Switch , Validation, LookUp, Stored Activity & Data flow :-

Ex:-

Switch Activity in ADF

- Switch Activity provides the same functionality that Switch statement provides in Programming languages.
- It evaluates a set of activities corresponding to a case that matches the condition evaluation.

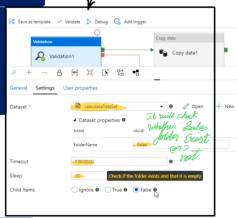
Default
Case - 1
Case - 2

requirement :

Pipeline should take file from source location and copy it to output1/Output2 folders based on Pipeline Parameter value

Validation Activity in ADF

- You can use a Validation in a pipeline to ensure the pipeline only continues execution once it has validated the attached dataset reference exists



Lookup Activity in ADF

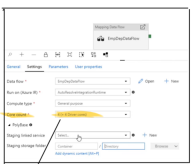
- Lookup activity can retrieve a dataset from any of the Azure Data Factory-supported data sources.
- Lookup activity reads and returns the content of a configuration file or table. It also returns the result of executing a query or stored procedure.
- The output from Lookup activity can be used in a subsequent activity

Stored Procedure Activity in ADF

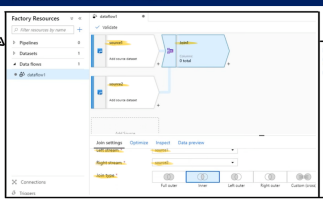
- Stored Procedure Activity is one of the transformation activities that Data Factory supports. We run stored procedure as one of the step using this Activity.

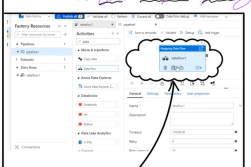
Data Flows in ADF

- Data flows feature in Azure data factory will allow you to develop graphical data transformation logic that can be executed as activities in ADF pipelines
- Your Data flow will execute on your own Azure data bricks cluster for scaled out data processing using spark.
- ADF internally handles all the code translation, spark optimization and execution of transformation.



Core Count is nothing but the no. of computer processing





* Create a Data flow in the DF Tab and Execute that in the pipe line tab

Types of Data flow :-

1. Mapping data flow
2. Wrangling data flow

Right click on the sink and select Edit Sink

Setting is used to Create output folder name

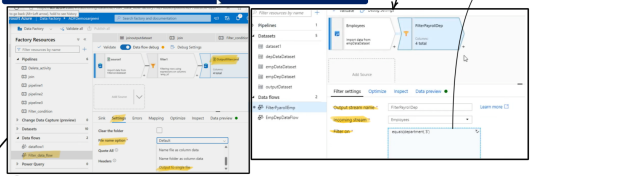
Map to output Setting what are the fields needed in the output

Filter / Aggregate / Conditional Split / Derived Column / Exists :-

Filter Transformation

- The Filter transforms allows row filtering based upon a condition. The output stream includes all rows that matching the filtering condition. The filter transformation is similar to a WHERE clause in SQL.

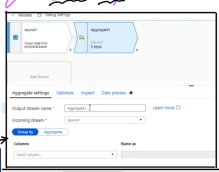
Enter the filter condition



While Executing the pipeline we get more like i.e. Source, Inp... etc. go to Setting of Sink tab and Select → Output to Single

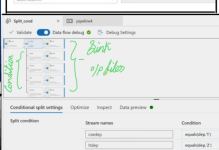
Aggregate Transformation

- The Aggregate transformation defines aggregations of columns in your data streams. Using the Expression Builder, you can define different types of aggregations such as SUM, MIN, MAX, and COUNT grouped by existing or computed columns.



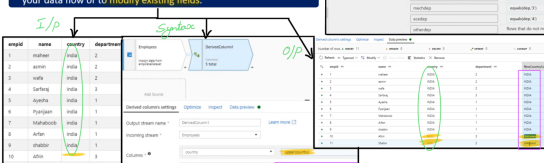
Conditional Split Transformation

- The conditional split transformation routes data rows to different streams based on matching conditions. The conditional split transformation is similar to a CASE decision structure in a programming language.



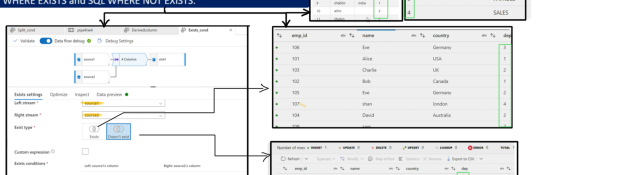
Derived Column Transformation

- Use the derived column transformation to generate new columns in your data flow or to modify existing fields.



Exists Transformation in Data Flow

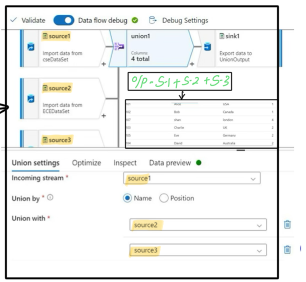
- The exists transformation is a row filtering transformation that checks whether your data exists in another source or stream. The output stream includes all rows in the left stream that either exist or don't exist in the right stream. The exists transformation is similar to SQL WHERE EXISTS and SQL WHERE NOT EXISTS.



Union / Lookup / Sort Transformation :-

Union Transformation in Data Flow

- Union will combine multiple data streams into one, with the SQL Union of those streams as the new output from the Union transformation.
- You can combine n-number of streams in the settings table by selecting the "+" icon next to each configured row, including both source data as well as streams from existing transformations in your data flow.



Source - 1

emp_id	name	country	dep
101	Alice	USA	1
102	Bob	Canada	1

Source - 2

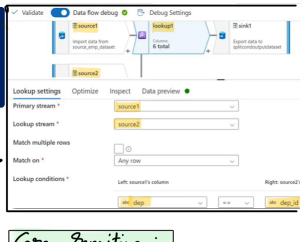
emp_id	name	country	dep
107	shan	london	4

Source - 3

emp_id	name	country	da
103	Charlie	UK	2
105	Eve	Germany	2
104	David	Australia	2

Lookup Transformation in Data Flow

- A lookup transformation is similar to a left outer join. All rows from the primary stream will exist in the output stream with additional columns from the lookup stream.



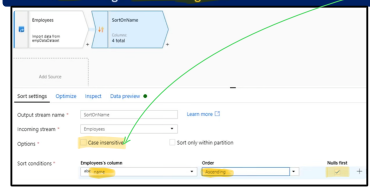
* based on dep and dep_id Column

Sort Transformation in Data Flow

- The sort transformation allows you to sort the incoming rows on the current data stream. You can choose individual columns and sort them in ascending or descending order.

Case Sensitive :-

OFF ON



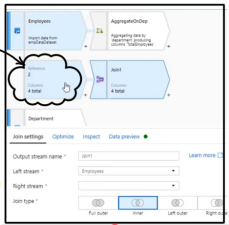
name
Afrin
Arfan
Ayesha
Imam
Mahaboob
Pyarjaan
Sarfaraj
Shahin
Shajarun
asmin
maheer

name
Afrin
Arfan
asmin
Ayesha
Imam
Mahaboob
maheer
Pyarjaan
Sarfaraj
shabbir
Shahin

New Branch / Select / Window / Alter Row :-

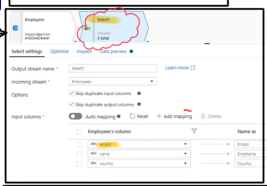
New Branch in Data Flow

- Add a new branch to do multiple sets of operations and transformations against the same data stream. Adding a new branch is useful when you want to use the same source to for multiple sinks.



Select Transformation in Data Flow

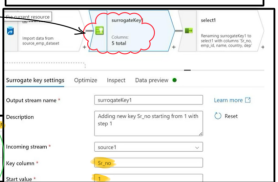
- Use the select transformation to rename, drop, or reorder columns. This transformation doesn't alter row data, but chooses which columns are propagated downstream.



* Simple Select Statement
i.e, SELECT * COLUMN_1

Surrogate Key Transformation in Data Flow

- Use the surrogate key transformation to add an incrementing key value to each row of data.



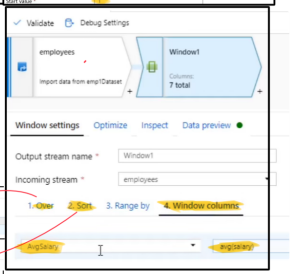
* It will Create a new row no

o/p →

emp_id	name	country	dep	Sr_no
106	Eve	Germany	3	1
101	Alice	USA	1	2
103	Charlie	UK	2	3
102	Bob	Canada	1	4

Window Transformation in Data Flow

- Window transformation is where you will define window-based aggregations of columns in your data streams. In the Expression Builder, you can define different types of aggregations that are based on data or time windows (SQL OVER clause) such as LEAD, LAG, NTILE, CUMEDIST, RANK, etc.). A new field will be generated in your output that includes these aggregations. You can also include optional group-by fields.



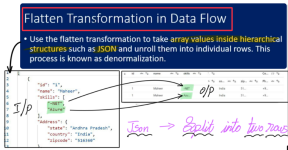
Alter Row Transformation in Data Flow

- Use the Alter Row transformation to set insert, delete, update, and upsert policies on rows. You can add one-to-many conditions as expressions. These conditions should be specified in order of priority, as each row will be marked with the policy corresponding to the first-matching expression. Each of those conditions can result in a row (or rows) being inserted, updated, deleted, or upserted.
- Alter Row transformations will only operate on database or CosmosDB sinks in your data flow.

Flatten / parameterize / validate Schema / Schema drift / wrangling

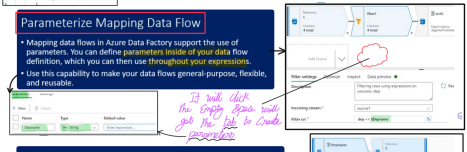
Flatten Transformation in Data Flow

- Use the flatten transformation to take array values inside hierarchical structures such as JSON and unroll them into individual rows. This process is known as denormalization.



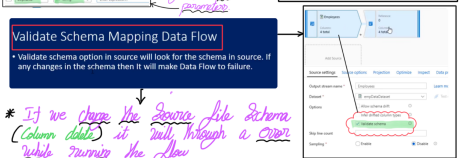
Parameterize Mapping Data Flow

- Mapping data flows in Azure Data Factory support the use of parameters. You can define parameters inside of your data flow definition, which you can then use throughout your expressions.
- Use this capability to make your data flows general-purpose, flexible, and reusable.



Validate Schema Mapping Data Flow

- Validate schema option in source will look for the schema in source. If any changes in the schema then It will make Data Flow to failure.



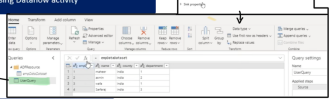
Schema Drift Mapping Data Flow

- Schema drift is the case where your sources often change metadata.
- Fields, columns, and, types can be added, removed, or changed on the fly.
- Without handling for schema drift, your data flow becomes vulnerable to upstream data source changes.

Feature	One-line Explanation	One-line Example
Allow Schema Drift	Automatically handles changes in data structure, like added or missing columns.	A new "Email" column is added, and ADF processes it without needing manual changes.
Infer Drifted Column Type	Automatically guesses the data type of new columns based on values.	A new "Salary" column appears with numbers, and ADF infers it as an integer.
Validate Schema	Ensures the incoming data matches the expected structure before processing.	If the "Age" column is missing, ADF stops and gives an error message.

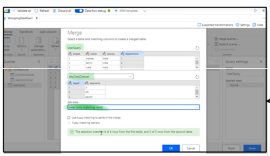
Wrangling Data Flow

- Wrangling data flows in Azure Data Factory allow you to do code-free data preparation at cloud scale.
- It uses industry leading power query technology that we see in excel.
- Data preparation logic defined in Wrangling data flows will gets translate in to spark and run on Bigdata clusters
- We can use this Wrangling data flows as one of the step inside your azure data factory pipelines using Dataflow activity



Merge Queries Wrangling Data Flow

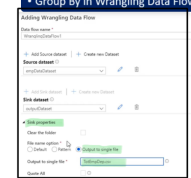
- Merge Queries in Wrangling Data Flows are like JOIN in T-SQL

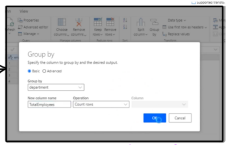


Merge Queries wrangling DF / Diff Author mode / Github / Annotation / template

Merge Queries Wrangling Data Flow

- Group By in Wrangling Data Flows is same as GROUP BY in T-SQL





If we Select the output to Single file option then only will get Single file. If no will get file for Each row. i.e, 5 rows of data → 5 Separate files.

Different Author Modes in Azure Data Factory

- By default, the Azure Data Factory authors directly against the data factory service. This will have below limitations
  1. ADF service doesn't include a code repository. Hence ADF components will be directly published to Data factory service and we cannot perform Partial saves and we cannot save JSON code for ADF and its components.
  2. ADF service is not optimized for collaboration and version control.
- Hence, to provide better Authoring Experience ADF allows you to configure a git repository with either Azure Repos or GitHub

```
Repositories -> parent Folder
branches --> Sub folders
        1

Collabartion branch --> final code of application

master(collabartion branch)

Features Branch:
employeeA(newcode -> IMplimentaion)
employeeB(code -> IMplimentations)

Pull Request:
A created pull request -> Approval(manager)
```

Version Control:
Ability to track/audit changes.
Ability to revert changes that introduced bugs.

Partial Saves:
When authoring against the data factory service, you can't save changes as a draft and all publishes must pass data factory validation

Collaboration and control:
If you have multiple team members contributing to the same factory, you may want to let your teammates collaborate with each other via a code review process.

Better CI\CD:

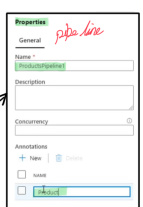
Better Performance:
An average factory with git integration loads 10 times faster than one authoring against the data factory service.

Advantages of Git Integration

- Version Control
- Partial saves
- Collaboration and control
- Better CI/CD
- Better Performance

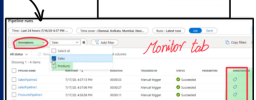
Setup GitHub Code Repo for ADF

- We can Author Azure Data Factory either with GitHub repository or with Azure DevOps Git Repos.
- In this part we are going to look at Authoring Azure data factory with GitHub repository
- A single GitHub account can have multiple repositories, but a GitHub repository can be associated with only one data factory.
- The GitHub integration with Data Factory supports both public GitHub (that is, https://github.com) and GitHub Enterprise.



Annotations/Tags

- Annotations are like Tags. We can Tag our pipelines with some name and then we can use that name to filter while monitoring.
- We can apply Annotations on below ADF components
  - Pipelines
  - Linked Services
  - Triggers



Templates

- Templates are predefined Azure Data Factory pipelines that allow you to get started quickly with Data Factory.
- Templates will reduce development time to build pipelines

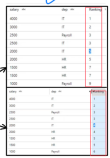
Global Parameters

- Global parameters are constants across a data factory that can be consumed by a pipeline in any expression.
- They're useful when you have multiple pipelines with identical parameter names and values.

Rank by / Session log in Copy / parse by / Fail Activity :

Rank Transformation in Data Flow

- Use the rank transformation to generate an ordered ranking based upon sort conditions specified by the user.



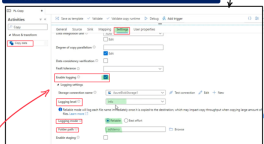
Session log in Copy Activity

- You can log your copied file names in copy activity, which can help you to further ensure the data is not only successfully copied from source to destination store, but also consistent between source and destination store by reviewing the copied files in copy activity session logs.
- When you enable fault tolerance setting in copy activity to skip faulty data, the skipped files and skipped rows can also be logged. This we will discuss more in another video.

Session log in Copy Activity

- After the copy activity runs completely, you can see the path of log files from the output of each copy activity run.

https://{your-blob-account}.blob.core.windows.net/{logFilePath}/copyactivity-logs/{copy-activity-name}/{copy-activity-run-id}/{auto-generated-id}.csv



Example :-

I/p

Source - 1
Source - 2
Source - 3

Copy Data Activity

O/p

Source - 1
Source - 2
Source - 3

Including the Session log file whether copied correctly or not

Schema of log file

Column	Description
Timestamp	The timestamp when ADF reads, writes, or skips the object.
Level	The log level of this item. It can be 'Warning' or 'Info'.
OperationName	ADF copy activity operational behavior on each object. It can be 'FileRead', 'FileWrite', 'FileSkip', or 'SkipMoreFiles'.
OperationItem	The file names or skipped rows.
Message	More information to show if the file has been read from source store, or written to the destination store. It can also be why the file or rows has being skipped.

Parse Transformation

- Use the Parse transformation to parse columns in your data that are in document form. The current supported types of embedded documents that can be parsed are JSON, XML, and delimited text.

Fail Activity

- You might occasionally want to throw an error in a pipeline intentionally. In such cases we can make use of fail activity to make our pipeline fail intentionally.

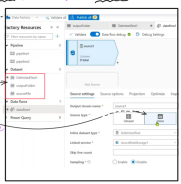
* If the Copy activity fails, Fail will be triggered. Whereas the web activity and Session will succeed the pipeline is successfully completed.

* But logically the pipeline is failed. So we are using Fail activity to throw an error in Monitor tab the pipeline is failed due to this Error.

IN Line Data Set

Two types of Data Sets :-

1. Data Set
2. In Line



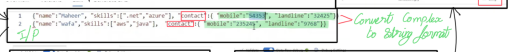
* In line Data Set are not reusable will not able to use in different Data flows.

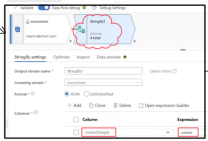
* It will not Create a Data Set under the Data Set tab

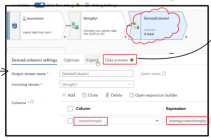
Stringify / Assert tf / user defined function / Fuzzy join :-

Stringify transformation

- Use the stringify transformation to turn complex data types into strings. This can be very useful when you need to store or send column data as a single string entity that may originate as a structure, map, or array type.



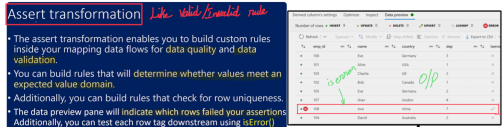


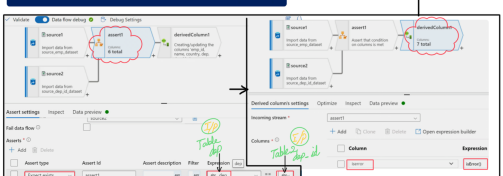


Assert transformation

Like Valid/Invalid rule

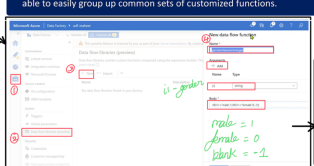
- The assert transformation enables you to build custom rules inside your mapping data flows for data quality and data validation.
- You can build rules that will determine whether values meet an expected value domain.
- Additionally, you can build rules that check for row uniqueness.
- The data preview pane will indicate which rows failed your assertions. Additionally, you can test each row tag downstream using isError() and hasError() for rows that failed assertions.

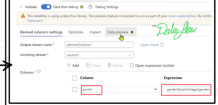




User defined functions

- A user defined function is a customized expression you can define to be able to reuse logic across multiple mapping data flows.
- User defined functions live in a collection called a data flow library to be able to easily group up common sets of customized functions.





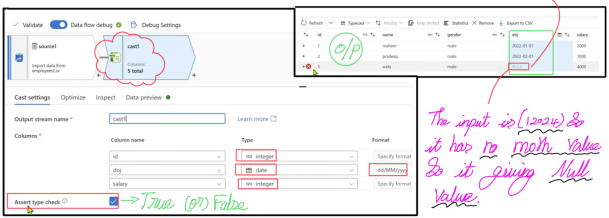
Cast typ / web table Connector / Alert rule :-

Cast Transformation

Use the cast transformation to easily modify the data types of individual columns in a data flow. The cast transformation also enables an easy way to check for casting errors.

I/p

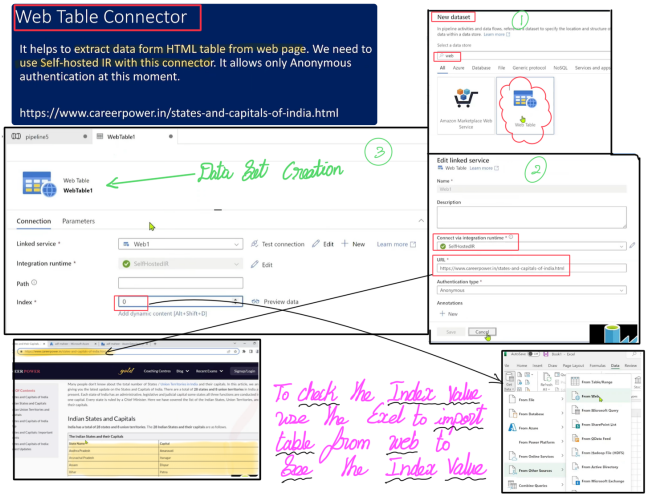
id	name	gender	doj	salary
1	maheer	male	1%2F1%2F2022	2000
2	pradeep	male	1%2F2%2F2022	1000
3	wafa	male	24%2F5303	4000



Web Table Connector

It helps to extract data form HTML table from web page. We need to use Self-hosted IR with this connector. It allows only Anonymous authentication at this moment.

https://www.careerpower.in/states-and-capitals-of-india.html



Alert Rules

You can raise alerts on supported metrics in Data Factory. Select **Monitor > Alerts & metrics** on the Data Factory monitoring page to get started.