IE6700 Data Management for Analytics

Use Case Study Report

Group No 19

Group Members-: Lei Duan
Saurav Rohidas Palekar

Topic-: Blood Donation Database System

Executive Summary-:

The goal of our project was to create a Blood donation database system for a blood donation organization. The relation database for the project is created on MySQL, with the data requirements such as, persons, patient, donor, doctor, donation records, transfusion records, pre-examination details etc. For the NoSQL implementation we have used MongoDB Compass and for analytical part we have connected the MySQL to Tableau, to get various data visualization insights. Apart from this we also implemented the possible user interface for the website or an app for the organization. More study should be done on how a unique relational database like this can be implemented in a NoSQL environment. Having implemented a few tables on MongoDB database and performing queries on it, this can certainly be used to build the database but the whole data input process benefits of using a relational database.

I. Introduction

Background and Introduction:-

Pulselife is a leading Blood donation organization in the state of Massachusetts, hosting blood donation drives throughout the year, distributing the blood to hospitals and health centers in the state of Massachusetts. The Pulselife has created an app and has a website for the blood donation and transfusion. Pulselife also has doctors which does the pre-examination of the patient and the donor before the donation and transfusion process. The organization then stores the transfusion and donation data. The Pulselife has a bond with all Blood banks, health centers and hospitals in state of Massachusetts and can get the Blood stock from them. The organization has various Blood Inventories in the State where the bloodstocks are stored. The Pulselife wants to create a database for the blood donation management system and has certain requirements…

1. All the person and the details such as people id, age, name, etc., must be in 1 table and people consists of Patient, Doctor and Donor.

2. All the blood donation and transfusion requests need to be sent to PulseLife which manages the requests depending upon the availability and then the request is then processed.

3. Before the transfusion and donation, the patient and donor needs to undergo pre-examination in which various traits of body such as temperature, blood pressure, pulse rate, hemoglobin is monitored by the doctor.

4. A person cannot be patient and doctor, patient and donor or doctor and donor at same time.

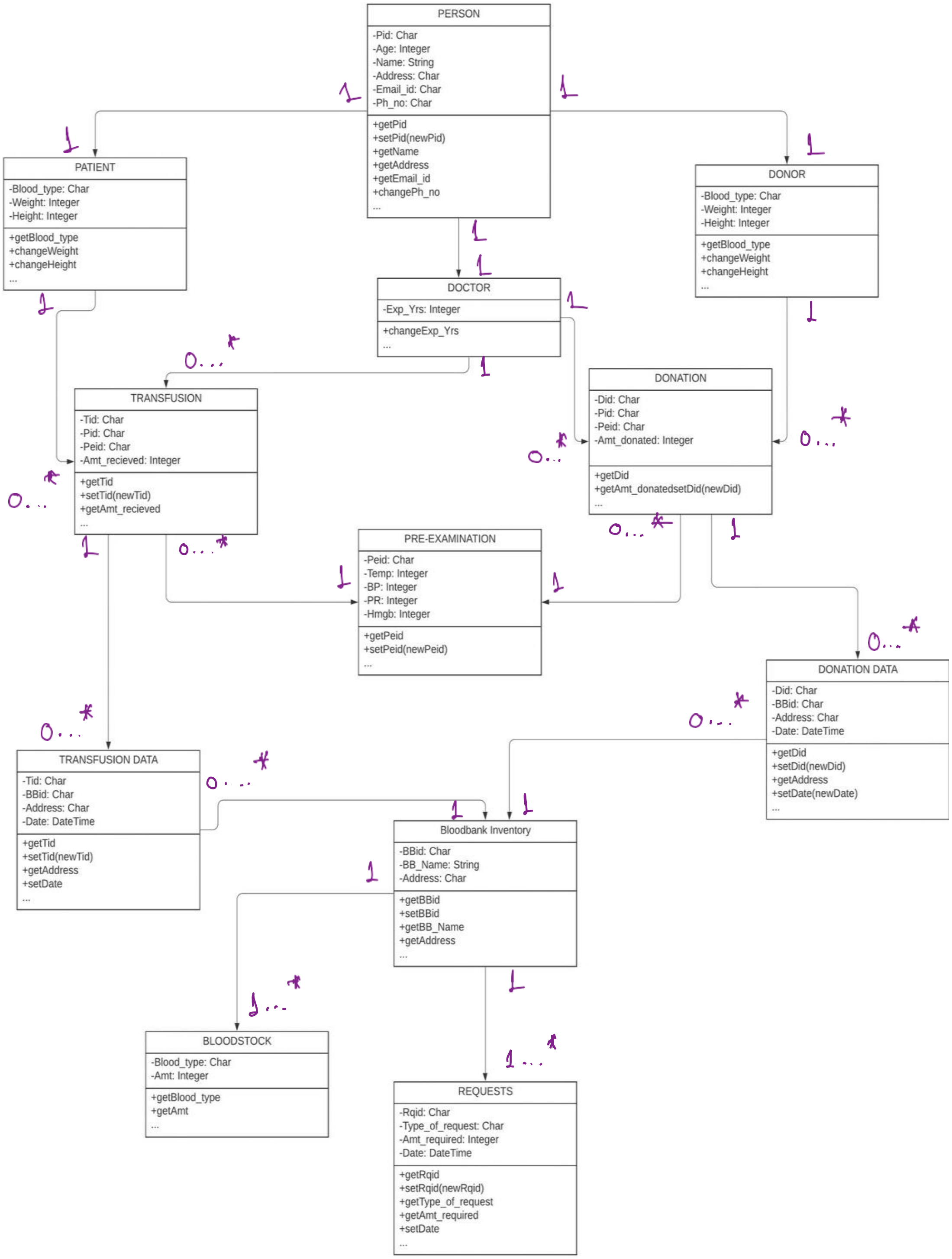
5. For patient and donor, height and blood with various body traits are very important for the donation and transfusion process.

Design:-

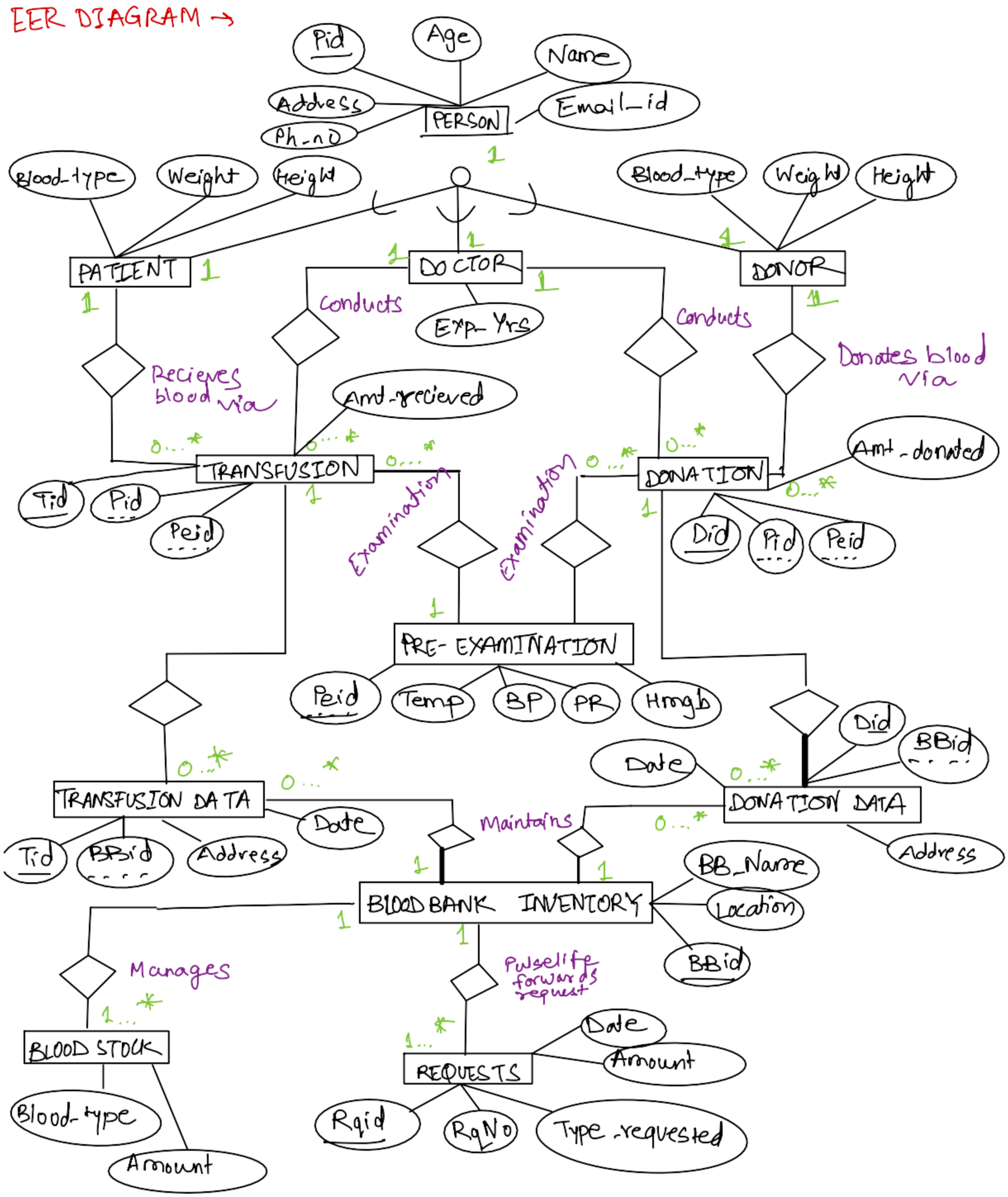
The design of the database requires various tables: People, patient, doctor, donor, transfusion data, donation data, requests, blood bank inventory, pre-examination etc. We also must establish various relationships between the entities. A person can be a doctor, patient, or donor, and not both or all at same time. A patient can receive blood many times he wants, and donor can donate blood as many times he wants but under the specified days limit after previous donation and same goes with patient. Similarly, a doctor can do the process of transfusion, donation, and pre-examination as many times he wants. The PulseLife manages all the data and process all the blood donation related requirements as per the need.

II. Conceptual Data Modeling

The below is the advanced UML diagram for our database with more information.



The below is the advanced proposed EER diagram with more information. This diagram is more detail than the database we used.



III. Mapping Conceptual Model to Relational Model

Relational Model: Primary keys are underlined; Foreign keys are in italics.

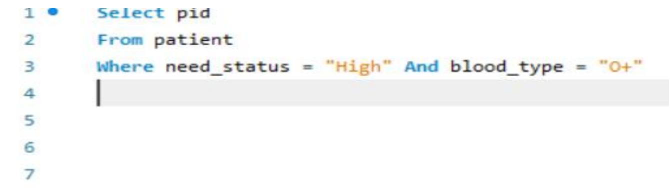
- PEOPLE (*Pid*, Age, Name, Address, Ph_no)
- PATIENT (*Pid*, Blood_type, Weight, Height)
- DONOR (*Pid*, Blood_type, Weight, Height)
- DOCTOR (*Pid*, Exp_Yrs)
- TRANSFUSION (Tid, *Pid*, *Peid*, Amt_recieved)
- DONATION (Did, *Pid*, *Peid*, Amt_donated)
- PRE-EXAMINATION (Peid, Temp, BP, PR, Hmgb)
- TRANSFUSION_RECORDS (Tid, *BBid*, Date)
- DONATION_RECORDS (Did, *BBid*, Date)
- BLOODBANK INVENTORY (BBid, donated_type, Amount_donated, blood_type)

IV. Implementation of Relation Model via MySQL and NoSQL

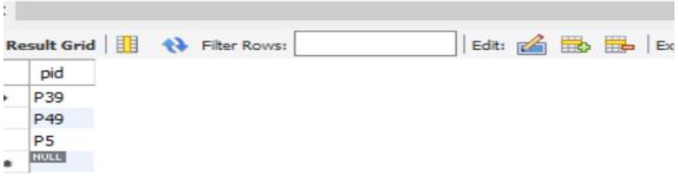
MySQL Implementation of the database:-

Query 1: Find the patients of all blood types with high need status.

(This could help the hospital to identify which patient of a certain blood type is in the high demand of blood when certain type of blood is supplied.)



```
Select pid
From patient
Where need_status = "High" And blood_type = "O+"
```



```
pid
P39
P49
P5
NULL
```

Select pid

From patient

Where need_status = “High”

Group by blood_type

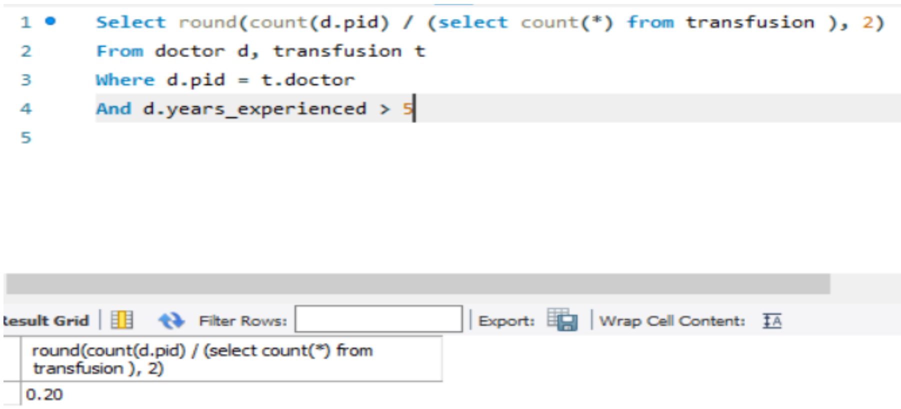
Query 2: Find out how the proportion that transfusions are conducted by the doctors with 5 years-experience or more in all transfusions.

(This could really help us out when we need to identify the exact proportion of experienced doctors conducting the transfusion and we could try to change that rate with more reasonable arrangement)

Select round(count(d.pid) / (select count(*) from transfusion), 2)

From doctor d, transfusion t

Where d.pid = t.doctorAnd d.years_experienced > 5



```
Select round(count(d.pid) / (select count(*) from transfusion ), 2)
From doctor d, transfusion t
Where d.pid = t.doctor
And d.years_experienced > 5
```

```
round(count(d.pid) / (select count(*) from
transfusion ), 2)
0.20
```

Query 3: Find out the donor donates blood within a month and if there is no record satisfied the requirement, return Null.

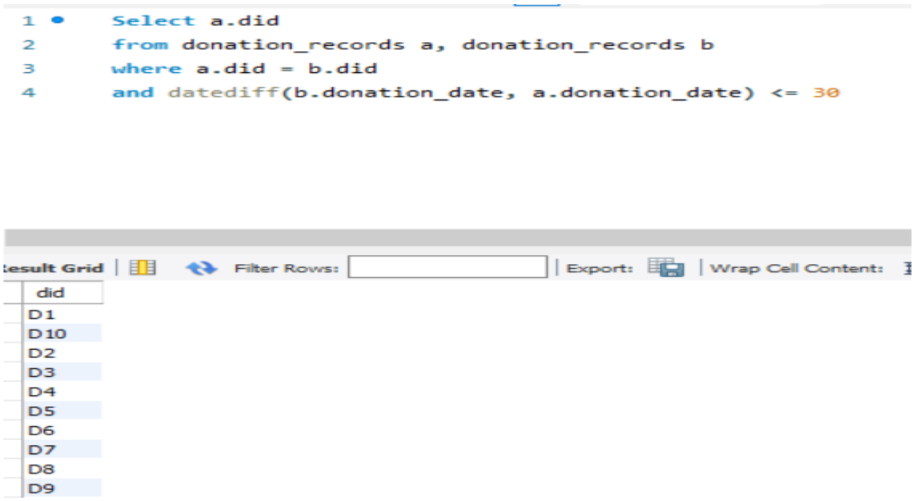
(This query is aimed to help us to find out which donor donates blood too frequently and might have a health issue if he or she keeps doing this).

Select a.did

from donation_records a, donation_records b

where a.did = b.did

and datediff(b.donation_date, a.donation_date) <= 30



```
Select a.did
from donation_records a, donation_records b
where a.did = b.did
and datediff(b.donation_date, a.donation_date) <= 30
```

```
did
D1
D10
D2
D3
D4
D5
D6
D7
D8
D9
```

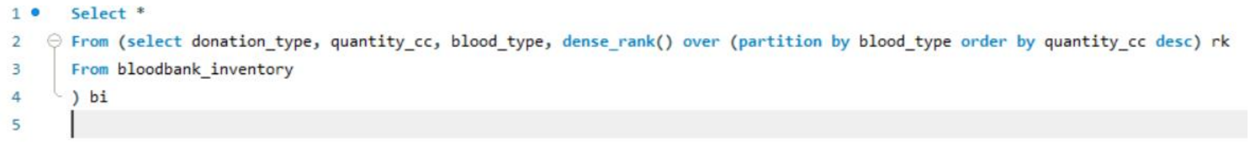
Query 4: Find out the rank of different blood types in the blood bank inventory according to their quantity_CC.

(This query could help us have a clear view about our blood inventory – which blood type we have the most inventory and which blood type is in alert)

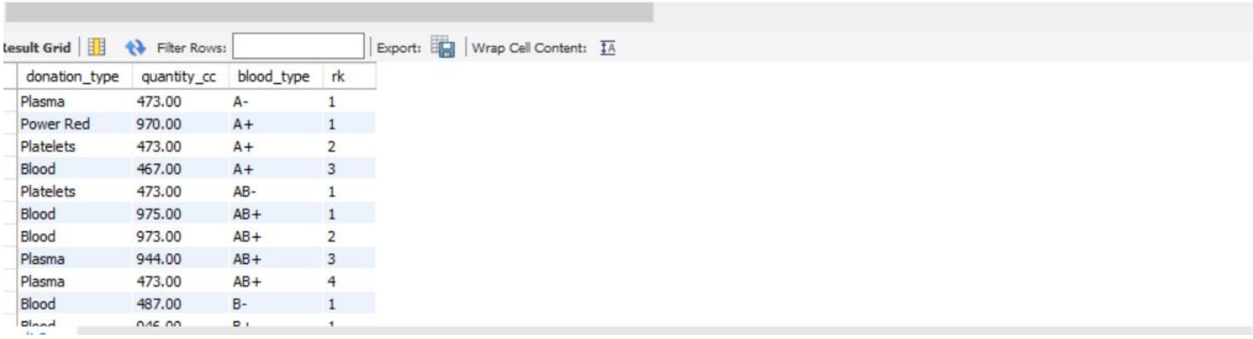
Select * From (select donation_type, quantity_cc, blood_type, dense_rank() over (partition by blood_type order by quantity_cc desc) rk

From bloodbank_inventory

) bi



```
Select *
From (select donation_type, quantity_cc, blood_type, dense_rank() over (partition by blood_type order by quantity_cc desc) rk
From bloodbank_inventory
) bi
```

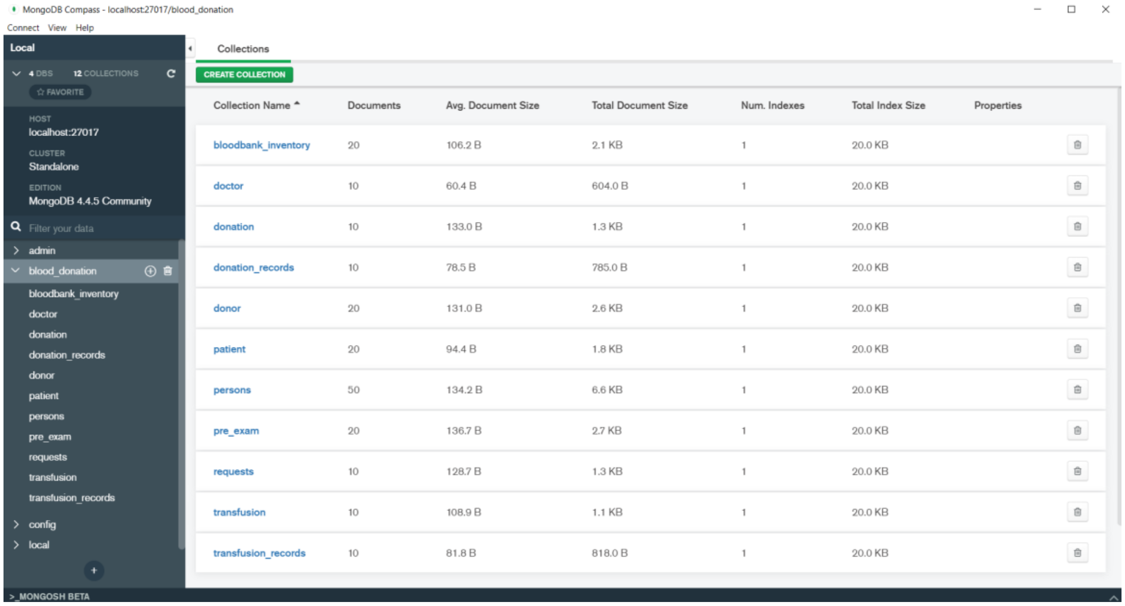


donation_type	quantity_cc	blood_type	rk
Plasma	473.00	A-	1
Power Red	970.00	A+	1
Platelets	473.00	A+	2
Blood	467.00	A+	3
Platelets	473.00	AB-	1
Blood	975.00	AB+	1
Blood	973.00	AB+	2
Plasma	944.00	AB+	3
Plasma	473.00	AB+	4
Blood	487.00	B-	1
Blood	946.00	B+	1

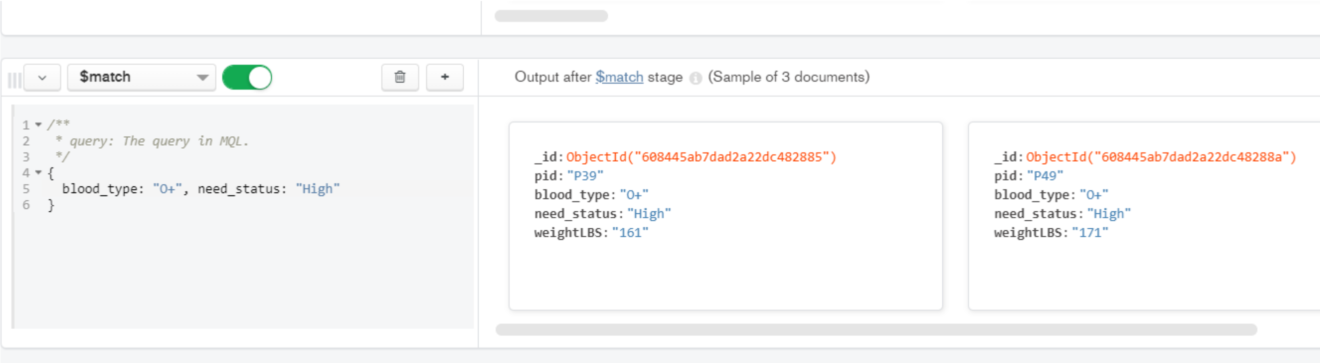
NoSQL Implementation:-

MongoDB

In the image below, we have created a database in the NoSQL database, here we have used MongoDB.



Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
bloodbank_inventory	20	106.2 B	2.1 KB	1	20.0 KB	
doctor	10	60.4 B	604.0 B	1	20.0 KB	
donation	10	133.0 B	1.3 KB	1	20.0 KB	
donation_records	10	78.5 B	785.0 B	1	20.0 KB	
donor	20	131.0 B	2.6 KB	1	20.0 KB	
patient	20	94.4 B	1.8 KB	1	20.0 KB	
persons	50	134.2 B	6.6 KB	1	20.0 KB	
pre_exam	20	136.7 B	2.7 KB	1	20.0 KB	
requests	10	128.7 B	1.3 KB	1	20.0 KB	
transfusion	10	108.9 B	1.1 KB	1	20.0 KB	
transfusion_records	10	81.8 B	818.0 B	1	20.0 KB	



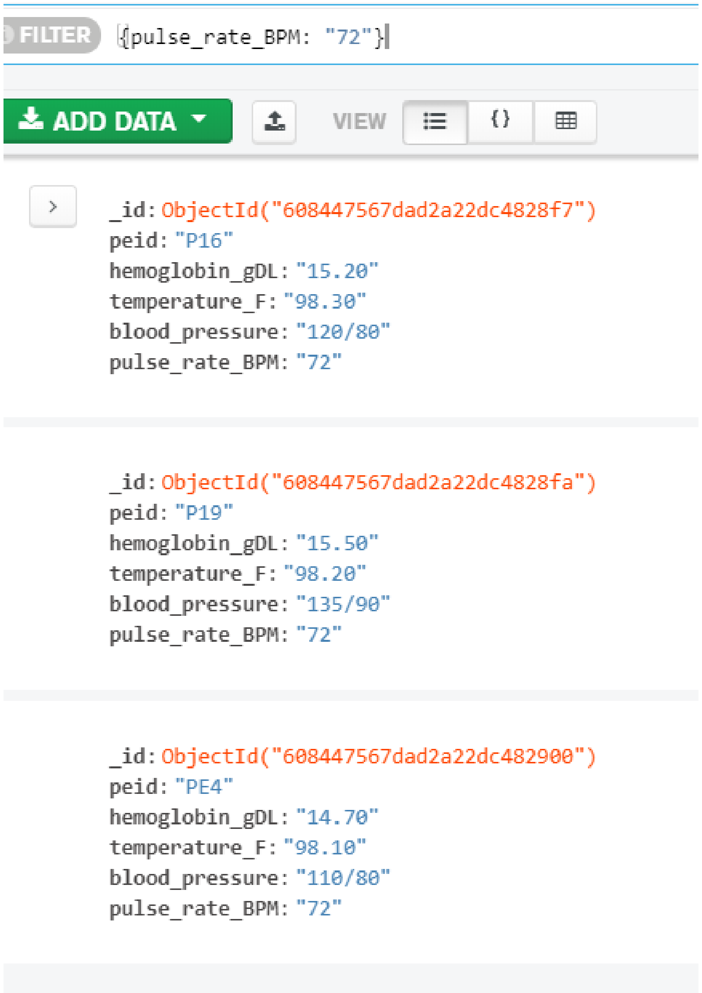
Output after $match stage (Sample of 3 documents)

```
/**
 * query: The query in MQL.
 */
{
  blood_type: "O+", need_status: "High"
}
```

```
_id: ObjectId("608445ab7dad2a22dc482885")
pid: "P39"
blood_type: "O+"
need_status: "High"
weightLBS: "161"
```

```
_id: ObjectId("608445ab7dad2a22dc48288a")
pid: "P49"
blood_type: "O+"
need_status: "High"
weightLBS: "171"
```

The above is the simple query performed in the MongoDB, where we used aggregate tab to find a patient who has blood group O+ and his need status is high.



```
{pulse_rate_BPM: "72"}
```

```
_id: ObjectId("608447567dad2a22dc4828f7")
peid: "P16"
hemoglobin_gDL: "15.20"
temperature_F: "98.30"
blood_pressure: "120/80"
pulse_rate_BPM: "72"
```

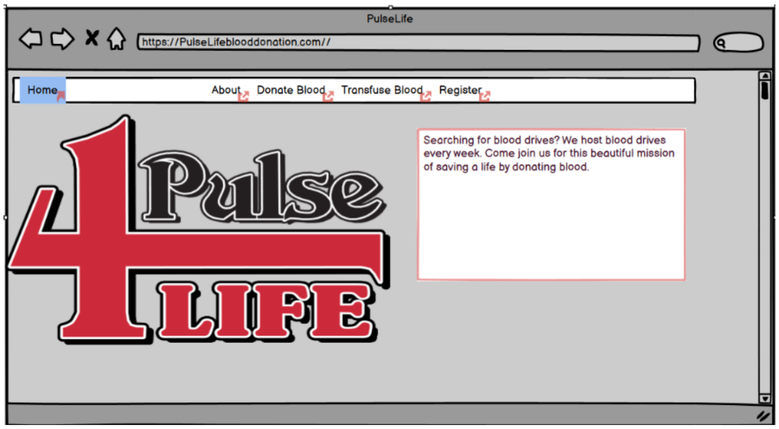
```
_id: ObjectId("608447567dad2a22dc4828fa")
peid: "P19"
hemoglobin_gDL: "15.50"
temperature_F: "98.20"
blood_pressure: "135/90"
pulse_rate_BPM: "72"
```

```
_id: ObjectId("608447567dad2a22dc482900")
peid: "PE4"
hemoglobin_gDL: "14.70"
temperature_F: "98.10"
blood_pressure: "110/80"
pulse_rate_BPM: "72"
```

The above is the simple filter query where we have find that how many pre-examination tests has pulse rate 72, which is an idea pulse rate for a human.

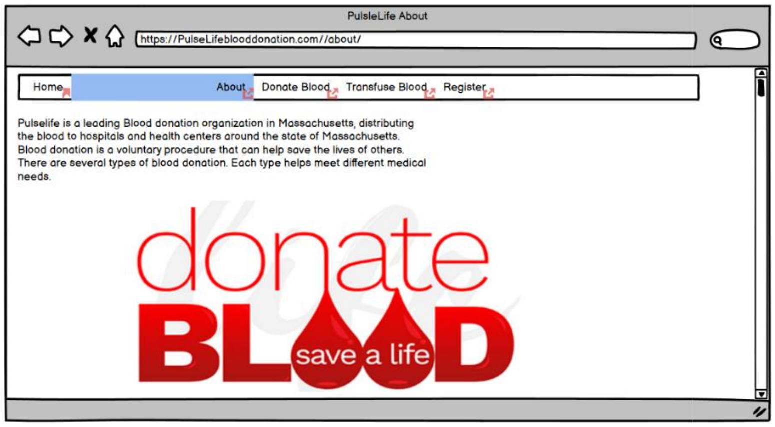
GUI Interface using Balsamiq Wireframes:-

We have used Balsamiq wireframes to show the GUI Interface, with home, about, register and donate blood pages.



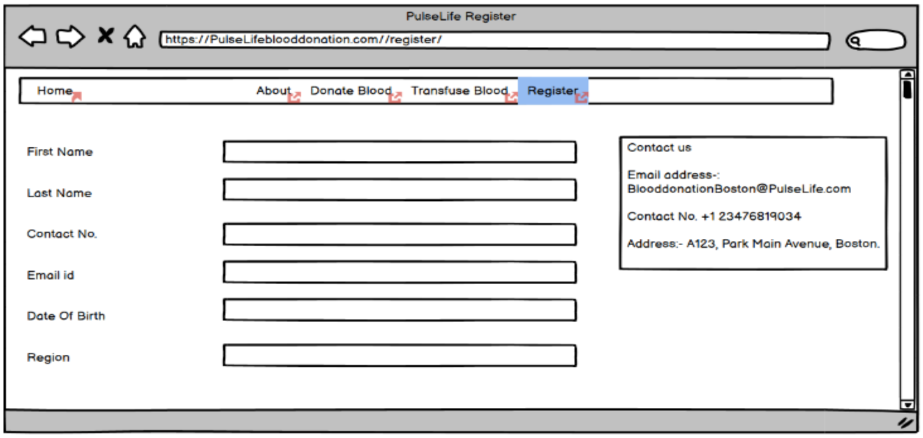
Searching for blood drives? We host blood drives every week. Come join us for this beautiful mission of saving a life by donating blood.

The above is the home page, which contains the motto and aim of the organization.



Pulselife is a leading Blood donation organization in Massachusetts, distributing the blood to hospitals and health centers around the state of Massachusetts. Blood donation is a voluntary procedure that can help save the lives of others. There are several types of blood donation. Each type helps meet different medical needs.

The above is the about page, which give the information about the organization.



First Name

Last Name

Contact No.

Email id

Date Of Birth

Region

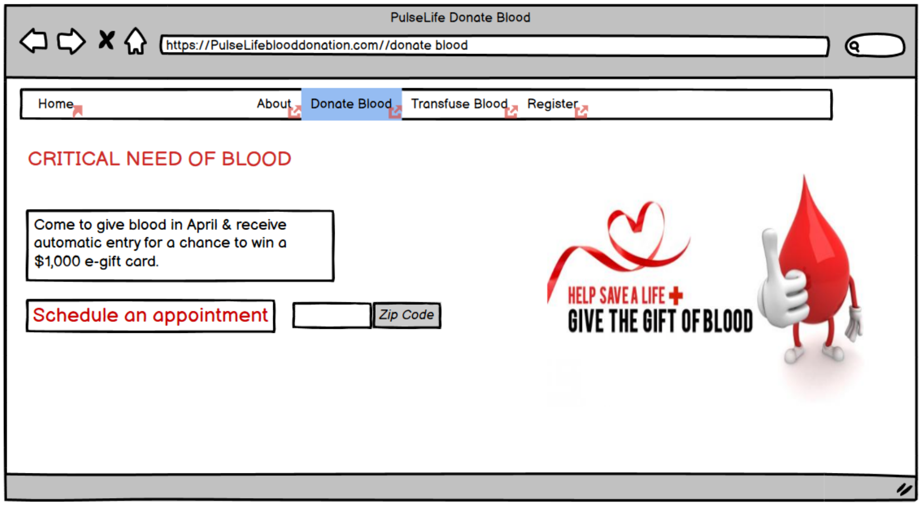
Contact us

Email address-:
BlooddonationBoston@PulseLife.com

Contact No. +1 23476819034

Address:- A123, Park Main Avenue, Boston.

The above is the registration page.

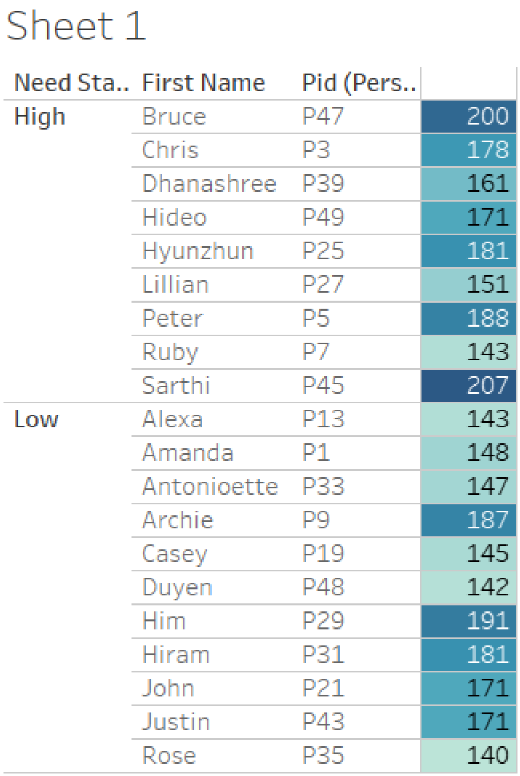


CRITICAL NEED OF BLOOD

Come to give blood in April & receive automatic entry for a chance to win a $1,000 e-gift card.

The above is the blood donation page, where a person can locate active blood donation drives around his area by inserting zipcode.

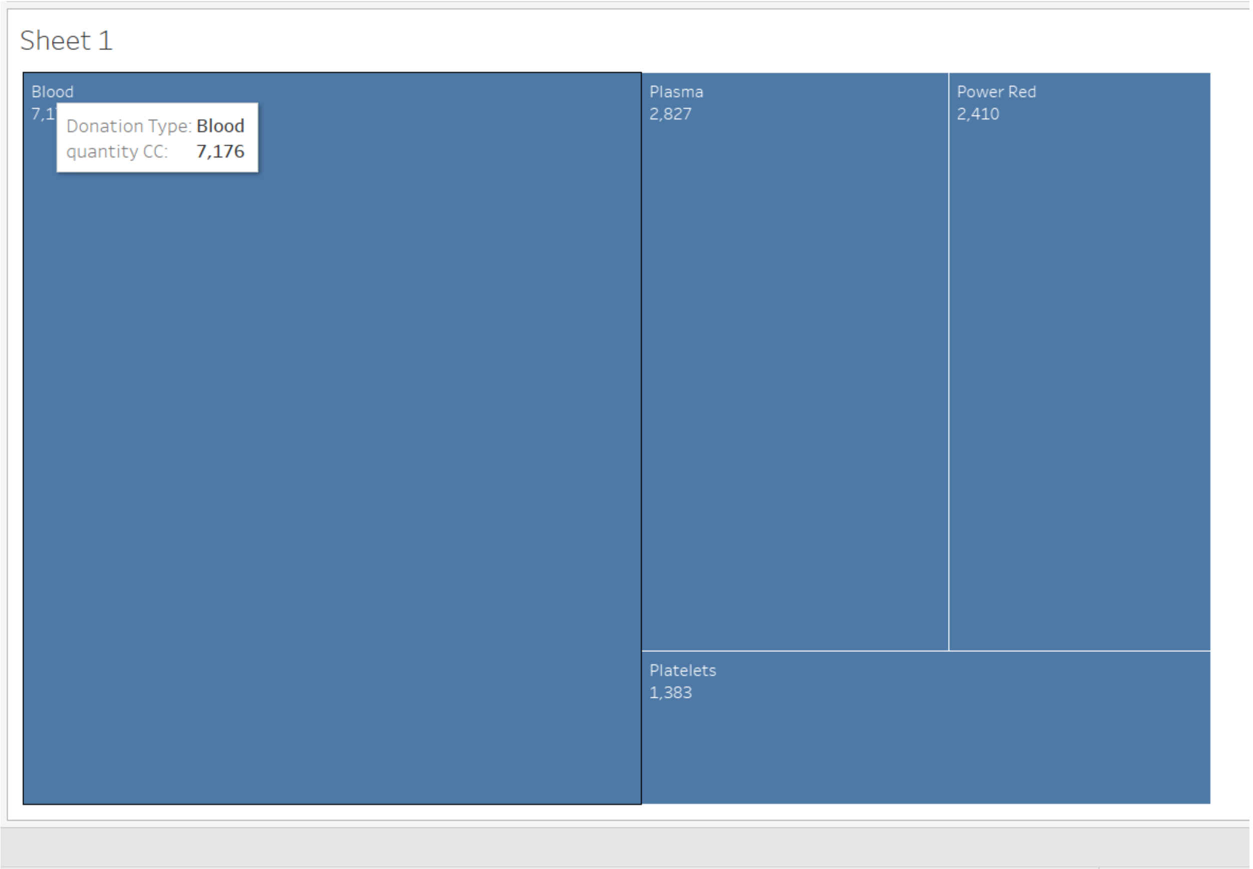
V. Database Access via Tableau



Sheet 1

Need Sta..	First Name	Pid (Pers..	
High	Bruce	P47	200
	Chris	P3	178
	Dhanashree	P39	161
	Hideo	P49	171
	Hyunzhun	P25	181
	Lillian	P27	151
	Peter	P5	188
	Ruby	P7	143
	Sarthi	P45	207
Low	Alexa	P13	143
	Amanda	P1	148
	Antonioette	P33	147
	Archie	P9	187
	Casey	P19	145
	Duyen	P48	142
	Him	P29	191
	Hiram	P31	181
	John	P21	171
	Justin	P43	171
	Rose	P35	140

In the above analytical table above, we have the blood need status of patient which is high or low, the name and the weight which is the one of the most essential body trait to perform blood transfusion. Here we have integrated MySQL with Tableau and performed visualizations, which gives helpful insights to us.



In the above two diagrams, we have connected Tableau with MySQL. The first diagram tells us the patients with blood status high and low with their weight in lbs.

VII. Summary and recommendation

The Blood Donation Database designed on MySQL is an industry ready relational database that can be implemented in for the Blood donation organization. It will help all the patients who are in need of Blood, and also donors who wants to donate the blood and provides great analytics capabilities, a small part of which is shown in this report utilizing Tableau. We went an extra step to design the front end UX wireframes for the database as well using Balsamiq Wireframe software, it showed the possible implementation of the website or an app for the organization. Improvement on the database would be the implementation of data governance measures on the database to ensure data quality as the entered donor or patients data can be reused and searched for various requirements. The shortcoming would in the NoSQL implementation of this Database on MongoDB. More study should be done on how a unique relational database like this can be implemented in a NoSQL environment. Having implemented a few tables on MongoDB database and performing queries on it, this can certainly be used to build the database but the whole data input process benefits of using a relational database and (n-n) relationships that are constantly updated in the stay data table has greater advantages in the real-world application, and this outweighs the benefits from a NoSQL database for the same.