# **Solution** –

Below is the step-by-step process for the solution.

* 1. Data Integration:
* Extracting data from the sources where customers claim,patient data, disease and subscriber along competitor data is located.
* Loading raw data into **AWS s3** in CSV/Parquet format.
  1. Data Cleaning and transformation:
* There is more possibility of having duplicate records or some incomplete data, so checking null data, counting the null values in specific column and replacing those null with ”NA” checking duplicates records and deleting is the part of requirement for data cleaning. Using **pyspark** to clean and preprocess data.
  1. Load into Redshift table:
* Design and create table in **redshift** for data storage.
* Design the partition and any key relationships as necessary.

# **Use Cases**- I have last 3 requirements picked. On bases of that I have below use cases.

**Requirement:**

* List all the patients below age of 18 who admit for cancer
* List patients who have cashless insurance and have total charges greater than or equal for Rs. 50,000.
* List female patients over the age of 40 that have undergone knee surgery in the past year.

**Use Cases:**

* Identifying and listing all patients below age of 18 who admit for cancer.

Looking into patient table with filter or disease\_name like”cancer” and calculating age from DOB.

* Identifying patients with cashless insurance having total charges of rs.50000 or more.

Looking into claims table to find the claims amount more the 50k

* Listing female patients over 40 who have undergone knee surgery in the past year.

Looking into patients table to filter for gender as Female and selecting on knee surgery list in past year.

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

## Tables Metadata Info with Pk/FK relationship –

***Patients Table: Column names***

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

***Claims Table: Column Names*** claim\_id(pk),patient\_id(fk),disease\_name,Galactosemia,SUB\_ID,Claim\_Or\_Rejected,claim\_type,claim\_amount,claim\_date

***Disease table: Column Names***

SubGrpID(fk), Disease\_ID(pk),Disease\_name

***Subscriber table:column name***

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

## ER diagram – *Optional*

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

*Claims Table: Column Names*  claim\_id(pk),patient\_id(fk),disease\_name,Galactosemia,SUB\_ID,Claim\_Or\_Rejected,claim\_type,claim\_amount,claim\_date

*Disease table: Column Names*

SubGrpID, Disease\_ID,Disease\_name

Disease id

Subgrp\_id

Claim\_ID

Patient\_id

Patient ID

SsS

Sub\_id

Disease\_table

Patient\_table

Sub\_id

Subgrp\_id

Claims\_table

Subscriber\_table

# Technologies and Platforms to be used in this solution:

List of tools use in this solution:

* **AWS S3**: Storage of raw and processed data.
* **AWS Redshift**: Data warehouse for structured data and analytics.
* **Databricks**: Data transformation and cleaning using PySpark.
* **PySpark**: Data cleaning, transformation, and ETL processing.
* **Jira**: Project tracking and management.
* **Github**: Project collaboration