# **Lakehouse Design Plan using Synapse**

Azure Service Identified - Dedicated SQL Pool in Synapse

**Why?**

- Horizontal Scale Out (Manual\Can be Automated using Function App)

- Compute can be paused and resumed based on use, which will save us the cost. This can be automated using the Function Apps

**Steps:**

1. Source can be DataLake or BLOB storage.
2. Using ADF or Pyspark or Spark SQL we can access the data source.
3. Transformation can be performed using ADF or Pyspark or Spark SQL.

Transformation at Lakehouse side for Data Warehouse

* 1. Having 3 level of data movement - LOAD, STAGE and FINAL
  2. LOAD layer import the raw transaction data into the LOAD table Ex: <table\_name>\_load
  3. Validation of data in the load table referencing the master table

Validation lists include:

* + - Record count check
    - Master Table Mapping
    - Validity of the reference
    - Reporting the unmapped records into new file for the recheck
  1. Validated records are pushed into the STAGE table.
  2. A count check is performed to decide where the LOAD table count match with STAGE.
  3. Mismatch of the count of record will be placed in bad folder create with daily date.
  4. Good Data will be loaded into FINAL layer FACT tables.
  5. Bad Data will be check on daily basis for the failure reason and fix and reprocess is next day or same day (depends on the aggregation)

1. Store the data in the Synapse Dedicated SQL Pool with 500DWU as the processing power to have Data Warehouse

**Things to be ready before we decide on the data loading into the above model:**

1. We have the list of required tables and its columns from the raw files.
2. Understand the table data and to have more atomic level of data (either by joining related with key columns)
3. Identify tables that undergo frequent update to split the column to a separate table using the third normal form.
4. Finalize the base data level Granularity for tables of interest.

# **Aggregated Data for Reporting**

Azure Service Identified - SQL Server

**Why?**

- This service supports more concurrent user sessions for the user to access the data via API. 32747 is the limit which is not available for other services.

**Advantages?**

- Cost effective when compared to SQL Dedicated Pool for frequent reads and writes

- Cost effective when compared to Document DB and Mongo DB

- This instance can be used for normal SQL checks

**Design**

1. Station Dimension Normalized Tables - Daily, Weekly and Monthly.
2. Battery Dimension Normalized Tables - Aggregation based on business needs.