USE CASE STUDY REPORT

Group No: 8

Student Names: Navya Pragathi Molugaram and Sanjana Konte

Executive Summary:

The main goal of this study was to design and implement a relational database model that is ready for use by a University's Health Center to analyse the health insurance spend patterns of the students, available policies other than the university policy that adhere to the guidelines. Students of a university enrolled in a part-time or full-time course structure are required to participate in a health insurance policy according to the Massachusetts law. The University has a set of guidelines to accept a policy to be considered as an potential insurance cover. Though the university has clear guidelines listed on their site, there exists no database which shows the student spends, network hospitals, potential outside policy providers to be considered or a comparison between various health insurance providers. Thus, we aim to model a relational database as a solution.The database also implements an analytics platform that could help the students and the university to understand their needs and cater a well rounded policy benefitting the students.

The database was designed by collecting the requirements of the policy that is eligible to be considered as an insurance cover to students. The data from the available policies, providers and the network hospitals was also gathered. The EER and UML diagrams were developed, followed by the mapping of the conceptual model to a relational model with the required constraints (foreign key, primary key). The database was fully implemented on MySQL and to study the feasibility of this database in a No SQL environment , a prototype with two tables and a relationship was implemented in the Neo4j NoSQL graph environment.

The database was successfully created and a connection has been established to the Python environment. Using Python- Seaborn and Matplotlib library the analytics capabilities were immense, some of which have been shown in the study. These queries can be very helpful to track the insurance spends. With the increasing trend of medical expenses, the database would also be helpful to students to find an economical insurance cover. The next stage of improvement would be to implement a front end UX interface of the database and a check to make sure that the created database system adheres to the health regulations set by the University.

I. Introduction:

Health care in the United States is a vital system that includes almost everyone with health insurance. Health insurance is an arrangement made to pay some or all unexpected medical costs. A student enrolled at any University in Massachusetts should choose a policy offered by their university or from any other provider offering similar coverage.Comparing and choosing the right health insurance is crucial for students if opting out of university health insurance.

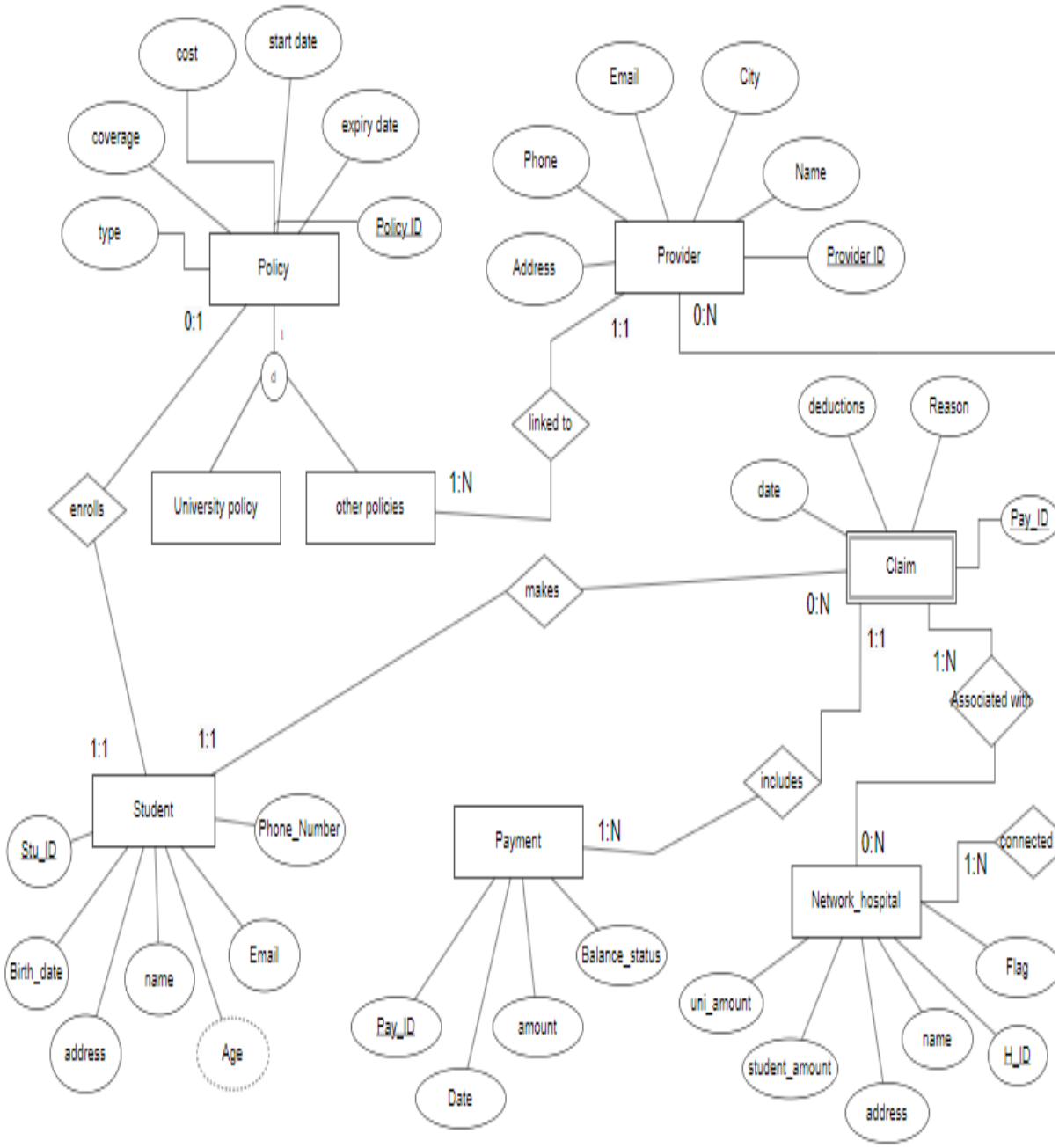
Northeastern University Student Health Plan(NUSHP) is the university's health insurance that is administered by Blue Cross Blue Shield of Massachusetts. It ensures all-over coverage(co-op, intern, or study abroad), various medical care at reduced costs, and round-the-clock emergency medical care and help. At present the university automatically enrols students into the policy and gives a option to unenroll and choose a different provider given it satisfies the rules. The relational database thus created will help the students to have access to all the other policy details at hand if they prefer to choose an outside provider. We've considered factors like coverage, cost, policy period, type of health policy and a unique policy ID to identify potential policies. Our database records the provider's details such as their unique company ID, Name, Email, Phone Number, City and Address. To understand the claim trends of students across the university, we are considering claim information, payment details, and corresponding hospitals in which they availed the services.

The claim information includes the unique claim Id, reason, Date, and Deductions. A student can make a claim at any of the network hospitals. For the hospitals we record the hospital ID, Name, Address, Student Amount, University Amount. A claim is processed only if a payment is done. The payment is tracked using a payment ID, date of payment, payment amount, and Balance status. We record student information which includes student ID, Name, Email ID, Phone number, Address, Age, and Nationality.

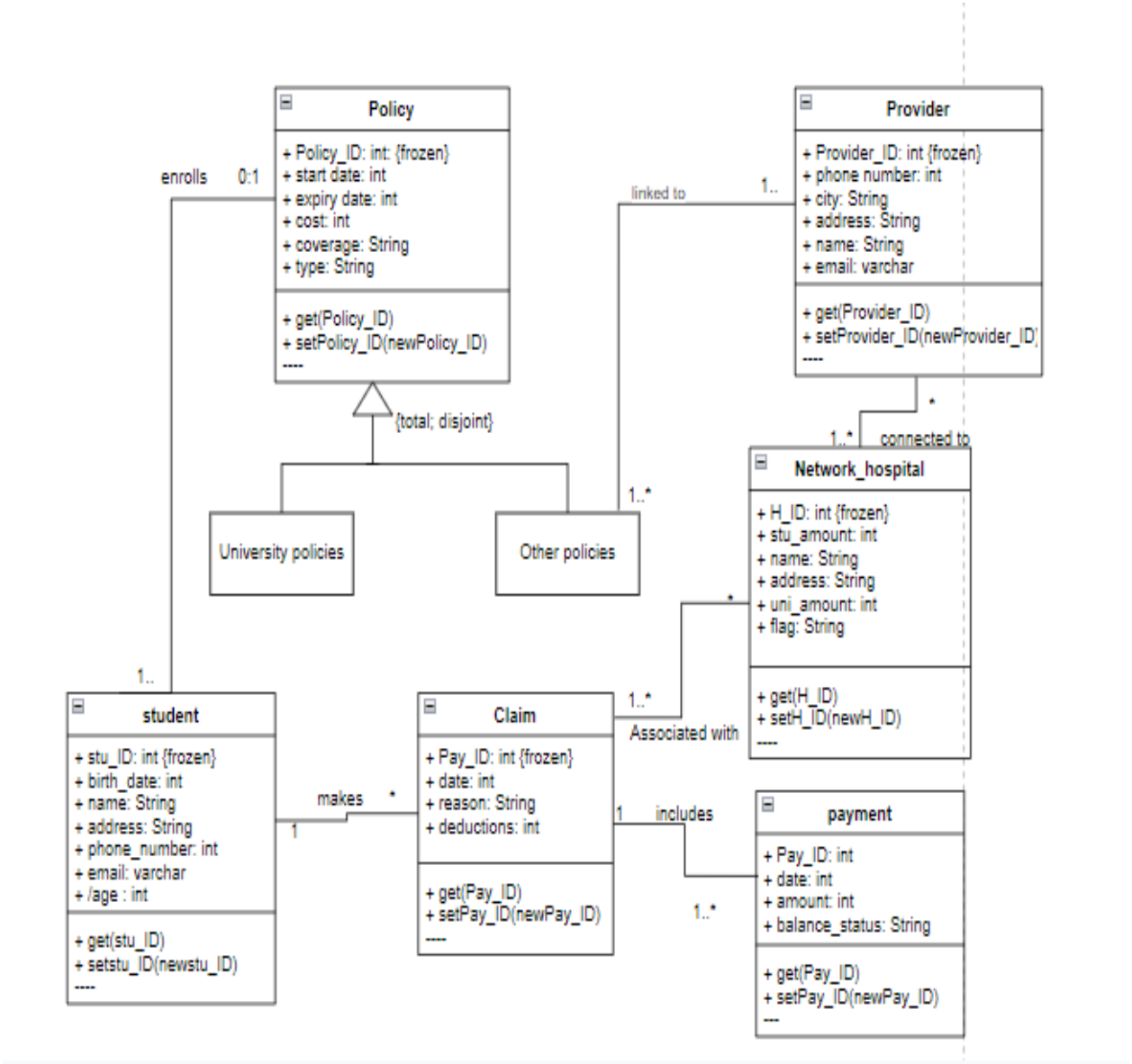
Thus, a relational database model is used to design the above structure. For a better understanding of the usage of each policy. The policy has been divided into two specialisations - university policy that represents the policy by default considered by a university and other policy that represents all the policies other than the university policy. All the data tables combined with their relationships gives a simple yet comprehensive database that provides a solution to the problem at hand in an efficient and structured manner.

II. Conceptual Data Modelling

1. EER Diagram



2. UML Diagram



III. Mapping Conceptual Model to Relational Model

Primary Key - <u>Underlined</u> **Foreign Key - *<u>Italicized</u>***

Policy(<u>policyID</u>, type, coverage, cost, startdate, expirydate,*<u>holderID</u>*)

- Foreign key holderID refers to studID from Student, NULL NOT ALLOWED

UniversityPolicy(*<u>unipolicyID</u>*)

- Foreign key unipolicyID refers to policyID from policy, NULL NOT ALLOWED

OtherPolicy(*<u>otherpolicyID</u>*, *<u>providerID</u>*)

- Foreign key otherpolicyID refers to policyID from policy, NULL NOT ALLOWED.
- Foreign key providerID refers to providerID from Provider, NULL NOT ALLOWED.

Provider(providerID, name, city, email, phone, address)
Student (studID, name, address, dateofbirth, email, phone,age)
Payment (payID, date, amount, balance_status)
Claim (*claimpayID*, date,deductions, reason, *studclaimID, hospID*)

- Foreign key claimpayID refers to the payID from Payment, NULL NOT ALLOWED
- Foreign key studclaimID refers to the studID from Student, NULL NOT ALLOWED
- Foreign key hospID refers to hospitalID from Network Hospital, NULL NOT ALLOWED.

NetworkHospital (hospitalID, name, address, student_amount, policy_amount)
connects(*connhospitalID, connproviderID*)

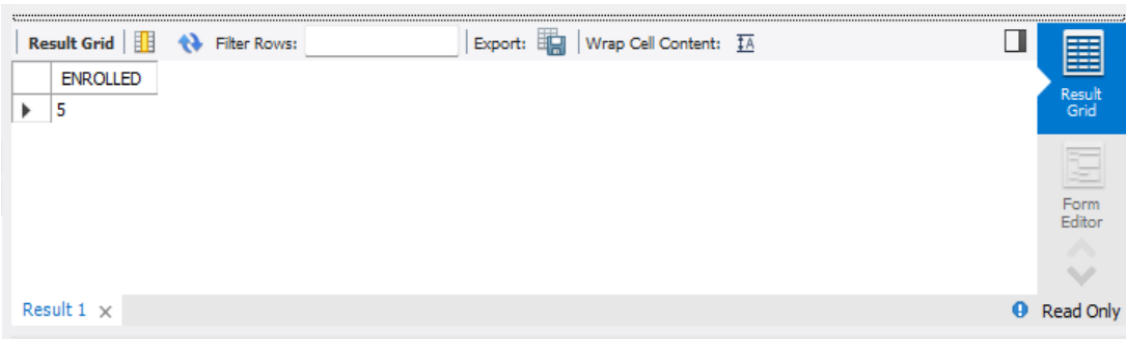
- Foreign key connhospitalID refers to hospitalID from NetworkHospital, NULL NOT ALLOWED .
- Foreign key connproviderID refers to providerID from Provider, NULL NOT ALLOWED.

IV. Implementation of Relation Model via MySQL and NoSQL

MySQL Implementation:
The database was created in MySQL and the following queries were performed:

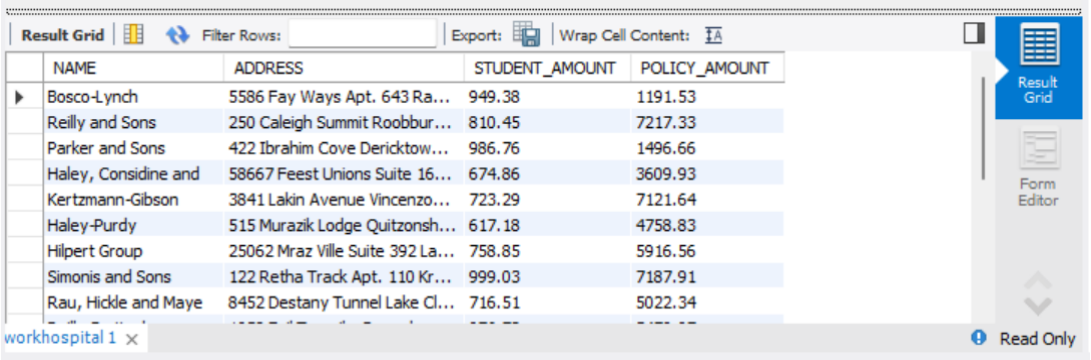
Query1 : Find the number of students who have enrolled in University Policy (Queries like these can be used to track the student behaviour on their preference of insurance)

```
SELECT count(*) AS ENROLLED FROM STUDENT
WHERE EXISTS (SELECT * FROM POLICY ,university_policy
WHERE university_policy.UNIPOLICYID=policy.POLICYID
AND policy.HOLDERID = student.STUDID) ;
```



Query2: Retrieve the name, address, student_amount and policy_amount from the Network hospital which are owned by other policyholders.

```
SELECT NAME,ADDRESS,STUDENT_AMOUNT,POLICY_AMOUNT
FROM networkhospital  AS N
WHERE N.HOSPITALID IN (SELECT CONNHOSPITALID FROM connects
WHERE CONNPROVIDERID IN (SELECT PROVIDERID FROM other_policy));
```



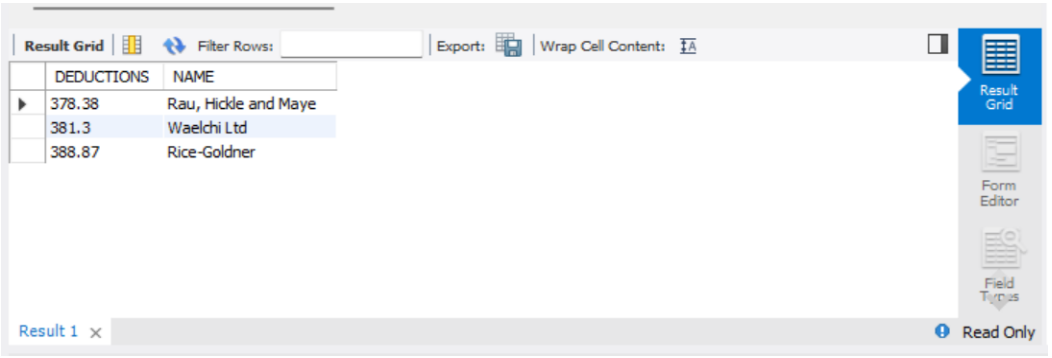
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

NAME	ADDRESS	STUDENT_AMOUNT	POLICY_AMOUNT
Bosco-Lynch	5586 Fay Ways Apt. 643 Ra...	949.38	1191.53
Reilly and Sons	250 Caleigh Summit Roobbur...	810.45	7217.33
Parker and Sons	422 Ibrahim Cove Dericktow...	986.76	1496.66
Haley, Considine and	58667 Feest Unions Suite 16...	674.86	3609.93
Kertzmann-Gibson	3841 Lakin Avenue Vincenzo...	723.29	7121.64
Haley-Purdy	515 Murazik Lodge Quitzonsh...	617.18	4758.83
Hilpert Group	25062 Mraz Ville Suite 392 La...	758.85	5916.56
Simonis and Sons	122 Retha Track Apt. 110 Kr...	999.03	7187.91
Rau, Hickle and Maye	8452 Destany Tunnel Lake Cl...	716.51	5022.34

Result Grid | Form Editor | workhospital 1 | Read Only

Query3: Find the top three claims made by students and the corresponding network hospital where the claim was made.

```
SELECT C1.DEDUCTIONS, N.NAME
FROM claim AS C1 inner join networkhospital as N
 on c1.HOSPID = N.HOSPITALID
WHERE 3 > (SELECT COUNT(*)
FROM claim AS C2
WHERE C1.DEDUCTIONS < C2.DEDUCTIONS )
ORDER BY C1.DEDUCTIONS;
```



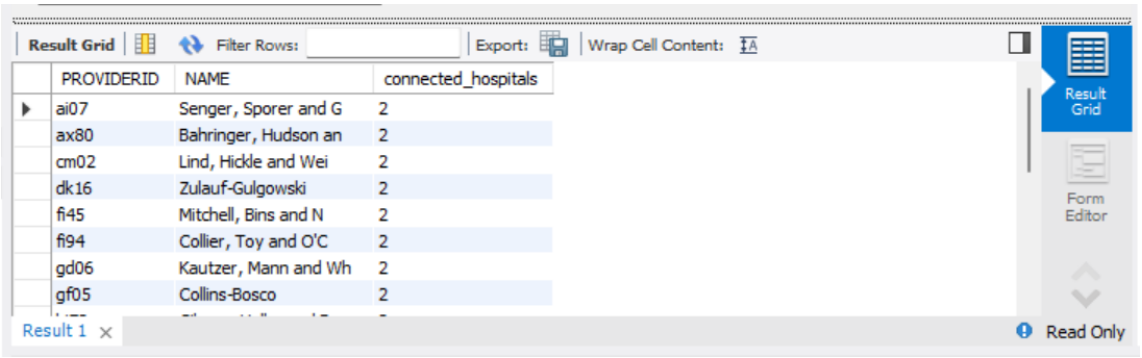
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

DEDUCTIONS	NAME
378.38	Rau, Hickle and Maye
381.3	Waelchi Ltd
388.87	Rice-Goldner

Result Grid | Form Editor | Field Types | Result 1 | Read Only

Query4: Find the total number of network hospitals for each provider.

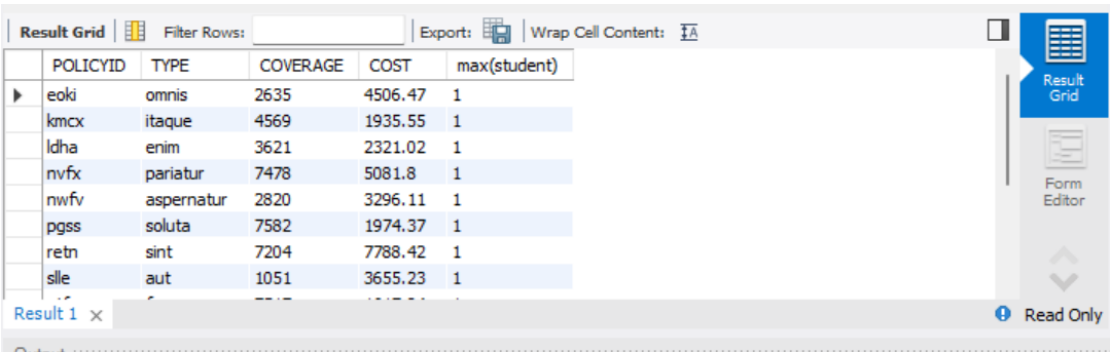
```
select PROVIDERID, NAME, count(*) as connected_hospitals
from provider as p inner join connects as c
where p.PROVIDERID = c.CONNPROVIDERID
group by p.PROVIDERID;
```



PROVIDERID	NAME	connected_hospitals
ai07	Senger, Sporer and G	2
ax80	Bahringer, Hudson an	2
cm02	Lind, Hickle and Wei	2
dk16	Zulauf-Gulgowski	2
fi45	Mitchell, Bins and N	2
fi94	Collier, Toy and O'C	2
gd06	Kautzer, Mann and Wh	2
gf05	Collins-Bosco	2

Query5: Find the policy name, policy type, coverage and policy period of the highest opted policy among students.

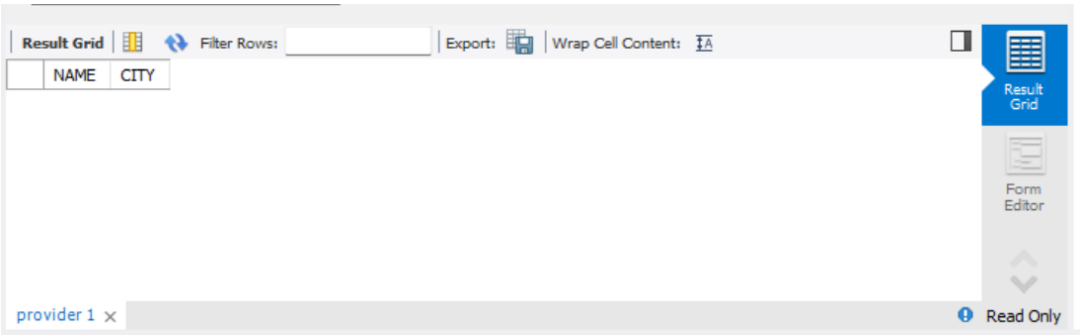
```
with cte as (
select POLICYID, count(HOLDERID) as student
from policy
group by TYPE)
select cte.POLICYID,policy.TYPE, policy.COVERAGE, policy.COST, max(student)
from cte,policy
where cte.POLICYID = policy.POLICYID
group  by POLICYID;
```



POLICYID	TYPE	COVERAGE	COST	max(student)
eoki	omnis	2635	4506.47	1
kmcx	itaque	4569	1935.55	1
ldha	enim	3621	2321.02	1
nvfx	pariatur	7478	5081.8	1
nwfv	aspernatur	2820	3296.11	1
pgss	soluta	7582	1974.37	1
retn	sint	7204	7788.42	1
slle	aut	1051	3655.23	1

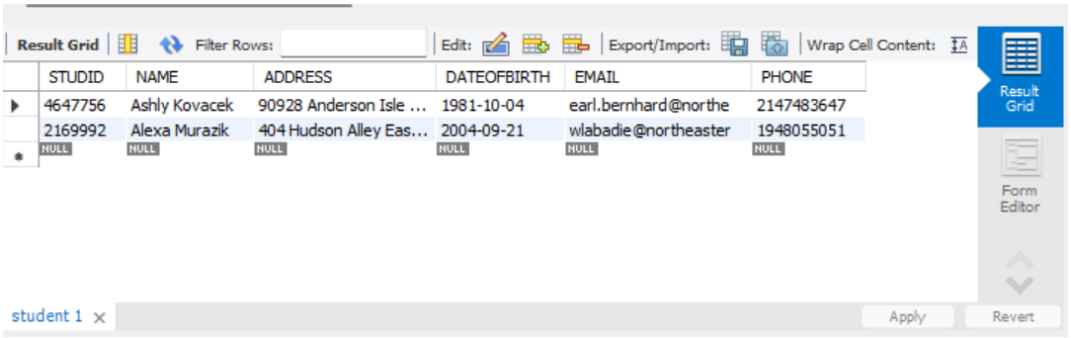
Query6: Retrieve the provider details of the policies opted by none of the students.

```
SELECT NAME,CITY FROM provider
WHERE EXISTS (SELECT * FROM other_policy
WHERE provider.PROVIDERID = other_policy.PROVIDERID
AND EXISTS (SELECT * FROM policy
WHERE policy.POLICYID= other_policy.OTHERPOLICYID
AND NOT EXISTS
(SELECT * from student where policy.HOLDERID = student.STUDID) ))
```



Query7: Find the student details who has made less than 10$ worth claim.

```
select * from student
where STUDID In ( select STUDENTCLAIMID from claim
where DEDUCTIONS < 10
and  claim.STUDENTCLAIMID = student.STUDID);
```

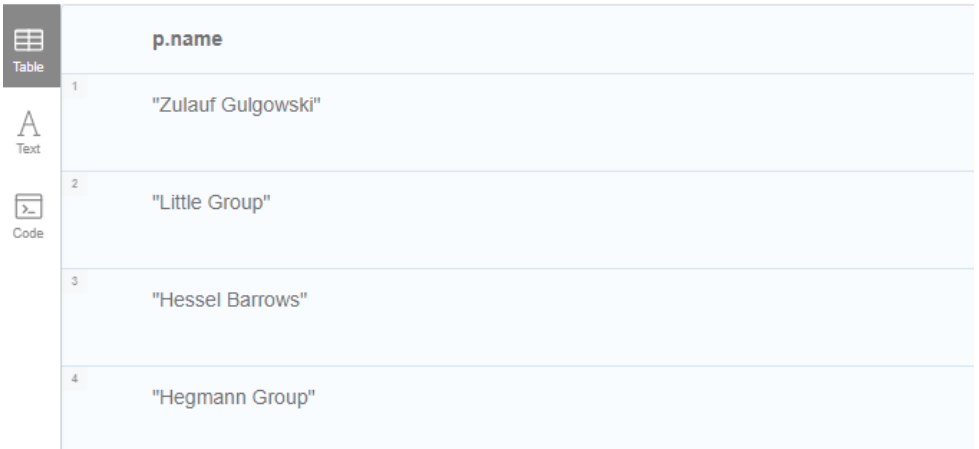


NoSQL Implementation:

Two tables(provider, network hospital) and a relation(connects) have been created in Neo4j playground. The following Cypher queries were done:

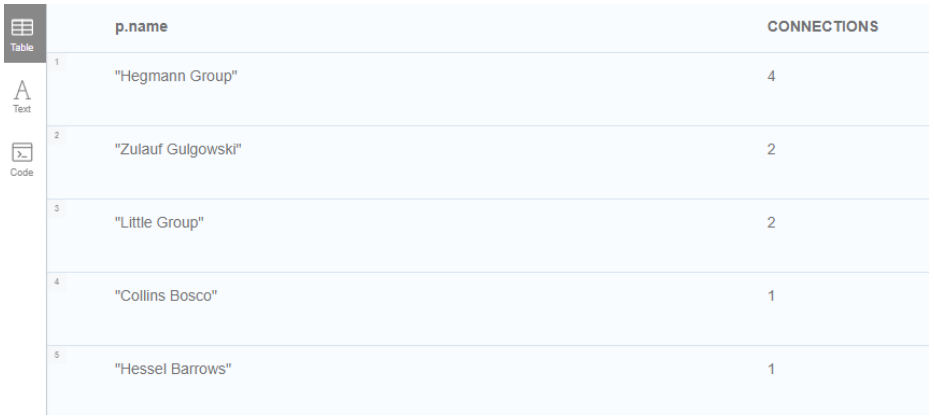
Query1: Find all the providers from the city Boston

```
MATCH (p:PROVIDER) WHERE p.city = 'Boston' RETURN p.name
```



Query2: Find the total number of network hospitals for each provider and arrange the providers from most to least.

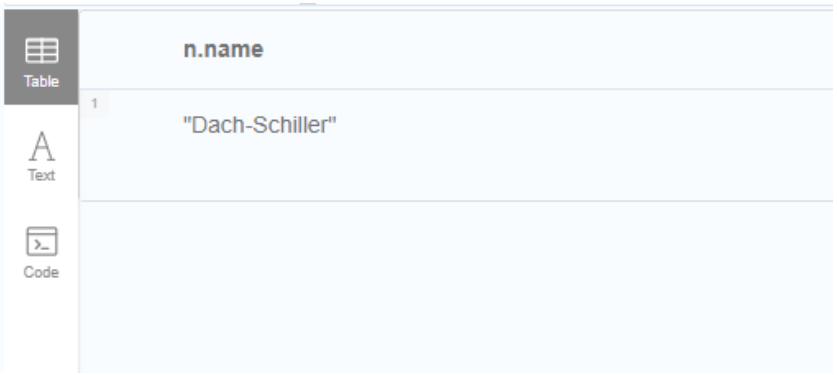
```
MATCH (p:PROVIDER)--(n:NETWORKHOSPITAL)
RETURN p.name, COUNT(*) AS CONNECTIONS
ORDER BY CONNECTIONS DESC
```



	p.name	CONNECTIONS
1	"Hegmann Group"	4
2	"Zulauf Gulgowski"	2
3	"Little Group"	2
4	"Collins Bosco"	1
5	"Hessel Barrows"	1

Query3: Find the name of the hospitals that has a provider not in Boston and student amount exceeds $650

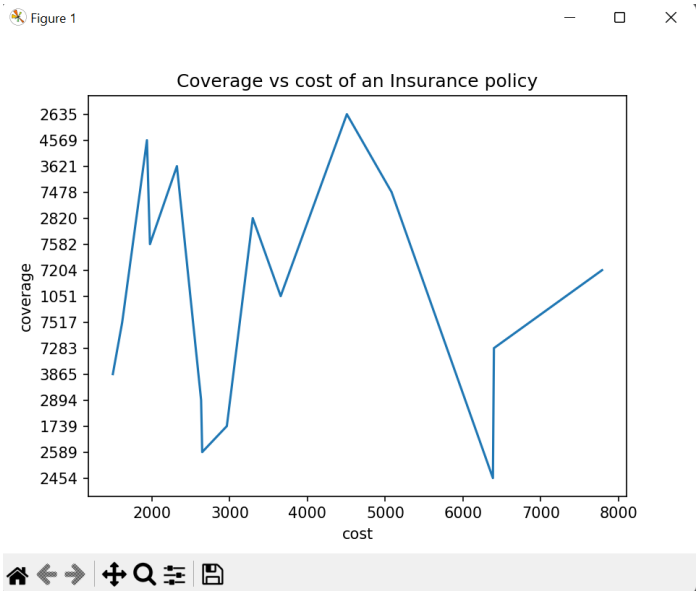
```
MATCH (p:PROVIDER) –(n:NETWORKHOSPITAL)
WHERE NOT p.city = ‘Boston’
AND n.student_amount > 650
RETURN n.name
```

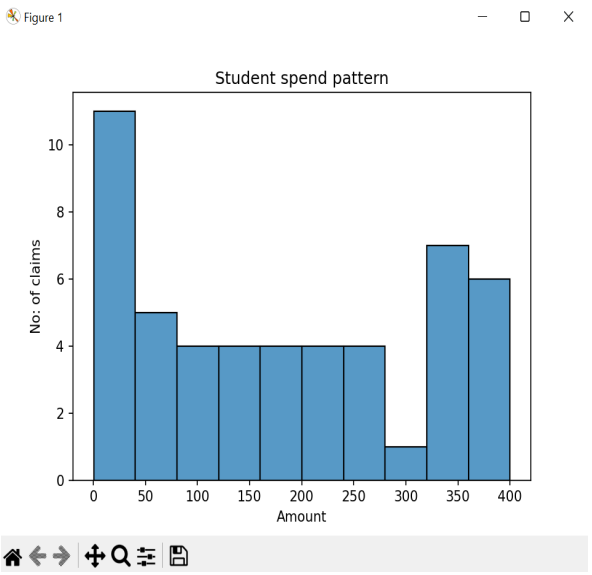


	n.name
1	"Dach-Schiller"

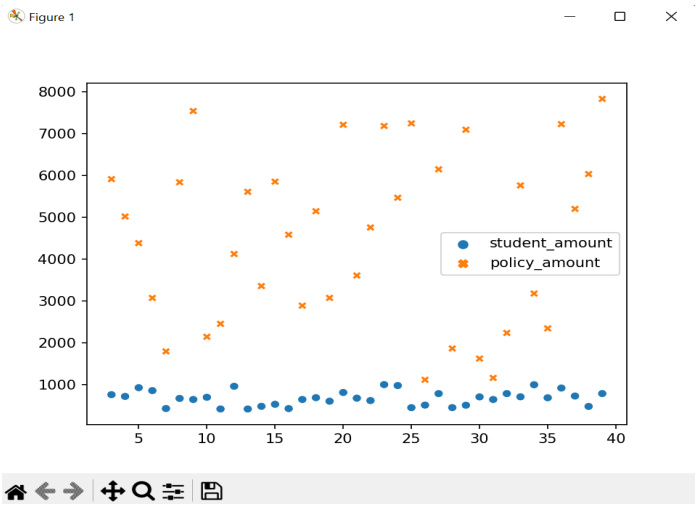
V. Database Access via Python

The database is accessed using Python and visualisation of analysed data is shown below. The connection of MySQL to Python is done using mysql.connector, followed by cursor.excecute to run and fetchall from query, followed by converting the list into a dataframe using pandas library and using matplotlib and seaborn to plot the graphs for the analytics.





Scatter plot of student amount vs coverage amount.



VII. Summary

The health insurance database created on MySQL as a relational model can be implemented by Universities. It will facilitate a great understanding of the university guidelines of the alternate health policy and provide the details of providers to students, a small part of which is shown in this report using Python.
The database model can be improved by adding data governance measures to ensure data quality as the policy changes happen over time.