Solution

1.Uploaded all given files into the S3 bucket inside the folder called data-input.

2.Created a cluster in Databricks and initiated work in the notebook.

3.Established a connection between Databricks and S3 using IAM-generated secret and access keys.

4.Created a DataFrame to read files from the S3 bucket, specifying the first line as a header.

5. Utilized DataFrames to perform various data transformations, including filter functions and counts.

6. Concluded by writing the resulting table into Redshift using df.write.format("redshift"), providing the necessary credentials.

Use Cases

Write sql for the functional requirements. Or which sql keywords you use?

1. Which disease has a maximum number of claims.

```
SELECT disease_name, COUNT(disease_name) AS dis_count

FROM claim

WHERE claim_id IS NOT NULL

GROUP BY disease_name

ORDER BY dis_count DESC

LIMIT 1
```

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

```
SELECT subs.first_name, subs.last_name , subs.Birth_date

FROM subs

INNER JOIN subgroup ON subs.Subgrp_id = subgroup.Subgrp_id
```

```
AND DATEDIFF(CURRENT_DATE(), CAST(subs.Birth_date AS DATE)) / 365.25 < 30
```

3. Find out which group has maximum subgroups.

```
SELECT SubGrp_ID, COUNT(*) AS Group_count

FROM subs

GROUP BY SubGrp_Id

ORDER BY Group_count DESC

LIMIT 1
```

4. Find out hospital which serve most number of patients.

```
SELECT h.Hospital_name, COUNT(h.Hospital_name) AS hos_count

FROM hospital h

INNER JOIN patient p ON p.Hospital_id = h.Hospital_id

GROUP BY h.Hospital_name

ORDER BY hos_count DESC

LIMIT 1
```

5. Find out which subgroups subscribe most number of times.

6. Find out total number of claims which were rejected.

```
df = spark.read.json("s3://merocapstonebucket/input-data/claims/claims.json")

df2 = df.filter("Claim_or_Rejected == 'Y'") df3 = df2.count()

print("The total numner of claims which were rejected were", df3,".")
```

7. From where most claims are coming (city).

```
cf=spark.read.option('header','True').csv("s3://merocapstonebucket/input-data/patient_records/Patient_records.csv")

cf2 = cf.groupBy("city").count()

cf3 = cf2.orderBy("count", ascending=False)
```

first = cf3.first() print(first)

8. Which groups of policies subscriber subscribe mostly Government or private.

```
SELECT *

FROM govtopri

WHERE Grp_Type LIKE 'Gov%'

ORDER BY Grp_Name
```

9. Average monthly premium subscriber pay to insurance company.

```
SELECT AVG(Monthly_Premium) AS Monthly_Premium

FROM subg
```

10. Find out Which group is most profitable .

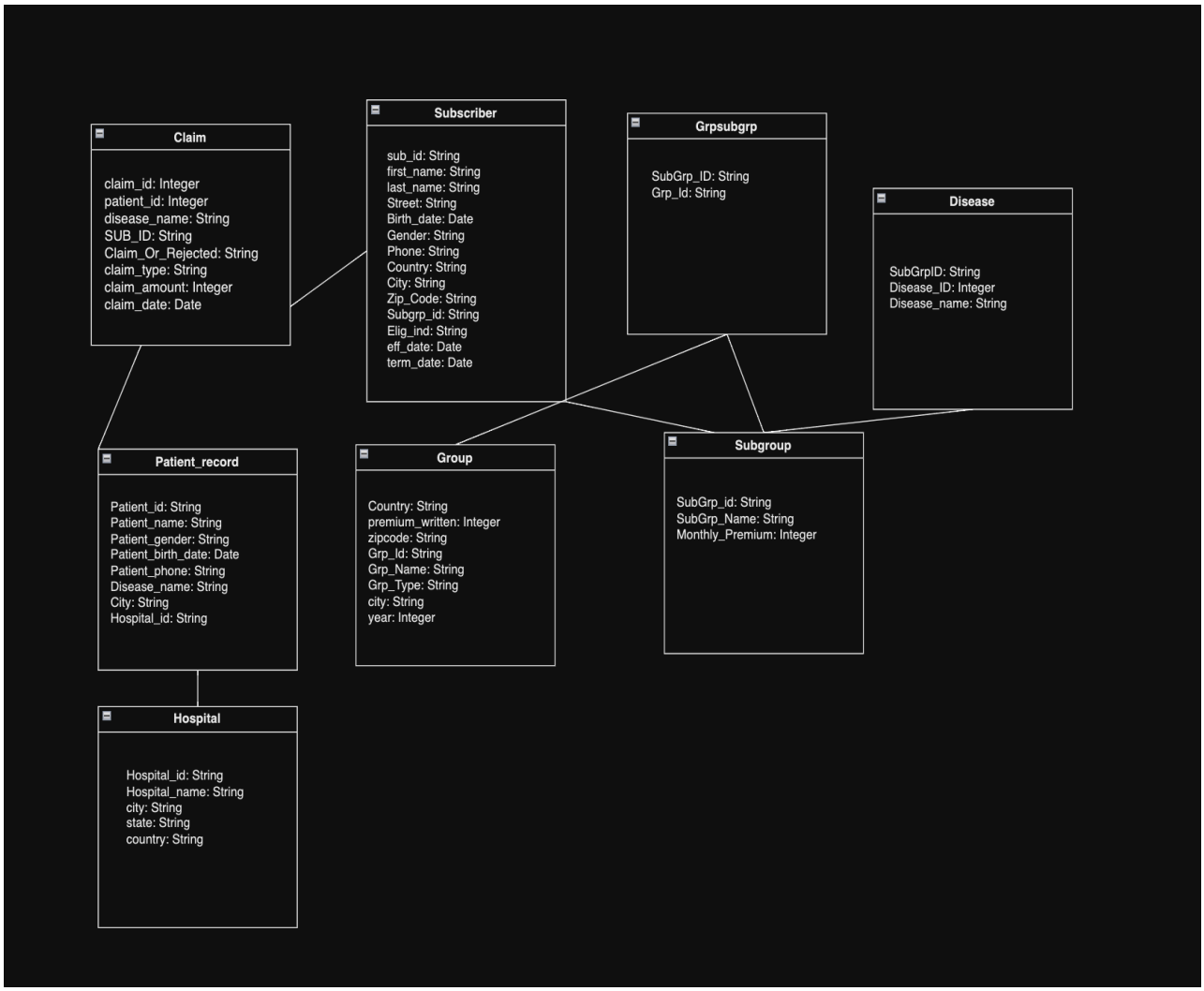
11. List all the patients below age of 18 who admit for cancer .

```
SELECT * FROM patient_record

WHERE YEAR('2024-01-20') - YEAR(patient_birth_date) >= 18

AND disease_name LIKE '% 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.

Database_Design



Technologies and Platforms to be used in this solution -*List down list of technologies like spark, aws and databricks etc.*

Databricks, Redshift, AWS S3, Pyspark