Azure Data Factory Masterclass

About Me

Alpa Buddhabhatti

/alpaBuddhabhatti/

/alpabuddhabhatti/

/AlpaB7

/@alpabuddhabhatti

/@meetalpa

Microsoft Certified: Azure Data Engineer Associate
Microsoft

Microsoft Certified: Azure Data Scientist Associate
Microsoft

Microsoft Certified: Azure Developer Associate
Microsoft

Microsoft Certified Trainer 2021-2022
Microsoft

Agenda

1. *ADF Dynamic solutions overview*
    I. *Parameters*
    II. *Variables*
    III. *Expressions & Functions*
    IV. *Global Variable*
2. *Labs*
    i. *Activities (Look up, Foreach loop, Get Metadata, Variables, Append Variables)*
    ii. *Delete files for one Business outcomes*
    iii. *Delete files for any Business outcomes*
    iv. *Moving .csv files for one Business outcomes*
    v. *Moving .csv files for any Business Outcomes*

3.Conclusions

Why we need dynamic solutions?

1. Reuse patterns for similar tasks
2. Reduce development time
3. Reduce maintenance cost
4. Lower risk of manual errors

What can make a solution dynamic?

Parameters and Variables

Pass input values and set or update values during runtime

Expressions and Functions:

Modify the content of values during runtime

Loops and Lookups:

Control logic and executions based on external configuration values

Parameters

Use Parameters to pass external value into :

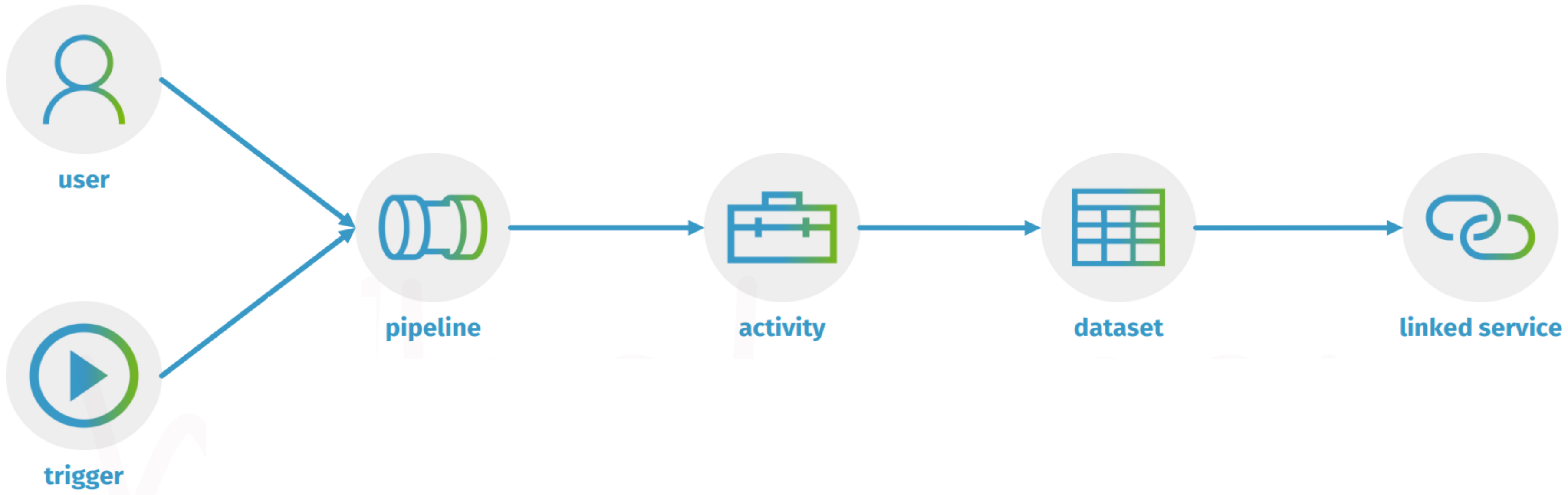
Pipelines

Activity

Datasets

Linked Services

Mapping Data Fl



How to pass parameters?

Pipeline Parameter:

@pipeline().parameters. ParameterName

Dataset Parameter:

@dataset(). ParameterName

Linked Service :

@linkedService().ParameterName

System Parameters :

@pipeline().DataFactory

@pipeline().TriggerTime

Variables

- ➢Variables live inside pipeline
- ➢Variables can be changeable during pipeline execution
- ➢Use for temporary calculations

How you can pass variable to other activity ?

```
@variables('VariableName‘)
@first(variables('VariableName‘)
@last(variables('VariableName‘)
```

Expressions and Function

- Expressions is a “modify values during runtime”.
- @ symbol
- Example :

Passing values as:

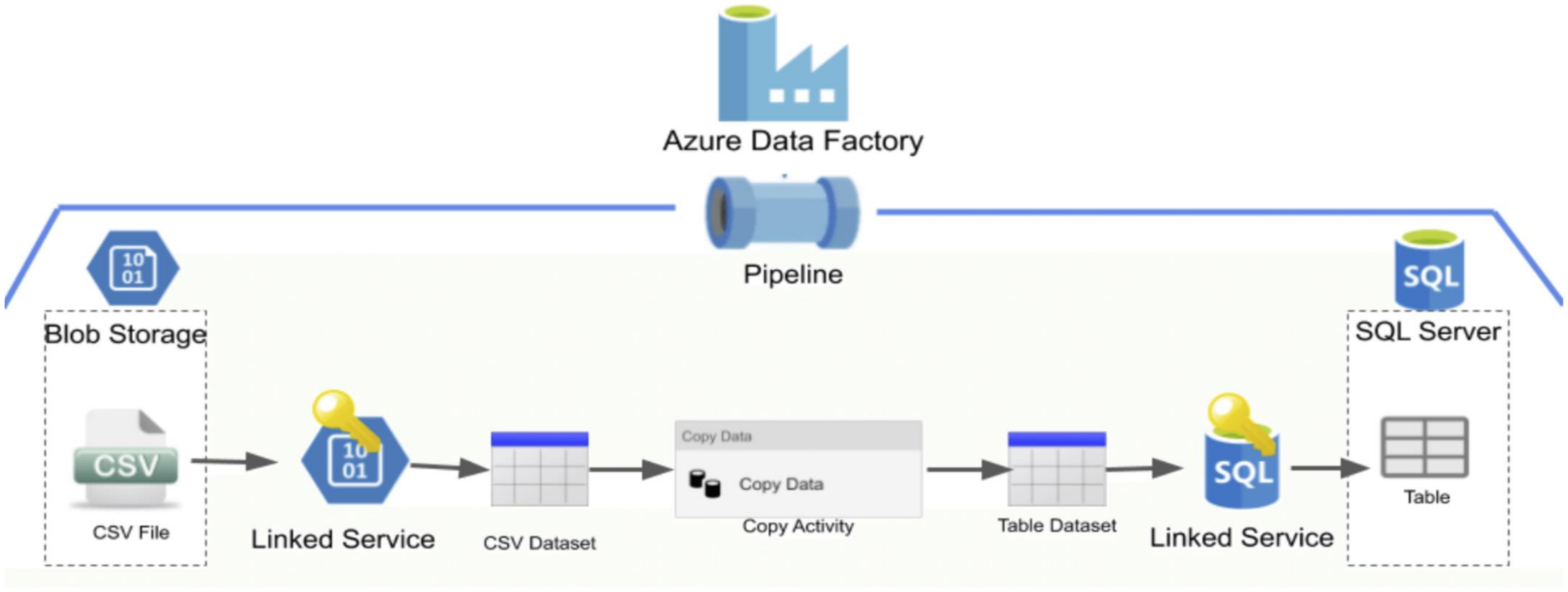
"@concat(pipeline().parameter.Speccode,’-in’)”

Output:

At runtime, expressions are evaluated to literal string values: ‘Feedback-in’

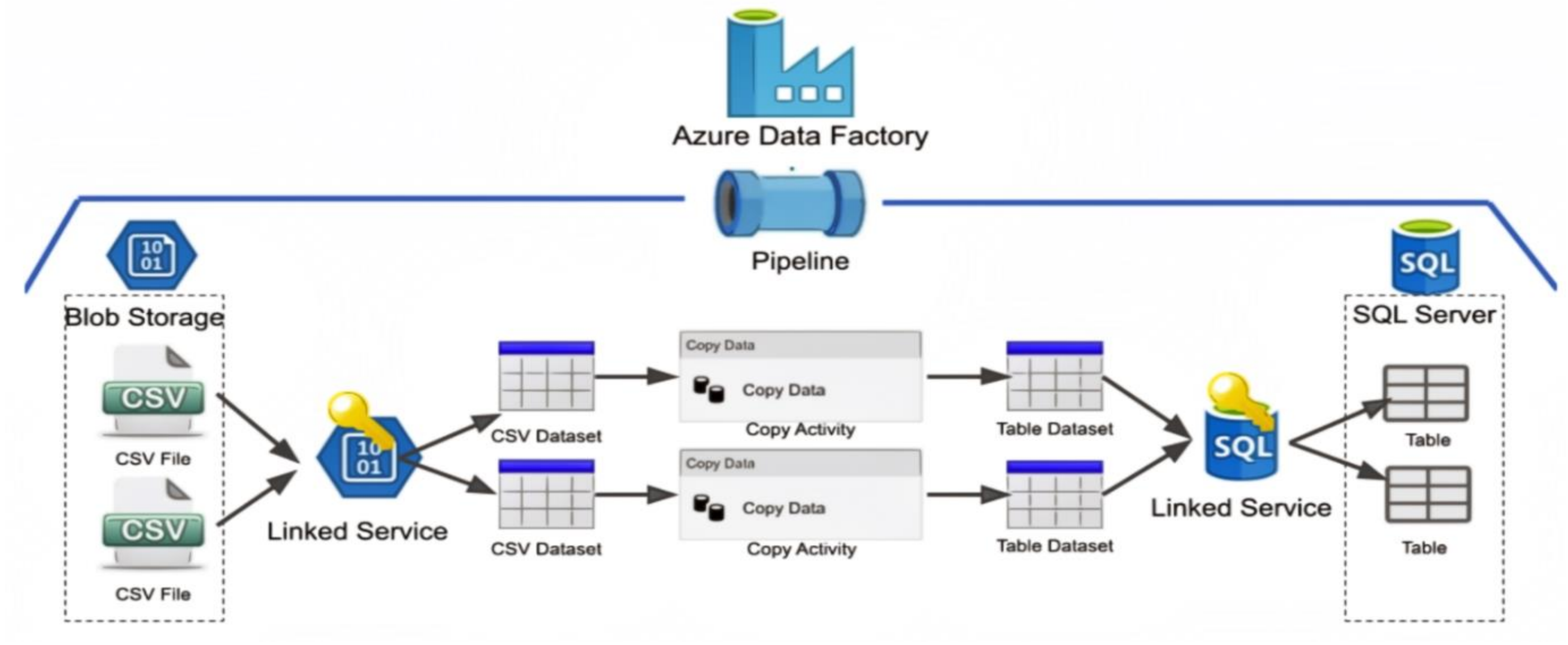
ADF Pipelines - Scenarios

Let's Imagine we have scenarios to move a CSV file(MovieDB.csv) from Azure Blob Storage to Azure SQL Server



ADF Pipelines - Scenarios

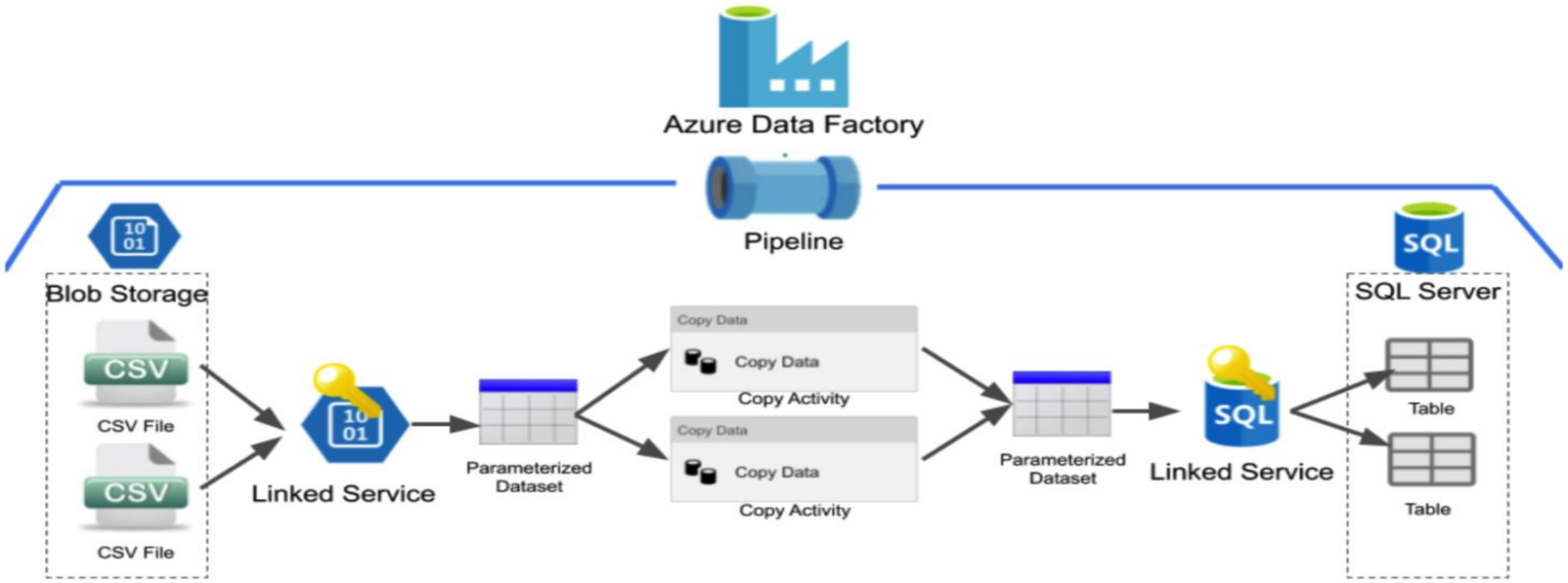
Moving two CSV files from Azure Blob Storage to Azure SQL Server.(Without Parametrization)



ADF Pipelines - Scenarios

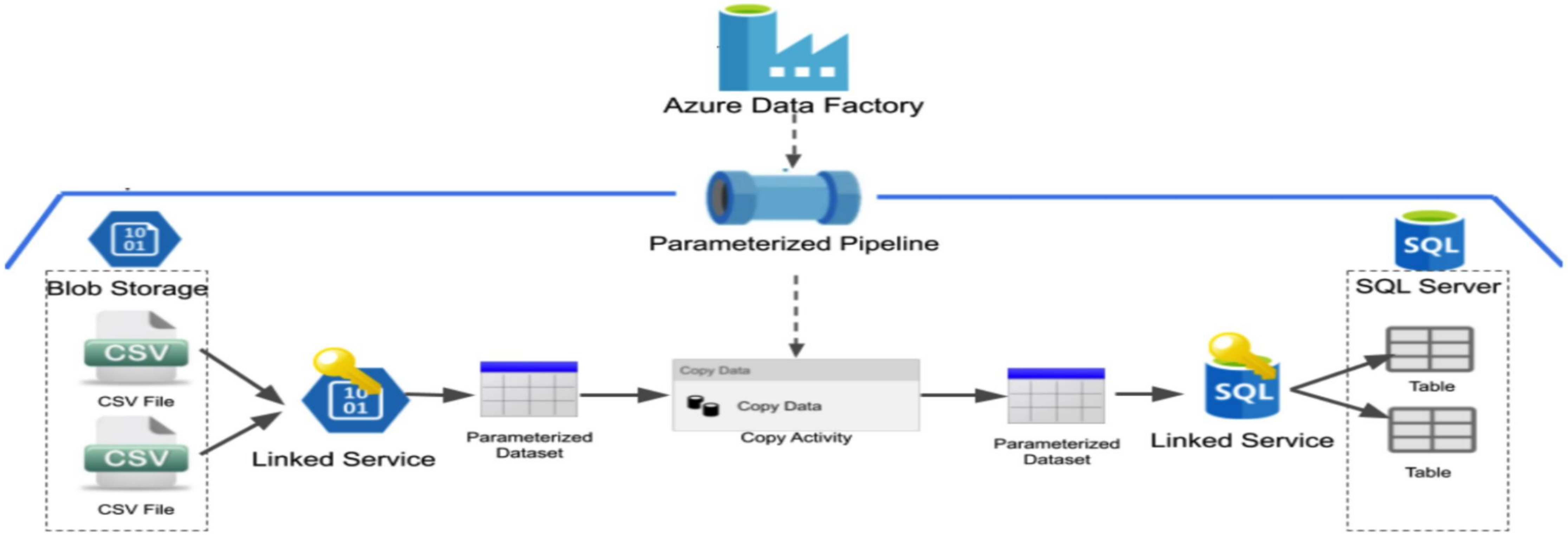
Moving more than one file from Azure Blob Storage to Azure SQL Server.

(with Parametrized Datasets)



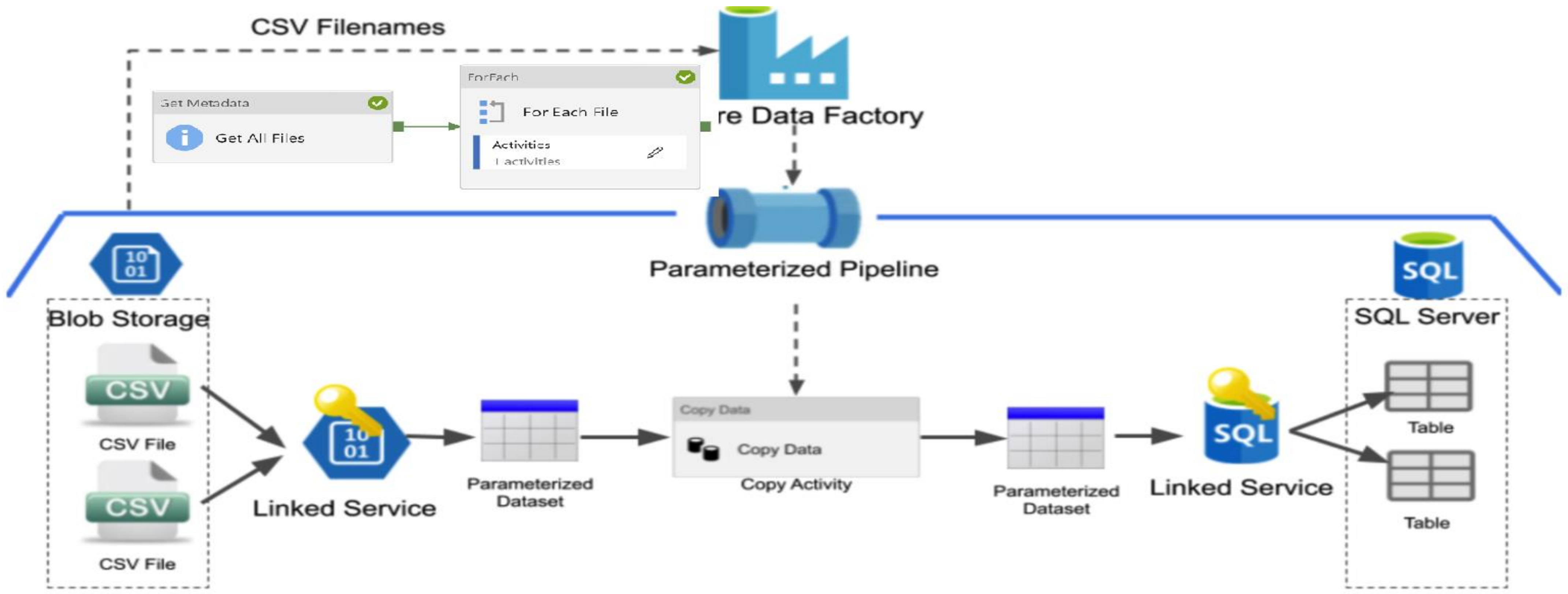
ADF Dynamic solution - 1

Moving more than one file from Azure Blob Storage to Azure SQL Server (With Parametrized Datasets and Pipelines)



ADF Dynamic solution - 2

Moving more than one file from Azure Blob Storage to Azure SQL Server (With Parametrized Datasets and Pipelines and Using loop and Metadata activity)



Steps for Dynamic solution creations

1. Create solution for *one* file or table
2. Replace hardcoded properties with parameters
3. Execute the parameterized solution in a loop
4. Control loop from configuration table

Conclusion

Dynamic Pipeline solutions

- Parameters
- Variables
- Expression and Functions
- Global Parameters

Activities :

- Get Metadata
- Looked up
- Foreach loop
- Variables
- Append Variable

Labs:

- Delete Files from one Business outcomes
- Delete Files from more any Business outcomes
- Moving All .csv files from one Business Outcomes
- Moving All .csv files from any Business outcomes

Look up

```
{
  "firstRow" :
    {
      "Column1" : "Value",
      "Column2" : "Value"
    }
}
```

```
@{activity('Lookup')
.output.firstRow
.Column1}
```

Look up

```
@{activity('Lookup')
.output.value}
```

```
{
  "count" : "2",
  "value" : [
    {
      "Column1" : "Value",
      "Column2" : "Value"
    },
    {
      "Column1" : "Value",
      "Column2" : "Value"
    }
  ]
}
```