Azure Data Factory (ADF) test case

Main goal of this test case:

This test case is designed to see what your approach is in creating a pipeline in ADF. It is not mandatory to implement it in that environment but, how you solve the problem and build this pipeline is more important to us. You can even sketch the whole pipeline in any documentation and present. Of course, there is no best and unique solution for any problem. We know people have different viewpoints for one problem and we appreciate that in our team. What is important to us is:

- How you perceive the problem
- What is your approach towards that
- Which module in ADF you are using for each step and your reasoning behind that

Process definition:

There is a process called “Unique Supersession”. This process takes data (like 1K rows per dealer and parts for that dealer). The order includes dealer number, part number and more information for part and dealer. The main responsibility of this process is to check if there is any part which is replaced by another part. In more detail, it checks the “status” and “structure” of each received part. If the status is “Final” and the structure is “Straight”, then, the process understands that the part is replaced by another specific part and replace the existing part number with the new part number else, everything is the same as before and the part has not been replaced.

Objective:

You want to implement this process in ADF as a pipeline which runs every day at specific time. You can use any module which you think suits from there.

Here are steps to create this pipeline:

Task 1:

Everyday, there is a process reading data from table called “INGESTION” and selects required columns and puts them in another table called “INPUT_DYNAMIC_DATA”.

Here are columns in “INGESTION” table:

- DEALER_ID
- SSO_ID
- STATUS
- STRUCTURE
- SP_ID
- ORDER_ID (PK)
- ORDER_NUMBER
- DISTRICT_ID
- LIBRARY_ID
- PRODUCT_ID
- COMPANY_CODE

Here are columns in “INPUT_DYNAMIC_DATA” table:

- DEALER_ID
- SSO_ID
- STATUS
- STRUCTURE
- SP_ID
- ORDER_ID (PK)
- ORDER_NUMBER

Notes:

- PK here means the primary key for this table.
- “INPUT_DYNAMIC_DATA” table shows which part number (SP_ID) has what STATUS and STRUCTURE.
- In “INPUT_DYNAMIC_DATA” table, each dealer (DEALER_ID) has unique part (SP_ID). So, the combination of part per dealer is unique in this table.

Define a process which extracts data from “INGESTION” table and insert all rows into a table named “INPUT_DYNAMIC_DATA”.

Task 2:

Create a process implementing the business rule for “Unique Supersession”. This process takes data from “INPUT_DYNAMIC_DATA” and streams them to table “RESULT” regarding the rule mentioned in the “Process definition” part.

It means that, for each part:

if the “STATUS” = ‘F’ and “STRUCTURE”=’S’, then, SP_ID = SSO_ID(the part number should be replaced by another part number while streaming to table “RESULT”).

Else: do nothing and stream the whole row without any change to “RESULT” (the SP_ID will remain as before)

Task 3:

Create the whole pipeline by gathering all these steps.

NOTES:

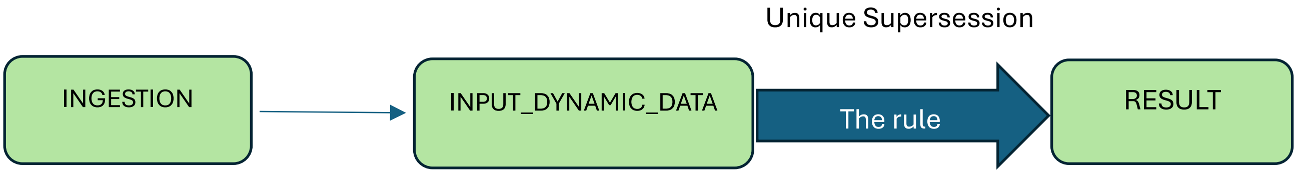
You decide how each table looks like.

You can add or remove any columns but please mention your reason behind that.

It is ok if you do not have time to prepare any data for all these tables. As long as you can explain the implementation, it is ok for us.

During the meeting you will be asked to go through your presentation and explain your approach.

Here is a brief overview of this process:


Unique Supersession
INGESTION
INPUT_DYNAMIC_DATA
The rule
RESULT