

HOSPITAL CLAIM MANAGEMENT SYSTEM ANALYSIS

Created by Batch 75

Cohort Code: INTCDB22DW075



Contents

1. Business Challenge / Requirement	3
2. Goal of the project	3
3. Project Architecture.....	3
4. Dataset Explanation and Schema	5
5. Problem Statement / Tasks.....	9
6. Code Templates.....	9
▪ Data Processing.....	9
▪ Conversion of Raw data into Processed Data.....	11
▪ Importing processed data to MYSQL.....	13
▪ Data Transformation	21
▪ Sqoop and Hive and HDFS.....	25
▪ Final processing from data.....	27
7. Output Screen.....	29
8. Business Benefits/ Conclusion.....	34
9. Further Enhancements/Recommendations	34
10. References / Bibliography.....	34

❖ Business Challenge / Requirement

This is a Hospital Claim Management System which has many databases stored like Patient information, Claim information, Insurance Company Names, Hospital Details etc. So, to correctly manipulate and analyse this huge sets of data, the authority has requested for our help in Big Data Analytics, for handling and analysing of this datasets.

❖ Goal of the Project

To meet our client requirements, we have to use many databases which store patient information, claim information, insurance company names etc. We need to do data cleansing to fix errors, duplicates and irrelevant data from the raw dataset. We have processed, filtered, transformed and analysed these data. Finally, we will plot a graph of 6months claim of patient and claims based on disease.

❖ Project Architecture

1. A linux file server receives 14 files in total in Comma Separated Value (CSV) format from the Hospital, Insurance Company and other sources. These are the source files that we will be using for our project.
2. Then we use data filtering in python to filter out NULL and invalid values from our data sources.
3. Now transformational logic is applied on the cleansed files and the data are imported to MYSQL for storage.
4. The file data and tables are validated, enriched, analysed, processed before loading into HDFS and HIVE.
5. Finally, after analysing of the tables and datasets, we are able to visualize the graph we need to get the relation between insurance claims and age group.

HOSPITAL CLAIM MANAGEMENT SYSTEM ANALYSIS ARCHITECTURE

Data Sources

Patient.csv

Claim.csv

Insurer.csv

Hospitals.csv

Group.csv

SubGroup.csv

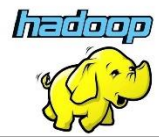
Data Preprocessing

Linux

Data Cleaning

Data Transformation

Store, Retrieve and Manage Data



Data Ingestion Services



Raw Data

Data
Preprocessing

Matplotlib
Pandas
Seaborn

Final Output

Data Storage



Data Processing



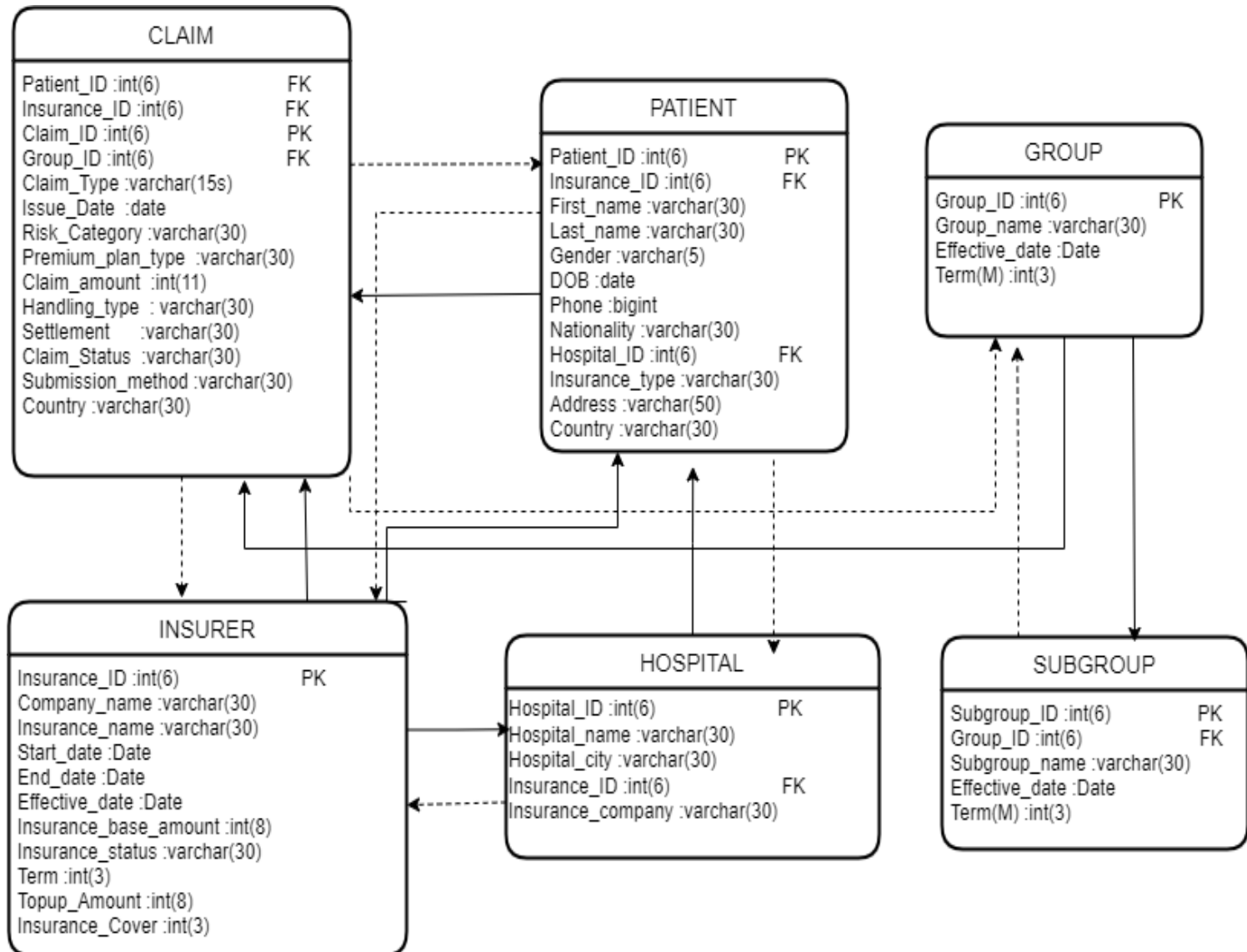
Analysed Data

1. Patient claims based on past 6 months
2. Patient claims based on disease

❖ Datasets and Schemas

1. Data coming from third party sources reside in our local directory of Unix and in CSV format.
2. A final master table is created from existing tables which is then inserted in HDFS using SQOOP.
3. Lastly, that table is analysed and visualized using python data analytical package, and matplotlib to map the datasets.

❖ DATA MODEL



❖ Description of the data source files

Claim CSV file fields

- Claim_ID : Unique, not null Primary Key
- Patient_ID: Unique, not null Primary key for Patient table
- Insurance_ID : Unique, not null Primary Key for Insurer table
- Group_ID : Unique, not null Primary Key for Group table
- Claim_type : Shows the type of claim
- Issue_date : Issue date for the claim made by patient
- Risk_Category : Level of risk
- Premium_plan_type : Shows the frequency of subscription
- Claim_Amount : Amount of claim made
- Handling_type : Type of handling
- Settlement : Status of settlement of claim
- Claim_status : Status of the claim made
- Submission_method : Method of claim submission
- Country : Country

Patient CSV file fields

- Patient_ID : Unique, not null Primary Key
- Insurance_ID : Unique, not null Primary Key for Insurer table
- Hospital_ID : Unique, not null Primary Key for Hospital table
- First_name : First name of the patient
- Last_name : Last name of the patient
- Gender : Gender
- DOB : Date of birth of patient
- Phone : Contact number of patient
- Nationality : Nationality of patient
- Insurance_type : Term of the insurance covered
- Address : Full address of patient
- Country : Country in which the patient belong to

Insurer CSV file fields

- Insurance_ID : Unique, not null Primary Key for Insurer table
- Company_name : Name of Company providing insurance
- Insurance_name : Name of Insurance Plan
- Start_Date : Date of start of insurance plan in dd/mm/yyyy
- End_Date : End Date of the plan in dd/mm/yyyy
- Effective_Date : Effective Date of the plan in dd/mm/yyyy
- Insurance_Base_Amount : Basic Coverage amount of the plan
- Term : Validity time period of plan in Years
- Topup_Amount : Amount of topup that can be added
- Insurance_Cover : Head coverage under the plan

Hospital CSV file fields

- Hospital_ID : unique, not null Primary key
- Hospital_Name : Name of the Hospital
- Hospital_City : City where the hospital is located
- Insurance_ID : Unique ID of Insurance as foreign key
- Insurance_Company : Name of Insurance Company

Group CSV file fields

- Group_ID : Unique, not null Primary key of group table
- Group_Name : Name of the group of Disease
- Effective_Date : Effective_date
- Term : Term of validity in Months

SubGroup CSV file fields

- SubGroup_ID : Unique, not null primary key
- Group_ID : Unique , foreign key of table Group
- SubGroup_Name : Name of the disease under the parent disease
- Effective_Date : Effective date
- Term : Term of validity in months

❖ Description of the final file

Final CSV file fields

- Patient_name : Name of the Patients
- Patient_address : Address of the Patients
- Claim_id : Unique, not null primary key
- Insurance_details : Name of the insurance and insurance company
- Claim_status_details : Status of the claim made
- Claim_issue_date : Issue date for the claim made by patient
- Insurance_term : Validity time period of plan in Years
- Patient_age : Age of Patient
- Group_details : Name of the disease and group of disease
- Insurance_amount : Base amount and top up amount

❖ Problem Statement

From this huge dataset, we process and analyze the data carefully and find out the following pattern :

- I. Insurance claims made by patient who are admitted in the Hospital from the past 6 months by the patient's age.
- II. Claims of patient based on disease.
- III. Claim uses and balance amount based on Insurance company.

❖ Code Templates

▪ Data Processing :

First we fetch all the raw source files and then cleanse it using python pandas, such as removing duplicate values, null values, then it gets converted to processed data. Here is some code snippets.

Cleansing of group.csv

```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [12]: df=pd.read_csv('raw_Group.csv')
```

```
In [13]: df.head()
```

Out[13]:

	Group ID	Group Name	Effective Date	Term(M)	Address
0	76891	Accident	12/12/2021	12	NaN
1	76892	Industrial Disease	3/1/2022	36	NaN
2	76893	Critical Illness	4/11/2021	30	NaN
3	76894	Surgeries	18/5/2022	24	NaN
4	76895	Genetic Illness	14/5/2022	26	NaN

```
In [14]: del df['Address']
```

```
In [15]: df.head()
```








Out[15]:

	Group ID	Group Name	Effective Date	Term(M)
--	----------	------------	----------------	---------

Cleansing of subgroup files by removing duplicates.

 **jupyter** Hospital Claim Management Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

       Run    Code 

In [15]: `df.head()`

Out[15]:

	Group ID	Group Name	Effective Date	Term(M)
0	76891	Accident	12/12/2021	12
1	76892	Industrial Disease	3/1/2022	36
2	76893	Critical Illness	4/11/2021	30
3	76894	Surgeries	18/5/2022	24
4	76895	Genetic Illness	14/5/2022	26

In [18]: `df.to_csv('Group.csv',index=False)`

In [19]: `df=pd.read_csv('raw_Subgroup1.csv')`

In [21]: `df.head(7)`

Out[21]:

	Subgroup ID	Group ID	Subgroup Name	Effective Date	Term(M)
0	56431	76891	Road-traffic accident	12/4/2021	32
1	56432	76891	Fractures caused by falls	22/9/2021	23
2	56433	76891	Gun shot	5/5/2022	15
3	56434	76891	Burn Cases	27/4/2022	26
4	56435	76891	Current Shock	1/1/2022	10
5	56436	76891	Broken Arm	NaN	8

In [40]: `df=df.dropna()`

In [41]: `df`

Out[41]:

	Subgroup ID	Group ID	Subgroup Name	Effective Date	Term(M)
0	56431	76891	Road-traffic accident	12/4/2021	32
1	56432	76891	Fractures caused by falls	22/9/2021	23
2	56433	76891	Gun shot	5/5/2022	15
3	56434	76891	Burn Cases	27/4/2022	26
4	56435	76891	Current Shock	1/1/2022	10

In [42]: `df = df.reset_index(drop = True)`

In [43]: `df`

Cleansing of subgroup files by removing null values.

```
In [66]: df=pd.read_csv('raw_Subgroup5.csv')
```

```
In [67]: df
```

```
Out[67]:
```

	Subgroup ID	Goup ID	Subgroup Name	Effective Date	Term(M)
0	56470	768925	Autism	NaN	25
1	56471	768925	Down syndrome	23/1/2022	30
2	56472	768925	FragileX syndrome	22/9/2021	27
3	56473	768925	Turner syndrome	4/4/2022	20
4	56474	768925	Trisomy 18	27/4/2022	28
5	56475	768925	Trisomy 13	1/1/2022	14

```
In [68]: df=df.dropna()
```

```
In [69]: df
```

```
Out[69]:
```

	Subgroup ID	Goup ID	Subgroup Name	Effective Date	Term(M)
1	56471	768925	Down syndrome	23/1/2022	30
2	56472	768925	FragileX syndrome	22/9/2021	27

■ Processed data

	Patient ID	Insurance ID	First name	Last name	Gender
0	45098.0	1003.0	Akriti	Singh	F
1	45099.0	1003.0	Pritam	Bannerjee	M
2	45100.0	1003.0	Salim	Khan	M
3	45101.0	1003.0	Yasoof	Akhtar	M
4	45102.0	1003.0	Wahida	Begum	F
5	45103.0	1003.0	Saheb	Chowdhury	M
6	45104.0	1003.0	Liza	Kusari	F
7	45105.0	1003.0	Rahul	Parmar	M
8	45106.0	1003.0	Koyel	Singh	F
9	45107.0	1003.0	Rai	Sen	F
10	45108.0	1003.0	Saina	Das	F
11	45109.0	1003.0	Saili	Goyel	F
12	45110.0	1003.0	Lianna	Reddy	F
13	45111.0	1003.0	Sanvi	Shetty	F
14	45112.0	1003.0	Sourav	Sanyal	M
15	45113.0	1003.0	Rounak	Bansal	M
16	45114.0	1003.0	Diya	Gupta	F

	Phone no	Nationality	Hospital ID	Insurance	Type
0	7.405288e+09	American	70001.0	Long	Term
1	4.014207e+09	American	70005.0	Short	Term
2	3.206315e+09	American	70004.0	Long	Term
3	4.057383e+09	American	70002.0	Long	Term
4	7.852044e+09	American	70003.0	Short	Term
5	8.567862e+09	American	70005.0	Short	Term
6	2.033718e+09	American	70004.0	Long	Term

allhost:8888/nbconvert/html/Untitled15.ipynb?download=false

8/22, 10:25 AM

Untitled15

7	6.176708e+09	American	70003.0	Short	Term
8	7.087782e+09	American	70002.0	Short	Term
9	6.204418e+09	American	70001.0	Short	Term
10	6.263711e+09	American	70001.0	Long	Term
11	3.472706e+09	American	70003.0	Short	Term
12	6.172935e+09	American	70005.0	Short	Term
13	9.319498e+09	American	70002.0	Long	Term
14	2.679530e+09	American	70004.0	Short	Term
15	8.136449e+09	American	70004.0	Long	Term
16	6.192009e+09	American	70002.0	Short	Term
17	4.804396e+09	American	70003.0	Long	Term
18	3.014947e+09	American	70001.0	Long	Term
19	7.142611e+09	American	70005.0	Short	Term
20	8.125245e+09	American	70004.0	Short	Term

	Address	Country
0	1660 Meadow DriveTennessee 27	USA
1	4427 Green hill road,Arkanas 08	USA
2	776 Harper street, kentucky 65	USA
3	53 Flynn Street Nebraska 16	USA
4	3085 simons road pennsylvania 34	USA
5	2929 forest drive kansas 05	USA
6	3 Herald New York 10001-3065	USA
7	42 W 35TH New York 10001-2233	USA
8	213 W 35th New York NY	USA
9	350 W 31st New York	USA
10	939 Stadium Drive Massachusetts	USA
11	3319 HORSESHOE LANE CALIFORNIA	USA
12	4170 POPLAR STREET, ILLIOIS	USA
13	1135 NORMAN STREET, CALIFORNIA	USA
14	3432 SYCAMORE STREET,CALIFORNIA	USA

Insurer.csv after processing and cleansing

	Insurance ID	Company Name	Insurance Name	Start Date(dd/mm/yyyy)	End Date(dd/mm/yyyy)	Effective date(dd/mm/yyyy)	Insurance Base Amount	Insurance Status	Term(Yrs)	TopUp Amount
1	1001	Aegon Life Insurance Co. Ltd.	Individual Health Insurance	1/1/2020	1/1/2036	15/1/2020	200000.0	Active	15	200000
4	1002	Edelweiss Life Insurance Co.	Family Health Insurance	3/1/2020	3/1/2039	15/3/2020	300000.0	Active	18	300000
7	1003	Birla Sun Life Insurance Co.	Critical illness Insurance	5/2/2020	5/2/2041	17/05/2020	400000.0	Active	20	150000

- Importing Processed data to MYSQL

List of all tables, that are imported to MySQL from the processed files.

```
Database changed
mysql> show tables;
+-----+
| Tables_in_project |
+-----+
| Claim              |
| Hospital            |
| Insurer            |
| Patient            |
| groups             |
| subgroup            |
+-----+
6 rows in set (0.00 sec)
```

- CLAIM TABLE

```
mysql> describe Claim;
```

Field	Type	Null	Key	Default	Extra
Patient_ID	int(6)	YES	MUL	NULL	
Insurance_ID	int(6)	YES	MUL	NULL	
Claim_ID	int(6)	NO	PRI	NULL	
Group_ID	int(6)	YES	MUL	NULL	
Claim_type	varchar(15)	YES		NULL	
Issue_date	date	YES		NULL	
Risk_Category	varchar(30)	YES		NULL	
Premium_plan_type	varchar(30)	YES		NULL	
Claim_amount	int(11)	YES		NULL	
Handling_type	varchar(30)	YES		NULL	
Settlement	varchar(30)	YES		NULL	
Claim_status	varchar(30)	YES		NULL	
Submission_method	varchar(30)	YES		NULL	
Country	varchar(30)	YES		NULL	

14 rows in set (0.00 sec)

```
mysql> select * from Claim limit 15;
```

Patient_ID	Insurance_ID	Claim_ID	Group_ID	Claim_type	Issue_date	Risk_Category	Premium_plan_type	Claim_amount	Handling_type	Settlement	Claim_status	Submission_method	Country
45104	1003	4007	76895	Cashless	2021-12-12	High	Monthly	20000	Complex	In Process	ACTIVE	Internet	India
45106	1003	4009	76894	Reimbursement	2022-05-09	High	Semi-annualy	29000	Accelerated	Invalid	INVALID	Mail	India
45111	1003	4014	76893	Cashless	2022-01-03	Low	Monthly	35000	Standard	Settled	RECOVERED	Internet	India
45113	1003	4016	76895	Reimbursement	2021-11-12	High	Monthly	20500	Standard	Settled	ACTIVE	Internet	India
45117	1003	4020	76891	Reimbursement	2021-11-04	Low	Quarterly	24100	Standard	Settled	RECOVERED	Mail	India
45121	1003	4024	76892	Cashless	2022-04-30	Low	Quarterly	19999	Express	In Process	ACTIVE	Mail	India
45123	1003	4026	76891	Cashless	2022-05-18	High	Semi-annualy	49500	Accelerated	In Process	ACTIVE	Call	India
45127	1003	4030	76894	Cashless	2022-04-21	High	Semi-annualy	41000	Standard	Invalid	INVALID	Mail	India
45128	1003	4031	76892	Reimbursement	2022-05-14	High	Quarterly	32000	Accelerated	In Process	ACTIVE	Call	India
45134	1003	4037	76891	Reimbursement	2022-04-07	Low	Semi-annualy	59000	Standard	Settled	RECOVERED	Mail	India
45135	1003	4038	76892	Reimbursement	2022-04-01	Medium	Monthly	20000	Express	Invalid	INVALID	Mail	India
45139	1003	4042	76891	Cashless	0000-00-00	Low	Semi-annualy	45000	Standard	In Process	ACTIVE	Mail	India
45142	1003	4045	76895	Cashless	2022-03-31	Medium	Quarterly	50000	Express	Invalid	INVALID	Internet	India
45145	1003	4048	76894	Cashless	2022-05-08	Low	Monthly	37000	Standard	Settled	RECOVERED	Internet	India
45147	1003	4050	76893	Reimbursement	2022-04-23	Low	Quarterly	22500	Standard	In Process	RECOVERED	Mail	India

- INSURER TABLE

```
mysql> describe Insurer;
```

Field	Type	Null	Key	Default	Extra
Insurance_ID	int(6)	NO	PRI	NULL	
Company_name	varchar(30)	YES		NULL	
Insurance_name	varchar(30)	YES		NULL	
Start_date	date	YES		NULL	
End_date	date	YES		NULL	
Effective_date	date	YES		NULL	
Insurance_base_amount	int(8)	YES		NULL	
Insurance_status	varchar(30)	YES		NULL	
Term	int(3)	YES		NULL	
Topup_Amount	int(8)	YES		NULL	
Insurance_cover	int(4)	YES		NULL	

```
11 rows in set (0.00 sec)
```

```
mysql> select * from Insurer;
```

Insurance_ID	Company_name	Insurance_name	Start_date	End_date	Effective_date	Insurance_base_amount	Insurance_status	Term	Topup_Amount	Insurance_cover
1001	Aegon Life Insurance Co. Ltd.	Individual Health Insurance	2020-01-01	2036-01-01	2020-01-15	200000	Active	15	200000	4
1002	Edelweiss Life Insurance Co.	Family Health Insurance	2020-01-03	2039-01-03	2020-03-15	300000	Active	18	300000	3
1003	Birla Sun Life Insurance Co.	Critical illness Insurance	2020-02-05	2041-02-05	2020-05-17	150000	Active	20	150000	3

```
3 rows in set (0.00 sec)
```


- PATIENT TABLE

```
mysql> describe Patient;
```

Field	Type	Null	Key	Default	Extra
Patient_ID	int(6)	NO	PRI	NULL	
Insurance_ID	int(6)	YES	MUL	NULL	
First_name	varchar(30)	YES		NULL	
Last_name	varchar(30)	YES		NULL	
Gender	varchar(5)	YES		NULL	
DOB	date	YES		NULL	
Phone	bigint(20)	YES		NULL	
Nationality	varchar(30)	YES		NULL	
Hospital_ID	int(6)	YES	MUL	NULL	
Insurance_type	varchar(30)	YES		NULL	
Address	varchar(50)	YES		NULL	
Country	varchar(30)	YES		NULL	

12 rows in set (0.00 sec)

```
mysql> select * from Patient limit 15;
```

Patient_ID	Insurance_ID	First_name	Last_name	Gender	DOB	Phone	Nationality	Hospital_ID	Insurance_type	Address	Country
45098	1003	Akriti	Singh	F	2002-05-09	8908908398	Indian	90001	Longterm	23	
sp road kolkata 64		India									
45099	1003	Pritam	Bannerjee	M	2001-02-02	9087988092	Indian	90003	Short term	12	
p Mukherjee road kolkata 87		India									
45100	1003	Salim	Khan	M	2001-06-09	9900887766	Indian	90005	Long term	34	
oy road howrah 34		India									
45101	1003	Yasoof	Akhtar	M	2000-08-07	9988455767	Indian	90002	Long term	34	
ellam road Chennai 01		India									
45102	1003	Wahida	Begum	F	2003-07-08	9090123467	Indian	90003	Short term	46	
roy road karnataka 04		India									
45103	1003	Saheb	Chowdhury	M	1992-03-09	9807890979	Indian	90004	Short term	36	
urnadas road kolkata 67		India									
45104	1003	Liza	Kusari	F	1990-04-10	9089757868	Indian	90004	Long term	34	
s sahani road. Mumbai 05		India									
45105	1003	Rahul	Parmar	M	2000-09-06	9999977685	Indian	90001	Short term	45	
inson road kolkata 98		India									
45106	1003	Koyel	Singh	F	2001-09-09	9999963667	Indian	90005	Long term	23	
oy road kolkata 98		India									
45107	1003	Rai	Sen	F	2002-09-08	9864680066	Indian	90002	Short term	88	
M road Chennai 67		India									
45108	1003	Saina	Das	F	1999-02-09	9876543210	Indian	90003	Long term	90	
oy road kolkata 89		India									
45109	1003	Sailli	Goyel	F	1997-03-08	9638527410	Indian	90001	Short term	34	
s road Bombay 01		India									
45110	1003	Lianna	Reddy	F	1992-07-22	9518742360	Indian	90002	Long term	44	
c road Malda 09		India									
45111	1003	Sanvi	Shetty	F	1995-03-22	9638521470	Indian	90003	Long term	34	
road punjab 99		India									
45112	1003	Sourav	Sanyal	M	1980-09-13	9032165478	Indian	90005	Long term	44	
uresh roy road kolkata 9		India									

- GROUPS TABLE

```
mysql> describe groups;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Group_ID       | int(6)        | NO   | PRI | NULL    |       |
| Group_Name     | varchar(30)   | YES  |     | NULL    |       |
| Effective_date | date          | YES  |     | NULL    |       |
| Term           | int(3)        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> select * from groups;
+-----+-----+-----+-----+
| Group_ID | Group_Name          | Effective_date | Term |
+-----+-----+-----+-----+
| 76891    | Accident            | 2021-12-12    | 12   |
| 76892    | Industrial Disease  | 2022-03-01    | 36   |
| 76893    | Critical Illness    | 2021-04-11    | 30   |
| 76894    | Surgeries           | 2022-05-18    | 24   |
| 76895    | Genetic Illness     | 2022-05-14    | 26   |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

- HOSPITAL TABLE

```
mysql> describe Hospital;
```

Field	Type	Null	Key	Default	Extra
Hospital_ID	int(6)	NO	PRI	NULL	
Hospital_name	varchar(30)	YES		NULL	
Hospital_City	varchar(30)	YES		NULL	
Insurance_ID	int(6)	YES	MUL	NULL	
Insurance_Company	varchar(30)	YES		NULL	

5 rows in set (0.00 sec)

```
mysql> create table Hospital( Hospital_ID int(6) Primary Key, Hospital_name varchar(30), Hospital_City varchar(30), Insurance_ID int(6), Insurance_Company varchar(30) );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> load data local infile "Desktop/Project/Hospitals.csv" into table Hospital fields terminated by "," lines terminated by "\n" ignore 1 rows;
Query OK, 15 rows affected, 2 warnings (0.01 sec)
Records: 15 Deleted: 0 Skipped: 0 Warnings: 2
```

```
mysql> select * from Hospital;
```

Hospital_ID	Hospital_name	Hospital_City	Insurance_ID	Insurance_Company
70001	The Johns Hopkins Hospital	Baltimore MD	1001	Aegon Life Insurance Co. Ltd.
70002	Massachusetts General Hospital	Boston MA	1001	Aegon Life Insurance Co. Ltd.
70003	UCSF Medical Center	San Francisco CA	1001	Aegon Life Insurance Co. Ltd.
70004	Mayo Clinic	Phoenix AZ	1001	Aegon Life Insurance Co. Ltd.
70005	Brigham And Women's Hospital	Boston MA	1001	Aegon Life Insurance Co. Ltd.
80001	Rechts der Isar Hospital	Munich	1002	Edelweiss Life Insurance Co.
80002	University Hospital Cologne	Cologne	1002	Edelweiss Life Insurance Co.
80003	University Medical Center	Freiburg	1002	Edelweiss Life Insurance Co.
80004	University Medical Center Schl	Keil	1002	Edelweiss Life Insurance Co.
80005	University Hospital Regensburg	Regensburg	1002	Edelweiss Life Insurance Co.
90001	Rainbow Hospitals	Hyderabad	1003	Birla Sun Life Insurance Co.
90002	Apollo Hospital	Visakhapatnam	1003	Birla Sun Life Insurance Co.
90003	Gauhati Medical College and Ho	Gauhati	1003	Birla Sun Life Insurance Co.
90004	AIIMS	Patna	1003	Birla Sun Life Insurance Co.
90005	Ruby General Hospital	Kolkata	1003	Birla Sun Life Insurance Co.

15 rows in set (0.00 sec)

- SUBGROUP TABLE

```
mysql> describe subgroup;
```

Field	Type	Null	Key	Default	Extra
Subrgoup_ID	int(6)	NO	PRI	NULL	
Group_ID	int(6)	YES	MUL	NULL	
Subgroup_name	varchar(30)	YES		NULL	
Effective_date	date	YES		NULL	
Term	int(3)	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
ignore 1 rows,
ERROR 2 (HY000): File '/Desktop/Project/Subgroup.csv' not found (Errcode: 2 - No such file or directory)
mysql> load data local infile "Desktop/Project/Subgroup.csv" into table subgroup fields terminated by "," lines terminated by "\n"
ignore 1 rows;
Query OK, 25 rows affected (0.01 sec)
Records: 25 Deleted: 0 Skipped: 0 Warnings: 0
```

```
mysql> select * from subgroup;
```

Subrgoup_ID	Group_ID	Subgroup_name	Effective_date	Term
56431	76891	Road-traffic accident	2021-12-04	32
56432	76891	Fractures caused by falls	2021-09-22	23
56433	76891	Gun shot	2022-05-05	15
56434	76891	Burn Cases	2022-04-27	26
56435	76891	Current Shock	2022-01-01	10
56441	76892	Industrial deafness	2022-01-23	32
56442	76892	Respiratory Problems	2021-09-22	23
56443	76892	Breathing Problems	2022-05-05	15
56444	76892	Dermatitis	2022-04-27	26
56445	76892	Musculoskeletal disorder	2022-01-01	10
56451	76893	Cancer	2022-01-23	30
56452	76893	Major organ transplant	2021-09-22	27
56453	76893	Kidney failure	2022-05-05	15
56454	76893	Paralysis	2022-04-27	26
56455	76893	First heart attack	2022-01-01	10
56461	76894	Neurological Surgery	2022-01-23	30
56462	76894	Oncology	2021-09-22	27
56463	76894	Chemotherapy	2022-04-04	20
56464	76894	Hystoscopy	2022-04-27	15
56465	76894	Nasal Concha	2022-01-01	10
56471	76895	Down syndrome	2022-01-23	30
56472	76895	FragileX syndrome	2021-09-22	27
56473	76895	Turner syndrome	2022-04-04	20
56474	76895	Trisomy 18	2022-04-27	28
56475	76895	Trisomy 13	2022-01-01	14

Final Table in MYSQL

```
mysql> select * from Final_table limit 15;
```

Patient_name	Patient_address	Claim_ID	Insurance_details	Claim_stat
us_details	Claim_issue_date	Insurer_term	Patient_age	Group_details
			Insurance_amount	
Liza Kusari	34- ks sahani road. Mumbai 05 India	4007	Critical illness Insurance,Birla Sun Life Insurance Co.	In Process
,ACTIVE	2021-12-12	20	32 Genetic Illness,Down syndrome	550000
Liza Kusari	34- ks sahani road. Mumbai 05 India	4060	Critical illness Insurance,Birla Sun Life Insurance Co.	In Process
,ACTIVE	2022-01-14	20	32 Accident,Road-traffic accident	550000
Sanvi Shetty	34-yk road punjab 99 India	4014	Critical illness Insurance,Birla Sun Life Insurance Co.	Settled,RE
COVERED	2022-01-03	20	27 Critical Illness,Kidney failure	550000
Sanvi Shetty	34-yk road punjab 99 India	4069	Critical illness Insurance,Birla Sun Life Insurance Co.	Settled,RE
COVERED	2022-03-03	20	27 Industrial Disease,Dermatitis	550000
Khushi Khan	9- ss roy road Delhi 09 India	4020	Critical illness Insurance,Birla Sun Life Insurance Co.	Settled,RE
COVERED	2021-11-04	20	32 Accident,Burn Cases	550000
Khushi Khan	9- ss roy road Delhi 09 India	4073	Critical illness Insurance,Birla Sun Life Insurance Co.	Settled,RE
COVERED	2022-01-23	20	32 Critical Illness,First heart attack	550000
Mollika Singhania	7- kk road Chennai 09 India	4026	Critical illness Insurance,Birla Sun Life Insurance Co.	In Process
,ACTIVE	2022-05-18	20	20 Accident,Current Shock	550000
Mollika Singhania	7- kk road Chennai 09 India	4074	Critical illness Insurance,Birla Sun Life Insurance Co.	In Process
,ACTIVE	2022-01-15	20	20 Surgeries,Nasal Concha	550000
Rohan Ghose	9- linkon street kolkata 9 India	4031	Critical illness Insurance,Birla Sun Life Insurance Co.	In Process
,ACTIVE	2022-05-14	20	29 Industrial Disease,Musculoskeletal disorder	550000
Rohan Ghose	9- linkon street kolkata 9 India	4090	Critical illness Insurance,Birla Sun Life Insurance Co.	Invalid,IN
VALID	2022-07-03	20	29 Accident,Fractures caused by falls	550000
Lokesh Ghose	2- sr road Pune 4 India	4038	Critical illness Insurance,Birla Sun Life Insurance Co.	Invalid,IN
VALID	2022-04-01	20	26 Industrial Disease,Respiratory Problems	550000
Lokesh Ghose	2- sr road Pune 4 India	4081	Critical illness Insurance,Birla Sun Life Insurance Co.	In Process
,ACTIVE	2022-11-02	20	26 Surgeries,Oncology	550000
Rani Chakraborty	6 - Fort William street kolkata 9 India	4042	Critical illness Insurance,Birla Sun Life Insurance Co.	In Process
,ACTIVE	0000-00-00	20	30 Accident,Burn Cases	550000
Afzal Alan	7- ju colony kolkata 7 India	4045	Critical illness Insurance,Birla Sun Life Insurance Co.	Invalid,IN
VALID	2022-03-31	20	23 Genetic Illness,Trisomy 13	550000
Shravan Ghosal	7- cu road Chennai 8 India	4048	Critical illness Insurance,Birla Sun Life Insurance Co.	Settled,RE
COVERED	2022-05-08	20	27 Surgeries,Neurological Surgery	550000

```
15 rows in set (0.00 sec)
```

■ DATA TRANSFORMATION

```
In [12]: df=pd.read_sql_query("SELECT * FROM PATIENT",connection)
(df.head())
```

/home/ubh01/anaconda3/lib/python3.9/site-packages/pandas/io/sql.py:761: UserWarning: pandas only support SQLAlchemy connectable(engine/connection) or database string URI or sqlite3 DBAPI2 connection or other DBAPI2 objects are not tested, please consider using SQLAlchemy
warnings.warn(

```
Out[12]:
```

	Patient_ID	Insurance_ID	Hospital_ID	First_name	Last_name	Gender	DOB	Phone	Nationality	Insurance_type	Address	C
0	45098	1003	90001	Akriti	Singh	F	2002-05-09	8908908398	Indian	Longterm	23- asp road kolkata 64	
1	45099	1003	90003	Pritam	Bannerjee	M	2001-02-02	9087988092	Indian	Short term	12- sp Mukherjee road kolkata 87	
2	45100	1003	90005	Salim	Khan	M	2001-06-09	9900887766	Indian	Long term	34- roy road howrah 34	
3	45101	1003	90002	Yasoof	Akhtar	M	2000-08-07	9988455767	Indian	Long term	34 chellam road Chennai 01	
4	45102	1003	90003	Wahida	Begum	F	2003-07-08	9090123467	Indian	Short term	46 bb roy road karnataka 04	

```
In [13]: cursor=connection.cursor()
```

```
In [14]: df["Name"]=df["First_name"]+" "+df["Last_name"]
```

```
In [18]: df.head()
df2=df.drop('First_name',axis=1)
df2=df2.drop('Last_name',axis=1)
df2.head()
```

```
Out[18]:
```

	Patient_ID	Insurance_ID	Hospital_ID	Gender	DOB	Phone	Nationality	Insurance_type	Address	Country	Name
0	45098	1003	90001	F	2002-05-09	8908908398	Indian	Longterm	23- asp road kolkata 64	India\r	Akriti Singh
1	45099	1003	90003	M	2001-02-02	9087988092	Indian	Short term	12- sp Mukherjee road kolkata 87	India\r	Pritam Bannerjee
2	45100	1003	90005	M	2001-06-09	9900887766	Indian	Long term	34- roy road howrah 34	India\r	Salim Khan
3	45101	1003	90002	M	2000-08-07	9988455767	Indian	Long term	34 chellam road Chennai 01	India\r	Yasoof Akhtar
4	45102	1003	90003	F	2003-07-08	9090123467	Indian	Short term	46 bb roy road karnataka 04	India\r	Wahida Begum


```
[1]: import pymysql
import pandas as pd
import sqlalchemy

[2]: conn=pymysql.
      ↪connect(host='localhost',port=int(3306),user='root',passwd='password',db='Hospital_DB')

[3]: df=pd.read_sql_query("SELECT * FROM Insurer",conn)
print(df.head())
```

	InsuranceID	CompanyName	InsuranceName \
0	1001	Aegon Life Insurance Co. Ltd.	Individual Health Insurance
1	1002	Edelweiss Life Insurance Co.	Family Health Insurance
2	1003	Birla Sun Life Insurance Co.	Critical illness Insurance

	StartDate	EndDate	EffectiveDate	InsuranceBaseAmount	InsuranceStatus \
0	2020-01-01	2036-01-01	2020-01-15	200000	Active
1	2020-01-03	2039-01-03	2020-03-15	300000	Active
2	2020-02-05	2041-02-05	2020-05-17	400000	Active

	Term	TopupAmount	InsuranceCover
0	15	200000	4
1	18	300000	3
2	20	150000	3

```
[4]: cursor=conn.cursor()
```

```
[5]: data=pd.DataFrame({
      'Insurance_amount':df["InsuranceBaseAmount"]+df["TopupAmount"]
    })
```

```
[6]: data
```

```
[6]: Insurance_amount
0      400000
1      600000
2      550000
```

```

38      In Process , ACTIVE
39      In Process , ACTIVE
40      Invalid , INVALID
41      In Process , ACTIVE
42      In Process , RECOVERED
43      In Process , ACTIVE
44      Settled , RECOVERED
45      Invalid , INVALID
46      Settled , RECOVERED
47      Settled , ACTIVE
48      Settled , RECOVERED
49      In Process , ACTIVE
50      In Process , ACTIVE
51      In Process , ACTIVE
52      Invalid , INVALID
53      In Process , ACTIVE REVIEW
54      Invalid , INVALID
55      Settled , RECOVERED
56      Invalid , INVALID

```

```
In [26]: CLAIMS=pd.DataFrame({'Claim': df['Claim_ID']})
```

```
In [27]: CLAIMS
```

```

Out[27]:      Claim
0      4007
1      4009
2      4014
3      4016
4      4020
5      4024
6      4026
7      4030
8      4031
9      4037
10     4038
11     4042
12     4045
13     4048
14     4050
15     4051
16     4053
17     4055
18     4059
19     4060
20     4061
21     4066
22     4069
23     4070
24     4072
25     4073
26     4074
27     4077
28     4080
29     4081
30     4086
31     4089
32     4090
33     4096
34     4098
35     4100
36     4101
37     4103
38     4105
39     4107
40     4109
41     4110
42     5003

```


13 2022-05-08
 14 2022-04-23
 15 2021-12-31
 16 2021-05-22
 17 2022-02-02
 18 2021-09-16
 19 2022-01-14
 20 2022-01-21
 21 2021-12-08
 22 2022-03-03
 23 2022-03-01
 24 2021-12-11
 25 2022-01-23
 26 2022-01-15
 27 2022-01-02
 28 2022-01-01
 29 2022-11-02
 30 2022-01-22
 31 2022-01-02
 32 2022-07-03
 33 2021-03-11
 34 2022-06-05
 35 2022-12-05
 36 2021-09-08
 37 2022-05-04
 38 2021-03-09
 39 2021-09-29
 40 2022-02-02
 41 2022-04-02
 42 2022-04-23
 43 2021-12-02
 44 2022-01-05
 45 2022-06-05
 46 2021-10-10
 47 2021-11-12
 48 2021-04-12
 49 2022-04-19
 50 2022-03-10
 51 2022-05-05
 52 2022-01-01
 53 2022-03-29
 54 2022-03-19
 55 2022-05-10
 56 2022-09-05

```
In [30]: df=pd.read_sql_query("SELECT * FROM INSURER",connection)
print(df.head())
```

	Insurance_ID	Company_name	Insurance_name	\
0	1001	Aegon Life Insurance Co. Ltd.	Individual Health Insurance	
1	1002	Edelweiss Life Insurance Co.	Family Health Insurance	
2	1003	Birla Sun Life Insurance Co.	Critical illness Insurance	

	Start_date	End_date	Effective_date	Insurance_Status	Term	Topup_amount	\
0	2020-01-01	2036-01-01	2020-01-15	200000	0	15	
1	2020-01-03	2039-01-03	2020-03-15	300000	0	18	
2	2020-02-05	2041-02-05	2020-05-17	400000	0	20	

	Insurance_Cover
0	200000
1	300000
2	150000

/home/ubh01/anaconda3/lib/python3.9/site-packages/pandas/io/sql.py:761: UserWarning: pandas only support SQLAlchemy connectable(engine/connection) or database string URI or sqlite3 DBAPI2 connection or other DBAPI2 objects are not tested, please consider using SQLAlchemy warnings.warn()

```
In [30]: df=pd.read_sql_query("SELECT * FROM INSURER",connection)
print(df.head())
```

	Insurance_ID	Company_name	Insurance_name	\
0	1001	Aegon Life Insurance Co. Ltd.	Individual Health Insurance	
1	1002	Edelweiss Life Insurance Co.	Family Health Insurance	
2	1003	Birla Sun Life Insurance Co.	Critical illness Insurance	

	Start_date	End_date	Effective_date	Insurance_Status	Term	Topup_amount	\
0	2020-01-01	2036-01-01	2020-01-15	200000	0	15	
1	2020-01-03	2039-01-03	2020-03-15	300000	0	18	
2	2020-02-05	2041-02-05	2020-05-17	400000	0	20	

▪ SQOOP and HIVE and HDFS

All tables are imported to HDFS from MYSQL

```
ubh01@ubh01:~$ hdfs dfs -ls
22/06/07 15:25:55 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 8 items
drwxr-xr-x - ubh01 supergroup          0 2022-06-07 15:19 Claim
drwxr-xr-x - ubh01 supergroup          0 2022-06-07 14:58 Final_table
drwxr-xr-x - ubh01 supergroup          0 2022-06-07 15:21 Hospital
drwxr-xr-x - ubh01 supergroup          0 2022-06-07 15:23 Insurer
drwxr-xr-x - ubh01 supergroup          0 2022-06-07 15:22 Patient
drwxr-xr-x - ubh01 supergroup          0 2022-06-07 15:24 groups
drwxr-xr-x - ubh01 supergroup          0 2022-06-07 15:25 subgroup
drwxr-xr-x - ubh01 supergroup          0 2022-06-07 11:11 ubh01
ubh01@ubh01:~$ hdfs dfs -ls Final_table
22/06/07 15:26:32 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 ubh01 supergroup          0 2022-06-07 14:58 Final_table/_SUCCESS
-rw-r--r-- 1 ubh01 supergroup    10229 2022-06-07 14:58 Final_table/part-m-00000
ubh01@ubh01:~$
```

We have used Sqoop for the import from HDFS to HIVE

```
ubh01@ubh01:~$ sqoop import --connect jdbc:mysql://ubh01/project -table Final_table --fields-terminated-by '|' --username root --password password -m 1
Warning: /home/ubh01/sqoop-1.4.7.bin__hadoop-2.6.0/./hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /home/ubh01/sqoop-1.4.7.bin__hadoop-2.6.0/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /home/ubh01/sqoop-1.4.7.bin__hadoop-2.6.0/./zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
22/06/07 14:57:36 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
22/06/07 14:57:36 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
22/06/07 14:57:36 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
22/06/07 14:57:36 INFO tool.CodeGenTool: Beginning code generation
Loading class 'com.mysql.jdbc.Driver'. This is deprecated. The new driver class is 'com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading of the driver class is generally unnecessary.
22/06/07 14:57:37 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM 'Final_table' AS t LIMIT 1
22/06/07 14:57:37 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM 'Final_table' AS t LIMIT 1
22/06/07 14:57:37 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /home/ubh01/hadoop-2.7.1
Note: /tmp/sqoop-ubh01/compile/bbfb98d630adc271fc0b148d42042c24/Final_table.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
22/06/07 14:57:40 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-ubh01/compile/bbfb98d630adc271fc0b148d42042c24/Final_table.jar
22/06/07 14:57:40 WARN manager.MySQLManager: It looks like you are importing from mysql.
22/06/07 14:57:40 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
22/06/07 14:57:40 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
22/06/07 14:57:40 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
22/06/07 14:57:40 INFO mapreduce.ImportJobBase: Beginning import of Final_table
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/ubh01/hadoop-2.7.1/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/ubh01/hbase-1.1.2/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
22/06/07 14:57:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
22/06/07 14:57:40 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
22/06/07 14:57:41 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
22/06/07 14:57:41 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
```

Like this, we have created the schema for every table in HIVE

```
hive> use project;
OK
Time taken: 0.038 seconds
hive> create table final_table (
  > Patient_name string,Patient_address string,Claim_ID int,Insurance_details string,
  > Claim_status_details string,Claim_issue_date Date,Insurer_term int,Patient_age int,Group_details string,Insurance_amount int )
  > row format delimited
  > fields terminated by '|'
  > location '/user/ubh01/Final_table';
OK
Time taken: 0.413 seconds
hive> select * from final_table;
OK
Liza Kusari 34- ks sahani road. Mumbai 05 India 4007 Critical illness Insurance,Birla Sun Life Insurance Co. In Process,ACTIVE 202
1-12-12 20 32 Genetic Illness,Down syndrome 550000
Liza Kusari 34- ks sahani road. Mumbai 05 India 4060 Critical illness Insurance,Birla Sun Life Insurance Co. In Process,ACTIVE 202
2-01-14 20 32 Accident,Road-traffic accident 550000
Sanvi Shetty 34-yk road punjab 99 India 4014 Critical illness Insurance,Birla Sun Life Insurance Co. Settled,RECOVERED 2022-01-03
20 27 Critical Illness,Kidney failure 550000
Sanvi Shetty 34-yk road punjab 99 India 4069 Critical illness Insurance,Birla Sun Life Insurance Co. Settled,RECOVERED 2022-03-03
20 27 Industrial Disease,Dermatitis 550000
Khushi Khan 9- ss roy road Delhi 09 India 4020 Critical illness Insurance,Birla Sun Life Insurance Co. Settled,RECOVERED 2021-11-04
20 32 Accident,Burn Cases 550000
Khushi Khan 9- ss roy road Delhi 09 India 4073 Critical illness Insurance,Birla Sun Life Insurance Co. Settled,RECOVERED 2022-01-23
20 32 Critical Illness,First heart attack 550000
Mollika Singhania 7- kk road Chennai 09 India 4026 Critical illness Insurance,Birla Sun Life Insurance Co. In Process,ACTIVE 202
2-05-18 20 20 Accident,Current Shock 550000
Mollika Singhania 7- kk road Chennai 09 India 4074 Critical illness Insurance,Birla Sun Life Insurance Co. In Process,ACTIVE 202
2-01-15 20 20 Surgeries,Nasal Concha 550000
Rohan Ghose 9- linkon street kolkata 9 India 4031 Critical illness Insurance,Birla Sun Life Insurance Co. In Process,ACTIVE 202
2-05-14 20 29 Industrial Disease,Musculoskeletal disorder 550000
Rohan Ghose 9- linkon street kolkata 9 India 4090 Critical illness Insurance,Birla Sun Life Insurance Co. Invalid,INVALID 202
2-07-03 20 29 Accident,Fractures caused by falls 550000
Lokesh Ghose 2- sr road Pune 4 India 4038 Critical illness Insurance,Birla Sun Life Insurance Co. Invalid,INVALID 2022-04-01 20 26
Industrial Disease,Respiratory Problems 550000
Lokesh Ghose 2- sr road Pune 4 India 4081 Critical illness Insurance,Birla Sun Life Insurance Co. In Process,ACTIVE 2022-11-02 20
26 Surgeries,Oncology 550000
Rani Chakraborty 6 - Fort William street kolkata 9 India 4042 Critical illness Insurance,Birla Sun Life Insurance Co. In Process,
ACTIVE NULL 20 30 Accident,Burn Cases 550000
Afzal Alam 7- ju colony kolkata 7 India 4045 Critical illness Insurance,Birla Sun Life Insurance Co. Invalid,INVALID 2022-03-31
20 23 Genetic Illness,Trisomy 13 550000
```

We have imported the tables : claim, final_table, groups, hospital, insurer, patient and subgroup to HIVE.

```
hive> show tables;
OK
claim
dept
dummy
dummy2
emp
final_table
groups
hospital
insurer
mydevices
patient
salgrade
subgroup
webpage
webpage_titanic
Time taken: 0.064 seconds, Fetched: 15 row(s)
```

▪ Final processing from data

Analyzing data from Final table to plot the final Graphs.

❖ Claims based on last 6 months.

```
[1]: import pymysql
import pandas as pd
import sqlalchemy

[2]: conn=pymysql.
      ↪connect(host='localhost',port=int(3306),user='root',passwd='password',db='Hospital_DB')

[3]: import matplotlib.pyplot as plt

[4]: mycursor = conn.cursor()

[5]: mycursor.execute("select Claim_issue_date,Patient_age from FinalTable where_
      ↪Claim_issue_date >=now()-interval 6 month;")
result = mycursor.fetchall

[6]: IssueDate = []
Age = []

for i in mycursor:
    IssueDate.append(i[0])
    Age.append(i[1])

print("Claim issue date = ", IssueDate)
print("Patient age = ", Age)

Claim issue date = [datetime.date(2021, 12, 12), datetime.date(2022, 5, 9),
datetime.date(2022, 1, 3), datetime.date(2022, 4, 30), datetime.date(2022, 5,
18), datetime.date(2022, 4, 21), datetime.date(2022, 5, 14), datetime.date(2022,
4, 7), datetime.date(2022, 4, 1), datetime.date(2022, 3, 31),
datetime.date(2022, 5, 8), datetime.date(2022, 4, 23), datetime.date(2021, 12,
31), datetime.date(2022, 2, 2), datetime.date(2022, 1, 14), datetime.date(2022,
1, 21), datetime.date(2021, 12, 8), datetime.date(2022, 3, 3),
datetime.date(2022, 3, 1), datetime.date(2021, 12, 11), datetime.date(2022, 1,
23), datetime.date(2022, 1, 15), datetime.date(2022, 1, 2), datetime.date(2022,
1, 1), datetime.date(2022, 1, 22), datetime.date(2022, 1, 2),
datetime.date(2022, 5, 4), datetime.date(2022, 2, 2), datetime.date(2022, 4, 2),
datetime.date(2022, 4, 23), datetime.date(2022, 1, 5), datetime.date(2022, 4,
19), datetime.date(2022, 3, 10), datetime.date(2022, 5, 5), datetime.date(2022,
1, 1), datetime.date(2022, 3, 29), datetime.date(2022, 3, 19),
datetime.date(2022, 5, 10)]
Patient age = [32, 21, 27, 18, 20, 26, 29, 28, 26, 23, 27, 23, 21, 10, 32, 26,
28, 27, 29, 27, 32, 20, 38, 39, 39, 35, 10, 28, 21, 27, 33, 36, 21, 24, 22, 23,
35, 30]

]: pl.rcParams['figure.figsize']=[20,8]
pl.bar(IssueDate, Age,width=3)
pl.xlabel("Claims based on last 6 months")
pl.ylabel("Patient Age")
pl.title("Claim Details")
pl.show()
```

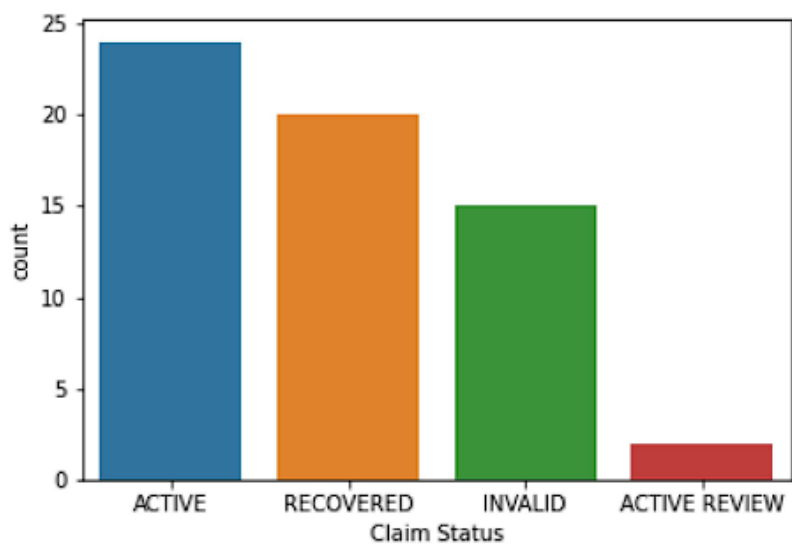
❖ Patient claims based on disease.

```
gender_dict= dict(zip(Patient_df.Patient_ID, Patient_df.Gender))
Claim_df["Gender"]=Claim_df.apply(lambda x:gender_dict[x[0]], axis=1)
print(Claim_df)
gendf=pd.DataFrame()
gendf=pd.pivot_table(Claim_df, index=['Group'], columns=['Gender'], values=['Patient_ID'], aggfunc='count')
print(gendf)
plt.figure()
gendf.plot.bar(title='STACKED BAR CHART', stacked=True)
plt.xlabel('Types of Group')
plt.ylabel('No.of Customers')
plt.savefig("groupgenderanalysis.png",bbox_inches='tight')
```

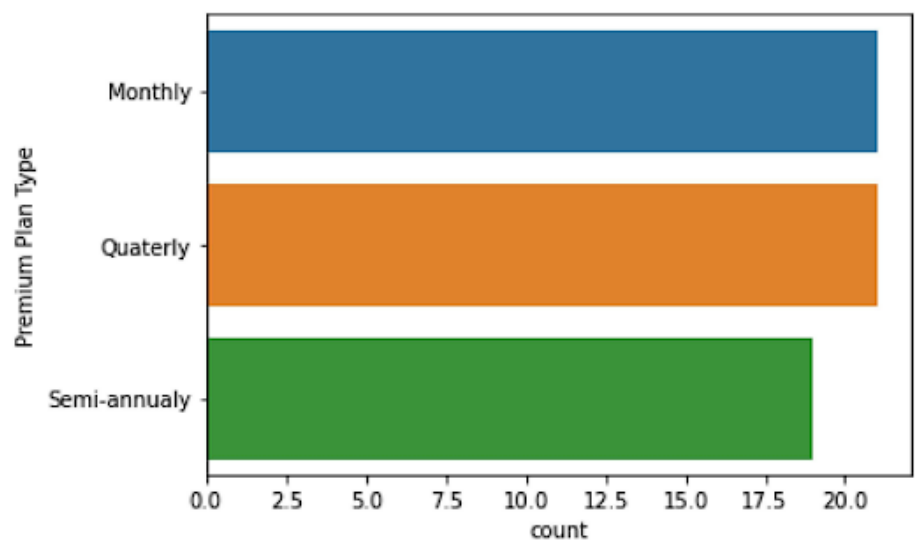
```
gdf= pd.DataFrame()
gdf=Claim_df["Group"].value_counts()
print(gdf)
plt.figure()
gdf.plot.bar(title='Patient and Group Analysis',color='red')
plt.xlabel('Types of Group')
plt.ylabel('No of patients')
plt.savefig("groupanalysis.png",bbox_inches='tight')
```

❖ OUTPUT SCREEN

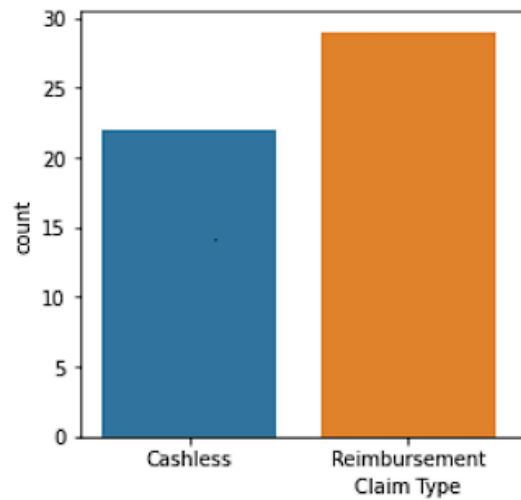
Different claims made and the claim status



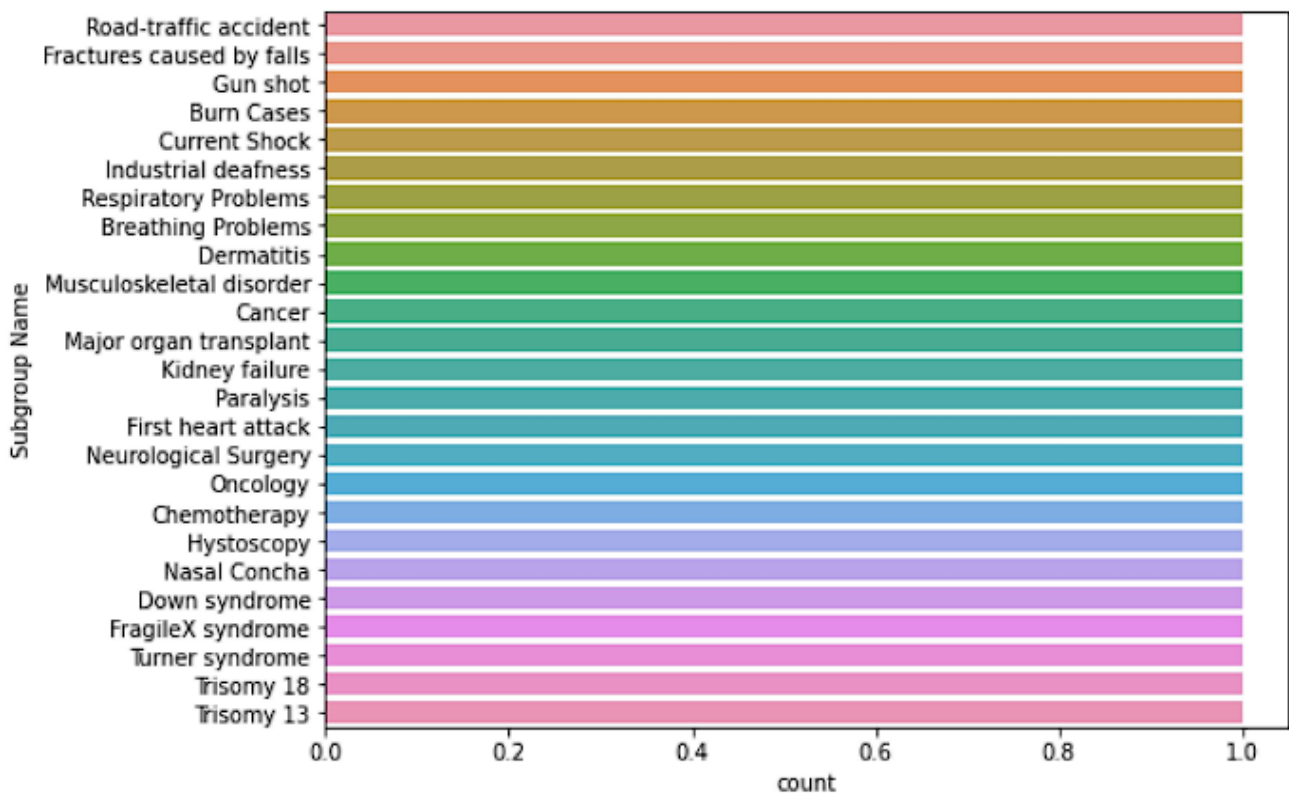
Insurance Premium Plan Type



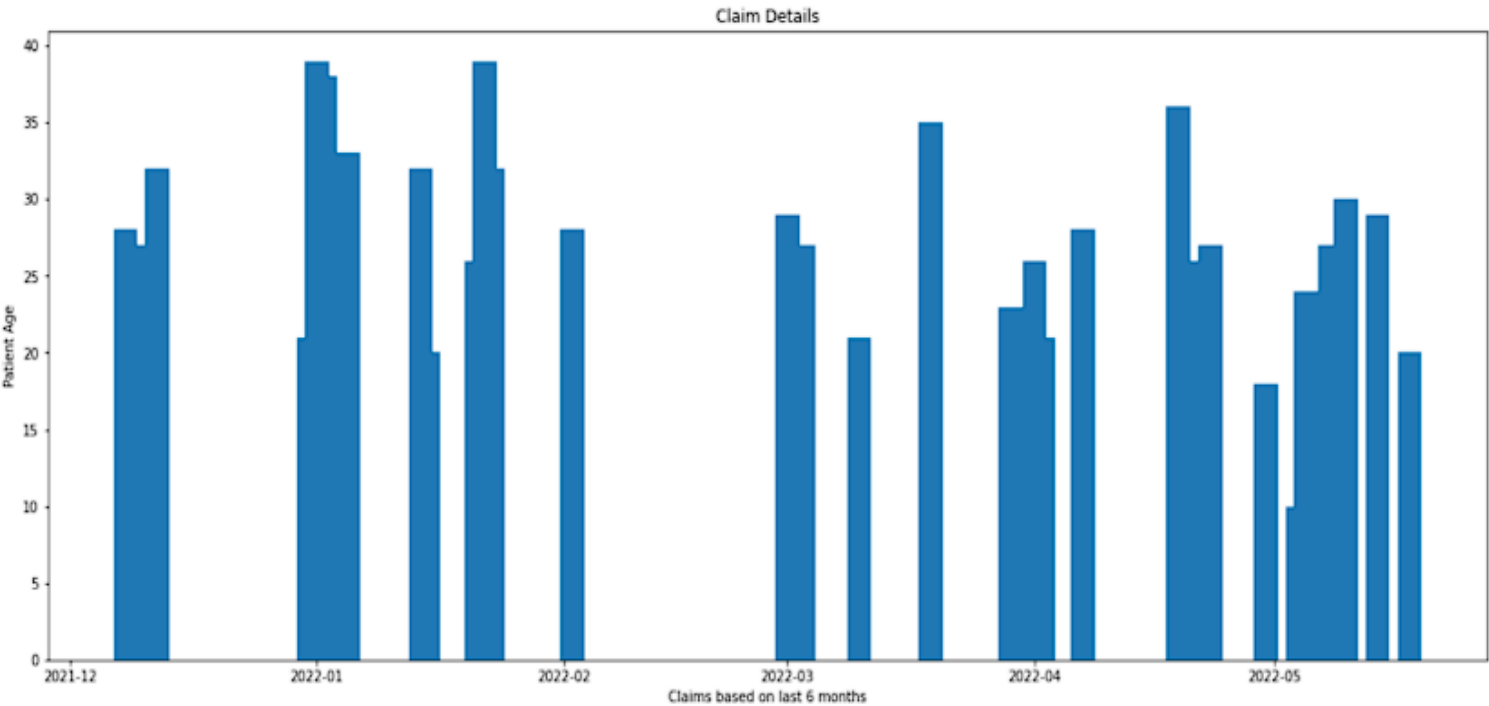
Type of claim request Cashless or Reimbursement



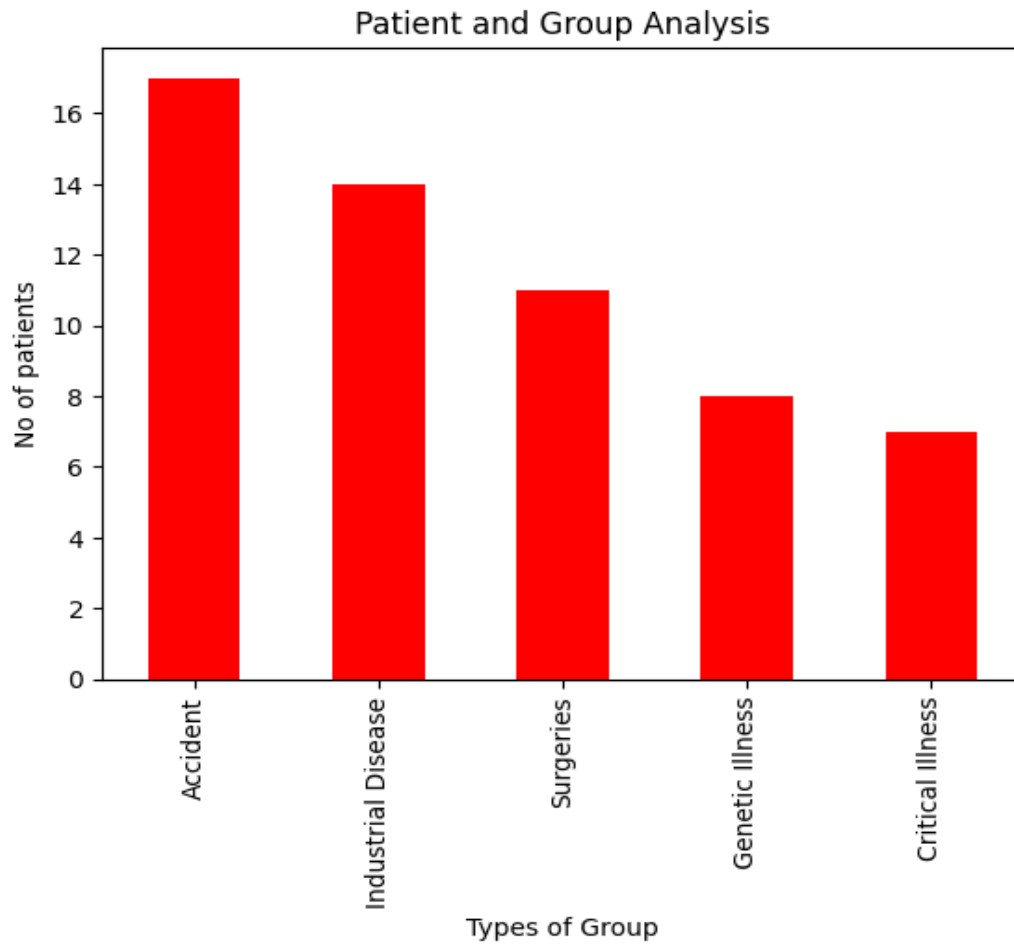
Subgroups of Disease



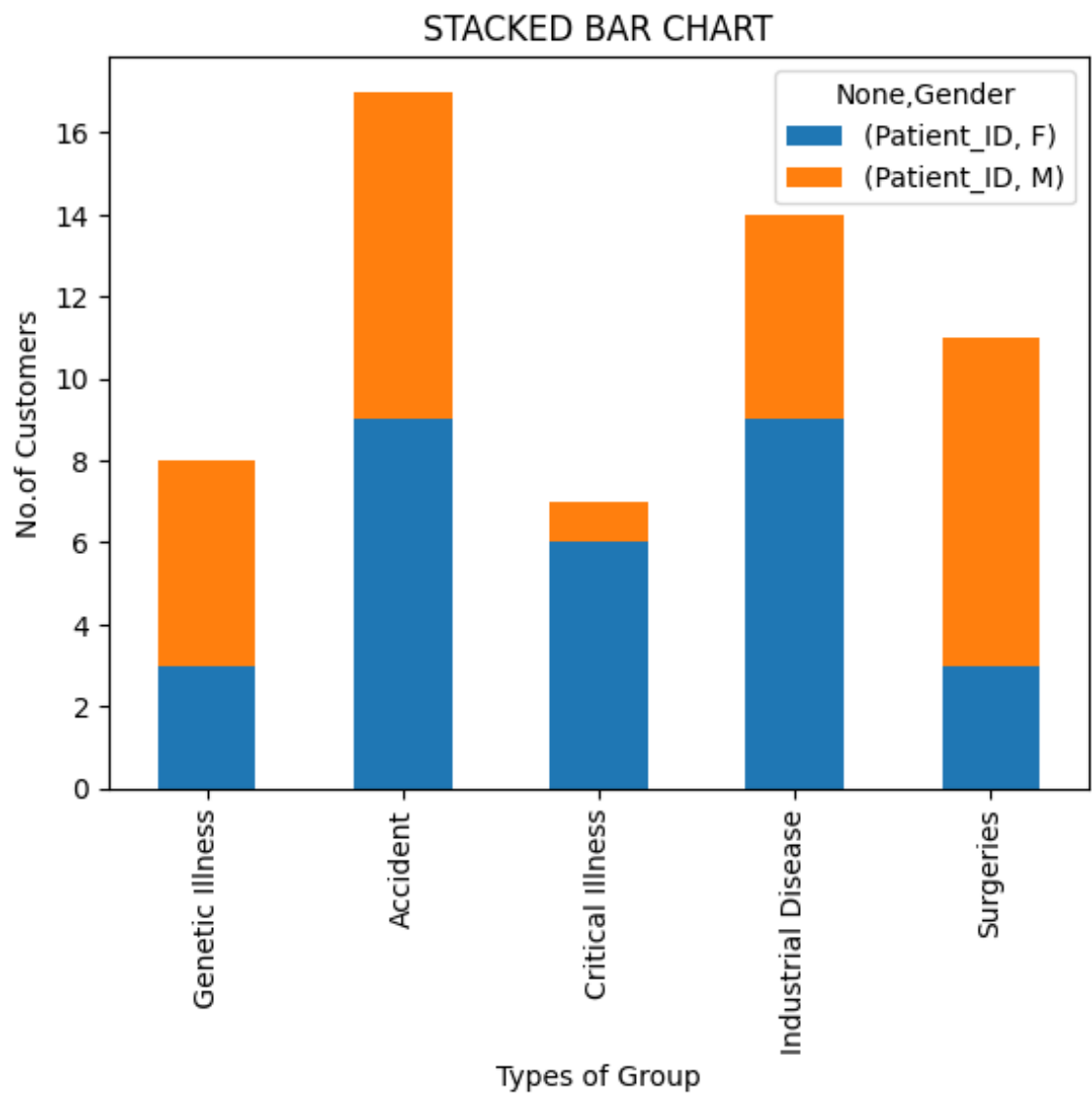
Patient claims based on past 6 months



Patient claims based on disease



Patient claims based on disease, classified by Gender



❖ Business Benefits/Conclusion

The company is facing challenges to get the insights they need to improve their services as the dataset is on a much bigger scale and may contain missing records. So the company wants to take the help of Data Analytics using Big Data Ecosystem in order to evaluate the quality of care provided by health care providers. To do this, we have been assigned to identify patterns of past claims made for insurance in order to understand the customers better, and provide targeted experience and improve customer retention.

❖ Further Enhancements/Recommendations

This project has a vast scope in future for all the Insurance Companies. Companies can get detailed analysis of data from this project and can serve the customers according to their special needs and benefits. They can also change their insurance plans and business needs for maximum profits and also provide with the optimal service to patients.

❖ References/Bibliography

- i. Python Data analytics
- ii. DrawIO
- iii. GeeksforGeeks
- iv. Hive
- v. Jupyter
- vi. Hadoop

❖ GitHub Link

[BigData-Project-Hospital-Claim-Management](#)