**1. Solution - Explain your solution here in a step-by-step manner.**

**Step-by-Step Solution:**

1. **Data Ingestion:**
   * Upload the given sample data to AWS S3 in a folder named **input-data**.
2. **Data Cleaning:**
   * Check for null values in the dataset.
   * Count the total null values for each column.
   * Replace null values for specific columns with **NA**.
   * Check for duplicate records and drop duplicates if found.
   * Clean the data for the following datasets: Patients, Subscribers, Claims, and Group\_subgroup.
   * Upload the cleaned data into corresponding Redshift tables.
3. **Database Schema Design:**
   * Create schema design documents for target tables.
   * Define primary and foreign key relationships.
4. **Data Analysis and Result Creation:**
   * Create separate Redshift tables for each use case output in a Redshift schema **Project-Output**.
   * Implement data pipelines to perform the required analyses, such as:
     + Identifying the disease with the maximum number of claims.
     + Finding subscribers under 30 who subscribe to any subgroup.
     + Identifying the group with the maximum subgroups.
     + Determining the hospital serving the most patients.
     + Finding the most frequently subscribed subgroups.
     + Counting the total number of rejected claims.
     + Determining the city from which most claims are coming.
     + Identifying whether subscribers mostly subscribe to government or private policies.
     + Calculating the average monthly premium paid by subscribers.
     + Identifying the most profitable group.
     + Listing patients under 18 admitted for cancer.
     + Listing patients with cashless insurance and total charges >= Rs. 50,000.
     + Listing female patients over 40 who underwent knee surgery in the past year.
5. **Implementation and Testing:**
   * Use Databricks for implementation and testing.
   * Push the final code to GitHub.
   * Deploy the code on AWS EMR or Databricks with the help of GitHub.

**2. Use Cases - List down all the use cases on which this solution will be applicable.**

**Use Cases:**

1. Identifying the disease with the maximum number of claims.
2. Finding subscribers under 30 who subscribe to any subgroup.
3. Identifying the group with the maximum subgroups.
4. Determining the hospital serving the most patients.
5. Finding the most frequently subscribed subgroups.
6. Counting the total number of rejected claims.
7. Determining the city from which most claims are coming.
8. Identifying whether subscribers mostly subscribe to government or private policies.
9. Calculating the average monthly premium paid by subscribers.
10. Identifying the most profitable group.
11. Listing patients under 18 admitted for cancer.
12. Listing patients with cashless insurance and total charges >= Rs. 50,000.
13. Listing female patients over 40 who underwent knee surgery in the past year.

**3. Database Design - List down all possible DB (Redshift) tables here**

**a. Tables Metadata Info with PK/FK relationship:**

1. **Patients Table:**
   * **Columns:** PatientID (PK), Name, Age, Gender, Disease, HospitalID (FK), InsuranceType, TotalCharges, AdmitDate, DischargeDate.
2. **Subscribers Table:**
   * **Columns:** SubscriberID (PK), Name, Age, Gender, PolicyGroupID, SubgroupID (FK), City, State, Premium, SubscriptionDate.
3. **Claims Table:**
   * **Columns:** ClaimID (PK), PatientID (FK), SubscriberID (FK), Disease, ClaimAmount, ClaimStatus, ClaimDate.
4. **Group\_Subgroup Table:**
   * **Columns:** GroupID (PK), SubgroupID (PK), GroupName, SubgroupName.
5. **Hospitals Table:**
   * **Columns:** HospitalID (PK), HospitalName, Location, TotalPatientsServed.

**4. Technologies and Platforms to be used in this solution - List down the list of technologies like Spark, AWS, and Databricks, etc.**

**Technologies and Platforms:**

1. **AWS S3** for data storage.
2. **AWS Redshift** for data warehousing and analysis.
3. **Databricks** for data processing and data cleaning
4. **AWS EMR Studio** for managing big data frameworks.
5. **PySpark (optional)** for data cleaning and transformation.
6. **Jira** for project management and sprint planning.
7. **GitHub** for version control and code repository.
8. **PyCharm** for code development and deployment.