ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 3, March 2017

# MongoDB vs MySQL

[1] Mr. Parag S. Naik, [2] Mr. Ashish Sambare [3] Mr. Nikesh Aote [4] Mr. Swapnil Dravyakar
[1] Asst. Professor, PIET, Hingna, Nagpur ,[2] Asst. Professor, PIET, Hingna, Nagpur, [3] Asst. Professor, PIET, Hingna, Nagpur, [4] Asst. Professor, PIET, Hingna, Nagpur

***Abstract*— MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling. MongoDB obviates the need for an Object Relational Mapping (ORM) to facilitate development. MongoDB stores data in JSON-like documents that can vary in structure.**

## I. INTRODUCTION

MongoDB is an open-source database developed by MongoDB, Inc. MongoDB stores data in JSON-like documents that can vary in structure. Related information is stored together for fast query access through the MongoDB query language. MongoDB uses dynamic schemas, meaning that you can create records without first defining the structure, such as the fields or the types of their values. You can change the structure of records (which we call documents) simply by adding new fields or deleting existing ones.

This data model gives you the ability to represent hierarchical relationships, to store arrays, and other more complex structures easily. Documents in a collection need not have an identical set of fields and denormalization of data is common. MongoDB was also designed with high availability and scalability in mind, and includes out-of-the-box replication and auto-sharding. MongoDB supports field, range queries, regular expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions. Queries can also be configured to return a random sample of results of a given size. Many concepts in MySQL have close analogs in MongoDB. This table outlines some of the common concepts in each system.

| MySQL | MongoDB |
|---|---|
| Table | Collection |
| Row | Document |
| Column | Field |
| Joins | Embedded documents, linking |
| Primary Key | Primary Key |
| Group by | Aggregation |

, where the database is presented graphically in the form of a table, in MongoDB, a database has the following graphic structure:

{
"_id": "d4acaf3a76e4378b853eb15fde216722",
"username": "parag",
"email": "paragspnaik@gmail.com",
}
{
"_id": "d4rvgf3a76e4378b853tr15fde216722",
"username": "ashish",
"email": "ashishsambare@gmail.com",
}

## II. FEATURE COMPARISON

Like MySQL, MongoDB offers a rich set of features and functionality far beyond those offered in simple key-value stores. MongoDB has a query language, highly-functional secondary indexes (including text search and geospatial), a powerful aggregation framework for data analysis, and more. With MongoDB you can also make use of these features across more diverse data types than with a relational database, and at scale. Organizations of all sizes are adopting MongoDB because it enables them to build application faster, handle highly diverse data types, and manage application more efficiently at scale.

| MySQL | MongoDB | |
|---|---|---|
| Rich Data Model | No | Yes |
| Dynamic Schema | No | Yes |
| Typed Data | Yes | Yes |
| Data Locality | No | Yes |
| Field Updates | Yes | Yes |
| Easy for Programmers | No | Yes |
| Complex Transactions | Yes | No |
| Auditing | Yes | Yes |
| Auto-Sharding | No | Yes |

***Table 2: Feature Comparison***

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 3, March 2017

## III. NOSQL

A NoSQL (originally referring to "non SQL", "non relational") database provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases. The primary advantage of a NoSQL database is that, unlike a relational database it can handle unstructured data such as documents, email, multimedia and social media efficiently. Non-relational databases do not use the RDBMS principles (Relational Data Base Management System) and do not store data in tables, schema is not fixed and have very simple data model. Instead, they use identification keys and data can be found based on the keys assigned.

### *3.1 NoSQL Database Types*

There are 4 types of NoSQL databases:

1. **Key-Value Store: -** It has a BigHash Table of keys & Values.
2. **Document-based Store-It: -** It stores documents made up of tagged elements.
3. **Column-based Store: -** Each storage block contains data from only one column.
4. **Graph-based: -** Each storage block contains data from only one column.

#### *3.1.1 Key-Value Store*

The schema-less format of a key value database like Riak is just about what you need for your storage needs. The key can be synthetic or auto-generated while the value can be String, JSON, BLOB (basic large object) etc.

The key value type basically, uses a hash table in which there exists a unique key and a pointer to a particular item of data. A bucket is a logical group of keys – but they don't physically group the data. There can be identical keys in different buckets. Performance is enhanced to a great degree because of the cache mechanisms that accompany the mappings. To read a value you need to know both the key and the bucket because the real key is a hash (Bucket+ Key).

Example:-Riak and Amazon's Dynamo are the most popular key-value store NoSQL databases.

| Key | Value |
|---|---|
| "Student1" | {"101, Parag, Mongodb, 78"} |
| "Student2" | {"102, Sachin, C, 79"} |
| "Student3" | {"103, Mahi, Database, 80"} |

***Table 3: Key-Value Pair***

The key can be synthetic or auto-generated while the value can be String, JSON, BLOB (basic large object) etc.

This key/value type database allows clients to read and write values using a key as follows:

Get (key), returns the value associated with the provided key.

Put (key, value), associates the value with the key.

Multi-get (key1, key2, .., keyN), returns the list of values associated with the list of keys.

Delete (key), removes the entry for the key from the data store.

While Key/value type database seems helpful in some cases, but it has some weaknesses as well. One is that the model will not provide any kind of traditional database capabilities (such as atomicity of transactions, or consistency when multiple transactions are executed simultaneously). Such capabilities must be provided by the application itself.

#### *3.1.2 Document Store NoSQL Database*

The data which is a collection of key value pairs is compressed as a document store quite similar to a key-value store, but the only difference is that the values stored (referred to as "documents") provide some structure and encoding of the managed data. XML, JSON (Java Script Object Notation), BSON (which is a binary encoding of JSON objects) are some common standard encodings.

{officeName:"Calibre Technology",
{Street: "B-25, City: "Nagpur", State: "MH", Pincode: "440017"}
}

Example: - Apache Couchbase and MongoDB are the most popular document based databases.

#### *3.1.3 Column Store NoSQL Database*

In column-oriented NoSQL database, data is stored in cells grouped in columns of data rather than as rows of data. Columns are logically grouped into column families. Column families can contain a virtually unlimited number of columns that can be created at runtime or the definition of the schema. Read and write is done using columns rather than rows. In comparison, most relational DBMS store data in rows, the benefit of storing data in columns, is fast search/ access and data aggregation. Relational databases store a single row as a continuous disk entry. Different rows are stored in different places on disk while Columnar databases store all the cells corresponding to a column as a continuous disk entry thus makes the search/access faster.

The best known examples are Google's BigTable and HBase & Cassandra that were inspired from BigTable.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 3, March 2017

BigTable, for instance is a high performance, compressed and proprietary data storage system owned by Google. It has the following attributes:
Sparse- some cells can be empty
Distributed- data is partitioned across many hosts
Persistent- store to disk
Multidimensional- more than 1 dimension
Map- key and value
Sorted- maps are generally not sorted but this one is

| City | Picode | Strength | Project |
|---|---|---|---|
| Nagpur | 440017 | 250 | 20 |
| Mumbai | 400001 | 300 | 45 |
| Pune | 411014 | 150 | 10 |

***Table 4: RDBMS Table Example***

For above RDBMS table a BigTable map can be visualized as shown below:

{
CalibreTechnology: {
City: Nagpur
Pincode: 440017
},
Details: {
strength: 250
project: 20
}
}

Example: - Google's BigTable, HBase and Cassandra are the most popular column store based databases.

Graph Base NoSQL Database

This strategy can support complex data queries which are also performed in a relatively smaller period of time compared to other databases using the strategies mentioned above [1].

Also, non-relational databases provide high flexibility at addition or deletion of an attribute from the database because they do not have a fixed database schema.

## IV. query language

Both MySQL and MongoDB have a rich query language. A comprehensive list of statements can be found in the MongoDB documentation.

**Document Database**

A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents.

{
name: "parag", ← field: value
age: "29",
← field: value
status: "A"
← field: value
}

**Collections**

MongoDB stores documents in collections. Collections are analogous to tables in relational databases. Unlike a table, however, a collection does not require its documents to have the same schema.

In MongoDB, documents stored in a collection must have a unique _id field that acts as a primary key.

| MySQL | MongoDB |
|---|---|
| INSERT INTO users (user_id, age, status) VALUES ('uid101',29, A) | db.users.insert({ user_id:'uid101', age:45, status:'A' }) |
| SELECT * FROM users | db.users.find() |
| UPDATE user SET status='C' where age>25 | db.users.update( {age: {$gt : 25} }, {$set : {status : 'C'} }, {multi : true} ) |

***Table 5: MongoDB vs MySQL Query***

conclusion

The advantage of using MongoDB is that MongoDB provided lower execution times than MySQL. We can choose MongoDB instead of MySQL if the application is data intensive and stores many data and queries lots of data. Thus, more and more applications are beginning to use a non-relational database because they provide a more flexible structure that can shape after each user' needs; they are designed to store large amounts of data and they are denormalized databases, which increases performance.

## REFERENCES

Cornelia Gy_rödi, Robert Gy_rödi, George Pecherle, Andrada Olah, "A Comparative Study: MongoDB vs. MySQL," 2015 13th International Conference on Engineering of Modern Electric Systems (EMES).

S. Hoberman, "Data Modeling for MongoDB", Publisher by Technics Publications, LLC 2 Lindsley Road Basking Ridge, NJ 07920, USA, ISBN 978-1-935504-70-2, 2014.
Z. Wei-Ping, LI Ming-Xin, H. Chen, "Using MongoDB to Implement Textbook Management System instead of MySQL", IEEE 3rd International Conference on Communication Software and Networks (ICCSN), ISSN 978-1-61284-486, 2011.