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

S1: Upload all the raw datasets on AWS S3 named “input-data”

S1.1 Use your AWS IAM account credentials to access the account.

S1.2 Create a directory “input-data” in S3.

S1.3 Upload all the datasets in that directory.

S2: Data cleaning activity using pyspark in databricks

Use your credentials to access azure account then open the databricks studio.

S2.1 Data cleaning activity for **patient\_records** dataset:

* First check all the columns if there are any null values or not.
* Count the total number of Null values for each column.
* And then replace the null values for specific columns with NA.
* After that, check If there are any duplicate records in Patient\_id column.
* If there are duplicates, then drop duplicates.

S2.2 Data cleaning activity for **subscriber** dataset:

* First check all the columns if there are any null values or not.
* Count the total number of Null values for each column.
* And then replace the null values for specific columns with NA.
* After that, check If there are any duplicate records in sub\_id column.
* If there are duplicates, then drop duplicates.

S2.3 Data cleaning activity for **claims** dataset:

* First check all the columns if there are any null values or not.
* Count the total number of Null values for each column.
* And then replace the null values for specific columns with NA.
* After that, check If there are any duplicate records in claim\_id column.
* If there are duplicates, then drop duplicates.

S2.4 Data cleaning activity for **group** dataset:

* First check all the columns if there are any null values or not.
* Count the total number of Null values for each column.
* And then replace the null values for specific columns with NA.
* After that, check If there are any duplicate records in group\_id column.
* If there are duplicates, then drop duplicates.

S2.5 Data cleaning activity for **subgroup** dataset:

* First check all the columns if there are any null values or not.
* Count the total number of Null values for each column.
* And then replace the null values for specific columns with NA.
* After that, check If there are any duplicate records in subgroup\_id column.
* If there are duplicates, then drop duplicates.

S3: Upload cleaned data corresponding to each dataset from databricks to redshift table. For that you must use the redshift database name, username, password, Redshift write access authority.

S4: Create a separate schema design document for all target tables.

S5: Analyse the data according to use cases using pyspark in databricks and create a separate redshift table for each use case output in a redshift schema named “Project-Output”.

S5.1 Open the Redshift Editor

S5.2 Create the schema “Project-Output”

S5.3 Write the analysed data from azure databricks to redshift table

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

* 1. The disease which has the maximum number of claims.
  2. Subscribers whose age is less than 30 and they have subscribed any subgroup.
  3. Group which has maximum number of subgroups.
  4. Hospitals which serve the greatest number of patients
  5. The subgroups which are subscribed maximum number of times.
  6. The total number of claims which were rejected.
  7. The city which has the maximum number of claims
  8. Groups of policies subscriber that subscribe mostly Government or private.
  9. Monthly average premium subscribers are paying to insurance company.
  10. The most profitable group
  11. All the patients who are below age of 18 and are admitted for cancer.
  12. All the patients who have cashless insurance and have total charges greater than or equal to Rs. 50,000.
  13. All the female patients who are over the age of 40 and 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 –

3.1 Patients\_records table columns:

patient\_id (PK), patient\_name, patient\_gender, patient\_birth\_date, patient\_phone, disease\_name, city, hospital\_id (FK)

3.2 subscriber table columns:

Sub\_id (PK), first\_name, last\_name, street, birth\_date, gender, phone, Country, City, Zipcode, Subgrp\_id (FK), Elig\_ind, eff\_date, term\_date

3.3 claims table columns:

claim\_id (PK), patient\_id (FK), disease\_name, sub\_id (FK), claim\_or\_rejected, claim\_type, claim\_amount, claim\_date

3.4 hospital table columns:

hospital\_id (PK), hospital\_name, city, state, country

3.5 group table columns:

Country, premium\_written, zipcode, grp\_id (PK), grp\_name, grp\_type, city, year

3.6 subgroup table columns:

SubGrp\_id (PK), SubGrp\_Name, Monthly\_Premium

3.7 disease table columns:

SubGrpID (FK), Disease\_ID (PK), Disease\_name

## 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
* AWS Redshift
* Databricks
* AWS EMR Studio
* PySpark
* Jira
* GitHub

## 