HEALTH INSURANCE DATABASE SYSTEM

Comprehensive Project Report

Group 23

ARYAK BODKHE

SHREYANS THESIA

bodkhe.a@northeastern.edu

thesia.s@northeastern.edu

Submission Date: 12/9/2022

Business Problem:

Many overseas students pursue their dreams of attending university in the United States each year. Dueto universities requiring students to have approved health insurance, students must obtain decent health insurance that covers treatment for most common diseases and injuries, where the student pays less, andthe insurance company covers most of the treatment costs. Students frequently lack access to resourceswhere they can view and compare the best-priced plans that are necessary for them. In order to addressthis issue, we are going to develop a health insurance database system where we can supply all requiredinformation about all insurance plans, and which would be the best option depending on student desire.

Our DBMS will provide access to various health insurance plans and affiliated hospitals and the cost of treatment for various diseases and injuries so that students can find the appropriate health plan in one place in their budget.

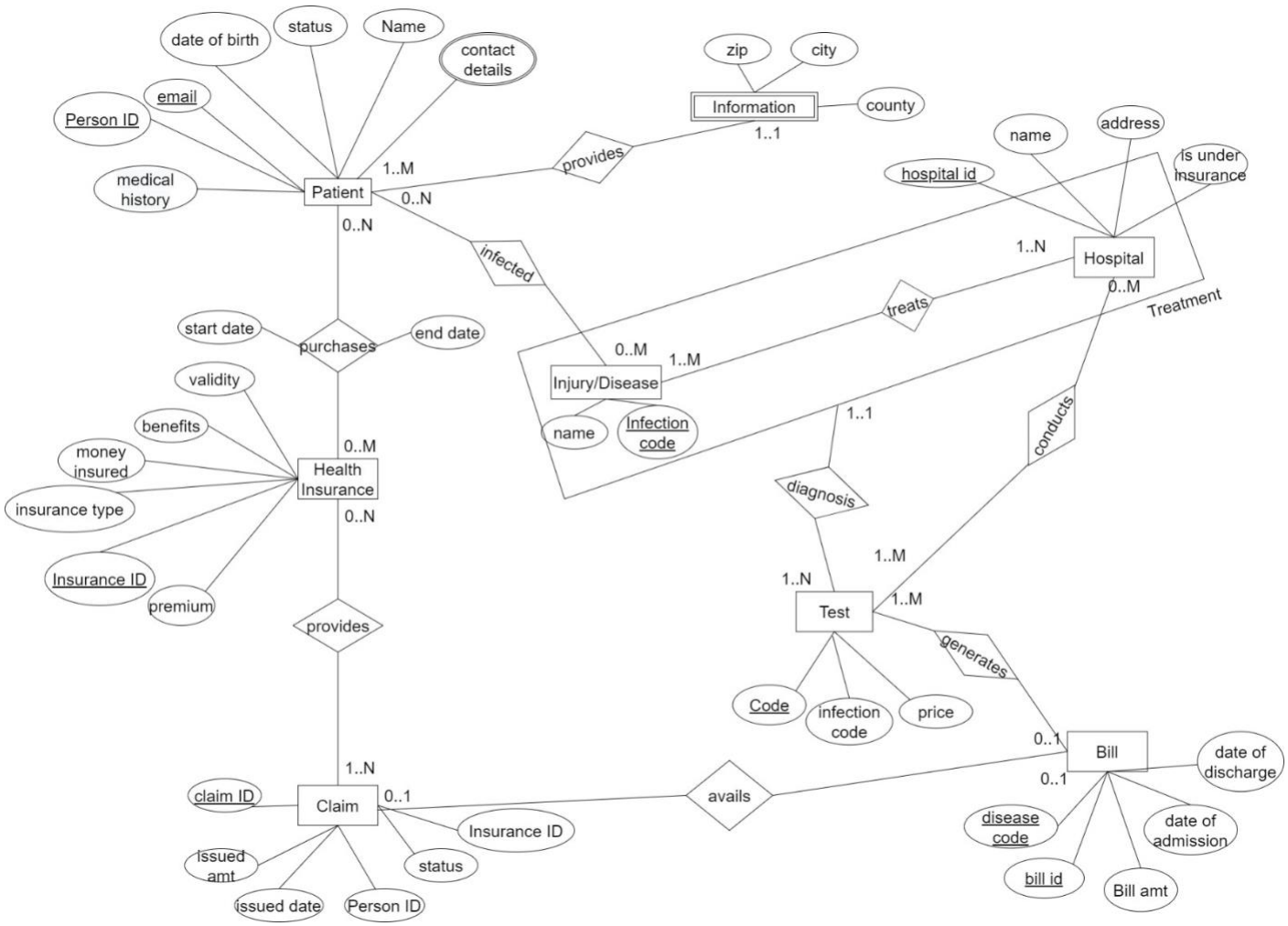
In the model if a person- student or a foreign visitor who comes to an accident and does not have any insurance, but he is a patient as he/she is injured.

We can claim the health insurance multiple times and a person can have multiple insurances such as generic health insurance, dental insurance, eye insurance etc.

Business Entities Considered in our Model:

- Patient
- Information
- Health Insurance
- Claim
- Injury/Disease
- Hospital
- Test
- Bill

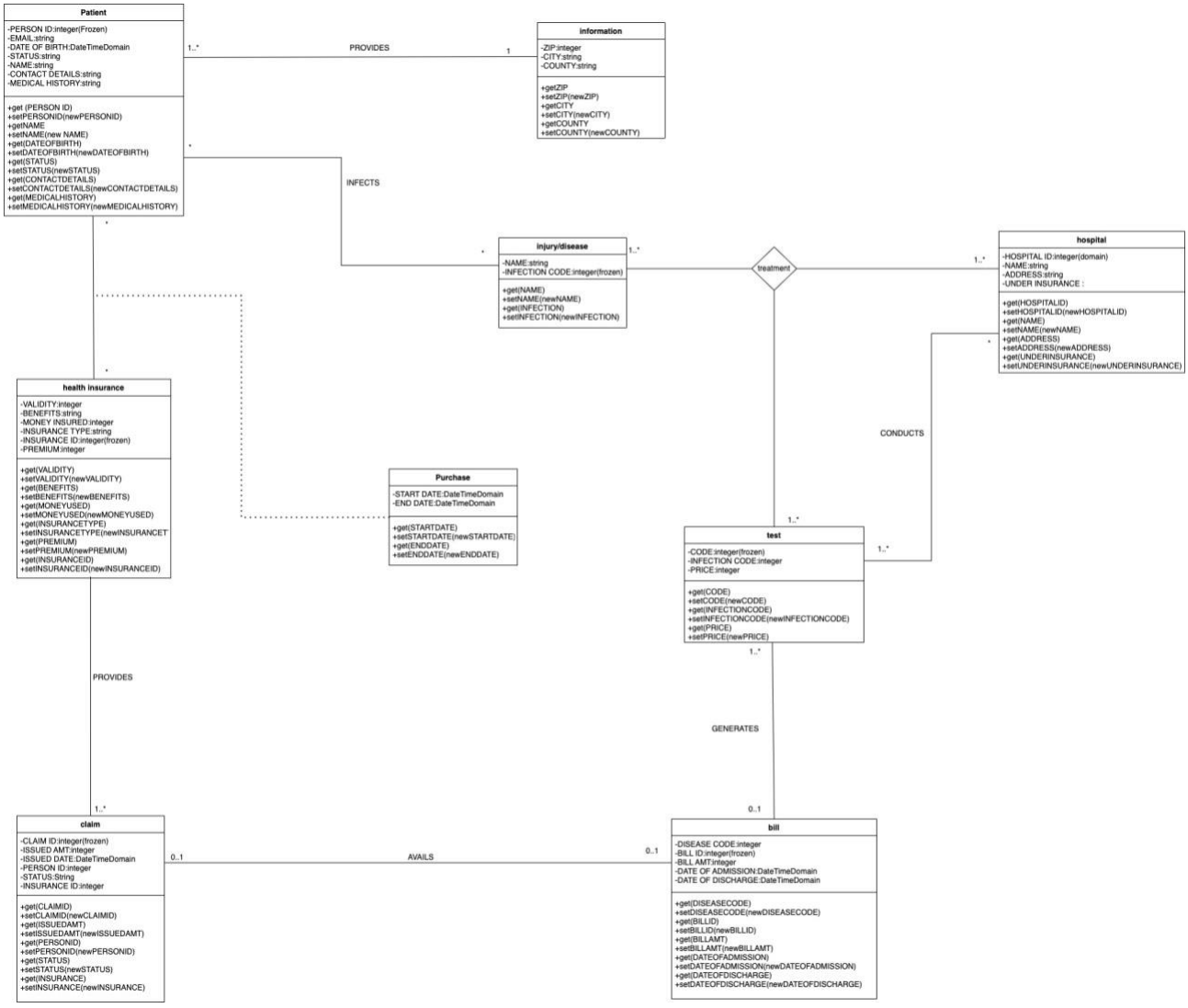
EER Diagram:



Here we have reference data in injury/disease entity where we have stored information about tests required for any disease in the database.

Here when health insurances are sold, the transaction referenced to the relevant Insurance ID and Patient ID. The insurance and patient records do not need to be modified for the new transactions.

UML Diagram:


Patient
-PERSON ID:integer(Frozen)
-EMAIL:string
-DATE OF BIRTH:DateTimeDomain
-STATUS:string
-NAME:string
-CONTACT DETAILS:string
-MEDICAL HISTORY:string
+get (PERSON ID)
+setPERSONID(newPERSONID)
+getNAME
+setNAME(new NAME)
+get(DATEOFBIRTH)
+set(DATEOFBIRTH(newDATEOFBIRTH)
+get(STATUS)
+setSTATUS(newSTATUS)
+get(CONTACTDETAILS)
+set(CONTACTDETAILS(newCONTACTDETAILS)
+get(MEDICALHISTORY)
+setMEDICALHISTORY(newMEDICALHISTORY)
1..*
PROVIDES
1
information
-ZIP:integer
-CITY:string
-COUNTY:string
+getZIP
+setZIP(newZIP)
+getCITY
+setCITY(newCITY)
+getCOUNTY
+setCOUNTY(newCOUNTY)
*
INFECTS
*
injury/disease
-NAME:string
-INFECTION CODE:integer(frozen)
+get(NAME)
+setNAME(newNAME)
+get(INFECTION)
+setINFECTION(newINFECTION)
1..*
treatment
1..*
hospital
-HOSPITAL ID:integer(domain)
-NAME:string
-ADDRESS:string
-UNDER INSURANCE :
+get(HOSPITALID)
+setHOSPITALID(newHOSPITALID)
+get(NAME)
+setNAME(newNAME)
+get(ADDRESS)
+setADDRESS(newADDRESS)
+get(UNDERINSURANCE)
+setUNDERINSURANCE(newUNDERINSURANCE)
*
CONDUCTS
health insurance
-VALIDITY:integer
-BENEFITS:string
-MONIEY INSURED:integer
-INSURANCE TYPE:string
-INSURANCE ID:integer(frozen)
-PREMIUM:integer
+get(VALIDITY)
+setVALIDITY(newVALIDITY)
+get(BENEFITS)
+setBENEFITS(newBENEFITS)
+get(MONEYUSED)
+setMONEYUSED(newMONEYUSED)
+get(INSURANCETYPE)
+setINSURANCETYPE(newINSURANCET
+get(PREMIUM)
+setPREMIUM(newPREMIUM)
+get(INSURANCEID)
+setINSURANCEID(newINSURANCEID)
Purchase
-START DATE:DateTimeDomain
-END DATE:DateTimeDomain
+get(STARTDATE)
+set(STARTDATE(newSTARTDATE)
+get(ENDDATE)
+setENDDATE(newENDDATE)
1..*
test
-CODE:integer(frozen)
-INFECTION CODE:integer
-PRICE:integer
+get(CODE)
+setCODE(newCODE)
+get(INFECTIONCODE)
+setINFECTIONCODE(newINFECTIONCODE)
+get(PRICE)
+setPRICE(newPRICE)
1..*
PROVIDES
1..*
GENERATES
0..1
1..*
claim
-CLAIM ID:integer(frozen)
-ISSUED AMT:integer
-ISSUED DATE:DateTimeDomain
-PERSON ID:integer
-STATUS:String
-INSURANCE ID:integer
+get(CLAIMID)
+setCLAIMID(newCLAIMID)
+get(ISSUEDAMT)
+setISSUEDAMT(newISSUEDAMT)
+get(PERSONID)
+setPERSONID(newPERSONID)
+get(STATUS)
+setSTATUS(newSTATUS)
+get(INSURANCE)
+setINSURANCE(newINSURANCE)
0..1
AVAILS
0..1
bill
-DISEASE CODE:integer
-BILL ID:integer(frozen)
-BILL AMT:integer
-DATE OF ADMISSION:DateTimeDomain
-DATE OF DISCHARGE:DateTimeDomain
+get(DISEASECODE)
+setDISEASECODE(newDISEASECODE)
+get(BILLID)
+setBILLID(newBILLID)
+get(BILLAMT)
+setBILLAMT(newBILLAMT)
+get(DATEOFADMISSION)
+setDATEOFADMISSION(newDATEOFADMISSION)
+get(DATEOFDISCHARGE)
+setDATEOFDISCHARGE(newDATEOFDISCHARGE)

Relational Mapping :

Primary key (underlined), foreign key (in italics)

- Patient(PersonID, *info_id*, medicalhistory, email, dateofbirth, status, Name, contactdetails)
    - Info_id

Here, Info_id is the foreign key which refers to Information and it is NULL NOT ALLOWED

- Purchases(*PersonID, InsuranceID*, startdate,enddate)
    - PersonID, InsuranceID

Here they are the foreign keys that refer to Patient and Health insurance respectively and it is NULL NOT ALLOWED

- Infected(*PersonID,Infectioncode*)
    - personID, Infectioncode

Here these are the foreign keys that refer to Patient and injury respectively and it is NULL NOT ALLOWED

- Information (zip, city, county)
- Health insurance (insuranceID, insurancetype,moneyinsured,benefits,validity,premium)
- Provides(*insuranceid* , *claimid*)
    - insuranceid, claimid

Here these are the foreign keys that refer to Health insurance and claim respectively and it is NULL NOT ALLOWED

- Claim(claimID, *billid* , issuedamt, issueddate, status, *personID, InsuranceID*)
    - personID, InsuranceID

Here they are the foreign keys that refer to Patient and Health insurance respectively and it is NULL NOT ALLOWED

- billid

this is the foreign key that refers to BILL and is NULL ALLOWED

- Bill(billid , dateofdischarge, dateofadmission, billamt, *injurycode*)
    - injurycode

Here it is the foreign key refering to injury and it is NULL NOT ALLOWED

- Injury(<u>infectioncode</u>, name)
- Treats(*infectioncode, hospitalid*)
    - infectioncode, hospitalid

Here it is the foreign key that refers to injury and hospital and it is NULL NOT ALLOWED

- Hospital(<u>hospitalid,</u> hname, haddress, isunderinsurance)
- Conducts(*hospitalid,testcode*)
    - hospitalid, testcode

Here it is the foreign key that refers to Hospital and Test and it is NULL NOT ALLOWED

- Test(<u>code</u>,*billid*, price)

    - Billid

Here it is the foreign key that refers to bill and it is NULL NOT ALLOWED

- Treatment(*infectioncode,hospitalid,code*)
    - hospitalid, testcode, code

Here it is a foreign key that refers to hospital, test and injury and it is NULL NOT ALLOWED.

Normalization :

- Patient(PersonID, *info_id*, medicalhistory, email, dateofbirth, status, Name, contactdetails)
    - Already in 1NF and 2NF
    - 3NF and BCNF: Patient(Name,contactdetails)

P_info(PersonID, *info_id*,medicalhistory,email,dateofbirth,status,Name)

- Purchases(*PersonID, InsuranceID*, startdate,enddate)
    - ALREADY IN 1NF and 2NF
    - 3NF and BCNF:

Purchases*(PersonID*, startdate, enddate) R(*PersonID, InsuranceID*)

- Infected(*PersonID*,*Infectioncode*)
    - Already normalized

- Information (zip, city, county)
    - Already normalized

- Health insurance (insuranceID, insurancetype,moneyinsured,benefits,validity,premium)
    - Already in 1NF and 2NF
    - 3NF and BCNF:

Healthinsurance (insurancetype,benefits,premium,moneyinsured,validity) Type (insuranceID,insurancetype)

- Provides(*insuranceid* , *claimid*)
    - Already normalized

- Claim(claimID, *billid* , issuedamt, issueddate, status, *personID, InsuranceID*)
    - Satisfies 1NF and 2NF
    - 3NF and BCNF: Insure(InsuranceID,claimID) Claim_details(claimID,status,billid) Bill_details(billid,issueddate,issuedamt) Person(*personID*,*InsuranceID*)

- Bill(billid , dateofdischarge, dateofadmission, billamt, *injurycode*)
    - ALREADY in 1NF, 2NF, 3NF, BCNF

    - Injury(infectioncode, name)
    - ALREADY in 1NF, 2NF, 3NF, BCNF

- Treats(*infectioncode, hospitalid*)

- ALREADY in 1NF, 2NF, 3NF, BCNF

- Hospital(<u>hospitalid,</u> hname, haddress, isunderinsurance)
  - ALREADY in 1NF, 2NF, 3NF, BCNF

- Conducts(*hospitalid*,*testcode*)
  - ALREADY in 1NF, 2NF, 3NF, BCNF

- Test(<u>code</u>,*billid*, price)
  - ALREADY in 1NF, 2NF, 3NF, BCNF

- Treatment(*infectioncode*,*hospitalid*,*code*)
  ALREADY in 1NF, 2NF, 3NF, BCNF

Analytical SQL Queries

Query1:

```
SELECT
    d.money_insured as MONEY_INSURED,
    d.insurance_type AS INSURANCE_TYPE,
    a.insurance_id AS INSURANCE_ID,
    b.status AS STATUS,
    f.P_Name as PatientName
FROM
    insure a,
    claim_details b,
    insurance_type c,
    health_insurance d,
    person e,
    patient_info f
WHERE
    a.claim_id = b.claim_id
    AND b.status = 'TRUE'
    AND a.insurance_id = c.insurance_id
    AND c.insurance_type = d.insurance_type
    and e.person_insurance_id=c.insurance_id
    and e.personID=f.Person_ID
LIMIT 10;
```

MONEY_INSURED	INSURANCE_TYPE	INSURANCE_ID	STATUS	PatientName
92254	PPO	6	TRUE	Erina Wye
92254	PPO	149	TRUE	Krystyna Quirk
74421	POS	141	TRUE	Barbara-anne Parbrook
74421	POS	195	TRUE	Chrisy Hessentaler
72931	EPO	131	TRUE	Sherman Joint
87407	HMO	184	TRUE	Tabb Borleace
74421	POS	167	TRUE	Nickey Pache
87407	HMO	41	TRUE	Noll Sayton
92254	PPO	108	TRUE	Layne Hagergham
92254	PPO	25	TRUE	Ferris Trustrie

This query will output the patient name whose insurance has been passed alongwith their insurance ID and which insurance type they have bought and the amount of money which is insured to them.

Query 2:

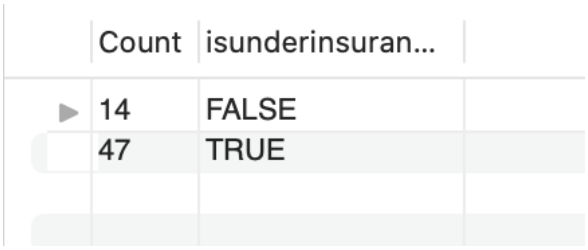
```
SELECT
    a.bill_amt AS Amount,
    b.injury_Name AS Injury,
    b.infection_code AS Infection_Code
FROM
    bill a,
    injury b
WHERE
    b.infection_code = a.injury_code
GROUP BY b.injury_Name
ORDER BY a.bill_amt DESC
LIMIT 10;
```

Amount	Injury	Infection_Code
99501	Poisoning by unsp topical agent, undetermined,…	20
99329	Fracture of mandible, unsp, subs for fx w routn…	150
97940	Displ commnt fx r patella, 7thR	21
97683	Cholera, unspecified	100
97157	Corrosion of second degree of left lower leg, init…	75
96573	Malignant neoplasm of appendix	187
96498	Nondisp commnt fx shaft of unsp femr, 7thE	86
96455	Allergy status to unsp drug/meds/biol subst status	184
96284	Non-pressure chronic ulcer of unspecified ankle	6

This query outputs the injuries and the infection code which has the highest billamount to treat that disease or injury.

Query 3:

```
SELECT
    COUNT(hospital_id) AS Count, isunderinsurance
FROM
    hospital
GROUP BY isunderinsurance
```



Count	isunderinsuran...
14	FALSE
47	TRUE

Query to retrieve the count of hospitals in Massachusetts that are under theinsurance, and which are not under the insurance.

Query 4:

```
SELECT
    COUNT(hospital_id) AS Count, isunderinsurance, h_address
FROM
    hospital
WHERE h_address = "Boston, MA"
GROUP BY isunderinsurance
```

	Count	isunderinsuran...	h_address
	7	TRUE	Boston, MA
	4	FALSE	Boston, MA

Query to fetch the count of hospitals that are under the insurance, and whichare not under insurance and are in Boston, Massachusetts.

Query 5:

```
SELECT
    status, COUNT(claim_id) AS COUNT, AVG(issued_amt)
FROM
    claim_details a,
    bill_details b
WHERE
    b.bill_id = a.claim_bill_id
        AND status = 'true'
```

	status	COUNT	AVG(issued_amt)
▶	TRUE	150	23367.046666666665

Query to return the average issued amount when the claim for the insurance isaccepted

Query 6:

```
SELECT
    DISTINCT(e.Person_ID),
    e.P_Name AS PatientName,
    b.treats_infection_code AS InfectionCode,
    c.Injury_Name AS Injury,
    a.hospital_name,
    a.h_address
FROM
    hospital a,
    treats b,
    injury c,
    infected d,
    patient_info e
WHERE
    a.hospital_id = b.treats_hospital_id
        AND b.treats_infection_code = c.infection_code
        AND c.infection_code = d.infection_cod
        AND d.person_id = e.Person_ID
        AND e.Status = 'true'
        AND a.h_address LIKE '%Boston%'
        AND a.isunderinsurance = 'TRUE'
LIMIT 10
```

Person_ID	PatientName	InfectionCode	Injury	hospital_name	h_address
67	Katina Pepall	2	Sltr-haris Type IV physl fx low end humer, l arm,...	Beth Israel Deaconess Medical Center	Boston, MA
192	Jacinda Shelliday	156	Benign neoplasm of connective and oth soft tiss...	Brigham and Women's Faulkner Hospital	Boston, MA
10	Cilka Frances	136	Scrotal pain	Brigham and Women's Faulkner Hospital	Boston, MA
22	Binky Marousek	136	Scrotal pain	Brigham and Women's Faulkner Hospital	Boston, MA
116	Bren Aizikovitch	30	Underdosing of unspecified systemic antibiotic, i...	Brigham and Women's Hospital	Boston, MA
73	Miles Dradey	24	Rheumatoid bursitis, left hand	Massachusetts Eye and Ear Infirmary	Boston, MA
15	Ruy Presland	167	Unsp fracture of navicular bone of right wrist, se...	Massachusetts General Hospital	Boston, MA
64	Jessie Whiteman	101	Toxic effect of petroleum products, self-harm, init	Massachusetts General Hospital	Boston, MA
159	Bobina Culcheth	105	Military operations involving other explosions an...	Tufts Medical Center	Boston, MA
106	Krystyna Quirk	62	Contusion of unspecified forearm	Tufts Medical Center	Boston, MA

Query to fetch Patient information who were treated in Hospitals in Boston andhad their insurance covered along with the injury and infection code that were treated in those hospitals.

Query 7:

```
SELECT
    COUNT(Person_ID) AS count, status,
FROM
    patient_info
GROUP BY status
```

count	status
90	true
110	false

Query to output the count of person who has active health insurance.

Query 8:

```
SELECT
    bill_id as Bill_ID,
    f.date_of_discharge,
    f.date_of_admission,
    a.infection_code as Infection_Code,
    a.Injury_name as Injury_Name,
    e.Price as TestPrice,
    f.bill_amt as Treatment_amount,
    (e.Price + f.bill_amt) AS Total_Bill
FROM
    injury a,
    Treats b,
    hospital c,
    conducts d,
    test e,
    bill f
WHERE
    a.infection_code = b.treats_infection_code
        AND b.treats_hospital_id = c.hospital_id
        AND c.hospital_id = d.hospital_id
        AND d.test_code = e.Test_code
        AND e.billid = f.bill_id
ORDER BY (e.Price + f.bill_amt) desc
LIMIT 10;
```

Bill_ID	date_of_dischar...	date_of_admissi...	Infection_Code	Injury_Name	TestPrice	Treatment_amount	Total_Bill
161	10/4/2022	9/24/2022	187	Malignant neoplasm of appendix	9685	99501	109186
161	10/4/2022	9/24/2022	142	Disp fx of 4th metatarsal bone, unsp ft, 7thG	9685	99501	109186
161	10/4/2022	9/24/2022	149	Corrosion of third degree of neck, subsequent encounter	9685	99501	109186
33	9/13/2022	9/4/2022	158	Mtrcy driver injured in collision w unsp mv in traf, subs	9215	96498	105713
33	9/13/2022	9/4/2022	118	Poisoning by appetite depressants, self-harm, sequela	9215	96498	105713
33	9/13/2022	9/4/2022	130	Subluxation and dislocation of T8/T9-T9/T10 thor vertebra	9215	96498	105713
16	5/24/2023	5/5/2023	168	Sprain of tibiofibular ligament of unsp ankle, sequela	7742	97683	105425
16	5/24/2023	5/5/2023	75	Corrosion of second degree of left lower leg, init encntr	7742	97683	105425
16	5/24/2023	5/5/2023	148	Contusion of right forearm, initial encounter	7742	97683	105425
91	6/23/2023	6/5/2023	124	Other effects of lightning, sequela	6693	97940	104633

Query to fetch the top 10 total Bill amount for the treatment of various injuries

Analytical NoSQL Queries:

Query to find the average of the entire bill amount of treating various injuries

```
db.getCollection("bill").find({})

db.bill.aggregate([
    {
        $group : {
            _id: "aggregates",
            average_insured: {
                $avg: "$bill_amt"
            },
        }
    }
])
```

```
{
    "_id" : "aggregates",
    "average_insured" : 54632.075
}
```

Query to find top 10 costliest test

```
db.getCollection("test").find({})

db.test.aggregate([
{$group:{_id:"$Test_code", Price:{$max:"$Price"}}},
{$project:{_id:0, "Test_code":"$_id", Price:1}},
{$sort:{Price:-1}},
{$limit: 10}
])
```

```
{
    "Price" : NumberInt(9962),
    "Test_code" : NumberInt(64)
}
{
    "Price" : NumberInt(9944),
    "Test_code" : NumberInt(193)
}
{
    "Price" : NumberInt(9938),
    "Test_code" : NumberInt(194)
}
{
    "Price" : NumberInt(9930),
    "Test_code" : NumberInt(129)
}
{
    "Price" : NumberInt(9909),
    "Test_code" : NumberInt(59)
}
{
    "Price" : NumberInt(9857),
    "Test_code" : NumberInt(36)
}
{
    "Price" : NumberInt(9851),
    "Test_code" : NumberInt(13)
}
{
    "Price" : NumberInt(9830),
```

Count of patients who were insured.

```
db.getCollection("patient_info").find({})

db.patient_info.aggregate(
  [
    {
      $match: {
        Status: {
          $eq: "true"
        }
      }
    },
    {
      $count: "Patients who had Insurance"
    }
  ]
)
```

```
{
    "Active_Patient_Count" : NumberInt(90)
}
```

Count of hospitals in Massachusetts that are under insurance cover

```
db.hospital.aggregate(
[
    {
    $match: {
    isunderinsurance: {
        $eq :true
    }
    }
    },
    {
    $count: "Hospital under Insurance"
    }
     ]
)
```

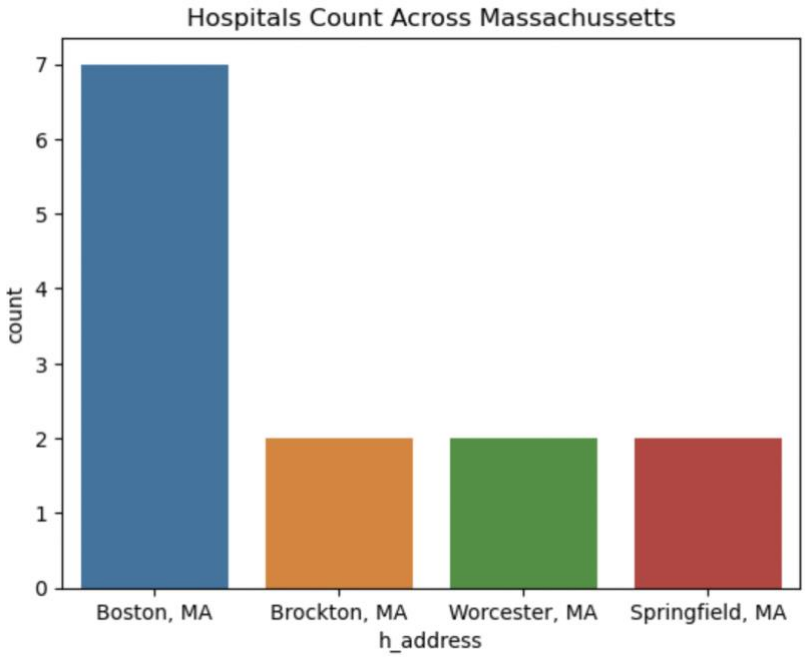
```
{
    "Hospital under Insurance" : NumberInt(47)
}
```

Database Access Python

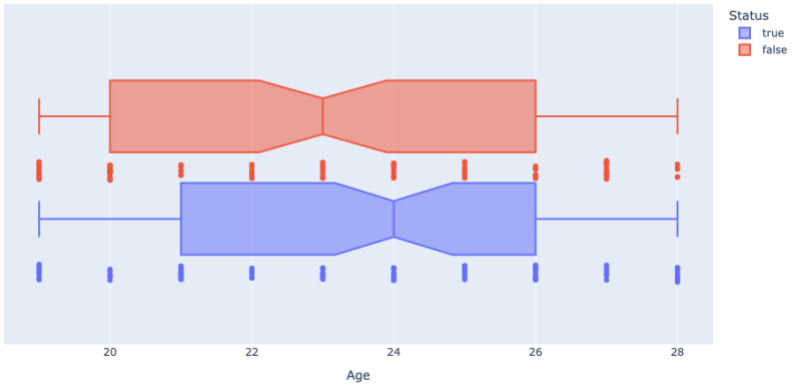
The database is accessed using Python's library such as Pandas library and using Matplotlib to plot the graphs for the analytics.

Analytics Using python:

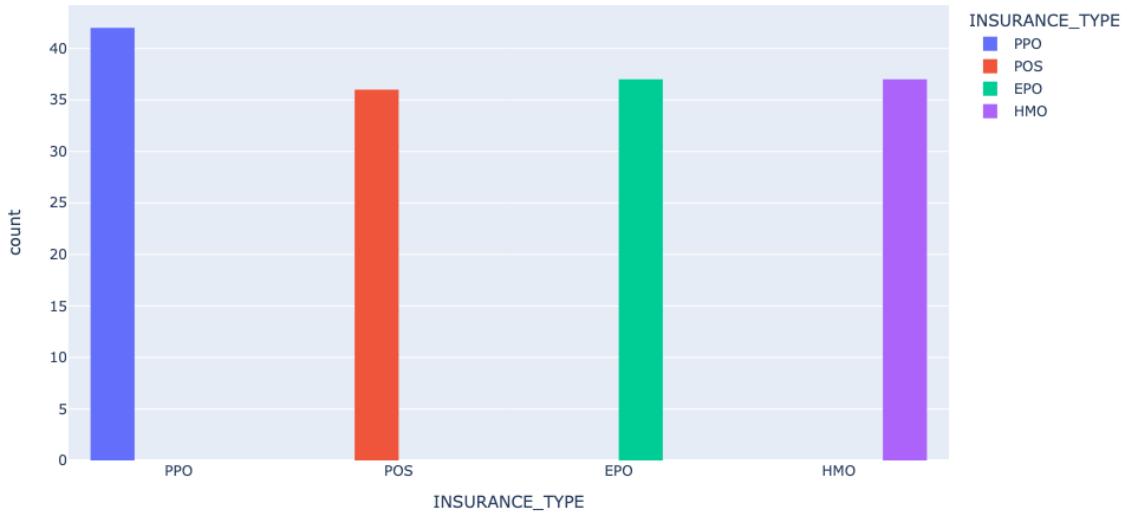
1. The number of hospitals that are under insurance of PPO type which has the most number of tests being performed at the hospital.



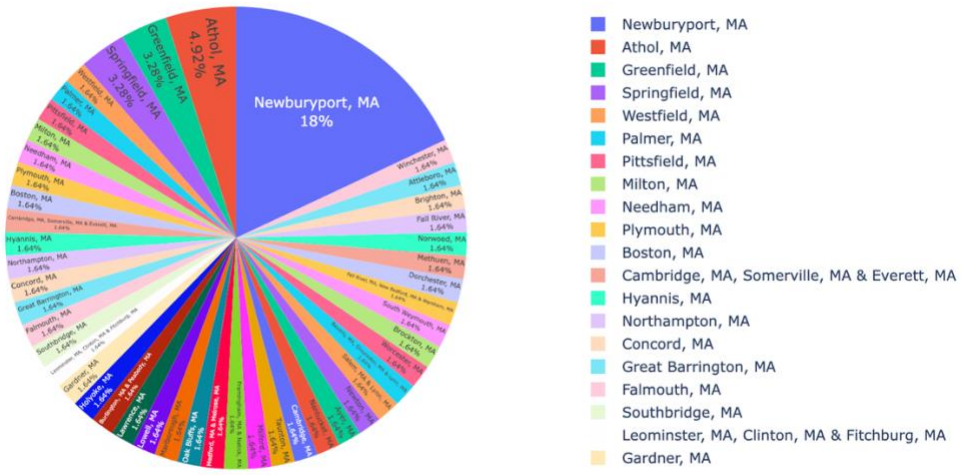
2. Box plot distribution of age of patients vs status of whether they have active insurance plans



3. Count of Insurance Plans that have been successfully claimed at least one time



4. Pie chart of no of hospitals in Massachusetts that are under insurance depending on the percentage



Project Conclusion:

-We have conveniently administered the problem statement, conceptual Modeling using EER & UML Diagram in SQL database as well as NoSQL database.

-Also, we have successfully implemented the database access via python & demonstrated practical visualization analytics.