MongoDB vs RE Model

A nosql solution to relational limitations

David Cosman

2017

MongoDB vs Relational Entity Model: A NoSQL Solution to Relational Limitations

The conception of the world as flat is an archaic and outdated theory of the Earth’s surface as a plane. However, the relational entity model family of query languages continue to push this paradigm of flat data instead of multi-dimensional data. [1] Flat data essentially remains static and persists throughout the lifetime of the application, structured in a way that allows little flexibility for future updates or changing data. This paper will consider a NoSQL language member, MongoDB [2], and contrast it with the relational language MySQL, [3] and will focus specifically on the implementation of the query language and how it shapes the design of the database structure. The relational data model is a business standard and used for its reliability in keeping data persistent throughout the application, while NoSQL databases exchange the consistency for flexibility [1]. Speaking a verbal language, such as English or French, and writing a programming language, such as Fortran or C, are both similar in the sense that they are a form of communication and arguing. Similarly, the design of the relational model vs the non-relational model both form a way of communicating and implementing a database. By analyzing studies in MySQL and MongoDB, as well as the applications of the relational model vs the non-relational model, this paper will ask and answer the following questions; how do MySQL and MongoDB implement and store data? [3] What do their inherent designs argue about the presentation and storage of data? [4] The world is not flat, and neither is the data that exists in the real world; does MongoDB provide a solution to the presentation of flexible data? In asking these questions, data will be presented on how to use MongoDB and the applications it can be used for [5] [6] [7]. Contrast will be made to how these applications are implemented in the relational model, and the experience from designing the database to user side interaction.

# References

|  |  |
| --- | --- |
| [1] | MongoDB, "Top 5 Considerations When Evaluating NoSQL Databases," November 2016. [Online]. Available: https://www.mongodb.com/collateral/top-5-considerations-when-evaluating-nosql-databases. [Accessed January 2017]. |
| [2] | MongoDB, "MongoDB," MongoDB, January 2017. [Online]. Available: https://www.mongodb.com/. [Accessed January 2017]. |
| [3] | P. Grover and R. Johari, "MVM: MySQL vs MongoDB," *Advances in Intelligent Systems and Computing,* vol. 436, pp. 899-909, 2016. |
| [4] | M. Flamm, "Asked & Answered: Dev Ittycheria, MongoDB," *Crain's New York Business,* vol. 33, no. 1, p. 8, 2017. |
| [5] | MongoDB, "Cybersecurity: Building the Next Generation of Threat Intelligence with MongoDB," October 2016. [Online]. [Accessed January 2017]. |
| [6] | H. Heni, "A Methodological Approach for Big Data Security: Application for NoSQL data stores," *SpringerLINK Lecture Notes in Computer Science,* pp. 685-692, 2015. |
| [7] | E. Bertheselen, "MongoDB," April 2014. [Online]. Available: https://www.mongodb.com/collateral/why-nosql-databases-internet-things-machina-research. [Accessed January 2017]. |