NO SQL Database: Graph database

M. A. Elsabagh[1]

[1]*Department of Machine Learning and Information Retrieval, Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh, Egypt.*

Corresponding Author: *Mahmoud Ahmed Mohsen Elsabagh, Faculty of Artificial Intelligence, Kafrelsheikh University 33511 Kafrelsheikh, Egypt.*
Email: Mahmoud_Mohsen@fci.kfs.edu.eg

ABSTRACT: The database of NoSQL is considered one the most significant technology in the current era of computer science especially, with the emergency of big data. The issue of processing and storing data is solved by utilizing the NoSQL databases. Planning to offer references to the users of No SQL databases, this survey examines the characterization, categories, and hypothetical premise of NoSQL dependent on the introduction of the rise, improvement, and development of relational database to NoSQL and the examination of its restrictions of relational databases in the current era. Also, this survey points to a type of NoSQL database called graph database. It can be characterized as those in which the architecture for instances and schema are demonstrated as graphs and data control is described by graph oriented processes and activities and type constructors. It started in eighties and nineties close by object arranged models. Their impact slowly ceased to exist with the rise of other models, specifically XML, spatial, and semi-structured. Lately, the requirement to manipulate information with graph like nature has restored the relationship of this field. The fundamental goal of this review is to introduce the work that has been proposed in the field of graph database.

Keywords: Big data; BASE; No SQL; CAP; Graph database.

1. Introduction and overview of NoSQL Database

With regards to the information society, some computerized marks will be left by anything and anyone that is additionally called data. Quick advancement and the continuous increase in the society of information produce a massive amount of data so, new applications are generated that results to a new data, and a massive amount of information is created due to this continuous increasing of data [1]. It is in this manner that the application and the information grow iteratively, empowering the data to become a snowball.

As per statistical experiment that led in 2012[2], in 2010 the size of data the world over has arrived to ZB, contrasted with 130EB in the year of 2005, implying that the data size has expanded by 10 folds every 5 years. This infers that at this continuous and the highly speed, the volume of the data arrived to 40ZB in 2020 all over the world and so on.

As indicated by the ZDNET yearly technical and specialized report in 2013: [1] in 2013, China made over 0.8ZB which is 2 fold the amount of as that for 2012 and equivalent to the amount created in 2009. In 2020, China approximately generated data over 8.5 ZB that is 10 times for 2013.

Productivity of application advancement, A NoSQL database may give a data model that better meets the application's requirements, that resulting less debug and code [3]. Also, huge scope of data that organizations are thinking that it important to catch and process a huge amount of data more speedy. They are thinking that it costly, if they do it with relational database. The essential explanation is that a single machine is utilized in relational database designing, but it is normally more financial to run enormous to execute a large amount of data and calculating process on less expensive and many smaller servers. Distributed servers are utilized in many NoSQL databases design so, so they make a superior fit for large data situations [3].

Till now, there has no a specific definition for NoSQL database. It is distributed, a general concept of open-source, and non-relational database tools. In 1991, NoSQL begins from the primary form of Berkeley DB [4]. Berkeley DB is based on a key-value pairs to store data and suited for applications with simple data but, require very quick reading and inserting [5]. Researchers from Amazon and Google published many researches on Big Table, describing the thoughts on the novel database they have embraced. BigTable presented the model of column store, exhibiting that the determined storage of data can be stretched out to many of nodes [6]. There are more than 100 of NoSQL databases all over the world [7]. Since 2009 in China, the NoSQL databases have been developed.

Despite the extensive energy for NoSQL development, it is not since a long time ago its turn of events and its database tools should be additionally refined in applications, due to there are both failed and productive applications. Twitter chose in April 2010 to suspend the arrangement to store data by utilizing Cassandra as a replace for MySQL database because Cassandra has

doi: 10.21608/EJAI.2022.132961.1002

inadequate experiences and cases of equal access to large amount of data and that it needs to bear a timeframe before the new produce settles [8].

The rest of this survey is organized as follows. Section 2 theoretical principles of NoSQL database. Section 3 describes the NoSQL classification that has the graph database which is the core of this survey. a new researches based on graph database are also mentioned and its advantages. Section 4 draws conclusions and discusses future work.

2. THEORETICAL PRINCIPLES OF NOSQL

The eventual consistency, BASE hypothesis, and CAP hypothesis are viewed as the three hypothetical establishments for NoSQL.

2.1 CAP

In 2009, The CAP hypothesis was first proposed by Eric Brewer [9]. In 2002, Seth and Nancy demonstrated the accuracy of CAP theory. The CAP hypothesis expresses that it is extremely difficult at the same time for distributed software to give consistency, availability and partition and that it can just meet two of the CAP theory, as illustrated in Fig. 1.

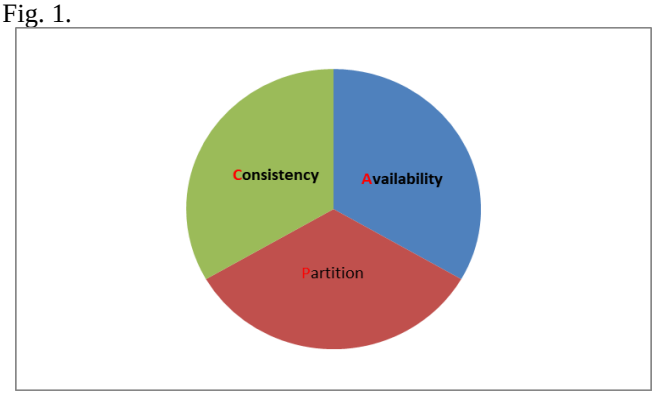

Fig. 1 CAP hypothesis.

The user must pass and tolerate the unsuccessful read and write, if the distributed database has high in consistency and partition because of the unavailability of the system. In the event that the client picks high availability and partition, at that point consistency will be an exceptionally testing issue and may be a dirty read is happen. Naturally, the classic relational database picks availability and consistency, bringing about both helpless level of improvement and scalability. NoSQL gives an answer for accomplish a compromise between the three decisions to meet different necessities.

CAP is not great. Designers in practical applications should adjust between the three choices. For instance, relational database picks availability and consistency with helpless of scalability. Redis [10], Mongo DB [11] and HBase [12] are instances of NoSQL database and are highly described by scalability and consistency but dissatisfied availability. Cassandra and Dynamo are instances of NoSQL databases that are better in availability and scalability but dissatisfied consistency. In this way, it very well might be vital for applications to merge different databases.

2.2 BASE

The genuine importance of the BASE model [13], [14] is Basically Available, which principally has three perspectives: Basically Available (BA), Soft-State (S) and Eventual Consistency (E).

BA: This implies that the distributed system can have a partial failure in certain parts and the remainder of the system keeps on working. For instance, without repeating data if a NoSQL is executing on ten machines and one of the machines is down, then 10% of queries are down, however 90% would pass. NoSQL databases frequently keep various duplicates of data on various machines. This permits that if one of the machines is down, the database still responds to the query [15].

S: Usually in computer science, the concept of Soft-State implies if the data is not updated, it will expire so, in NoSQL tasks, it means the new data may overwrite the old data eventually. S property overlaps with E property [15].

E: The database is inconsistent in many times. For instance, some databases keep numerous duplicates of data on different machines. So, in one time, numerous duplicates may be inconsistent. This can happens when one of the data on a machine is updated and the other still have the old one. E property updates all duplicates of data, but the duplicates are not consistent in meantime [15], [16].

3. NOSQL CLASSIFICATION

This section describes briefly the common types of NoSQL databases that are Key-Value, Document, Column-Family, and Graph databases.

3.1 Key-Value

Key-value databases are the least difficult database of NoSQL and are demonstrated on two segments: **(A) keys** that are identifiers associated with values. The significant rule to recollect keys is that they must be unique. Obviously, any one creating a key-value database at an organization might utilize the same keys at an organization B. This is not an issue due to the isolation of the two databases and the organization A has no chance to write or read to the other data of organization B. They are in a distinct namespaces that are a cluster of keys as shown in Fig. 2. **(B) Values** that are data put away alongside keys. They may be a complex such as binary instance or image and a simple such as integer or string. Key-value databases have no tables, no attributes related to tables, no joins between tables, no query, and no foreign key [15].

doi: 10.21608/EJAI.2022.132961.1002

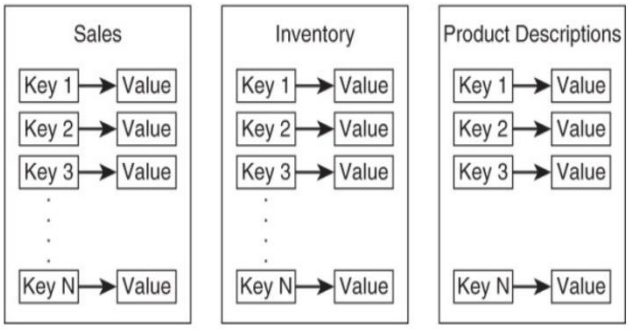

Fig. 2 Key-Value database with various namespaces.

The following subsection presents the document database that is another kind of NoSQL database.

3.2 Document database

Rather than putting away each feature with a different key, a lot of features are stored in a single document. As shown in the following simple instance in JSON:

{
F_Name: “John”,
L_Name: “Mark”,
position: “CIO”,
office_ID: “3-130”,
Phone: “444-333-1234”,
}

Document database does not require a predefined schema before data is inserted to the database is considered one of the most significant features of document database. This feature provides designers of document database flexibility compared with relational databases. It gives query or API to access documents. A one document can have several of values that make the designers do not need the join terminology in relational. Finally, document databases are likely the most common database of NoSQL databases [15].
The following subsection presents the column-family database that is another kind of NoSQL database that have some features with relational database.

3.3 Column Family Databases

The fundamental unit of column-family database is the column that composed of name and value. A row is made up from a collection of column. It may have different or similar columns as shown in Fig. 3.

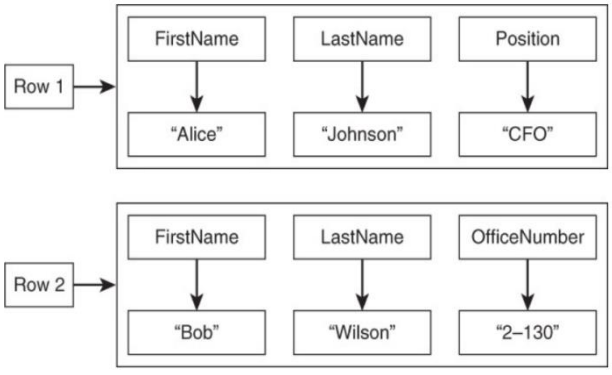

Fig. 3 Column-Family structure.

Enormous number of columns may assist with gathering them into a cluster of related columns. Similar to document database, it does not required a schema before inserting data to database so, columns are inserted as needed by designers of database. This type of database is intended for rows with several columns. It is like to relational database in terms of columns and rows, supports query language and many tasks such as CREATE operation, but it does not support the concept of joining [15].

The next subsection proposes the graph database that is another form of NoSQL database and it is the core of our survey that will be explained in details.

3.4 Graph Databases

One of the most features of relational database is used in different industries for implementing and designing software. The graph database with NoSQL is considered the future of relational database, which is arising beyond of the relational. Relational database has some properties such as: tables are utilized to store data in (columns and rows), easy to utilize through SQL to retrieve data by complex join, reengineering for maintenance, and fast search by using indexing. Because of these previous inherent highlights of SQL and relational database, it is important to investigate and contrast relational with NoSQL strategies to avoid the activity of complex join. Various software researchers are attempting to reengineer the old relational to other structure by means of vertices and edges that stored information easily. So, it is a major difficulties for any industry how rapidly they can re-engineer their old relational to graph database [17]. Fig. 4 presents a graphical perspective of beyond the development of relational.

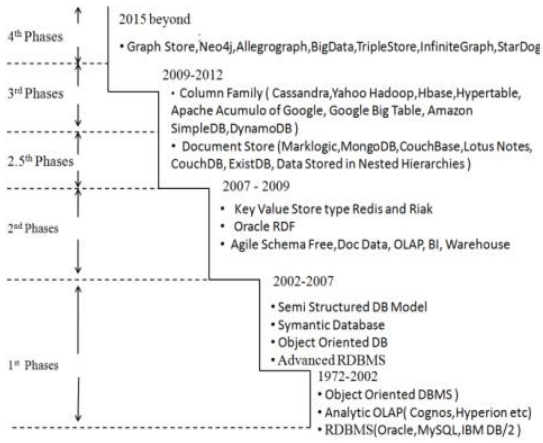

Fig. 4 A graphical representation of database development.

In graph database, vertices that represent the nodes and edges that represent the relationship between nodes are the basic units used in graph database. For example, any person can be a node and friendship is represented as a relationship between them and a distance between towns is also represented by graph database such towns are nodes and the distance is the relationship between them as in Fig. 5.

doi: 10.21608/EJAI.2022.132961.1002

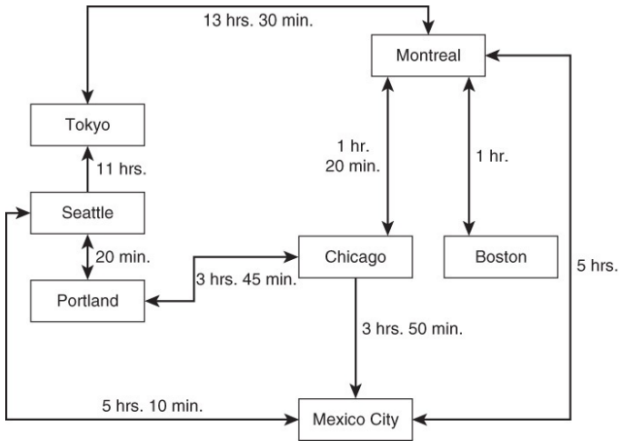

Fig. 5 An example of Graph Database

3.4.1 3.4.1 Pros of graph database

Researchers of graph database presented a lot of experiments in this field. Angles et al. [18] presented a review in graph database and its features. Extraordinary style has been proposed in that review about architecture, storing data, query language, prior researches (pre-NoSQL) in graph database and its truly [18], [19]. The most common advantages of graph database are described as follow:

- This form of database is intended to show the interaction between vertices.
- Every object in database points to another object that increases the speed of operations than in the relational one.
- Effective query is permitted in graph database when ways or edges through nodes are included.
- It considers a more normal demonstrating of data. It is noticeable to user and it permits a characteristic method for accessing data such as geographic data interaction.
- A single node can have the benefit of having the option to keep all the information about an entity and displaying related information by edges associated with it [20].
- Integrity supports consistency of data by collecting the constraints of integrity in schema [21].
- Easy to implement and effective algorithms for performing operations
- The most popular examples of graph database are Neo4J, InfoGrid, and Titan.

3.4.2 Use Cases and Criteria for choosing Graph Databases

This database is useful for displaying networks, additionally utilized to a wide scope of issues [15] such as:

- Displaying Geographic positions.
- Displaying Infectious Diseases, Infectious sicknesses can spread from individual to individual. Vertices describe patient, whereas edges describe relationship between patients.
- Representing fixed entities.
- Representing social media
- One approach to evaluate the usefulness of this database is to decide whether occurrences of elements have relations to different occasions of elements.
- Managing infrastructure of IT and network
- Railways connecting towns
- Highways connecting towns
- Management of business process
- Multimedia systems and visual interface
- Fields that its size, analysis, and management are become problem due to the data are gathered automatically such as network of biology.

3.4.3 Researches based on graph database

Thoba lose et al. [22] presented a recent progress in genomic innovations have empowered high throughput financially creation of 'omics' data from isolation of M.tuberculosis (M.tb), which at that point gets shared by means of various heterogeneous openly accessible database of biology. Though helpful, divided curation adversely impacts the specialist's capacity to use the data by means of unified queries. They proposed Combat-TB-NeoDB, an incorporated M.tb 'omics' knowledge-base. Neo4j is the basis of Combat-TB-NeoDB that was made by restricting the named property graph database to a reasonable philosophy in particular Chado. Combat-TB-NeoDB empowers analysts to execute complex combined queries by connecting unmistakable biological databases [22].

Labat et al. [23] utilized a graph database due to its easy management of the software/ product offering. The explored forms give a system that encourages the support and execution of a product offering. Within the experiment, the system acquires model data related with a multidimensional model of the product offering. Then, a graph database is utilized to store data model by the system as explained in Fig. 6.

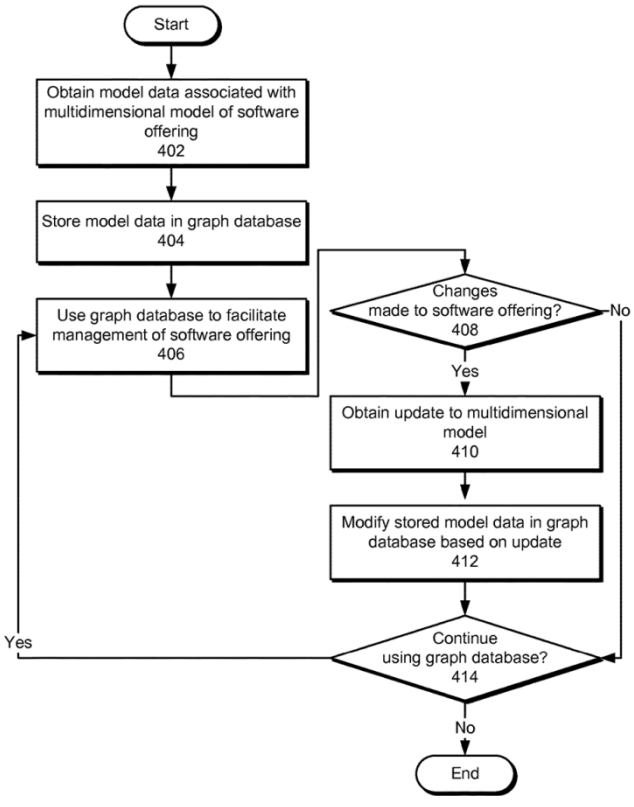

Fig. 6 the process flow used by [23] utilizing graph database.

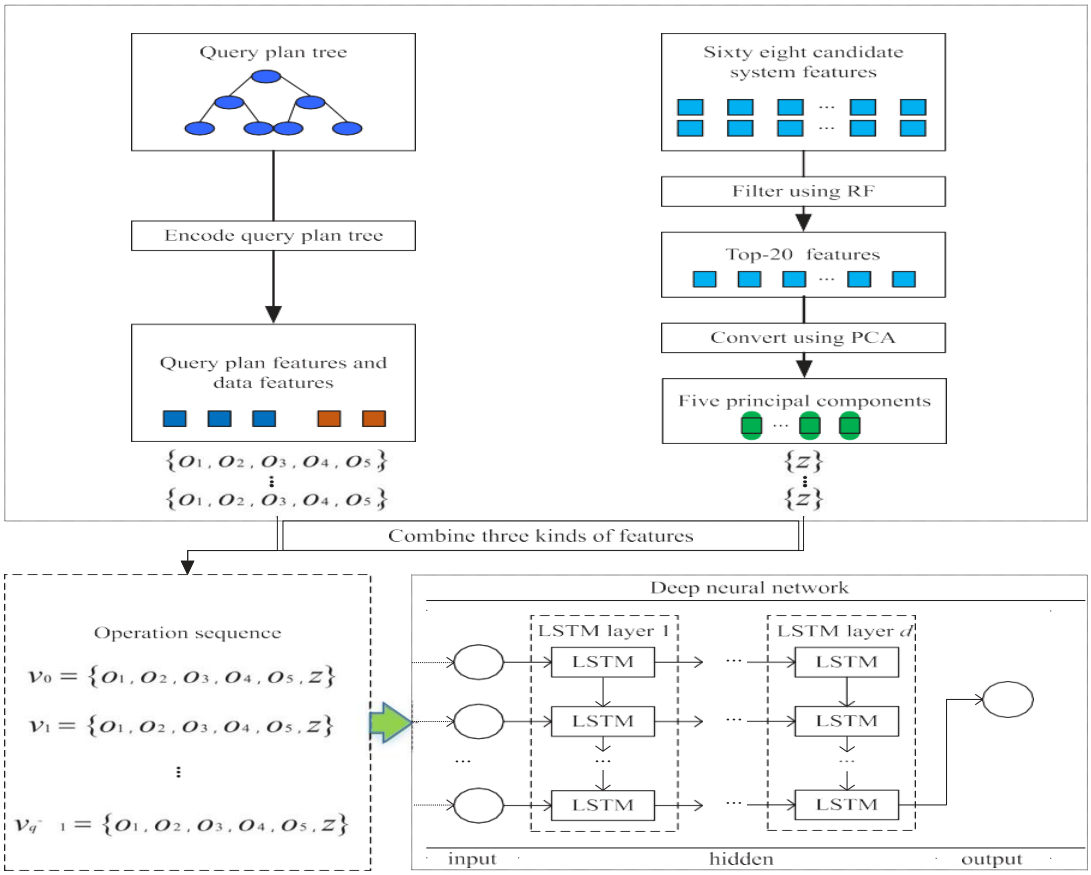

Fig. 7 the proposed method of [24].

The running time forecasting for query operation in graph database has gotten troublesome and a big issue because of query and system plan complexity. It is hard for DBMS and DBA to find the precise running time within and before the running of a query operation. Before running the query operation, forecasting its running time may assist the DBMS or DBA effectively management in the load management fields, progress monitoring, permission control, and task scheduling. So, precisely and proficiently forecasting the running time for query operation is a critical innovation in these fields.

Due to this challenge, Zheng Chu et al. [24] presented a research inspired by the techniques of AI and graph database. They proposed a deep learning technique to forecast the running time for query operation in graph database. First, every query task is encoded into query arrangement. Second, random forest algorithm is used as a feature selection algorithm to select top- 20 attributes from 68 attributes of the system and principle component analysis is utilized to reduce 20 attributes to 5 attributes. Finally, an exact and proficient model dependent on the long short term memory is utilized to forecast the running time. The method can forecast the running time before running a query operation in graph database [24] as shown in Fig. 7.

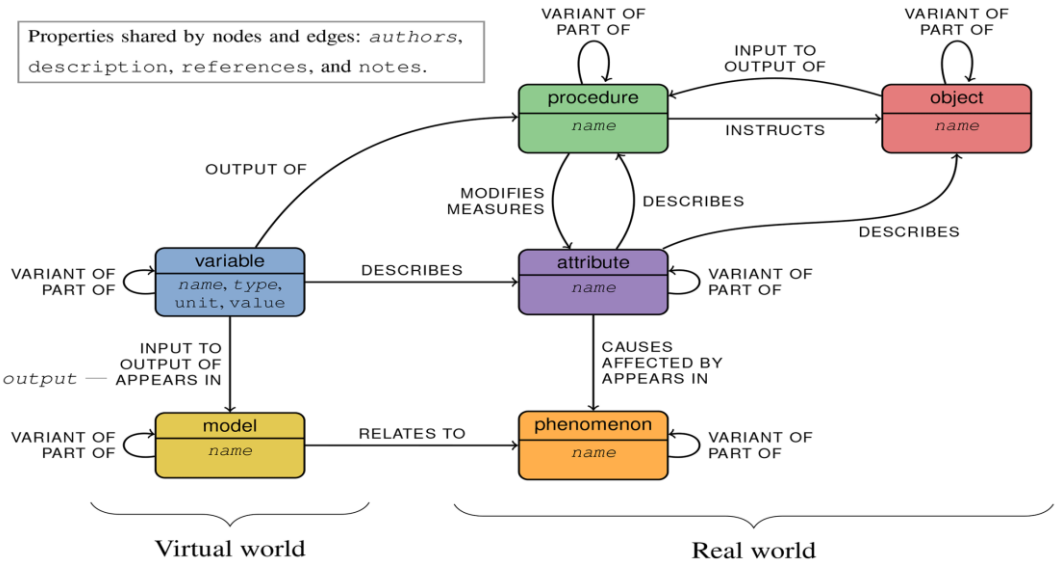

Fig. 8 Overview of database schema.

doi: 10.21608/EJAI.2022.132961.1002

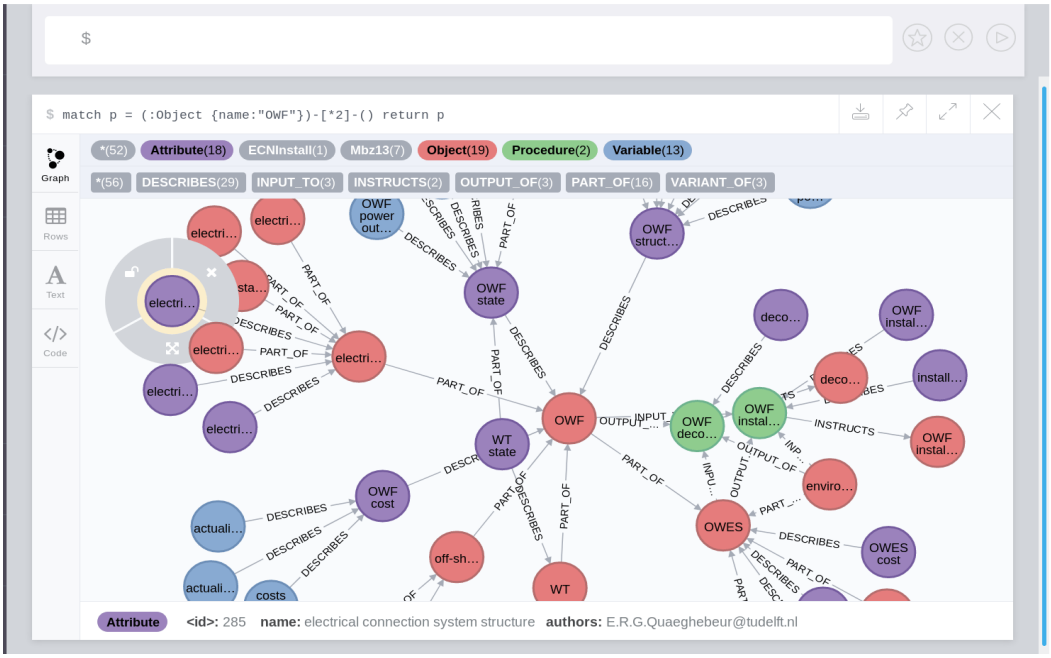

Fig 9 the Neo4J interface

The development and construction of wind farm include numerous controls. It is difficult for a single developer or engineer to keep an overview of all concepts, tools, and models. However, this is required when performing incorporated analysis or modeling. To assist researchers keep this outline, the Wind Energy System graph(WES graph) is made by Erik et al. [25], , a knowledge base for the wind farm domain, implemented as a graph database. It at present contains 1222 concepts and 1725 relations between them. Their research proposed the architecture of this graph database such nodes is used as storages for content and the relation between them which classifies the concepts. Their research additionally examines various by and large troublesome cases that find when adding content to such a knowledge base. It also discusses the expected uses of WESgraph and shows its utilization for calculation pathway discovery as illustrated in Fig. 8 and Fig. 9.

In medical services applications, building up a data model for storing the relationship between patient and doctor is significant. Despite the relational models are well known and common for some business and commercial applications, they may not be suitable for demonstrating the relationship between patient and doctor because of their inherent irregular nature and complexities. Mondal et al. [26] proposed a case study as a recommendation system between patient and doctor. This proposed system is based on top of multilayer graph database. The modern research manuscripts have already demonstrated that multilayer graph model can be effectively utilized in numerous applications where huge, heterogeneous data are to be modeled. As a feature of the recommendation system, their research additionally presents an idea of confidence which is one significant element of any kind of recommendation. The confidence factor presented in their research exploited certain attributes of the multilayer graph database and it also introduced some examination to show the effectiveness of the graph database compared with relational database [26] as shown in Fig. 10.

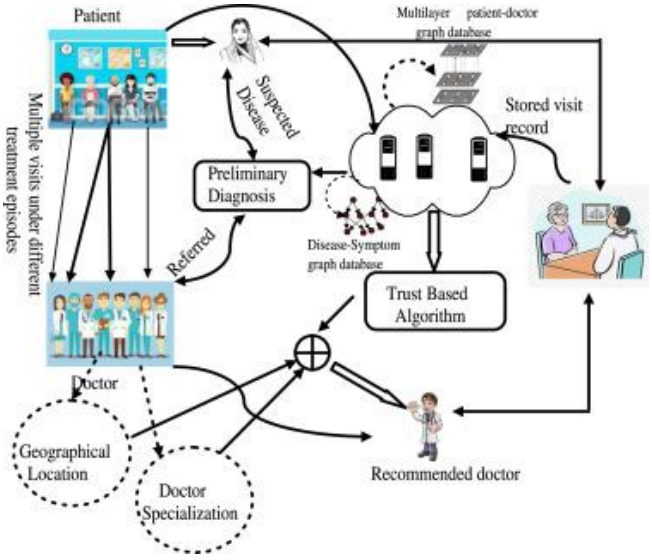

Fig. 10 the recommendation system.

4. Conclusions and future work

In the current era, the state of art becomes more interesting for researchers. NoSQL database has gotten a lot of consideration from the business and industry because of its ability to sore a large amount of data, effectiveness, flexibility, and scalability. No SQL database may stop the conventional relational database in the future, but it isn't since a long time ago NoSQL is born. Rather than filling in for the conventional relational database altogether, NoSQL database is just appropriate for certain applications. In some sense, it is only an enhancement to the conventional database. The requirements of any system should be

doi: 10.21608/EJAI.2022.132961.1002

specified, and then the designers or developers should specify the type of database used. This survey initially examines the need of NoSQL, totally studies the improvement of NoSQL database, and discusses the CAP and BASE theories, the NoSQL database categories such as key-value, document, column family, and graph databases. Graph database is one the most significant database in No SQL that is discussed in details in this survey in terms the definition of graph database, graph database structure, graph database advantages, use Cases and criteria for choosing graph databases, and the latest researches based on graph database.

References

[1] C. He, "Survey on NoSQL database technology," J. Appl. Sci. Eng. Innov., vol. 2, no. 2, 2015.

[2] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," IDC iView IDC Anal. Futur., vol. 2007, no. 2012, pp. 1–16, 2012.

[3] P. J. Sadalage and M. Fowler, NoSQL distilled: a brief guide to the emerging world of polyglot persistence. Pearson Education, 2013.

[4] H. Yadava, The Berkeley DB Book. Apress, 2007.

[5] S. Vijaykumar and S. G. Saravanakumar, "Implementation of NOSQL for robotics," in INTERACT-2010, 2010, pp. 195–200.

[6] F. Chang et al., "Bigtable: A distributed storage system for structured data," ACM Trans. Comput. Syst., vol. 26, no. 2, pp. 1–26, 2008.

[7] J. Jose et al., "Memcached design on high performance rdma capable interconnects," in 2011 International Conference on Parallel Processing, 2011, pp. 743–752.

[8] M. Y. Becker and P. Sewell, "Cassandra: Flexible trust management, applied to electronic health records," in Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004., 2004, pp. 139–154.

[9] S. Simon, "Brewer's cap theorem," CS341 Distrib. Inf. Syst. Univ. Basel, 2000.

[10] J. L. Carlson, Redis in action. Manning Publications Co., 2013.

[11] 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.

[12] T. Fan, L. Yan, and Z. Ma, "Storing and querying fuzzy RDF (S) in HBase databases," Int. J. Intell. Syst., vol. 35, no. 4, pp. 751–780, 2020.

[13] P. Tu et al., "Effect of Doping Metal Oxide in ZnO/SBA-15 on Its Acid-Base Properties and Performance in Ethanol-to-Butadiene Process," ChemistrySelect, vol. 5, no. 24, pp. 7258–7266, 2020.

[14] X. Han, H. An, X. Zhao, and Y. Wang, "Influence of acid-base properties on the catalytic performance of Ni/hydroxyapatite in n-butanol Guerbet condensation," Catal. Commun., vol. 146, p. 106130, 2020.

[15] D. Sullivan, NoSQL for mere mortals. Addison-Wesley Professional, 2015.

[16] W. Vogels, "Eventually consistent," Queue, vol. 6, no. 6, pp. 14–19, 2008.

[17] A. Bhattacharyya and D. Chakravarty, "Graph Database: A Survey," in 2020 International Conference on Computer, Electrical & Communication Engineering (ICCECE), 2020, pp. 1–8.

[18] R. Angles and C. Gutierrez, "Survey of graph database models," ACM Comput. Surv., vol. 40, no. 1, pp. 1–39, 2008.

[19] R. Angles, "A comparison of current graph database models," in 2012 IEEE 28th International Conference on Data Engineering Workshops, 2012, pp. 171–177.

[20] J. Paredaens, P. Peelman, and L. Tanca, "G-Log: A graph-based query language," IEEE Trans. Knowl. Data Eng., vol. 7, no. 3, pp. 436–453, 1995.

[21] G. Klyne, "Resource description framework (RDF): Concepts and abstract syntax," http//www. w3. org/TR/2004/REC-rdf-concepts-20040210/, 2004.

[22] T. Lose, P. van Heusden, and A. Christoffels, "COMBAT-TB-NeoDB: fostering tuberculosis research through integrative analysis using graph database technologies," Bioinformatics, vol. 36, no. 3, pp. 982–983, 2020.

[23] J. Labat, R. Venkataraman, J. E. Edward, and R. Varadharajan, "Graph databases for storing multidimensional models of software offerings." Google Patents, 07-Jan-2020.

[24] Z. Chu, J. Yu, and A. Hamdulla, "A novel deep learning method for query task execution time prediction in graph database," Futur. Gener. Comput. Syst., 2020.

[25] E. Quaeghebeur, S. Sanchez Perez-Moreno, and M. B. Zaaijer, "WESgraph: a graph database for the wind farm domain," Wind Energy Sci., vol. 5, no. 1, pp. 259–284, 2020.

[26] S. Mondal, A. Basu, and N. Mukherjee, "Building a trust-based doctor recommendation system on top of multilayer graph database," J. Biomed. Inform., vol. 110, p. 103549, 2020.

doi: 10.21608/EJAI.2022.132961.1002