



SQL Saturday Toronto 2023

Metadata-driven pipelines in Azure Data Factory

Rayis Imayev

Lead Cloud Data Engineer, OMERS



Toronto Data Professionals Community (TDPC)



SQL Saturday (#1064)

Our Sponsors



Community Support

[Toronto Data Professionals Community \(TDPC\)](#), one of the largest data professional's community in Toronto, host monthly event which offers interactive learning built by community and guided by trusted data experts.

TDPC Event Partners

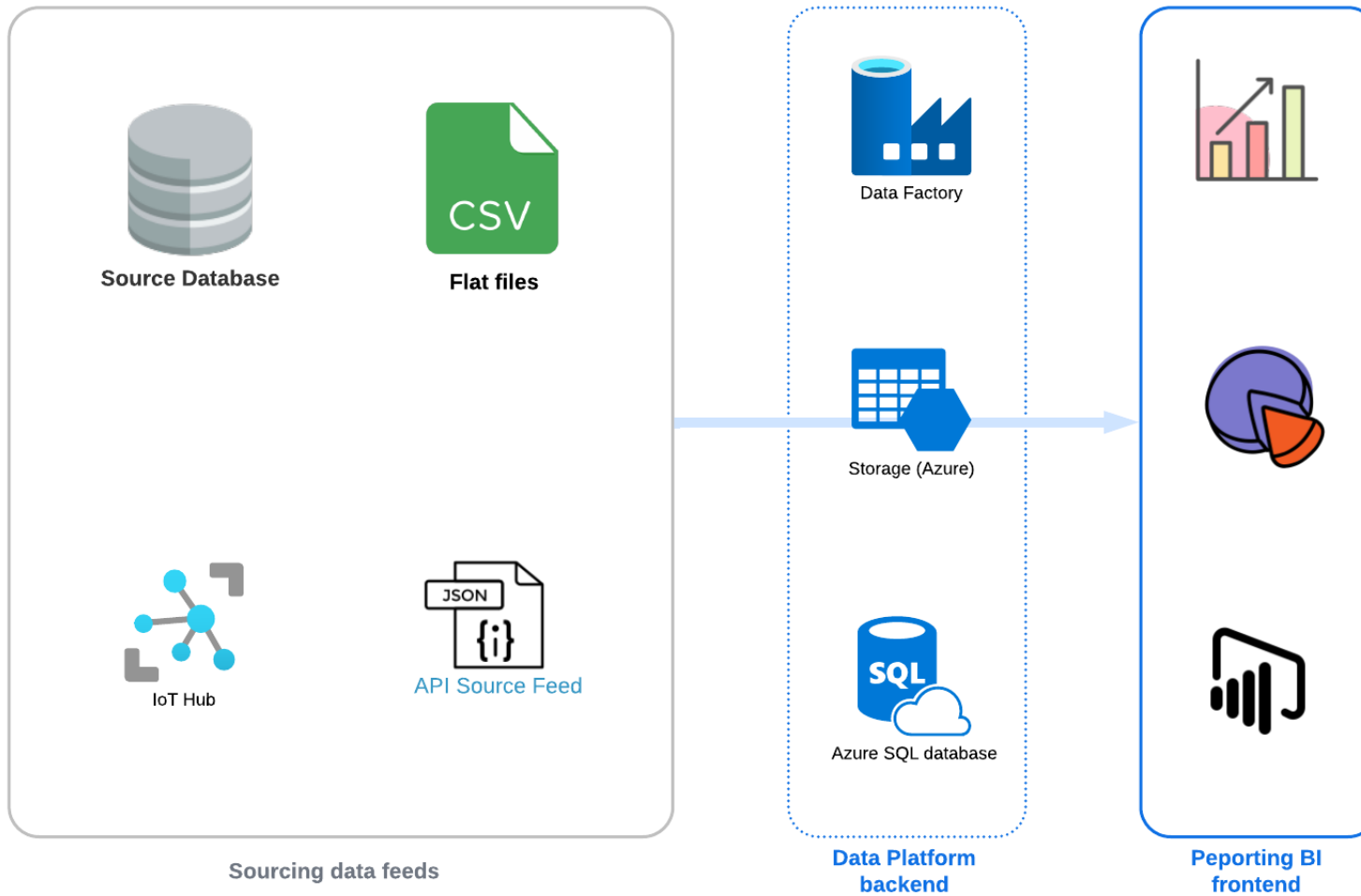


Toronto Data Professionals Community (TDPC) - [Metadata-driven pipelines in Azure Data Factory](#) = Rayis Imayev

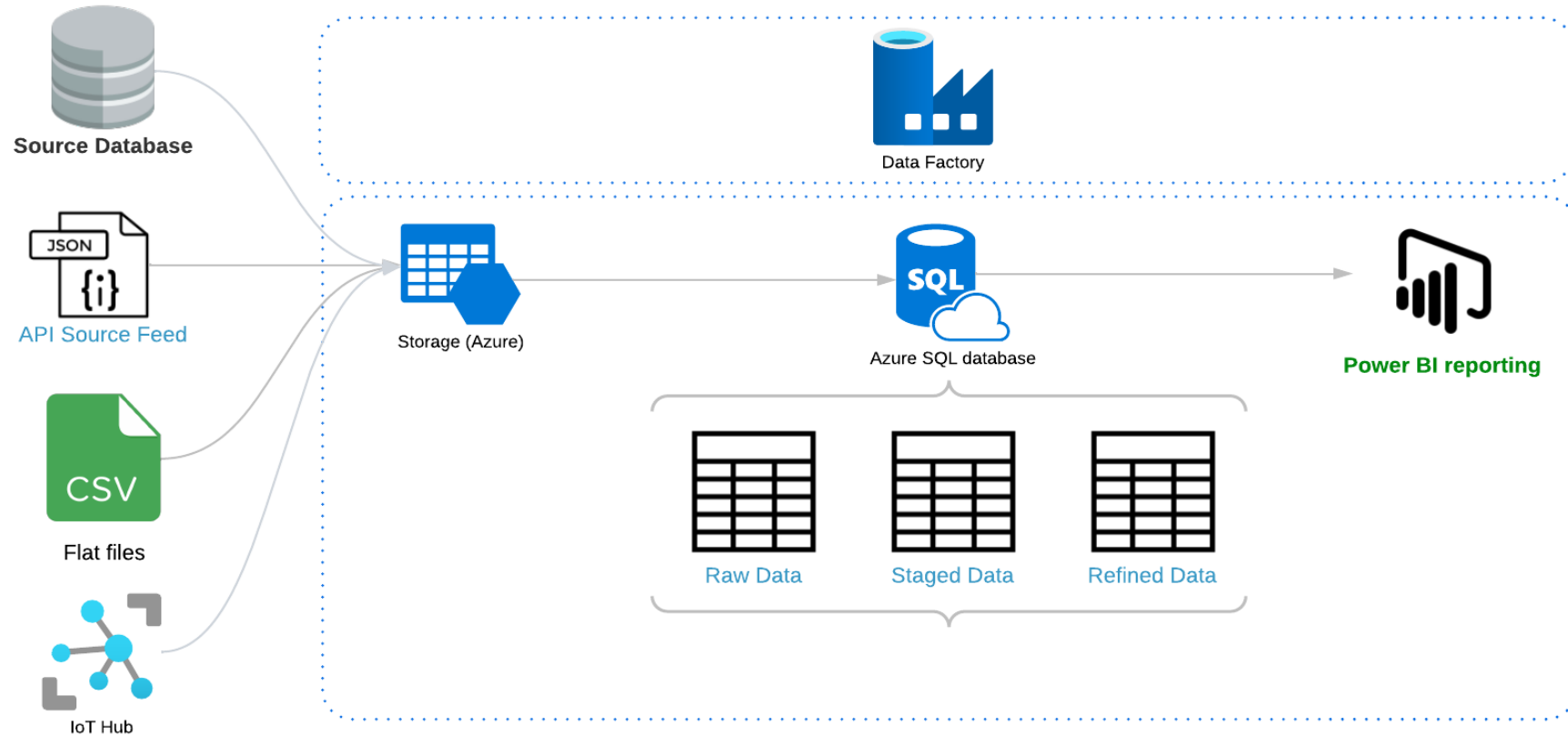


SQL Saturday (#1064)

Typical Data Integration Solution



Typical Data Integration Solution





Metadata can be described as "information that describes other information in order to help you understand or use it."

Oxford dictionary





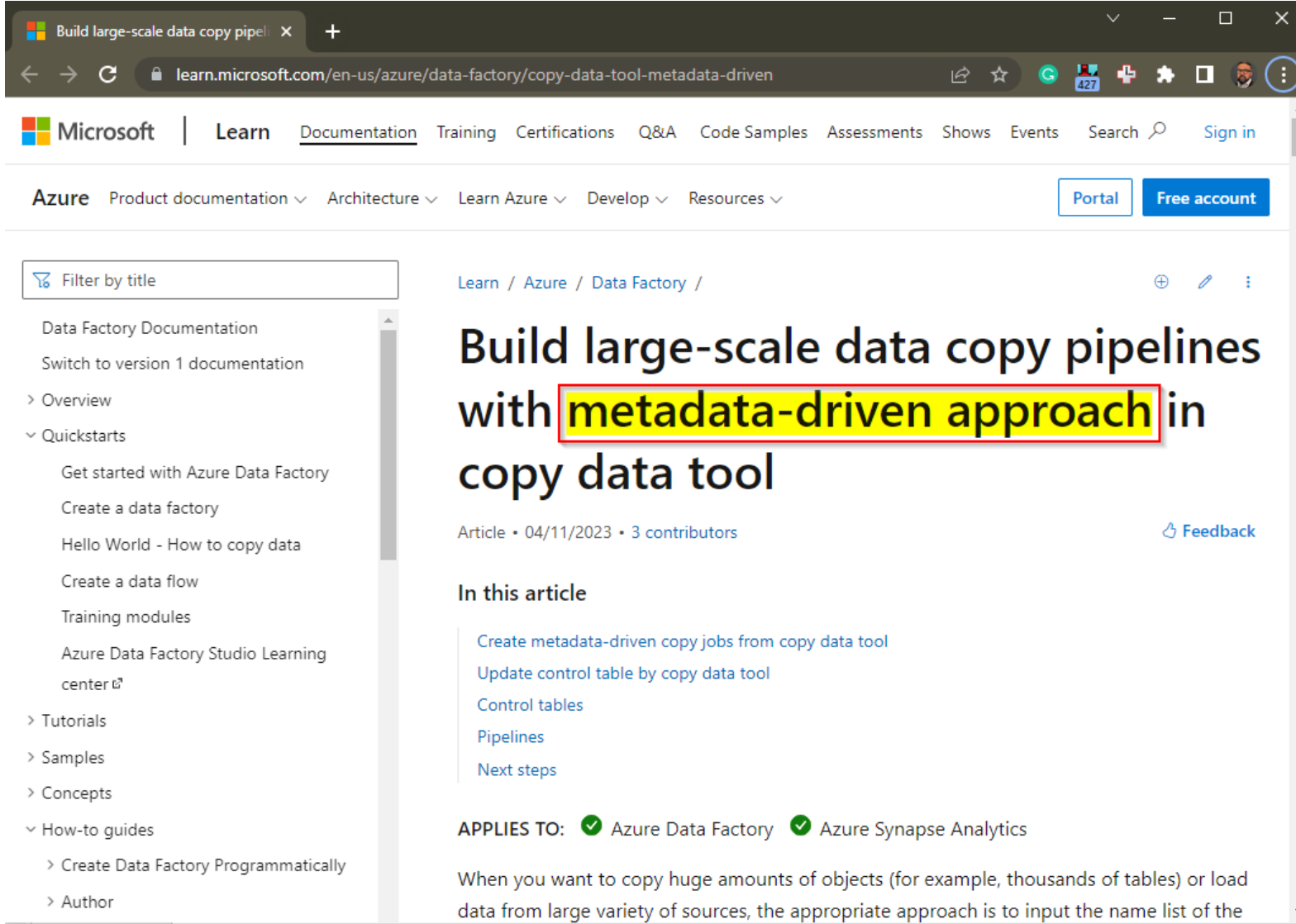
Pipeline is "a series of pipes that are usually underground and are used for carrying oil, gas, etc. over long distances "

Oxford dictionary



Existing solutions

<https://learn.microsoft.com/en-us/azure/data-factory/copy-data-tool-metadata-driven>

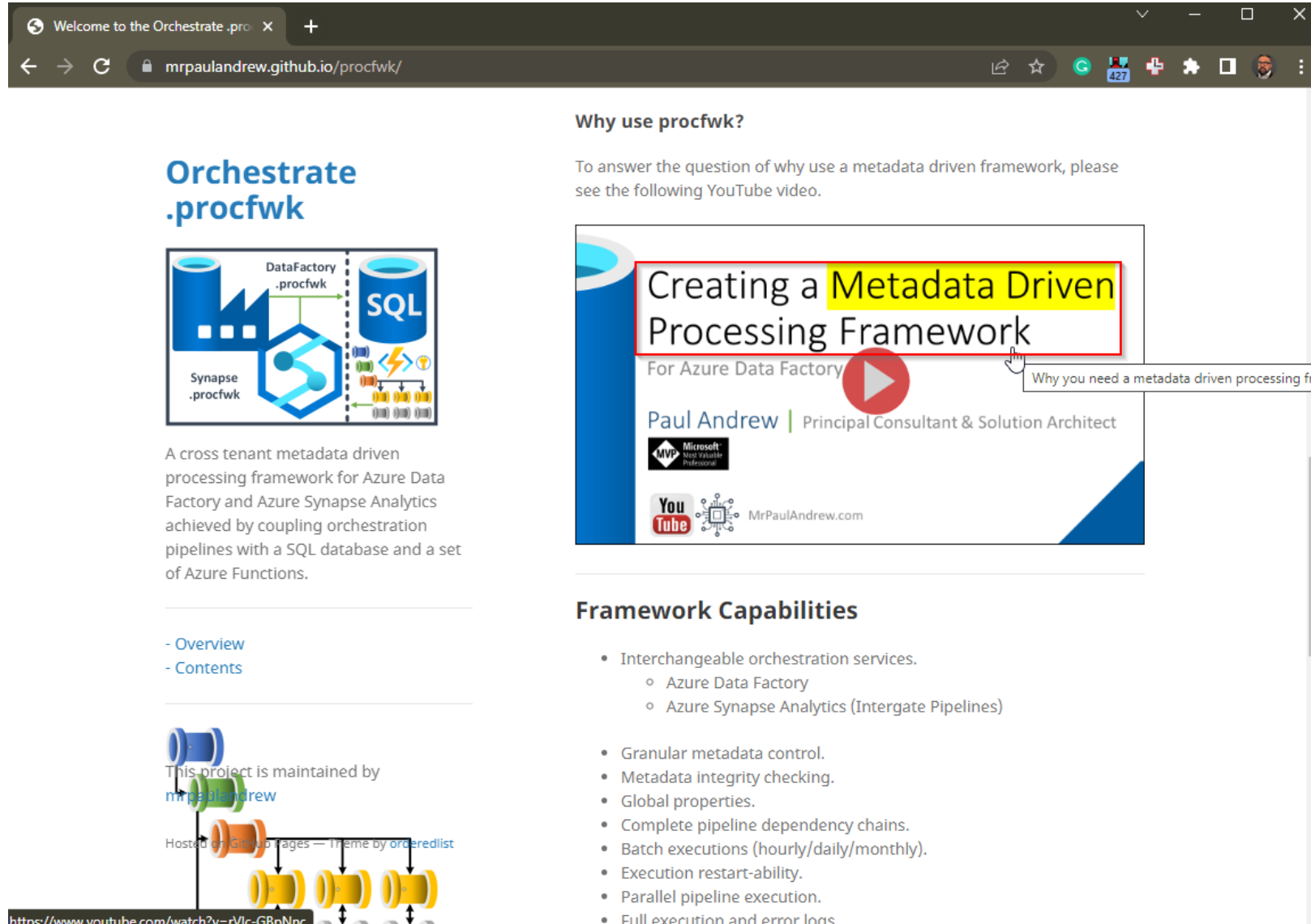


The screenshot shows a web browser window displaying a Microsoft Learn article. The browser's address bar shows the URL <https://learn.microsoft.com/en-us/azure/data-factory/copy-data-tool-metadata-driven>. The page header includes the Microsoft logo and navigation links for Learn, Documentation, Training, Certifications, Q&A, Code Samples, Assessments, Shows, Events, Search, and Sign in. Below the header, there are links for Azure, Product documentation, Architecture, Learn Azure, Develop, and Resources, along with buttons for Portal and Free account. The main content area features a sidebar on the left with a search bar and a list of navigation items: Data Factory Documentation, Switch to version 1 documentation, Overview, Quickstarts (Get started with Azure Data Factory, Create a data factory, Hello World - How to copy data, Create a data flow, Training modules, Azure Data Factory Studio Learning center), Tutorials, Samples, Concepts, and How-to guides (Create Data Factory Programmatically, Author). The main article title is "Build large-scale data copy pipelines with metadata-driven approach in copy data tool", with "metadata-driven approach" highlighted in a yellow box. Below the title, it says "Article • 04/11/2023 • 3 contributors" and a Feedback link. The "In this article" section lists: Create metadata-driven copy jobs from copy data tool, Update control table by copy data tool, Control tables, Pipelines, and Next steps. At the bottom, it states "APPLIES TO: Azure Data Factory, Azure Synapse Analytics" and provides a brief introduction: "When you want to copy huge amounts of objects (for example, thousands of tables) or load data from large variety of sources, the appropriate approach is to input the name list of the".



Existing solutions

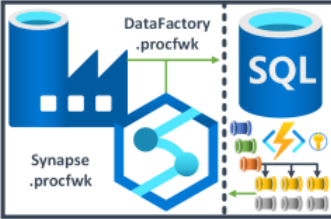
<https://mrpaulandrew.github.io/procfwk/>



Welcome to the Orchestrate .procfwk

mrpaulandrew.github.io/procfwk/

Orchestrate .procfwk



A cross tenant metadata driven processing framework for Azure Data Factory and Azure Synapse Analytics achieved by coupling orchestration pipelines with a SQL database and a set of Azure Functions.

- Overview
- Contents

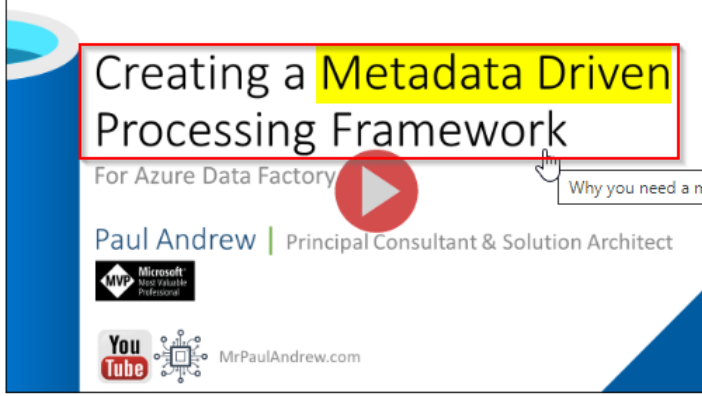
This project is maintained by [mrpaulandrew](#)

Hosted on GitHub Pages — Theme by [orderedlist](#)

<https://www.youtube.com/watch?v=rVlc-GBnNnc>

Why use procfwk?

To answer the question of why use a metadata driven framework, please see the following YouTube video.



Why you need a metadata driven processing framework

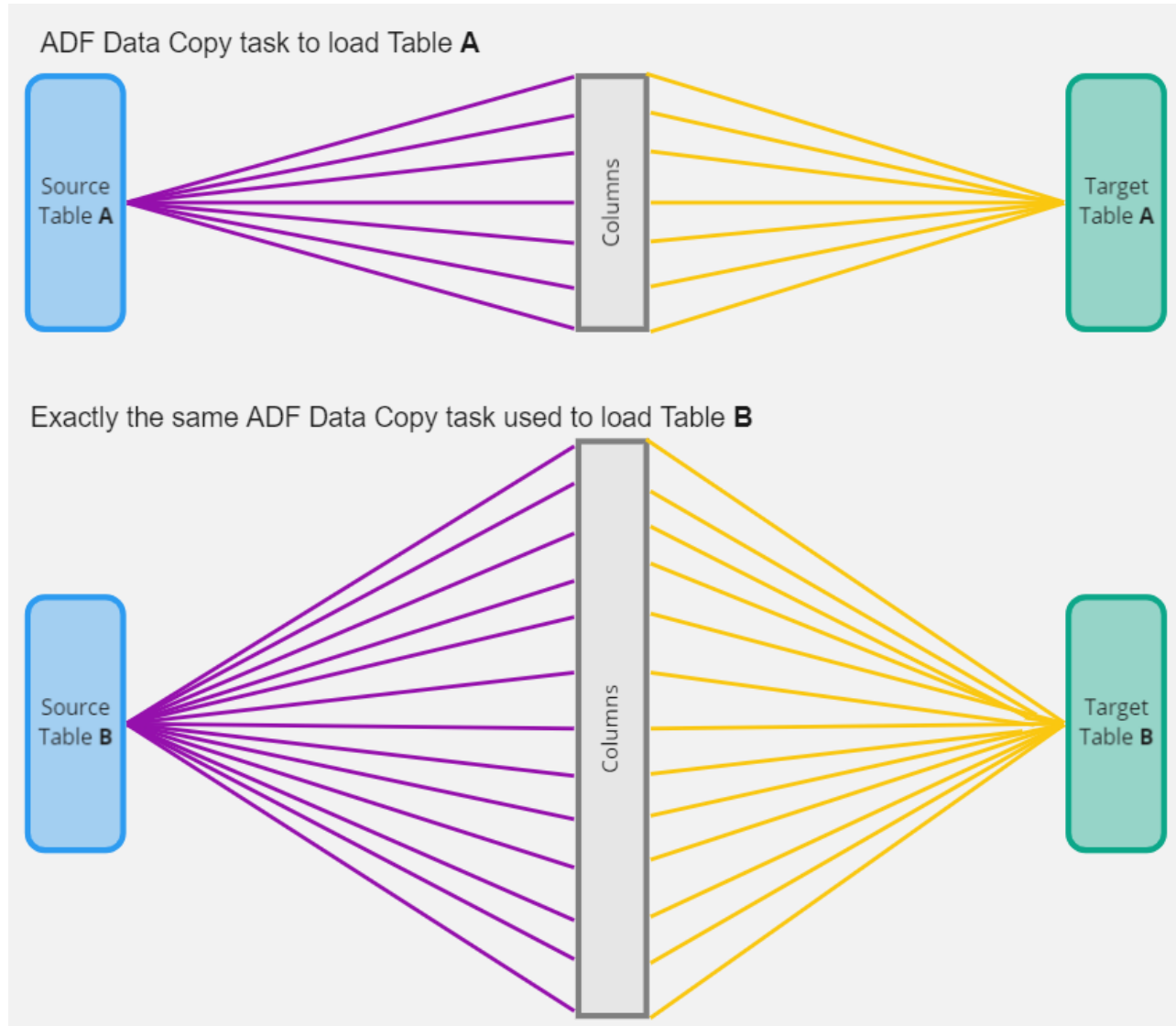
Framework Capabilities

- Interchangeable orchestration services.
 - Azure Data Factory
 - Azure Synapse Analytics (Intergate Pipelines)
- Granular metadata control.
- Metadata integrity checking.
- Global properties.
- Complete pipeline dependency chains.
- Batch executions (hourly/daily/monthly).
- Execution restart-ability.
- Parallel pipeline execution.
- Full execution and error logs.



Data Copy activity task

<https://learn.microsoft.com/en-us/azure/data-factory/copy-activity-overview>



Stages Zones Layers

(1)

Possible names: **Raw, Landing, Bronze.**

Purpose: Entry point for raw sourcing data; not suitable for end-user consumption, but might present some value for a data scientist team

(2)

Possible names: **Staging, Structured, Enriched, Silver**

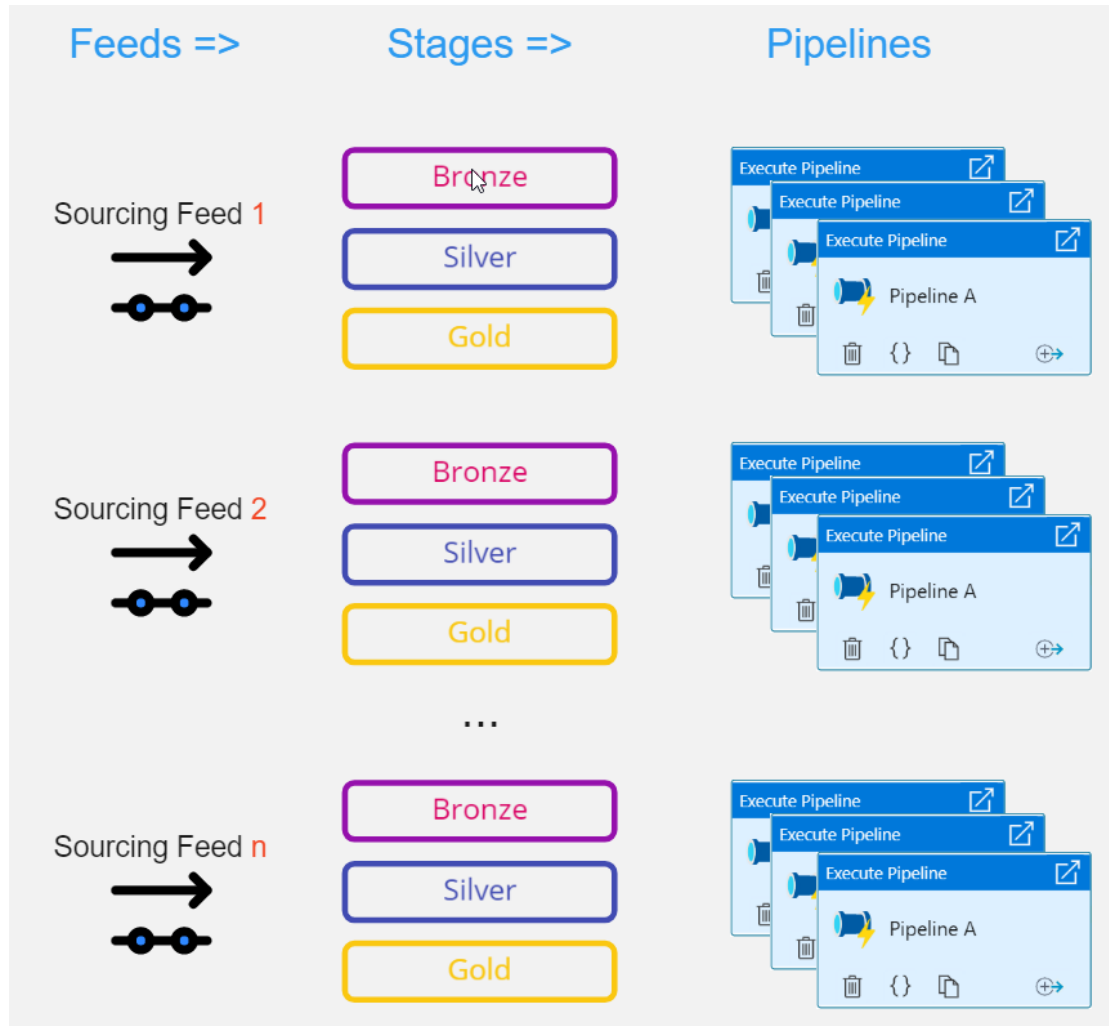
Purpose: Some data transformation is applied, conformed to a defined set of data types, list of attributes/columns can also be revised

(3)

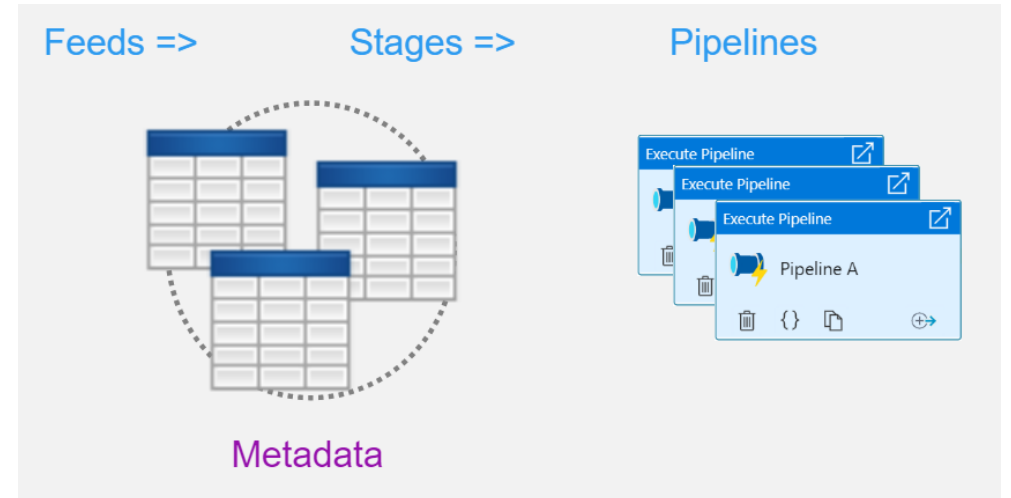
Possible names: **Refined, Curated, Product, Reporting, Gold**

Purpose: Data is ready-available for reporting and analytics; usually feeds an organization's data warehouse or serves as a data model for it; additional aggregations and calculations may be applied.





VS.



Configuration

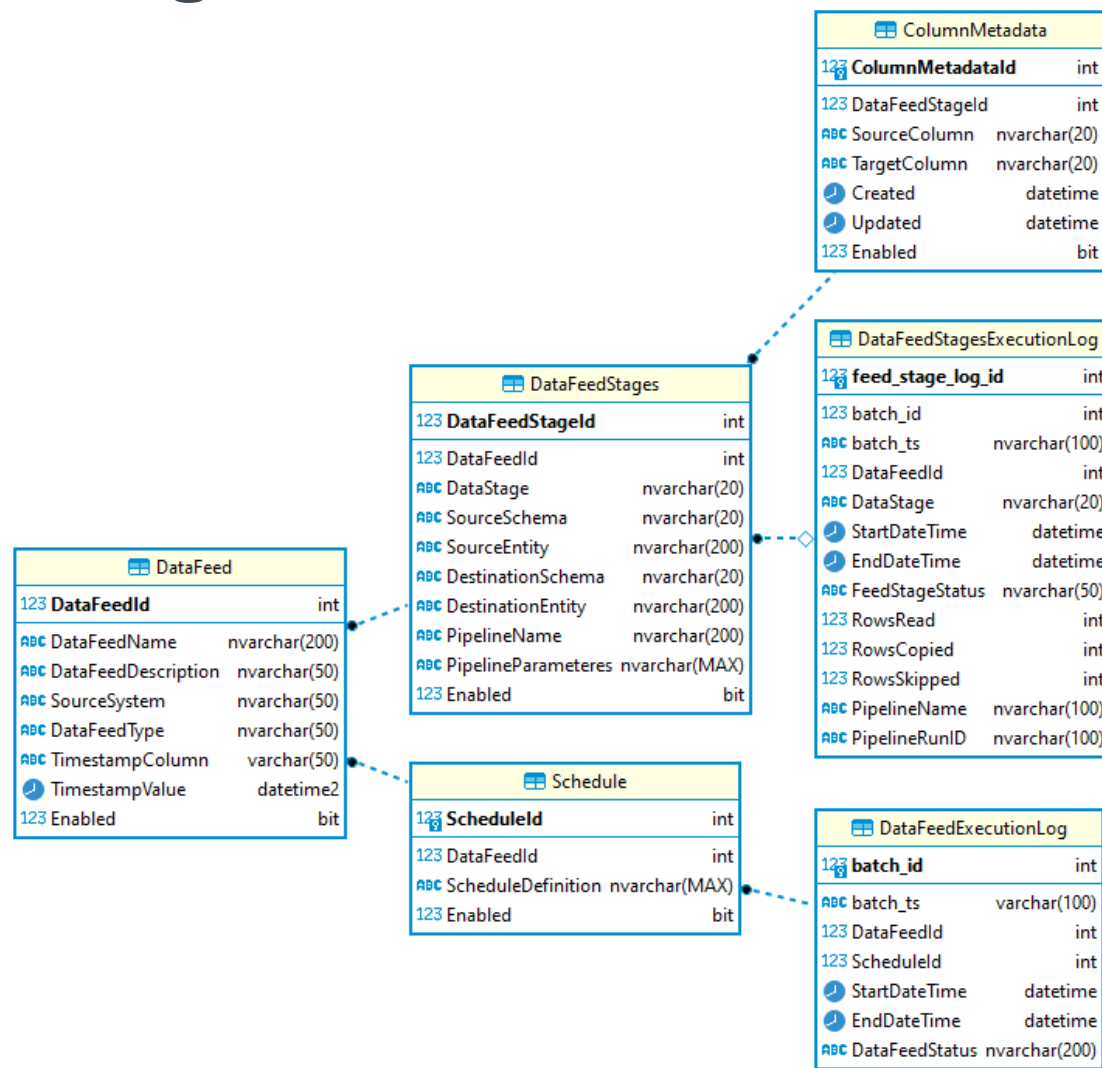


Table	Description
DataFeed	Data Feed configuration, e.g. sourcing table or file
DataFeedStages	Data Feed Stages, e.g. raw, staging, edw, etc.
Schedule	Schedule definition for a particular Data Feed, currently daily or hourly schedules are supported.
DataFeedExecutionLog	Execution Log for scheduled data feeds
DataFeedStagesExecutionLog	Execution Log for data feed stages of triggered data feeds
ColumnMetadata	Column metadata that maps columns between RAW & STG SQL schemas for a table feed

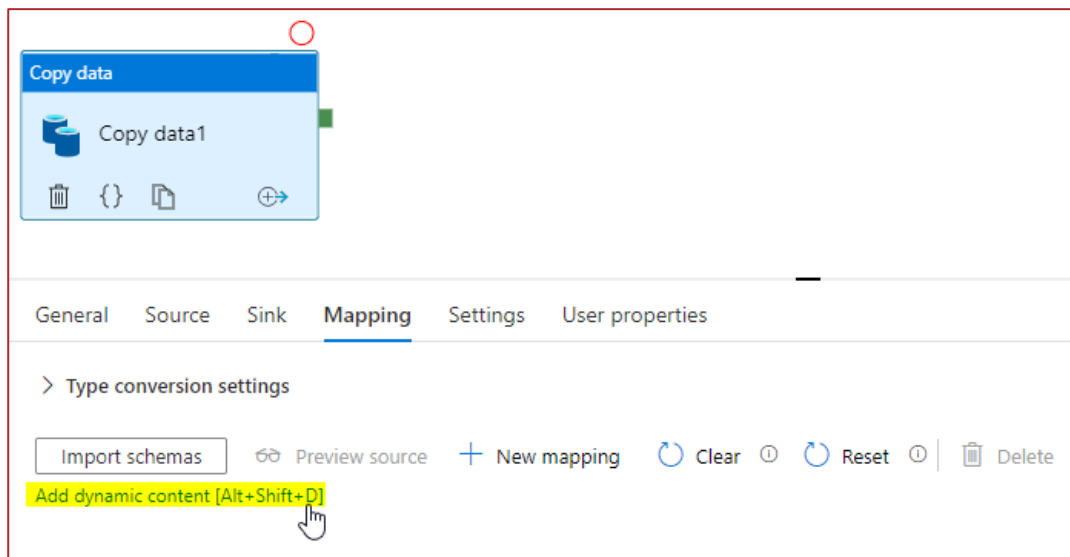
Demo:

Sourcing Feeds Data Load



Column Mapping

For example, mapping between 3 columns may look like this:



```
{
  "type": "TabularTranslator",
  "mappings": [
    {
      "source": {
        "name": "columnA"
      },
      "sink": {
        "name": "columnA"
      }
    },
    {
      "source": {
        "name": "columnB"
      },
      "sink": {
        "name": "columnB"
      }
    },
    {
      "source": {
        "name": "columnC"
      },
      "sink": {
        "name": "columnC"
      }
    }
  ]
}
```



Column Mapping

<https://sqlitybi.com/dynamically-set-copy-activity-mappings-in-azure-data-factory-v2/>

ColumnMetadata		
123	ColumnMetadataId	int
123	DataFeedStageId	int
ABC	SourceColumn	nvarchar(20)
ABC	TargetColumn	nvarchar(20)
	Created	datetime
	Updated	datetime
123	Enabled	bit

```
select * from [cfg].[ColumnMetadata]
```

Results							
	ColumnMetadataId	DataFeedStageId	SourceColumn	TargetColumn	Created	Updated	Enabled
1	1	1	columnA	columnA	2022-06-07 02:07:23.667	NULL	1
2	2	1	columnB	columnB	2022-06-07 02:07:46.047	NULL	1
3	3	1	columnC	columnC	2022-06-07 02:07:56.443	NULL	1

```
CREATE PROCEDURE [cfg].[GetColumnMapping]
    @DataFeedStageId int
AS
BEGIN
    DECLARE @json_construct varchar(MAX) = '{"type": "TabularTranslator", "mappings": {X}}'
    DECLARE @json VARCHAR(MAX);

    SET @json = (
        SELECT
            c.[SourceColumn] AS 'source.name',
            c.[TargetColumn] AS 'sink.name'
        FROM [cfg].[ColumnMetadata] as c
        WHERE c.DataFeedStageId = @DataFeedStageId
        FOR JSON PATH );

    SELECT REPLACE(@json_construct, '{X}', @json) AS json_output;
END
```

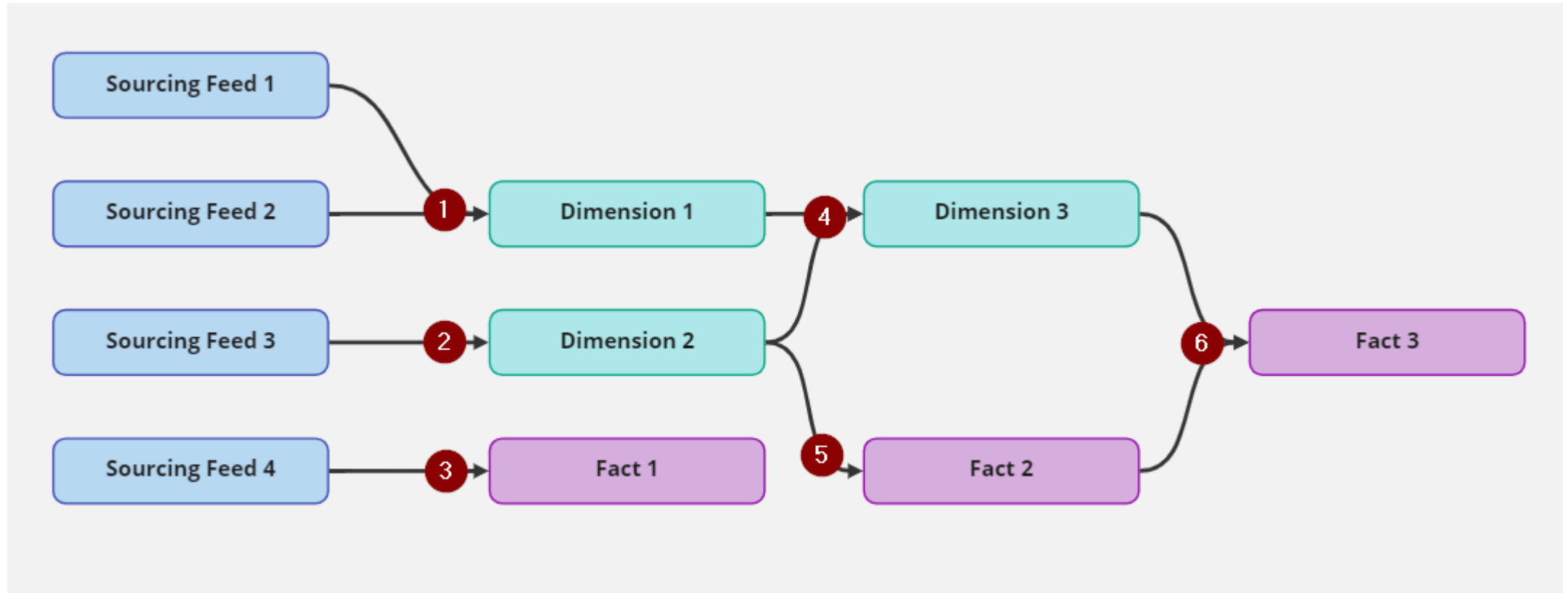
```
EXEC [cfg].[GetColumnMapping] @DataFeedStageId = 1
```

```
GO
```

Results	
	json_output
1	{ "type": "TabularTranslator", "mappings": [{ "source": { "name": "columnA" }, "sink": { "name": "columnA" } }, { "source": { "name": "columnB" }, "sink": { "name": "columnB" } }, { "source": { "name": "columnC" }, "sink": { "name": "columnC" } }] }



Dependency



Dependency

```
1 SELECT [DataFeedDependencyId]
2       , [DataObject]
3       , [DataObjectDependency]
4       , [ProcessingRoutine]
5       , [Enabled]
6       , [master_batch_type]
7 FROM [dbo].[DataFeedDependency]
```

	DataFeedDependencyId	DataObject	DataObjectDependency	ProcessingRoutine	Enabled	master_batch_type
1	1	dbo.Dimension_1	stg.Sourcing_Feed_1 ,stg.Sourcing_Feed_2	dbo.LoadDimension_1	1	Daily
2	2	dbo.Dimension_2	stg.Sourcing_Feed_3	dbo.LoadDimension_2	1	Daily
3	3	dbo.Dimension_3	dbo.Dimension_1 ,dbo.Dimension_2	dbo.LoadDimension_3	1	Daily
4	4	dbo.Fact_1	stg.Sourcing_Feed_4	dbo.LoadFact_1	1	Daily
5	5	dbo.Fact_2	dbo.Dimension_2	dbo.LoadFact_2	1	Daily
6	6	dbo.Fact_3	dbo.Dimension_3 ,dbo.Fact_2	dbo.LoadFact_3	1	Daily

```
2 SELECT *
3 FROM [dbo].[DataFeedDependencyExecutionLog]
```

	feed_dependency_log_id	master_batch_id	master_batch_ts	feed_batch_id	DataObjectParent	DataObjectChild	StartDateTime	EndDateTime	FeedDependencyStatus	ValidFrom
1	16	1	20230101000000	NULL	dbo.Dimension_1	dbo.Dimension_1	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
2	17	1	20230101000000	NULL	dbo.Dimension_1	stg.Sourcing_Feed_1	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
3	18	1	20230101000000	NULL	dbo.Dimension_1	stg.Sourcing_Feed_2	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
4	19	1	20230101000000	NULL	dbo.Dimension_2	dbo.Dimension_2	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
5	20	1	20230101000000	NULL	dbo.Dimension_2	stg.Sourcing_Feed_3	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
6	21	1	20230101000000	NULL	dbo.Dimension_3	dbo.Dimension_1	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
7	22	1	20230101000000	NULL	dbo.Dimension_3	dbo.Dimension_2	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
8	23	1	20230101000000	NULL	dbo.Dimension_3	dbo.Dimension_3	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
9	24	1	20230101000000	NULL	dbo.Fact_1	dbo.Fact_1	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
10	25	1	20230101000000	NULL	dbo.Fact_1	stg.Sourcing_Feed_4	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
11	26	1	20230101000000	NULL	dbo.Fact_2	dbo.Dimension_2	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
12	27	1	20230101000000	NULL	dbo.Fact_2	dbo.Fact_2	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
13	28	1	20230101000000	NULL	dbo.Fact_3	dbo.Dimension_3	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
14	29	1	20230101000000	NULL	dbo.Fact_3	dbo.Fact_2	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283
15	30	1	20230101000000	NULL	dbo.Fact_3	dbo.Fact_3	NULL	NULL	Start Pending	2023-02-20 18:26:59.0283



Demo:

Dependency Framework



Lesson Learned

1. Configuration is a key
 1. Deployment concept
 2. Changes from many team members
2. Monitor your workload:
 1. Number of parallel pipelines running
 2. Use case or Azure or Self-hosted Integration Runtime
3. Change is painful



Links and contact information

Session information:

Part 1 - <https://datanrg.blogspot.com/2022/03/metadata-driven-pipelines-in-azure-data.html>

Part 2 - <https://datanrg.blogspot.com/2022/05/metadata-driven-pipelines-in-azure-data.html>

Part 3 - <https://datanrg.blogspot.com/2022/06/metadata-driven-pipelines-in-azure-data.html>

Part 4 - <https://datanrg.blogspot.com/2023/02/metadata-driven-pipelines-in-azure-data.html>

Blog: <https://datanrg.blogspot.com/>

Twitter: <https://twitter.com/RayisImayev>

LinkedIn: <https://www.linkedin.com/in/rimayev/>

