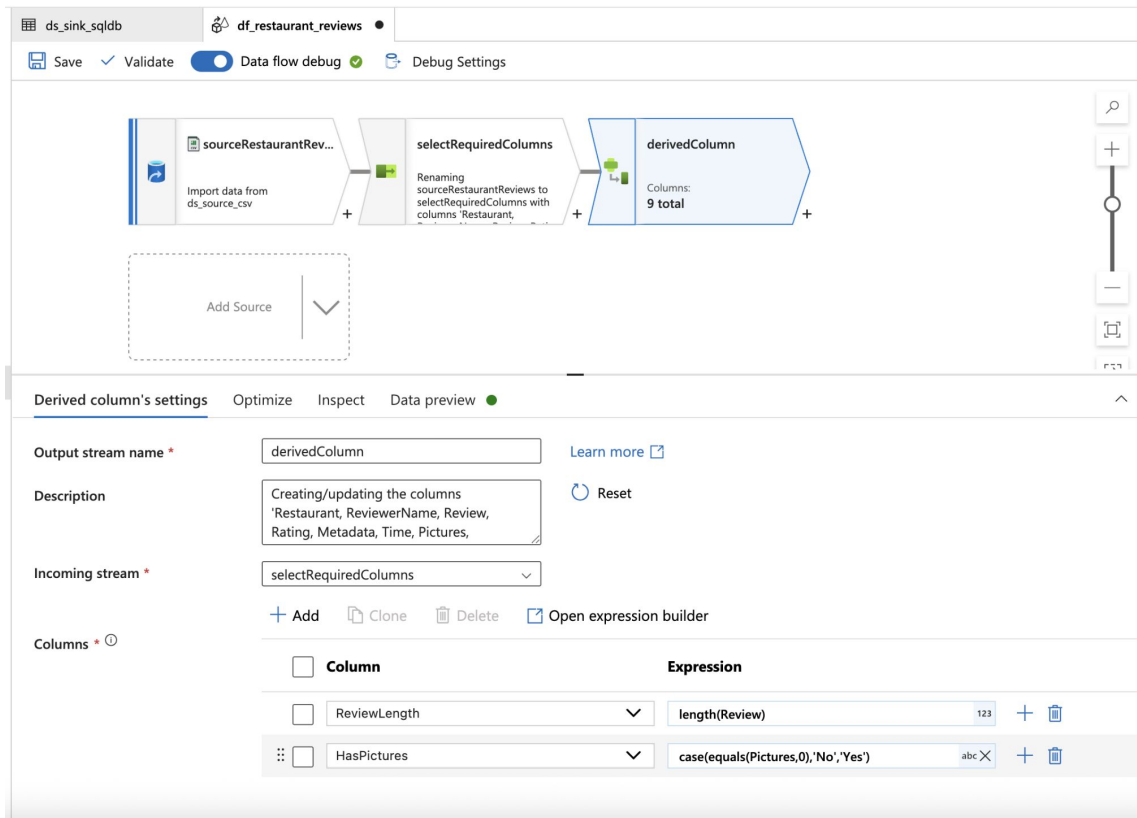
Derived Column, Select, Aggregate

- Row modifier transformations

Sort, Filter, Alter Row

- Multiple inputs/outputs transformations

Join, Conditional Split, Union, Lookup



The screenshot displays the Microsoft Data Flow Designer interface. At the top, the workspace shows a data flow starting with a source named 'sourceRestaurantReviews' (Import data from ds\_source\_csv), followed by a transformation 'selectRequiredColumns' (Renaming sourceRestaurantReviews to selectRequiredColumns with columns 'Restaurant', 'ReviewerName', 'Review', 'Rating', 'Metadata', 'Time', 'Pictures'), and finally a derived column transformation 'derivedColumn' (Columns: 9 total). Below the workspace, the 'Derived column's settings' pane is open, showing the 'Columns' section with two columns: 'ReviewLength' and 'HasPictures'. The 'ReviewLength' column has the expression 'length(Review)' and the 'HasPictures' column has the expression 'case(equals(Pictures,0),'No','Yes')'. The 'Incoming stream' is set to 'selectRequiredColumns'.

Column	Expression
ReviewLength	length(Review)
HasPictures	case(equals(Pictures,0),'No','Yes')

<https://learn.microsoft.com/en-us/training/modules/code-free-transformation-scale/3-describe-transformation-types>



## 4. WRITING TO DATA SINK

The Sink transformation is the endpoint in a data flow where transformed data is written to a target destination.

It supports various sinks like Azure Blob Storage, Azure SQL Database, Data Lake, Cosmos DB, etc.

This is where data is stored after all processing is completed.

The screenshot displays the Databricks Data Flow interface. At the top, a tab labeled 'df\_restaurant\_reviews' is active. Below the tab, there are buttons for 'Saved', 'Validate', and 'Data flow debug'. The main area shows a data flow pipeline with five transformations: 'selectRequiredColumns' (Renaming sourceRestaurantReviews to selectRequiredColumns with columns 'Restaurant', 'Review', 'Rating', 'Metadata', 'Time', 'Pictures', 'ReviewLength'), 'derivedColumn' (Creating/updating the columns 'Restaurant', 'ReviewName', 'Review', 'Rating', 'Metadata', 'Time', 'Pictures', 'ReviewLength'), 'filterRatingsWithThree' (Filtering rows using expressions on columns 'Rating'), 'sortByRestaurantName' (Sorting rows on columns 'Restaurant'), and 'sinkProcessedRestau...' (Write order: 1, Columns: 9 total). Below the pipeline, there is a tabbed interface with 'Sink' selected. The 'Sink' tab shows the following configuration:

- Output stream name: sinkProcessedRestaurantReviews
- Description: Export data to ds\_sink\_sqldb
- Incoming stream: sortByRestaurantName
- Sink type: Dataset (selected), Inline, Cache
- Dataset: ds\_sink\_sqldb
- Options: ☒ Allow schema drift, ☐ Validate schema

Buttons for 'Test connection', 'Open', and 'New' are also visible.

## 5. ADD DATAFLOW ACTIVITY TO PIPELINE AND TRIGGER IT

Create a new pipeline.

Drag and drop the Data Flow activity onto the pipeline canvas.

Configure the Data Flow Activity with required parameters.

Trigger the Pipeline and monitor the data flow run to see if the data is written to sink after performing the expected transformations.

The screenshot displays the Azure Data Studio interface. On the left, the 'Factory Resources' pane shows a tree view with 'Pipelines' (1), 'demo' (1), 'pl\_process\_restaurant\_reviews' (selected), 'Change Data Capture (preview)' (0), 'Datasets' (2), 'Data flows' (1), 'demo' (1), 'df\_restaurant\_reviews', 'Power Query' (0), and 'Templates' (0). The main canvas shows a pipeline named 'pl\_process\_restaurant\_reviews' with a 'Data flow' activity named 'ac\_df\_restaurant\_reviews' added. The 'Data flow' activity is highlighted with a blue box. Below the canvas, the 'Settings' tab is active, showing the configuration for the 'Data flow' activity. The 'Data flow' dropdown is set to 'df\_restaurant\_reviews'. The 'sourceRestaurantReviews parameters' section includes 'container\_name' (data), 'folder\_name' (raw\_files), and 'file\_name' (restaurant\_reviews.csv). The 'sinkProcessedRestaurantReviews parameters' section includes 'schema\_name' (dbo) and 'table\_name' (RestaurantReviewsProcessed). The 'Run on (Azure IR)' dropdown is set to 'AutoResolveIntegrationRuntime'. The 'Compute size' dropdown is set to 'Small'. The 'Logging level' is set to 'Verbose'.

**Factory Resources**

- Pipelines 1
  - demo 1
    - pl\_process\_restaurant\_reviews
- Change Data Capture (preview) 0
- Datasets 2
- Data flows 1
  - demo 1
    - df\_restaurant\_reviews
- Power Query 0
- Templates 0

**Data flow**

df\_restaurant\_reviews

sourceRestaurantReviews parameters

Name	Value
container_name	data
folder_name	raw_files
file_name	restaurant_reviews.csv

sinkProcessedRestaurantReviews parameters

Name	Value
schema_name	dbo
table_name	RestaurantReviewsProcessed

Run on (Azure IR) AutoResolveIntegrationRuntime

Compute size Small

Advanced

Logging level Verbose

# DEMO

*You are working for a company that runs a large online restaurant review platform*

You are to extract data from the source which has thousands of new reviews from different customers.

Before moving the data to the required destination, you are also expected to perform some transformations

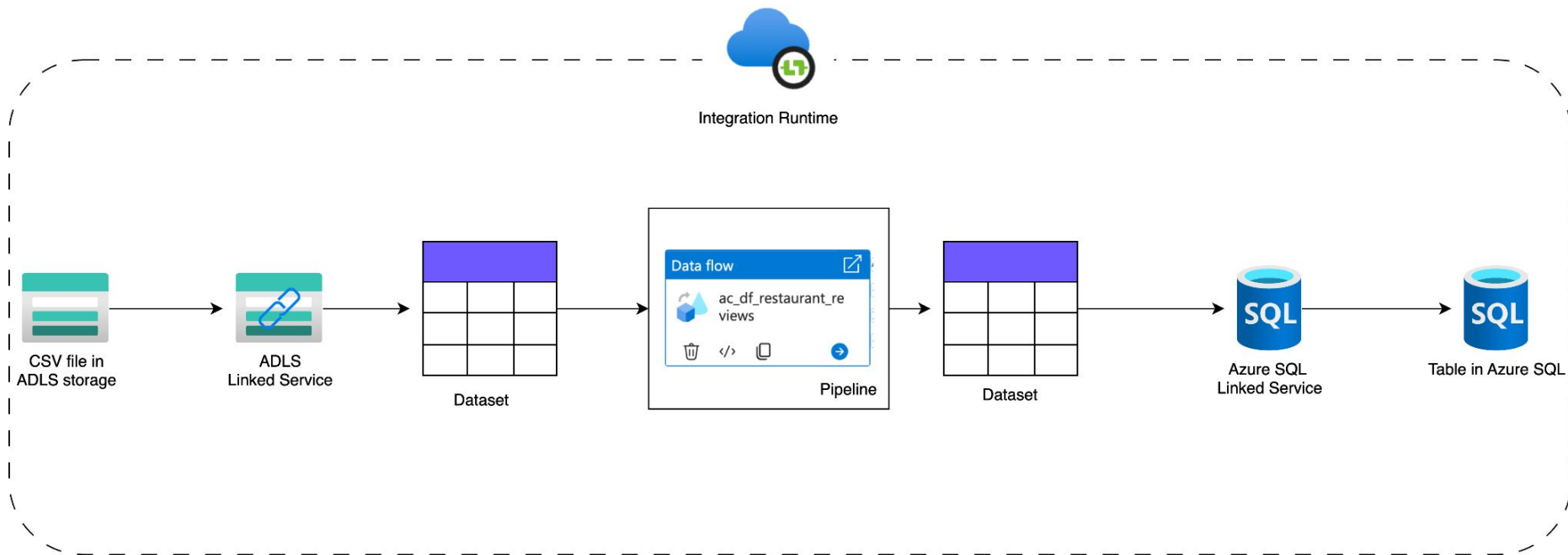
- Drop irrelevant columns
- Create new features like Review Length and HasPictures.
- Select rows with rating not equal to 3

Create a Mapping Data Flow to clean the data, add meaningful columns, summarize key insights, and prepare it for analytics."

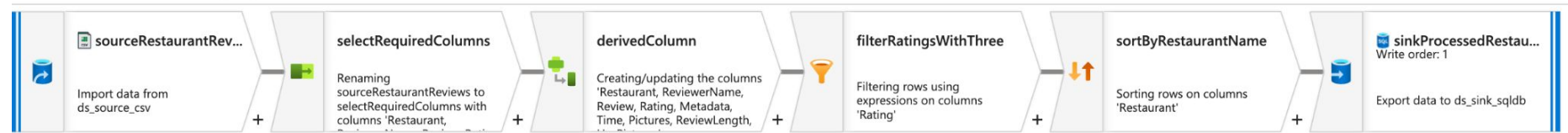
# DATASET

Restaurant	Reviewer	Review	Rating	Metadata	Time	Pictures	DummyColumn
Beyond Flavours	A	The ambience was good, food was quite good . had Saturday lunch , which was cost effective . Good place for a sate brunch. One can also chill with friends and or parents. Waiter Soumen Das was really courteous and helpful.	5	1 Review , 2 Followers	5/25/2019 15:54	0	2447
Beyond Flavours	B	Ambience is too good for a pleasant evening. Service is very prompt. Food is good. Over all a good e	5	3 Reviews , 2 Followers	5/25/2019 14:20	0	
Beyond Flavours	C	A must try.. great food great ambience. Thnx for the service by Pradeep and Subroto. My personal re	5	2 Reviews , 3 Followers	5/24/2019 22:54	0	

<https://www.kaggle.com/datasets/joebeachcapital/restaurant-reviews/data>

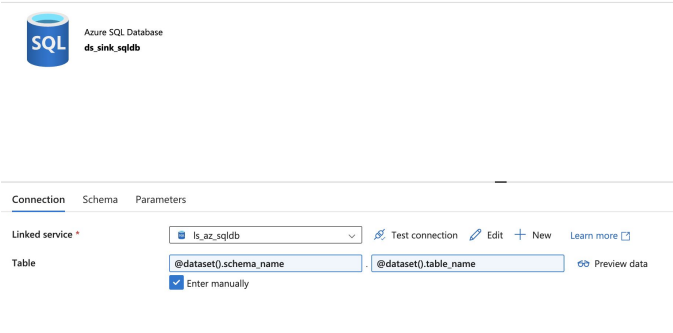


# Data Flow



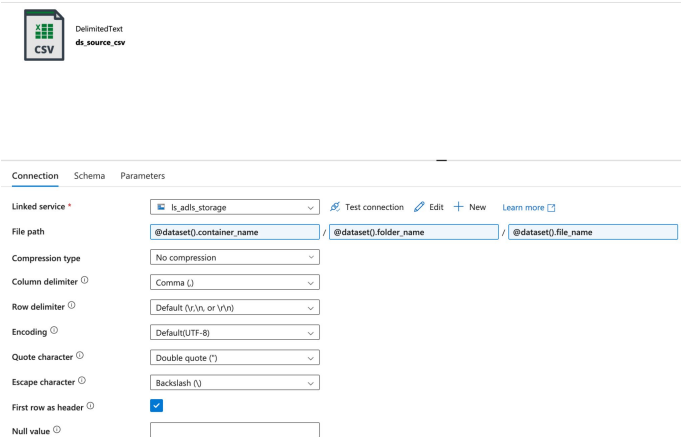
## ADF Structure

└ Pipelines	1
└ demo	1
pl_process_restaurant_reviews	
└ Change Data Capture (preview)	0
└ Datasets	2
demo	2
ds_sink_sqldb	
ds_source_csv	
└ Data flows	1
demo	1
df_restaurant_reviews	








Azure SQL table dataset

## Azure Storage dataset



# Linked Services

 <b>ls_adls_storage</b>	  <b>Azure Data Lake Storage Gen2</b>	<b>1</b>
 <b>ls_az_sqldb</b>	<b>Azure SQL Database</b>	<b>1</b>

 Azure Data Lake Storage Gen2 [Learn more](#)

**Name \***  
ls\_adls\_storage

**Description**

**Connect via integration runtime \*** ⓘ  

✔ AutoResolveIntegrationRuntime

**Authentication type**  

Account key

**Account selection method** ⓘ  

☐ From Azure subscription ☒ Enter manually

**URL \***  

https://dlsgeeknight.dfs.core.windows.net/

Storage account key Azure Key Vault

**Storage account key \***  

\*\*\*\*\*

**Test connection** ⓘ  


☒ To linked service ☐ To file path

**Annotations**  

+ New

> Parameters

> Advanced ⓘ

 Azure SQL Database [Learn more](#)

ⓘ To avoid publishing immediately to Data Factory, please use Azure Key Vault to retrieve secrets securely. [Learn more](#) [here](#)

**Name \***  
ls\_az\_sqldb

**Description**

**Connect via integration runtime \*** ⓘ  

✔ AutoResolveIntegrationRuntime

**Version**  

☒ 2.0 ☐ 1.0

**Account selection method** ⓘ  

☐ From Azure subscription ☒ Enter manually

**Fully qualified domain name \***  

geeknight.database.windows.net

**Database name \***  

sqldb-geeknight

**Authentication type \***  

SQL authentication

**User name \***  

admin\_sa

Password Azure Key Vault

**Password \***

# DISADVANTAGES

**Not Ideal for Highly Custom Transformations** - using custom code in Databricks or other services might be necessary

**Limited Control Over Spark Environment** - Spark environment used in data flows is abstracted, limiting fine tuning and custom configurations.

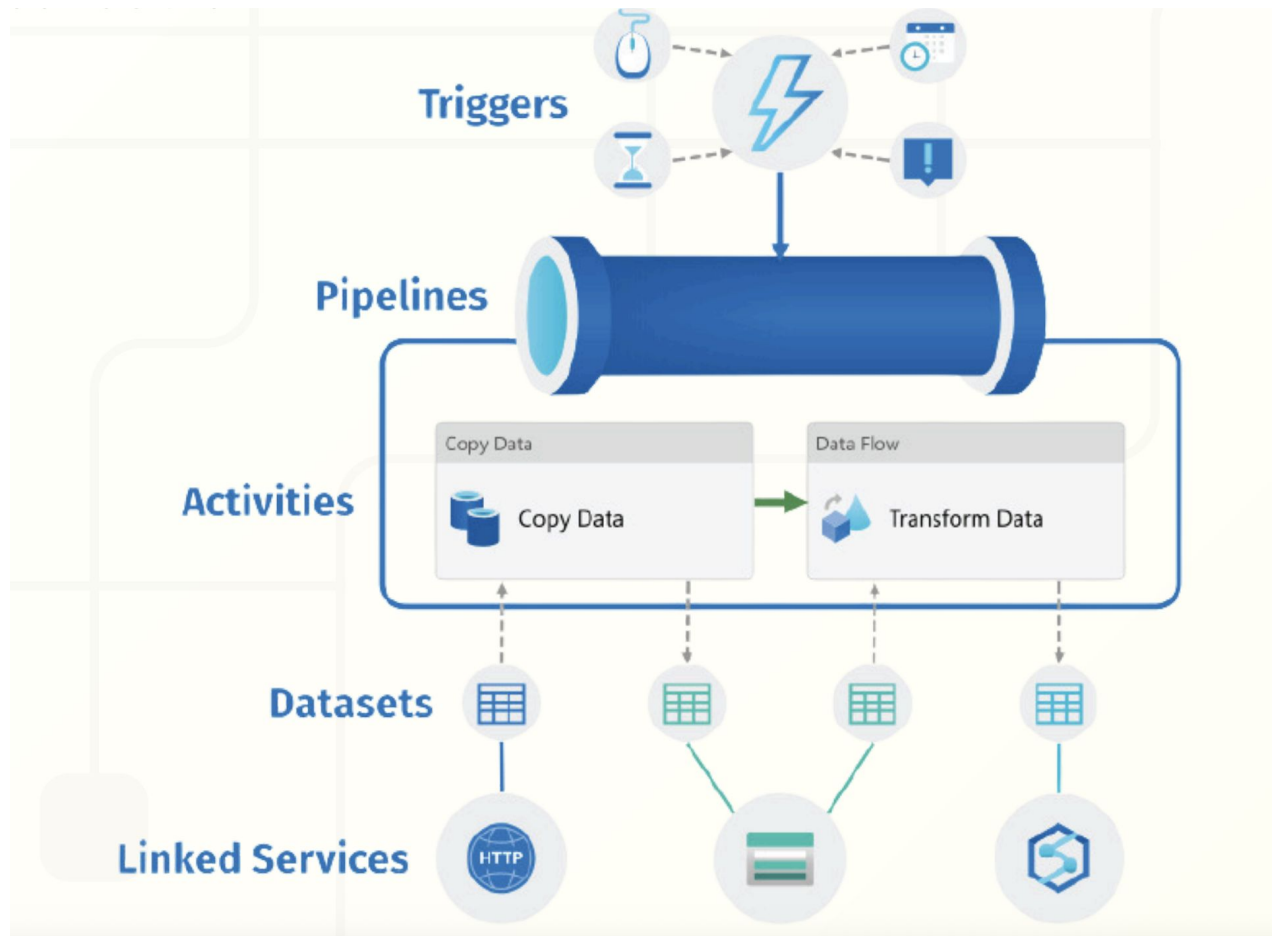
**Higher Costs for Large Data Volumes and Data Flow Debug** - Can become costly when processing large volumes of data frequently and also during debugging.

**Complex Debugging for Large Flows** - Debugging complex data flows with multiple transformations can become challenging

**Performance Overhead** - Data flows rely on Spark clusters, which introduce startup time and may increase latency.

**Limited Real-Time Processing** - Primarily designed for batch processing, not real-time or streaming data.





# Thank you

[https://github.com/antoprince001/geek\\_night\\_adf\\_data\\_flows](https://github.com/antoprince001/geek_night_adf_data_flows)