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

The solution involves creating a robust data pipeline to analyze competitors' data and customer behaviors for a Health Care insurance company. This pipeline will leverage AWS services, Databricks, and PySpark to ingest, clean, store, and analyze large datasets. Here is the step-by-step solution:

1.Data Ingestion:

Collect data from various sources, including web scraping and third-party APIs.

Store raw data in AWS S3 in a folder named input-data.

1. Data Cleaning:

Null Values Handling:

* Check for null values in the dataset.
* Count total null values for each column.
* Replace null values with 'NA' for specific columns.

Duplicates Handling:

* Check for duplicate records.
* Remove duplicates from the dataset.
* Clean data for the following datasets:

Patients

Subscriber

Claims

Group\_subgroup

1. Data Storage:

Upload cleaned data to AWS Redshift tables.

Create schema and table design for storing cleaned data.

1. Data Analysis:

Create Redshift tables to store analysis results for each use case.

Write PySpark scripts to perform the required analysis and generate insights

.

1. Reporting and Visualization:

Use Databricks for data processing and visualization.

Create dashboards and visualizations to display key metrics and insights.

1. Deployment:

Deploy the PySpark scripts on AWS EMR or Databricks.

Use GitHub for version control and collaboration.

Implement a CI/CD pipeline to automate the deployment process.

1. Documentation:

Create comprehensive documentation for requirements, solution design, and database schema.

Maintain a Jira board for tracking user stories, tasks, and project progress.

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

The solution will address the following use cases:

* Identify the disease with the maximum number of claims.
* Find subscribers aged less than 30 who have subscribed to any subgroup.
* Determine which group has the maximum subgroups.
* Identify the hospital serving the most number of patients.
* Determine which subgroups are subscribed to the most number of times.
* Find the total number of rejected claims.
* Identify the city from where most claims are coming.
* Determine whether subscribers mostly subscribe to government or private policies.
* Calculate the average monthly premium paid by subscribers to the insurance company.
* Identify the most profitable group.
* List all patients below the age of 18 admitted for cancer.
* List patients with cashless insurance having total charges greater than or equal to Rs. 50,000.
* List female patients over the age of 40 who have undergone knee surgery in the past year.

1. Database Design - List down all possible db(Redshift) tables here

## Tables Metadata Info with Pk/FK relationship –

* 1. Tables Metadata Info with PK/FK Relationships

1. Disease

Columns:

Disease\_ID (PK)

Disease\_name

Relationships:

Linked to Claims and Patient\_Records via Disease\_ID

2. GrpSubgrp

Columns:

Grp\_ID (PK)

SubGrp\_ID (PK)

Grp\_Name

Grp\_Type

SubGrp\_Name

Monthly\_Premium

Relationships:

Linked to Subscriber via SubGrp\_ID

Linked to Group via Grp\_ID

Linked to Subgroup via SubGrp\_ID

3. Claims

Columns:

Claim\_ID (PK)

Patient\_ID (FK)

Sub\_ID (FK)

Disease\_ID (FK)

Claim\_or\_Rejected

Claim\_Type

Claim\_Amount

Claim\_Date

Relationships:

Linked to Patients via Patient\_ID

Linked to Subscriber via Sub\_ID

Linked to Disease via Disease\_ID

4. Group

Columns:

Grp\_ID (PK)

Grp\_Name

Grp\_Type

Relationships:

Linked to GrpSubgrp via Grp\_ID

5. Hospital

Columns:

Hospital\_ID (PK)

Hospital\_name

City

State

Country

NumberOfPatients

Relationships:

Linked to Patients indirectly through Claims

6. Patient\_Records

Columns:

Patient\_ID (PK)

Patient\_name

Patient\_Gender

Patient\_Birth\_Date

Patient\_Phone

Disease\_ID (FK)

City

Hospital\_ID (FK)

SubGrp\_ID (FK)

Relationships:

Linked to Claims via Patient\_ID

Linked to Disease via Disease\_ID

Linked to Hospital via Hospital\_ID

Linked to Subgroup via SubGrp\_ID

7. Subscriber

Columns:

Sub\_ID (PK)

First\_name

Last\_name

Street

Birth\_date

Gender

Phone

Country

City

Zip\_Code

SubGrp\_ID (FK)

Elig\_ind

Eff\_date

Term\_date

Relationships:

Linked to GrpSubgrp via SubGrp\_ID

Linked to Claims via Sub\_ID

8. Subgroup

Columns:

SubGrp\_ID (PK)

SubGrp\_Name

Relationships:

Linked to GrpSubgrp via SubGrp\_ID

Linked to Subscriber via SubGrp\_ID

## ER diagram - *Optional*

# Technologies and Platforms to be used in this solution -*List down list of technologies like spark, aws and databricks etc.*

## AWS S3: For storing raw and cleaned data.

## AWS Redshift: For storing and querying cleaned data and analysis results.

## Databricks: For data processing, analysis, and visualization.

## AWS EMR Studio: For managing EMR clusters and running PySpark jobs.

## PySpark: For data cleaning and analysis scripts.

## Jira: For project management and tracking user stories.

## GitHub: For version control and collaboration.

## APIs and Web Scraping Tools: For data ingestion from various sources.