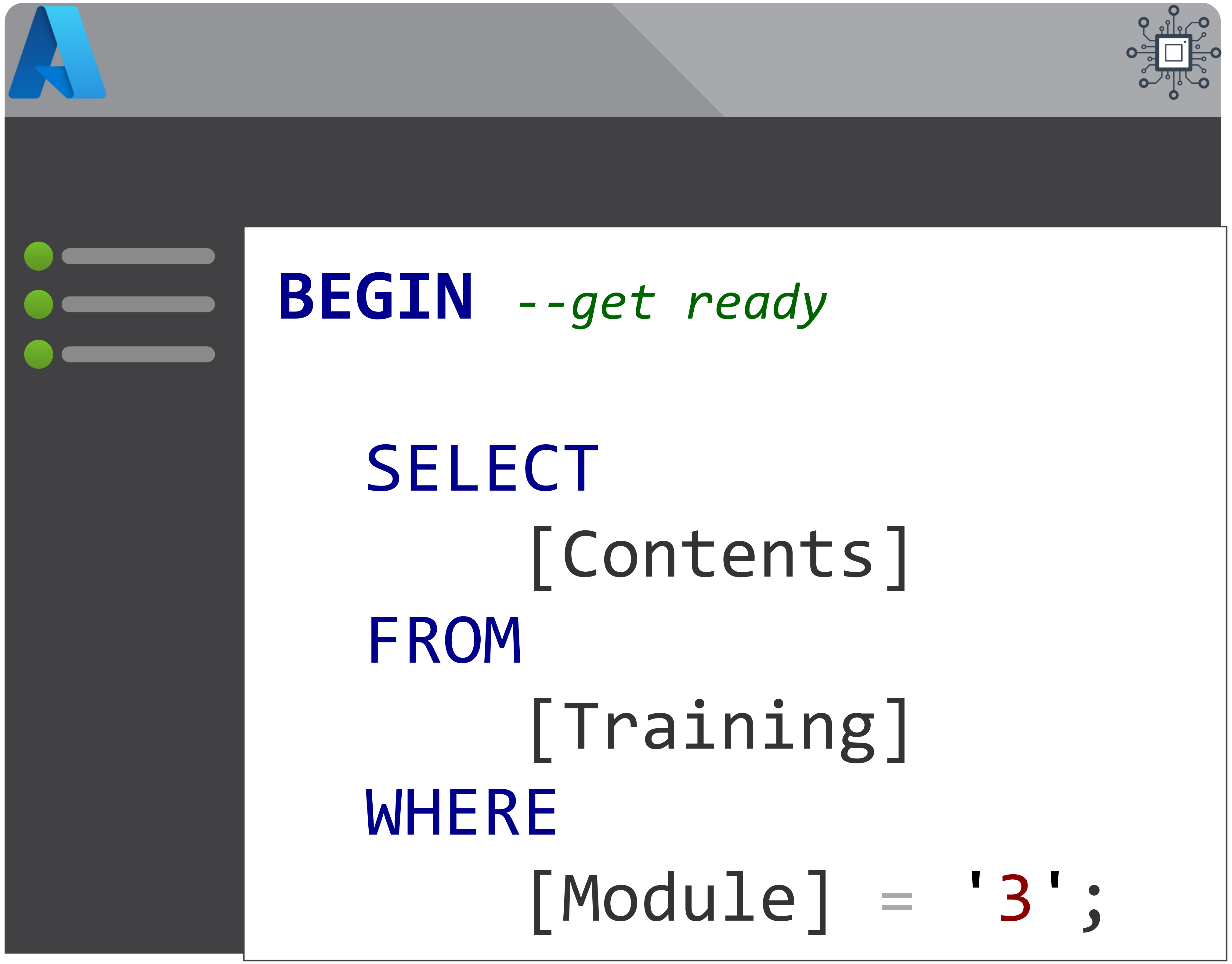
Module 3

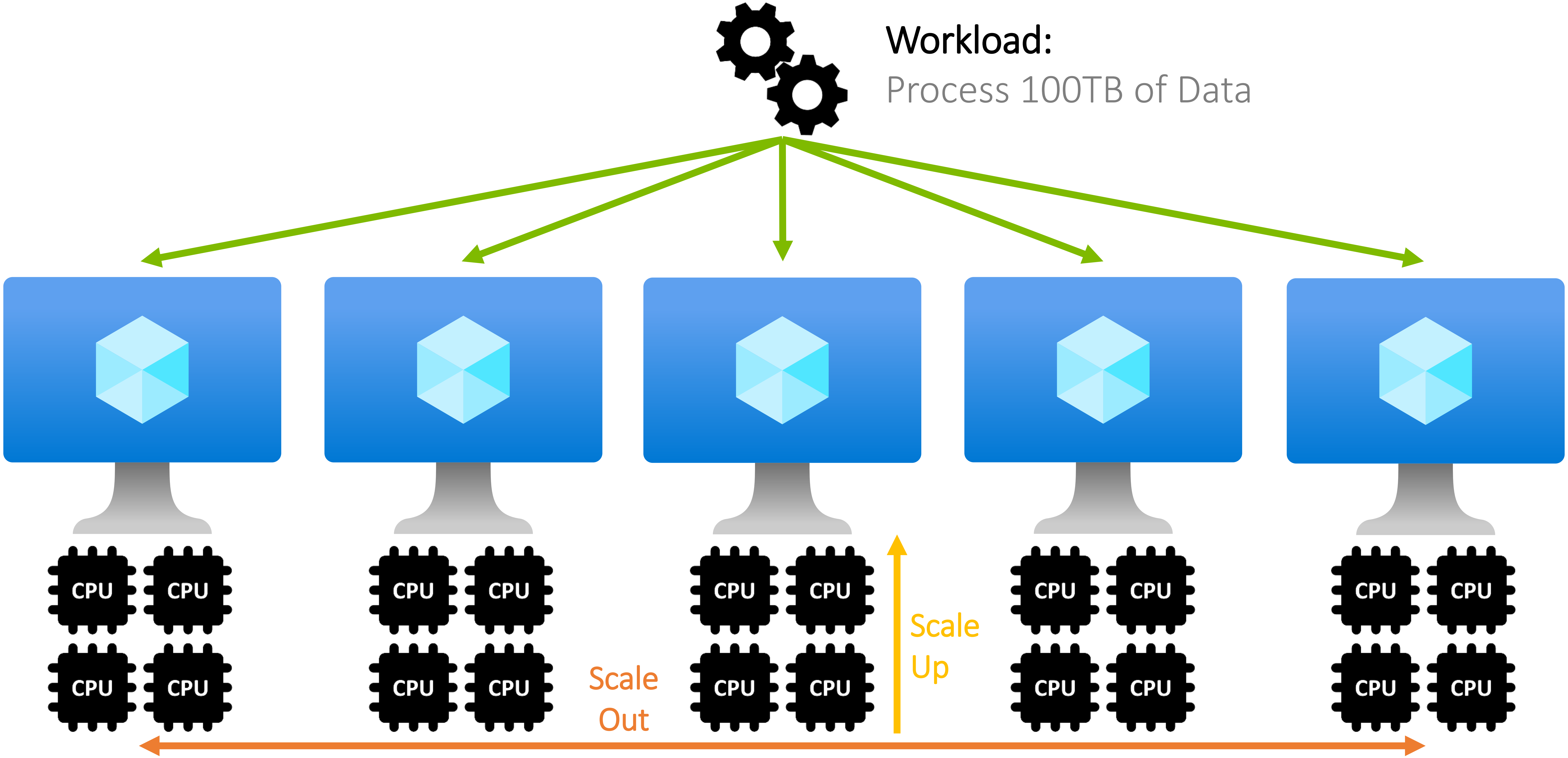
Data Transformation



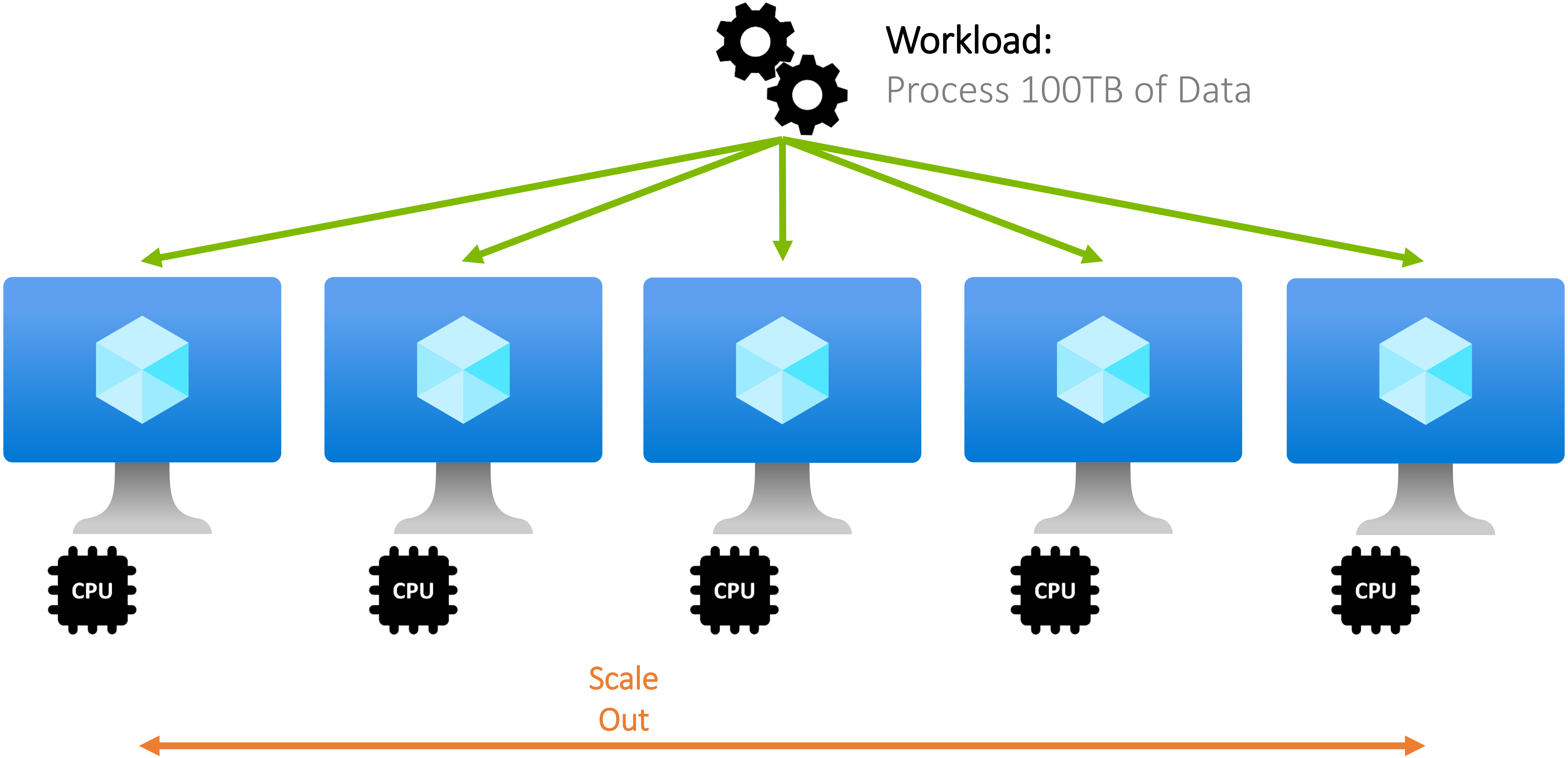
```
BEGIN --get ready

   SELECT
         [Contents]
   FROM
         [Training]
   WHERE
         [Module] = '3';
```

- Data Flows
- Power Query Injection
- Spark Configuration
- Use Cases

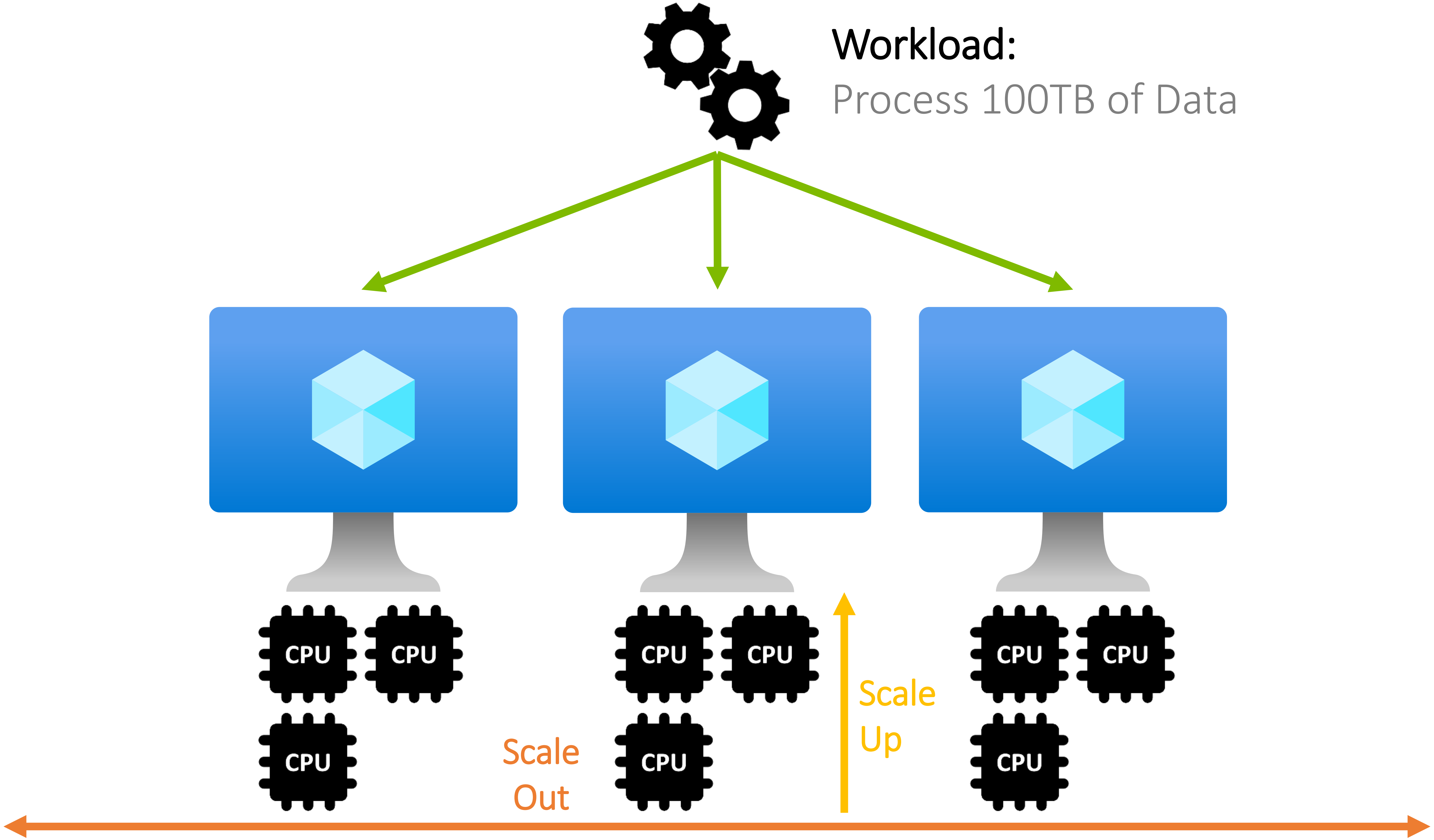
Scaling Up and/or Scaling Out



Scaling Up and/or Scaling Out

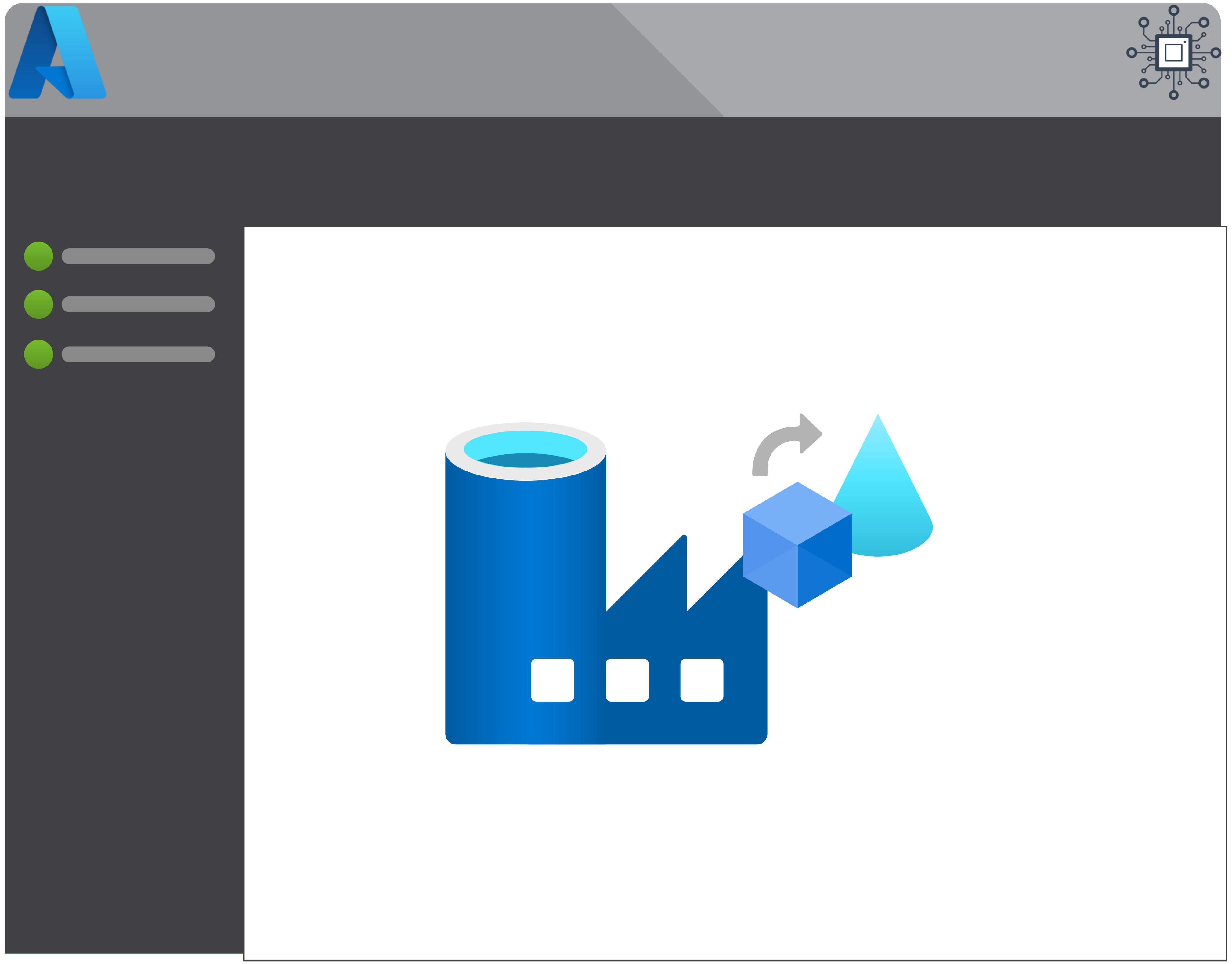

Workload:
Process 100TB of Data

Scaling Up and/or Scaling Out



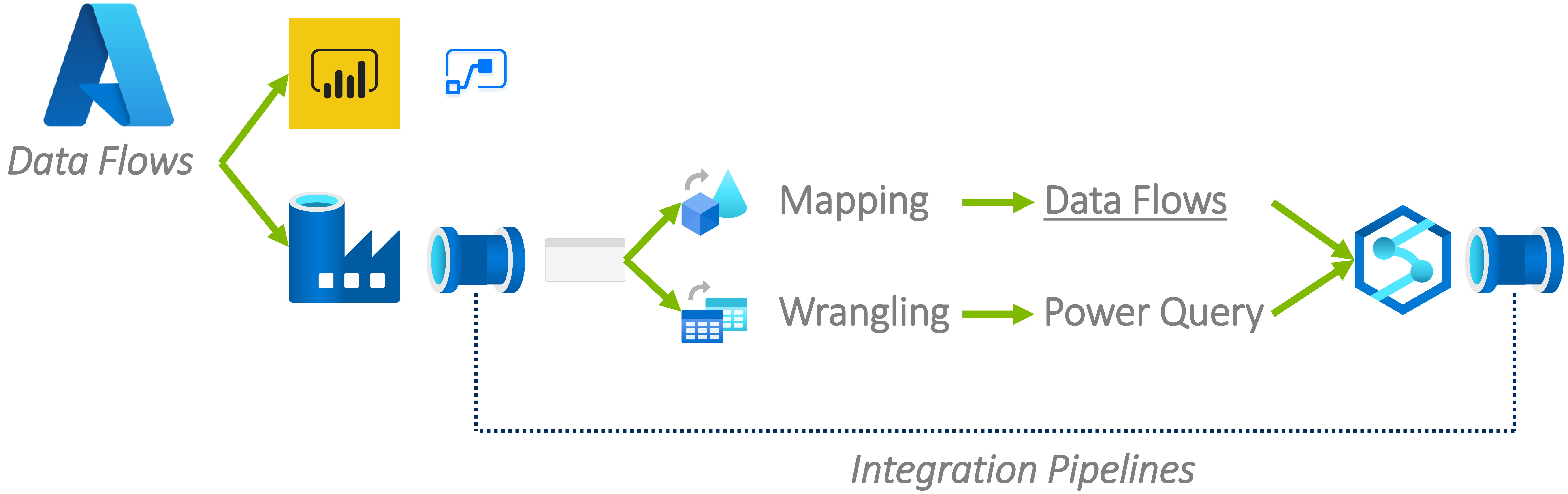
Module 3

Data Transformation



- Data Flows
- Power Query Injection
- Spark Configuration
- Use Cases

Terminology Clarification



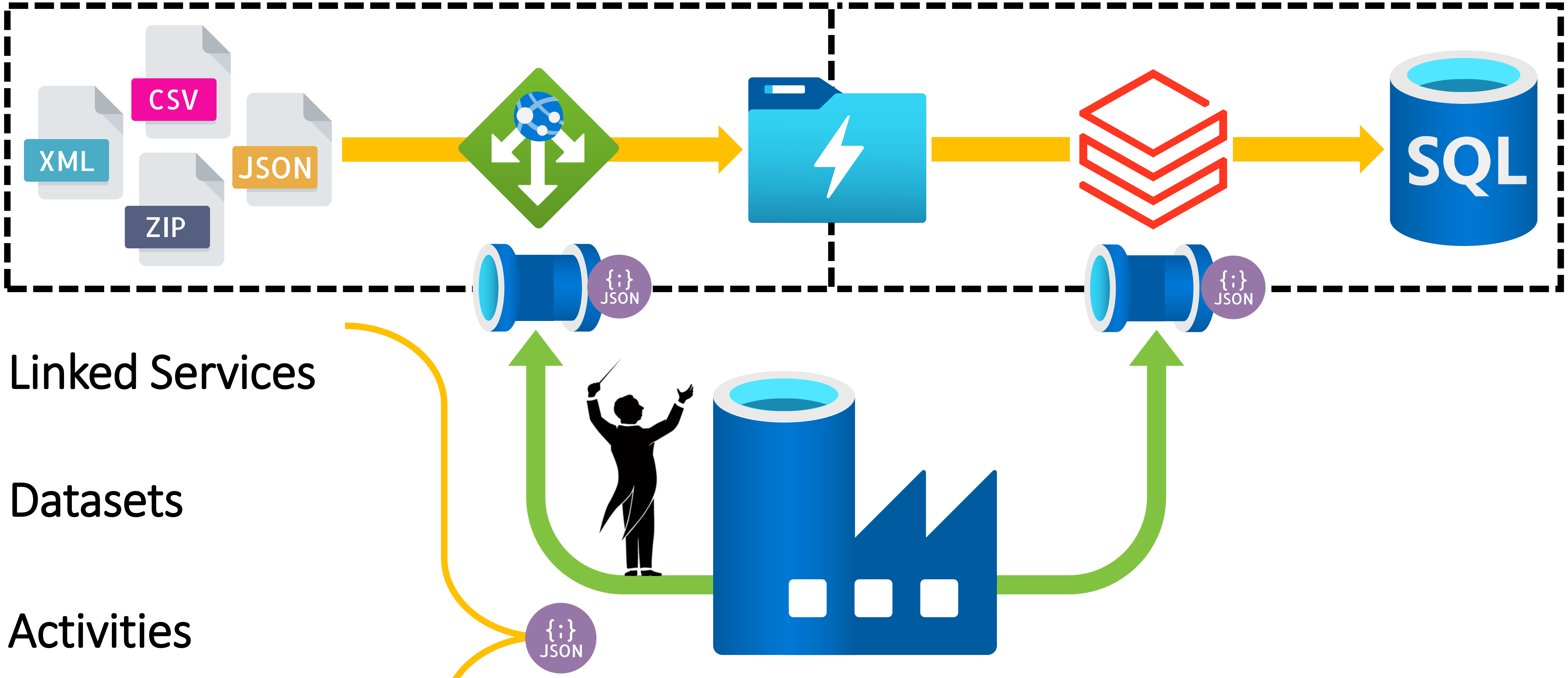
Data Flows

Mapping → Data Flows

Wrangling → Power Query

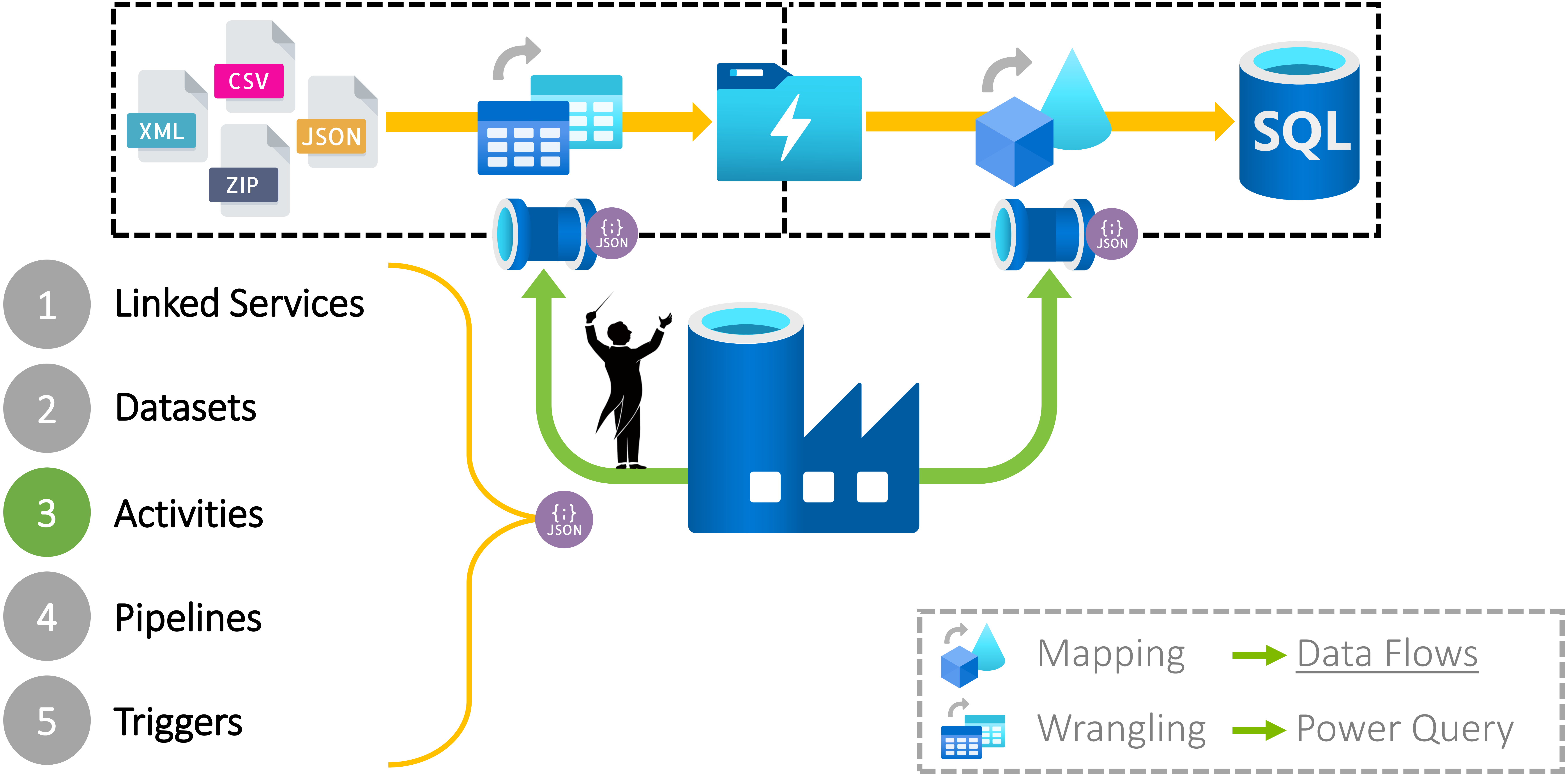
Integration Pipelines

Control Flow Components



1. Linked Services
2. Datasets
3. Activities
4. Pipelines
5. Triggers

~~Control~~ Data Flow Components



1 Linked Services

2 Datasets

3 Activities

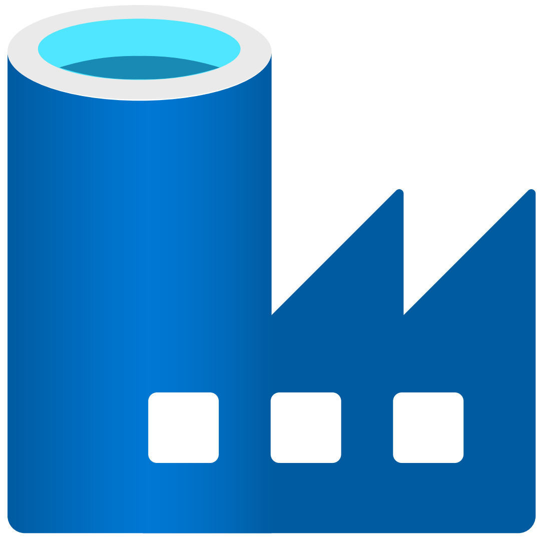
4 Pipelines

5 Triggers

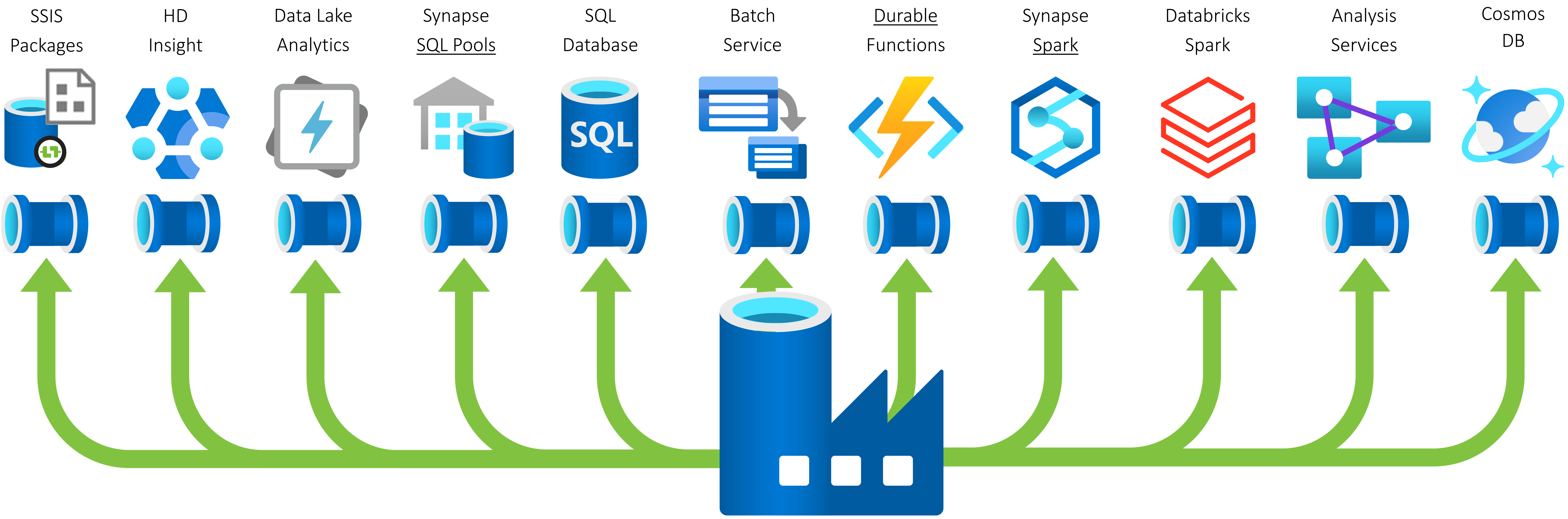
Mapping → Data Flows

Wrangling → Power Query

Data Transformation in Azure

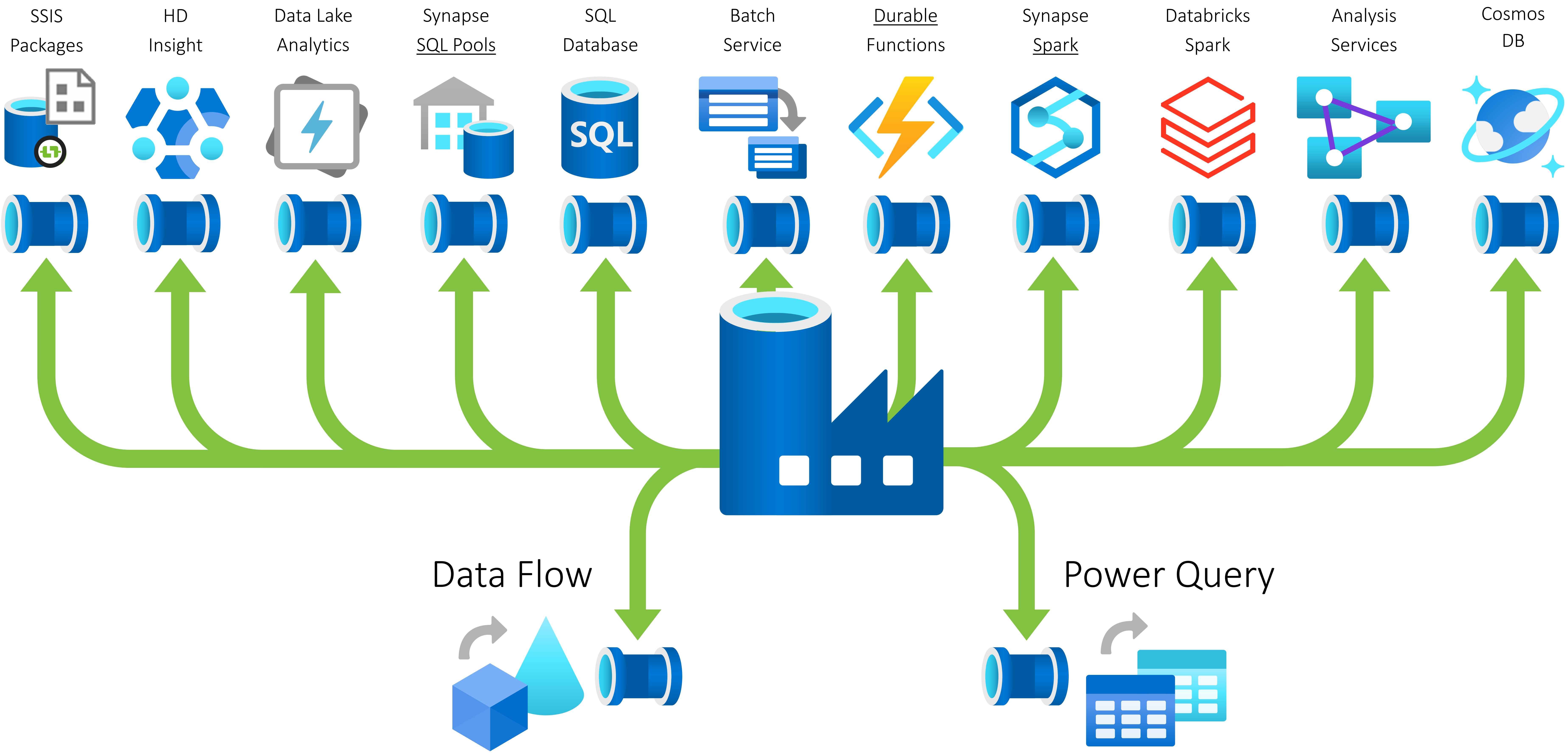


Other Data Transformation Services in Azure



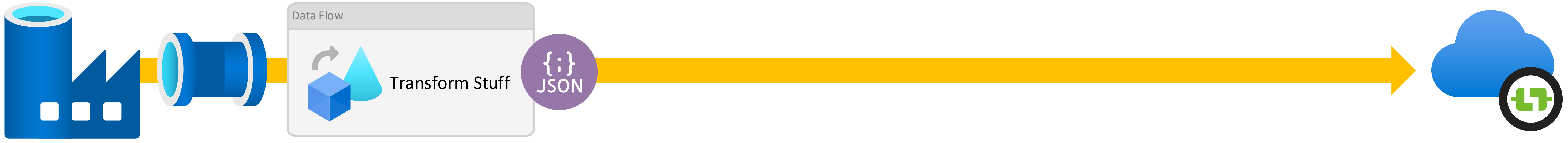
Other Data Transformation Services in Azure

When Should We Use These Integration Pipeline Transformation Activities?

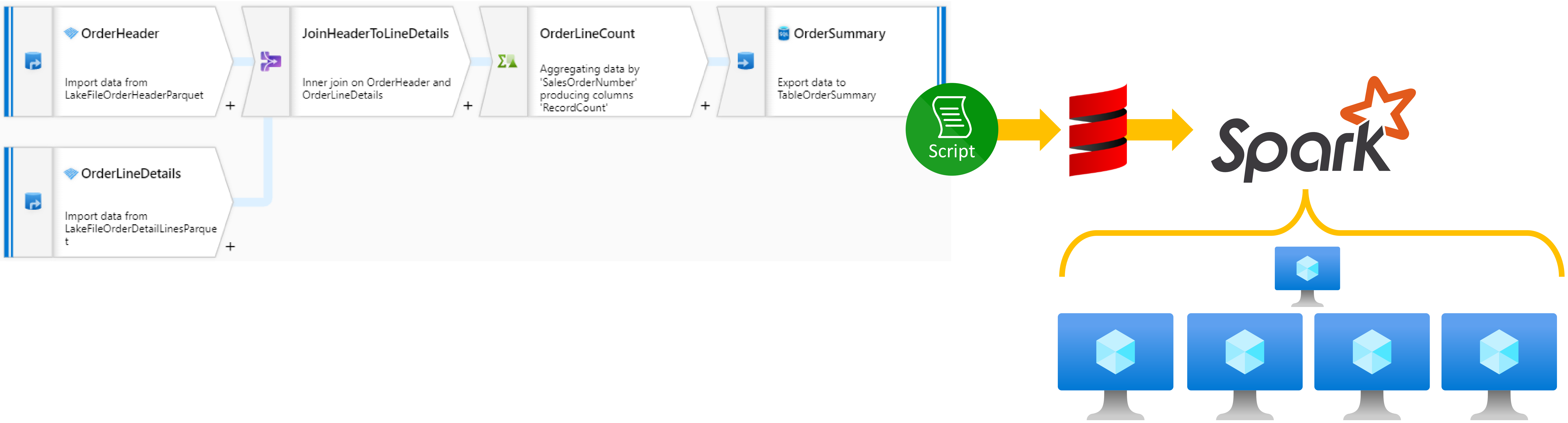


What is a ~~Mapping~~ Data Flow?

Control Flow



Data Flow



OrderHeader
Import data from LakeFileOrderHeaderParquet

OrderLineDetails
Import data from LakeFileOrderDetailLinesParquet

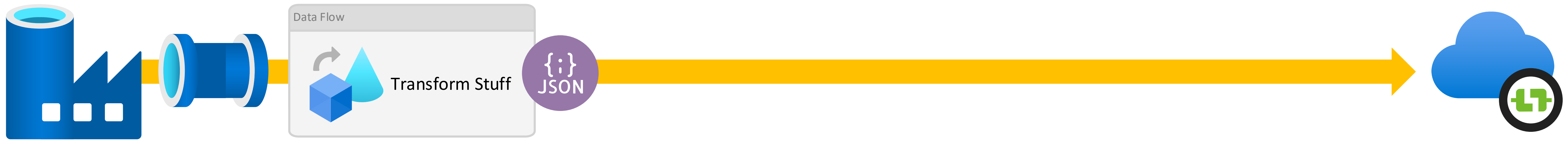
JoinHeaderToLineDetails
Inner join on OrderHeader and OrderLineDetails

OrderLineCount
Aggregating data by 'SalesOrderNumber' producing columns 'RecordCount'

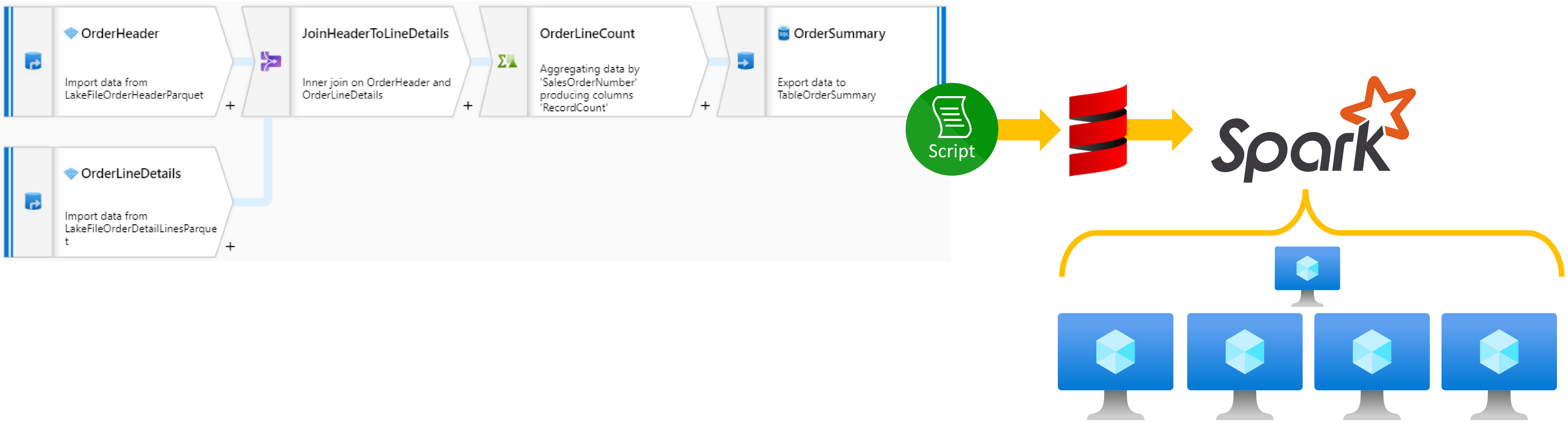
OrderSummary
Export data to TableOrderSummary

Q: What is a ~~Mapping~~ Data Flow?

Control Flow



Data Flow



OrderHeader

Import data from LakeFileOrderHeaderParquet

JoinHeaderToLineDetails

Inner join on OrderHeader and OrderLineDetails

OrderLineCount

Aggregating data by 'SalesOrderNumber' producing columns 'RecordCount'

OrderSummary

Export data to TableOrderSummary

OrderLineDetails

Import data from LakeFileOrderDetailLinesParquet

A: Graphic no low/low code data transformation tool that sits on top of Apache Spark.

Data Flows – Inputs & Outputs

Source & Sink		
Linked Services		SQL My SQL
Source Types	Dataset ▦	AVRO X X CSV JSON XML
	Inline ▨	HIVE

Data Flows – Transformations

New Branch

Join

Conditional Split

Exists

Union

Lookup

Derived Column

Select

Aggregate

Surrogate Key

Pivot/Unpivot

Window

Rank

External Call

Cast

Flatten

Parse

Stringify

Filter

Sort

Alter Row

Assert

Flowlet

Key

Input & Output Modifiers

Schema Modifiers

Formatters

Row Modifiers

Data Flows – Transformations

https://sqlplayer.net/2018/12/azure-data-factory-v2-and-its-available-components-in-data-flows/

New Branch

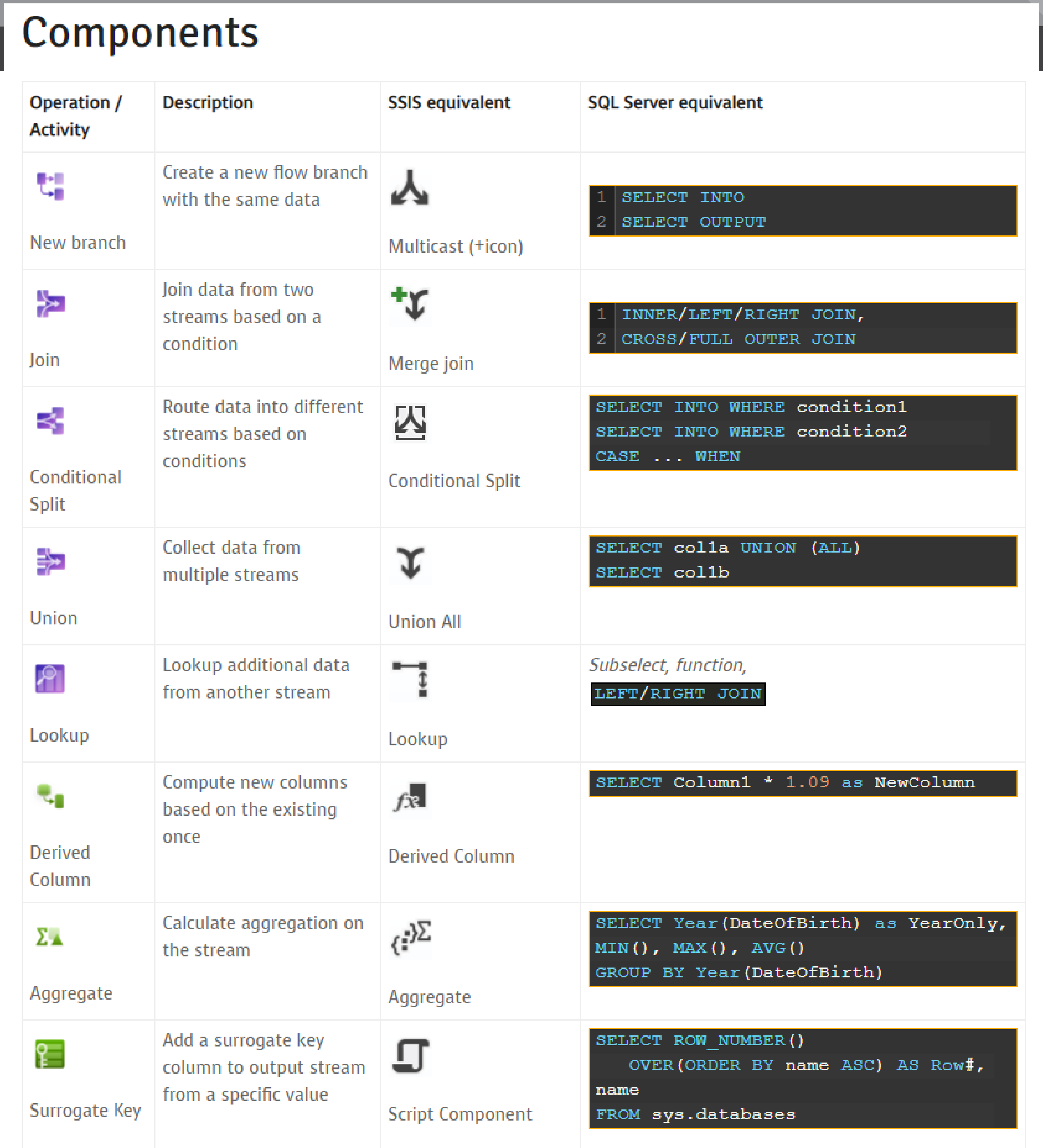
Join

Conditional Split

Exists

Union

Lookup



Components

Operation / Activity	Description	SSIS equivalent	SQL Server equivalent

New branch	Create a new flow branch with the same data	Multicast (+icon)	`SELECT INTO`
			`SELECT OUTPUT`

Join	Join data from two streams based on a condition	Merge join	`INNER/LEFT/RIGHT JOIN,`
			`CROSS/FULL OUTER JOIN`

Conditional Split	Route data into different streams based on conditions	Conditional Split	`SELECT INTO WHERE condition1`
			`SELECT INTO WHERE condition2`
			`CASE ... WHEN`

Union	Collect data from multiple streams	Union All	`SELECT col1a UNION (ALL)`
			`SELECT col1b`

Lookup	Lookup additional data from another stream	Lookup	*Subselect, function,* `LEFT/RIGHT JOIN`
Derived Column	Compute new columns based on the existing once	Derived Column	`SELECT Column1 * 1.09 as NewColumn`

Aggregate	Calculate aggregation on the stream	Aggregate	`SELECT Year(DateOfBirth) as YearOnly,`
			`MIN(), MAX(), AVG()`
			`GROUP BY Year(DateOfBirth)`

Surrogate Key	Add a surrogate key column to output stream from a specific value	Script Component	`SELECT ROW_NUMBER()`
			`   OVER(ORDER BY name ASC) AS Row#,`
			`name`
			`FROM sys.databases`

Filter

Sort

Alter Row

Key

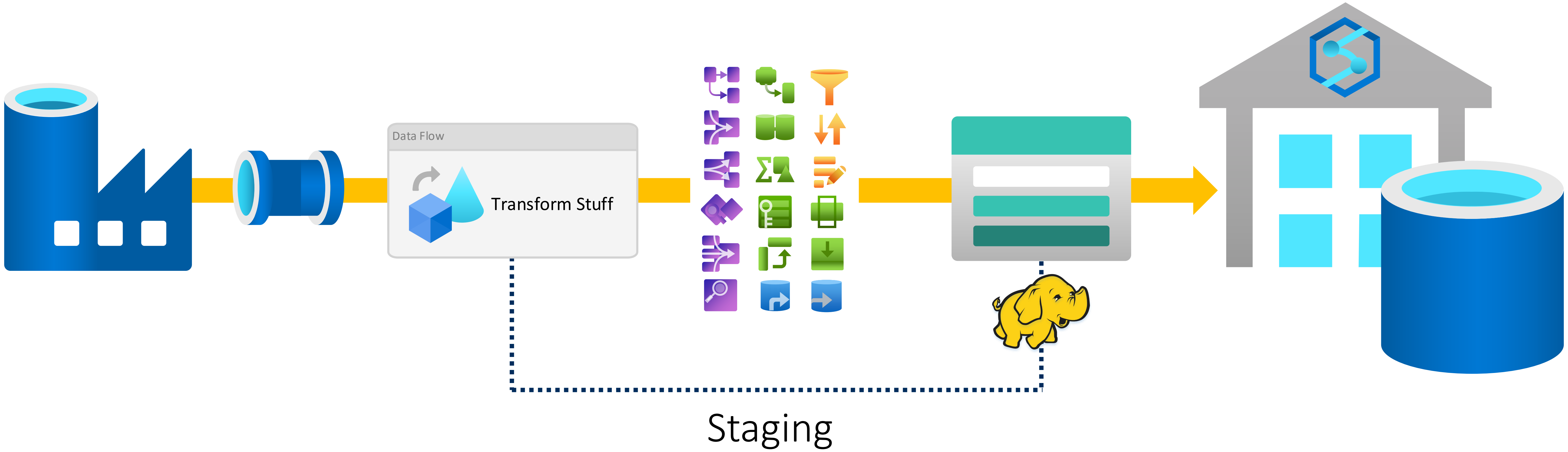
Input & Output Modifiers

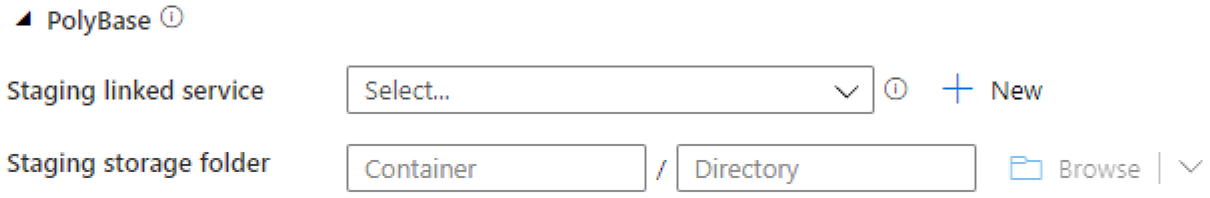
Schema Modifiers

Formatters

Row Modifiers

Data Flows – Data Warehouse Loading (PolyBase)





◢ PolyBase ⓘ

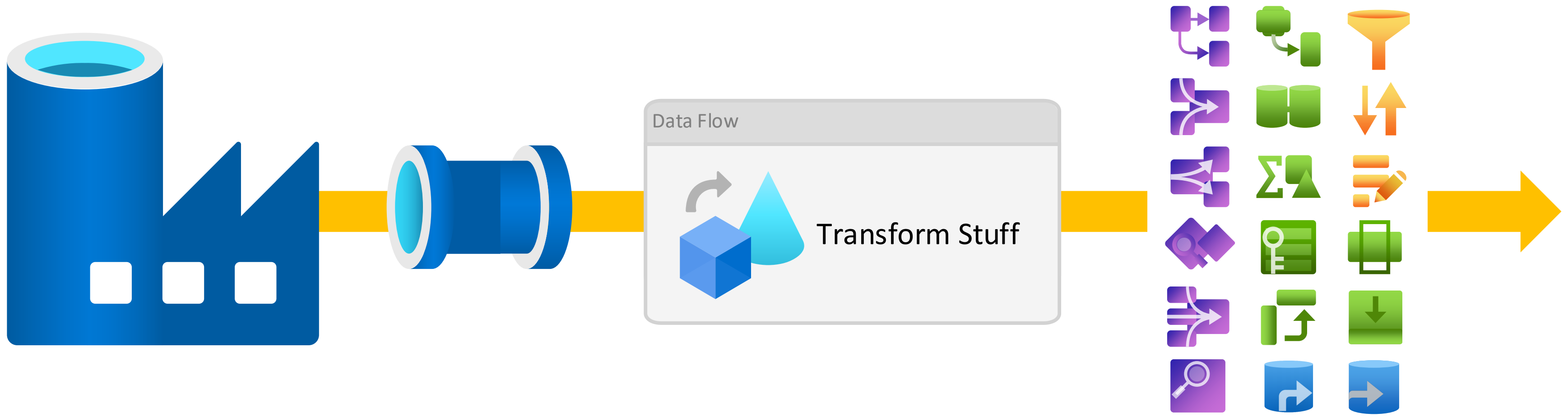
Staging linked service: Select... ⓘ

＋ New

Staging storage folder: Container / Directory

Browse

Data Flows – Expression Builder

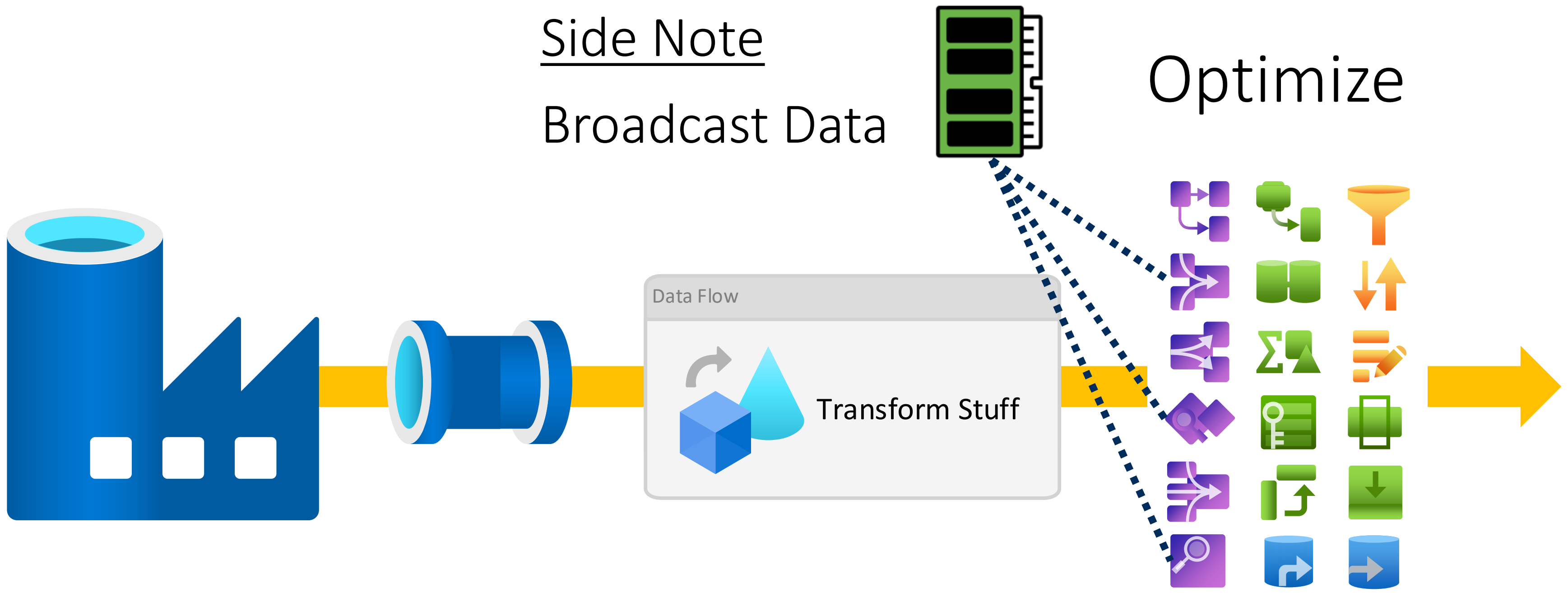


Expression Builder

- Count If
- Sum If
- Regex Extract
- Regex Match
- Regex Split
- R-Like

https://docs.microsoft.com/en-gb/azure/data-factory/data-flow-expression-functions

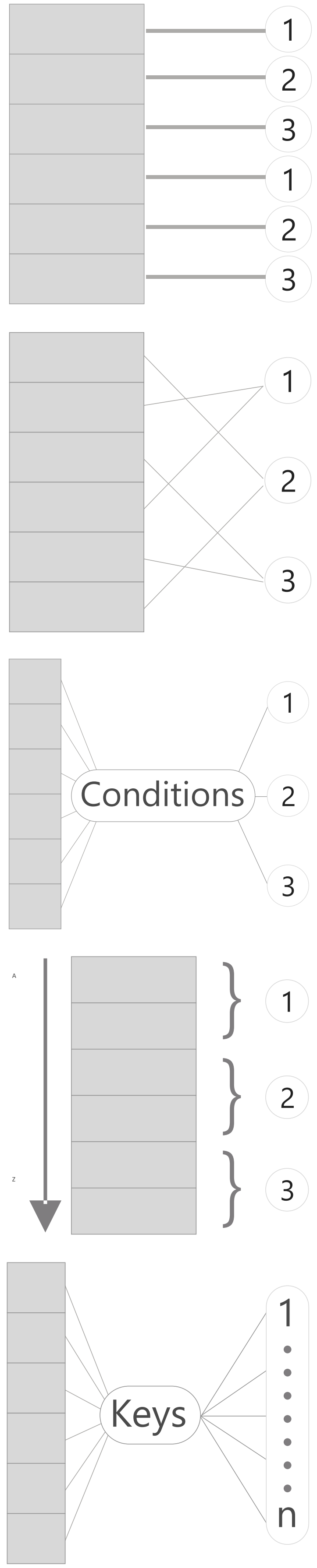
Data Flows – Data Distribution



Side Note

Broadcast Data

Optimize



Round Robin

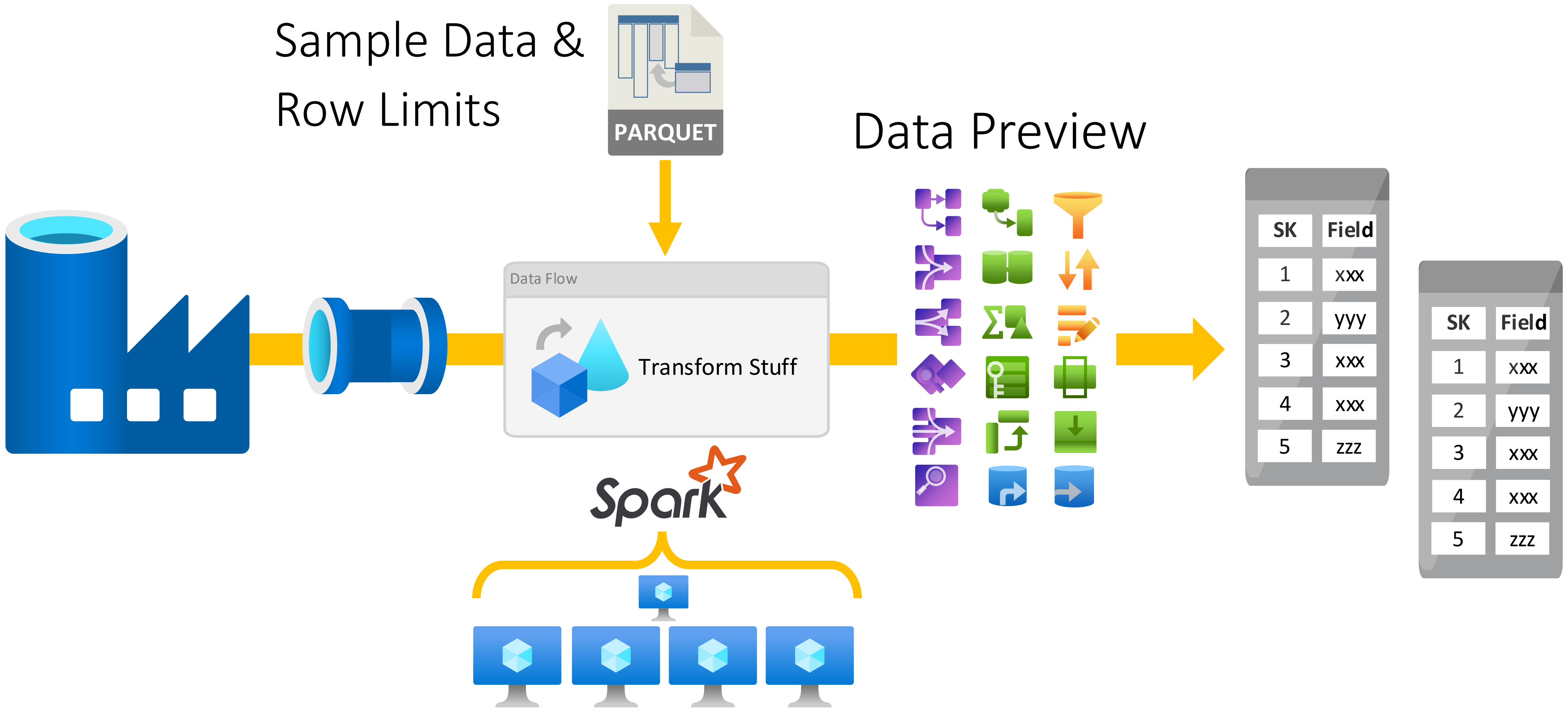
Hashed

Conditional

Dynamic Ranges

Key Based

Data Flows – Debugging

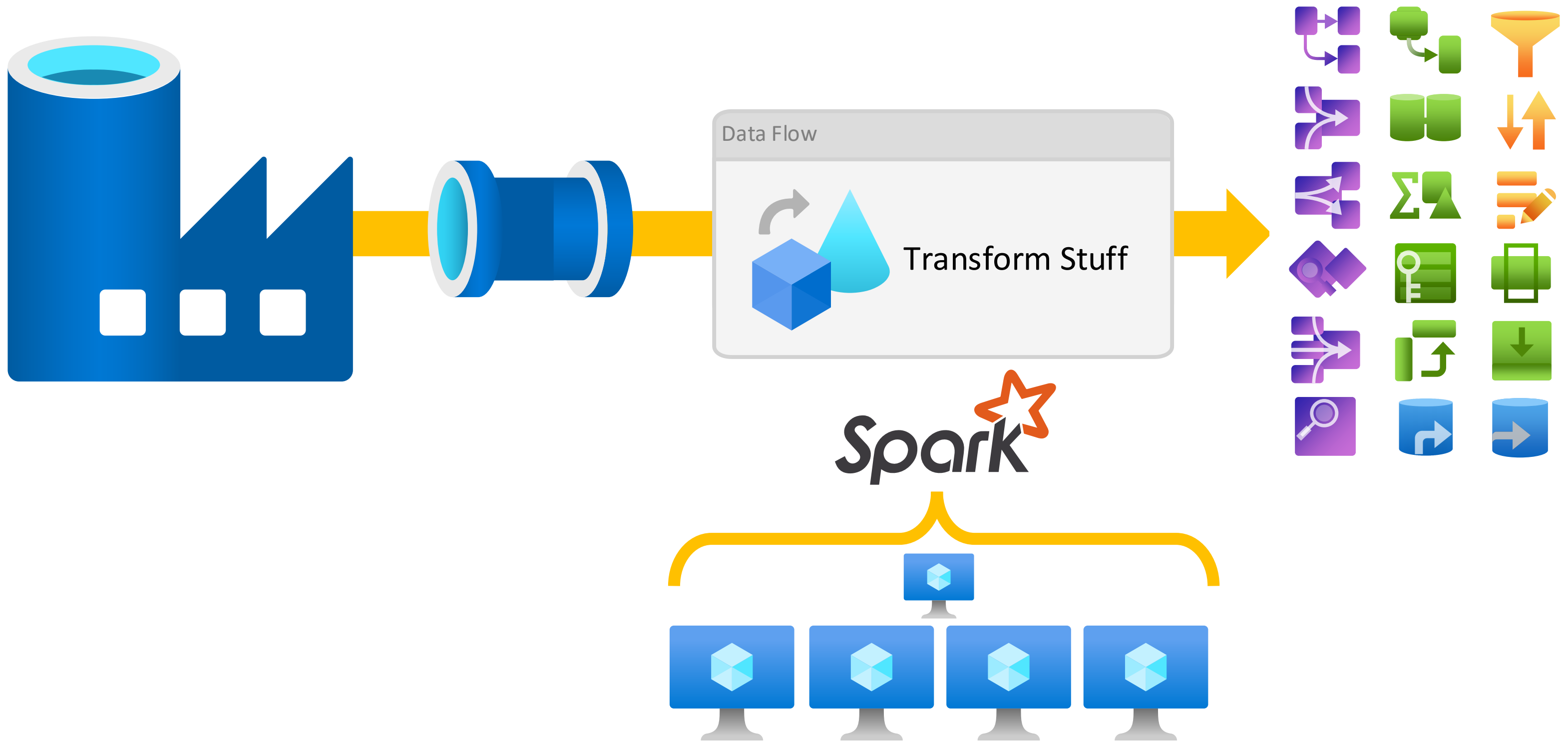


SK	Field
1	xxx
2	yyy
3	xxx
4	xxx
5	zzz

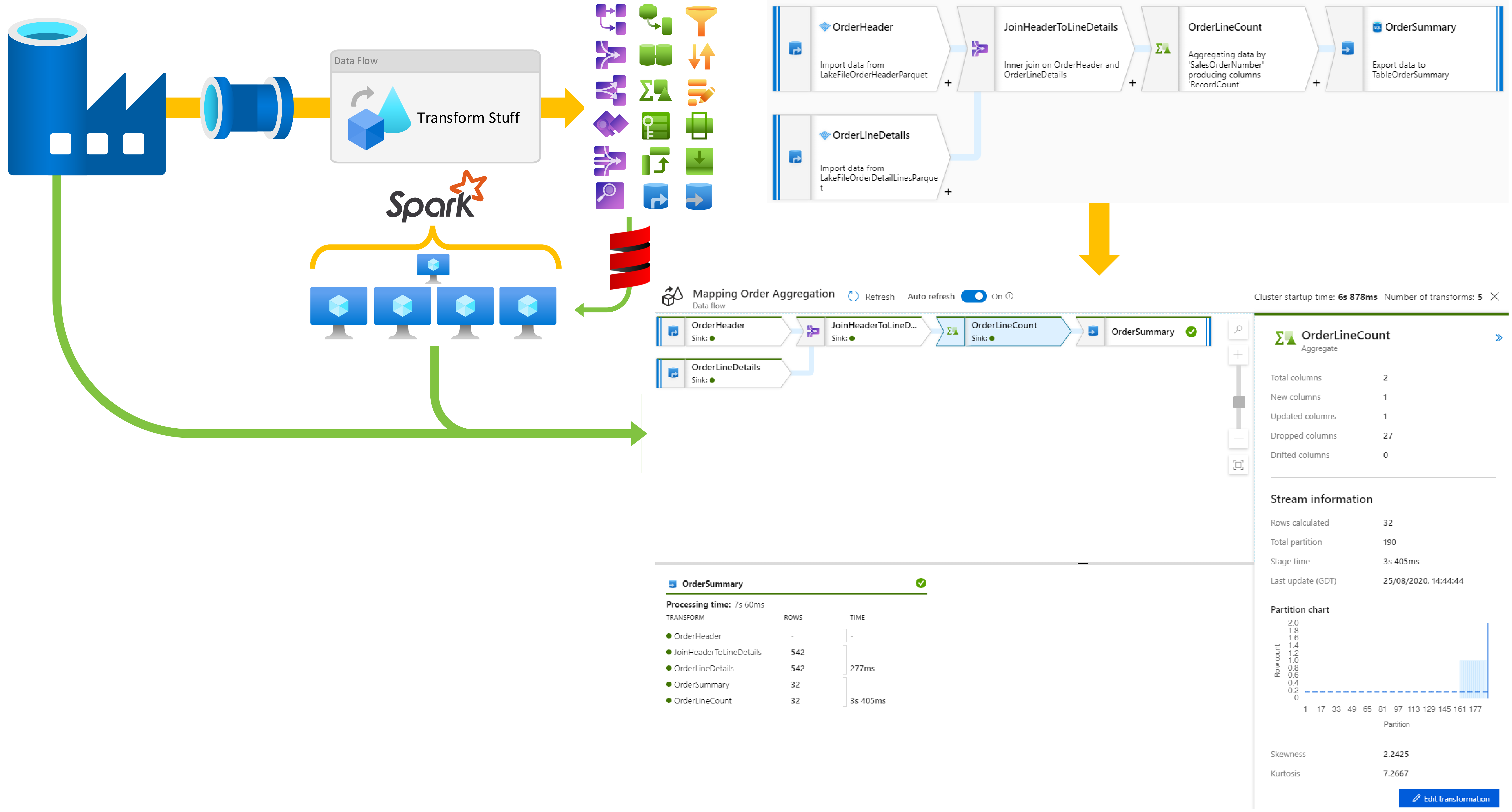
SK	Field
1	xxx
2	yyy
3	xxx
4	xxx
5	zzz

Enable Data Flow Debug Mode

Data Flows – Monitoring



Data Flows – Monitoring



OrderHeader
Import data from LakeFileOrderHeaderParquet

JoinHeaderToLineDetails
Inner join on OrderHeader and OrderLineDetails

OrderLineCount
Aggregating data by 'SalesOrderNumber' producing columns 'RecordCount'

OrderSummary
Export data to TableOrderSummary

OrderLineDetails
Import data from LakeFileOrderDetailLinesParque t

Mapping Order Aggregation
Data flow

Cluster startup time: **6s 878ms** Number of transforms: **5**

OrderHeader
Sink: ●

JoinHeaderToLineD...
Sink: ●

OrderLineCount
Sink: ●

OrderSummary ✓

OrderLineDetails
Sink: ●

OrderSummary ✓

Processing time: 7s 60ms

TRANSFORM	ROWS	TIME
● OrderHeader	-	-
● JoinHeaderToLineDetails	542	
● OrderLineDetails	542	277ms
● OrderSummary	32	
● OrderLineCount	32	3s 405ms

OrderLineCount
Aggregate

Total columns	2
New columns	1
Updated columns	1
Dropped columns	27
Drifted columns	0

Stream information

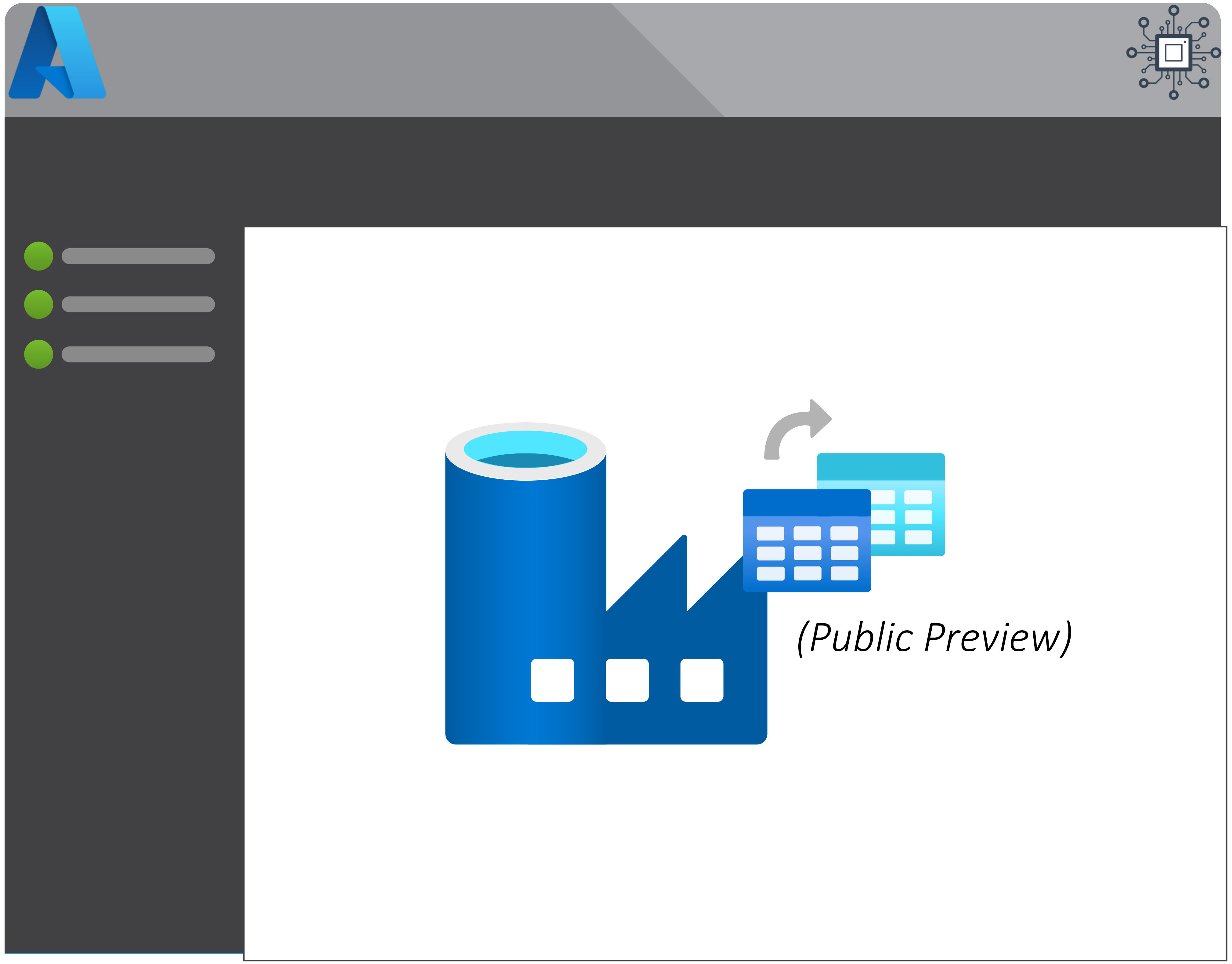
Rows calculated	32
Total partition	190
Stage time	3s 405ms
Last update (GDT)	25/08/2020, 14:44:44

Partition chart

Skewness	2.2425
Kurtosis	7.2667

Module 3

Data Transformation

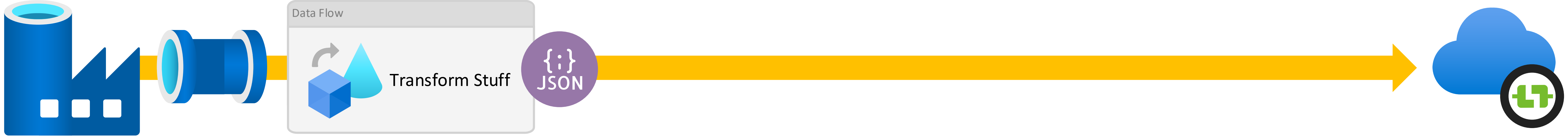


(Public Preview)

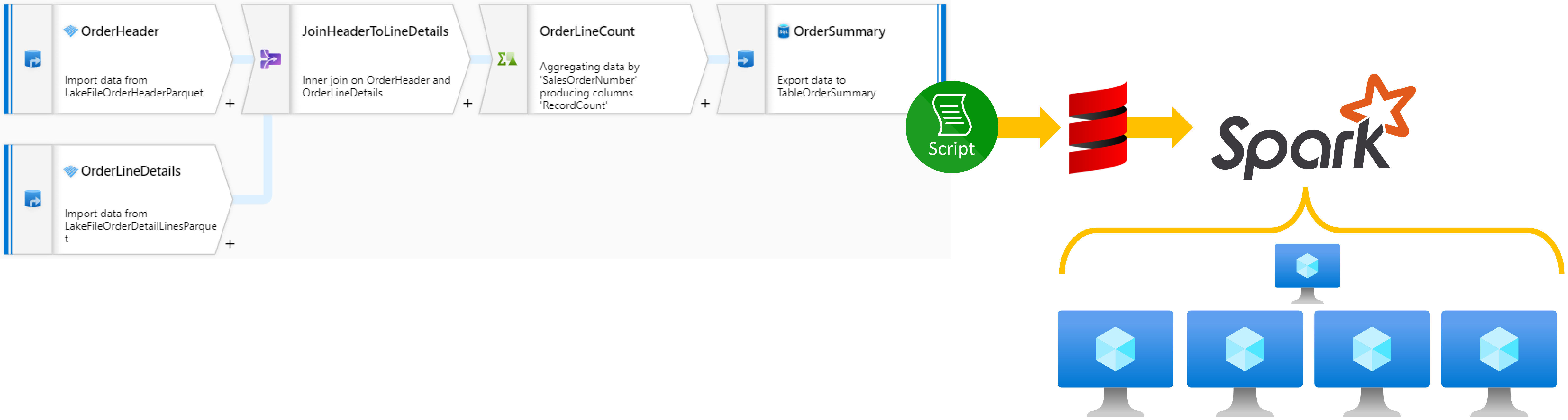
- Data Flows
- Power Query Injection
- Spark Configuration
- Use Cases

What is a Data Flow?

Control Flow



Data Flow



OrderHeader

Import data from LakeFileOrderHeaderParquet

JoinHeaderToLineDetails

Inner join on OrderHeader and OrderLineDetails

OrderLineCount

Aggregating data by 'SalesOrderNumber' producing columns 'RecordCount'

OrderSummary

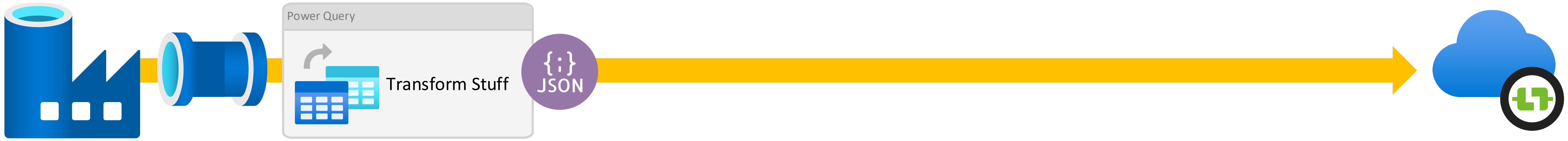
Export data to TableOrderSummary

OrderLineDetails

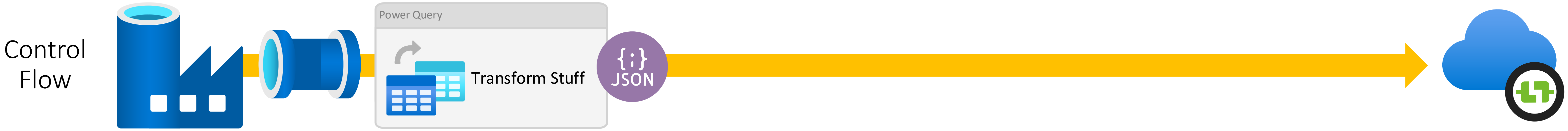
Import data from LakeFileOrderDetailLinesParquet

What is a Power Query Activity?

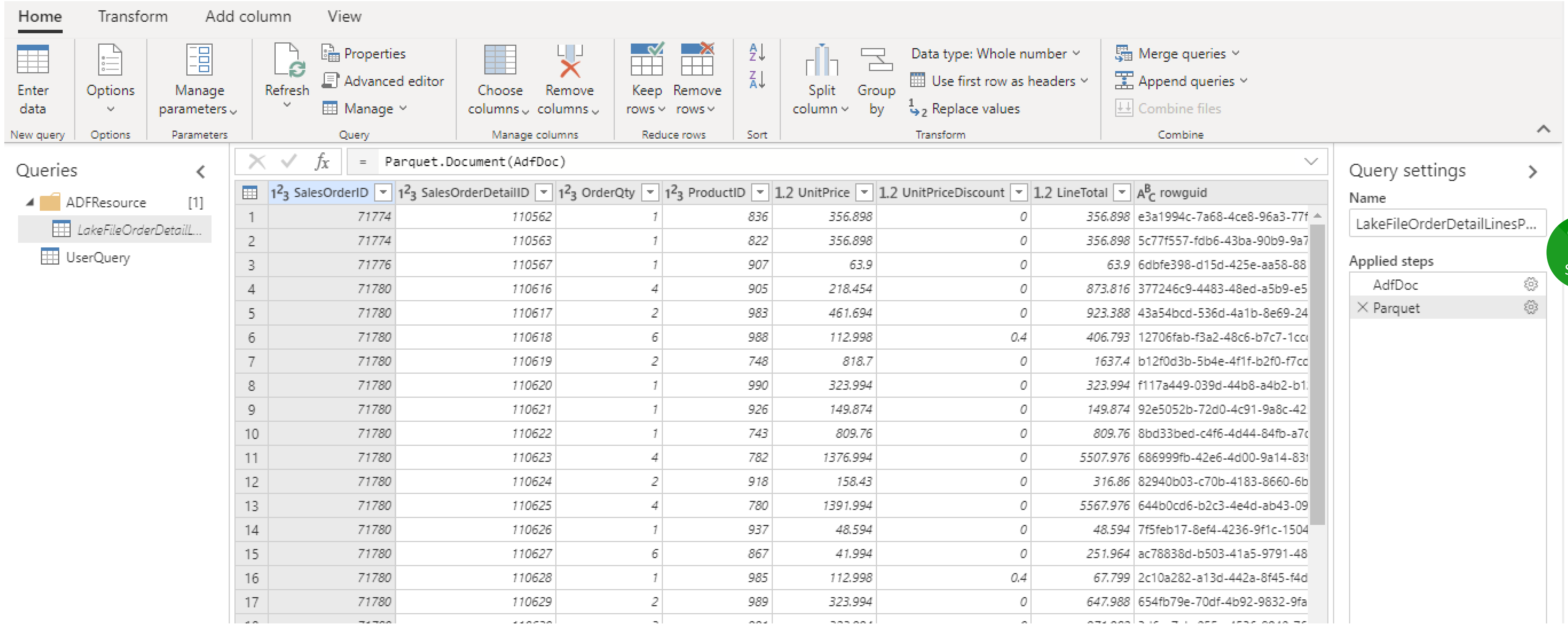
Control Flow



What is a Power Query Activity?



Power Query



```
= Parquet.Document(AdfDoc)
```

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid
1	71774	110562	1	836	356.898	0	356.898	e3a1994c-7a68-4ce8-96a3-77f
2	71774	110563	1	822	356.898	0	356.898	5c77f557-fdb6-43ba-90b9-9a7
3	71776	110567	1	907	63.9	0	63.9	6dbfe398-d15d-425e-aa58-88
4	71780	110616	4	905	218.454	0	873.816	377246c9-4483-48ed-a5b9-e5
5	71780	110617	2	983	461.694	0	923.388	43a54bcd-536d-4a1b-8e69-24
6	71780	110618	6	988	112.998	0.4	406.793	12706fab-f3a2-48c6-b7c7-1cc
7	71780	110619	2	748	818.7	0	1637.4	b12f0d3b-5b4e-4f1f-b2f0-f7cc
8	71780	110620	1	990	323.994	0	323.994	f117a449-039d-44b8-a4b2-b1
9	71780	110621	1	926	149.874	0	149.874	92e5052b-72d0-4c91-9a8c-42
10	71780	110622	1	743	809.76	0	809.76	8bd33bed-c4f6-4d44-84fb-a7c
11	71780	110623	4	782	1376.994	0	5507.976	686999fb-42e6-4d00-9a14-83
12	71780	110624	2	918	158.43	0	316.86	82940b03-c70b-4183-8660-6b
13	71780	110625	4	780	1391.994	0	5567.976	644b0cd6-b2c3-4e4d-ab43-09
14	71780	110626	1	937	48.594	0	48.594	7f5feb17-8ef4-4236-9f1c-1504
15	71780	110627	6	867	41.994	0	251.964	ac78838d-b503-41a5-9791-48
16	71780	110628	1	985	112.998	0.4	67.799	2c10a282-a13d-442a-8f45-f4d
17	71780	110629	2	989	323.994	0	647.988	654fb79e-70df-4b92-9832-9fa

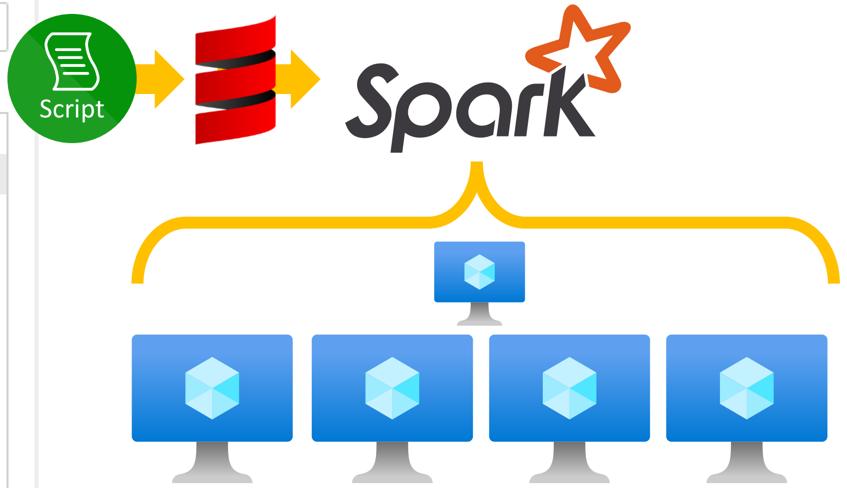
Query settings

Name

LakeFileOrderDetailLinesP...

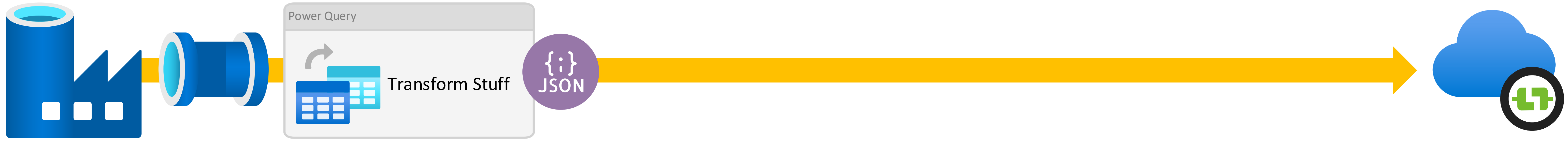
Applied steps

- AdfDoc
- Parquet

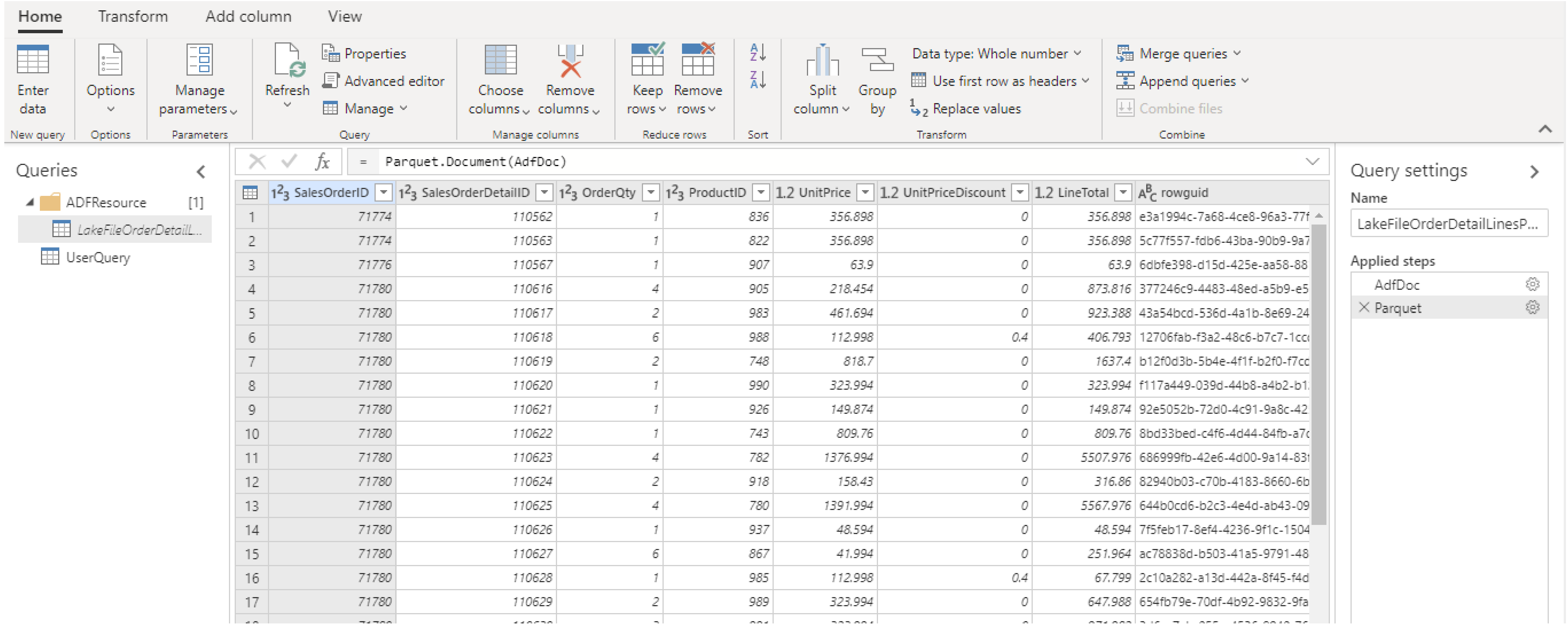


What can a Power Query Activity do?

Control Flow



Power Query



```
= Parquet.Document(AdfDoc)
```

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid
1	71774	110562	1	836	356.898	0	356.898	e3a1994c-7a68-4ce8-96a3-77f
2	71774	110563	1	822	356.898	0	356.898	5c77f557-fdb6-43ba-90b9-9a7
3	71776	110567	1	907	63.9	0	63.9	6dbfe398-d15d-425e-aa58-88
4	71780	110616	4	905	218.454	0	873.816	377246c9-4483-48ed-a5b9-e5
5	71780	110617	2	983	461.694	0	923.388	43a54bcd-536d-4a1b-8e69-24
6	71780	110618	6	988	112.998	0.4	406.793	12706fab-f3a2-48c6-b7c7-1cc
7	71780	110619	2	748	818.7	0	1637.4	b12f0d3b-5b4e-4f1f-b2f0-f7cc
8	71780	110620	1	990	323.994	0	323.994	f117a449-039d-44b8-a4b2-b1
9	71780	110621	1	926	149.874	0	149.874	92e5052b-72d0-4c91-9a8c-42
10	71780	110622	1	743	809.76	0	809.76	8bd33bed-c4f6-4d44-84fb-a7c
11	71780	110623	4	782	1376.994	0	5507.976	686999fb-42e6-4d00-9a14-83
12	71780	110624	2	918	158.43	0	316.86	82940b03-c70b-4183-8660-6b
13	71780	110625	4	780	1391.994	0	5567.976	644b0cd6-b2c3-4e4d-ab43-09
14	71780	110626	1	937	48.594	0	48.594	7f5feb17-8ef4-4236-9f1c-1504
15	71780	110627	6	867	41.994	0	251.964	ac78838d-b503-41a5-9791-48
16	71780	110628	1	985	112.998	0.4	67.799	2c10a282-a13d-442a-8f45-f4d
17	71780	110629	2	989	323.994	0	647.988	654fb79e-70df-4b92-9832-9fa

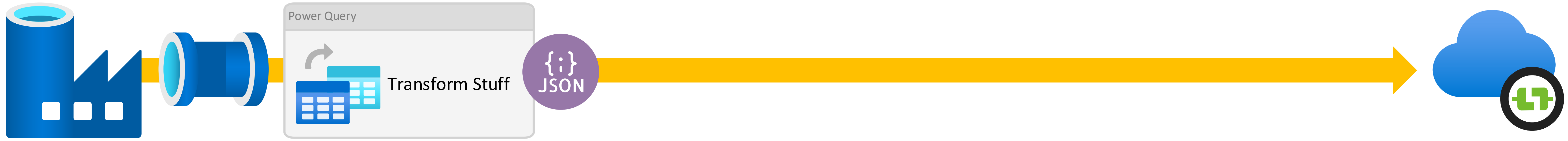
Name: LakeFileOrderDetailLinesP...

Applied steps
- AdfDoc
- Parquet

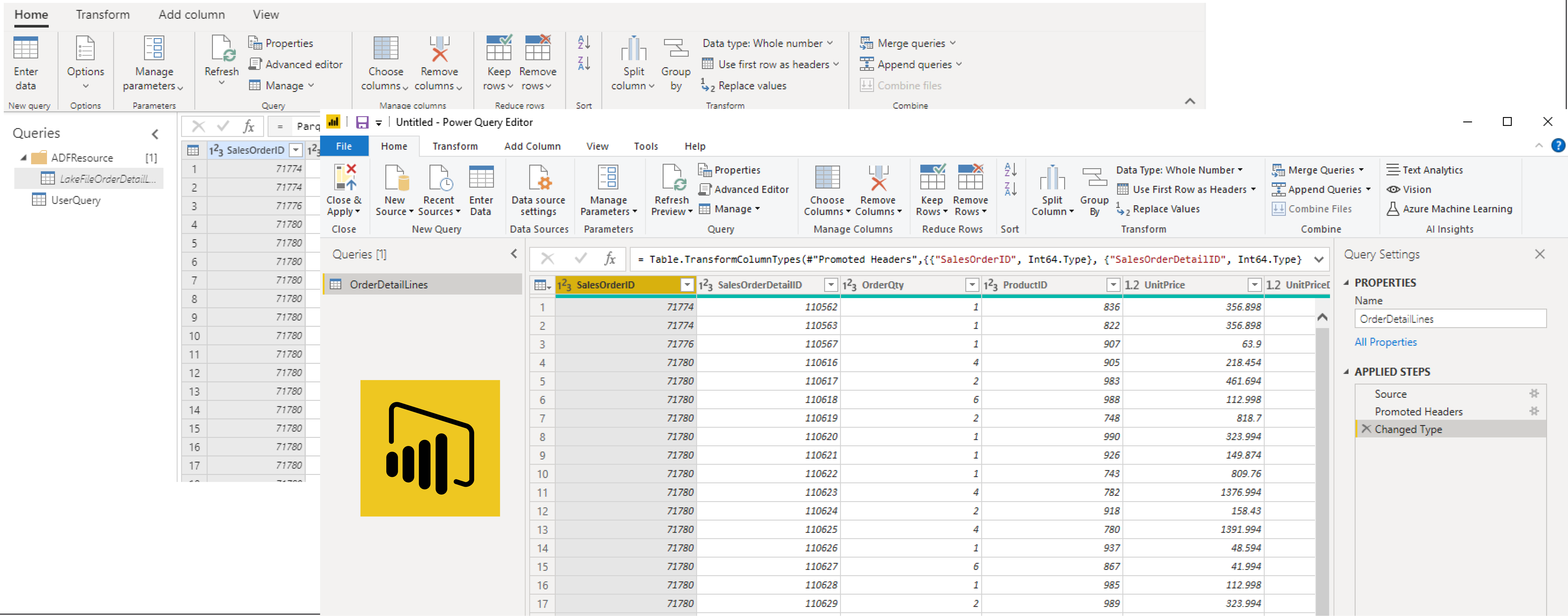
What can a Power Query Activity do?

Home

Control Flow



Power Query



= Parq

	SalesOrderID
1	71774
2	71774
3	71776
4	71780
5	71780
6	71780
7	71780
8	71780
9	71780
10	71780
11	71780
12	71780
13	71780
14	71780
15	71780
16	71780
17	71780

```
= Table.TransformColumnTypes(#"Promoted Headers",{{"SalesOrderID", Int64.Type}, {"SalesOrderDetailID", Int64.Type}
```

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPriceD
1	71774	110562	1	836	356.898	
2	71774	110563	1	822	356.898	
3	71776	110567	1	907	63.9	
4	71780	110616	4	905	218.454	
5	71780	110617	2	983	461.694	
6	71780	110618	6	988	112.998	
7	71780	110619	2	748	818.7	
8	71780	110620	1	990	323.994	
9	71780	110621	1	926	149.874	
10	71780	110622	1	743	809.76	
11	71780	110623	4	782	1376.994	
12	71780	110624	2	918	158.43	
13	71780	110625	4	780	1391.994	
14	71780	110626	1	937	48.594	
15	71780	110627	6	867	41.994	
16	71780	110628	1	985	112.998	
17	71780	110629	2	989	323.994	

Query Settings

PROPERTIES

Name

OrderDetailLines

All Properties

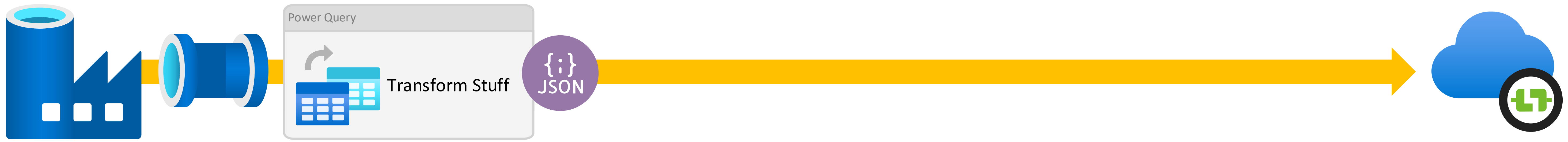
APPLIED STEPS

- Source
- Promoted Headers
- Changed Type

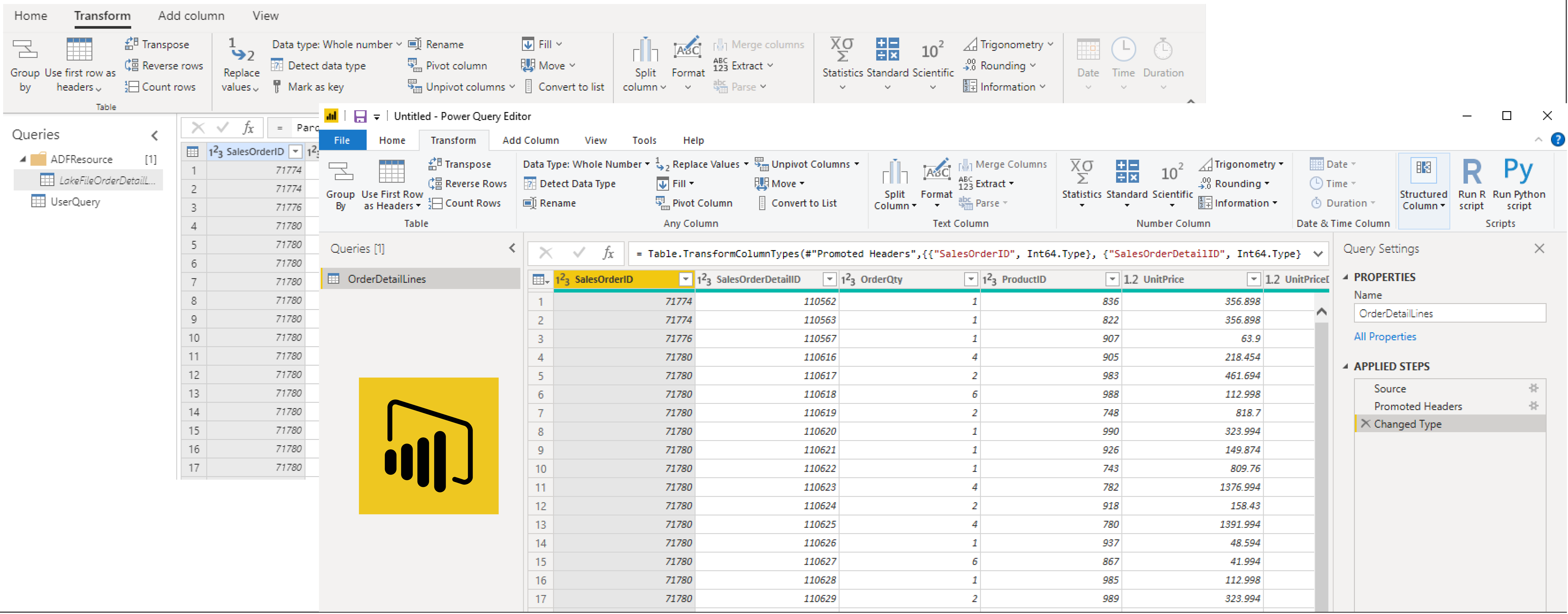
What can a Power Query Activity do?

Transform

Control Flow



Power Query



= Parc

	SalesOrderID
1	71774
2	71774
3	71776
4	71780
5	71780
6	71780
7	71780
8	71780
9	71780
10	71780
11	71780
12	71780
13	71780
14	71780
15	71780
16	71780
17	71780

```
= Table.TransformColumnTypes(#"Promoted Headers",{{"SalesOrderID", Int64.Type}, {"SalesOrderDetailID", Int64.Type}
```

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPriceD
1	71774	110562	1	836	356.898	
2	71774	110563	1	822	356.898	
3	71776	110567	1	907	63.9	
4	71780	110616	4	905	218.454	
5	71780	110617	2	983	461.694	
6	71780	110618	6	988	112.998	
7	71780	110619	2	748	818.7	
8	71780	110620	1	990	323.994	
9	71780	110621	1	926	149.874	
10	71780	110622	1	743	809.76	
11	71780	110623	4	782	1376.994	
12	71780	110624	2	918	158.43	
13	71780	110625	4	780	1391.994	
14	71780	110626	1	937	48.594	
15	71780	110627	6	867	41.994	
16	71780	110628	1	985	112.998	
17	71780	110629	2	989	323.994	

PROPERTIES

Name: OrderDetailLines

All Properties

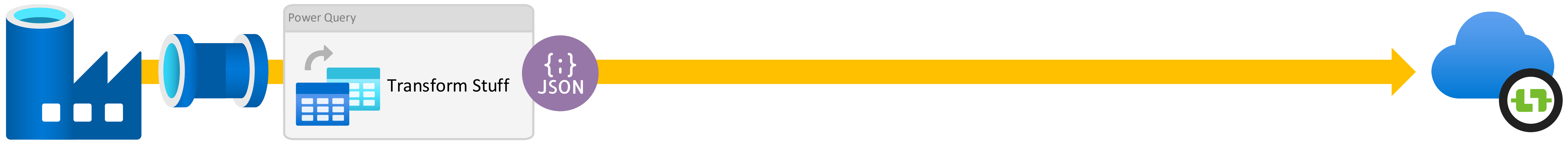
APPLIED STEPS

- Source
- Promoted Headers
- Changed Type

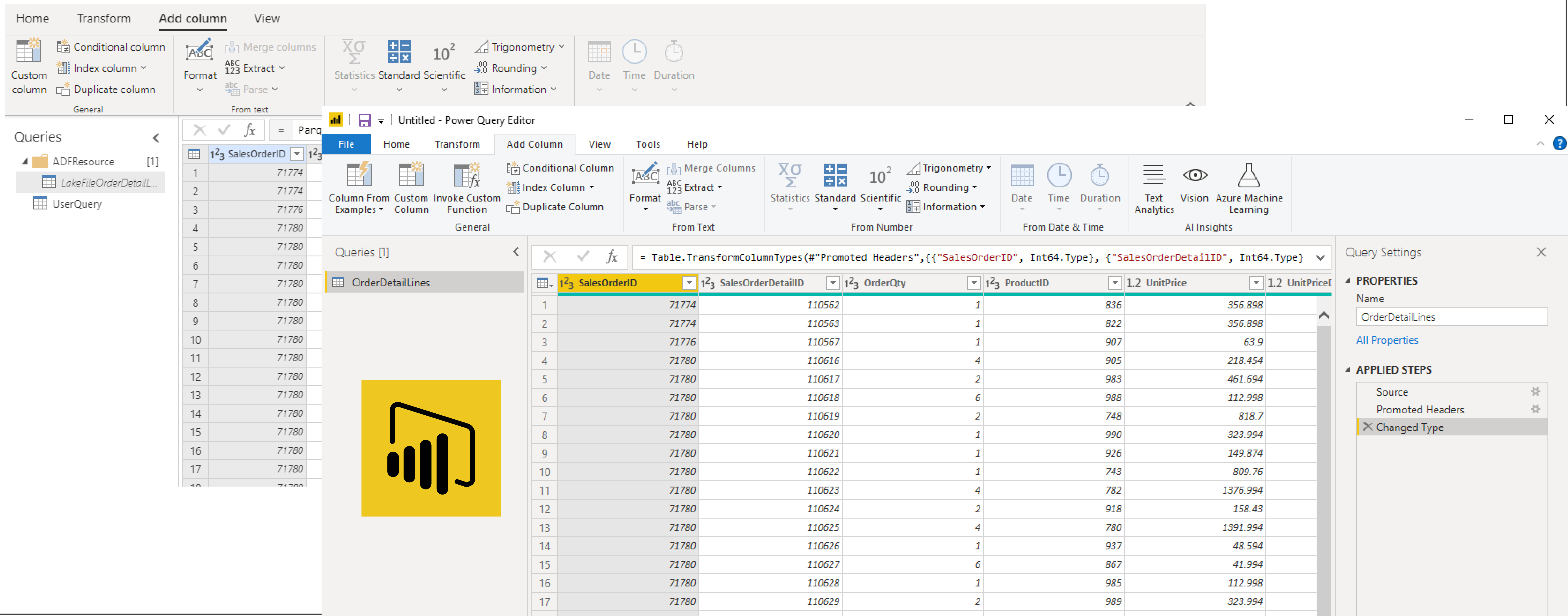
What can a Power Query Activity do?

Add Column

Control Flow



Power Query



= Parq

	SalesOrderID
1	71774
2	71774
3	71776
4	71780
5	71780
6	71780
7	71780
8	71780
9	71780
10	71780
11	71780
12	71780
13	71780
14	71780
15	71780
16	71780
17	71780

```
= Table.TransformColumnTypes(#"Promoted Headers",{{"SalesOrderID", Int64.Type}, {"SalesOrderDetailID", Int64.Type}
```

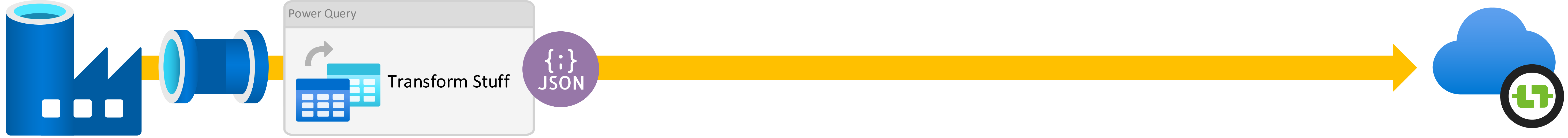
	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPriceD
1	71774	110562	1	836	356.898	
2	71774	110563	1	822	356.898	
3	71776	110567	1	907	63.9	
4	71780	110616	4	905	218.454	
5	71780	110617	2	983	461.694	
6	71780	110618	6	988	112.998	
7	71780	110619	2	748	818.7	
8	71780	110620	1	990	323.994	
9	71780	110621	1	926	149.874	
10	71780	110622	1	743	809.76	
11	71780	110623	4	782	1376.994	
12	71780	110624	2	918	158.43	
13	71780	110625	4	780	1391.994	
14	71780	110626	1	937	48.594	
15	71780	110627	6	867	41.994	
16	71780	110628	1	985	112.998	
17	71780	110629	2	989	323.994	

Name: OrderDetailLines

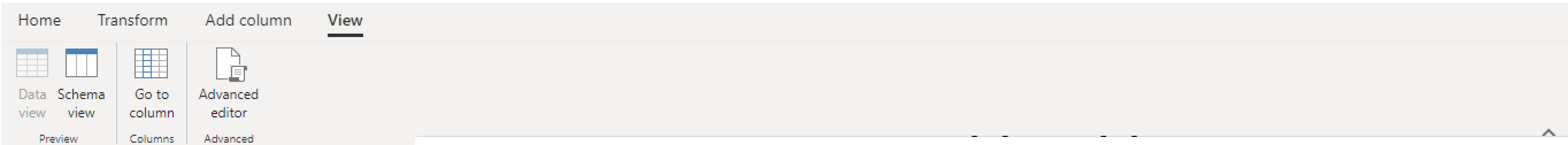
What can a Power Query Activity do?

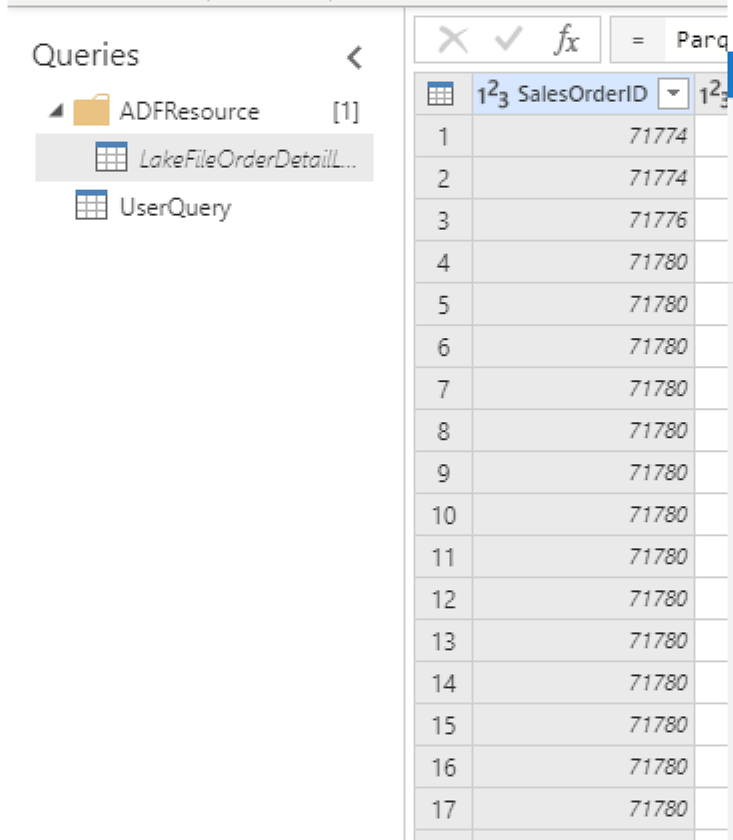
View

Control Flow

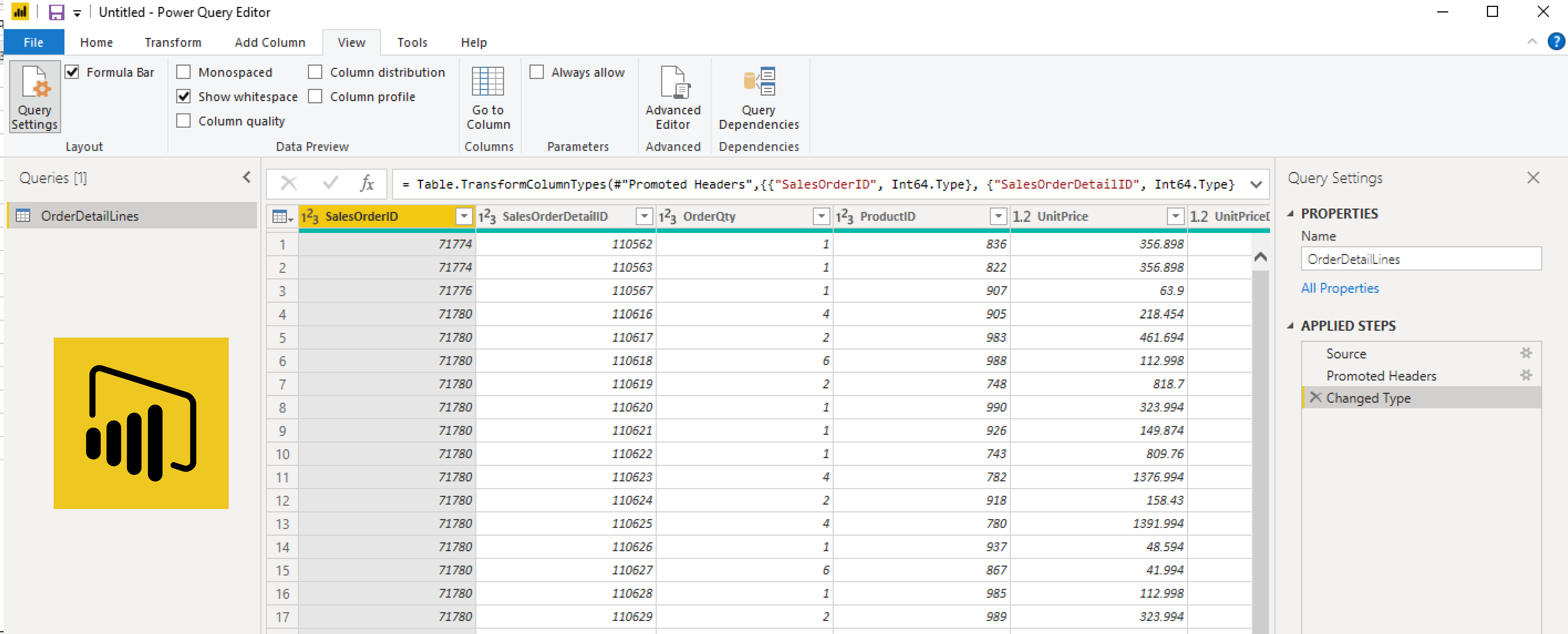


Power Query





	SalesOrderID
1	71774
2	71774
3	71776
4	71780
5	71780
6	71780
7	71780
8	71780
9	71780
10	71780
11	71780
12	71780
13	71780
14	71780
15	71780
16	71780
17	71780



```
= Table.TransformColumnTypes(#"Promoted Headers",{{"SalesOrderID", Int64.Type}, {"SalesOrderDetailID", Int64.Type}
```

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPriceD
1	71774	110562	1	836	356.898	
2	71774	110563	1	822	356.898	
3	71776	110567	1	907	63.9	
4	71780	110616	4	905	218.454	
5	71780	110617	2	983	461.694	
6	71780	110618	6	988	112.998	
7	71780	110619	2	748	818.7	
8	71780	110620	1	990	323.994	
9	71780	110621	1	926	149.874	
10	71780	110622	1	743	809.76	
11	71780	110623	4	782	1376.994	
12	71780	110624	2	918	158.43	
13	71780	110625	4	780	1391.994	
14	71780	110626	1	937	48.594	
15	71780	110627	6	867	41.994	
16	71780	110628	1	985	112.998	
17	71780	110629	2	989	323.994	

PROPERTIES

Name: OrderDetailLines

All Properties

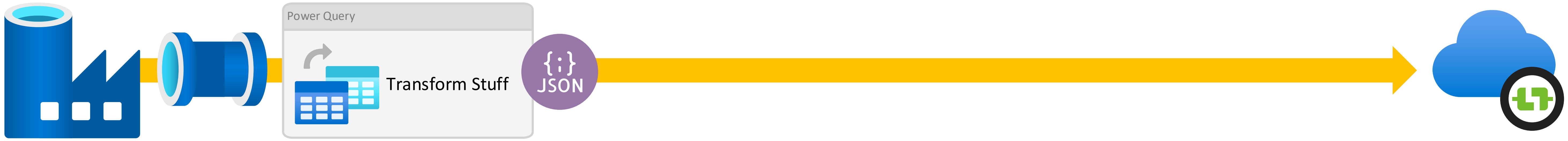
APPLIED STEPS

- Source
- Promoted Headers
- Changed Type

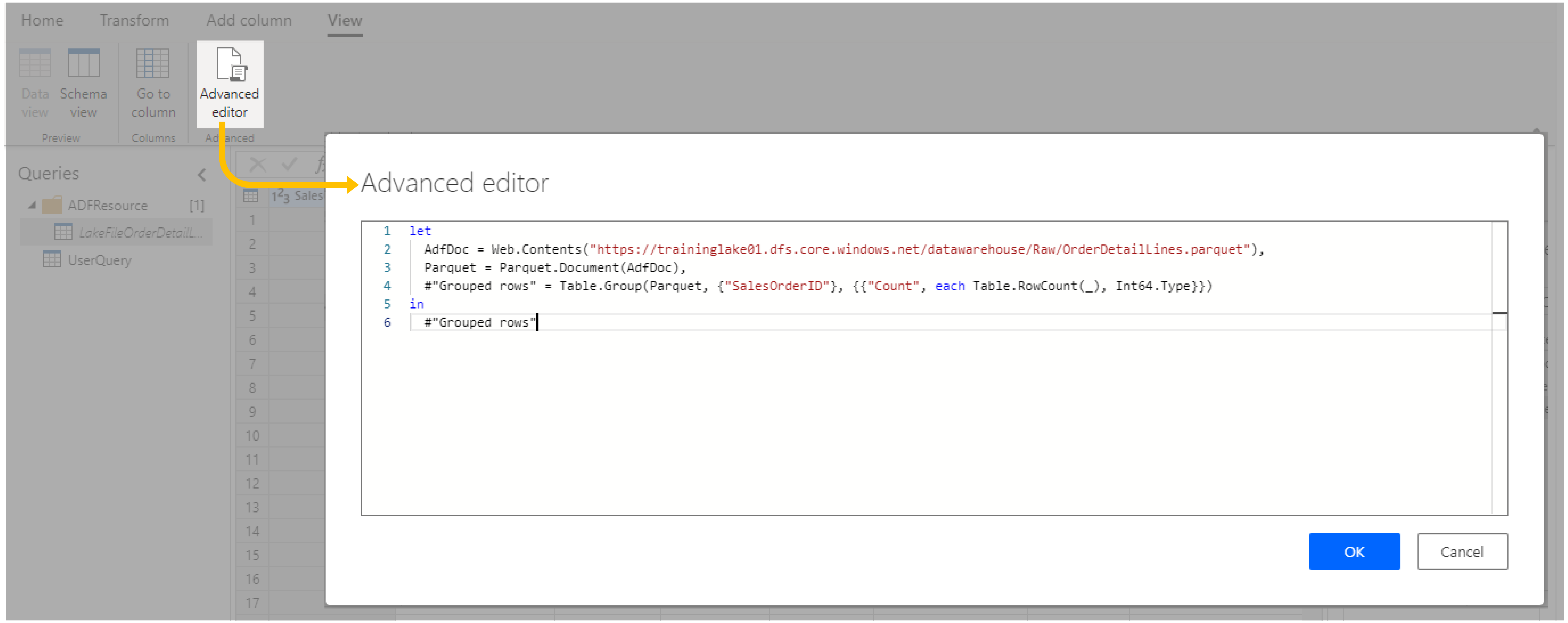
What can a Power Query Activity do?

View

Control Flow



Power Query

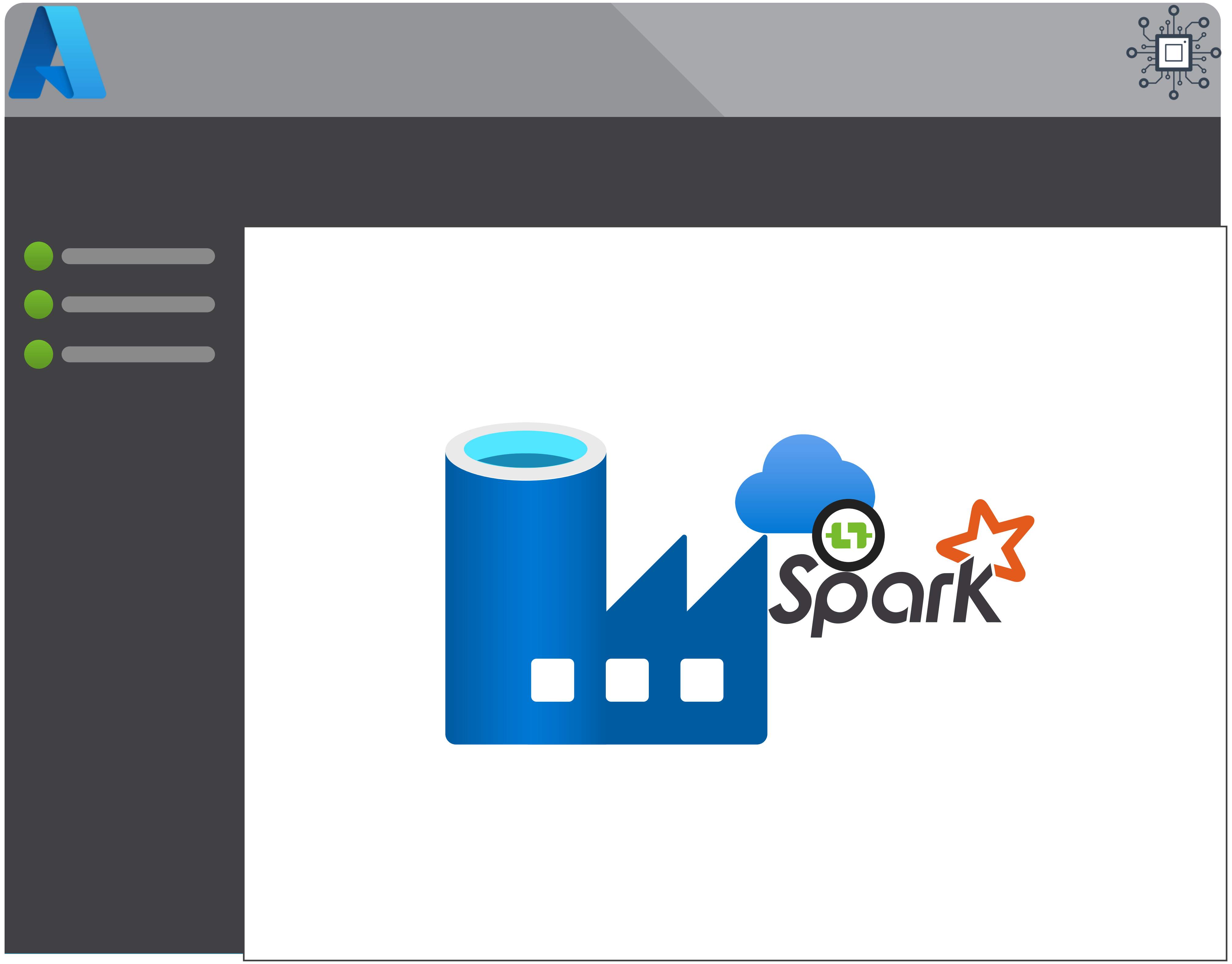


Advanced editor

```
let
  AdfDoc = Web.Contents("https://traininglake01.dfs.core.windows.net/datawarehouse/Raw/OrderDetailLines.parquet"),
  Parquet = Parquet.Document(AdfDoc),
  #"Grouped rows" = Table.Group(Parquet, {"SalesOrderID"}, {{"Count", each Table.RowCount(_), Int64.Type}})
in
  #"Grouped rows"
```

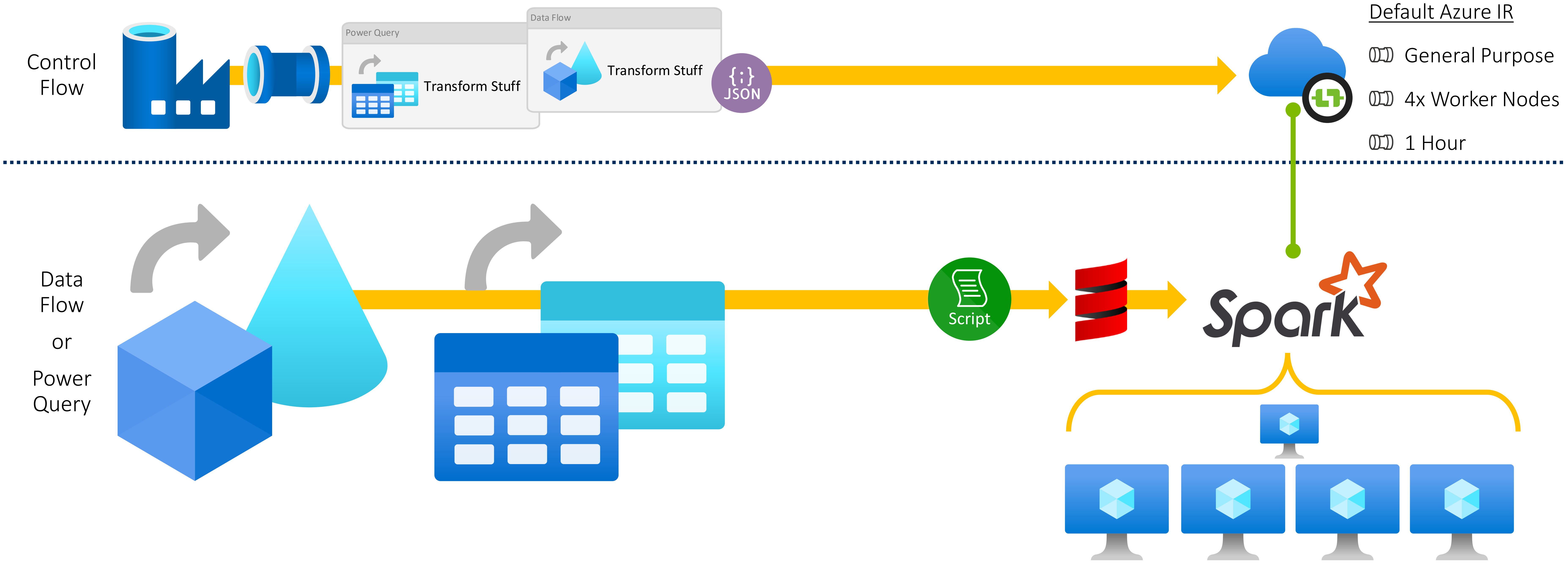
Module 3

Data Transformation



- Data Flows
- Power Query Injection
- Spark Configuration
- Use Cases

Spark Configuration

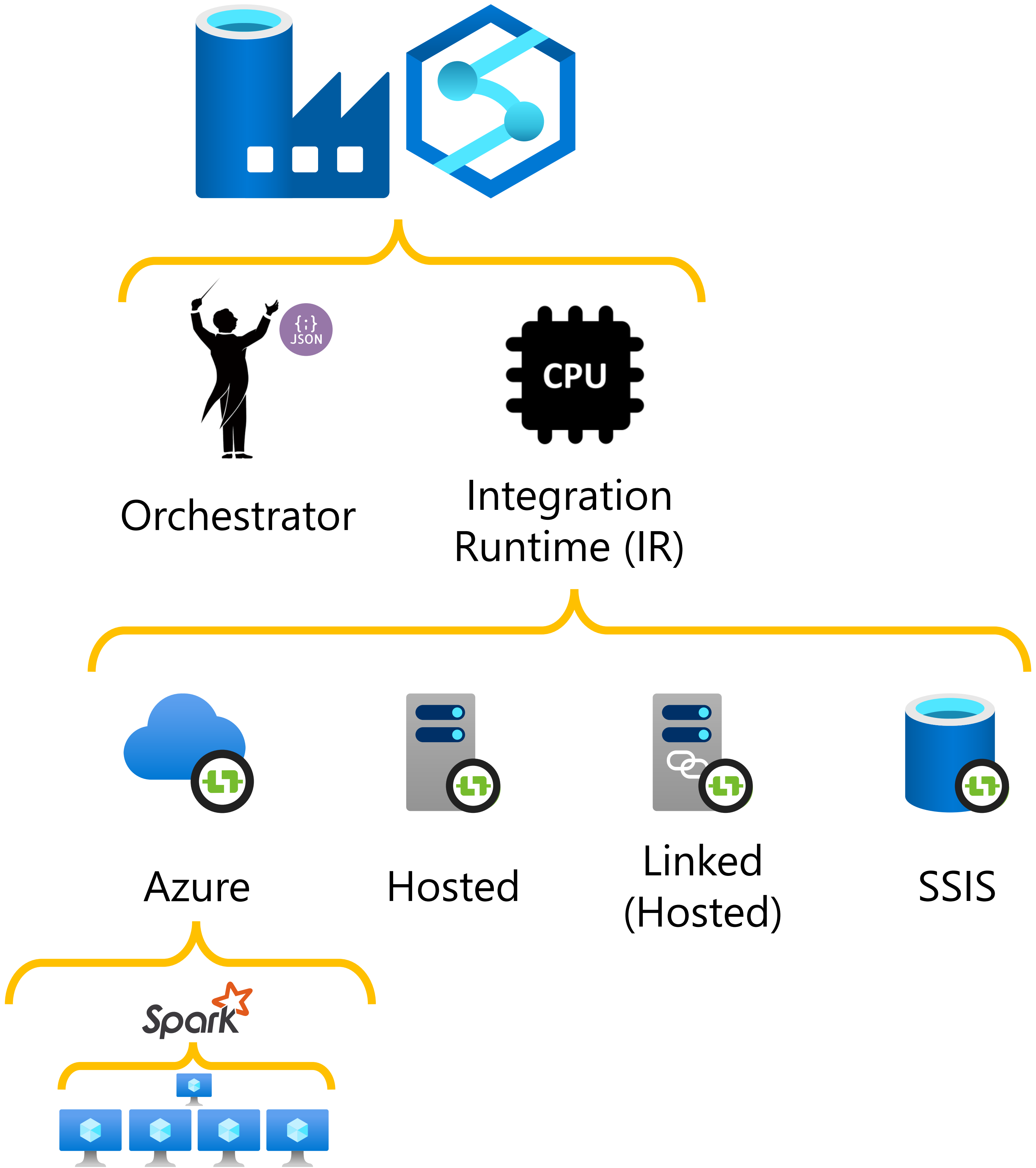


Default Azure IR

- General Purpose
- 4x Worker Nodes
- 1 Hour

- Compute Type
- Number of Worker Nodes
- Cluster Time to Live

Data Flow Compute – IR's vs Spark



Orchestrator

Integration Runtime (IR)

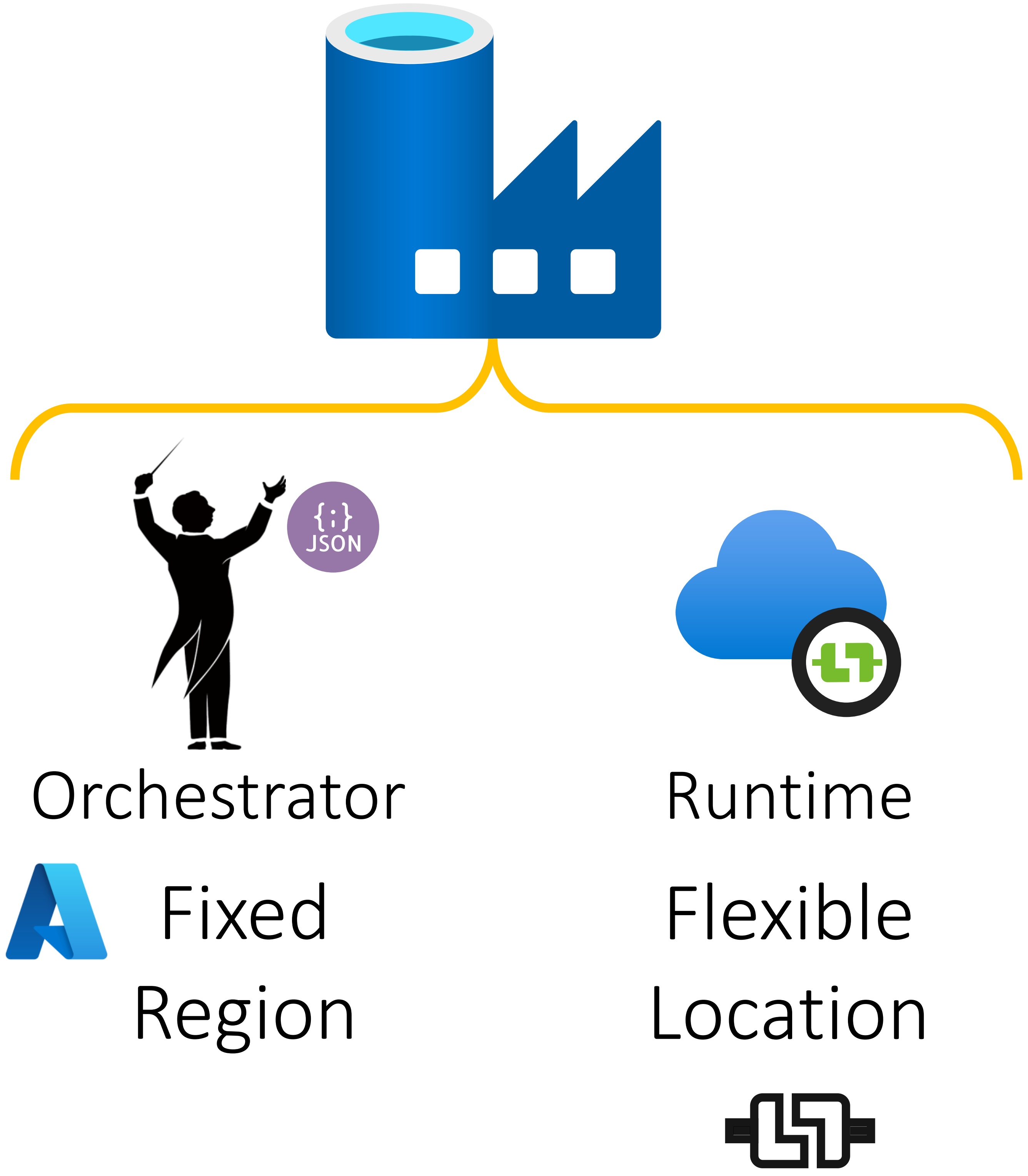
Azure

Hosted

Linked (Hosted)

SSIS

What is an Integration Runtime?



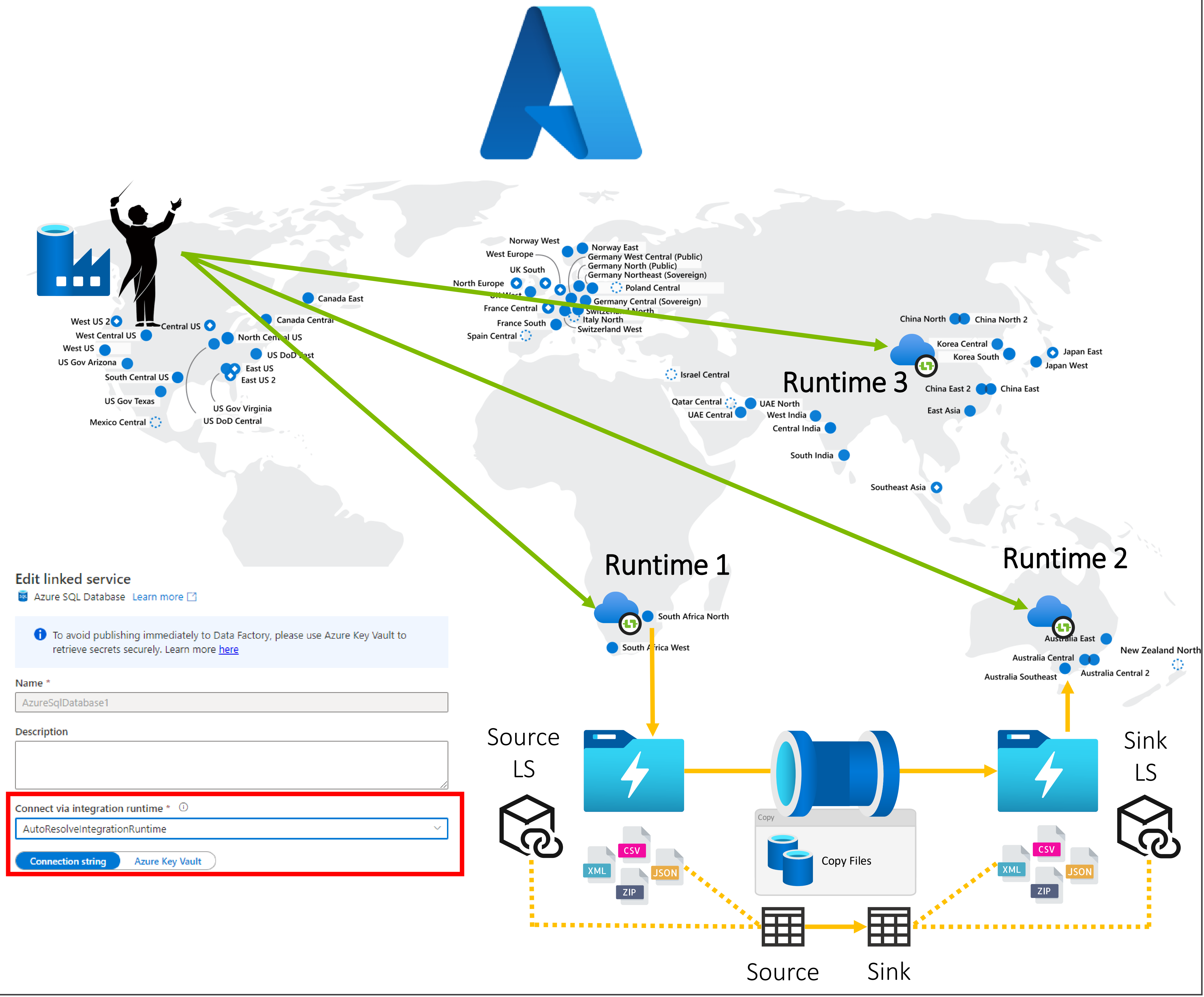
Orchestrator

Fixed Region

Runtime

Flexible Location

AutoResolveIntegrationRuntime



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 *

AzureSqlDatabase1

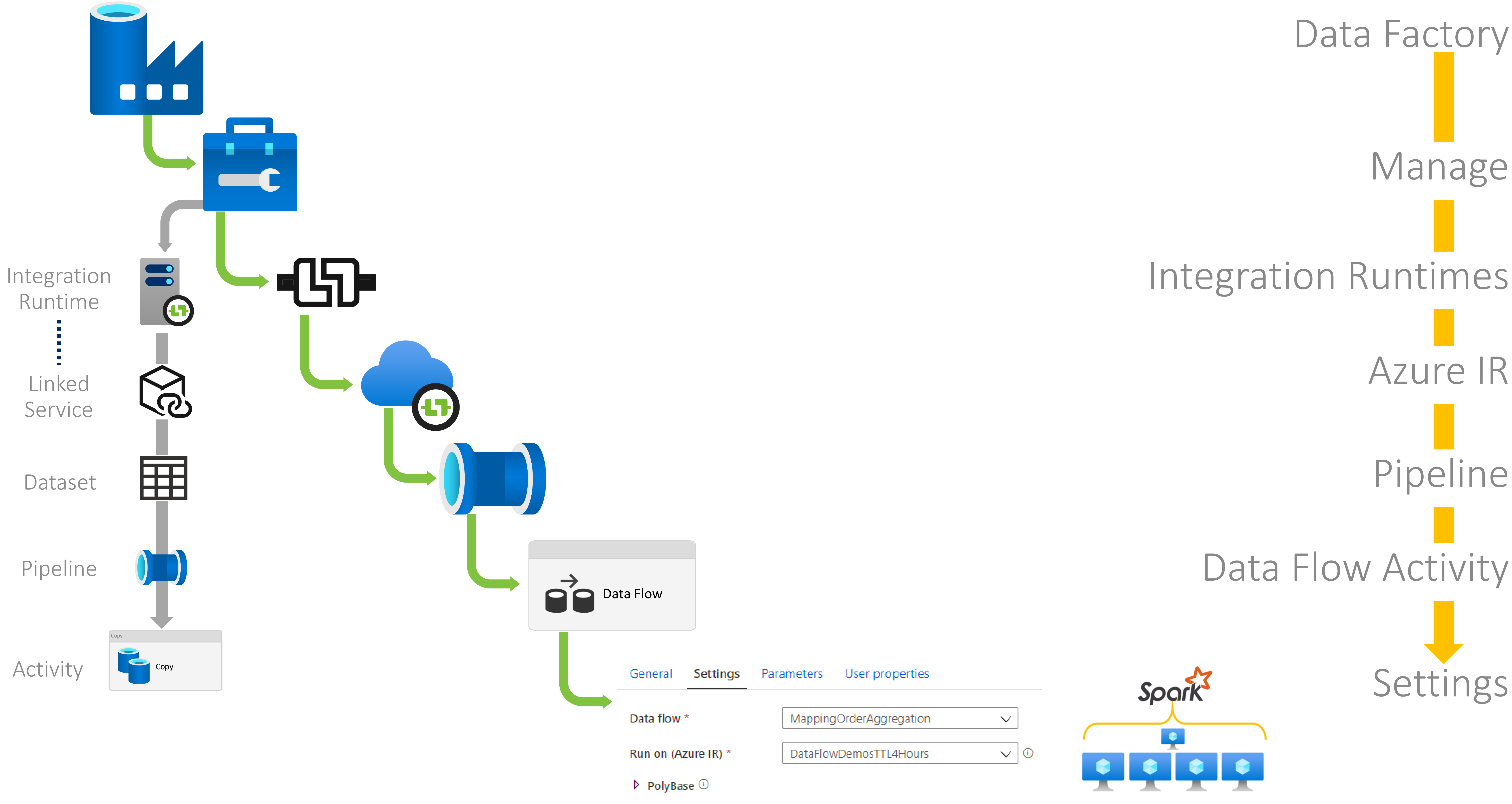
Description

Connect via integration runtime *

AutoResolveIntegrationRuntime

Connection string | Azure Key Vault

Setting the Data Flow Cluster (IR Configuration)



Data Flow * MappingOrderAggregation

Run on (Azure IR) * DataFlowDemosTTL4Hours

PolyBase

Data Factory

Manage

Integration Runtimes

Azure IR

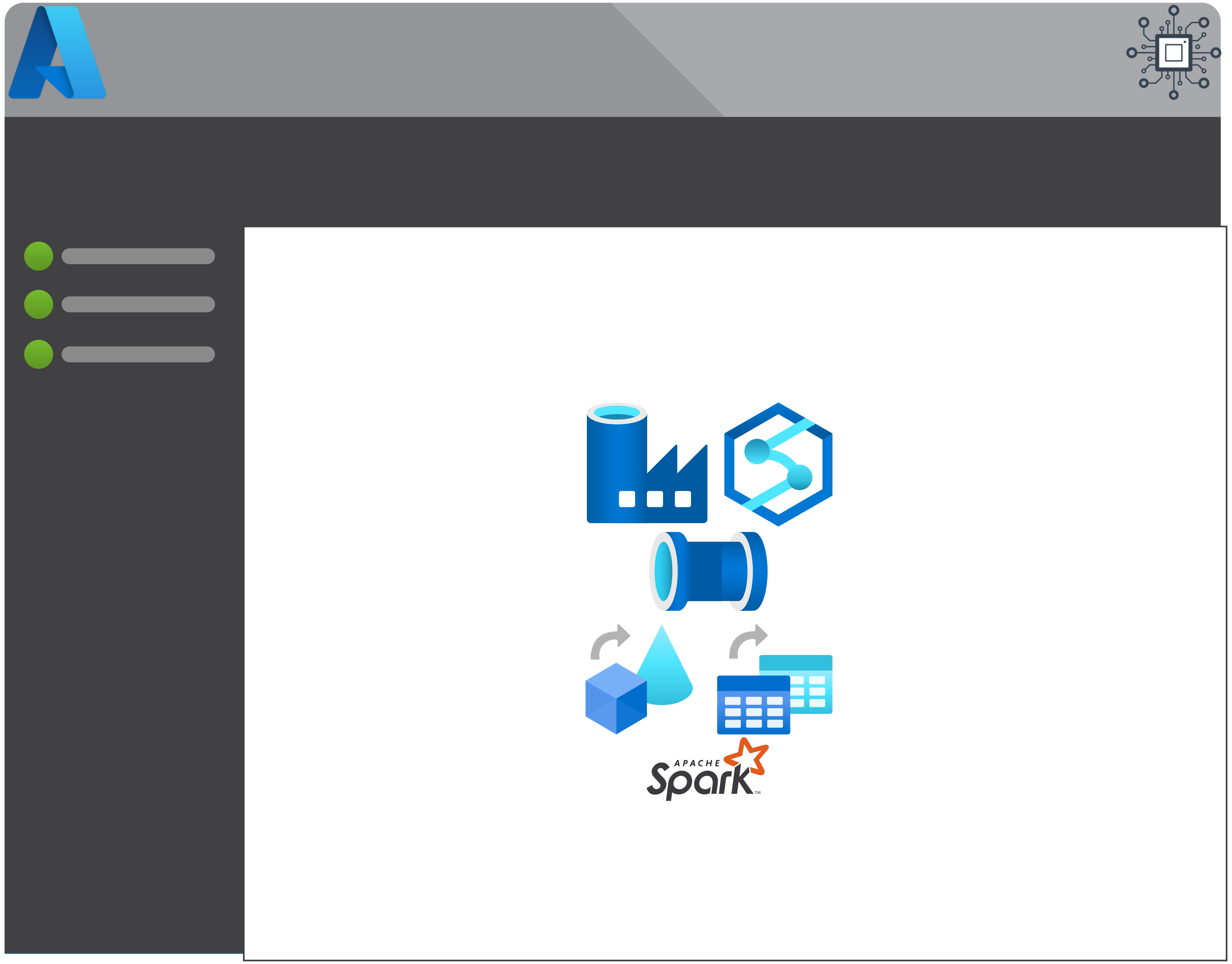
Pipeline

Data Flow Activity

Settings

Module 3

Data Transformation



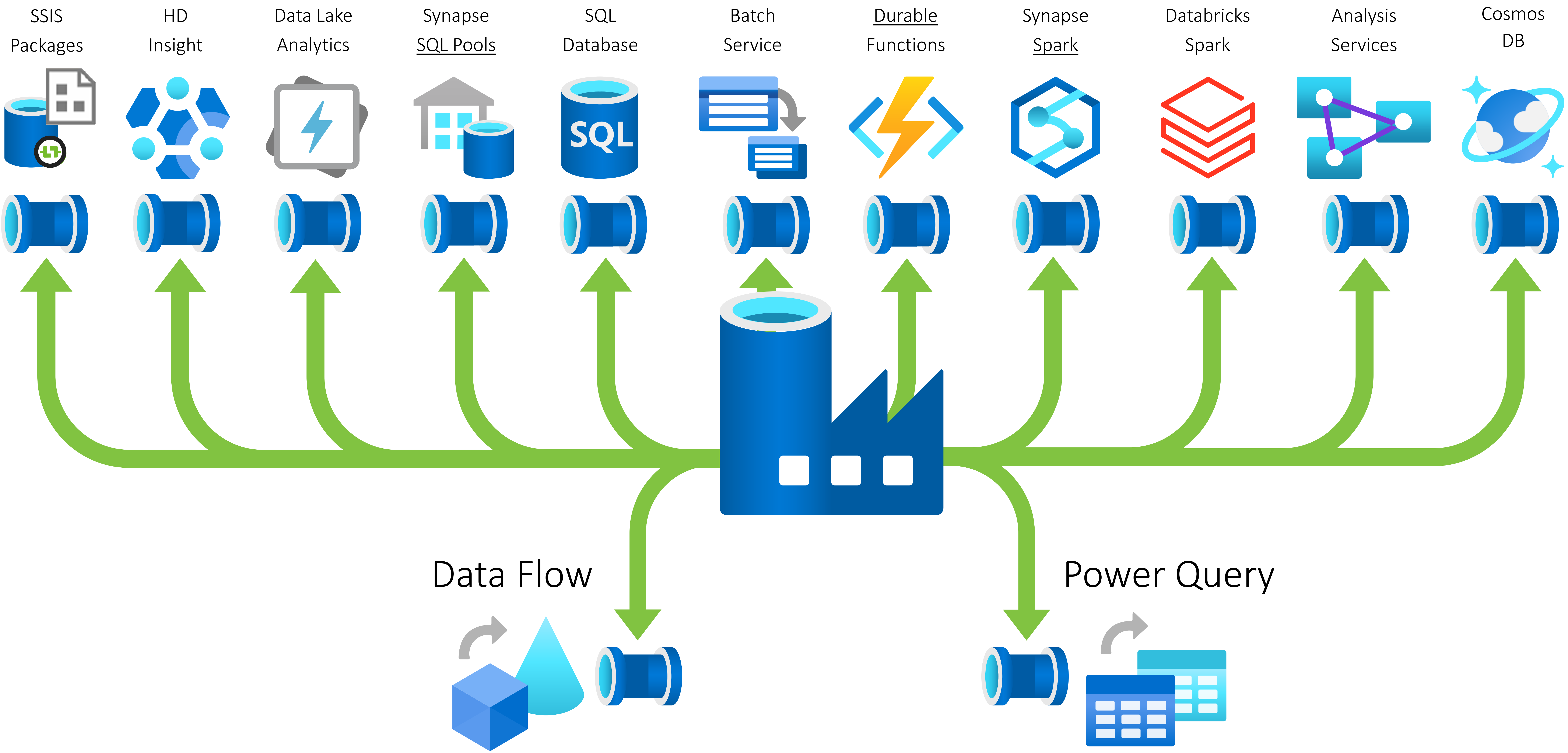
- Data Flows
- Power Query Injection
- Spark Configuration
- Use Cases

Data Transformation Resources in Azure Comparison

Transformation Tools	Graphical UI (Low/No Code)	Scales Out	Scales Up	Cloud Native Tech
T-SQL with SQLDB	✗	✗	✓	✗
SSIS Packages	✓	✗	✓	✗
Scala/Python/SQL with Databricks	✗	✓	✓	✓
Data Flows & Power Query	✓	✓	✓	✓

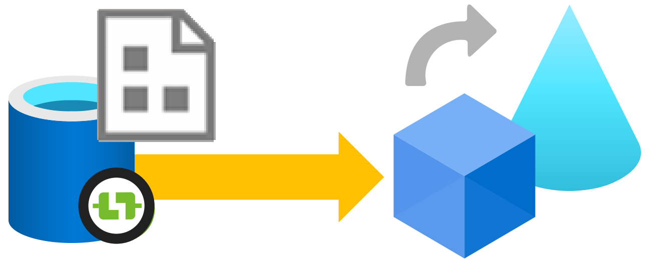
Other Data Transformation Services in Azure

When Should We Use These Integration Pipeline Transformation Activities?

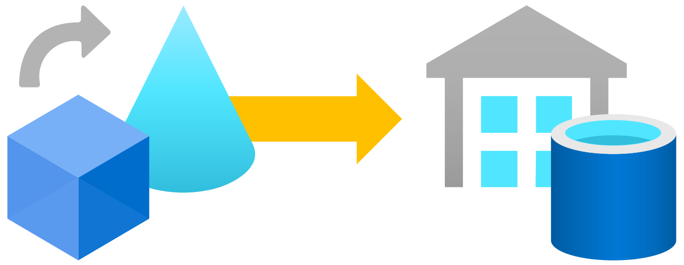


Use Cases

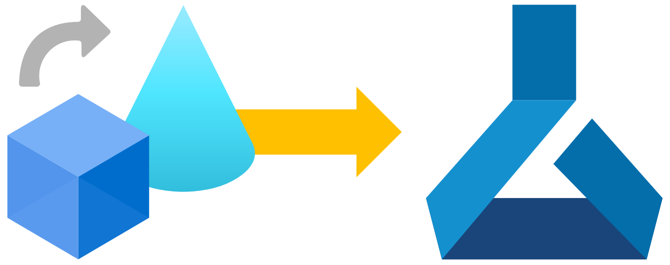
SSIS Package rebuild and skills migration.

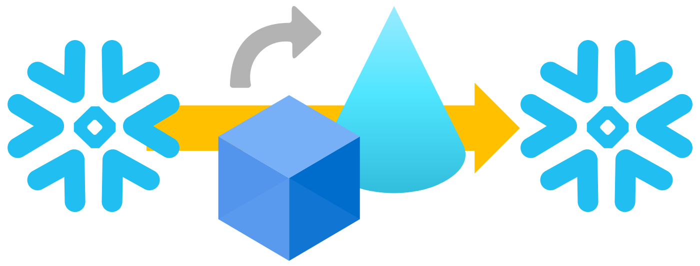


Warehouse data distribution & loading.

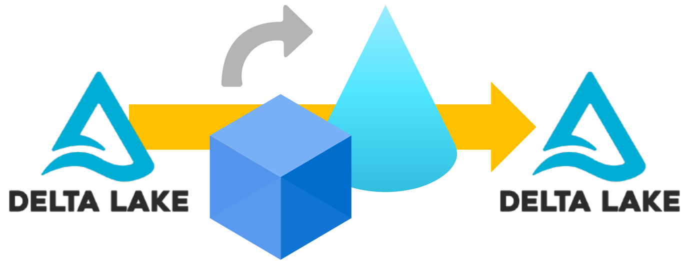


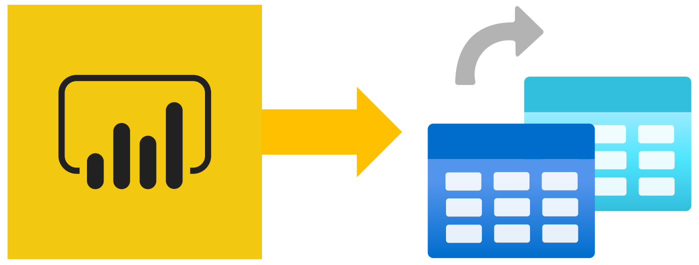
Data model dataset preparation.





Inline dataset transformations.

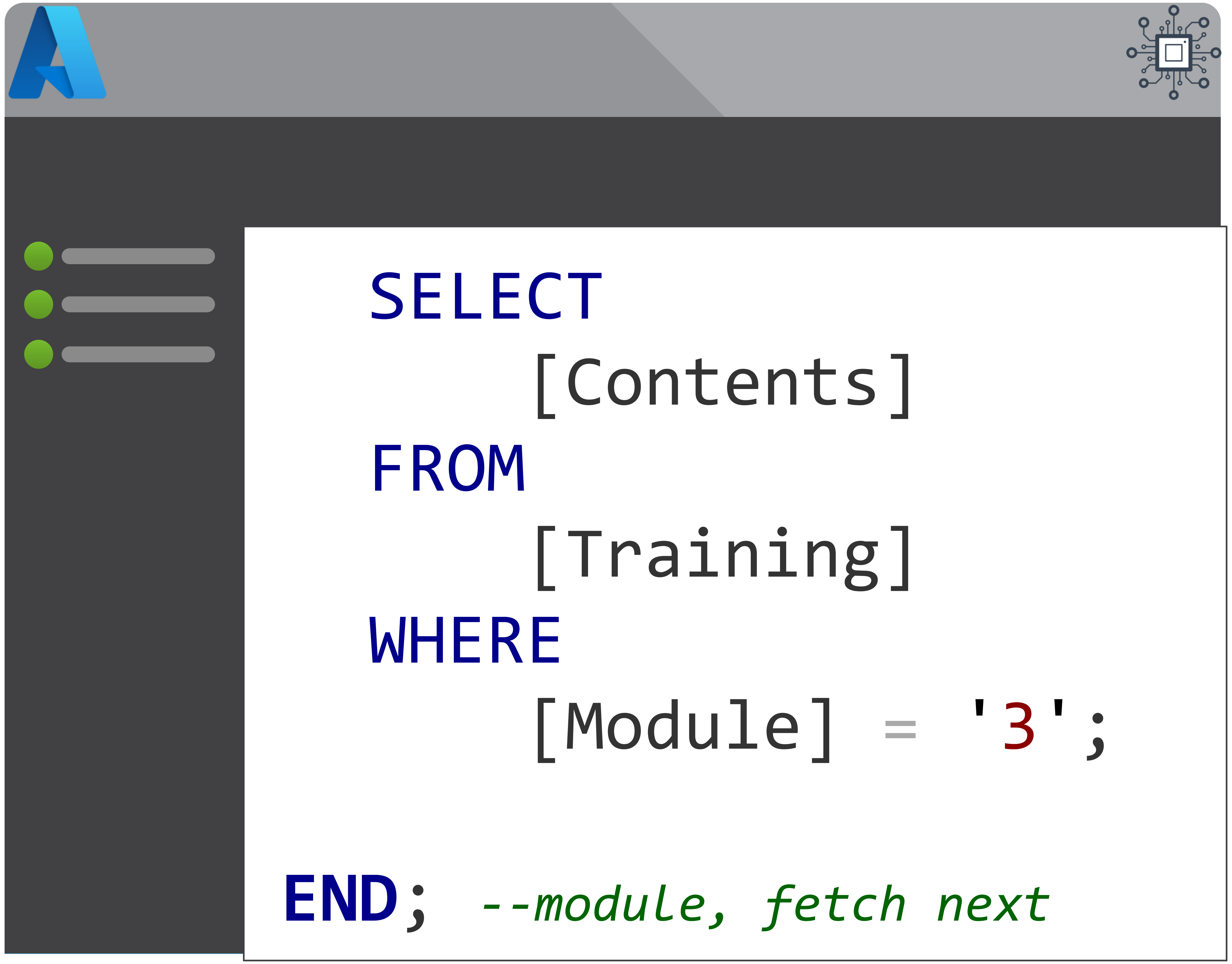




Power Query industrialisation.

Module 3

Data Transformation



```
  SELECT
      [Contents]
  FROM
      [Training]
  WHERE
      [Module] = '3';

END; --module, fetch next
```

- Data Flows
- Power Query Injection
- Spark Configuration
- Use Cases