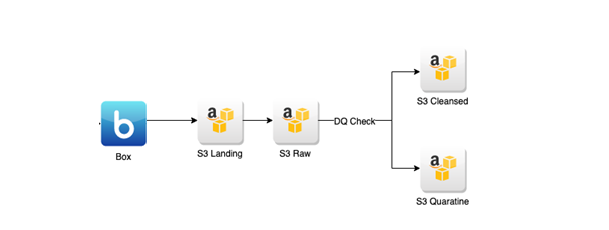
The goal is to ingest data from flat file available on external storage to AWS S3 storage. Business requirements provided from client’s side give definition of bad data that should be rejected and moved to quarantine until further instructions.

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:

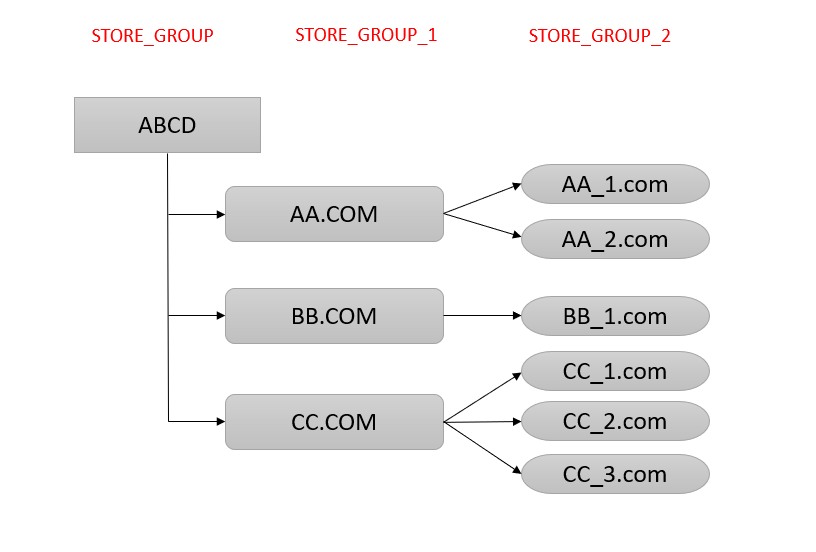
|  |  |  |  |
| --- | --- | --- | --- |
| **Target Column Name (CLEASED)** | **Nullable** | **Data Type** | **Other data requirements** |
| YEAR\_MONTH | N | date |  |
| STORE\_GROUP | N | nvarchar (100) | Has one value only ='ABCD' |
| FCST\_RATE | Y | decimal (5,3) |  |
| STORE\_GROUP\_1 | N | nvarchar (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 |  | X | X |
| Uniqueness |  |  | X |
| Timeliness | x | X | X |
| Validity | x | X | X |
| Accuracy |  |  | X |
| Consistency | x | X | X |

Task2

Provide list of all DQ checks for all layers in details.

**Landing Zone**

* **Validity**: the incoming files should be valid, there must be a mechanism about what we expect to receive like filename, format (csv, tsv, txt). Column names.
  + **Data should be sent to the location:** s3://test-bucket/lz/forecast/.
  + Other location is not acceptable.
  + The name of the file should be valid.
  + The date within the file name should be valid.
* **Consistency**: The data should be consistent in terms of columns, data type, column name. Data from source should match with target tables, if not this file should be rejected.
* **Timelines.**  What happen if the same day the landing zone receives same file twice, which file should pick first? the system should be able to pick the last and most recent file.

**Raw Zone**

* **Completeness.** If there are 1000 rows in users table within landing zone from datetime 01-01-2023 load there should be 1000 rows in Raw zone from users table for the date 01-01-2023 load date. Data among landing zone and raw layer should be the same in all columns.
* **Timelines.** Data should be partitioned by DT and should include date load, so we can track by load date time. The data should be stored by sub folders, the name of the folder should include the datetime.
* **Validity.** The data type included on the documentation, should match Vs. the metadata from each table for instance, we need to make sure integers, date time, floats are properly aligned with the documentation.
  + Verify data in the RL is persisted in Parquet format.
  + Verify the target location is S3: s3://test-bucket/raw/forecast/
    - Other locations are not acceptable.
* **Consistency.** Columns between source and target tables are the same. The incoming files are consistent with the information already stored in the DB.

**Cleansed zone.**

1. **Completeness.** Data received from raw zone should match the number of rows.
2. **Uniqueness.** As the data is merged, there should not be duplicated rows for each row.
3. **Timeliness.** On this layer, we don’t have partitions, but we should be able to get the latest version of the row.
4. **Validity.** We need to review the documentation in order to see what is considered a valid record. In case some critical and mandatory columns are null, we should send these records to quarantine.
   1. Records should be persisted in parquet format.
   2. Validate cleansed and incremental data is sent to:
      1. s3://test-bucket/cleansed/incremental/forecast/
   3. Validate cleaned and merged data is sent to:
      1. s3://test-bucket/cleansed/merged/forecast/
   4. Validate bad data is sent to:
      1. s3://test-bucket/quarantine/forecast/
5. **Accuracy.** As the data is merged, we can only have one row per record, so we need to make sure we have the latest version of the row.
   1. **Validate the scenarios below are met:**
      1. 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
      2. If new forecast rates are coming – then it is just added.
6. **Consistency.** The updates from existing rows should be accepted in terms of data consistency. The insertions for new records should be properly stored in the right columns with the right data type.

**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.

The test data covers dates for a whole year, from April 2019 to April 2020.

As a Positive scenario I provided file: Forecast\_20200401**.**xlsx

As a negative scenario I provided file Forecast\_20200401\_n.xlsx

1. There is no data for the month and year included on the file name.
2. There are data in the wrong format for the column year month.
3. There are incorrect values on store group column.
4. Incorrect values on column store group 1.
5. Incorrect values on column Fcst Rate1
6. Incorrect values on column storegroup 2
7. Percentage values greater than 100% in Rate columns.

Task 4

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