Solution:

**Create/Configure Ecosystem**

The first thing we should do is to configure an environment. As we are getting the data in various formats and the data are being stored in local databases or files to our AWS storage (s3 bucket), we will first need to configure our AWS ecosystem in such a way that the AWS services can be integrated with the local system so that it can read the data.

*Data Ingestion*

* Create a storage in AWS like s3 bucket to store the original data. We can create folder that denotes input and output. Input folders would contain all the incoming data and the output folders would contain all the processed/transformed data. The requirements want us to publish the result data in the AWS Redshift however we can still save the output data after cleaning into s3 and only select the data that we need to be pushed to RedShift.
* Learn the types/format: From the requirements we can see that we have 7 datasets. 6 of them are .csv files and 1 on them is .json file. We have to create a script that reads the data in various files and formats. This script will be written in any IDE that can run Python and Spark. We will be using Python to import the spark and utilize spark to load the data into S3 using various file formats.
* We should also make sure on how we want to process the data either batch or streaming or both. This depends on the volume of data we are generating every second/minute/hour.

*Data Processing*

* After the data is in s3 buckets, we can now create our ETL pipeline.
* The ETL would consist of various data cleaning and transformation techniques. This process would deal with any missing values, any duplicates, necessary columns, etc.

*Data Warehousing*

* After, we have our processed data. We will create various tables with specific schema in our Redshift. We would probably have at least 7 tables in our redshift, and we will increase and decrease the tables and schema if necessary.
* Now, that we have our cleaned data and the organized location where we can store them, we can finally publish our clean data to the Redshift table. When all the required data are in Redshift, we can begin SQL to query and get answers to our requirements.

*Reporting*

* Moreover, redshift can also be integrated with various visualization tools for the visual representation of the data and easier understanding of the analysis.

# **Use Cases**

# Migrating Data from one ecosystem to another ecosystem

# When dealing with different data sources and formats and bring all data into one storage.

# Need to do some analysis on the past/historical data and the data are not organized properly.

**Database Design**

## Tables Metadata Info with Pk/FK relationship

Patient Table

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| patient\_id | INT | Primary key for the patient |
| patient\_name | VARCHAR | Name of the patient |
| patient\_gender | VARCHAR | Gender of the patient |
| patient\_birth\_date | DATE | Birth date of the patient |
| patient\_phone | VARCHAR | Phone number of the patient |
| disease\_name | VARCHAR | Name of the disease |
| city | VARCHAR | City where the patient is located |
| hospital\_id | INT | Foreign key referencing Hospitals |

Subscriber Table

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| sub\_id | INT | Primary key for the subscriber |
| first\_name | VARCHAR | First name of the subscriber |
| last\_name | VARCHAR | Last name of the subscriber |
| Street | VARCHAR | Street address of the subscriber |
| Birth\_date | DATE | Birth date of the subscriber |
| Gender | VARCHAR | Gender of the subscriber |
| Phone | VARCHAR | Phone number of the subscriber |
| Country | VARCHAR | Country of the subscriber |
| City | VARCHAR | City where the subscriber is located |
| Zip Code | VARCHAR | Zip code of the subscriber |
| Subgrp\_id | INT | Foreign key referencing Group\_subgroup |
| Elig\_ind | VARCHAR | Eligibility indicator for the subscriber |
| eff\_date | DATE | Effective date for the subscriber |
| term\_date | DATE | Termination date for the subscriber |

## ER diagram - *Optional*

# **Technologies and Platforms to be used in this solution**

Technologies and Platforms to be used are as follows:

* 1. Amazon Web Services Platform
     1. AWS S3
     2. AWS Redshift
  2. Programming/Scripting Languages
     1. Python
     2. PySpark
     3. SQL
  3. Reporting Services
     1. Databricks
  4. Project Management
     1. JIRA
  5. CI/CD
     1. Jenkins
     2. GitHub