ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 4, October 2016

Comparison of SQL Server and Mongo DB

Ahmed ShamelAbdullah, Abdulbary Raouf Suleiman

Abstract*—The topic of this paper is to compare SQL Server database and MongoDB database. Both of these databases can be used to store data of different types of applications such as web, mobile, and desktop applications. In many situations, choosing a certain database for an application is critical decision that needs a study for each type of database. This paper can help programmers who want to know the best database for a given application.

***Index Terms*—Database, SQL Server, MongoDB, SQL, NoSQL, BSON.**

I. INTRODUCTION

Storing data of application (web, mobile or desktop application) can be done using one (in some cases, it could be two) of the available databases such as SQL Server, Oracle, MongoDB, and CouchDB. Almost all databases can be classified according to their structure into two types: relation databases and NoSQL databases.

Relational database can be seen as a collection of organized and inter-related data on a related subject or topic [1]. This data is organized into sets (relations or tables) and each set of data may be related to another set of data. Also, there are rules regarding how each set of data is made up and how one set of data can be related to another set of data. Almost all relational database systems use SQL (Structured Query Language) as the language for querying and maintaining the database. The most commonly used relational database is SQL Server, MySQL, and Oracle [1].

NoSQL databasedoesn't store data in tables; instead, the data is stored in documents (which it is the most popular), graphs or in XML files [2]. The term NoSQL database was chosen for a loosely specified class of non-relational data stores. Such databases (mostly) do not use SQL as their query language. The term NoSQL is therefore confusing and in the database community is interpreted rather as “not only SQL” [2]. Sometimes the term post-relational is used for these data stores [2]. NoSQL database exists since 1960s but with no practical implementation. In 1998, Strozzi NoSQL database was released and it is considered the first well-know NoSQL database. From 1998 to now days, many NoSQL databases are come to world such as MarkLogic, MongoDB,Couchbase, CouchDB, and Datastax. All NoSQL database share a three common properties: they didn't use a fixed schema, they avoid joins (linking set of data to another set of data), and they scale horizontally [3].

SQL Server is one of the famous relational databases and it is used in widely in many applications [10]. On the other hand, MongoDB is the favorite choice when developer decides to use document NoSQL database [10]. This paper presents a comparison between these two databases. The goal of this paper is not to say that one of these databases is better than the other. It should be clear that MongoDB is not a replacement of SQL Server and vice versa. The aim of this paper is to identify the type of applications that will be ideal for SQL Server and MongoDB.

II. DEFINITIONAL COMPARISON

At its core, SQL Server is an enterprise-class database management system that is capable of running anything from personal databases only, a few megabytes in size on a handheld Windows Mobile device, up to a multi-server database system managing terabytes of information [4]. SQL Server is a relational database developed by Microsoft. As a database server, the primary function of it is to store and retrieve data as requested by other software applications which may run either on the same computer or on another computer across a network [4]. The first release of SQL Server was in 1989 and since that time, many features are added to it. SQL Server can be considered as general-purpose database management system that used mainly in web applications [4].

MongoDB (derived from the word humongous) is a relatively new breed of database that has no concept of tables, schemas, SQL, or rows [5]. Many features that exist in relation database are not available in MongoDB such as transaction integrity, data integrity, and foreign keys [5].MongoDB is probably a very different database than what earlier developers are used to, especially if developer has used a relational database management system (RDBMS) in the past. MongoDB is a document-orientated database [5]. The first MongoDB version was released in 2009. The aim was to create a database that worked with documents rather than rows, was blindingly fast, massively scalable, and easy to use [5]. To do this, the team had to leave some features behind, which means that MongoDB is not an ideal candidate for certain situations [5].

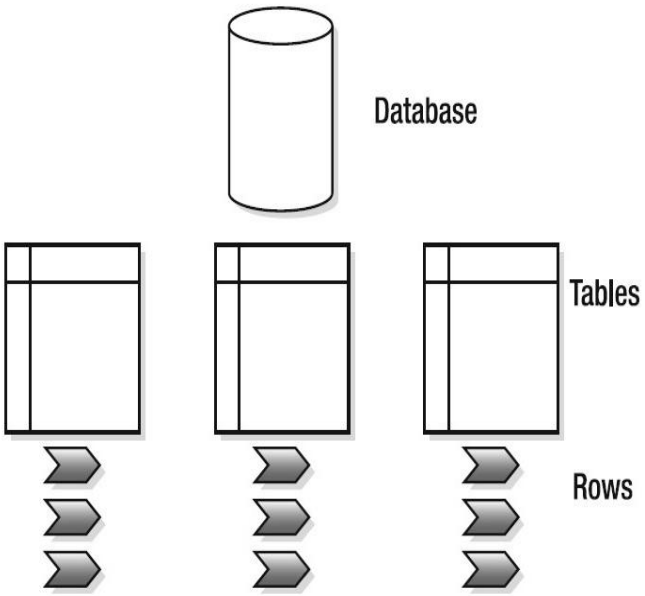
III. DATABASE STRUCTURE

SQL Server database consists of tables and tables consist of rows. Data is stored in the rows of the tables. SQL Server uses structured query language (SQL) to access database. In SQL Server, the developer should pre-define the database schema based on the requirements and set up rules to govern the relationships between rows in the table [4]. When creating a table in SQL Server, the developer should specify the name and data type of each column. If user makes wrong input, such as incorrect data type or out of range value, the database will refuse this input [4].

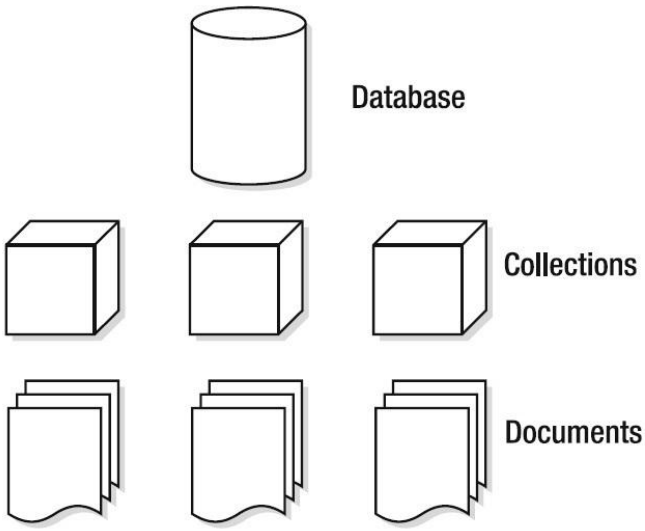
MongoDB database consists of collections and each collection consists of documents [5]. Data is stored in documents in the form of binary-JSON (BSON) with no-fixed structure [5].BSON is a binary-encoded serialization of JavaScript object notation (JSON) documents that supports the embedding of documents and arrays within other documents and arrays [7]. Data access to MongoDB database is done through the use of MongoDB query language [5].

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 4, October 2016

MongoDB uses dynamic schemas, which means that the developer can create a document without defining its structure. The developer can change the structure of documents simply by adding new fields or deleting existing ones [5]. This dynamic schema comes with advantage and disadvantage. The advantage is that it give the ability to store data in a flexible way without any constrain on storing process. The disadvantage is that the developer may accidently put wrong data in a document and the document will not refuse this wrong data because MongoDB designed to store data in the form their received [5]. Fig (1) shows the structure of SQL Server and MongoDB. Fig (2) shows data stored in SQL Server table and data stored in MongoDB collection (The collection is in JSON format for simplicity).



(a) Database structure in SQL Server



(b) Database structure in MongoDB

Fig (1) Database structure in SQL Server and MongoDB

```
{
Id:1,
Name:"User1",
Job:"Engineer"
}
{
Id:2,
Name:"User2",
Job:"Doctor",
Phone:"87644359"
}
{
Id:3,
Name:"User3",
Job:"Doctor",
Phone:{
        WorkPhone:"98032795",
        HomePhone:"89042759"
       }
}
```

(a) Data in SQL Server table

	Id	Name	Job	Phone
1	1	User1	Engineer	NULL
2	2	User2	Doctor	87644359
3	3	User3	Doctor	79987899

(b) Data in MongoDB document

Fig (2) Differences between SQL Server table and MongoDB document

IV. TRANSACTIONS INTEGRITY

A transaction is a sequence of operations executed as a single logical unit of work [6]. For example, changing the phone number or increasing the salary of employee is a transaction.

In SQL Server, the integrity of the transaction is ensured by forcing the transaction to have four properties, atomicity, consistency, isolation, and durability (ACID) [5]. In simple words, these four properties mean:

1. Atomicity: Each transaction is atomic. If one part of it fails, the entire transaction fails (and is rolled back) [6].
2. Consistency: Every transaction is subject to a consistent set of rules (constraints, triggers, cascades) [6].
3. Isolation: No transaction should interfere with another transaction [6].
4. Durability: The committed results in the database will survive on disks in spite of possiblesystem failures [6].

ACID adds overhead to the execution of SQL query because it forces database servers to use sequential evaluation.

MongoDB, on the other hand, does not have the concept of ACID and thus doesn't guarantee the integrity of the transaction [5]. In addition, MongoDB use an update method called lazy update [5]. Lazy update means that if value in document is needed to be updated more than once in the same second, then the update is performed only once [5]. For example, if the database receives 1000 transaction to increment a certain value in document, it will only be incremented once. Thus MongoDB is not suitable for applications that deal with accounts or banking information. Lack of transaction integrity is a solid tradeoff decision made

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 4, October 2016

by developers of MongoDB to make it simple, fast, and scalable [5].

V. DATABASE NORMALIZATION

The term "normalization" in database context refers to the methods used to organize columns and tables in order to avoid data redundancy and increase data integrity.

In SQL Server, database normalization is implemented with the help of foreign keys and joins features [4]. To illustrate the database normalization in SQL Server, suppose a simple articlesdatabase consisting of two tables: posts and authors. Table 1 shows the posts table and table 2 shows the authors table.By using foreign keys and joins, a single author in author table can have many posts in posts table. The Author_Id column in posts table is the foreign key that can be used to join the two tables. Organizing data in this way minimize data redundancy (author information is not repeated for every row in posts table). Also, normalizing database makes it easy to update author information without any change in posts table. Finally, using the foreign key increases data integrity because if user, for example, insert wrong Author_Id in the posts table (the wrong Author_Id means that there is no such Author_Id value in authors table), the database will refuse this insert operation.

MongoDB doesn't have foreign key [5]; but for the join feature, all versions prior to version 3.2.6 (which is released in 2016) don't support joins [8]. MongoDB provide two methods to save related documents: embedded data models and normalized data models [9]. With embedded data models, related data is saved in one document, which means that the data in this case will be denormalized [9]. The articles database, in case of using embedded data models, doesn't consist of two collections; instead, it is one collection that contains posts and authors information. However, in each post document the author information is embedded. Denormalize database leads to faster queries [5]; however, it makes updating author information very slow because all embedded documents need to be updated. Fig (3-a) shows the collection for the articles database. If MongoDB uses the normalized data models to store related data, then the related documents can be stored in different collections or databases [9]. Normalized data models use references in order to create relationships between documents. However, reading the related data in this case require two queries operations: the first query is to read the first document and the second query is to read the data in the other document [9]. Thus, using normalized data models produce slower read operations but it decrease data redundancy. Fig (3-b) shows the articles database in the case of using normalized data models.

Table (1) Posts table in articlesdatabase

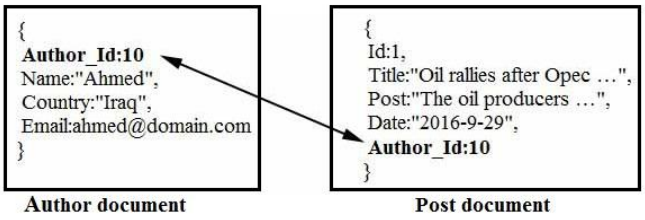
Id	Title	Post	Author_Id	Date
1	Oil rallies Opec …	The oil producers …	9	2016-9-29
2	Roman skeleton…	Archaeologists made …	8	2016-8-23
3	Austria's Graz airport …	Bemused customs officials in …	8	2016-8-26

Table (2) Authors table in articlesdatabase

Author_Id	Name	Country	Email
8	Ali	Iraq	ali@domain.com
9	Alex	USA	alex@domain.com

```
{
 Id:1,
 Title:"Oil rallies after Opec …",
 Post:"The oil producers …",
 Date:"2016-9-29",
 Author:{
       Name:"Ahmed",Country:"Iraq",Email:"ahmed@domin.com"
       }
}
{
 Id:2,
 Title:"Roman skeleton with …",
 Post:"Archaeologists made discovery …",
 "Date":"2016-8-23"
 Author:{
       Name:"Ahmed",Country:"Iraq",Email:"ahmed@domin.com"
       }
}
```

(a) Articles collection in MongoDB database using embedded data models



(b) Articles database in MongoDB database using embedded data models

Fig (3) Saving related data in MongoDB

In MongoDB version 3.2.6, MongoDB team decides to add the join feature to the database [10]. This decision was a surprise for the developers who use MongoDB because one of the bases that MongoDB build on is that it doesn't support joins. However, join feature in MongoDB performs left outer join only comparing to join feature in SQL Server which can performs seven different types of join operations (inner join, left outer join, right outer join, right excluding join, left excluding join, outer excluding join, and full outer join) [10].

VI. POPULARITY

Although popularity may be nota good scientific factor in the comparison of SQL Server and MongoDB, it is important factor for the developer. This is so because the more popular database means that there are more resources about it and more solutions for most problems that developers may face when using the popular database. According to db-engines.comwebsite, SQL Server is more popular than MongoDB [11] as shown in fig (4).

VII. CONCLUSION

SQL Server and MongoDB do the same thing but in different ways. SQL Server is a popular database that provides a fixed schema approach to store data. The integrity of transactions and data is guaranteed and data redundancy is

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 4, October 2016

avoided with SQL Server. Thus, SQL Server is general purpose database that can be used in most applications. However, SQL Server is more ideal for applications that need to store related data or when the data integrity is issue. Examples of these applications include web, banking data and data mart. On the other hand, MongoDB is less popular database that use dynamic schema when storing data. MongoDB is faster database; however, the integrity of transaction and data is not kept and is some cases the data may be redundant. Thus, MongoDB is ideal for applications that store unrelated data or when the integrity of data is not issue. MongoDB is suited for applications such as real time analytics, games, geospatial and mobile.

Rank			DBMS	Database Model	Score		
Sep 2016	Aug 2016	Sep 2015			Sep 2016	Aug 2016	Sep 2015
1.	1.	1.	Oracle	Relational DBMS	1425.56	-2.16	-37.81
2.	2.	2.	MySQL	Relational DBMS	1354.03	-3.01	+76.28
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1211.55	+6.51	+113.72
4.	↑ 5.	↑ 5.	PostgreSQL	Relational DBMS	316.35	+1.10	+30.18
5.	↓ 4.	↓ 4.	MongoDB	Document store	316.00	-2.49	+15.43
6.	6.	6.	DB2	Relational DBMS	181.19	-4.70	-27.95
7.	7.	↑ 8.	Cassandra	Wide column store	130.49	+0.26	+2.89
8.	8.	↓ 7.	Microsoft Access	Relational DBMS	123.31	-0.74	-22.68
9.	9.	9.	SQLite	Relational DBMS	108.62	-1.24	+0.97
10.	10.	10.	Redis	Key-value store	107.79	+0.47	+7.14

Fig (4) Popularity of SQL Server and MongoDB [10]

VIII. FUTURE ENHANCEMENT

This paper presents the major differences between SQL Server and MongoDB. There are other differences that are not mentioned in this paper such as query syntax complexity, storing data in main memory, and scaling (scaling in database context refer to the ability of the database to handle efficiently more work than it typically performs). These differences are recommended to be a topic of future paper.

Also, this paper can be used as a start point to compare other types of SQL and NoSQL databases such as a comparison of SQL Server and Couchbase or comparison of Oracle and MongoDB.

REFERENCES

[1] Fidel A Captain, Six-Step Relational Database Design(TM): A non-theoretical approach to relational database design and development, 1st Edition, ISBN: 1475039212, 2012.

[2] JaroslavPokorny, "NoSQL databases: a step to database scalability in web environment", International Journal of Web Information Systems, Vol. 9, issue 1, 2013.

[3] Adam Lith and JakobMattsson, "Investigating storage solutions for large data: A comparison of well performing and scalable data storage solutions for real time extraction and batch insertion of data", Department of Computer Science and Engineering, Chalmers University of Technology, Sweden,2010.

[4] Ray Rankins, Paul Bertucci, Chris Gallelli and Alex T. Silverstein, Microsoft SQL Server 2014 Unleashed, 1st Edition, ISBN-10: 0672337290, 2015.

[5] EelcoPlugge, Peter Membrey and Tim Hawkins, The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing, 1st Edition, ISBN-13 (electronic): 978-1-4302-3052-6, 2010.

[6] MarttiLaiho, Dimitris A. Dervos and Kari Silpiö, SQL Transactions Theory and hands-on exercises, ISBN 978-952-93-2421-7 (PDF).

[7] BSON Spec, BSON Definition, http://www.bsonspec.org/

[8] MongoDB, Database References,https://docs.mongodb.com/manual/reference/database-references/

[9] MongoDB, Data Model Design, https://docs.mongodb.com/manual/core/data-model-design/

[10] MongoDB, $lookup (aggregation), Definition, https://docs.mongodb.com/manual/reference/operator/aggregation/lookup/

[11] DB-Engines, Ranking database management systems according to their popularity [Online], Available at: http://db-engines.com/en/ranking, [Accessed 1 October 2016].