**Table of Content**

|  |  |
| --- | --- |
| Introduction | Page No. |
| SQL server | 2 |
| Azure Data Lake Storage | 2 |
| Azure Data Factory | 3 |
| Integration Runtime | 4 |
| Linked services | 4 |
| Lookup Activity | 5 |
| For-Each Activity | 6 |
| If Condition | 7 |
| Full Load | 8 |
| Storage Procedure | 9 |
| Delta Load | 10 |
| SAP Table | 13 |

Documentation on Full Load and Delta load

**Introduction:-**

* We need to create a Azure Pipeline From Source SAP Table to Destination Azure SQL for copy data. In this we perform Full load and Delta Load in a Same Pipeline.
* We have to create a Lookup Table we define which table is required Delta Load and Full Load.
* It is connected to Each activity in which we pass the Source Table name and Destination Table name with their Details.
* In each activity we have If condition which determine if the table will do full Load or Delta Load.

**SQL Server:-**

* It is a software, developed by Microsoft, which is implemented from the specification of RDBMS.
* It is also an ORDBMS.
* It is platform dependent.
* It is both GUI and command based software.
* It supports SQL (SEQUEL) language which is an IBM product, non-procedural, common database and case insensitive language.

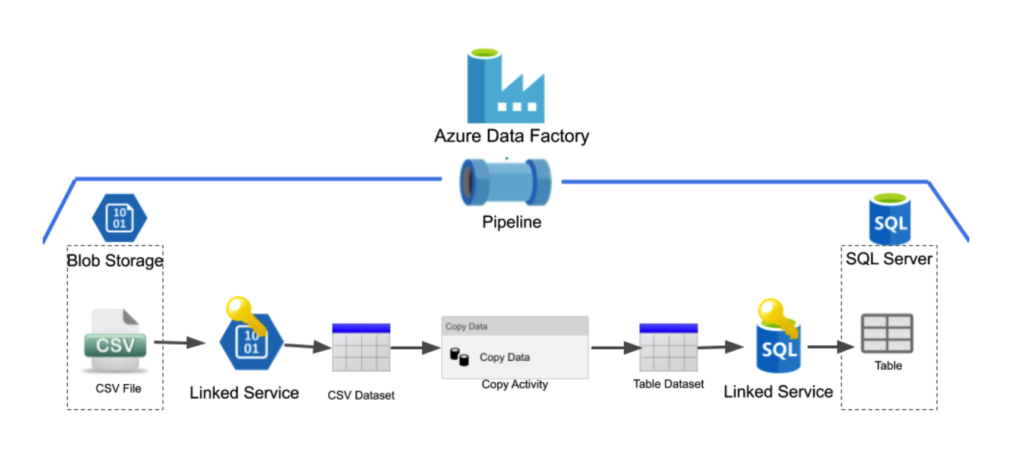
**Azure Data Lake Storage:-**

Itis a massively scalable and secure data lake for high-performance analytics workloads. Azure Lake Data Storage was formerly known and is sometimes still referred to as the Azure Data Lake Store. Designed to eliminate data silos, Azure Data Lake Storage provides a single storage platform that organizations can use to integrate their data. Azure Data Lake Storage can help optimize costs with tiered storage and policy management. It also provides role-based access controls and single sign-on capabilities through Azure Active Directory. Users can manage and access data within Azure Data Lake Storage using the Hadoop Distributed File System (HDFS). Therefore any tool that you’re already using that is based on HDFS will work with Azure Data Lake Storage.

**Azure Data Factory:-**

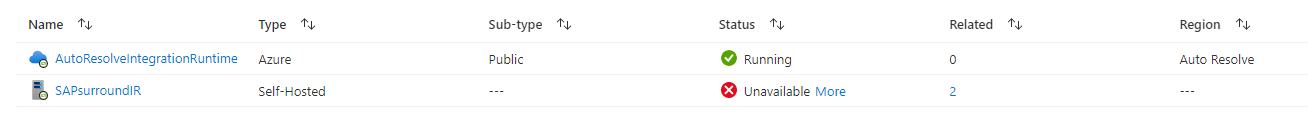
Azure Data Factory is a cloud-based data integration service that allows you to create data-driven workflows in the cloud for orchestrating and automating data movement and data transformation.

It allows you to create data-driven workflows to orchestrate the movement of data between supported data stores and then process the data using compute services in other regions or in an on-premise environment. It also allows you to monitor and manage workflows using both programmatic and UI mechanisms.



**Main Components of Azure Data Factory are:**

**Integration Runtime:-**

****

Integration runtime is the compute infrastructure used by Azure Data Factory (ADF) to provide various data integration capabilities across different network environments.

There are 3 types of Integration Runtime:-

* Azure Integration Runtime
* Self -Hosted Integration Runtime
* Azure-SQL Server Integration Services (SSIS) integration runtime

There are 2 integration runtime we used in this pipeline.

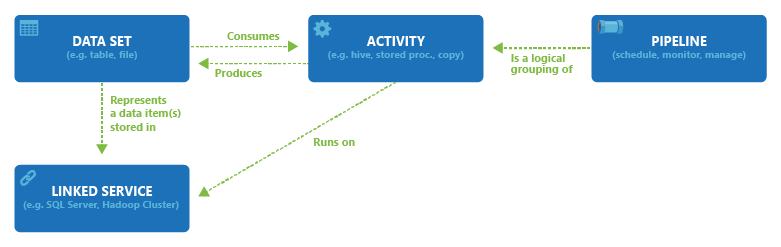
* **AutoResolveIntegrationRuntime**: - Every installation of ADF has a default IR: he **AutoResolveIntegrationRuntime**.

It's called auto resolve, because it will try to automatically resolve the geographic region the compute will need to run. This is determined for example by the data store of the sink in a Copy Data activity. If the sink is located in West Europe, it will try to run the compute in the West Europe region as well.

* **SAPsurroundIR**: - This IR is Self-Hosted. This IR is used to connect SAP table to Azure Data Factory for which we can do ETL (Extract, Transform and Load).

**Linked services:-**

Linked services are much like connection strings, which define the connection information needed for the service to connect to external resources.



Some of the Linked Service we used in this Pipeline is:-

|  |  |  |  |
| --- | --- | --- | --- |
| Linked Service | IR | Type Of Linked Service | Description |
| Is\_SapTable | SAPsurroundIR | SAP Table | This Linked Service is used to connect  Sap table to Azure data Factory. |
| AzureSqlDatabase3 | AutoResolveIntegrationRuntime | Azure SQL Database | This Linked Service Is used to Connect  Azure Data factory to Azure SQL  Database |
| AshutoshAzureBlobStorage3 | AutoResolveIntegrationRuntime | Blob Storage- Delimited Text | This Linked Service is used to connect Blob Storage to ADF. |
| AzureSqlDatabaseForFUll\_Load\_SapToSSMS | AutoResolveIntegrationRuntime | Azure SQL Database | This Linked Service Is used to Connect  Azure Data factory to Azure SQL  Database |

**PipeLine:-** A pipeline is a logical grouping of activities that performs a unit of work. Together, the **activities** in a pipeline perform a task.

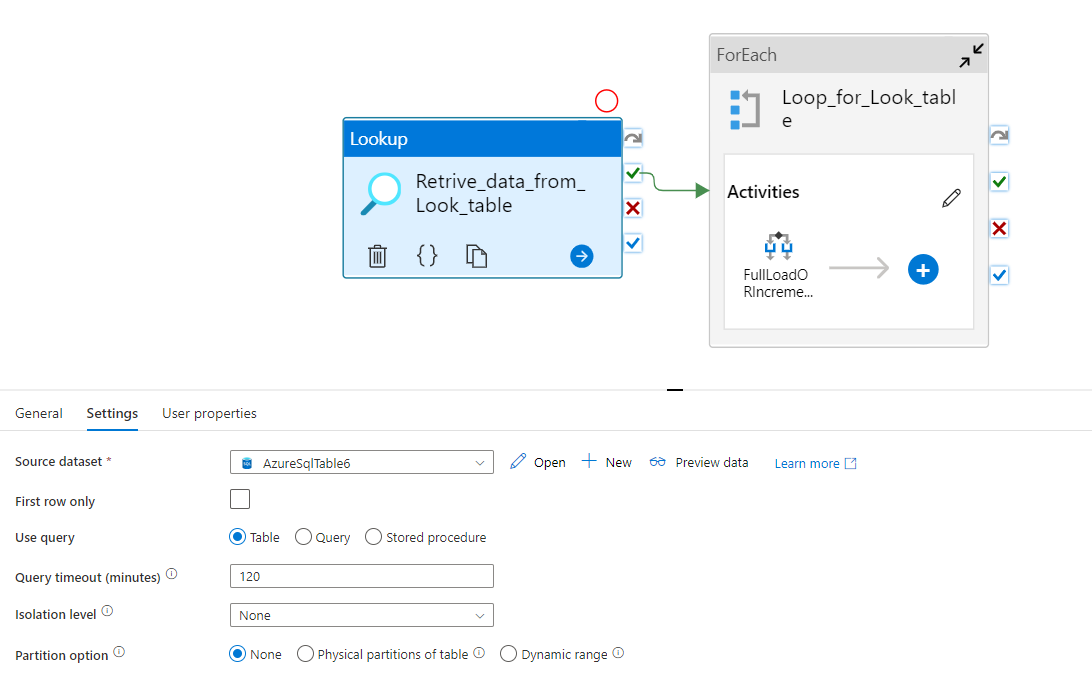
**Datasets:-** Datasets are used in source and sink transformations. The datasets define the basic data schemas.

**Data Flows:-** Data flows are visually designed data transformations that allow us to develop data transformation logic without writing code.

**Trigger:-** Trigger is a processing unit that determines when to begin or invoke an end-to-end pipeline execution. In other words, Triggers are used to schedule a Data Pipeline runs without any interventions.

**Activity:-** The activity is the task we performed on our data. We use activity inside the Azure Data Factory pipelines.

**Lookup Activity:-** Lookup activity can retrieve a dataset from any of the data sources supported by data factory and Synapse pipelines.



In above Lookup Activity (Retrive\_data\_from\_Look\_table) we retrieve SQL table which have all the source table name and which type of copy(full or Delta load) activity is used maximum value, and etc.

[Source1]:- It store the table name from SAP table which we want to copy data from SAP to SQL database.

[Sink1]:- It store the table name of the Destination table store in SQL Database.

[SCHEMA1]:- It store the schema name of destination table store in SQL Database.

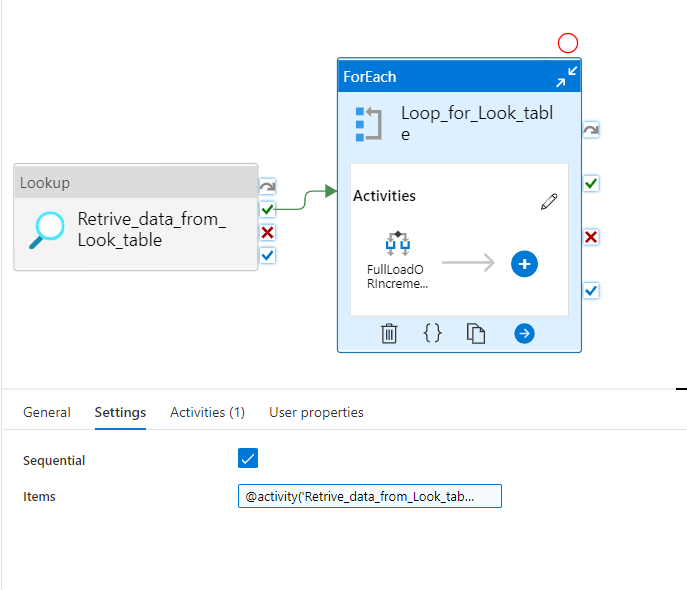
[PullFlag]:- This is Boolean datatype. It store data if we want to do copy activity on the Table.

[DeltaFlag]:- This is Boolean datatype. If the value is zero then it will do full copy an if the value Is one then it will be Delta Load.

[DeltaColumn]:-It store the column name in which we store the column name on which we have to do Delta Load.

[MaxOfValue]:- It store the maximum value on which last time we have updated the Table.

**ForEach Activity:-** The ForEach Activity defines a repeating control flow in an Azure Data Factory or Synapse pipeline. This activity is used to iterate over a collection and executes specified activities in a loop.



In above we make for loop for Lookup table in which we have store name of table and storage table name with the condition.

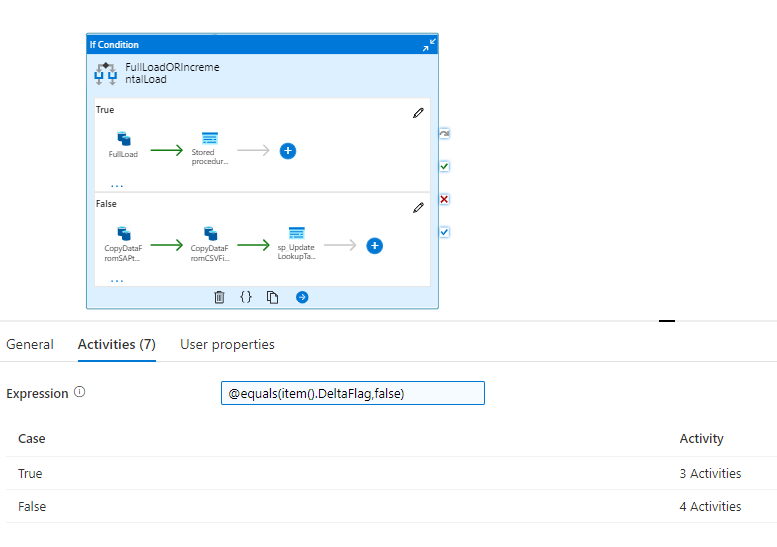
IN above we make loop in sequential mode so that each row work on proper manner.

There need to be an condition on for loop and we put condition:

|  |
| --- |
| @activity('Retrive\_data\_from\_Look\_table').output.value |

There is One activity in the for-loop that is If-else activity.

**If-Condition Activity:-** The If Condition activity provides the same functionality that an if statement provides in programming languages. It executes a set of activities when the condition evaluates to true and another set of activities when the condition evaluates to false.



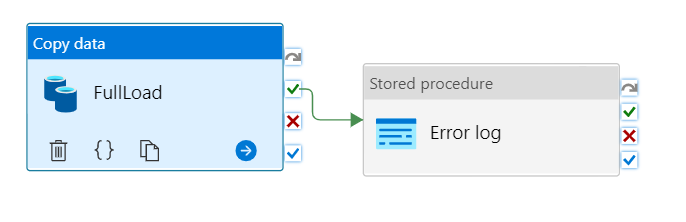
In above there is If-Condition by which we can find if we have to do full Load or Delta Load activity in the above Pipeline.

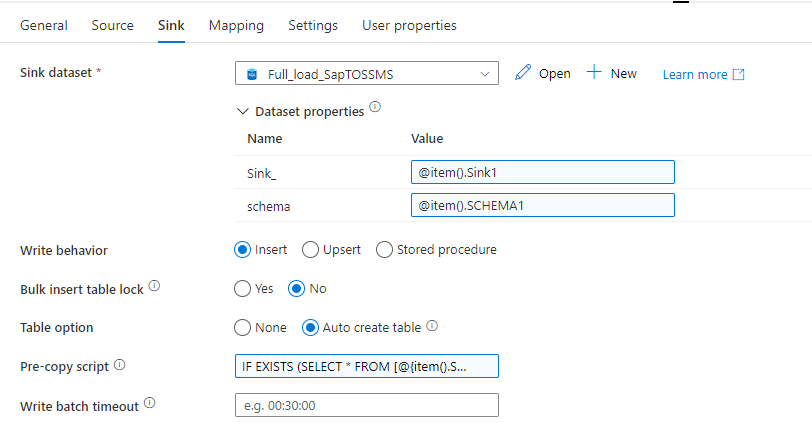
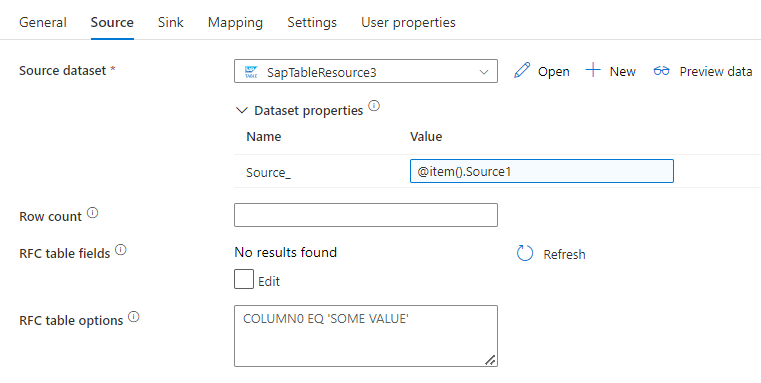
In the above If –else Condition we need a condition on which it can decide which part he have to execute and condition we have is

|  |
| --- |
| @equals(item().DeltaFlag,false) |

From above condition if delta flag is false then it will perform Full Load, and if The delta Load is True then Above condition is False and it will execute activity which is in the False condition.

**Full Load:-** In Full load we have to copy from SAP table to SQL database. So we already identify which table we have to do in Full Load. In this we have to provide Source dataset in Source. The Source Dataset have the LinkedService [Is\_SapTable] and Dataset is [SapTableResource3].In dataset properties we give name of the table which we want to retrieve from SAP table dynamically that is Source\_ @item().Source1 {This condition will have Source1 From Lookup Table}.

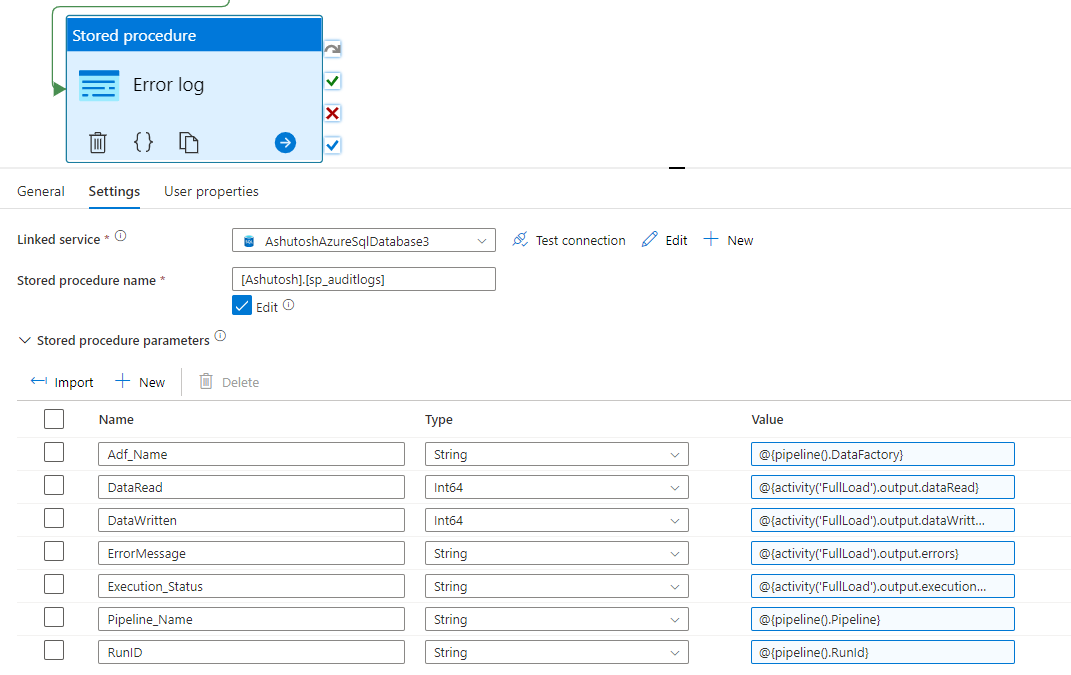
****



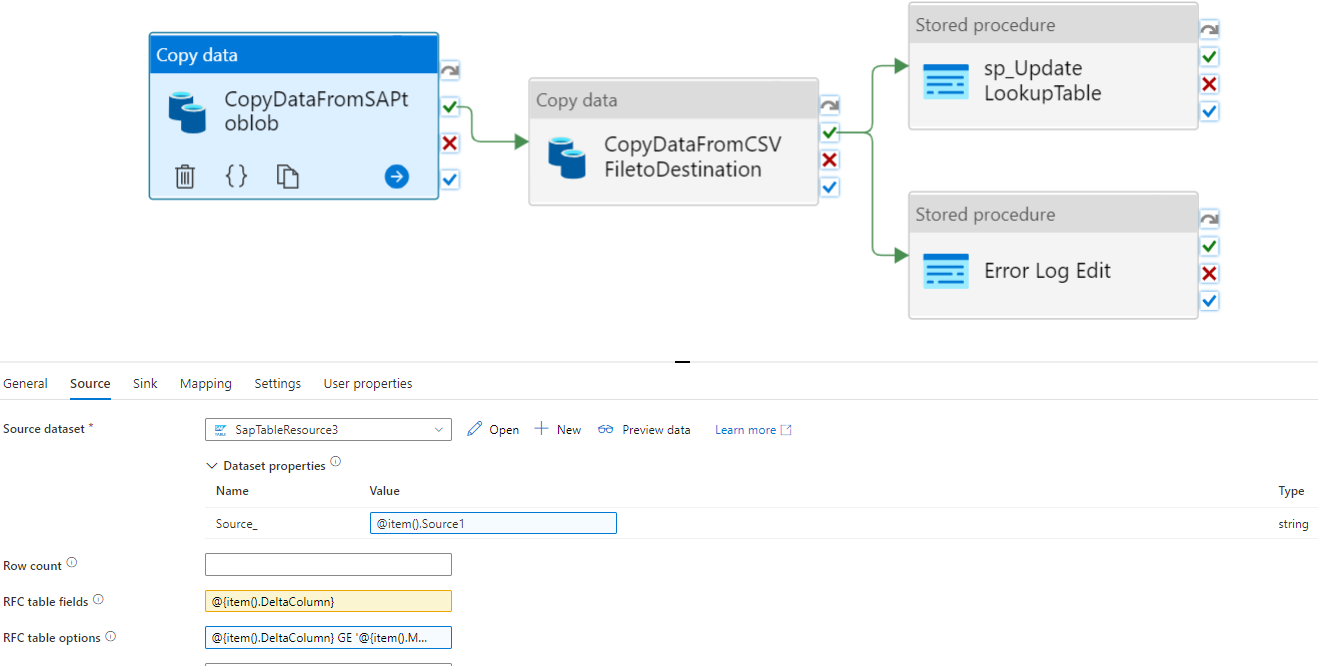
In Sink Section we have to put data details where we have to copy data[destination location details]. Here we have dataset name **Full\_Load\_SapToSSMS** with Linkled servies name is **AzureSqlDatabaseForFull\_Load\_sapToSSMS**. It have 2 Parameter have **Schema** which Store the Schema name of the table and **Sink\_** in which we pass the name of the table on which we want to store.

There is **Pre-copy script** in which we put condition if there is table of same name then it will be drop that table from the database from the destination side.

**Log Edit:-** It is a stored Procedure by which we can store the details of the activity or situation of activity . Here we connect the Stored Procedure to {Ashutosh}.[sp\_auditlogs]using Linked Service **AshutoshazureSqlDatabase3.** From here we pass parameters which we will store in table on the database which is present in SQL Database.

****

**Delta Load:-**

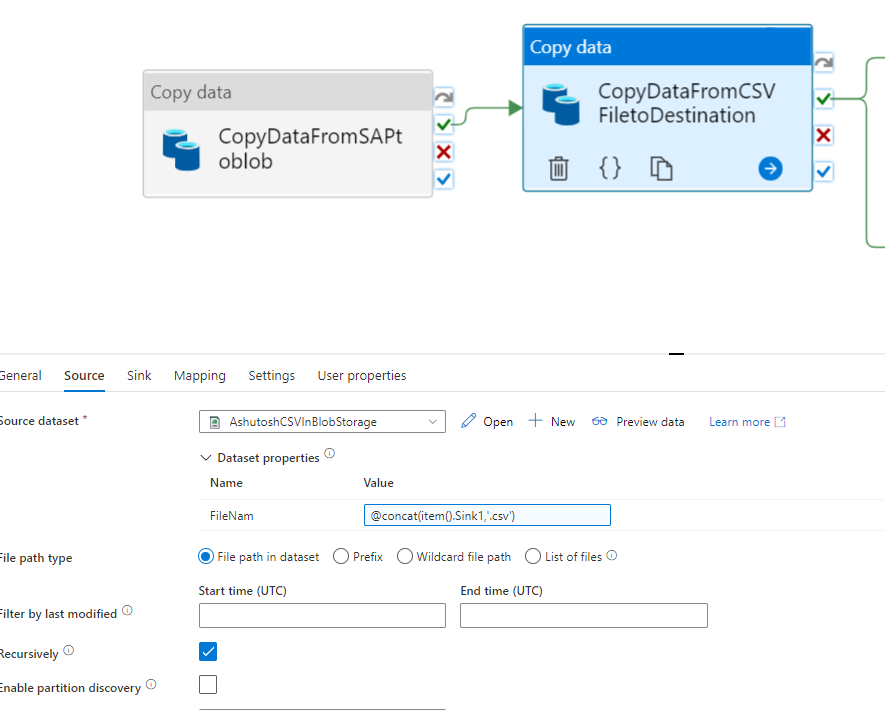


In above image there are 4 activities.

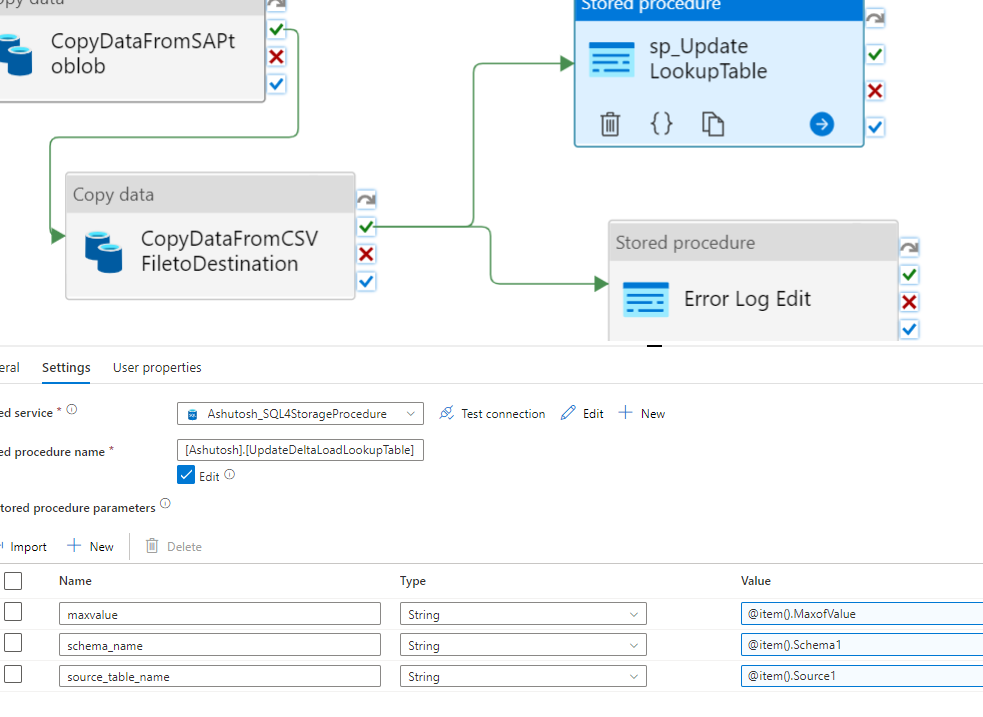
In the first 1st activity we copy data from SAP table using **Dataset** [SapTableResource3] having **Linked Service** [Is\_SapTable] To Azure Blob Storage using **Dataset** [AshutoshCSVInBlobStorage] having **Linked Service** [AshutoshAzureBlobStorage3].

Here in source , we are selecting RFC table field dynamically from lookup table to check the date that should be greater than the last load date by giving a condition in RFC table option.

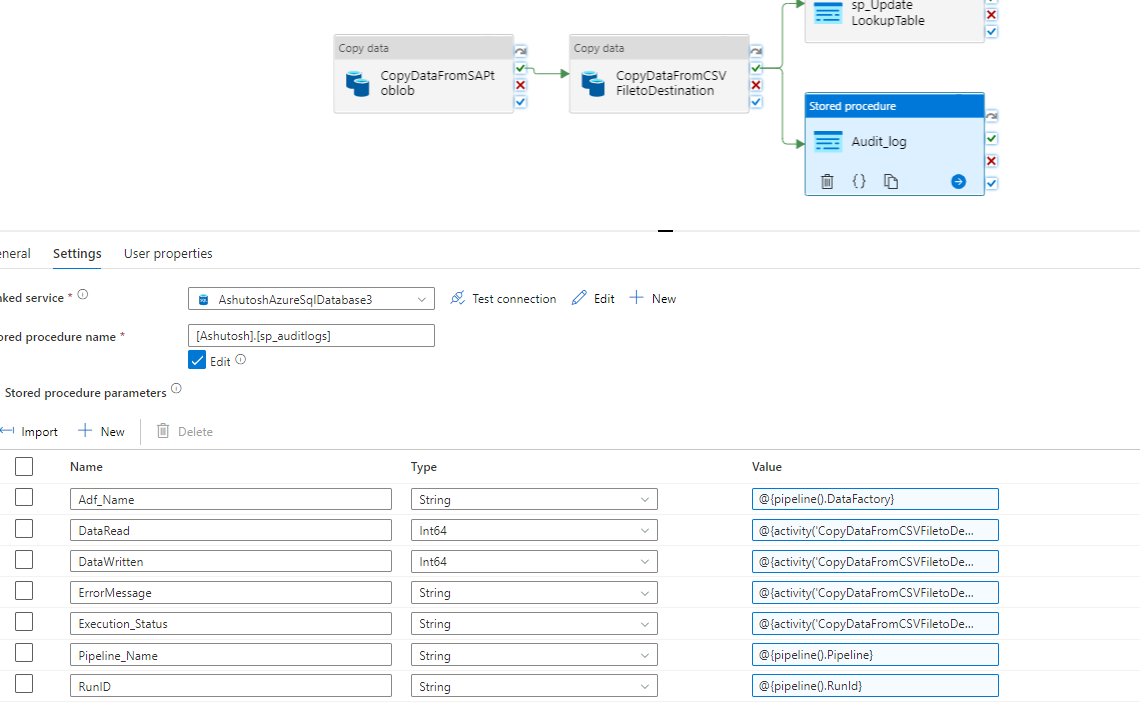
|  |
| --- |
| @{item().DeltaColumn}  GE  '@{item().MaxofValue}' |



In above image now after copying data from Sap table to Azure Bob Storage then we will copy data from Blob Storage to SQL database. Here we put name of file Dynamically like above.



After Successful copy from Blob storage to SQL database then it will perform 2 Storage Procedure at the Same time. One of the Storage Procedure in above we update Last Max value so next time it will copy data from there.



Now in Second Storage Procedure it pass parameters of the pipelines to Storage Procedure and store it in a table for in which we know which pipeline have worked and which have failed with thee reason.

Audit log:

Added a stored procedure activity  at the to add the logs like data factory name , Pipeline name , error message, run id , trigger name etc. to add them in lookup table.

**The tables which are Loaded From SAP table are:-**

|  |  |  |
| --- | --- | --- |
| Table Name | Load Type | Detail about table name |
| VBEP | Incremental Load | VBEP is a standard Sales Transparent Table in SAP SD application, which stores Sales Document: Schedule Line Data data. |
| TKA01 | FULL | Controlling areas / assignment |
| TSPAT | FULL | TSPAT is a standard SAP Table which is used to store Organizational Unit: Sales Divisions: Texts data and is available within R/3 SAP systems depending on the version and release level. |
| T001W | FULL | T001W is a standard Plant Master Transparent Table in SAP Logistics application, which stores Plants/Branches data. |
| CEPC | Incremental Load | Profit Center Accounting Transparent Table |
| TGSB | FULL | TGSB is a standard Financial Accounting Transparent Table in SAP FI application, which stores Business Areas data |
| TGSBT | FULL | Business Areas |
| ISDSLSORDERITEM | FULL | ISDSLSORDERITEM is a standard Sales Orders General View Structure in SAP SD application. |
| ISDSALESORDER | FULL | ISDSALESORDER is a standard Sales Orders General View Structure in SAP SD application |
| ICALENDERDATE | FULL | ICALENDARDATE is a standard General Application Functions General View Structure in SAP CA application. |
| CSKS | Incremental Load | Cost center Master Dataset |
| KNA1 | Incremental Load | Customer Master Transparent Table |
| MAKT | FULL | MAKT is a standard Material Master Transparent Table in SAP Logistics application, which stores Material Descriptions data. |
| SKA1 | Incremental Load | SKA1 is a standard Financial Accounting Transparent Table in SAP FI application, which stores G/L Account Master (Chart of Accounts) data |
| SKB1 | Incremental Load | SKB1 is a standard Basic Functions Transparent Table in SAP FI application, which stores G/L account master (company code) data. |
| TKA02 | FULL | TKA02 is a standard Controlling Transparent Table in SAP CO application, which stores Controlling area assignment data |