**How to Create Solutions in steps by steps?**

How would you structure your pipeline? Moreover, considering the potential emergence of new requirements in the future, wherein the source shifts to a Rest API and the destination transforms into Azure Data Lake Gen2, with a stipulated file format of .txt or something else?

We have established two distinct tables to manage data load configurations. The first table is designated for transferring data for a specific Business outcome from the relevant source to the relevant sink, while the second table is specifically designed for moving data for Delta Load. Each of these tables is populated with rows that represent unique source-destination pairs, encompassing crucial information such as the source table name, source schema, Operation (whether it’s a full or incremental load, Deletion, Updating, executing Store procedure, Calling API, etc). Furthermore, second table which gets updated upon completion of each delta load operation. These result fields encompass statistics like the number of inserted rows, updated rows, and the maximum transaction date associated with the respective tables.

**Step 1: Config Tables Designs**

There are two main tables which handle delta load, full load, deletion(list of tables based on condition), updating (list of tables based on conditions), etc

**Pipelines\_Config**

CREATE TABLE config.**Pipelines\_**Config (  
 Id int NOT NULL identity Primary Key,  
 SpecCode VARCHAR(100),  
 SourceName VARCHAR(50) NOT NULL,  
 TargetName VARCHAR(50) NOT NULL,  
 SourceType VARCHAR(20) NOT NULL,  
 TargetType VARCHAR(20) NOT NULL,  
 WhereCondition VARCHAR(500),  
 Operation VARCHAR(20) NOT NULL,  
 FieldList VARCHAR(500),  
 Directory VARCHAR(500) default ‘ ‘,  
 TargetFileExtention VARCHAR(10) default ‘’  
);

In the context of the Delta Load process, it’s worth noting that we’ve chosen to use a primary key as the condition for data extraction, rather than relying on a date window as a basis.

**Pipeline\_DeltaLoad**

CREATE TABLE [config].[Pipeline\_DeltaLoad] (  
 [Id] INT IDENTITY (1, 1) NOT NULL,  
 [SpecCodeId] INT NULL,  
 [MinWindowStartValue] INT DEFAULT ((0)) NULL,  
 [LastProcessedTimestamp] DATETIME2 (7) DEFAULT (getdate())   
);

You can consider utilizing the ‘**MaxWindowStartDate**’ and ‘**MinWindowStartDate**’ columns in the table above as alternatives to ‘**MinWindowStartValue**,’ depending on the data available in your source system. If your source system includes a ‘createdDate’ field, it would be appropriate to make use of the ‘**MaxWindowStartDate**’ and ‘**MinWindowStartDate**’ fields. On the other hand, if your source system relies on a primary key for data loading decisions, then ‘MinWindowStartValue’ would be the suitable choice.

**Functionality based config Email\_Config**

To meet the specific needs of business outcomes that require email functionality, it’s essential to establish a dedicated configuration table. This table will serve as the repository for capturing and managing relevant information, such as email templates, recipient lists, and other related settings. By maintaining this separation, you can streamline the management of email-related configurations, ensuring they are distinct and easily accessible for the relevant business processes. Additionally, this approach promotes a more organized and efficient system for handling email functionality, making it easier to update and customize as needed.

CREATE TABLE config.PipelinesEmail\_Config (  
 Id int NOT NULL identity Primary Key,  
 SpecCode VARCHAR(20),  
 SpecCodeId int NOT NULL,  
 ToEmail VARCHAR(100) NOT NULL,  
 FrmEmail VARCHAR(50) NOT NULL,  
 CcEmail VARCHAR(100) ,   
 Title VARCHAR(100) ,  
 CustomizedMessage VARCHAR(1000) NOT NULL,  
 IsAttachedMent bit default 0,  
 Path VARCHAR(100) NOT NULL  
 );

**Step 1.1: Preparing list of Pipelines based Business outcome**

As per the provided requirements, three primary business objectives are identified, leading to the creation of three main pipelines. These pipelines are labelled with respective Spec codes: BO001, BO002, and BO003.

**Step 1.2: Preparing Scripts for above config tables for above three business outcomes.**

Provide a sample script for each of the three items pertaining to each business outcome. If you have 100 sources, ensure there are 100 corresponding entries.

**Please find sample table data from GitHub**[**MetadataDrivenPipeine/Script.sql at main · alpaBuddhabhatti/MetadataDrivenPipeine (github.com)**](https://github.com/alpaBuddhabhatti/MetadataDrivenPipeine/blob/main/Script.sql)

Also, we need sample data for perform operation on target tables. Here You can find sample data, I have used for demo.

**BO001: Footboll data Tables:** [MetadataDrivenPipeine/Footboll.zip at main · alpaBuddhabhatti/MetadataDrivenPipeine (github.com)](https://github.com/alpaBuddhabhatti/MetadataDrivenPipeine/blob/main/Footboll.zip)

**BO002: Crhicket Data Tables :** [MetadataDrivenPipeine/Cricket.zip at main · alpaBuddhabhatti/MetadataDrivenPipeine (github.com)](https://github.com/alpaBuddhabhatti/MetadataDrivenPipeine/blob/main/Cricket.zip)

**BO003: Movie related data files :** [MetadataDrivenPipeine/Movie.zip at main · alpaBuddhabhatti/MetadataDrivenPipeine (github.com)](https://github.com/alpaBuddhabhatti/MetadataDrivenPipeine/blob/main/Movie.zip)

**Step 3: Designing ADF Pipelines**

If you are using a Virtual Network, you will need to plan the design for Vnet, Subnet, Managed IR, and Managed private endpoints.

**3.1 Create Linked Services for Sources and Sinks.**

a. SFTP:

* If it’s on-premise SFTP, you’ll require a self-hosted IR setup.
* If it’s Azure Blob Storage with SFTP, you only need an Azure Blob Storage Linked Service.

b. Azure Blob Storage:

* If “a” is Azure Blob Storage with SFTP, and you need the same Azure Storage Account, there’s no need to create this linked service. Otherwise, you should create it.

c. Azure SQL Server/Database.

**3.2 Create Datasets for Source and Sink.**

a. SFTP/Azure Storage Account(F**or BO001 as sink)**:

* One dataset is required since there is only one SFTP source for Business Outcome 1.

b. Azure Storage Account (F**or BO002 as sink and BO003 as Source)**:

* Two datasets are needed as there are two business outcomes utilizing Azure Blob storage either as a source or sink.

c. Azure SQL Database (For **BO002 as sink and BO001 & BO003 as Source**):

* Two datasets are necessary as there are two business outcomes utilizing Azure SQL Database either as a source or sink.

**3.3 Create Pipelines**

While it is possible to create a single pipeline to encompass all three business outcomes or opt to maintain three distinct main pipelines, each catering to an individual business outcome. We have created a single pipeline “BOS”.