Contents

[Introduction 2](#_Toc32427934)

[Architectural Diagram 2](#_Toc32427935)

[Source Systems 2](#_Toc32427936)

[File Format 2](#_Toc32427937)

[Schema 2](#_Toc32427938)

[Landing layer 2](#_Toc32427939)

[Staging Layer 2](#_Toc32427940)

[Analytic Mart Layer 2](#_Toc32427941)

Introduction

The efficient flow of data from one location to the other - from source system to a data warehouse, for example - is one of the most critical operations in today's data-driven enterprise. After all, useful analysis cannot begin until the data becomes available. Data flow can be precarious, because there are so many things that can go wrong during the transportation from one system to another: data can become corrupted, it can hit bottlenecks (causing latency), or data sources may conflict and/or generate duplicates. As the complexity of the requirements grows and the number of data sources multiplies, these problems increase in scale and impact. The key thing is to test each layer thoroughly from Data and functional perspective

Architectural Diagram

EDW



CRM

GCP Integration Server



FMW

Source Systems

File Format

Files are landed as dat file or as text file with pipe delimiter as a gzip compressed file.

Schema

**Schema for the landing table :**

* We need to define the schema in a json file. The best approach would be to have the datatype for all the fields as String. We can cast to the appropriate datatype in the staging layer.
* Datatypes of the landing table should be put as string , so that we won’t have the risk of losing data or any data corruption.

Landing layer

Staging Layer

Analytic Mart Layer

**Transfer files from Integration Server to Google Cloud storage bucket** :

cntrl-m jobs will begin transferring data to the p-asna-aadp-landing-01 bucket automatically.

**File Format :** Pipe delimited files compressed as gzip

**Schema for the landing table :**

We need to define the schema in a json file. The best approach would be to have the datatype for all the fields as String. We can cast to the appropriate datatype in the staging layer.

**Datatypes of the landing table:**

Datatypes of the landing table should be put as string , so that we won’t have the risk of losing data or any data corruption. Check the datatype of the landing table.

Basic Checks in the Staging bqs ::

* Datatypes are mentioned as string or not in landing table.
* **For transaction related table and item and store** : Two separate LB/CA landing tables are getting created or not
* **For CRM tables** : Check the landing tables bq load command are correctly written or not

Landing

Staging

Analytic Mart

* Replace the original staging table as the source in the mart script
* Go through the UTs of staging bqs and see the results. Make sure the test cases are all in the passed stage. As that will give you a confirmation for you to use that in the mart script.
* Check, is history being maintained or not in staging.
* Check the datatype of the key columns in staging is compatible with the datatype of the mart table.
* Post loading, The keys of the new mart tables are in sync with the keys of the existing mart table for overlapping data.
* Check the decimal value measure which are coming from staging table are numeric and have precision till 2 decimal places
* Check the uniqueness of the mart table.
* Check the foreign key integrity.
* Check the validity of the UTs at least over two consecutive days of load.
* Check the archival is happening correctly or not by checking the archival path.