# Solution :

* ***Creating S3 bucket – with bucket name bootcamp/capstone\_project1/input***
* ***Creating individual folder for individual dataset***
  + 1. bootcamp/capstone\_project1/input/Claims
    2. bootcamp/capstone\_project1/input/ Disease
    3. bootcamp/capstone\_project1/input/Group
    4. bootcamp/capstone\_project1/input/Grpsubgrp
    5. bootcamp/capstone\_project1/input/Hospital
    6. bootcamp/capstone\_project1/input/Patient\_records
    7. bootcamp/capstone\_project1/input/Subgroup
    8. bootcamp/capstone\_project1/input/Subscriber
* ***Uploading the dataset with their corresponding folder***
* ***Writing the data files from s3 to the Databriks using Access and Secrete Keys of AWS IAM role***
* ***Changing into data frame***
* ***Writing into a datalake temporary folder***
* ***Reading from datalake***
* ***Cleaning data***
  + 1. Write program to count null, NaN, spaces for each column
    2. Fill null values with NA using fillna
    3. Check if schema has any spaces on their names
    4. If schema has spaces, then remove it by renaming
    5. Check duplicates and drop it
* ***Changing the database into table using temporary view***
* ***Performing query to select and join according to the use case***

***Use Case-I***

Solution: Use only Claims

Count(Patiant\_id), group by (disease\_name), when count is max

***Use Case-II***

Solution: Use Subscribers, change the null values as Not

Select subscriber\_name when dob >1997 and subgroup\_id!==No

***Use Case-III***

Solution: Use group and grpsubgrp

Join on: Grp\_Id

Count(SubGrp), group by (Grp\_name), print when count is max

***Use Case-IV***

Solution: Use hospital and Patient\_records

Join on: City

Count(Patient\_id), group by (Patient\_id), print when count is max

***Use Case-V***

Solution: Use Subscriber and subgroup

Join on: subgroup\_id

Count(sub\_id), group by (subsgroup\_id), print when count is max

***Use Case-VI***

Solution: Use only Claim

count(claim\_id), where Claim\_Or\_Rejected='N'

***Use Case-VII***

Solution: Use Claims and Patient\_records

Join on:Patient\_id

count(claim\_id), group by city, print when count is max

***Use Case-VIII***

* ***Uploading final dimensional table into redshift as schema=Project-Output***

df.write.format("redshift").option("url"," ").\

option("dbtable", "Project\_Output.databriks\_capstone\_project1").\

option("aws\_iam\_role", " ").\

option("driver","com.amazon.redshift.jdbc42.Driver").\

option("tempdir", "s3a://xxx/tmpdir/").\

option("user", " ").\

option("password", " ").save()

# Use Cases :

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

## Tables Metadata Info with Pk/FK relationship:

A black background with white rectangles

Description automatically generated

## ER diagram - *Optional*

# Technologies and Platforms to be used in this solution:

1. AWS S3
2. Draw.io for flow chart
3. Databricks
4. Pyspark
5. AWS EMR Studio
6. AWS Redshift
7. GitHub
8. Jira