

Transformation

June 3, 2022

```
[15]: import os
import mysql.connector
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pymysql
import sqlalchemy
import warnings
warnings.filterwarnings('ignore')
```

```
[16]: user_name=os.environ.get('DB_USER')
password=os.environ.get('DB_PASS')
```

```
[17]: connection=mysql.connector.connect(host = 'localhost',
                                         user = 'root',
                                         passwd='password',
                                         db=' Hospitalmanagement_db')
```

```
[18]: connection
```

```
[18]: <mysql.connector.connection_cext.CMySQLConnection at 0x7fb3e0055f40>
```

```
[19]: connection.is_connected()
```

```
[19]: True
```

```
[20]: hospital_db=pd.read_sql_query('SHOW TABLES FROM『
→Hospitalmanagement_db』',connection)
```

```
[21]: hospital_db
```

```
[21]:    Tables_in_Hospitalmanagement_db
0                      CLAIM
1                      GROUPS
2                     HOSPITAL
```

```
3           INSURER
4           PATIENT
5           SUBGROUPS
```

```
[23]: tables=hospital_db['Tables_in_Hospitalmanagement_db']
```

```
[24]: tables
```

```
[24]: 0      CLAIM
1      GROUPS
2      HOSPITAL
3      INSURER
4      PATIENT
5      SUBGROUPS
Name: Tables_in_Hospitalmanagement_db, dtype: object
```

```
[25]: for table_name in tables:
    output=pd.read_sql_query('DESCRIBE {}'.format(table_name),connection)
    print(table_name)
    print(output, '\n')
```

CLAIM

	Field	Type	Null	Key	Default	Extra
0	Patient_ID	int(6)	NO	MUL	None	
1	Insurance_ID	int(6)	NO	MUL	None	
2	Claim_ID	int(6)	NO	PRI	None	
3	Group_ID	int(6)	NO	MUL	None	
4	Claim_Type	varchar(15)	NO		None	
5	Issue_date	date	NO		None	
6	Risk_category	varchar(30)	NO		None	
7	Premium_plan_type	varchar(30)	NO		None	
8	Claim_Amount	int(11)	NO		None	
9	Handling_type	varchar(30)	NO		None	
10	Settlement	varchar(30)	NO		None	
11	Claim_status	varchar(30)	NO		None	
12	Submission_method	varchar(30)	NO		None	
13	Country	varchar(30)	NO		None	

GROUPS

	Field	Type	Null	Key	Default	Extra
0	Group_ID	int(6)	NO	PRI	None	
1	Group_name	varchar(30)	YES		None	
2	Effective_date	date	YES		None	
3	Term	int(3)	YES		None	

HOSPITAL

Field	Type	Null	Key	Default	Extra
-------	------	------	-----	---------	-------

0	Hospital_ID	int(6)	NO	PRI	None
1	Insurance_ID	int(6)	YES	MUL	None
2	Hospital_name	varchar(30)	YES		None
3	Hospital_city	varchar(30)	YES		None
4	Insurance_company	varchar(30)	YES		None

INSURER

	Field	Type	Null	Key	Default	Extra
0	Insurance_ID	int(6)	NO	PRI	None	
1	Company_name	varchar(30)	NO		None	
2	Insurance_name	varchar(30)	NO		None	
3	Start_date	date	YES		None	
4	End_date	date	YES		None	
5	Effective_date	date	YES		None	
6	Insurance_Status	varchar(30)	YES		None	
7	Term	int(3)	NO		None	
8	Topup_amount	int(8)	NO		None	
9	Insurance_Cover	int(3)	NO		None	

PATIENT

	Field	Type	Null	Key	Default	Extra
0	Patient_ID	int(6)	NO	PRI	None	
1	Insurance_ID	int(6)	YES	MUL	None	
2	Hospital_ID	int(6)	YES	MUL	None	
3	First_name	varchar(30)	YES		None	
4	Last_name	varchar(30)	YES		None	
5	Gender	varchar(5)	YES		None	
6	DOB	date	YES		None	
7	Phone	bigint(20)	YES		None	
8	Nationality	varchar(30)	YES		None	
9	Insurance_type	varchar(30)	YES		None	
10	Address	varchar(50)	YES		None	
11	Country	varchar(30)	YES		None	

SUBGROUPS

	Field	Type	Null	Key	Default	Extra
0	Subgroup_ID	int(6)	NO	PRI	None	
1	Group_ID	int(6)	YES	MUL	None	
2	Subgroup_name	varchar(30)	YES		None	
3	Effective_date	date	YES		None	
4	Term	int(3)	YES		None	

```
[66]: patient_df=pd.read_sql_query("SELECT * FROM PATIENT",connection)
patient_df
```

```
[66]: Patient_ID Insurance_ID Hospital_ID First_name Last_name Gender \
0      45098      1003      90001    Akriti      Singh      F
1      45099      1003      90003    Pritam     Bannerjee      M
2      45100      1003      90005    Salim       Khan      M
3      45101      1003      90002   Yasoof     Akhtar      M
4      45102      1003      90003   Wahida     Begum      F
..
145     ...        ...        ...      ...      ...
145     56046      1001      70002    Kelly      Griffin      F
146     56047      1001      70003    Megan      Ford      F
147     56048      1001      70004    Roger      Marshall      M
148     56049      1001      70005   Janice      Wells      M
149     56050      1001      70002   Jordan     Holmes      M

          DOB      Phone Nationality Insurance_type \
0      2002-05-09  8908908398      Indian      Longterm
1      2001-02-02  9087988092      Indian      Short term
2      2001-06-09  9900887766      Indian      Long term
3      2000-08-07  9988455767      Indian      Long term
4      2003-07-08  9090123467      Indian      Short term
..
145     ...        ...        ...      ...
145     1999-11-16  6199202675    American      Long Term
146     1992-10-23  9257669394    American      Short Term
147     1984-03-21  4436766439    American      Long Term
148     1992-04-05  5103345478    American      Long Term
149     1988-05-26  6478894305    American      Short Term

          Address  Country
0      23- asp road kolkata 64  India\r
1      12- sp Mukherjee road kolkata 87  India\r
2      34- roy road howrah 34  India\r
3      34 chellam road Chennai 01  India\r
4      46 bb roy road karnataka 04  India\r
..
145     ...        ...
145     4063 ROSEWOOD LANE GEORGIA  USA\r
146     952 STILL STREET OHIO      USA\r
147     1861 BUENA VISTA AVENUE OREGON  USA\r
148     242 YORKIE LANE CALIFORNIA  USA\r
149     132 MUDLICK ROAD WASHINGTON  USA\r

[150 rows x 12 columns]
```

```
[67]: patient_df["Country"] = patient_df['Country'].str.replace(r"\r", "")  
patient_df
```

```
[67]: Patient_ID Insurance_ID Hospital_ID First_name Last_name Gender \
0      45098      1003      90001    Akriti      Singh      F
1      45099      1003      90003    Pritam     Bannerjee      M
```

2	45100	1003	90005	Salim	Khan	M
3	45101	1003	90002	Yasoof	Akhtar	M
4	45102	1003	90003	Wahida	Begum	F
..
145	56046	1001	70002	Kelly	Griffin	F
146	56047	1001	70003	Megan	Ford	F
147	56048	1001	70004	Roger	Marshall	M
148	56049	1001	70005	Janice	Wells	M
149	56050	1001	70002	Jordan	Holmes	M

	DOB	Phone	Nationality	Insurance_type	\
0	2002-05-09	8908908398	Indian	Longterm	
1	2001-02-02	9087988092	Indian	Short term	
2	2001-06-09	9900887766	Indian	Long term	
3	2000-08-07	9988455767	Indian	Long term	
4	2003-07-08	9090123467	Indian	Short term	
..	
145	1999-11-16	6199202675	American	Long Term	
146	1992-10-23	9257669394	American	Short Term	
147	1984-03-21	4436766439	American	Long Term	
148	1992-04-05	5103345478	American	Long Term	
149	1988-05-26	6478894305	American	Short Term	

	Address	Country
0	23- asp road kolkata 64	India
1	12- sp Mukherjee road kolkata 87	India
2	34- roy road howrah 34	India
3	34 chellam road Chennai 01	India
4	46 bb roy road karnataka 04	India
..
145	4063 ROSEWOOD LANE GEORGIA	USA
146	952 STILL STREET OHIO	USA
147	1861 BUENA VISTA AVENUE OREGON	USA
148	242 YORKIE LANE CALIFORNIA	USA
149	132 MUDLICK ROAD WASHINGTON	USA

[150 rows x 12 columns]

```
[68]: cursor=connection.cursor()
```

```
[69]: patient_df["Name"] = patient_df["First_name"] + " " + patient_df["Last_name"]
patient_df
```

```
[69]: Patient_ID Insurance_ID Hospital_ID First_name Last_name Gender \
0      45098      1003      90001   Akriti     Singh     F
1      45099      1003      90003   Pritam   Bannerjee   M
2      45100      1003      90005   Salim     Khan     M
```

3	45101	1003	90002	Yasoof	Akhtar	M
4	45102	1003	90003	Wahida	Begum	F
..
145	56046	1001	70002	Kelly	Griffin	F
146	56047	1001	70003	Megan	Ford	F
147	56048	1001	70004	Roger	Marshall	M
148	56049	1001	70005	Janice	Wells	M
149	56050	1001	70002	Jordan	Holmes	M

	DOB	Phone	Nationality	Insurance_type	\
0	2002-05-09	8908908398	Indian	Longterm	
1	2001-02-02	9087988092	Indian	Short term	
2	2001-06-09	9900887766	Indian	Long term	
3	2000-08-07	9988455767	Indian	Long term	
4	2003-07-08	9090123467	Indian	Short term	
..
145	1999-11-16	6199202675	American	Long Term	
146	1992-10-23	9257669394	American	Short Term	
147	1984-03-21	4436766439	American	Long Term	
148	1992-04-05	5103345478	American	Long Term	
149	1988-05-26	6478894305	American	Short Term	

	Address	Country	Name
0	23- asp road kolkata 64	India	Akriti Singh
1	12- sp Mukherjee road kolkata 87	India	Pritam Bannerjee
2	34- roy road howrah 34	India	Salim Khan
3	34 chellam road Chennai 01	India	Yasoof Akhtar
4	46 bb roy road karnataka 04	India	Wahida Begum
..
145	4063 ROSEWOOD LANE GEORGIA	USA	Kelly Griffin
146	952 STILL STREET OHIO	USA	Megan Ford
147	1861 BUENA VISTA AVENUE OREGON	USA	Roger Marshall
148	242 YORKIE LANE CALIFORNIA	USA	Janice Wells
149	132 MUDLICK ROAD WASHINGTON	USA	Jordan Holmes

[150 rows x 13 columns]

```
[70]: patient_df=patient_df.drop('First_name',axis=1)
patient_df=patient_df.drop('Last_name',axis=1)
patient_df
```

```
[70]: Patient_ID Insurance_ID Hospital_ID Gender          DOB      Phone \
0        45098       1003     90001      F  2002-05-09  8908908398
1        45099       1003     90003      M  2001-02-02  9087988092
2        45100       1003     90005      M  2001-06-09  9900887766
3        45101       1003     90002      M  2000-08-07  9988455767
4        45102       1003     90003      F  2003-07-08  9090123467
```

..
145	56046	1001	70002	F	1999-11-16	6199202675	
146	56047	1001	70003	F	1992-10-23	9257669394	
147	56048	1001	70004	M	1984-03-21	4436766439	
148	56049	1001	70005	M	1992-04-05	5103345478	
149	56050	1001	70002	M	1988-05-26	6478894305	

	Nationality	Insurance_type		Address	Country	\
0	Indian	Longterm	23- asp road	kolkata 64	India	
1	Indian	Short term	12- sp Mukherjee road	kolkata 87	India	
2	Indian	Long term	34- roy road howrah	34	India	
3	Indian	Long term	34 chellam road	Chennai 01	India	
4	Indian	Short term	46 bb roy road	karnataka 04	India	
..	
145	American	Long Term	4063 ROSEWOOD LANE	GEORGIA	USA	
146	American	Short Term	952 STILL STREET	OHIO	USA	
147	American	Long Term	1861 BUENA VISTA AVENUE	OREGON	USA	
148	American	Long Term	242 YORKIE LANE	CALIFORNIA	USA	
149	American	Short Term	132 MUDLICK ROAD	WASHINGTON	USA	

	Name
0	Akriti Singh
1	Pritam Bannerjee
2	Salim Khan
3	Yasoof Akhtar
4	Wahida Begum
..	..
145	Kelly Griffin
146	Megan Ford
147	Roger Marshall
148	Janice Wells
149	Jordan Holmes

[150 rows x 11 columns]

```
[71]: patient_df["Address"] = patient_df["Address"] + " "+patient_df["Country"]
patient_df=patient_df.drop('Country',axis=1)
patient_df
```

	Patient_ID	Insurance_ID	Hospital_ID	Gender	DOB	Phone	\
0	45098	1003	90001	F	2002-05-09	8908908398	
1	45099	1003	90003	M	2001-02-02	9087988092	
2	45100	1003	90005	M	2001-06-09	9900887766	
3	45101	1003	90002	M	2000-08-07	9988455767	
4	45102	1003	90003	F	2003-07-08	9090123467	
..	
145	56046	1001	70002	F	1999-11-16	6199202675	

146	56047	1001	70003	F	1992-10-23	9257669394
147	56048	1001	70004	M	1984-03-21	4436766439
148	56049	1001	70005	M	1992-04-05	5103345478
149	56050	1001	70002	M	1988-05-26	6478894305

	Nationality	Insurance_type		Address	\
0	Indian	Longterm	23-	asp road kolkata 64	India
1	Indian	Short term	12- sp Mukherjee road kolkata 87	India	
2	Indian	Long term	34- roy road howrah 34	India	
3	Indian	Long term	34 chellam road Chennai 01	India	
4	Indian	Short term	46 bb roy road karnataka 04	India	
..	
145	American	Long Term	4063 ROSEWOOD LANE	GEORGIA USA	
146	American	Short Term	952 STILL STREET	OHIO USA	
147	American	Long Term	1861 BUENA VISTA AVENUE	OREGON USA	
148	American	Long Term	242 YORKIE LANE	CALIFORNIA USA	
149	American	Short Term	132 MUDLICK ROAD	WASHINGTON USA	

	Name
0	Akriti Singh
1	Pritam Bannerjee
2	Salim Khan
3	Yasoof Akhtar
4	Wahida Begum
..	..
145	Kelly Griffin
146	Megan Ford
147	Roger Marshall
148	Janice Wells
149	Jordan Holmes

[150 rows x 10 columns]

```
[74]: patient_df.to_csv('PATIENT_TRANSFORMED.csv')
```

```
[ ]:
```

```
[19]:
```

```
[ ]:
```

```
[18]:
```

```
[ ]:
```

```
[76]: insurer_df=pd.read_sql_query("SELECT * FROM INSURER",connection)
insurer_df
```

```
[76]: 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
```

```
[77]: insurer_df['Handling_risk']=insurer_df['Company_name']+insurer_df['Insurance_name']
insurer_df=insurer_df.drop('Company_name',axis=1)
insurer_df=insurer_df.drop('Insurance_name',axis=1)
insurer_df
```

```
[77]: Insurance_ID  Start_date  End_date Effective_date Insurance_Status  Term \
0      1001  2020-01-01  2036-01-01    2020-01-15        200000      0
1      1002  2020-01-03  2039-01-03    2020-03-15        300000      0
2      1003  2020-02-05  2041-02-05    2020-05-17        400000      0

Topup_amount  Insurance_Cover \
0            15      200000
1            18      300000
2            20      150000

Handling_risk
0  Aegon Life Insurance Co. Ltd.Individual Health...
1  Edelweiss Life Insurance Co.Family Health Ins...
2  Birla Sun Life Insurance Co.Critical illness I...
```

```
[22]: Insurance
```

```
[22]:           Insurer
0  Individual Health Insurance , Aegon Life Insur...
1  Family Health Insurance , Edelweiss Life Insu...
2  Critical illness Insurance , Birla Sun Life In...
```

```
[23]: df=pd.read_sql_query("SELECT * FROM CLAIM",connection)
print(df.head())
```

```
Patient_ID  Insurance_ID  Claim_ID  Group_ID      Claim_Type  Issue_date \
0      45104        1003      4007      76895    Cashless  2021-12-12
```

```

1      45106        1003      4009      76894 Reimbursement 2022-05-09
2      45111        1003      4014      76893 Cashless    2022-01-03
3      45113        1003      4016      76895 Reimbursement 2021-11-12
4      45117        1003      4020      76891 Reimbursement 2021-11-04

   Risk_category Premium_plan_type Claim_Amount Handling_type Settlement \
0      High           Monthly       20000     Complex    In Process
1      High      Semi-annualy      29000   Accelerated Invalid
2      Low           Monthly       35000     Standard   Settled
3      High           Monthly       20500     Standard   Settled
4      Low           Quaterly      24100     Standard   Settled

   Claim_status Submission_method Country
0      ACTIVE          Internet  India\r
1      INVALID          Mail     India\r
2      RECOVERED         Internet  India\r
3      ACTIVE          Internet  India\r
4      RECOVERED         Mail     India\r

/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 other DBAPI2 objects are not
tested, please consider using SQLAlchemy
    warnings.warn(

```

```
[24]: CLAIMS=pd.DataFrame({'Claim': df['Settlement'] + " , " + df['Claim_status']})
```

```
[25]: CLAIMS
```

```

[25]:          Claim
0      In Process , ACTIVE
1      Invalid , INVALID
2      Settled , RECOVERED
3      Settled , ACTIVE
4      Settled , RECOVERED
5      In Process , ACTIVE
6      In Process , ACTIVE
7      Invalid , INVALID
8      In Process , ACTIVE
9      Settled , RECOVERED
10     Invalid , INVALID
11     In Process , ACTIVE
12     Invalid , INVALID
13     Settled , RECOVERED
14     In Process , RECOVERED
15     In Process , ACTIVE
16     Settled , RECOVERED
17     Settled , RECOVERED

```

```
18      In Process , ACTIVE
19      In Process , ACTIVE
20      In Process , ACTIVE
21          Invalid , INVALID
22          Settled , ACTIVE
23  In Process , ACTIVE REVIEW
24          Invalid , INVALID
25          Settled , RECOVERED
26      In Process , ACTIVE
27          Settled , RECOVERED
28          Invalid , INVALID
29          Invalid , ACTIVE
30          Settled , RECOVERED
31          Settled , RECOVERED
32          Invalid , INVALID
33          Invalid , INVALID
34          Process , ACTIVE
35          Process , ACTIVE
36          Settled , RECOVERED
37          Settled , RECOVERED
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
```

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

```
[27]: CLAIMS
```

```
[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
43	5005
44	5008
45	5013
46	5018
47	5021
48	5022

```
49  5029  
50  5032  
51  5035  
52  5038  
53  5041  
54  5044  
55  5047  
56  5050
```

```
[28]: CLAIMS=pd.DataFrame({'Claim': df['Issue_date']})
```

```
[29]: CLAIMS
```

```
[29]:      Claim  
0    2021-12-12  
1    2022-05-09  
2    2022-01-03  
3    2021-11-12  
4    2021-11-04  
5    2022-04-30  
6    2022-05-18  
7    2022-04-21  
8    2022-05-14  
9    2022-04-07  
10   2022-04-01  
11   None  
12   2022-03-31  
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
```

```
[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 other DBAPI2 objects are not
tested, please consider using SQLAlchemy
  warnings.warn(
```

```
[88]: 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 other DBAPI2 objects are not
tested, please consider using SQLAlchemy
    warnings.warn(
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```