# Solution -

# *Assumption: The data types and structure of the tables or datasets released will be constant.*

8 tables are present: claims, disease, group, grpsubgrp, hospital, Patient\_records, subgroup, subscriber. (In csv and json formats)

For setting up the pipeline, these needs to be cleaned: Patients, Subscriber, Claims, Group\_subgroup, using databricks.

The raw data are will be stored in S3.

They will be called for staging in databricks and cleaned. The cleaned data will be stored in S3 for staging.

Finally, we will perform analysis on the cleaned data and store the result table on AWS Redshift.

Database design and metadata will be done in Redshift.

Cleaned, Validated and Analyzed data corresponding to each data set stored into individual redshift table.

### A separate redshift table for each use case output in a redshift schema

* + - Schema = Project-Output

Needs proper connection (userID, password and URL/URI connectors)

* + Cleaning Activity
    - First check if there are null values in dataset
    - Count the total Null values for each column
    - And then replace the null values for specific columns by NA
    - Check the If three are duplicates records
    - If there are duplicates then drop duplicates
  + Clean data for at least for following datasets
    - Patients
    - Subscriber
    - Claims
    - Group\_subgroup

After cleaning the datasets, they were stored in s3 bucket in folders as per their name. However, when calling the tables from those folders, they lost their headers.

1. Use Cases -

* Which disease has a maximum number of claims.

Disease table, claims table, joins on disease\_name,

count the number of claim\_id where disease\_name is same in both tables

* Find those Subscribers having age less than 30 and they subscribe any subgroup

Subscriber table, Birth\_date (calculation and < 30) Subgrp\_id

Subgroup table, SubGrp\_id

Joins on SubGrp\_id

* Find out which group has maximum subgroups.

Grpsubgrp table, count how many times each grp\_Id appears

* Find out hospital which serve most number of patients.

Patient\_records table, hospital table. Joins on hospital\_id (Hospital\_id). First count the sum of Patient\_id per hospital\_id, then get the name ‘Hospital\_name’ associated with ‘Hospital\_id’.

* Find out which subgroups subscribe most number of times.

subgroup table, subscriber table. Joins on Subgrp\_id. Get the subGrp\_id, SubGrp\_Name, and sub\_id, Birth\_date

* Find out total number of claims which were rejected

claims table; sum of Y in ‘Claim\_Or\_Rejected’ column.

* From where most claims are coming (city)

claims table, subscriber table, joins on SUB\_ID(sub\_id), get the unique count of city and sum how many times they appear in subscriber table.

* Which groups of policies subscriber subscribe mostly Government or private (group type)

Grp\_ID

group table, grpsubgrp table, joins on Grp\_Id

subscriber table, grpsubgrp table, joins on Subgrp\_id (SubGrp\_ID)

find out which sub\_id belongs to which SubGrp\_ID map those SubGrp\_ID to Grp\_Id

Map the Grp\_Id to Grp\_Type.

* Average monthly premium subscriber pay to insurance company.

subgroup table, subscriber table, joins on SubGrp\_id (Subgrp\_id)

sum the count of sub\_id

divide by the total Monthly\_Premium

* Find out Which group is most profitable

List the SubGrp\_id and then sum the Monthly\_Premium related to the respective SubGrp\_id.

The SubGrp\_id paying the most Monthly\_Premium is the most profitable.

And/Or

Find out the most rejected claims, map the sub\_id to the subgrp\_id, calculate the monthly\_premium paid by those subgrp\_id (s). Now compare.

* List all the patients below age of 18 who admit for cancer.

Patient\_records table, calculate age based on birth\_date ( age < 18 and disease =’cancer’)

* List patients who have cashless insurance and have total charges greater than or equal for Rs. 50,000.

Claims table, claim\_type

Patient\_records table

* List female patients over the age of 40 that 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 -

## A diagram of a computer Description automatically generated ER diagram – *Optional*

# Technologies and Platforms to be used in this solution

# *AWS S3, AWS Redshift, PySpark, Databricks (*14.3 LTS (includes Apache Spark 3.5.0, Scala 2.12) *and SQL.*

In Databricks:  
##### df\_dis = disease

##### df\_grp = group

##### df\_gs = grpsubgrp

##### df\_hp = hospital

##### df\_pr = patient\_records

##### df\_sg = subgroup

##### df\_sb = subcriber

##### df\_cl = claims