European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.23-31, February 2020
Published by ***ECRTD- UK***
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

AN IMPROVED LAND MAPPING AND GEOGRAPHICAL INFORMATION MANAGEMENT SYSTEM USING GEODATABASE

Chidi E. Egwu and Edward E. Ogheneovo*
Department of Computer Science
, University of Port Harcourt,
Port Harcourt, Nigeria

ABSTRACT: *With data constantly increasing at a tremendous speed, it is crucial to have better knowledge of how information is manipulated and stored for subsequent retrieval and use. The data storage geodatabase strategy is introduced as dependable alternative based on acknowledged relational database concepts, which form foundation of selected database handling system. Simple but well-defined tables composed of distinctive features selected to store and handle spatial data and rule-base for every topographical dataset. In this paper, we developed an improved land mapping and Geographical Information System (GIS) using geodatabase. The study provides an enhanced approach to storing and managing data using a geodatabase, contributing to further research into alternative way of handling big data. Development of a web application that interacts with geodatabase for big data storage without the need of running multiple servers or enterprise class software. Thus this research is useful for those who have need for efficient data storage and management in today's world of data size and complexities. By storing data within a geodatabase, one can draw from the benefits that come with its data management capabilities to leverage spatial information.*

KEYWORDS: *Big Data, RDBMS, ORDBMS, Geodatabase, NoSQL, SQL, PostgreSQL*

INTRODUCTION

Rapid computing, simple access to internet-based devices, steady growth in mobile networks, cloud-based infrastructure and new technologies have created an incomprehensibly big data world, described as "Big Data" [1] [2] [3]. Large-scale data typically deemed to have grown so large in volume as to consist mainly of uncommon "structured, semi-structured and unstructured" information being constantly updated in real time, making it difficult to manage or use standard data management instruments efficiently, e.g., classic RDBMS. In the easiest form, "Big Data" is employed in defining information whose "semi-structured and unstructured" size and structure is larger than information deemed appropriate for periodic database handling scheme for storage or processing. The information is not organized to suit periodic architectures of database [4]. The massive increment in data offers likelihood for great scientific innovations, better business models idea and better procedure for handling banking sector, food production and surroundings in general [5] [6]. "Big data" covers broad variety of structured, semi-structured and unstructured data. Geodatabases presents substitute for storing GIS data as single file, which is able to hold polygons, polyline layers or multiple points at same time [7] [8].

Geodatabase comprises of multiple platform-independent components of software that provides services to support GIS. The key reason for geodatabases, is to allow usage of RDBMS for spatial data [9]. A multiuser geodatabase utilizes a multitier architecture that supports advanced implementation logic and behaviors in applications stages, (e.g., Aeronautical Reconnaissance Coverage Geographical Information System (ArcGIS) software) on storage tier, a simple supported RDBMS [10] [11]. The responsibility for managing GIS information when working with multiuser geodatabase is a collective task, handled by ArcGIS and the RDBMS software selected. A relational database offers a direct formal structure for storing and managing data in tables. Data storage and retrieval are handled via simple tables. The multiuser geodatabase utilizes power of RDBMS. Certain characteristics of topographical information organization, For example, disk storage, attribute type definition, query processing and multi-user transaction processing are reserved for RDBMS to handle [12] [13].

All the different elements of GIS data are supported by geodatabase, offering easy implementation, and inbuilt data management capabilities to leverage spatial information on all geodatabase stored data. Geodatabase offers robust and extendable data model when compared to Relational database, shape files and coverages [14]. Deciding on the best approach to store date remains a topic of importance, it remains a difficult problem to address as data state and nature is constantly changing. This is merely a consequence of increasing the number of Internet-based users. Companies suddenly need to collect data appropriate to their company in order to optimize productivity and ROI (Return on Investment). Due to these limitations, developers of application that are required to process big data have turned to an emerging database approach called Geodatabase for big data storage and management.

RELATED WORK

The “Microsoft Access database”, or multi-use relationship DB Multi-User (such as “Oracle, Microsoft SQL Server, PostgreSQL,” or IBM D B2) database is collection of numerous geodatabases set. Geodatabases are available to many users in various dimensions and can range from tiny, user-based, file built-in databases to large-scale geodatabases for departments, companies and many users [15]. However, geodatabase is more than set of datasets. The geodatabase is key information format used in editing and information management for the ArcGIS indigenous information structure. While Arc-GIS works with geographical information in many GIS formats, ArcGIS works with capacity of geodatabase and leverages them [16]. It is used mainly by the database management scheme (DBMS) or file system to physical store geographical information. You can access and operate through ArcGIS or a SQL-enabled database management system with this physical example of your data collection.

Geodatabases have an extensive model of data, for geographical representation and management. Geodatabases have a model transaction to manage GIS information workflows.

Geodatabases are on basis models specified to storing interactive database concepts while utilizing underlying database scheme. For every geographic dataset, simple tables and defined kind of attributes employed for storing schema, rule and spatial features data [17]. This strategy offers formal model for your data storage and collaboration. Through this technique SQL are employed to generate, alter and query tables and information components as sequence for interactive features and operators. Accessing contents of this table is possible, including shape when stored as a spatial SQL type [18].

The geodatabase in practice makes use of same architecture of multitier software found in other sophisticated DBMS applications; its implementation is common and is equally called Object interactive model. The objects in a geodatabase is made up rows like regular relational databases, but identity and set behaviors are supplied by application logic of the geodatabase. The core of every geodatabase is collection of standard tables, indexes and similar objects of relational databases. Schema are persisted in geodatabase scheme collection tables in DBMS which defines integrity and features of geographic information. These geo-tables are kept on either disk files or without enterprise-based databases like PostgreSQL or Oracle [19]. Well defined columns are employed for storage of spatial or raster data.

Brand [20] proposed a NoSQL approach to big data storage in HER. Both structured and unstructured data stored in MongoDB NoSQL database. This technique is very scalable but cannot handle complex transactions and joins. Transactions and Joins implementation are required at the application. Manson [21] proposed a NoSQL approach for storing big data. NoSQL offers scalability improvements compared to interactive databases. It attempts retain the scalability capability of NoSQL’s while maintain the ACID functionalities of relational data base technology. It uses sharing mechanism to distributed data across servers in a cluster. Therefore, some limitations imposed by centralized storage in conventional database systems are eliminated. The model architecture comprises of model in administrative, transaction and storage tier. The model aimed at developing model that provides functionality of both relational database and NoSQL. However, NoSQL cannot scale farther when data is beyond a few terabytes.

Churnithipaisan et al. [22] identified preference reliable framework comparatively firmer than Basically Available, Soft state, Eventual consistency systems by extending Brewer’s theory on Consistency Availability and Partitioning in distributed environ using best proportions. They also proposed a procedure based on time-stamping that imposes prompt uniformity along with horizontal scalability and resilience. This framework is however weaker than ACID framework. Alsubaiee et al. [23] proposed a framework anatomy for predominant wide volume data stores to categorize familiar frameworks and compare potentials for scaling. They investigated and analyzed scalable systems intensively and came up with exhaustive classification scheme for diverse crucial prospects surrounding framework and system model. The prevalent and vastly expandable systems were connected to scheme of classification to categorize the popular technique to provide footing for monitoring the extent which such system can grow to adapt.

METHODOLOGY

We proposed the introduction of Geodatabase management system to store structured and unstructured data. The core of the proposed system makes use of a Geodatabase system (PostgreSQL and POSTGIS) to store series of information from aeronautical platforms "structured and unstructured data". Geodatabase is an essential storage system which includes the schema and rule base for each geographic dataset plus simple, tabular storage of the spatial and attributes data. Tables are used to store all three primary datasets in the geodatabase (feature classes, tables of attributes, and raster datasets) and other geodatabase elements. The spatial depictions in the geographic datasets are stored as either vector features or rasters. These geometries are stored and managed in attribute columns along with the usual tabular attribute fields.

The proposed system design involves going through the architectural design process and the modules / components design process. It also involves the design of the user interface and the logic of the backend that will enable the system meet the required task. This describes the proposed system, explaining in details each module and component that work together to achieve a working application. The system is designed to meet the need for an improved land mapping and GIS management system using geodatabase, a database system capable of handling structured, unstructured and semi-structured data without little or none of the trade-offs that comes with using RDBMS or NSQL database alone. The design also captures the major components that serve as building block needed to understand the workings of the proposed system.

European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.23-31, February 2020
Published by *ECRTD- UK*
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

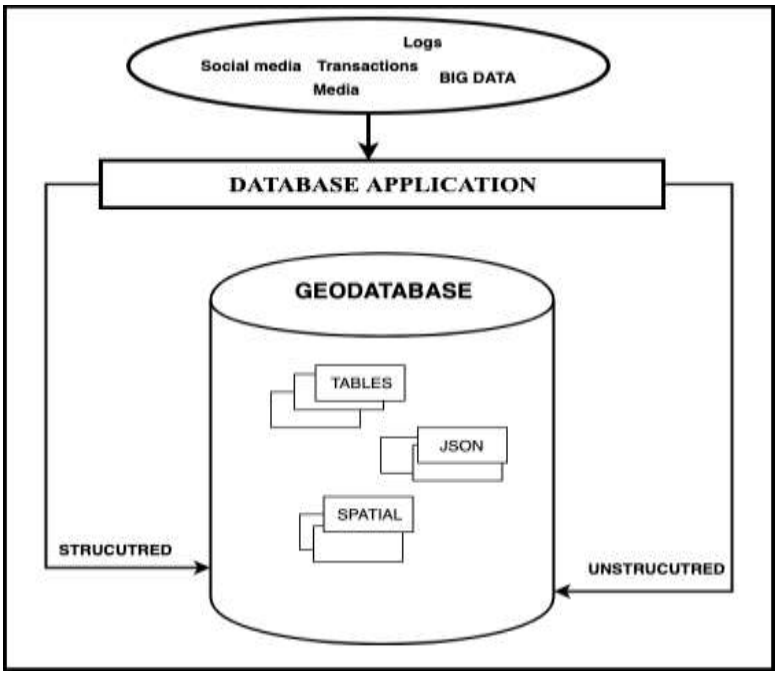

Fig. 1: Architecture of geodatabase system

RESULTS AND DISCUSSION

Multiple tests were carried out, with the best results recorded in the tables below, the Improved Land and GIS Management system was tested against MySQL in Table 1 below, tested against MongoDB in Table 2, before finally getting compared to the existing system in Table 3.

Table 1: Experiment Data Result 1 for MySQL and the Geodatabase Application

TIME OF DATA INTERACTION	MySQL (seconds)	PostgreSQL (seconds)
Structured data (Single Read) 1000	0.42	0.79
Unstructured data (Single Read) 1000	Not Applicable	15.32
Structured data (Multiple Read) 2000	14	9.3
Unstructured data (Multiple Read) 2000	Not Applicable	21.44
Structured data (Single Write) 1000	7.42	10.79
Unstructured data (Single Write) 1000	Not Applicable	15.32
Structured data (Multiple Write) 2000	33	24.2
Unstructured data (Multiple Write) 2000	Not Applicable	27.44

European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.23-31, February 2020
Published by ***ECRTD- UK***
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

Table 2: Experiment Data Result 2 for MongoDB and the Geodatabase Application

TIME OF DATA INTERACTION	MongoDB (seconds)	PostgreSQL (seconds)
Structured data (Single Read) 1000	2.2	1.32
Unstructured data (Single Read) 1000	6.1	9.7
Structured data (Multiple Read) 2000	22.54	23
Unstructured data (Multiple Read) 2000	15.3	18.83
Structured data (Single Write)	10.42	9.79
Unstructured data (Single Write)	19.2	15.38
Structured data (Multiple Write) 100	27.3	24.2sec
Unstructured data (Multiple Write) 100	27.22	27.44

Table 3: Experiment Data Result 3 for the Hybrid (MySQL and MongoDB) and the Geodatabase application

TIME OF DATA INTERACTION	Existing System (seconds)	PostgreSQL (seconds)
Structured data (Single Read) 1000	2.7	1.42
Unstructured data (Single Read) 1000	11.4	10.7
Structured data (Multiple Read) 2000	22.54	20.2
Unstructured data (Multiple Read) 2000	19.1	18.83
Structured data (Single Write)	10.42	9.79
Unstructured data (Single Write)	19.2	15.38
Structured data (Multiple Write) 100	32.3	22.2
Unstructured data (Multiple Write) 100	29.44	29.22

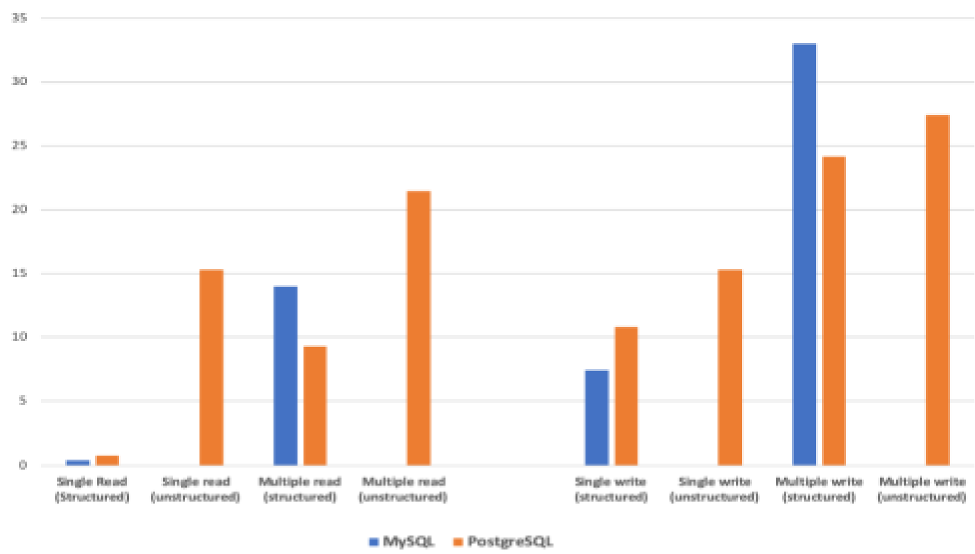

Fig. 6: Comparison between MySQL and the PostgreSQL

European Journal of Computer Science and Information Technology
Vol.8, No.1, pp.23-31, February 2020
Published by *ECRTD- UK*
Print ISSN: 2054-0957 (Print), Online ISSN: 2054-0965 (Online)

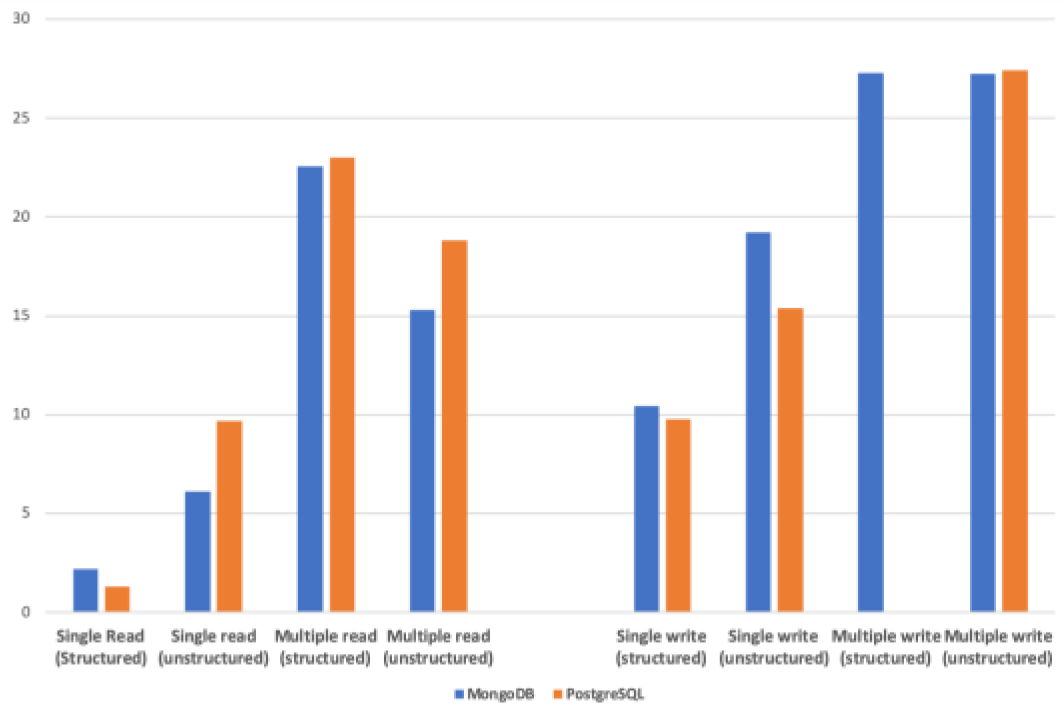

Fig. 7: Comparison between MongoDB and PostgreSQL

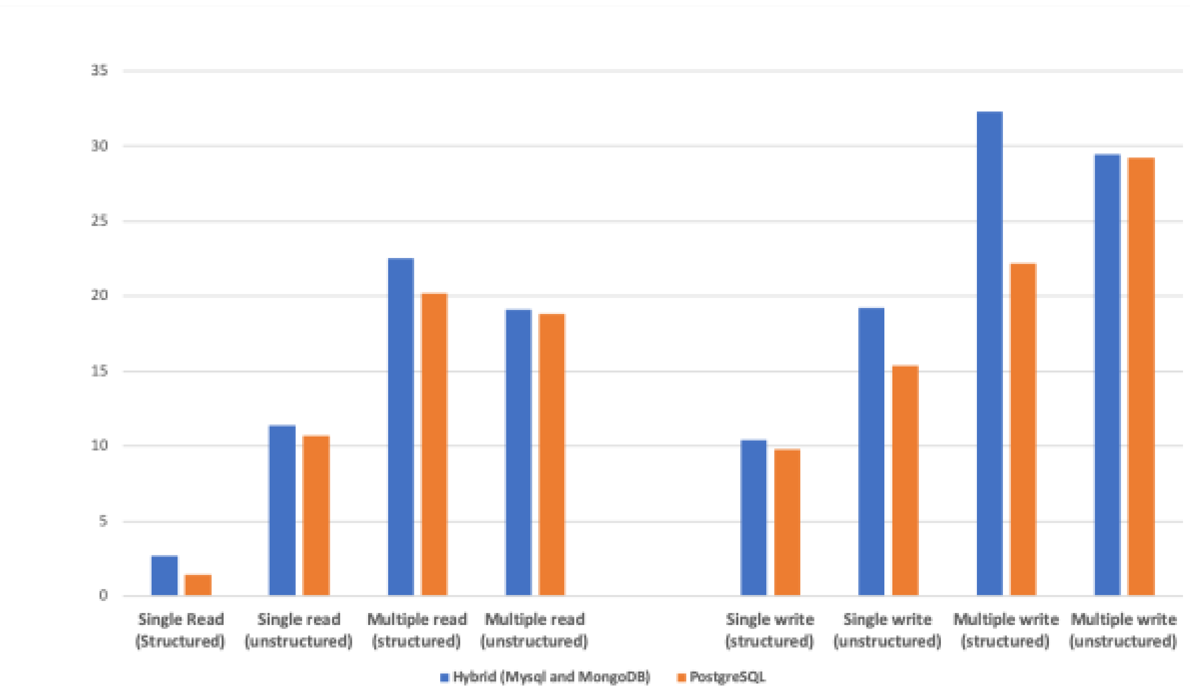

Fig. 8: Comparison between MongoDB and PostgreSQL

on-structured data. From the results it is possible to observe that speed at which data is being read and written on both the existing system and new system might be consider negligible to non-enterprise user. The testing of the Geodatabase application is further visualized using histograms and from the Figure 6, Figure 7 and Figure 8, While the read and write speed difference is negligible to most regular users, a few microseconds can serve as the deciding factor for enterprise industries interacting with big data.

CONCLUSION

In conclusion, we introduced a method that combines the capabilities of PostgreSQL (ORDBMS) which belongs to the relational group of database systems, JSONB (a NoSQL derivative for PostgreSQL) and PostGIS (spatial derivative of PostgreSQL) being a NoSQL database, to efficiently store and manage big data which are both structured, unstructured or semi-structured data. By storing data within a geodatabase, you can draw from the benefits that come with its data management capabilities to leverage spatial information. The work presents the following contributions to knowledge. The study discusses the option of achieving NoSQL database flexibility and scalability as well as the stability and transactional components of a relational database, while maintaining a single database management system. The study provides an enhanced approach to storing and managing data using a geodatabase, contributing to further research into alternative way of handling big data. Development of a web application that interacts with geodatabase for big data storage without the need of running multiple servers or enterprise class software. Thus this research is useful for those who have need for efficient data storage and management in today's world of data size and complexities.

References

1. Garside, J. and S. Cox (2013). Concise Process Improvement Definition with Case Studies. International Journal of Science Engineering and Research (IJSER), Vol. 2, No. 1, pp. 67-78.
2. Kumar R., C. Shilpi, B., Somya, (2015). Effective Way to Handling Big Data Problems Using NoSQL Database (MongoDB). Journal of Advanced Database Management & Systems, Vol. 2, No. 2, pp. 42–48.
3. Yuan and You (2014). Ensuring Consistency in Big Data Storage. Int'l Journal of Advanced Engineering and Global Technology, Vol. 2, No. 4, pp. 40-46.
4. Kang, J. and G. Kim, (2003). Query Type Classification for Web Document Retrieval. In Proceedings of the 27th Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval, 77-98
5. Kumar, R., N. H., Gupta, Charu, S. Jangir, (2014). Manage Big Data Through NoSQL. In Proceedings of the National Conference on Innovation in Wireless Communication and Networking Technology. Jaipur. doi: 10.13140/2.1.3965.3768.
6. Wu, J., P. Xinx, P., Joe. (2014). Anatomy of Big Data. International Journal of Advanced Engineering and Global Technology, Vol. 1, No. 2, pp. 44-55.
7. MacPherson, G. (2011). Student's Directory in Academic Work, Journal of Information Systems Education, Vol. 20, No. 1, pp. 60-67.
8. Burke, S. (2015). Time and Space: Geographical and Temporal Grounding of News Articles. Retrieved from Edinburgh Achieve: https:/www.era.lib.ed.ac.uk/handle/1842/6452. Retrieved 10/7/2019.

9. Alexander, K. (2008). Applications of Data Mining Techniques In Life Insurance. *International Journal of Data Mining & Knowledge Management Process* (IJDKP), Vol. 2, Vol. 4, pp. 87-98.
10. Morris, G. and A. Truskowski (2003). Database System Concepts, Implementations and Organizations: A Detailed Survey. International Journal of Scientific Engineering and Research (IJSER), Vol. 2, No. 2, pp. 45-56
11. James, B and P. O. Asagba, (2016): Hybrid Big Data Storage Management System Using Related Database Management System and Mongo DB (NoSqL), International Journal of Computer and Information Technology, Vol. 4, No. 4, pp. 111-123.
12. Sreekanth,R., G.V. Rao, and S. Nanduri, 2015). Big Data Electronic Health Records Data Management and Analysis on Cloud with MongoDB: A NoSQL Database. In Proceedings of the International Journal of Advanced Engineering and Global Technology, Vol. 3, No. 7, pp. 89-97.
13. Mordini E, and A. Rebera (2011). No Identification Without Representation: Constraints on the Use of Biometric Identification Systems. Review of Policy Research, Vol. 29, No. 1, pp. 5-20.
14. Ganed, H. and S. Cox (2013) Predicting Database Result with Defined Pattern. International Journal of Scientific Engineering and Research (IJSER), Vol. 2, No. 1, pp. 45-55.
15. Okpala, A. (2015). Using GIS to Model and Validate Small Scale Spatial Urban Air Polution Concentration. Retrieved from University of Edinburgh School of Geoscience: https://www.geos.ed.ac.uk/internal/databases/dms/public/abstract_report.htm?username:text=s1436706. Retrieved 10/7/2019.
16. Nowe, A. (2012). Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design (2nd ed.). New York: John Wiley & Sons. 13,6 :377–87.
17. Kumar, R., B. P. Bhushan, S. Gupta, Y. Sharma, N.,Gupta, (2014). Apache Hadoop, NoSQL and oSQL solutions of big data. International Journal of Advance Foundation and Research in Science & Engineering (IJAFRSE), Vol. 1, No.6, pp. 55-62.
18. Eze, F. U., C., Etus, J., Uzukwu (2014). Database System Concepts, Implementations and Organizations: A Detailed Survey. *International Journal of Scientific Engineering and Research* (IJSER), Vol. 2, No. 2, pp.
19. Byun, H. and S. W. Lee, (2002). Application of Support Vector Machines for Pattern Recognition: A Survey, Verri (Eds.): LNCS 2388, pp. 213-236.
20. Brand, H. (2013). Classification of Big Data Through NoSQL. In Proceedings of the National Conference on Innovation in Wireless Communication and Network Technology, Jaipur.doi:10.131140/2.1.3965.3768.
21. Mason, R. T. (2015). NoSQL Databases and Data Modelling Techniques for a Document-Oriented NoSQL Database. *Proceedings of Informing Science & IT Education Conference(InSITE)*2015,259268.Retrievedfromhttp://Proceedings.InformingScience.org/InSIT E2015/InSITE15p259-268Mason1569.pdf
22. Churnithipaisan, S., Z. A.Majeed, P.James, D. Parker, S. Abele (2003). Geospatial Interoperability via the Web. Supporting Land Adminstration in Kuala Lumpur. In Proceedings of MApAsia 2003 Conference, Kuala Lumpur, Malaysia, 13-15 October, 2003.
23. Alsubaiee, S., Y. Altowim, H. Altwaijry, H. Behm, V. Borkar, Y. Bu, T. Westman (2014). AsterixDB: A Scalable, Open Source DBMS. *Proceedings of the 40th International Conference on Very Large Data - Bases. 7(14)* Hangzhou, China.