

Database Management and Database Analytics

Practicum – 3 Non-Relational Database

Instructor: Dr. Alan Brandyberry

Student: Nikhil Kumar Sampath (811222899)

Save Lives – Donate Blood

Idea Generation:

Save Lives is an NPO which creates awareness among individuals by letting them know what's the importance of donating blood and how good it makes an individual feel by donating blood to a person in need. It also plays an important role in driving the blood donors on their list to the right person in need at the right time, so that they can **“Save Lives”**.

The founder of Save Lives got to realize that on average in the United States only 3% of the entire population comes forward to donate blood. Most of the time the supply can't always meet the demand, moreover the global pandemic i.e., coronavirus has taken a greater hit on the fear of the people and the numbers started falling as they were fewer donors visiting blood collection centers.

NPO Requirement:

Save Lives wanted to create two different databases, the first one would be for storing all the blood donors listed who can donate blood based on their blood group, this would ideally be done with the help of an application which would be launched by Save Lives, here the people who are interested to donate blood can register themselves as a blood donor. The second database which it wanted to build upon was a structured database which has a live server connected from various hospitals in the United States where the hospital staff can put a request for the blood needed along with all the details such as Blood Group, Patient Details, Point of Contact etc....later a dedicated team working at Save Lives will use the first database to contact the right donor based upon his blood group and location.

Why NoSQL, why not SQL? Also, if NoSQL is best which application is useful?

In SQL applications there would be a structured format involved with tuples and attributes thereby limiting the flexibility of the users in understanding the data, if we want to query the desired output mostly we would have to join the information from two tables for which we

need to understand the relationships which have been built using the primary key and foreign key constraints not only this each attribute has its data type which would thereby lead to computational intensity in querying the results. Whereas, when it comes to a NoSQL database especially referring to the Node4j you have a lot of flexibility in visualizing and understanding the data. Graph databases like Node4j are special purposed databases which primarily focus on problems where the relationship between objects is considered more important rather than the characteristic of the objects. Also, the use of cypher language helps in building the relationships effectively thereby mitigating the complexities which were generated in SQL. Hence, at Save Lives we are more considerate towards the relationships rather than just the data being stored, so I believe it is going to be beneficial for Save Lives to have NoSQL as their database.

Why should graph databases be beneficial to use for Save Lives?

Graph databases are visually more informative the properties of the database can be illustrated both via edges and nodes, the objects on which we are storing some data are called nodes and the relationship between these nodes is called edges. In relational databases, the data about a relationship is stored by redesigning it as an associative entity. But graph databases are structured to store data about the relationships themselves. Moreover, Node4j is based on the property graph model where they store data associated with both nodes and edges. At Save Lives we need to map and understand which donors belong to which blood group and where do they reside. This can help the NGO contact the right donor at the right time and provide them with the details of the requirement. Also, the secondary motive of using these graphs is to create awareness campaigns among individuals by showing them the sleek graph that is obtained based on a donor's journey so far in saving lives.

Illustration:

The hospitals, health centres and pharma centres post their requests through a form for blood/platelets which will be thereby received to Save Lives in the form of a structured format since the database pools this information in a set of attributes as listed below:

Hospital Name	Location	Patient Name	Age	Health Issue	Blood - RBC	Blood - Platelets	Blood Group
University Edge Medical Centre	Raveena, Kent	Chris Mclair	39	Heart Surgery	5 Units	NA	O+ve
Ohio State Hospital	Ohio	Michael Blair	67	Brain Haemorage	3 Units	NA	O+ve
CLS Pharma	Streetsbro, Kent	Henry	56	Breakbone Fever	1 Unit	1 Unit	O-ve
DeWeese Health Centre	KSU, Kent	Sophia	45	Malarial Virus	1 Unit	2 Units	AB+ve
DeWeese Health Centre	KSU, Kent	Imhoff Scot	54	Kidney Transplant	2 Units	NA	AB-Ve
CLS Pharma	Cleveland, Ohio	Kattie Marshall	33	Breast Cancer	1 Unit	NA	B-Ve
Michigan Pharma Centre	Central Michigan, Michigan	Joy Mary	50	Muscle Repair	2 Units	NA	A+ve

Then later the main process begins i.e., Save Lives now will start looking for blood donors using the neo4j database based on a few criteria i.e.,

1. Is the donor from the same blood group?
2. Is the donor eligible to donate blood or platelets?
3. When did the donor has done his last blood work?
4. What was the latest haemoglobin level of the donor upon his latest medical check?
5. Will a single donor be able to donate the required amount of blood i.e., units?

(Note: Currently we are limiting identifying the donors to just 2 criteria i.e., the blood group of the donor and the location of the donor)

Now, since Save Lives has a list of criteria to look for in the blood donors, now we can start to look at the updated Neo4j database for eligible donors,

Before that let's look at how the nodes and edges are created using *cypher language*;

Here we are creating a node called **"Group"** which refers to the blood groups, which is then named concerning all the blood groups using the given query, Similarly, we create another node which refers to the **"Persons"**.

```
savelives$ CREATE(n:Group{name:'O+ve'})
```



Table

Added 1 label, created 1 node, set 1 property, completed after 11 ms.

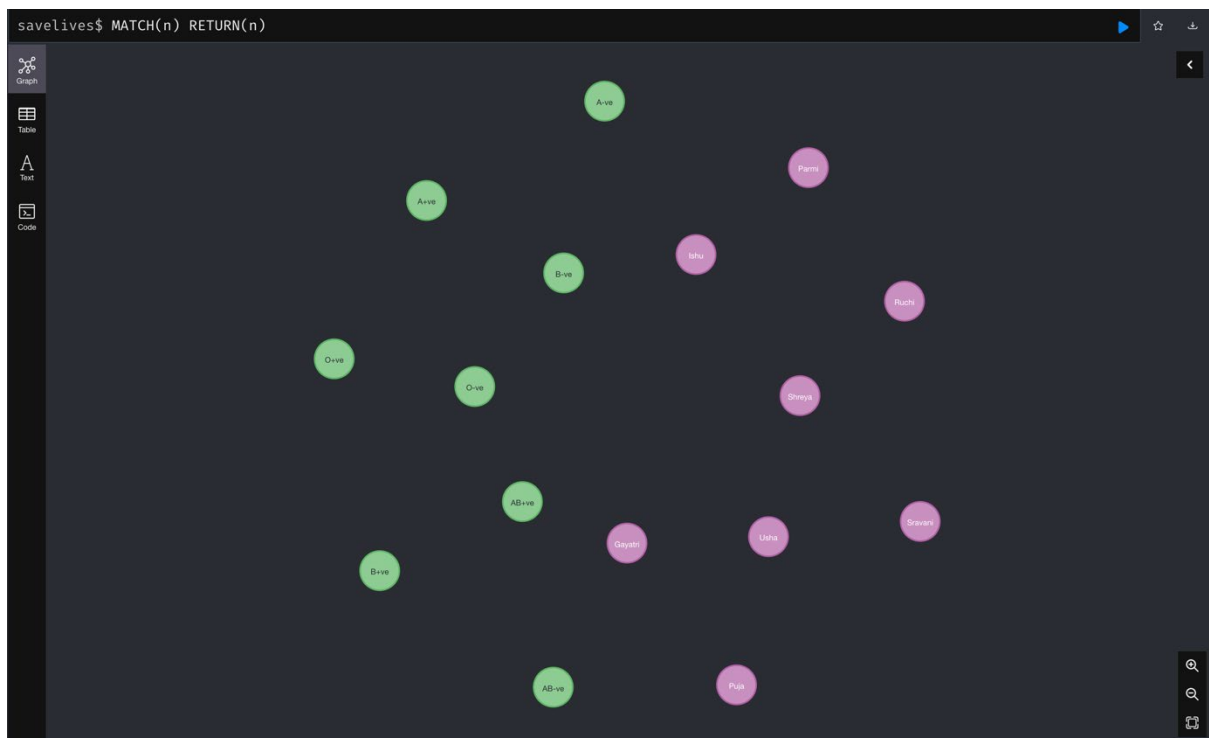
```
savelives$ CREATE(n:Person{name:'Khutso'})
```

```
savelives$ CREATE(n:Person{name:'Khutso'})
```

Added 1 label, created 1 node, set 1 property, completed after 6 ms.

Table

Here, the green circles (nodes) refer to the 8 blood groups and the purple circles (nodes) refer to the existing donors at Save Lives who got registered themselves as Blood Donors.



Let's look at the “*Relationships*”,

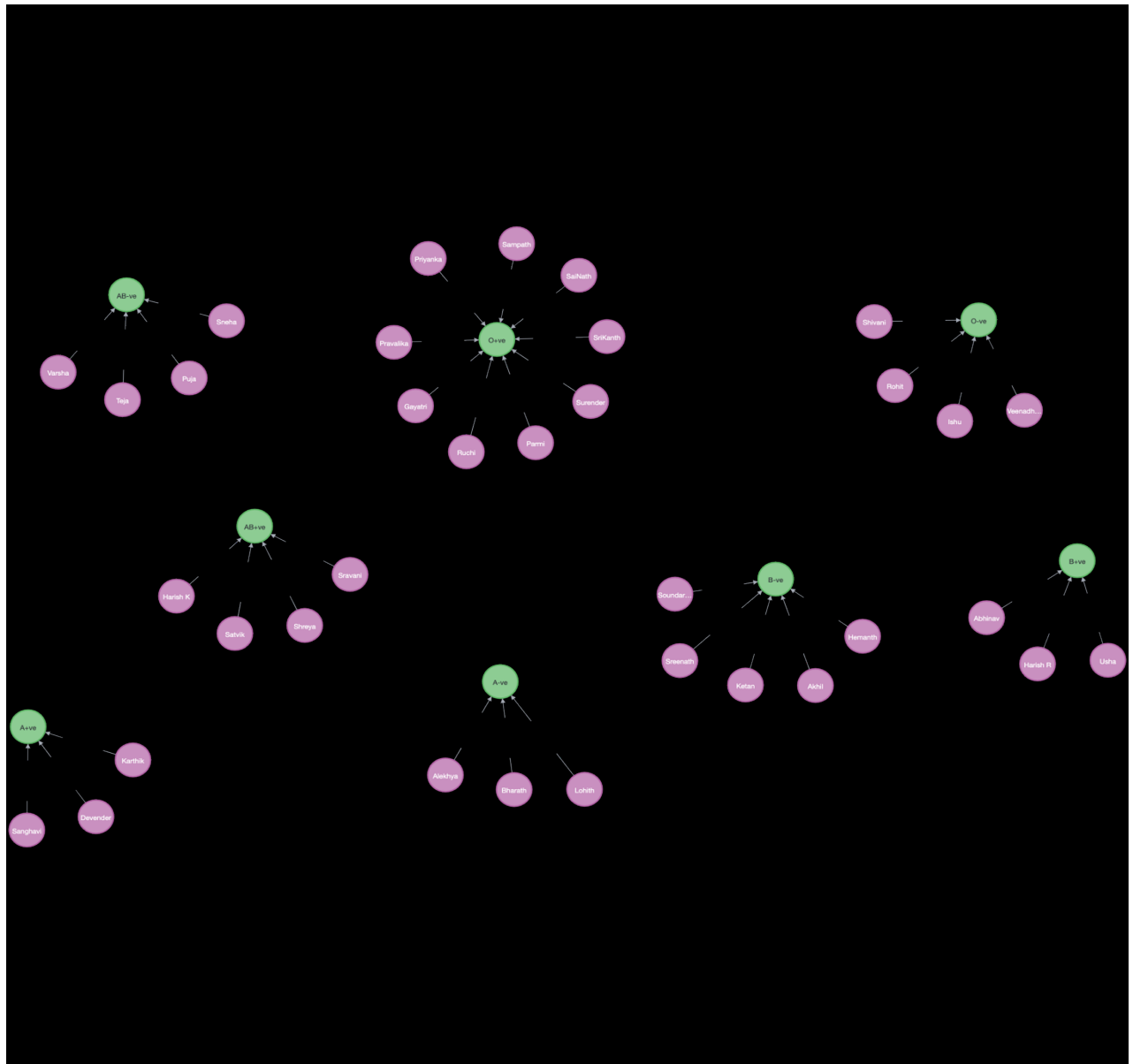
In the following query, we are mapping each person in the graph database to their respective blood groups using a relationship/edge – *BELONGS_TO*.

```
1 MATCH(p:Person),(g:Group)
2 WHERE p.name = 'Parmi' AND g.name = 'O+ve'
3 CREATE (p)-[bt:BELONGS_TO]-(g)
```

Created 1 relationship, completed after 4 ms.

Table

Now, we can see the green dots representing the blood group and all the lines/edges joining each person to their respective blood groups.

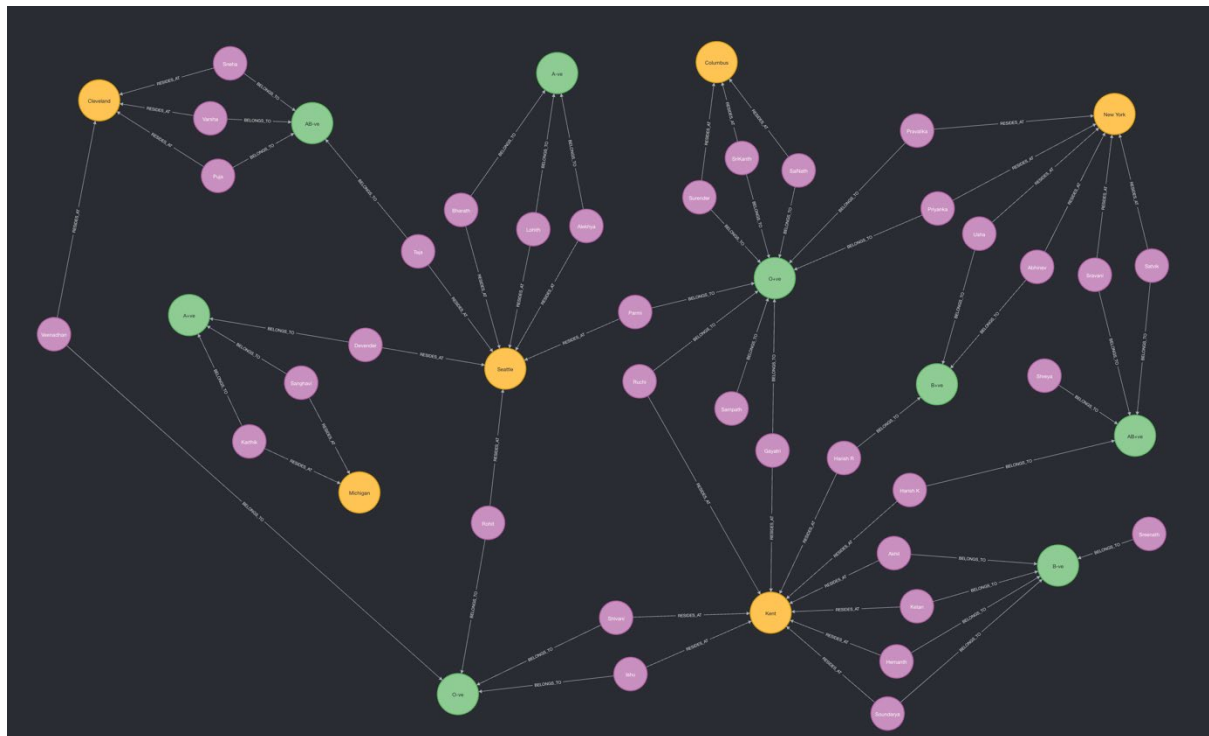


Finally, we are creating another node called ***“Location”*** to map each person to their available locations by a relationship/edge called ***“RESIDES_AT”*** so that we can use this information at the right time to contact the nearest available ***“Person”***,

For each location thus created, we get to have an understanding of which donor is available to donate, his/her blood group and at which place.

By this, we can track the donors based on the requirement and later direct the donor to the hospital as per their location availability.

Yellow – Locations, Green – Blood Group, Purple - Persons



Example: University Edge Medical Centre dropped a request for 5 Units of O+ve Blood Cells for Chris McIair, Save Lives can now take the help of this graph and direct Gayatri and Ruchi with the above request so that they can save the life of Chris at the right time.

Also, we can use ***MATCH(n) & RETURN(n)*** commands to query the desired output without any complexities like SQL. That's the key advantage of using NoSQL Databases for ***Save Lives***, especially a graph database like Neo4j is going to be more beneficial.

