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

1. Upload all the files in S3.
2. Create a new bucket(healthInsuranceBucket) to store your data.
3. Create a folder(uncleansedData) and Upload all the provided data files to the selected bucket.
4. Cleanse the data using Pyspark
5. Access the PySpark environment(PyCharm) and establish a connection to AWS S3.
6. Load each data file from the "uncleansedData" folder into a PySpark DataFrame.
7. Let’s take the patient’s table as an example to cleanse the data: check null values in the data set for the patients and after that count those values and replace all the null values with NA for specific columns. Convert date columns to appropriate date formats.
8. Save the cleaned Data Frame to redshift in an output table for further analysis.
9. Data Analysis for use case results
10. Load the cleaned data into a new Spark Data Frame.
11. Start performing the specific analysis and calculations based on the functional requirements. Calculate statistics like maximum, minimum, average, etc. Group data and aggregate information. Filter data based on specific conditions.
12. Let’s take use case “i”, here we must find out which disease has the maximum number of claims. For this, we must use the disease table and claims table and join them.

(SELECT disease\_name, count(claim\_id) as max\_number\_of\_claims

FROM disease

JOIN claims ON disease.disease\_name = claims.disease\_name

GROUP BY claim.disease\_name

ORDER BY max\_number\_of\_claim DESC

LIMIT 1;)

1. Load all the results obtained from data analysis into Redshift tables.
2. Create Redshift tables to store the analysis results.
3. Load the results obtained from data analysis into the corresponding Redshift tables.

For example, the result for the use case “i” will be a table

CREATE TABLE DiseaseClaimAnalysis (

disease\_name VARCHAR(100),

max\_number\_of\_claims INT);

1. Use Cases –
2. Which disease has the maximum number of claims?
3. Find those Subscribers having aged less than 30 and they subscribe to any subgroup
4. Find out which group has the maximum subgroups.
5. Find out the hospital which serves most number of patients
6. Find out which subgroups subscribe the most number of times
7. Find out the total number of claims which were rejected
8. From where most claims are coming (city)
9. Which groups of policies subscribers subscribe mostly Government or private
10. Average monthly premium subscriber pay to the insurance company.
11. Find out Which group is most profitable.
12. List all the patients below the age of 18 who admit to cancer.
13. List patients who have cashless insurance and have total charges greater than or equal to Rs. 50,000.
14. List female patients over the age of 40 that have undergone knee surgery in the past year
15. Database Design - List down all possible db(Redshift) tables here

## Tables Metadata Info with Pk/FK relationship –

For claim table

|  |  |
| --- | --- |
| **Column** | **Data Type** |
| Claim\_id(PK) | INT |
| Patient\_id(FK) | INT |
| Disease\_name | Varchar(100) |
| Sub\_id(Fk) | INT |
| Claim\_or\_rejected | Varchar(20) |
| Claim\_type | Varchar(20) |
| Claim\_amount | Decimal(10,2) |
| Claim\_date | Date |

Luicd Diagram

A diagram of a patient

Description automatically generated

**Output tables**

1. Disease\_name,max\_number\_of\_claims
2. Subscribers, age\_below\_30
3. Subgroups, max\_subgroup
4. Hospital\_name, max\_number\_of\_patients
5. Subgroups, subscribe\_most\_number\_times

# 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