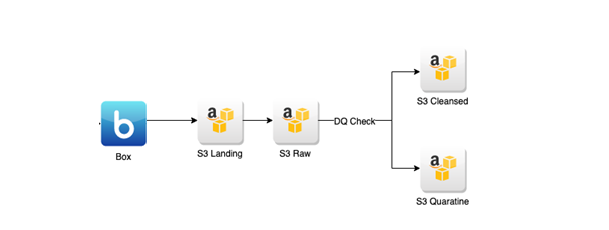
The goal is to ingest data from flat file available on external storage to AWS S3 storage. Business requirements provided from client’s side give definition of bad data that should be rejected and moved to quarantine until further instructions.

On S3 data passes through the following layers: Landing Zone, RAW, CLEANSED.



Types of loads supported:

* Full (Update)
* Incremental

Landing Zone Layer Characteristics:

* The LZ is a secure intermediate location where the data acquired from various external data sources will be stored before loading it into the RawZone.
* The data structure and content in the LZ mirrors that from the source.
* Target location on S3: s3://test-bucket/lz/forecast/

Raw Layer Characteristics:

* Data is sourced via the LZ
* Data in the RL is immutable. New data is always appended to existing data.
* Partitioned by run\_dt – date time of running pipeline, subfolder with a date component under the table or entity folder.
* Data in the RL is persisted in Parquet format.
* Target location on S3: s3://test-bucket/raw/forecast/

Cleansed Layer Characteristics:

* Data is read from the Raw Zone and checked for quality before being cleansed and standardized
* Data that does not pass the rules for completeness, correctness and coherence is moved to a quarantine location for later remediation.
* Data in Cleansed is persisted in Parquet format.
* The process of cleansing happens after the data is profiled and data quality
* Data Merging: data is collected from incremental single-source data sets, de-duped and merged into the full data set of that single-source.
* No partitioning on this layer
* Target location on S3 for incremental load: s3://test-bucket/cleansed/incremental/forecast/
* Target location on S3 for merged data: s3://test-bucket/cleansed/merged/forecast/
* Target location on S3 for bad data: s3://test-bucket/quarantine/forecast/

Source file format requirements:

FileName should follow ForecastData\_YYYYMMDD.xlsx

where YYYYMMDD is a date when file uploaded to source location

Source data description:

File contains forecast rates for different stores by month.

Incremental rules:

On cleansed layer, when we do any kind of update data is getting into Incremental folder first and then being merged with data from Merged folder, so that merge folder always contains relevant data to be used for further processing.

* If forecast rates were updated for already existing combination of store group + store\_group1+ store\_group2 + year\_month then data is being updated in target
* If new forecast rates are coming – then it is just added.

Target Data Description:

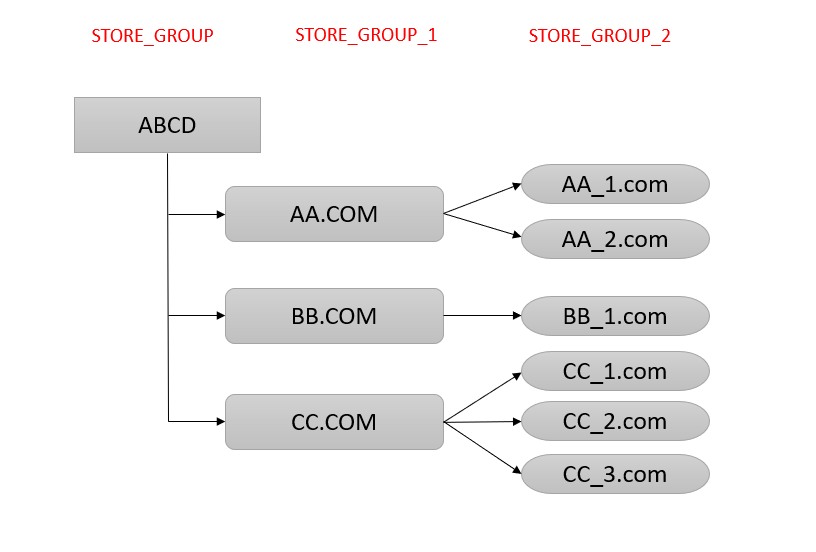
|  |  |  |  |
| --- | --- | --- | --- |
| **Target Column Name (CLEASED)** | **Nullable** | **Data Type** | **Other data requirements** |
| YEAR\_MONTH | N | date |  |
| STORE\_GROUP | N | nvarchar (100) | Has one value only ='ABCD' |
| FCST\_RATE | Y | decimal (5,3) |  |
| STORE\_GROUP\_1 | N | nvarchar (100) | 3 allowed values: ‘AA.COM’, ‘BB.COM’, ’CC.COM’ |
| FCST\_RATE\_1 | Y | decimal (5,3) |  |
| STORE\_GROUP\_2 | N | nvarchar (100) | 6 allowed values: ‘AA\_1.COM’, ‘AA\_2.COM’,’BB\_1.COM’, ‘CC\_1.COM’, ‘CC\_2.COM’, ’CC\_3.COM’ |
| FCST\_RATE\_2 | Y | decimal (5,3) |  |
| InsertDateTime | N | datetime | System column, full load timestamp |
| UpdateDateTime | N | datetime | System column, incremental load timestamp |
| file\_name | N | nvarchar (100) | System column, store source file name |
|  |  |  |  |

Other Data requirements:

1. General format for date fields should be YYYY-MM-DD
2. File type is excel (xlsx)
3. PK = YEAR\_MONTH + STORE\_GROUP + STORE\_GROUP1 + STORE\_GROUP2
4. Dates should be specified in UTC time zone
5. Forecast rate cannot be greater than 100 % (0 and 100 included)
6. FRCT\_RATE\_2 can be empty in case FCST\_RATE and FCST\_RATE\_1 is not null
7. FRCT\_RATE\_1 can be empty if FCST\_RATE is not null
8. Hive table should be created once data is processed based on cleansed layer data:

table name = forecast\_cleansed

1. STORE GROUP HIERARCHY exists:



Pipeline requirements:

1. Frequency = Daily
2. Trigger = By Time (5 PM UTC)
3. File can be processed only once.
4. If no files arrived by required time – trigger alert notification to client
5. In case there are no new files on storage next day, pipeline should not process yesterday’s file again

Task1

Define which DQ measures should be checked on different layers

|  |  |  |  |
| --- | --- | --- | --- |
| DQ | LZ | RAW | CLEANSED |
| Completeness | Yes | Yes | Yes |
| Uniqueness | Yes | Yes | Yes |
| Timeliness | Yes | Yes | Yes |
| Validity | Yes | Yes | Yes |
| Accuracy |  | Yes | Yes |
| Consistency | Yes | Yes | Yes |

Task2

Provide list of all DQ checks for all layers in details

Task3

Design test data to cover all specified requirements (provide both negative and positive cases). Set of excel files is expected to be received from you.

Task 4

Provide test cases and scenarios to cover all possible testing types. Please follow provided test case format.