# Solution:

**Step 1: Data Ingestion**

* Upload raw datasets (Patients, Subscribers, Claims, Group\_Subgroup) to an S3 bucket (input-data folder).
* Read these datasets using PySpark on Databricks or AWS EMR.

**Step 2: Data Cleaning**

* Check and count null values for each column.
* Replace nulls with 'NA' or appropriate defaults.
* Remove duplicate records from each dataset.
* Ensure consistent data types and validate integrity across datasets.

**Step 3: Schema Design**

* Define Redshift schemas and create corresponding tables for Patients, Subscribers, Claims, and Group\_Subgroup.
* Design relationships using primary and foreign keys.

**Step 4: Data Transformation & Load**

* Apply business logic and transformations using PySpark.
* Load cleaned and transformed data into Redshift target tables using JDBC.

**Step 5: Use Case Query Execution**

* Write SQL queries in Redshift for each defined use case.
* Store results in dedicated Redshift tables under project\_output schema.

**Step 6: Visualization & Validation**

* Create snapshots or charts using Databricks for visual interpretation.
* Validate query outputs using test cases in JIRA.

**Step 7: Deployment**

* Push PySpark code and SQL scripts to GitHub.
* Use AWS EMR/Databricks jobs to automate pipeline execution using GitHub integration.

**Step 8: Project Tracking**

* Organize sprints and tasks in JIRA.
* Track documentation (Week 1) and development/testing (Week 2) separately.

# Use Cases:

* Identify the disease with the maximum number of claims.
* Find subscribers under age 30 who subscribe to any subgroup.
* Determine which group has the maximum number of subgroups.
* Identify the hospital that serves the most number of patients.
* Find the subgroup that is subscribed to most frequently.
* Count total number of rejected claims.
* Determine the city from where most claims are coming.
* Analyze which type of policy (Government or Private) is subscribed the most.
* Calculate the average monthly premium paid by subscribers.
* Identify the most profitable group.
* List patients under 18 admitted for cancer.
* List cashless insurance patients with charges ≥ ₹50,000.
* List female patients over age 40 who had knee surgery in the past year.

1. Database Design - List down all possible db(Redshift) tables here:
   * + 1. Claims
       2. Disease
       3. Group
       4. Sub\_Group
       5. Hospital
       6. Patient\_Records
       7. Grp\_Subgrp
       8. Subscriber

# Technologies and Platforms to be used in this solution:

* **AWS S3**: Storage layer for raw and cleaned datasets.
* **AWS Redshift**: Data warehouse for final structured and analytics-ready data.
* **AWS EMR Studio**: Cluster-based execution environment for PySpark jobs.
* **Databricks (Community Edition)**: Development, processing, and visualization environment.
* **PySpark**: Data cleaning, transformation, and loading tool.
* **JIRA**: Task management and sprint planning.
* **GitHub**: Source code version control and deployment integration.
* **JDBC**: Connector for PySpark to communicate with Redshift.