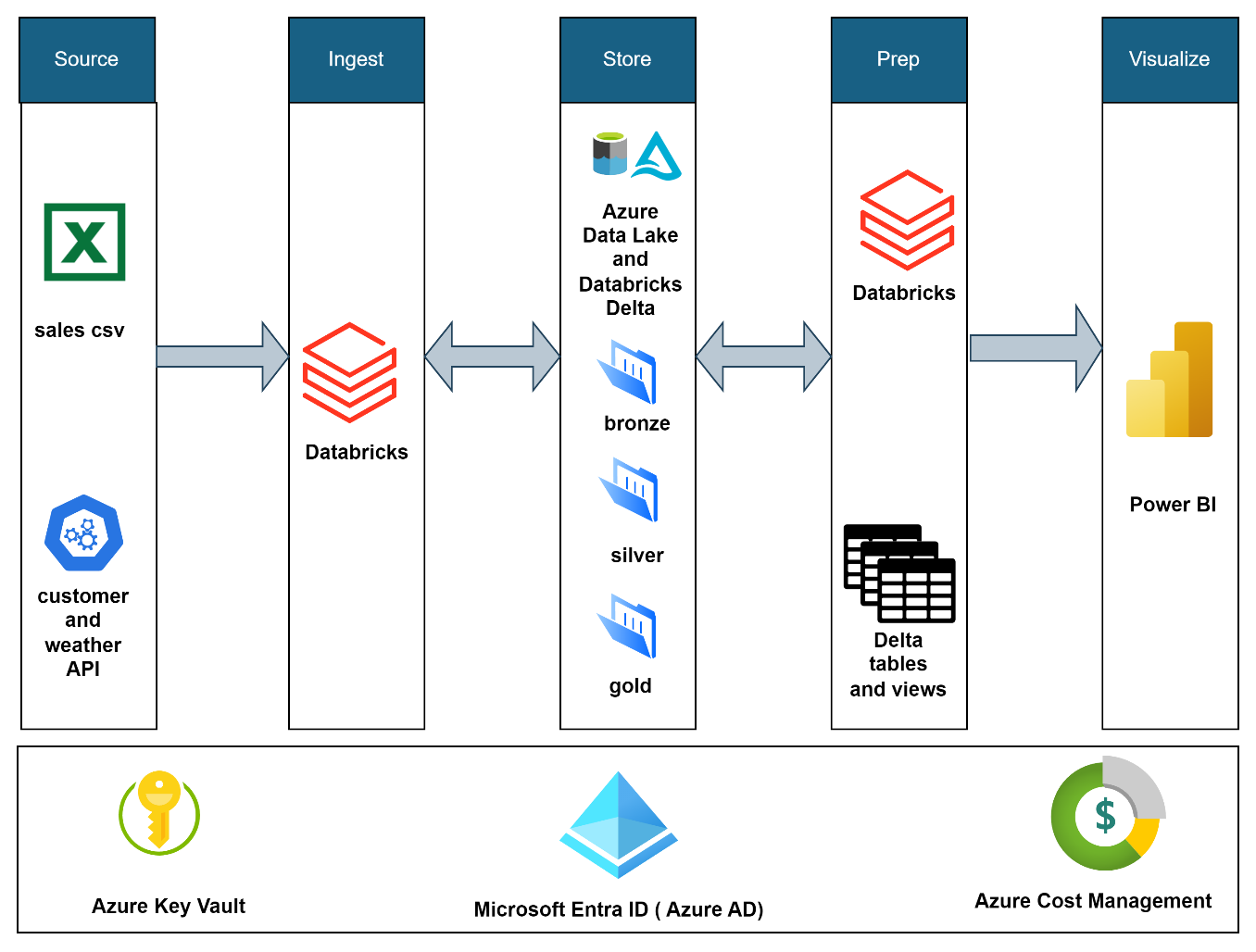
**AIQ Sales Analytics Assignment**

**Assignment Solution Architecture**

Data Ingestion, processing architecture is built on top of Azure Cloud. There are various data sources from which data is harvested, cleansed, transformed, and ingested in the Data Lake. It is using Azure Databricks Notebooks for data engineering activities, and orchestration of the data pipelines using databricks workflows.

This solution follows **Lake House** architecture which combines the best features from Datawarehouse and Data Lake

Lake house = Data **Lake** (low-cost storage) **+** Dataware**house** (relational attributes ACID)



**Notes**:

* Azure Key vault is a service used to store the credentials and enable us not to expose the passwords in our development code.
* In this assignment we have used this feature to store the weather API key
* Keys stored in this key vault will be accessed by creating **scoped credentials** in databricks

**Architecture Description**

**Source:**

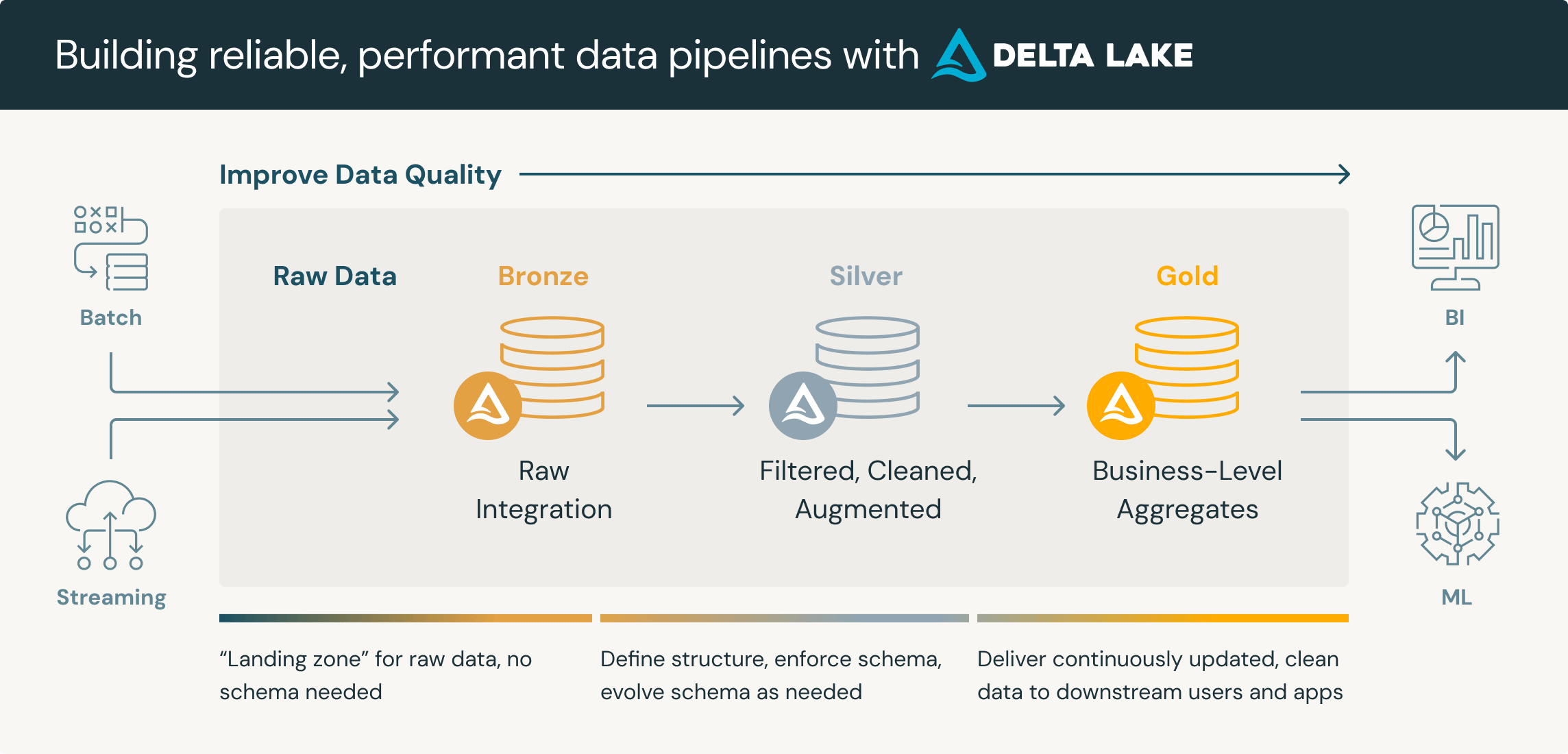
We have manual file and rest API sources; manual files will be placed under inbound folder of data lake (Refer Readme file step 1 to 12 for how to navigate and upload a file)

**Ingest:**

Ingestion has been taken care by databricks workflow job that runs the notebooks in it. Below table will explain the list of notebooks. (Refer Readme file step 14 to 27 for how to run a job)

**Storage:**

Data will be stored in 3 layers such as **bronze, silver and gold** in (**medallion architecture**) as below (image source https://www.databricks.com/glossary/medallion-architecture)



**Formats and load strategy followed in ingestion:**

Notebooks in each layer will be explained with their usage in later section.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No** | **Source** | **Layer** | **Format** | **Load Type** |
| 1 | Sales Data | Bronze | CSV | Full |
| 2 | Weather API | Bronze | JSON | Full |
| 3 | Customer API | Bronze | JSON | Full |
| 4 | Sales | Silver | Parquet | Increment |
| 5 | Customer and Weather | Silver | Parquet | Full |
| 6 | Sales | Gold | Delta Table (in Unity Catalog meta store) | Delta Append |
| 7 | Customer and Weather | Gold | Delta Table (in Unity Catalog meta store) | Delta Merge |

**Preparation:**

Final data preparations created as schematic views, will be explained in later sections.

**Visualization:**

Visuals have been created by Power BI along with a new date dimension added to enhance the analytics.

**Azure Data Lake Gen-2**

* A data lake is a single, centralized repository where you can store all your data, both structured and unstructured.
* Azure Data Lake Storage Gen2 is a set of capabilities dedicated to big data analytics, built on Azure Blob Storage.
* Data Lake Storage Gen2 provides file system semantics, file-level security, and scale. Because these capabilities are built on Blob storage, you also get low-cost, tiered storage, with high availability/disaster recovery capabilities.

**Databricks Delta Lake**

* Delta Lake is the optimized storage layer that provides the foundation for tables in a lake house on Databricks.
* Delta Lake is open-source software that extends Parquet data files (in Azure Data Lake) with a file-based transaction log for ACID transactions and scalable metadata handling.
* Delta Lake is fully compatible with Apache Spark APIs, and was developed for tight integration with Structured Streaming, allowing us to easily use a single copy of data for both batch and streaming operations and providing incremental processing at scale.

**Unity Catalog**

It’s a unified data governance model for Databricks lake house unified approach to **governance** accelerates data and AI initiatives while simplifying regulatory compliance.

**3 level namespaces in unity catalog**

A three-level namespace in Unity Catalogue consists of three levels of hierarchy — catalog, schema, and objects. Catalogs are like databases, schemas are like folders or directories, and objects can be tables, views, functions, or other entities.

* **Catalog**: halian\_aiq
* **Schema**: aiq\_sales
* **Objects**: Tables and views will be explained further in Ingestion process

All objects can be referred with **select \* from <catalog>.<schema>.<object>**

**ELT Config Table**

* **Object type: Table**
* **Name: aiq\_elt\_config**

It’s a table created to **drive the ELT data ingestion** flow it will store the following key details,

|  |  |  |
| --- | --- | --- |
| **S. No** | **Column** | **Usage** |
| 1 | source\_file\_name | Source file name in case of manual file processing |
| 2 | index\_if\_excel | Index value to refer sheet if the source is Excel file format |
| 3 | source\_query\_if\_db | Used to store SQL query to fetch data from database sources |
| 4 | last\_load\_date | A watermark column to be used for incremental load |
| 5 | endpoint\_if\_api | To store and pass API endpoint to the Notebook |
| 6 | api\_qry\_parm | To store and pass API query parameters to the Notebook |
| 7 | adls\_inbound\_path | Source path for bronze |
| 8 | adls\_path | Target path for bronze, silver, and gold |
| 9 | bronze\_nb | Bronze notebook name used as a filter to get the config details |
| 10 | silver\_nb | Silver notebook name used as a filter to get the config details |
| 11 | gold\_nb | Gold notebook name used as a filter to get the config details |

**ELT Audit Log Table**

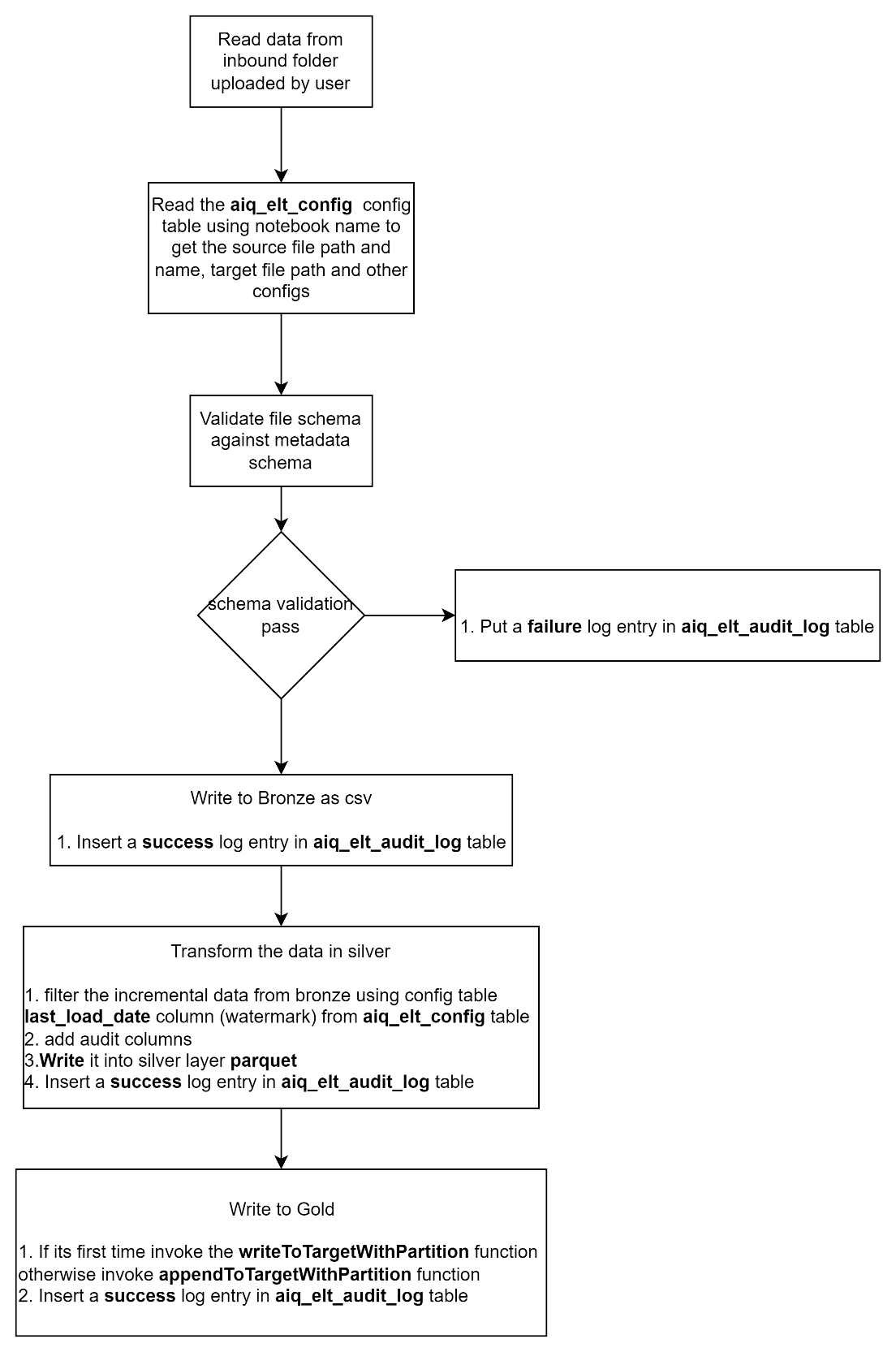
It is a table to store all notebook **execution logs.**

* **Object type: Table**
* **Name: aiq\_elt\_audit\_log**

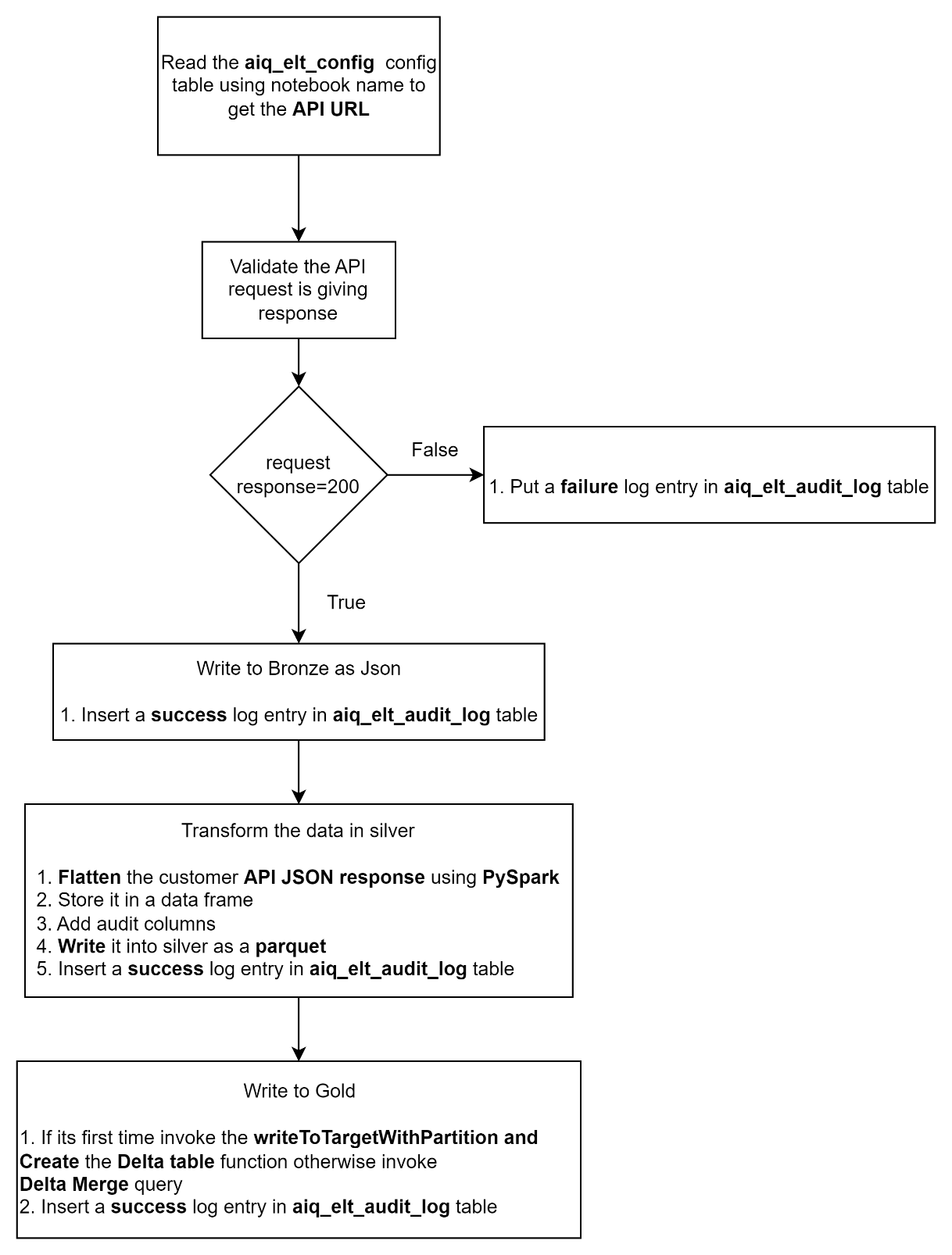
|  |  |  |
| --- | --- | --- |
| **S. No** | **Column** | **Usage** |
| 1 | project\_name | Project name of the notebook |
| 2 | file\_name | Source file it is processing |
| 3 | notebook\_name | Notebooks name |
| 4 | notebook\_path | Notebooks path in databricks workspace |
| 5 | notebook\_status | Execution status Success or Failure |
| 6 | zone | Notebook zone i.e., bronze, silver, and gold |
| 7 | start\_time | Notebook execution start time |
| 8 | end\_time | Notebook execution end time |
| 9 | log\_message | Log message of notebook executed |

**Data Ingestion Flow**

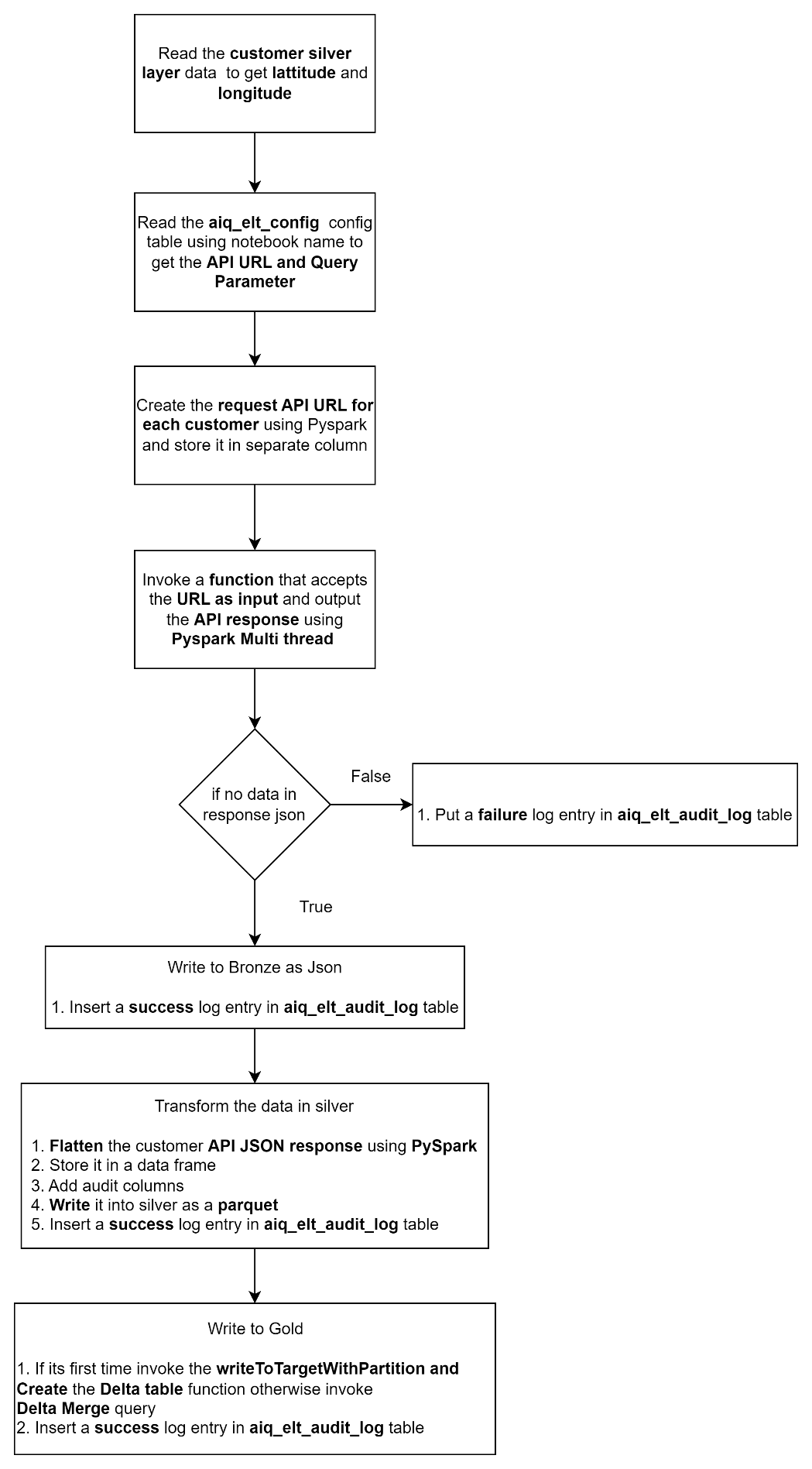
1. **Sales data flow**



1. **Customer data Flow**



1. **Weather data flow**

****

**Orchestrating the data flow**

Orchestrating the dataflow (bronze, silver, gold notebooks) has been done in databricks workflow (Refer Readme file step 14 to 27 for how to run a job) to execute them in the sequence it supposed to be. **A table below will map the layers mentioned in above flow in Data flow column.**



**Job name**: job\_aiq\_sales\_orchestration

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Flow** | **Job Task** | **Notebook Name** | **Notebook Path** |
| Customer | 1-customer\_bronze | aiq\_customer\_api\_bronze | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/bronze/aiq\_customer\_api\_bronze |
| Customer | 2-customer\_silver | aiq\_customer\_api\_silver | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/silver/aiq\_customer\_api\_silver |
| Customer | 3-customer\_gold | aiq\_customer\_api\_gold | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/gold/aiq\_customer\_api\_gold |
| Weather | 4-weather\_bronze | aiq\_weather\_api\_bronze | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/bronze/aiq\_weather\_api\_bronze |
| Weather | 5-weather\_silver | aiq\_weather\_api\_silver | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/silver/aiq\_weather\_api\_silver |
| Weather | 6-weater\_gold | aiq\_weather\_api\_gold | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/gold/aiq\_weather\_api\_gold |
| Sales | 7-sales\_bronze | aiq\_sales\_manual\_bronze | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/bronze/aiq\_sales\_manual\_bronze |
| Sales | 8-sales\_silver | aiq\_sales\_manual\_silver | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/silver/aiq\_sales\_manual\_silver |
| Sales | 9-sales\_gold | aiq\_sales\_manual\_gold | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/gold/aiq\_sales\_manual\_gold |

**Other Notebooks**

|  |  |  |
| --- | --- | --- |
| **Notebook Name** | **Purpose** | **Notebook Path** |
| aiq\_common\_functions | Notebook which contains config variables, path variables, credential secrets, common functions used in the assignment | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/lib/aiq\_common\_functions |
| aiq\_customer\_view | Schematic layer view preparation for customer dimension | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/view/aiq\_customer\_view |
| aiq\_sales\_view | Schematic layer view preparation for sales fact | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/view/aiq\_sales\_view |
| aiq\_consilidated\_sales\_dataset | Consolidated data of sales, customer, and weather dataset preparation | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/datasets/aiq\_consilidated\_sales\_dataset |
| aiq\_customer\_sales\_dataset | Consolidate customer and sales related analytics | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/datasets/aiq\_customer\_sales\_dataset |
| aiq\_customer\_weather\_sales\_dataset | Consolidate customer weather and sales related analytics | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/datasets/aiq\_customer\_weather\_sales\_dataset |
| aiq\_product\_sales\_dataset | Consolidate product and sales related analytics | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/datasets/aiq\_product\_sales\_dataset |
| aiq\_output | To View all dataset results as a select query output | /Workspace/Repos/aiq-sales-assignment/halian-aiq/aiq/output/aiq\_outputs |

**Meta store database objects list**

Please refer the below attachment for the list of objects (Tables and views).



**Features Implemented**

1. Modern Lakehouse architecture using Azure cloud (Databricks and Data Lake)
2. A key vault to manage credentials.
3. Unity catalog for the data governance (enables RLS and Object Level Security)
4. An ELT Config table a meta data driven framework that drives the ingestion.
5. An ELT log table to monitor the pipeline runs and logs.
6. Exception and error handling and logging
7. Email job notification from Databricks for success and failure (can be viewed in outlook.live.com link with the same credentials available in Readme doc)
8. For API source retry 2nd time in 1 min interval has been enabled
9. A new date dimension implemented in Power BI to analyze trends and growth (Time intelligence)

**Data Model**



**Data Model Relationship**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Source** | **Destination** | **Cardinality** |
| 1 | vw\_dim\_aiq\_customer\_weather | vw\_fct\_aiq\_sales | One to many (Single Direction) |

**Important Assumptions and Considerations**

1. Since it’s a trail account its valid only for 14 days and we have only 8 to 9 days to complete the validations
2. Both Azure portal and Databricks must be used with same credentials provided in Readme doc
3. Sales data pipeline has been designed to be executed in an incremental approach, if new file is going to be placed it should follow the same file name and schema otherwise job will be failed.
4. If there is a file name change that needs to be updated in aiq\_etl\_config table source\_file\_name column
5. Schema mismatch will be logged as failure job status in aiq\_elt\_audit\_log table for Sales.
6. Each run will update the aiq\_elt\_config table last\_load\_date column for sales data with max(order\_date) from current file – currently it is 2023-06-01.
7. Fact sales data has been portioned into year and month generated from order date column.
8. If new file contains records Order Date less than or equal to above last\_load\_date, they will not be loaded.
9. For weather data to get the weather information REST API Query parameter has been used from the API documentation, it accepts 2 query parameters latitude and longitude and the same has been configured in aiq\_elt\_config table, weather will be as on date and should be tested on same day
10. The latitude and longitude are sourced from customer API data.
11. Tried implementing the solution in Lakehouse architecture with Python, PySpark and Spark SQL languages to achieve distributed processing to the maximum extent possible.
12. Period wise trends have been analyzed in Power BI
13. For Github please use the same credential (Readme doc) in case of OTP is asked

**Results**

All results can be viewed in aiq\_output notebook by following the steps provided in **ReadmeToViewResult** document, and in **Power BI** also since all output views are added in there as a table.