**New Trends on Database Management Systems**

**Introduction**

The field of database management systems is experiencing revolutionary changes due to the trends and insights taking place in the field. Today, the concept of a database management system has encountered improvements in a bid to enhance the efficiency and effectiveness of data storage, processing, and sharing. One of the recent trends in the field of database management is the use of graphs database and NoSQL as the data language to code and decode the data and information stored in the database. This paper will examine the recent development in a database management system with a focus on how data language is changing the field of database management.

**Data Language (NoSQL, Graph Database)**

The world of computing is always characterized by new developments. One of the developments that have revolutionized the industry is the introduction and development of NoSQL(not only SQL), which is an approach in the design of a database that offers flexible schemas for data storage and retrieval beyond the table structures that characterize the traditional relational databases (Daniel et al., 2016). It should be noted that NoSQL databases were not introduced just recently; they have been in existence for many years. However, it only became popular with recent developments in database management such as the cloud era, mobile applications, big data as well as high-volume web.

Today, NoSQL databases are chosen by many companies for their distinguishing features such as performance, scale as well as the ease through which they can be used. Some of the well-known types of NoSQL databases include document, graph, key-value, and column databases. It should be noted that the "No" in the NoSQL stands for "not only," and this implies that apart from the attribute of NoSQL databases to support SQL queries, the two are also applicable together when it comes to dealing with micro-services as well as polyglot persistence (Daniel et al., 2016).

**NoSQL against SQL**

NoSQL databases are unique in that they are not subject to all the rules followed by relational databases. In particular, the NoSQL databases don’t employ the use of traditional row/ column/ table design of the database. Again, it does apply the structured query language when doing a data query. SQL and other relational databases operate efficiently for large servers as well as storage mediums (Comyn-Wattiau & Akoka, 2017). However, currently, the world is characterized by data sets that are frequently changing. Activities such as e-commerce applications provided programmers with a challenge of a more flexible database to replace the SQL, and this is what has brought about the emergence and development of NoSQL databases.

NoSQL databases are developed for specific models of data and are characterized by flexible schemas that give the programmers permission for the creation and management of modern applications. Additionally, NoSQL databases are considered more agile as they are not constructed on the principles of tables and do not also employ the use of SQL when it comes to data manipulation or analysis (Comyn-Wattiau & Akoka, 2017). Further, NoSQL databases cover structure, unstructured, semi-structured as well as polymorphic data. Besides, it also gives the user more freedom for agility, flexibility, and quicker iteration. It makes it possible to have a simpler design, improved scalability as well as enhanced control over availability. As highlighted earlier, there are various types of NoSQL databases. These include;

**Key-Value Store**

In this type of structure, the key is always characters in the form of a simple string, and the value can be described as a group of uninterrupted bytes, which are opaque to the database. In most cases, the dataset itself is usually data type, which is primitive, in the form of integer, array, or string. Similarly, it can be a complex material that requires the application to access directly (Comyn-Wattiau & Akoka, 2017). This form of NoSQL database replaces relational ones as they portray more flexible models of data through which developers can modify the fields and objects easily with the evolution of the applications. Generally, key-value stores are not characterized by any query language. They are currently adopted by many organizations because they provide a way of data storage, retrieval, and update through the simple GET, PUT as well as DELETE commands (de Sousa & Cura, 2018). As a result, they are considered easy in usability, fast, portable, scalable, and flexible.

**Document-Based**

A document is considered an object and keys or strings which values whose types can be recognized, such as Booleans, numbers, and strings. They also include nested dictionaries and nested arrays. The main purpose of designing document databases is for flexibility. In their design, they are not primarily subjected to have schemas and thus can be easily modified (de Sousa & Cura, 2018). For instance, in the event, an application requires the capability of storing varying attributes with specific amounts of data, then the best option that people currently use is the document database.

**Column-based**

Column-based NoSQL databases or wide column are those models which make it possible to quickly access data through the use of row key, name of the column as well as a timestamp. The flexible schema of Column-based NoSQL databases implies that all the columns in all the records don’t have to be consistent (Castelltort & Laurent, 2016). A column can be added to specific rows without the user being forced to do the same to every single record.

**Graph-based NoSQL databases**

Graph database or what others call NoSQL graph refers to a data management tendency that is designed to handle very large sets of unstructured, structured, and semi-structured data. In this regard, this type of database is useful to organizations since it enhances accessibility, integration, and analysis of data derived from some sources. In this regard, this helps many companies’ big data and social media data analytics.

**NoSQL Graph Database against Relational Database**

The periods of the 1970s saw the development of traditional models to data management, the relational database. This was primarily introduced to help companies or organizations in issues of storage of structured data. Progressively, the relational database requires the use of schema, which is the definition in which the organization of data occurs as well as how the relations are associated, to have definitions before any new information is added. However, today's changes in the dynamics of the world are characterized by social media data, mobile data, and the Internet of Things (loT) data, which has dominated the business environment. Additionally, companies face issues of big unstructured data, which piles up every minute as time goes by.

Through the use of a graph database, it is possible to handle such complex data sets. Further, other than handling huge amounts of data of different kinds, the NoSQL graph database is technically designed so that it does not require a redefinition of its schema before adding any new data. These features and capabilities thus comparatively enhance NoSQL graph database when it comes to flexibility, dynamicity as well as reduction of costs when it comes to integrating new data sources as compared to the old relational databases. In comparison to handling moderate velocity of data from a single or fewer location of the relational databases, NoSQL graph databases can relatively be able to store, integrate, retrieve, and analyze data at high velocity originating from some locations.

**Semantically Rich NoSQL Database**

The semantic graph database can be described as a category of NoSQL graph database with the capability of integrating data, which is heterogeneous and originating from several sources and constructing links between datasets. The semantic graph database, which is also called the RDF triplestore, concentrates on the association between entities and has the capability of deriving new knowledge from the already existing information. A semantic graph database is a powerful tool that can be used in the relationship-oriented analytics as well as the discovery of knowledge.

Additionally, the technical capability to handle huge amounts of datasets and the fact that it does not rely on the redefinition of schemas before any addition of new information are among the features that make NoSQL semantic graph database be employed when it comes to the analytics of real-time big data. In this regard, there is always the need for the addition of schemas defined before adding new information in relational databases. This aspect restricts the integration of data that is originating from the news sources as it requires that the entire schema is freshly changed (Holzschuher & Peinl, 2016). On the other hand, through the use of the NoSQL semantic graph database, there is no need to every time change the schema the event a new data source is to be added. This ensures that the integration of data by enterprises is at low cost and requires relatively less effort.

In this regard, among all other graph databases, the semantic graph database is considered one of the best if not the best because of its technical capability to besides support rich semantic data schema, also known as ontologies. The NoSQL semantic graph database is considered advantageous on both ends. On one hand, it enhances the flexibility of data as it does not rely on the schema (Holzschuher & Peinl, 2016). On the other hand, through the ontologies, the NoSQL semantic graph database is provided with the freedom and ability to come up with logical models based on any patterns the companies consider useful when it comes to their applications, with no restriction of having to alter data.

**The Advantages of NoSQL Semantic Graph Database**

Other than the rich semantic models, NoSQL semantic graph database employs the use of globally developed W3C standards when it comes to representation of data on the Web (Parmar & Roy, 2018). As a result, the employment of the use of standard procedures and practices ensure integration exchange and mapping of data to other datasets is easier and minimizes the risk of vendor lock-in as one is in the process of working with a graph database.

A specific scenario of the above-mentioned standards is the Uniform Resource Identifier (URI), which is some kind of unique ID for all the linked materials so that individuals can differentiate between them or get to know that one of the elements in the is similar to another in a different database (Parmar & Roy, 2018). It is important to note that the employment of the URIs does not only minimize costs in integrating databases from sources considered desperate but, at the same time, makes publishing and sharing of data within mapping to open or linked data.

Ontotext’s graph database can employ the use of inference, which implies inferring new links from the original explicit statements in the RDF triplestore. Through the inference, the graph database is enriched through the creation of new knowledge. This improves the organization's ability to observe all their data, which is highly interlinked (Castelltort & Laurent, 2017). This, therefore, equips organizations with detailed and clear information from which they can make sound decisions.

**Practical Areas Where NoSQL is Applicable**

Relational databases have existed for decades, and technology has been on rapid change. Relational databases employ the use of SQL to carry out certain activities such as data updating in the database or data retrieval from the database. It is expensive to maintain high-end relational database management systems as they need things like purchasing a license, trained personnel to manage them as well as very powerful hardware to keep them running (Fabregat et al., 2018). NoSQL databases enable faster, enhanced agility in storage and processing, implying that NoSQL databases are the best option for modern and complex applications such as mobile applications or e-commerce.

NoSQL databases’ horizontal scaling and flexibility imply that they can handle huge volumes of data, rendering them most effective for agile development, frequent code pushes, and quick iterations. Generally, the main distinction between relational and NoSQL databases comes in terms of availability, performance, as well as scalability (Fabregat et al., 2018). In this regard, currently, there are specific cases where NoSQL databases are preferred to relational databases.

First, when there is a need to store huge amounts of unstructured data characterized by changing schemas. This is because NoSQL databases have properties such as horizontal scaling that enables them to store and carry out the processing of huge data volumes. Similarly, NoSQL databases make it possible to have ad-hoc changes in schemas. Secondly, when using cloud computing and storage, NoSQL databases are the best option currently used by many (Johnpaul & Mathew, 2017). This is because NoSQL databases are designed in a way that they can be scaled across multiple centers of data while at the same time run as distributed systems. This attribute enables the NoSQL databases to be advantageous when it comes to cloud computing.

Third, in the event, the user wants to develop rapidly. In most cases, NoSQL is the best choice to store agile methods of developing software, especially when it requires sprint cycles (Castelltort & Laurent, 2016). In the case of NoSQL databases, data doesn’t have to be prepared like it is done in relational databases, and rather than mitigating the structured data every time there is a change in the application, NoSQL databases have dynamic schemas which change and develop with the application (Castelltort & Laurent, 2016). Lastly, when there is more sense in using a hybrid data environment. NoSQL means not only SQL, and this implies that when required, it can be used as a complement to the relational databases and, in this scenario, provide the system with the flexibility of choosing the most appropriate tool or method of carrying out the task.

**Conclusion**

NoSQL data language is one of the recent trends and insights that are changing the concept of a database management system. The use of NoSQL in database management has increased the efficiency and efficiency of data storage, processing, accessibility, gathering, and sharing through the increased interference and interoperability of different databases. It is also referred to as the graph database because it provides graphs that can be used to visualize the data and information that is stored in the database. The use of this concept has increased the speed of processing, compatibility of data and information, comparability of data firm different databases, and speed of retrieving and processing data. The accuracy, integrity, availability, and credibility of data stored in these databases have also increased with the application of this concept of database.

**References**

Castelltort, A., & Laurent, A. (2016). Extracting fuzzy summaries from NoSQL graph databases. In *Flexible Query Answering Systems 2015* (pp. 189-200). Springer, Cham.

Castelltort, A., & Laurent, A. (2017). Exploiting NoSQL graph databases and in-memory architectures for extracting graph structural data summaries. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *25*(01), 81-109.

Comyn-Wattiau, I., & Akoka, J. (2017, December). Model drove reverse engineering of NoSQL property graph databases: The case of Neo4j. In *2017 IEEE International Conference on Big Data (Big Data)* (pp. 453-458). IEEE.

Daniel, G., Sunyé, G., & Cabot, J. (2016, November). UMLtoGraphDB: mapping conceptual schemas to graph databases. In *International Conference on Conceptual Modeling* (pp. 430-444). Springer, Cham.

de Sousa, V. M., & Cura, L. M. D. V. (2018, November). Logical design of graph databases from an entity-relationship conceptual model. In *Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services* (pp. 183-189).

Fabregat, A., Korninger, F., Viteri, G., Sidiropoulos, K., Marin-Garcia, P., Ping, P., ... & Hermjakob, H. (2018). Reactome graph database: Efficient access to complex pathway data. *PLoS computational biology*, *14*(1), e1005968.

Holzschuher, F., & Peinl, R. (2016). Querying a graph database–language selection and performance considerations. *Journal of Computer and System Sciences*, *82*(1), 45-68.

Johnpaul, C. I., & Mathew, T. (2017, January). A Cypher query-based NoSQL data mining on protein datasets using the Neo4j graph database. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 1-6). IEEE.

Parmar, R. R., & Roy, S. (2018). MongoDB as an Efficient Graph Database: An Application of Document Oriented NoSQL Database. *Data-Intensive Computing Applications for Big Data*, *29*, 331.