Name: Oscar Boateng Acheampong

Date: October 23rd, 2022

Assignment No.: Module 1.3 - Basic Comparison of Relational vs NoSQL Databases

In the context of relational databases, relations are the connections between two or more tables. This type of relationship is enforced using foreign keys and Joins. Foreign keys are referential constraints that references the primary key in a different table. Joins on the other hand are statements that are used to combine rows from two or more tables based on a common field between them.

There are several advantages of relational databases which justifies its extensive use by corporations and most business. Among these benefits are the following:

1. Ease of Use: Users can easily access information stored in the database in seconds or less without any complex query writing.
2. Accuracy: One major aspect of relational database is that it is strictly defined and well- organized, so data does not necessarily get duplicated without reason. This brings in the topic of normalization.
3. Normalization: As data becomes more complex and needs proper management in storing, normalization breaks it down to manageable sets to reduce redundancy. The opposite of this approach is de-normalization which also comes with its own benefits.
4. Data Integrity: In relational databases there are several integrity constraints that are put in place to serve as a check on data. Popular among them is the primary key constraint which uniquely identifies a row in a table and prevents duplicate records from being inserted into the table.

Also, non-relational databases also serve unique purposes in the database industry and comes with their own package of benefits. According to [www.mongodb.com](http://www.mongodb.com), these are some of the benefits of NoSQL:

1. Handle large volumes of data at high speed with a scale-out architecture
2. Store unstructured, semi-structured, or structured data
3. Enable easy updates to schemas and fields
4. Be developer-friendly
5. Take full advantage of the cloud to deliver zero downtime

However, both relational and non-relational databases have their disadvantages and among these disadvantages for relational databases are the following:

1. Maintenance Problem: When data increases in size, performance starts to become an issue and as a result people with the technical expertise needs to be brought on to manage the database and improve performance.
2. Physical storage: As relational database consists of rows and columns, storage becomes an issue when data grows, and the disk space becomes inadequate for both storage and performance.
3. Cost: The cost of maintaining a relational database is expensive, aside the price of the software, which some are open source and free, hiring a professional with expert knowledge of database development and/or administration can be expensive for small businesses.

Also, for NoSQL databases some of the disadvantages are the following:

1. Queries are less flexible as in, they cannot enforce or guarantee uniqueness for keys within documents like traditional relational systems do. So, this makes them less ideal if your application requires strict enforcement and unique key values across all records without having to write custom code or configurations each time an end-user creates a new record.
2. The concept is less mature and has few to no support group. Unlike SQL which has been around a lot linger and is generally universal and more mature with a vibrant support group online, it may be difficult for users of NoSQL to find solutions to problems online because of the non-existent support groups.

MySQL is a subset of SQL which is free to use and easy to understand. It is based on relational databases and is mostly used as the database system for web development and applications.

MySQL is an open source software, which means anyone can download and use or modify to suit their project as well as it does not breach the General Public License of the application. It also provides a secure interface since it has a password system which is flexible and ensures that it is verified based on the host before accessing the database. The password is encrypted while connecting to the server.

MongoDB also has features like Atlas, which is a leading global cloud database service for modern applications. Using Atlas, developers can deploy fully managed cloud databases across AWS, Azure, or Google Cloud. Best-in-class data security and privacy standards practices means that developers can rest easy knowing that they have instant access to the availability, scalability, and compliance they require for enterprise-level application development. It also has the Load Balancing feature which is seen as the holy grails of large-scale database management for growing enterprise applications. Properly distributing millions of client requests to hundreds or thousands of servers can lead to a noticeable (and much appreciated) difference in performance. There is no need to add an external load balancer, MongoDB ensures that each user has a consistent view and quality experience with the data they need to access

References:

<https://databasetown.com/relational-database-benefits-and-limitations/>

[www.mongodb.com](http://www.mongodb.com)

<https://www.adservio.fr/post/what-are-the-pros-and-cons-of-nosql>