Recommendation System Based on Graph Database Techniques

Noor Mohammedali
The University of Northampton

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 10 | Oct 2019 www.irjet.net p-ISSN: 2395-0072

Recommendation System Based on Graph Database Techniques

Noor Mohammedali

Dept. of Computer Network Engineering, University of Northampton, Northampton, UK

---***---

Abstract – *Nowadays, the World's leading Graph Database to collect a massive amount of data that generate every day on the search engine and then used these data to announce new staff to the users. The aim of this research to propose a graph-based model for a recommendation system that deals with the restaurants and the customers data loading from a CSV file that upload to cloud service on a google drive using google spreadsheet. In this paper, a recommendation system presented and described based on Neo4j techniques using a graph database. These techniques explained, organized and implemented in a sequence order by starting with Content-based filtering, Collaborative filtering and Hybrid filtering to develop a recommendation approach in an efficient way.*

Key Words: Recommender system, NoSQL, Graph Database, algorithms, Graph-based model, Content-based filtering, Collaborative filtering, Hybrid filtering.

1. INTRODUCTION

NoSQL Graph database has used with a real-time system such as search engine, online marketing, social media and recommendation system. So, the recommendation system uses in a real-time to represent the newest product that seemlier to the user search history or recommend a thing that customers' friend like. Recommendation system uses many algorithms to recommend items to the user may like it or not. The benefits from using these algorithms to discover a new item because they have not come across it before because of the massive amount of information that these web sites have. In [1], the author mention that the recommendation system came under unsupervised learning, that means it a subcategory of the machine learning mining algorithm.

The advantages of using a Graph database: 1. including new association sorts will be easy. 2. evolving existing nodes and their relationships is comparative will information migration, since these transforms will must make carried for each node also every association in the current information. 3. Performance: - Chart databases would exceedingly useful at it goes with inquiry performance, Indeed to profound and complex queries. 4. Flexibility: The structures and pattern of a graph model flex as applications and industries change. The admin can add data to the existing graph structure without endangering current functionality.5. Agility: Advanced chart databases need aid prepared to frictionless improvement What is more graceful frameworks upkeep. On other hand, there are some disadvantages such as 1. they are not proficient toward transforming high volumes of transactions. 2.they need aid not great during taking care of queries that compass the whole database. 3. A graph database is exactly a data store and does not provide a business-facing client interface to an inquiry or deal with associations. Furthermore, it will not provide advanced match and survivorship purpose alternately information nature competencies. 4. All data stored on one server.5. A graph database is not optimized for large-volume analytics queries commonplace from claiming information warehousing.

All these data stored in a node and the connection of these nodes called a relationship. Also, it is used different types of clauses to get the solution from the requirement queries. Writing a cypher query requires a less time execution, easy to read for others compering to SQL query. Besides, converting a Cypher query to SQL query, especially for the complex query that contains many JOINS. Moreover, there are many ways to load data as a CSV file from cloud storage like Dropbox and Google drive. In this work will use a google spreadsheet to save our data then will load them in Cypher command. [2]

The algorithms that used to implement this system will discover in the following section with some explanation for each algorithm as well as the advantage and disadvantage for graph database will be explained.

This model tackle with restaurants and customers. Each restaurant deals with a different type of meals, and it has many customers that ordered this meal. The restaurant will get reviews from the customers about their meal. The meal that has a high rate will recommend to the customer. Sometime will be recommended a meal to the customer based on their location to find the nearest restaurant to them. Furthermore, how these restaurants are recommended by the system to the user depend on the amount of the review that the restaurant has or by friend recommendation will be seen.

The remaining part of this paper is organized as follows: Section II discusses the previous work. In Section III, several techniques that support this system will be explained. Section IV explains the content of the system model. In Section, V explains the implementation and the result. Finally, in Section VI contains the paper conclusion and future work.

2. LITERATURE REVIEW

In this section will cover all the background research then will list some of the use cases for the graph database and explaining some of them. In this paper [3], they present a graph-based recommendation system for the digital library. They combined a two-layer graph model to implement their model. Their result evaluation is done based on a machine learning algorithm called hold out test result.

Graph databases map more directly to the object-oriented programming models Besides might snappier will profoundly familiar with data sets and graph queries. Furthermore, they typically support ACID transaction properties in the same way as most RDBMS. There are many types of graph databases such as Neo4J, Infinite Graph, OrientDB or FlockDB. In addition, their current project: Online analytical, Online shop and Marketing online like, Amazon [4].

Furthermore, there are many use cases of graph databases have been existed, such as Fraud Detection, Knowledge Graph, Network and IT Operation, Real-Time Recommendation Engines, Master Data Management, Social Network, Identity and Access Management, Geo Routing (Public transport), Bioinformatics, Mobile Social Application, Portfolio Analytics, Insurant Risk analysis, Network cell Analysis and Connect management and access control. So, this a little explanation about some of the use cases such as Social Network and Network and IT Operation that is very important in the business outcomes side and their challenges. [5]

1.1 Network and IT Operation

In this paper [6], Neo4j explained the size and unpredictability of the network and IT infrastructure required a configuration management database (CMDB), and it is better than using relational databases. Using Neo4j for the network and data center helps us to troubleshoot, analysis the data in the network. As well as, we can see the capacity and outage in the network. So, the graph database allowed us to use monitoring tools and connected them with different network or data center to see the complexity of the relationships between the networks and the data center operations. The business outcomes from using Graph database in a Network and IT operation will be:

- **Impact analysis and network planning:**

  Fined alternative routes in a network if one of the nodes failures, incursion or outages.

- **Root-cause analysis:**

  Any network or infrastructure problem will quickly identify the root cause of tracing back dependencies easily. Furthermore, making the IT infrastructure up will provide service desks visibility into all components and relationships in the network.

- **Routing and quality-of-service (QoS) mapping:**

  The best location in the network will depend on a best, shortest or least-busy path to Introduce a new service in this location, then will complete quality-of-service (QoS) mapping from the base level to a high level in the network.

- **IT infrastructure management:**

  Virtual infrastructure components in the network will be mapping to the IT services, this optionally mapping up to cost centres.

On other hand, there are many challenges when used in the Graph database in a Network and IT operation such as:

- **Highly interrelated elements**

  The right technology cannot manage a network efficiently even if it provides security-related access, optimization a network, application infrastructure or data centre.

- **Non-linear and non-hierarchical relationships**

  There are different types of the relationship between the nodes in the network such as, linear and hierarchical and non-linear and non-hierarchical so it is making it difficult to use traditional RDBMS, these relationships become complex to describe.

- **Growing physical and virtual nodes**

  Organization must develop systems that support the existing and future requirements because the network users and services are increasing day by day.

1.2 Social Network

In [6] [7], a relationship in a social network based on activity and it is already functioning as graphs, so no needless work such as store a data in a table then make a graph database to model these data. They are using Neo4j to reduce the time and improve the quality of the social network. For instance, Twitter stores relationships between users to provide their tweet service. Social media network is based on location services, knowledge representation and recommendation systems to get the complex relationships between the nodes. The business outcomes from using Graph database in a Social Network will be:

- **Collaboration and Sharing.**

  In a social media can connect and share things that make the application interactive for a multitude of worldwide users.

- **Friend-of-friend recommendations**

  Recommendation systems so powerful because it helps users to connect and build networks faster.

- **Discover unique relationships.**

  The database helps the developer to understand how people are connected in a social network, even when he does not know their initial relationships.

- **Faster time to market.**

  A graph database in the social network will reduce the cost and decrease time to market because of social networks already graphs.

On another hand, there are many challenges when the Graph database used in a Social Network such as:

- **Highly dynamic networks.**

  Networks change quickly, so the applications must be updated to detect new trends.

- **The high density of connections.**

  Networks are thickly associated and become more over time and require parsing this relationship rapidly for better benefits of the business insights.

- **Relationships are equally important.**

  A social network, the user's behaviour and the relationships between users significant to process the data.

1.3 Public Transport

The Graph database Used in transportation network such as railway network. Find a path from start station to a destination depending on the schedule that the railway has. The travelling time for the train depends on the time of the journey. In graph database will link the station directly, storing schedule information in station node or edge [8].

In a graph below, the blue nodes for the station node, the green nodes for the route nodes and it contain a stop zoon in a specific time. As well as, there is leaving time between the route nodes. Through the transfer time will change the train at a specific station, new edges are created between the route nodes and the station nodes. The weight of each edge **$(w(\tau)=\tau A(i+1)-\tau A(i))$** which is the equivalent of waiting time at station plus travel time.

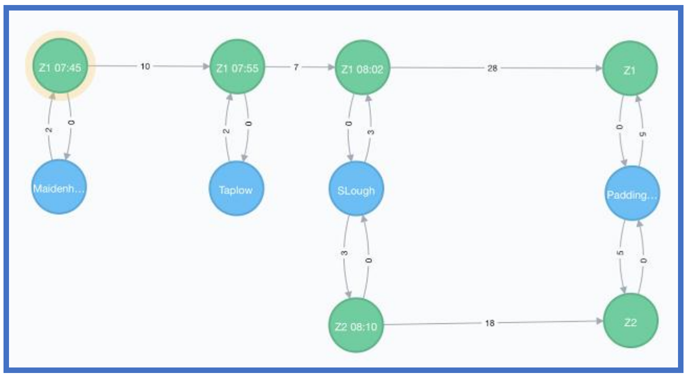


Fig. 1 Graph DB for railway network

3. Recommendation Techniques

There are many techniques used to build the recommendation system that represents in the diagram below:

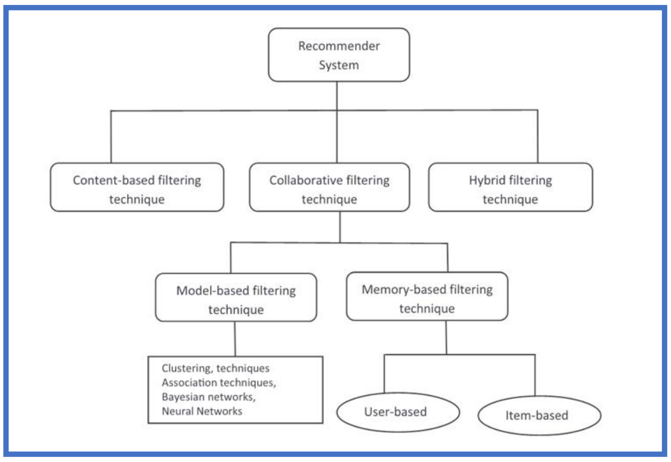

Recommender System
Content-based filtering technique
Collaborative filtering technique
Hybrid filtering technique
Model-based filtering technique
Memory-based filtering technique
Clustering, techniques
Association techniques,
Bayesian networks,
Neural Networks
User-based
Item-based

Fig. 2 Recommendation techniques [9].

3.1 Content-based Filtering Technique.

Content-based filtering technique is used to analyses a set of items that rated before by the user or other users that interest in to generate a prediction. This technique used with a web page, news to make some recommendation for the user. these recommendations done based on the user evaluation with some item in the past. To generate a meaning full recommendation will use different types of models to find the similarity between documents such as Vector Space Model, Probabilistic models and Neural Networks. In this recommendation, we do not need a user profile because it was done by using machine learning or statistical analysis. The advantage of this technique is 1. Ensures privacy that means non-need for sharing the user profile. 2. The recommendation has done with a short period. 3. Recommend new items even if there is no rate in it.

On the other hand, the disadvantage: 1. Content-based filtering depends on the item metadata. 2. limited content analysis means need a detailed description of the items. 3. Content overspecialization [10].

3.2 Collaborative Filtering Technique.

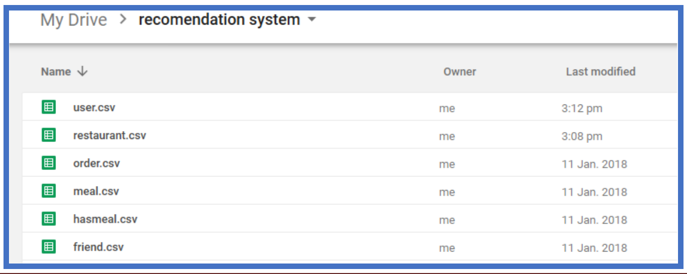
This technique cannot easily describe by using metadata because it is a domain-independent prediction technique. This technique uses to build a database between the user-item as an array of performance for items by users then will calculate the similarity by matching the users with relevant items after that will make a recommendation based on the similarity result between users. Recommended item for the user depends on how similar users rate this item or other users rate another item. Moreover, this technique has two types: Model-based filtering technique that depends on the clustering techniques, association techniques, Bayesian network and neural network and Memory-based filtering technique: such as user-based, item-based. Also, it has some advantages, such as 1. Need the users rating to find the similarity between them to set the recommendation. 2. Display recommendation items that unknown user like or rate. 3. A new item can be suggested even if they are no rating. On the other hand, it has a disadvantage: 1. For the new user, the recommendation will not be provided correctly. 2. Items will not be recommended if there is no enough information to discriminate [11].

3.3 Hybrid filtering technique

This technique is done by combining many techniques to avoid limitation in the systems. So, the result will be more accurate rather than a single algorithm. Each technique has weaknesses could be overcome by complaining it with another technique. There are different ways to do the combination: 1. Implement the algorithms separately then will combine the result. 2. Use content-based filtering in the collaborative filtering approach. 3. Use collaborative filtering in content-based filtering approach [12].

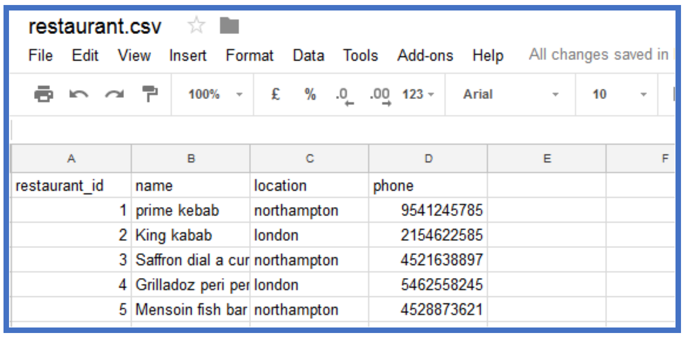
4. System Model Node and Relationships

In this section, Google sheets will be used as a cloud service to save all system data and their relationship as a CSV file and used them remotely by loading them in Neo4j desktop server. The screenshot below displays all the files used in our system.



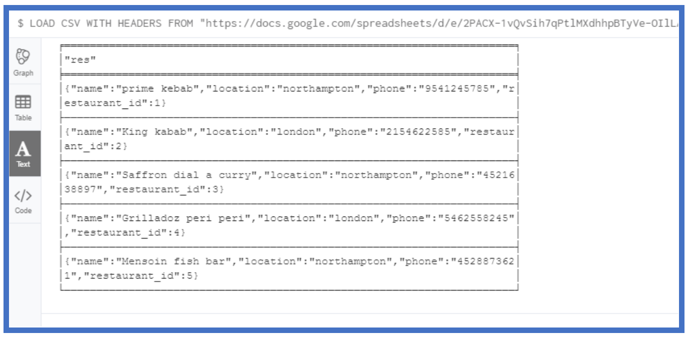
Name	Owner	Last modified
user.csv	me	3:12 pm
restaurant.csv	me	3:08 pm
order.csv	me	11 Jan. 2018
meal.csv	me	11 Jan. 2018
hasmeal.csv	me	11 Jan. 2018
friend.csv	me	11 Jan. 2018

restaurant.csv contains all restaurant information such as, restaurant_id, restaurant name, location and phone number as in a screenshot below.


restaurant.csv
File Edit View Insert Format Data Tools Add-ons Help All changes saved in
100% £ % .0 .00 123 Arial 10

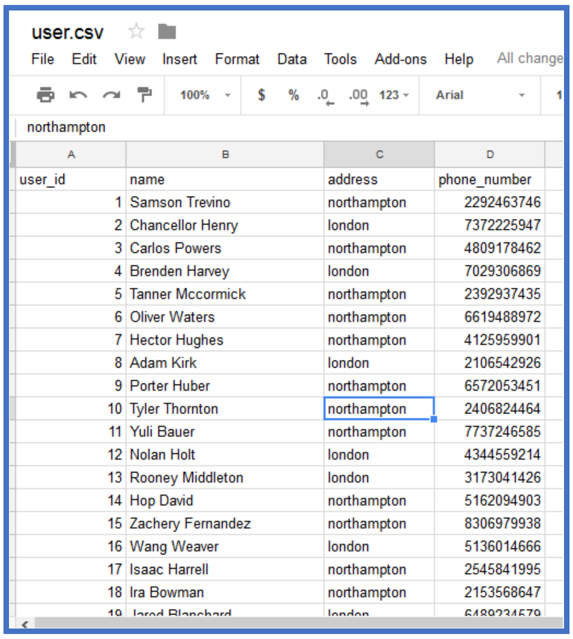
A	B	C	D
restaurant_id	name	location	phone
1	prime kebab	northampton	9541245785
2	King kabab	london	2154622585
3	Saffron dial a cur	northampton	4521638897
4	Grilladoz peri per	london	5462558245
5	Mensoin fish bar	northampton	4528873621

These data will be loaded from the file in graph model in Neo4j. From the screenshot below, we can see the properties in each node.



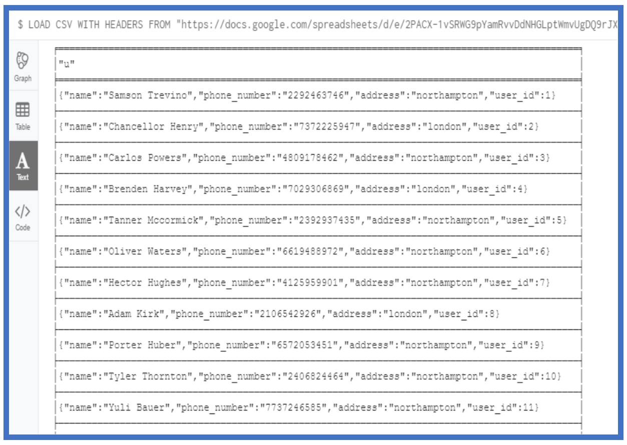
```
$ LOAD CSV WITH HEADERS FROM "https://docs.google.com/spreadsheets/d/e/2PACX-1vQvSih7qPtlMXdhhpBTyVe-OIlL
"res"
{"name":"prime kebab","location":"northampton","phone":"9541245785","r
estaurant_id":1}
{"name":"King kabab","location":"london","phone":"2154622585","restaur
ant_id":2}
{"name":"Saffron dial a curry","location":"northampton","phone":"45216
38897","restaurant_id":3}
{"name":"Grilladoz peri peri","location":"london","phone":"5462558245"
,"restaurant_id":4}
{"name":"Mensoin fish bar","location":"northampton","phone":"452887362
1","restaurant_id":5}
```

users.csv that contains user_id, name of the user, their address and phone number as on screenshot below.


user.csv
File Edit View Insert Format Data Tools Add-ons Help All change
100% $ % .0 .00 123 Arial
northampton

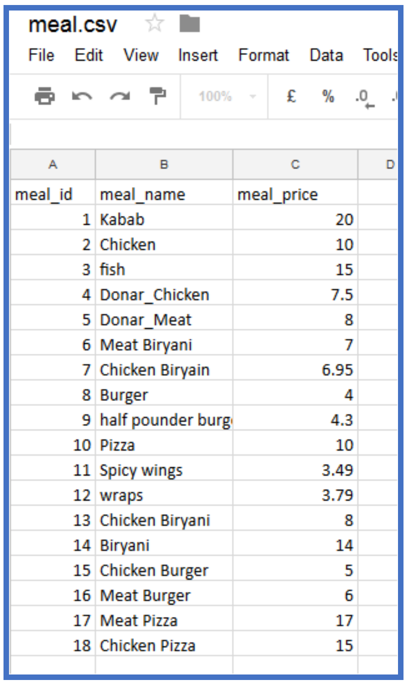
A	B	C	D
user_id	name	address	phone_number
1	Samson Trevino	northampton	2292463746
2	Chancellor Henry	london	7372225947
3	Carlos Powers	northampton	4809178462
4	Brenden Harvey	london	7029306869
5	Tanner Mccormick	northampton	2392937435
6	Oliver Waters	northampton	6619488972
7	Hector Hughes	northampton	4125959901
8	Adam Kirk	london	2106542926
9	Porter Huber	northampton	6572053451
10	Tyler Thornton	northampton	2406824464
11	Yuli Bauer	northampton	7737246585
12	Nolan Holt	london	4344559214
13	Rooney Middleton	london	3173041426
14	Hop David	northampton	5162094903
15	Zachery Fernandez	northampton	8306979938
16	Wang Weaver	london	5136014666
17	Isaac Harrell	northampton	2545841995
18	Ira Bowman	northampton	2153568647
19	Jarod Blanchard	london	6489234579

the screenshot below represents the result after load the file.



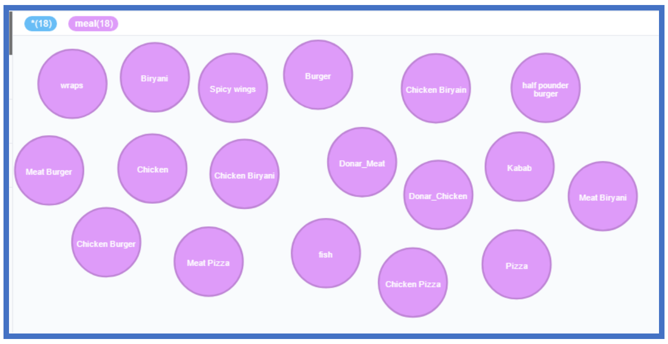
```
$ LOAD CSV WITH HEADERS FROM "https://docs.google.com/spreadsheets/d/e/2PACX-1vSRWG9pYamRvvDdNHGLptWmvUgDQ9rJX
"u"
{"name":"Samson Trevino","phone_number":"2292463746","address":"northampton","user_id":1}
{"name":"Chancellor Henry","phone_number":"7372225947","address":"london","user_id":2}
{"name":"Carlos Powers","phone_number":"4809178462","address":"northampton","user_id":3}
{"name":"Brenden Harvey","phone_number":"7029306869","address":"london","user_id":4}
{"name":"Tanner Mccormick","phone_number":"2392937435","address":"northampton","user_id":5}
{"name":"Oliver Waters","phone_number":"6619488972","address":"northampton","user_id":6}
{"name":"Hector Hughes","phone_number":"4125959901","address":"northampton","user_id":7}
{"name":"Adam Kirk","phone_number":"2106542926","address":"london","user_id":8}
{"name":"Porter Huber","phone_number":"6572053451","address":"northampton","user_id":9}
{"name":"Tyler Thornton","phone_number":"2406824464","address":"northampton","user_id":10}
{"name":"Yuli Bauer","phone_number":"7737246585","address":"northampton","user_id":11}
```

meal.csv that contains meal_id, meal_name and meal_price as in a screenshot below.



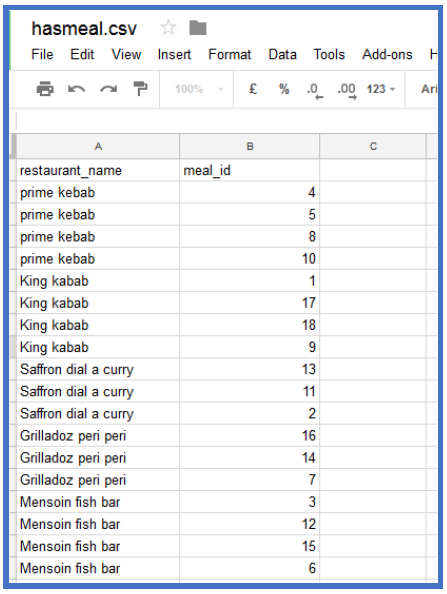
A	B	C
meal_id	meal_name	meal_price
1	Kabab	20
2	Chicken	10
3	fish	15
4	Donar_Chicken	7.5
5	Donar_Meat	8
6	Meat Biryani	7
7	Chicken Biryain	6.95
8	Burger	4
9	half pounder burg	4.3
10	Pizza	10
11	Spicy wings	3.49
12	wraps	3.79
13	Chicken Biryani	8
14	Biryani	14
15	Chicken Burger	5
16	Meat Burger	6
17	Meat Pizza	17
18	Chicken Pizza	15

In the screenshot below, Neo4j result after load the CSV file.



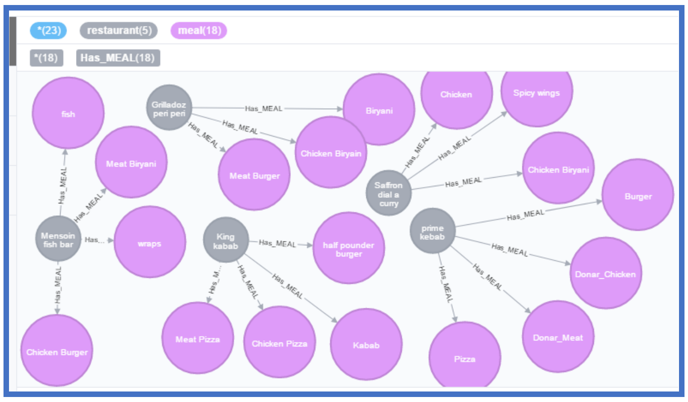
Moreover, this system has a different type of relationship that saved in csv file as well:

1- Restaurant has meal: this relationship saved in a file called hasmeal.csv that contains restaurant_name and meal_id as in screenshot below we can see each restaurant many meals.

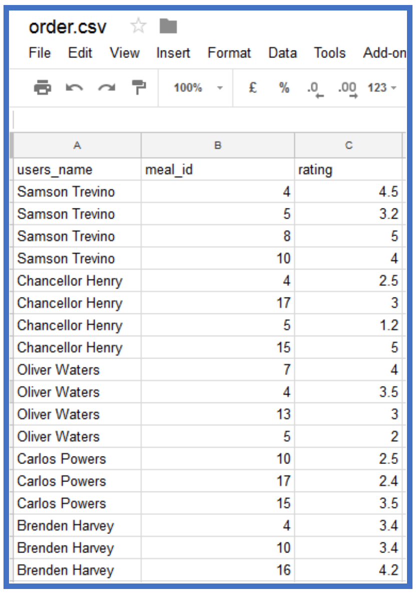


A	B	C
restaurant_name	meal_id	
prime kebab	4	
prime kebab	5	
prime kebab	8	
prime kebab	10	
King kabab	1	
King kabab	17	
King kabab	18	
King kabab	9	
Saffron dial a curry	13	
Saffron dial a curry	11	
Saffron dial a curry	2	
Grilladoz peri peri	16	
Grilladoz peri peri	14	
Grilladoz peri peri	7	
Mensoin fish bar	3	
Mensoin fish bar	12	
Mensoin fish bar	15	
Mensoin fish bar	6	

This file will be loaded in neo4j, each restaurant has many meals as in the screenshot below.



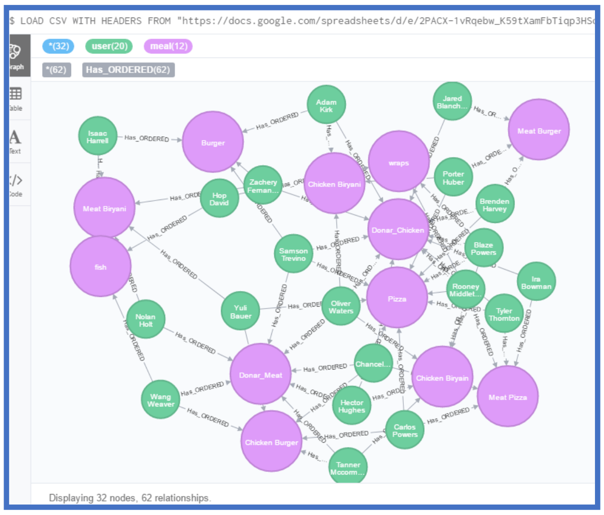
2- Customer has ordered this meal: this is in a file called order.csv that contains users_name, meal_id and rating for this meal as in a screenshot below.



order.csv

users_name	meal_id	rating
Samson Trevino	4	4.5
Samson Trevino	5	3.2
Samson Trevino	8	5
Samson Trevino	10	4
Chancellor Henry	4	2.5
Chancellor Henry	17	3
Chancellor Henry	5	1.2
Chancellor Henry	15	5
Oliver Waters	7	4
Oliver Waters	4	3.5
Oliver Waters	13	3
Oliver Waters	5	2
Carlos Powers	10	2.5
Carlos Powers	17	2.4
Carlos Powers	15	3.5
Brenden Harvey	4	3.4
Brenden Harvey	10	3.4
Brenden Harvey	16	4.2

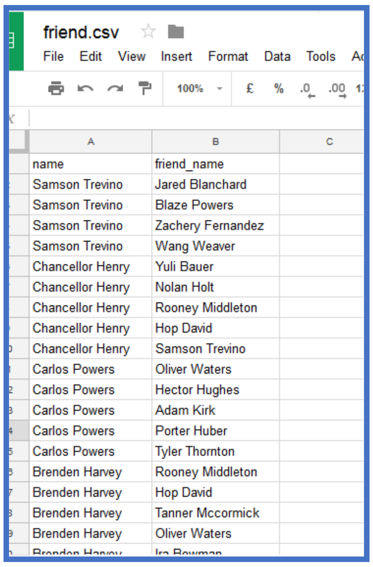
Loading a CSV file in Neo4j, as in result below.



```
$ LOAD CSV WITH HEADERS FROM "https://docs.google.com/spreadsheets/d/e/2PACX-1vRqebw_K59tXamFbTiqp3HS
```

```
Displaying 32 nodes, 62 relationships.
```

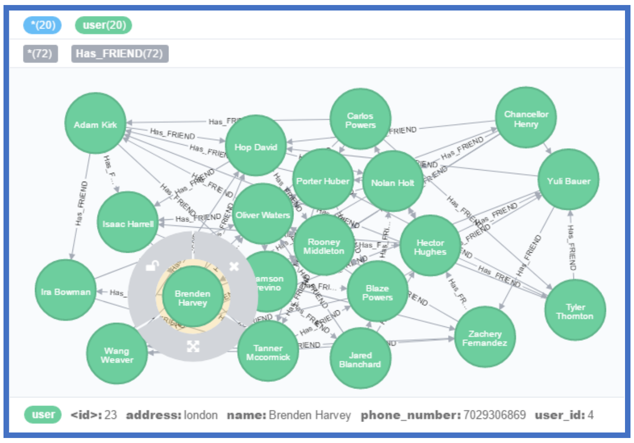
3- Customer has friend: this relationship saved in file called friend.csv that contains the name of the user and their friend name as in a screenshot below.



friend.csv

name	friend_name
Samson Trevino	Jared Blanchard
Samson Trevino	Blaze Powers
Samson Trevino	Zachery Fernandez
Samson Trevino	Wang Weaver
Chancellor Henry	Yuli Bauer
Chancellor Henry	Nolan Holt
Chancellor Henry	Rooney Middleton
Chancellor Henry	Hop David
Chancellor Henry	Samson Trevino
Carlos Powers	Oliver Waters
Carlos Powers	Hector Hughes
Carlos Powers	Adam Kirk
Carlos Powers	Porter Huber
Carlos Powers	Tyler Thornton
Brenden Harvey	Rooney Middleton
Brenden Harvey	Hop David
Brenden Harvey	Tanner Mccormick
Brenden Harvey	Oliver Waters

This file will be loaded in graph database using Cypher query in Neo4j.



```
user <id>: 23 address: london name: Brenden Harvey phone_number: 7029306869 user_id: 4
```

5. Proposed System

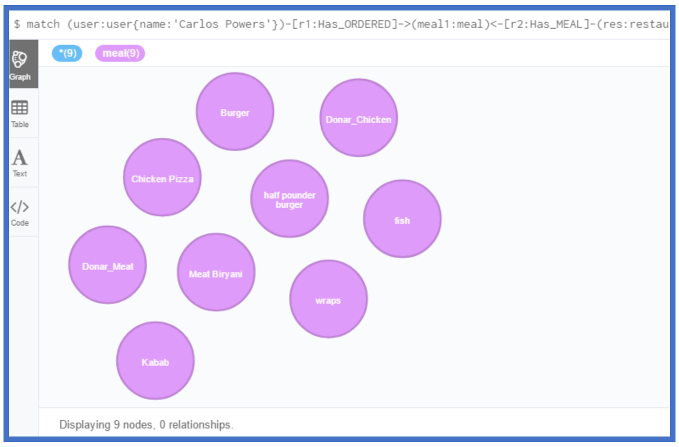
There are many techniques of recommendation system as it explained in the literature review. So, in this section will demonstrate all these techniques using Cypher commands.

5.1 Content-based Filtering Technique.

This technique analyses the attributes of the item to give a new suggestion to other customers. To implement this technique in Neo4j will recommend a meal to a specific user that he did not order before, like in a query below.

```
match (user:user{name:'Carlos Powers'})-
[r1:Has_ORDERED]->(meal1:meal)<-[r2:Has_MEAL]-
(res:restaurant)-[r3:Has_MEAL]-(meal2:meal)
where NOT (user)--(meal2)
RETURN meal2
```

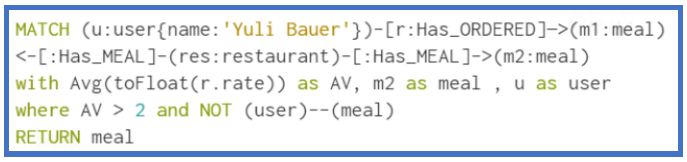
From the query above 'Carlos Powers' ordered meal from the restaurant and this restaurant has many meals that he did not ordered before so we will recommend this meal to him.



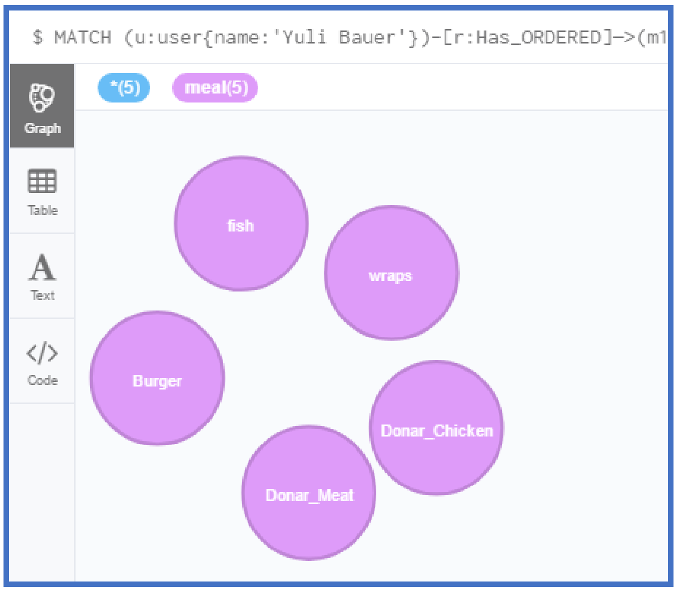
```
$ match (user:user{name:'Carlos Powers'})-[r1:Has_ORDERED]->(meal1:meal)<-[r2:Has_MEAL]-(res:restau
```

From the screenshot, we can see all the meals that recommended to the user.

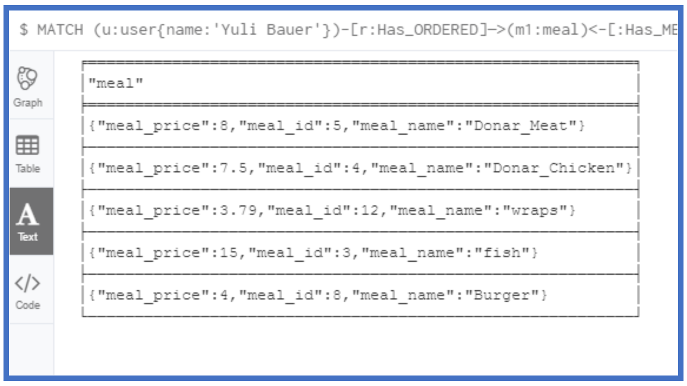
For the second query will return all meals that rated greater than 2 and the user not rate or order these meals before.



```
MATCH (u:user{name:'Yuli Bauer'})-[r:Has_ORDERED]->(m1:meal)
<-[:Has_MEAL]-(res:restaurant)-[:Has_MEAL]->(m2:meal)
with Avg(toFloat(r.rate)) as AV, m2 as meal , u as user
where AV > 2 and NOT (user)--(meal)
RETURN meal
```



```
$ MATCH (u:user{name:'Yuli Bauer'})-[r:Has_ORDERED]->(m1
```



```
$ MATCH (u:user{name:'Yuli Bauer'})-[r:Has_ORDERED]->(m1:meal)<-[:Has_ME

"meal"
{"meal_price":8,"meal_id":5,"meal_name":"Donar_Meat"}
{"meal_price":7.5,"meal_id":4,"meal_name":"Donar_Chicken"}
{"meal_price":3.79,"meal_id":12,"meal_name":"wraps"}
{"meal_price":15,"meal_id":3,"meal_name":"fish"}
{"meal_price":4,"meal_id":8,"meal_name":"Burger"}
```

From the screenshot see all the meals properties.

5.2 Collaborative Filtering Technique.

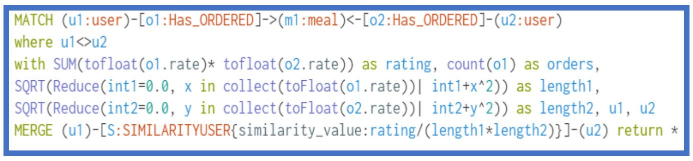
This technique uses a user-to-user collaborative method, and the recommendation has done depend on the users' experience in this product. Memory-based filtering technique will be implemented in this part:

User-based: this technique will be used to find the similarity between users by comparing their ratings on the same item.

User to use collaborative filtering that will be used within the cosine similarity algorithm. The cosine similarity between two vectors to calculate the cosine of the angle between them on the Vector Space. This is the cosine similarity formula will generate a metric that shows how two documents are related to each other by looking at the angle instead of magnitude, that has a range between -1 and 1, -1 means perfectly dissimilar (opposite directions) and 1 means perfectly similar means (same direction) [13].

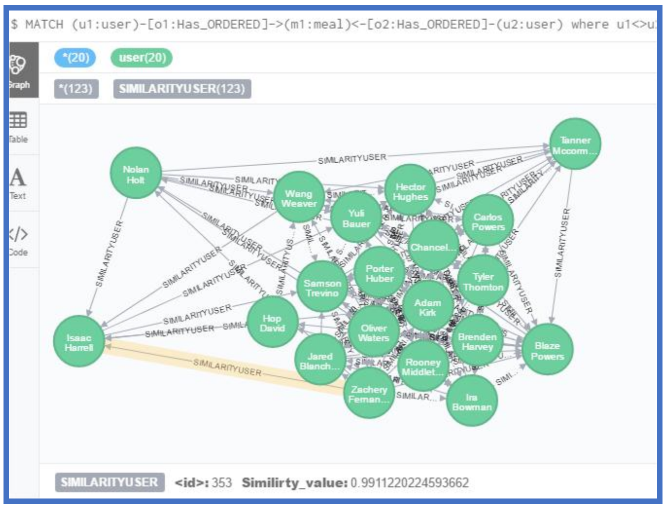
$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

This query will be run by using Neo4j, after implementing this technique in Cypher command: by getting the other users that have a relationship called [: SIMILIRTYUSER] with specific user, the similarity will be done by using the code below. The algorithm will be converted to aggregation function and other expression that the Neo4j understand as in the screenshot below to get the similarity between users.



```
MATCH (u1:user)-[o1:Has_ORDERED]->(m1:meal)<-[o2:Has_ORDERED]-(u2:user)
where u1<>u2
with SUM(tofloat(o1.rate)* tofloat(o2.rate)) as rating, count(o1) as orders,
SQRT(Reduce(int1=0.0, x in collect(toFloat(o1.rate))| int1+x^2)) as length1,
SQRT(Reduce(int2=0.0, y in collect(toFloat(o2.rate))| int2+y^2)) as length2, u1, u2
MERGE (u1)-[S:SIMILARITYUSER{similarity_value:rating/(length1*length2)}]-(u2) return *
```

From the screenshot below, we can see the similarity between all users when we click on SIMILARTYUSER relationship we can see the similarity value between “Zachery and Isaac”.



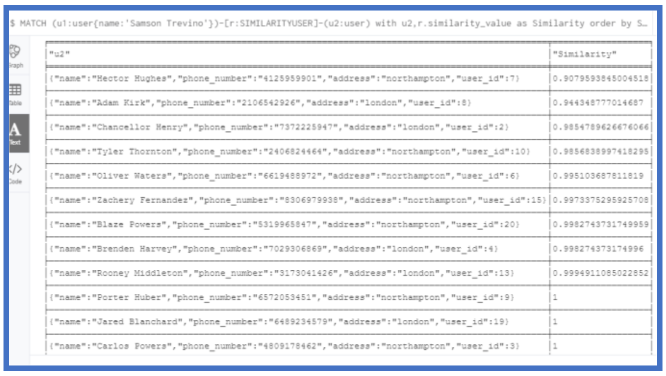
```
$ MATCH (u1:user)-[o1:Has_ORDERED]->(m1:meal)<-[o2:Has_ORDERED]-(u2:user) where u1<>u
```

```
SIMILARITYUSER  <id>: 353  Similirty_value: 0.9911220224593662
```

From the example below will see the similarity between the 'Samson Trevino' and other users.

```
MATCH (u1:user{name:'Samson Trevino'})-[r:SIMILARITYUSER]-(u2:user)
with u2,r.similarity_value as Similarity
order by Similarity
return u2, Similarity
```

From this query will get all users that have a similarity relationship with a specific user 'Samson Trevino'.



```
$ MATCH (u1:user{name:'Samson Trevino'})-[r:SIMILARITYUSER]-(u2:user) with u2,r.similarity_value as Similarity order by S…

"u2"                                                                                  "Similarity"
{"name":"Hector Hughes","phone_number":"4125959901","address":"northampton","user_id":7}        0.9079593845004518
{"name":"Adam Kirk","phone_number":"2106542926","address":"london","user_id":8}                 0.94434877014687
{"name":"Chancellor Henry","phone_number":"7372225947","address":"london","user_id":2}          0.9854789626676066
{"name":"Tyler Thornton","phone_number":"2406824464","address":"northampton","user_id":10}      0.9856838997418295
{"name":"Oliver Waters","phone_number":"6619488972","address":"northampton","user_id":6}        0.99510368781819
{"name":"Zachery Fernandez","phone_number":"8306979938","address":"northampton","user_id":15}   0.9973375295925708
{"name":"Blaze Powers","phone_number":"5319965847","address":"northampton","user_id":20}        0.9982743731749959
{"name":"Brenden Harvey","phone_number":"7029306869","address":"london","user_id":4}            0.99827437317 4996
{"name":"Rooney Middleton","phone_number":"3173041426","address":"london","user_id":13}         0.9994911085022852
{"name":"Porter Huber","phone_number":"6572053451","address":"northampton","user_id":9}         1
{"name":"Jared Blanchard","phone_number":"6489234579","address":"london","user_id":19}          1
{"name":"Carlos Powers","phone_number":"4809178462","address":"northampton","user_id":3}        1
```

Each user ordered a meal from many restaurants and give a rate for each meal. From the table below we can see the similar rate that different users give to the same meal.

Table 1 Similar rate from different users to the same meal

Meal User	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Samson Trevino				4.5	3.2			5		4							
Chancellor Henry				2.5	1.2										5		3
Carlos Powers										2.5					3.5		2.4
Brenden Harvey				3.4						3.4						4.2	
Tanner McCormick					4		2.5			2.1					2.5		
Oliver Waters				3.5	2		4						3				
Hector Hughes					4.2		2.5			2.1							
Adam Kirk				2.1				5					2				
Porter Huber										3.5			4		2.4		
Tyler Thornton				4						5							2
Yuli Bauer						2				3					3.5		
Nolan Holt						2.5	1.2										
Rooney Middleton				2.4			3.2			2		2.1					5
Hop David			4									2.1					
Zachery Fernandez				2.5	1.2			2.4									
Wang Weaver			2.4		3.2										2.4		
Isaac Harrell			2			1.5		3.5									
Ira Bowman				4.2													3
Jared Blanchard				2.5												2	
Blaze Powers				4.5			3			4.5		3.2					

Item based: will be used to find the similarity between items.

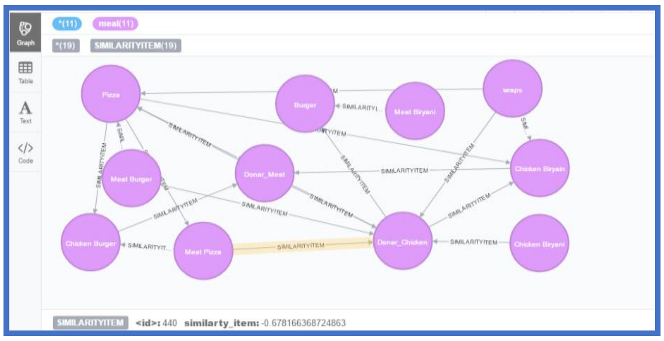
This technique uses to recommend a new item based on the similarity between items. As well as, the similarity between a pair of vectors will use adjusted cosine similarity algorithm [9].

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R_u})(R_{u,j} - \bar{R_u})}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R_u})^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R_u})^2}}.$$

In item-based measure the Similarity between meals have the similar rate. similarity

```
MATCH (m1:meal),(m2:meal)where m1<>m2
match (m1)<-[r:Has_ORDERED]-(u:user)
with AVG(toFloat(r.rate)) as m1rate,m1,m2
match (m2)<-[r:Has_ORDERED]-(u:user)
with AVG(toFloat(r.rate)) as m2rate,m1,m2,m1rate
match (m1)<-[r1:Has_ORDERED]-(u:user)-[r2:Has_ORDERED]-(m2)
with SUM((toFloat(r1.rate)-m1rate)*(toFloat(r2.rate)-m2rate)) as summ1m2,
SQRT(SUM((toFloat(r1.rate)-m1rate)^2) * SUM((toFloat(r2.rate)-m2rate)^2)) AS sqrtm1m2,
m1,m2, COUNT(r1) as countrelation
where sqrtm1m2<> 0 and countrelation>1
MERGE (m1)-[q:SIMILARITYITEM{similarity_item:(summ1m2/sqrtm1m2)}]-(m2) return *
```

Run the query in Neo4j.

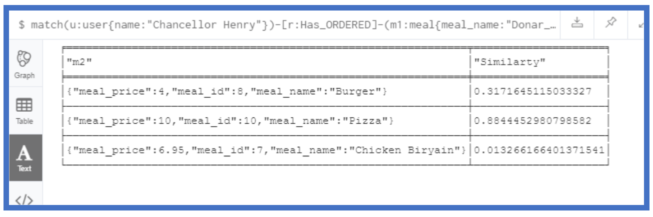


SIMILARITYITEM <id>: 440 similarty_item: -0.678166368724863

From the query below, we can see Meal like Chicken Burger has Similarity rate with other meals.

```
match(u:user{name:"Chancellor Henry"})-[r:Has_ORDERED]-
(m1:meal{meal_name:"Donar_Chicken"})-[s:SIMILARITYITEM]->(m2:meal)
with m2,s.similarity_item as Similarty
return m2, Similarty
```

Run the query in Neo4j to get the similar meal that this user order and have a similarity with another meal.

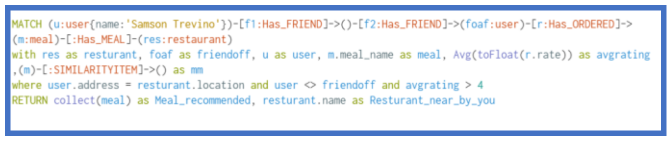


```
$ match(u:user{name:"Chancellor Henry"})-[r:Has_ORDERED]-(m1:meal{meal_name:"Donar_…

╒══════════════════════════════════════════════════════════════╤═════════════════════╕
│"m2"                                                          │"Similarty"          │
╞══════════════════════════════════════════════════════════════╪═════════════════════╡
│{"meal_price":4,"meal_id":8,"meal_name":"Burger"}             │0.3171645115033327   │
├──────────────────────────────────────────────────────────────┼─────────────────────┤
│{"meal_price":10,"meal_id":10,"meal_name":"Pizza"}            │0.8844452980798582   │
├──────────────────────────────────────────────────────────────┼─────────────────────┤
│{"meal_price":6.95,"meal_id":7,"meal_name":"Chicken Biryain"} │0.013266166401371541 │
└──────────────────────────────────────────────────────────────┴─────────────────────┘
```

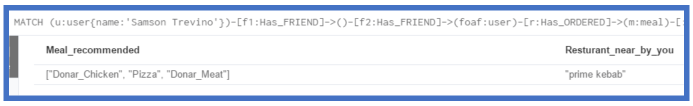
5.3 Hybrid filtering technique

This technique is a compensation between two techniques. This system will Recommend a meal and restaurant near by user depend on user location.



```
MATCH (u:user{name:'Samson Trevino'})-[f1:Has_FRIEND]->()-[f2:Has_FRIEND]->(foaf:user)-[r:Has_ORDERED]->
(m:meal)-[:Has_MEAL]-(res:restaurant)
with res as resturant, foaf as friendoff, u as user, m.meal_name as meal, Avg(toFloat(r.rate)) as avgrating
,(m)-[:SIMILARITYITEM]->() as mm
where user.address = resturant.location and user <> friendoff and avgrating > 4
RETURN collect(meal) as Meal_recommended, resturant.name as Resturant_near_by_you
```

Run the query in Neo4j.



```
MATCH (u:user{name:'Samson Trevino'})-[f1:Has_FRIEND]->()-[f2:Has_FRIEND]->(foaf:user)-[r:Has_ORDERED]->(m:meal)-[
```

Meal_recommended	Resturant_near_by_you
["Donar_Chicken", "Pizza", "Donar_Meat"]	"prime kebab"

6. Conclusions

In this research, after writing a query in graph-based database. We can conclude that graph database faster than relational database. Query in Graph databases depend on latency and that means we can choose the number of graphs that can be explore in a query and is not depend on the amount of data stored. We notice that, a query in graph has attractive meaning so we can understand eastly what this query about and modelling a graph take a less time than relational.

Nevertheless, Learning the basic query and moving to advance concept in NoSQL Graph Database lead me think to develop a recommendation system and move it to the advance. Finally, in the future will improve a solution for the hardest or complex Cypher query that contain a lot of node and relationship. Then will use a document database like MongoDB in advance with complex query. As well as, develop a web page with MongoDB using a PHP or JAVA rather than a normal SQL query.

REFERENCES

[1] S. R. R. N. Dipali R.Dubey, "An integrated recommendation system using fuzzy linguistic approach," no. June, pp. 3541–3546, 2019.

[2] W. McKnight, *Graph Databases*. 2014.

[3] Z. Huang, W. Chung, T. H. Ong, and H. Chen, "A graph-based recommender system for digital library," *Proc. ACM Int. Conf. Digit. Libr.*, no. January, pp. 65–73, 2002.

[4] A. Nayak, A. Poriya, and D. Poojary, "Type of NOSQL databases and its comparison with relational databases," *Int. J. Appl. Inf. Syst.*, vol. 5, no. 4, pp. 16–19, 2013.

[5] J. Webber and I. Robinson, "The Top 5 Use Cases of Graph Databases - Unlocking New Possibilities with Connected Data," *Neo4j Whitepaper*, 2017.

[6] I. Analysis, "How Graph Databases Solve Problems in Network & Data Center Management : a Close Look at Two Deployments," 2013.

[7] A. Chaudhary and A. FAISAL, "Role of graph databases in social networks," no. June, 2016.

[8] D. Delling, B. Katz, and T. Pajor, "Parallel Computation of Best Connections in Public Transportation Networks," *ACM J. Exp. Algorithmics*, vol. 17, pp. 4.1-4.26, 2012.

[9] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egypt. Informatics J.*, vol. 16, no. 3, pp.

261–273, 2015.

[10] R. Van Meteren and M. Van Someren, “Using Content-Based Filtering for Recommendation,” *ECML/MLNET Work. Mach. Learn. New Inf. Age*, pp. 47–56, 2000.

[11] P. Bittner, “Modulempfehlungen über Vo rgänger- und Nachfolgemodule,” *Lect. Notes Informatics (LNI), Proc. - Ser. Gesellschaft fur Inform.*, vol. 246, no. Section 3, pp. 725–733, 2015.

[12] V. Vekariya and G. R. Kulkarni, “Hybrid recommender systems: Survey and experiments,” *2012 2nd Int. Conf. Digit. Inf. Commun. Technol. its Appl. DICTAP 2012*, pp. 469–473, 2012.

[13] N. Team, “The Neo4j Graph Algorithms User Guide v3.5,” *Neo4j*, p. 250, 2019.