Database Systems Lab

Lab#14 – MongoDB Introduction

Contents

MongoDB

MongoDB is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. Collections contain sets of documents and function which is the equivalent of relational database tables. MongoDB is a database which came into light around

MongoDB Features

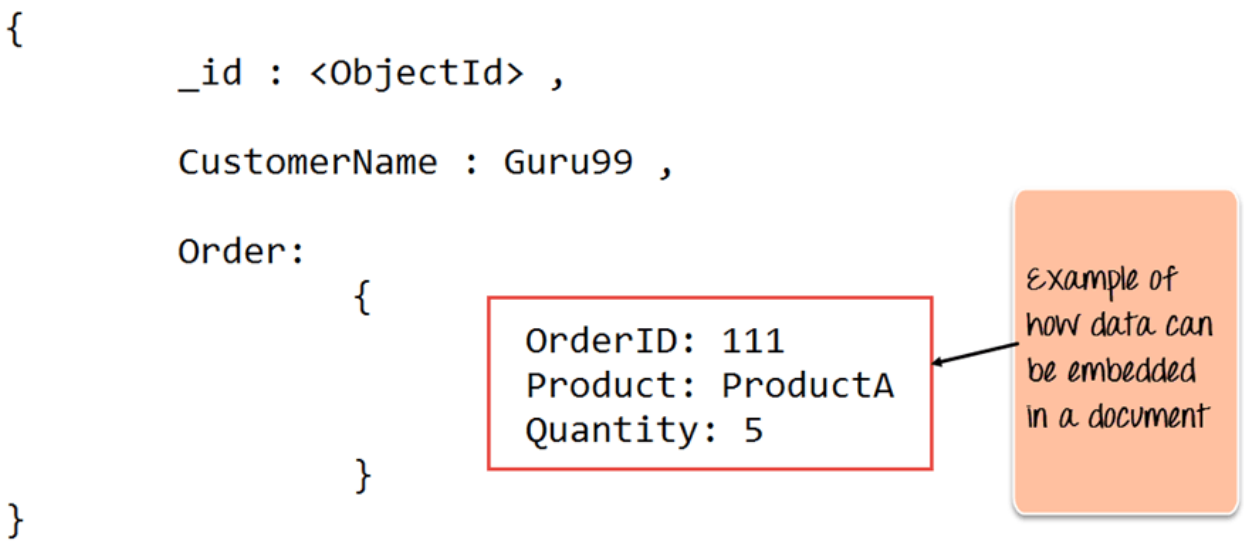
1. Each database contains collections which in turn contains documents. Each document can be different with a varying number of fields. The size and content of each document can be different from each other.
2. The document structure is more in line with how developers construct their classes and objects in their respective programming languages. Developers will often say that their classes are not rows and columns but have a clear structure with key-value pairs.
3. The rows (or documents as called in MongoDB) doesn't need to have a schema defined beforehand. Instead, the fields can be created on the fly.
4. The data model available within MongoDB allows you to represent hierarchical relationships, to store arrays, and other more complex structures more easily.

1. Scalability – The MongoDB environments are very scalable. Companies across the world have defined clusters with some of them running 100+ nodes with around millions of documents within the database

MongoDB Example

The below example shows how a document can be modeled in MongoDB.

1. The _id field is added by MongoDB to uniquely identify the document in the collection.
2. What you can note is that the Order Data (OrderID, Product, and Quantity) which in RDBMS will normally be stored in a separate table, while in MongoDB it is actually stored as an embedded document in the collection itself. This is one of the key differences in how data is modeled in MongoDB.



Key Components of MongoDB Architecture

Below are a few of the common terms used in MongoDB

1. **_id** – This is a field required in every MongoDB document. The _id field represents a unique value in the MongoDB document. The _id field is like the document's primary key. If you create a new document without an _id field, MongoDB will automatically create the field. So for example, if we see the example of the above customer table, Mongo DB will add a 24 digit unique identifier to each document in the collection.

_Id	CustomerID	CustomerName	OrderID
563479cc8a8a4246bd27d784	11	Guru99	111
563479cc7a8a4246bd47d784	22	Trevor Smith	222
563479cc9a8a4246bd57d784	33	Nicole	333

2. **Database** – This is a container for collections like in RDMS wherein it is a container for tables. Each database gets its own set of files on the file system. A MongoDB server can store multiple databases.
3. **Collection** – This is a grouping of MongoDB documents. A collection is the equivalent of a table which is created in any other RDMS such as Oracle or MS SQL. A collection exists within a single database. As seen from the introduction collections don't enforce any sort of structure.
4. **Document** - A record in a MongoDB collection is basically called a document. The document, in turn, will consist of field name and values.
5. **Field** - A name-value pair in a document. A document has zero or more fields. Fields are analogous to columns in relational databases.

   The following diagram shows an example of Fields with Key value pairs. So in the example below CustomerID and 11 is one of the key value pair's defined in the document.

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field

Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by MongoDB itself)

Difference between MongoDB & RDBMS

Below are some of the key term differences between MongoDB and RDBMS

RDBMS	MongoDB	Difference
Table	Collection	In RDBMS, the table contains the columns and rows which are used to store the data whereas, in MongoDB, this same structure is known as a collection. The collection contains documents which in turn contains Fields, which in turn are key-value pairs.
Row	Document	In RDBMS, the row represents a single, implicitly structured data item in a table. In MongoDB, the data is stored in documents.
Column	Field	In RDBMS, the column denotes a set of data values. These in MongoDB are known as Fields.
Joins	Embedded documents	In RDBMS, data is sometimes spread across various tables and in order to show a complete view of all data, a join is sometimes formed across tables to get the data. In MongoDB, the data is normally stored in a single collection, but separated by using Embedded documents. So there is no concept of joins in MongoDB.

Apart from the terms differences, a few other differences are shown below

1. Relational databases are known for enforcing data integrity. This is not an explicit requirement in MongoDB.
2. RDBMS requires that data be normalized first so that it can prevent orphan records and duplicates Normalizing data then has the requirement of more tables, which will then result in more table joins, thus requiring more keys and indexes.

   As databases start to grow, performance can start becoming an issue. Again this is not an explicit requirement in MongoDB. MongoDB is flexible and does not need the data to be normalized first.

Installation

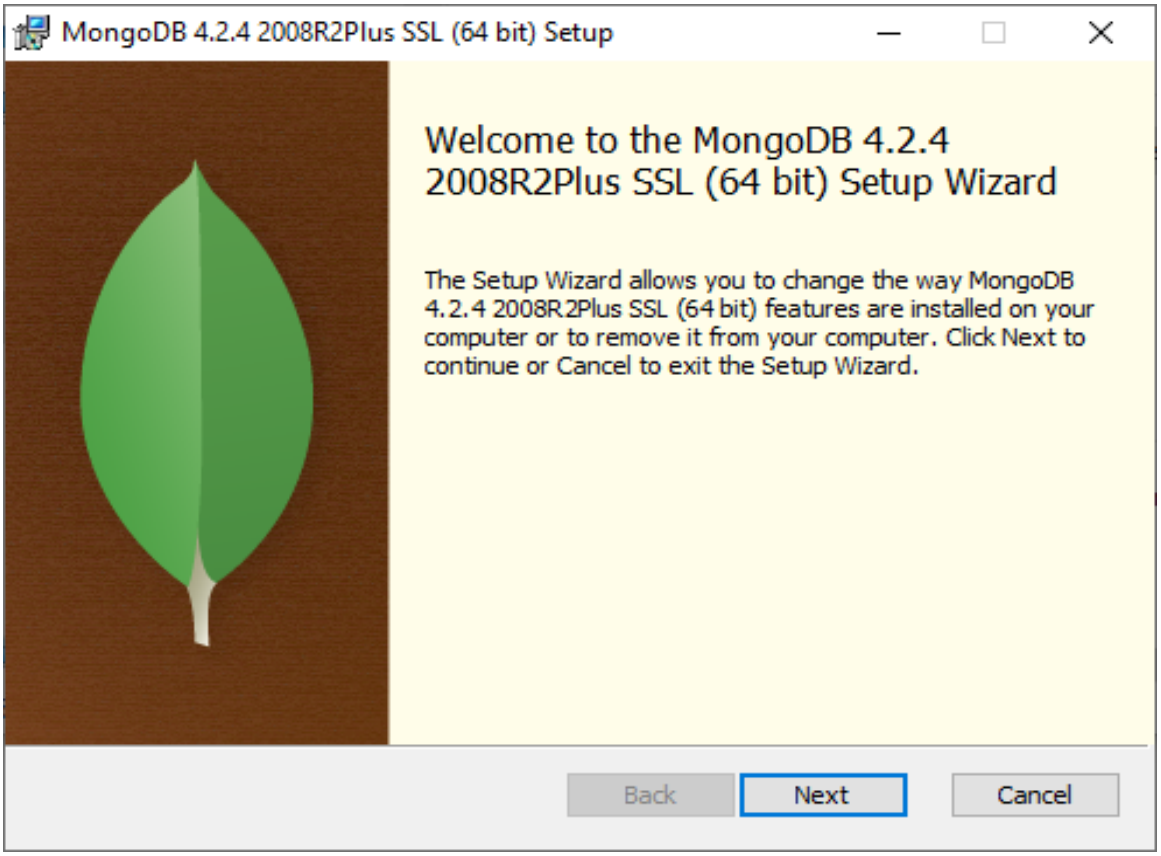
1. Installing Mongodb on windows

Downloading

Download from web: https://www.mongodb.com/download-center/community

Running the setup

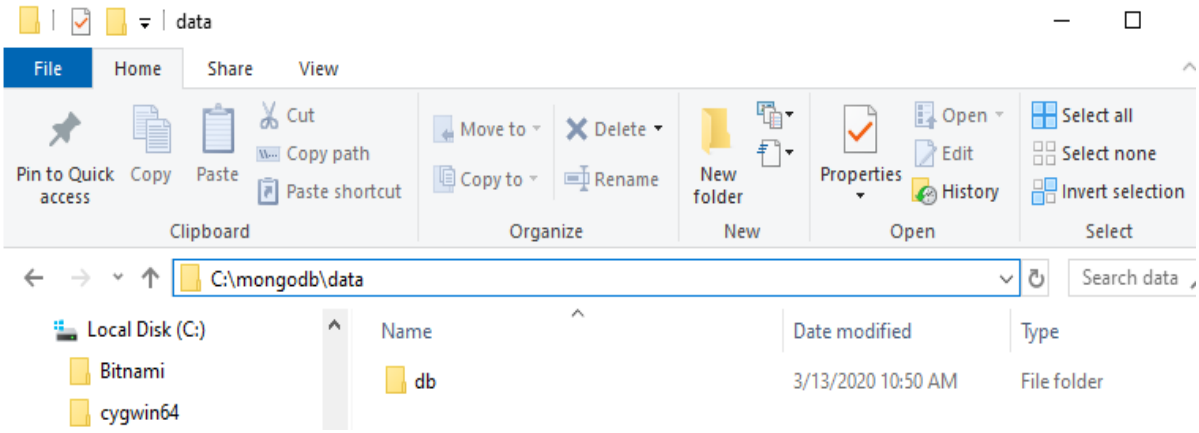
Double click the downloaded .msi to install the setup **[while installing chose custom option and change its location to a simpler path e.g C:\mongodb]**



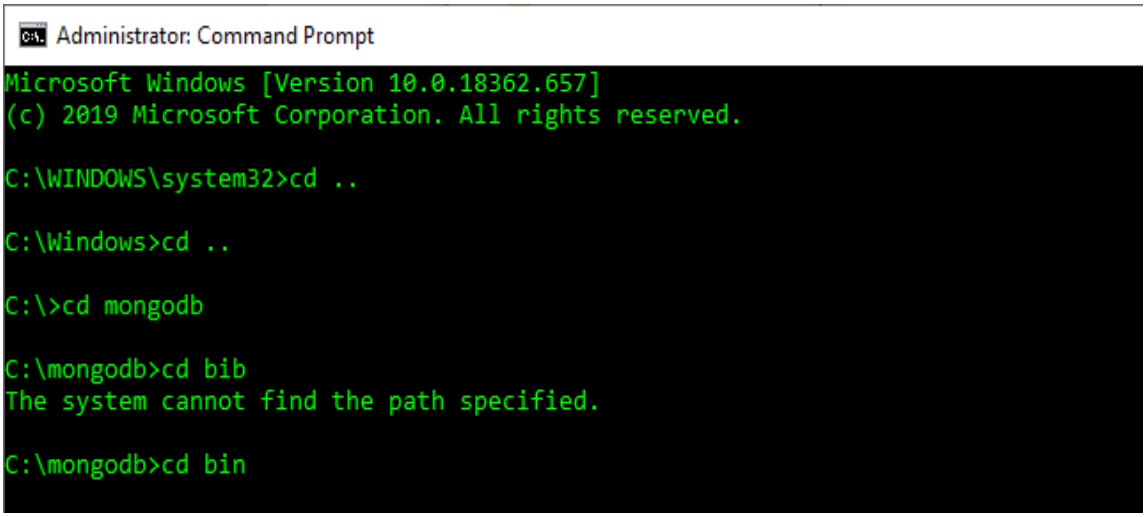
After finishing the setup. Open command prompt as an administrator

Configuration

To go the installation folder of mongodb -> data and make a new folder name db



Navigate to mongo directory, into bin folder



After that run this command:

```
mongod --directoryperdb --dbpath C:\mongodb\data\db --logpath C:\mongodb\log\mongo.log --logappend –install
```

After that to run services, run the following command:

```
net start Mongodb
```

incase to get an error run the following commands:

```
mongod --remove
```

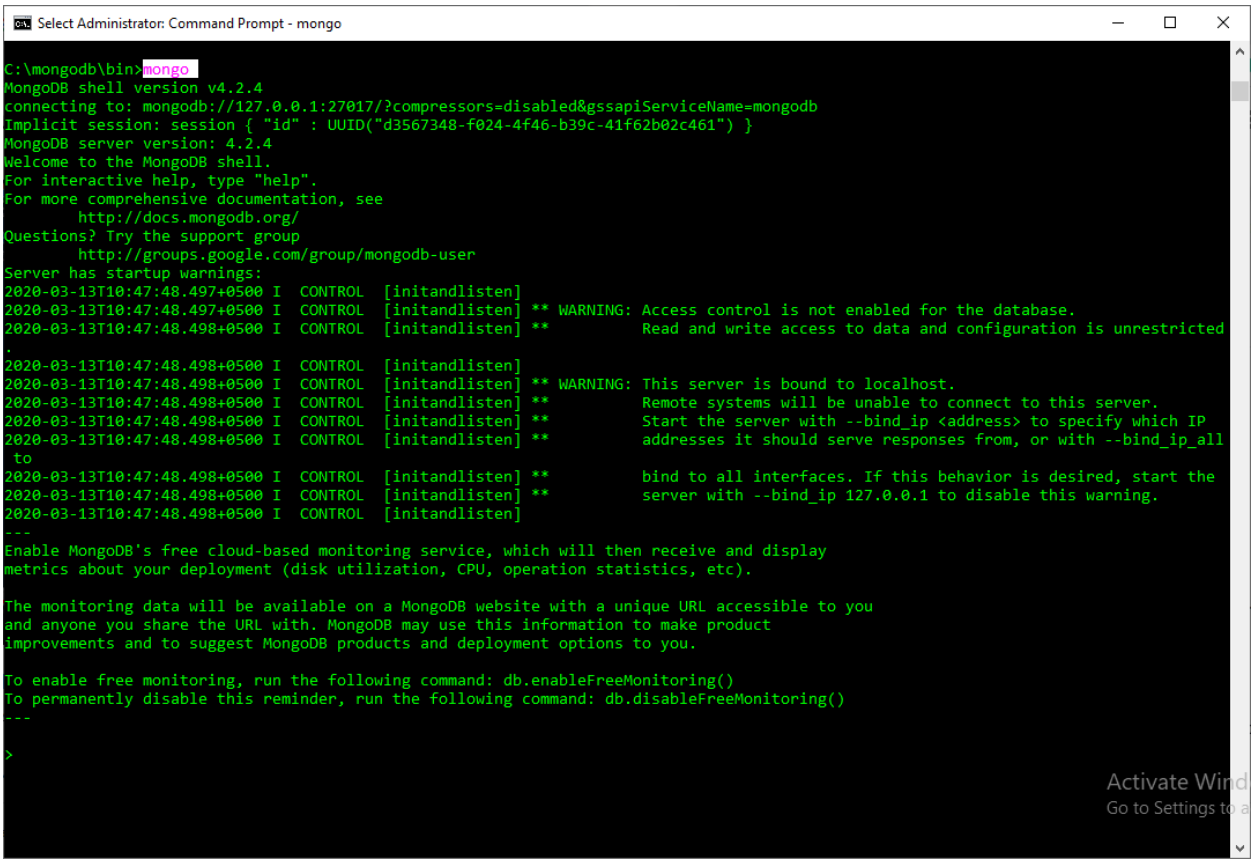
After that run the above command again.

```
C:\mongodb\bin>net start Mongodb
The service is not responding to the control function.

More help is available by typing NET HELPMSG 2186.


C:\mongodb\bin>mongod --remove
2020-03-13T10:47:16.897+0500 I  CONTROL  [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2020-03-13T10:47:16.904+0500 W  ASIO     [main] No TransportLayer configured during NetworkInterface startup
2020-03-13T10:47:16.905+0500 I  CONTROL  [main] Trying to remove Windows service 'MongoDB'
2020-03-13T10:47:16.906+0500 I  CONTROL  [main] Service 'MongoDB' removed

C:\mongodb\bin>mongod --directoryperdb --dbpath C:\mongodb\data\db --logpath C:\mongodb\log\mongo.log --logappend --install

C:\mongodb\bin>net start mongodb
The MongoDB service is starting..
The MongoDB service was started successfully.


C:\mongodb\bin>bsondump.exeI
```


Now the services are running in background

Running a mongo shell

After running the services successfully we will be working on mongo shell. For that just type **mongo**



To clear the screen just type “cls”, it will clear the screen

2. Installing Mongodb on Ubuntu

To go this link and watch the video I uploaded.

https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/

References:

https://www.tutorialspoint.com/mongodb/mongodb_overview.htm

https://www.w3resource.com/mongodb/introduction-mongodb.php