MongoDB Interview Questions & Answers

By Debasis Saha

Technical Manager, Author and Mentor

DotNetTricks

MongoDB Interview Questions & Answers

Release History

- Initial Release 1.0 - 18th Dec 2018
- Second Release 1.1 - 4th Jan 2019

About Dot Net Tricks

Dot Net Tricks is founded by Shailendra Chauhan (Microsoft MVP), in Jan 2010. Dot Net Tricks came into existence in form of a blog post over various technologies including .NET, C#, SQL Server, ASP.NET, ASP.NET MVC, JavaScript, Angular, Node.js and Visual Studio etc.

The company which is currently registered by a name of Dot Net Tricks Innovation Pvt. Ltd. came into the shape in 2015. Dot Net Tricks website has an average footfall on the tune of 300k+ per month. The site has become a cornerstone when it comes to getting skilled-up on .NET technologies and we want to gain the same level of trust in other technologies. This is what we are striving for.

We have a very large number of trainees who have received training from our platforms and immediately got placement in some of the reputed firms testifying our claims of providing quality training. The website offers you a variety of free study material in form of articles.

Dot Net Tricks Training Solutions

Dot Net Tricks provide you training in traditional as well as new age technologies, via various formats.

Master Courses (Instructor-led)

For a beginner who needs regular guidance, we have a fully packed Master Courses. They are almost equal to semester courses taught in engineering colleges when it comes to length, breadth of content delivery, the only difference instead of 5-6 months, they take approx. 16-weekend classes (2 months).

The detail about Master courses can be found here: https://www.dotnettricks.com/instructor-led-courses

Hands-On Learning (Learn to code)

Hands-On Learning courses give you the confidence to code and equally helpful to work in real-life scenarios. This course is composed of hands-on exercise using IDE or cloud labs so that you can practice each and everything by yourself. You can learn to code at your own pace, time and place.

The detail about Hands-On Learning courses can be found here: https://www.scholarhat.com

Skill Bootcamps (Instructor-led)

Professionals who don't have two months' time and want to get skilled up in least possible time due to some new project that their company has to take in very near future, we have designed Skill Bootcamps Concept, where you will get trained on consecutive days in a fast-paced manner, where our full focus is going to be on hands-on delivery of technological exercises.

The detail about Skill Bootcamps can be found here: https://www.dotnettricks.com/skill-bootcamp

Self-paced Courses

Self-paced courses give you the liberty to study at your own pace, time and place. We understand everyone has their own comfort zone, some of you can afford to dedicate 2 hours a day, some of you not. Keeping this thing in

mind, we created these self-paced courses. While creating these courses we have ensured that quality of courses doesn't get compromise at any parameter, and they also will be able to produce the same results as our other course formats, given the fact you will be able to put your own honest effort.

The detail about Self-paced courses can be found here: https://www.dotnettricks.com/self-paced-courses

Corporate Training (Online and Classroom)

Dot Net Tricks having a pool of mentors who help the corporate to enhance their employment skills as per changing technology landscape. Dot Net Tricks offers customized training programs for new hires and experienced employees through online and classroom mode. As a trusted and resourceful training partner, Dot Net Tricks helps the corporate to achieve success with its industry-leading instructional design and customer training initiatives.

The detail about Corporate Training can be found here: https://www.dotnettricks.com/corporate-training

Learning Platform

We have a very robust technology platform to answer the needs of all our trainees, no matter which program they enrolled in. We have a very self-intuitive Learning Management System (LMS), which help you in remain focused and keeping an eye over your progress.

We offer two Learning Platforms as given below:

1. Dot Net Tricks: https://www.dotnettricks.com
2. Scholar Hat: https://www.scholarhat.com

Apart from these, we also provide on-demand Skill bootcamps and personalized project consultation.

Dedication

My Parents and my wife deserve to have theirs name on the cover as much as I do for all theirs support made this possible. I would like to say thanks to all my friends, team member, to you and to readers or followers of my blog www.dotnettricks.com/mentors/debasis-saha to encourage me to write this book.

-Debasis Saha

Introduction

Writing a book has never been an easy task. It takes a great effort, patience and consistency with strong determination to complete it. Also, one should have a depth knowledge over the subject is going to write.

So, what where my qualification to write this book? My qualification and inspiration come from my enthusiasm for and the experience with the technology and from my analytic and initiative nature. Being a trainer, developer, consultant and blogger, I have through knowledge and understandings of different types of database technologies with both SQL & NOSQL database. My inspiration and knowledge have also come from many years of my working experience and research over it.

So, the next question is who this book is for? This book covers useful Interview Questions and Answers on MongoDB. This book is appropriate for novice as well as for senior level professionals who wants to strengthen their skills before appearing for an interview on MongoDB. This book is equally helpful to sharpen their programming skills and understanding MongoDB in a short time. This book is updated to latest version of MongoDB version 4.0.

What You'll Learn

This book is for those who want to learn MongoDB and also for those who are going to appear for the MongoDB interview to have a bright future in No SQL Database.

- MongoDB features
- Mongo DB Core concepts
- Mongo Shell
- Various Query Syntaxes
- Query Operators
- Database Operations
- MongoDB Indexes
- Aggregation
- Database Replication
- Sharding

All the best for your interview and happy programming!

About the Author

Debasis Saha - An Author, Architect and Mentor

With more than 10 years in hand experience, **Debasis Saha** is a great expertise in the domains of **Microsoft .NET technologies** and an array of other technologies including **JavaScript, AngularJS, Node.js and NoSQL Databases** to name but a few.

His extensive knowledge has been broadcasted as **articles-of-the-day** section on the Official **Microsoft ASP.NET Community Site.**

Debasis Saha is renowned for sharing his working experience, research and knowledge through his reputed and widely subscribed to blog www.dotnettricks.com/mentors/debasis-saha. His blog provides a vast storehouse of knowledge and support resources in the field of .NET technologies worldwide.

Moreover, and to his credit he has delivered **50+ technical event & workshop sessions** to professionals across different parts of India in Microsoft .NET technologies and other technologies including JavaScript, AngularJS, Node.js and NoSQL Databases.

How to Contact Us

Although the author of this book has tried to make this book as accurate as it possible but if there is something strikes you as odd, or you find an error in the book please drop a line via e-mail.

The e-mail addresses are listed as follows:

- mentor@dotnettricks.com
- info@dotnettricks.com

We always happy to hear from our readers. Please provide your valuable feedback and comments!

You can follow us on YouTube, Facebook, Twitter, LinkedIn and Google Plus or subscribe to RSS feed.

Table of Contents

1

MongoDB Introduction

Q1. What is MongoDB?

Ans. MongoDB is a document-oriented database. MongoDB is a powerful, scalable and much more data flexible database which provides a high performance in case of a large volume of data and also leads to a NoSQL Database. MongoDB stores the data in a BSON format.

Q2. What is NoSQL Database?

Ans. NoSQL stands for Not Only SQL. NoSQL is a category of Database Management System (DBMS) which does maintain all the rules of traditional RDBMS systems. It also does not use the conventional SQL syntaxes to fetch the data from the database. This type of database systems is typically used in case of a very large volume of data. Some of well-known NoSQL database systems are – Cassandra, BigTable, DynamoDB, MongoDB etc.

Q3. What are the types of NoSQL Database?

Ans. There are four types of NoSQL Database is available:

- **Document Database** – This type of NoSQL database is always based on a Document-Oriented approach to store data. The main objective of the Document Database is to store all data of a single entity as a document and all documents can be stored as a Collections. Some example of the Document Database – MongoDB, CosmosDB, CouchDB, PostgreSQL etc.
- **Key-Value Database** – This type of database stored data in a schema-less way since key-value storage is the simplest way of storing data. A key can be a point to any type of data, like an object, string or any other types of data. The main advantages of these databases are easy to implement and add data into it. Example – Redis, DynamoDB etc.
- **Column Store Database** – These types of databases store data in columns within a keyspace. The key space is always defined on a unique name, value and a timestamp. Example – Cassandra, BigTable, HBase, Vertica, HyperTable.
- **Graph Store Database** - These types of database mainly designed for data that can be easily represented as graph data. This means that data are interconnected with an undetermined number of data relations between them like family and social relations etc. Example – AllegroDB, GraphDB, OrientDB, Titan.

Q4. What are the advantages of MongoDB?

Ans. The main advantages of MongoDB are:

1. It can deal with a high volume of data.

2. It supports cross-platform
3. It provides High Performance
4. It is easily scalable
5. It does not require any complex joins to retrieve data
6. It supports both type of scaling – Horizontal & Vertical
7. It is available on any cloud-based environment like Azure, AWS etc

Q5. What are the main features of MongoDB?

Ans. The main features of the MongoDB are:

1. It supports JSON data models with dynamic schemas
2. We can perform a search operation in any fields of the documents
3. MongoDB support secondary indexes which helps us to search a variety of data
4. It supports aggregation pipeline
5. It supports Master-Slave replication
6. It supports automatic load balancing
7. It supports auto-sharding for horizontal scaling
8. It basically uses JavaScript objects in place of the procedure
9. It supports special collection type like Time-To-Live (TTL) for data storage which will expire automatically at a fixed time.

Q6. What are Documents?

Ans. The document is actually being the heart of MongoDB. In simple word, the document is an ordered set of keys with associated values. It is similar to the table rows in the RDBMS systems. It always maintains dynamic scheme so that it does not require any predefined structure or fields.

Q7. What is Collection?

Ans. A collection is a group of documents. The collection is basically representing the table objects in the RDBMS systems.

Q8. What are Dynamic Schemas?

Ans. In MongoDB, Collections always have dynamic Schemas. Dynamic Schemas means the documents within a single collection may contain different types of structure or shapes. For an example, both the below documents can be stored in a single collection:

```
{"message" : "Hello World"}
{"id" : 10, "description" : "India"}
```

Q9. What is a database?

Ans. In MongoDB, database means a group of collections. A single instance of MongoDB can be containing multiple databases where each of them can be grouping zero or more collections. Every database has its own permissions and each database is normally stored in a separate file in a disk.

Q10. What is the difference between MongoDB & RDBMS?

Ans. Below is the difference between MongDB & RDBMS systems

MongoDB	RDBMS
Database	Database
Collection	Table
Document	Rows
Field	Column
Embedded Document	Table Join

Q11. What is Mongo Shell?

Ans. MongoDB Shell is a JavaScript shell which allows us to interactions with MongoDB instances using the command line. It is very much useful to perform any administrative work along with any other data operations related commands. Mongo Shell is automatically installed when we installed the MongoDB in our computers.

Q12. What is mongod?

Ans. MongoDB always runs as a network server so that clients can connect with the server and perform operations on it. So, to start the server, we need to execute the command mongod. It basically starts the host process for the database and once it is started, it continued running in the background.

Q13. What is the default port of Mongod?

Ans. The default port of MongoDB is 27017.

Q14. What is Mongo?

Ans. Mongo is one of the important commands of the Mongo Shell which is mainly used to connect with a specific instance of the mongod. When we try to connect mongo command without any parameters, then it will try to connect the localhost with the port no 27017.

Q15. Where we can use MongoDB?

Ans. MongoDB can use used in the following areas –

a. Content Management System
b. Mobile Apps where data volume is very large and required high readability of data
c. Data Management
d. Big Data

Q16. Which languages supports MongoDB?

Ans. There are several languages which are supported by MongoDB like

- C++
- C
- C#
- Java
- Node.Js

- Perl
- PHP etc

Q17. How to retrieve the list of databases in MongoDB?

Ans. To display the list of databases in a MongoDB Server, use the command –

```
show dbs
```

Q18. How to create a New Database in MongoDB?

Ans. To create a new database, we need to use

```
use <databasename>
```

If the database exists then this command will return the database otherwise it will create the database.

Q19. How to drop the Database?

Ans. To drop a database, use the command –

```
db.dropDatabase()
```

This command will drop the selected database.

Q20. How to find the list of Collections in Database?

Ans. To display all the collections of the database, use the command –

```
show collections
```

Q21. What data types are supported by MongoDB?

Ans. MongoDB supports a wide range of data types a value in documents. The below is the available data types in the MongoDB.

Data Types	Descriptions
String	It is most commonly used data types. A string must be UTF-8 valid in MongoDB
Integer	It is used to store the numeric values. It may either 32 bit or 64 bits.
Boolean	It is used to store Boolean data types. It's valued either true or false.
Double	It is used to store floating point values.
Arrays	This data type is used to store a list or multiple values in a single key
Objects	This data type is used to store embedded data
Null	It is used to store null data
Date	This data type is used to store current date or time value in Unix time format.

Q22. What is ObjectId in MongoDB?

Ans. Each document stored in MongoDB must contain a "_id" key. The default value type of the "_id" is ObjectId. In a single collection, every document always contains a unique value of the "_id" field, so that every document can be identified easily. ObjectId always uses 12 bytes of storage. It always represents 24 hexadecimal digit string values.

Q23. What is BSON data types?

Ans. BSON stands for Binary JSON objects. BSON is a binary serialization of JSON documents. It also allows extensions which basically responsible for storing some data type which is not a part of the JSON objects. Also, BSON is a language independent data interchange format.

Q24. What is the maximum data size of a Document in MongoDB?

Ans. In MongoDB, the maximum size of a document is 16 MB.

Q25. What is the role of the MongoDB Profiler?

Ans. In MongoDB, the database profiler provides us with the performance characteristics of each operation in the database. Also, we can estimate any query performance in respect of the speed that which query is slow and which one is first.

Q26. What component is contains in the ObjectId value?

Ans. In MongoDB, object id contains the following four values –

1) Timestamp
2) Client Machine Id
3) Client Process Id
4) 3 bytes incremental counter

2
Database Operations

Q1. How to Insert single documents in MongoDB?

Ans. To Insert a Single document, we need to use –

```
db.<CollectionName>.insert({“Id”:1,”name”:”Kamalesh”,”Code”:”A001”})
```

Q2. How to Insert multiple documents in MongoDB?

Ans. To Insert a Multiple document at a time, we need to use –

```
db.<CollectionName>.insert(
    [
    {“Id”:1,”name”:”Kamalesh”,”Code”:”A001”},
    {“Id”:2,”name”:”Sumit”,”Code”:”A002”}
    ]
)
```

Q3. How to remove a Documents?

Ans. To remove a document, we need to use –

```
db.<CollectionName>.remove()
```

This command will remove all the documents within the collections. It will not remove the collections. If we want to remove any particular document based on some condition, then we need to use query option in the above command as below –

```
db.<CollectionName>.remove(“Id”:3)
```

Q4. How to Update a Document?

Ans. To update an existing document, we need to use –

```
      db.<CollectionName>.update(
    { “Id”:1},
    $set:{“name”:”Kamal”}}
)
```

Q5. What are the uses of Upserts?

Ans. Upserts is a special type of update command used in the MongoDB. If the search condition is not matched any document in the collection then this command will insert the mentioned document as a new one. If the search condition matched, then it will update the document normally.

```
db.<CollectionName>.update(
    { “Id”:1},
    {$set:{“name”:”Kamal”}},
true)
```

Q6. How to update multiple documents at a time?

Ans. Update command by default updates only the first document. If the search condition matched with multiple documents then also it will update the first document and rest of the documents will remains the same. So if we want to update all the document then we need to pass the fourth parameter value as true.

```
db.<CollectionName>.update(
    { “Id”:1},
    {$set:{“name”:”Kamal”}},
true,
true)
```

Q7. How to Find a Document using Find Command?

Ans. To find a document using find command, we need to use

```
db.<CollectionName>.find()
db.<CollectionName>.find{})
```

Q8. How to use conditions in Find Command?

Ans. To find a document using find command with any type of condition, we need to use

```
db.<CollectionName>.find(Search Condition)
db.<CollectionName>.find(“name”:”kamal”)
```

Q9. How to use OR operator in Find Command?

Ans. To use the OR operation, the syntax as below –

```
db.<CollectionName>.find(
    {"$or" : [{"Id" : 12}, {"name" : “Sumit”}]}
)
```

Q10. How to use AND operator in Find Command?

Ans. To use the AND operation, the syntax as below –

```
db.<CollectionName>.find(
    {"$and" : [{"Id" : 12}, {"name" : “Sumit”}]}
)
```

Q11. How to use IN operator in Find Command?

Ans. To use the IN operation, the syntax as below –

```
db.<CollectionName>.find({“Id” : {"$in" : [1,10,26]}})
```

Similarly, we can use the not in operator as below –

```
db.<CollectionName>.find({“Id” : {"$nin" : [1,10,26]}})
```

Q12. How to use $ne command?

Ans. To use the $ne or not equal to operation, the syntax as below –

```
db.<CollectionName>.find({“Id” : {"$ne" :10}})
```

Q13. How to use $lt command?

Ans. To use the $lt or less than equal to operation, the syntax as below –

```
db.Employee.find({“age”:{“$lt”:40}})
```

Q14. How to use the $lte command?

Ans. To use the $lte or less than and equal to the operation, the syntax as below –

```
db.Employee.find({“age”:{“$lte”:40}})
```

Q15. How to use the $gt command?

Ans. To use the $gt or greater than equal to operation, the syntax as below –

```
db.Employee.find({“age”:{“$gt”:40}})
```

Q16. How to use $gte command?

Ans. To use the $gte or greater than and equal to the operation, the syntax as below –

```
db.Employee.find({“age”:{“$gte”:40}}
```

Q17. How to use limit command?

Ans. This command is used to limiting the total number of records during the execution of the find() commands. This command is normally set the upper limit of records. The syntax of this command is

```
db.<CollectionName>.find().limit(5)
```

Q18. How to use sort command?

Ans. This command is working as similar to the limit() syntax. The main difference between this two syntax is that it first skip the number of records which is mentioned and then display the remaining records. The syntax of this command is

```
db.<CollectionName>.find().skip(5)
```

The above syntax will skip the first 5 records of the document and then return the rest of the documents.

Q19. How to use skip command?

Ans. This command is used to display the searched document results in a particular order. This command takes two parameters – first one for mentioning field name or object name on which sort need to be done and the second one for mentioning sort order (its accept 1 or -1 as value, 1 for ascending order and -1 for descending order). The syntax of this command is

```
db.<CollectionName>.find().sort({“salary”:1})
```

3
Index

Q1. Why index is required in MongoDB?

Ans. Indexes are required to execute the query faster. If we don't use an index, then MongoDB perform the table-scan on the entire document when we execute any search condition. But in the case of an index, it just looks for the ordered list of the documents as per the index key.

Q2. How to create an index in MongoDB?

Ans. To create Index on any MongoDB collection, we need to use the below command –

```
db.<CollectionName>.ensureIndex({ColumnName:1})
```

Q3. What are different types of Index in MongoDB?

Ans. In MongoDB, there are several types of indexes are available. They are –

a) Single Field Index
b) Compound Index
c) Multikey Index
d) Text Index
e) Geospatial Index
f) Hashed Index

Q4. What are different index properties are available in MongoDB?

Ans. In MongoDB, there are several types of index properties are available. They are –

a) Unique Index
b) Partial Index
c) Sparse Index
d) TTL Index (Time-To-Live)

Q5. How to retrieve index information of a collection?

Ans. In MongoDB, we can retrieve information related to the created indexes on any documents using the command –

```
db.<CollectionName>.getIndexes()
```

Q6. How to drop an Index in MongoDB?

Ans. To drop an existing index from MongoDB, we can use this command –

```
db.<CollectionName>.dropIndex(<indexname>)
```

Q7. What is the default index in MongoDB?

Ans. In MongoDB, the default index is a Unique Index.

Q8. What is the concept of explain()?

Ans. Explain() is the one of the most important diagnostics command in the MongoDB. By executing these commands, we can find out which indexes are being used in the query. Basically, these commands will produce the total query plan in the MongoDB. Explain() method always returned two types of output: indexed and non-indexed queries. The most popular type of explain() is that collection does not have an index on which query will be executed. It tells us that query doesn't use an index because it uses a "BasicCursor". On the other hand, most queries which use indexes are used "BtreeCursor". The execution command of explain() is –

```
db.<CollectionName>.find({"age":10}).explain()
```

Q9. What is a unique index?

Ans. Unique Index is always ensuring us that each value in the documents within a collection is always coming once. So, for an example, if we decide that in the Employee collection, employee name can't be the same for one than one collection then we need to use a unique index on that collections.

```
db.Employee.ensureIndex(
    {"employeeName":1},
    {"unique":true}
)
```

Q10. What is Sparse Index?

Ans. As we know, a unique index is always counted null as a value, so that we can't use a unique index with more than one document missing that key. But, in real life scenario, many times we want to implement a unique index if the key or column contains any value. So, if we have a field which may or may not exist but it must be unique when it contains any value, then we can use the unique index along with the sparse option. To create a sparse index, we need to include the sparse options. For example, if we want to implement an email address of employee collection to be unique if the email value is provided, then we need to execute the below command –

```
db.Employee.ensureIndex(
    {"email":1},
    {"unique":true, "sparse" : true}
)
```

Q11. What is Capped Collection in MongoDB?

Ans. In MongoDB, normal collections are created dynamically at the time of data insertion and automatically grow in size so that additional data can be fit. But in MongoDB, there is also another type of collections which a pre-created collection is always along with fixed size. These types of collection are known as Capped Collection.

But one question is that What will happen when the size of the capped collection is exceeded? Actually, capped collection always behaves like circular queues. So, when the capped collection is filled up and new a document is inserted then the oldest one document will be deleted automatically. So, to create a capped collection, the command will be –

```
db.createCollection(“CollectionName”, {“capped”:true, “size” : 100000})
```

Q12. What is the TTL Index?

Ans. As we know, capped collections always give us limited control over their content. So, if we need more control over it, then we can use Time-To-Leave (TTL) indexes so that we can set a lifespan of each document in the collection and whenever the lifespan is over that document will be deleted automatically. This type of index is very much useful for storing caching type data. To create a TTL index, we need to run a command as

```
db.<CollectionName>.ensureIndex(
    {“CreatedOn”:1},
    {“expireAfterSecs”:60*60*24}
)
```

The above command creates a TTL index on CreatedOn with a lifespan of 1 day. Since once the time crosses the expire after secs value of that document then it will automatically be deleted.

Q13. What is the Full Text Index?

Ans. Full-Text Index is one of the special types of Index in MongoDB for searching text within the documents. Full-Text index always gives us the ability to search text quickly. But this type of index is basically expensive for uses concern. So, creating a full-text index on a busy collection can overload the MongoDB Server. So it is always recommended to use this type of index in an offline mode. To create a full-text index, the command is –

```
db.<CollectionName>.ensureIndex({“name” : “text”})
```

Q14. What is Geospatial Index?

Ans. Geospatial indexes are used in MongoDB for indexing any geospatial data like GeoJSON objects or coordinate pairs. MongoDB supports two types of Geospatial indexes – 2dsphere and 2d

Q15. What is GridFS?

Ans. Since in MongoDB, every document size limit is 16 MB. So, if we want to insert any large binary data file, then we need to use GridFS. GridFS is a mechanism through which we can store any type of large file data like an audio file, video file or image etc. It is basically just like a file system to store these large files and also, its related data stored in the MongoDB collection.

Q16. What are the benefits of using GridFS?

Ans. The benefits of the GridFS are –

a) GridFS is providing a choice to store a large file within MongoDB.
b) GridFS is applicable in any existing replication or auto sharding implementation.
c) We can achieve the great disk locality with GridFS since MongoDB allocates each data files in a 2 GB data chunk.

Q17. What are the disadvantages of GridFS?

Ans. The disadvantages of GridFS are –

a) GridFS is slower in the performance point-of-view. The reading files from MongoDB using GridFS is slower compared to the reading file in the file system
b) We can only modify the documents by deleting the document and resaving the whole document. Since MongoDB always stores the file in multiple document chunks.

Q18. How to create 2dSphere Indexes?

Ans. To Create 2dSphere index, we need to run the below command –

```
db.<CollectionName>.ensureIndex({“fieldName” : “2dsphere”})
```

In the above command, index field must be containing value type of GeoJSON objects or a coordinate pair. This type of index mainly supports queries that calculate geometrics on an earth-like data sphere.

Q19. How to create 2D Indexes?

Ans. To Create 2d index, we need to run the below command –

```
db.<CollectionName>.ensureIndex({“fieldName” : “2d”})
```

In the above command, index field must be containing value type of a coordinate pair. This type of index mainly supports queries that calculate geometrics on a two-dimensional data plane.

Q20. What is Index Cardinality in MongoDB?

Ans. Index Cardinality always refers to how many distinct values there are in the field of the collections. The fields which contain two or three possible value always have low cardinality like “gender”, “title” etc. On the other hand, fields like “name” or “email” always have a high cardinality since they normally contain a unique value in the collection. In general concept, the higher cardinality of a field always helps the index on that field. Because in this case, the index can quickly narrow down the search option to a smaller result set.

4
Aggregation

Q1. What is the purpose of using the Aggregation Framework?

Ans. Aggregation Framework is mainly used to transform and combine the documents within a collection. This framework always returns the array of the document as a result set. It is first performing the transformation of the documents and then combines the documents within a collection and return the result. Actually, the aggregation framework builds a pipeline which processes the stream of documents through several building blocks – filtering, projecting, grouping, sorting, limiting and skipping.

Q2. What is Aggregation Pipeline Operators?

Ans. In MongoDB aggregation framework, every command operates on a stream of documents, then performs some type of transformations and then return the result of the transformation. If it is the last operator in the pipeline then it will return the result to the user. Otherwise, the result transferred to the next operator as input.

Q3. What is the purpose of use $match?

Ans. $match command is used to filter the document so that we will able to run the associated aggregation commands on a set of documents. If our main objective to filter the documents then we can use the $match command as below –

```
{$match : {“city” : “DELHI”}}
```

In general, the $match can be used to all available question operators (“$gt”, “$lt”, “$in” etc). But, it is not applicable to any geospatial operators.

Q4. What is the purpose of use $project?

Ans. The $project is actually acting as a projection operator within the aggregation pipeline. This command does not use as normal command outside the aggregation framework or pipeline. The $project actually used to extract fields from any subdocuments or rename any fields. If we want to use $project then we need to use the command as below

```
db.Employee.aggregation(
    {“$project” : {“name” : 1, “_id” : 0 }}
)
```

In the above syntax, the name is the field name which is given to the aggregation framework.

Q5. What is the purpose of use $group?

Ans. $group syntax is used to bundle or group the documents of a collection on the basis of the one or more fields. So, if we want to group the data on depends on one or more than one fields, then we need to pass those fields name within the group method to create a group key and normally the group key name is "_id".

```
{"$group" : {"_id" : {"state" : "$state", "city" : "$city"}}}
```

We can use any type of arithmetic operator with the group command as below –

```
db.sales.aggregate(
{
 "$group" : {
 "_id" : "$country",
 "totalRevenue" : {"$average" : "$revenue"},
 "numSales" : {"$sum" : 1}
}
})
```

Q6. What is the purpose of use $unwind?

Ans. $unwind command is basically used to create every field of any subdocuments array into a separate document. Suppose, we have a document like as below –

```
{
    "author" : "k",
    "post" : "Hello, world!",
    "comments" : [
        {
            "author" : "mark",
            "date" : ISODate("2013-01-10T17:52:04.148Z"),
            "text" : "Nice post"
        },
        {
            "author" : "bill",
            "date" : ISODate("2013-01-10T17:52:04.148Z"),
            "text" : "I agree"
        }
    ]
}
```

Now, if we run the unwind command on the above document, then the result document will be below –

```
{
"results" :
    {
        "_id" : ObjectId("50eeffc4c82a5271290530be"),
        "author" : "k",
```

```
        "post" : "Hello, world!",
        "comments" : {
            "author" : "mark",
            "date" : ISODate("2013-01-10T17:52:04.148Z"),
            "text" : "Nice post"
        }
    },
    {
        "_id" : ObjectId("50eeffc4c82a5271290530be"),
        "author" : "k",
        "post" : "Hello, world!",
        "comments" : {
            "author" : "bill",
            "date" : ISODate("2013-01-10T17:52:04.148Z"),
            "text" : "I agree"
        }
    }
],
"ok" : 1
}
```

The syntax of the unwind command is –

```
db.<CollectionName>.aggregate({"$unwind" : "$comments"})
```

Q7. What is the purpose of use $sort?

Ans. $sort command is used to sort the document of the collection as per the given field name. $sort command accepts one or more than one fields as input. Also, we can provide the sort order against every field name. The value of the sort order is either 1 (for ascending order) or -1 (for descending order). The syntax of the sort command is

```
db.<CollectionName>.aggregate(
    {“$project” :
        {“name” : 1, “_id” : 0 }
    },
    {“$sort” :
        {“name” : 1}
    }
)
```

Q8. What is the purpose of use $limit?

Ans. $Limit always takes a number as input, say n and return first n result of the documents.

```
db.<CollectionName>.aggregate(
    {“$project” :
```

```
        {“name” : 1, “_id” : 0 }
    },
    {“$sort” :
        {“name” : 1}
    }
).limit(5)
```

Q9. What is the purpose of use $skip?

Ans. $skip also takes a number, say n and then remove the first n documents from the result set and then return the result.

```
db.<CollectionName>.aggregate(
    {“$project” :
        {“name” : 1, “_id” : 0 }
    },
    {“$sort” :
        {“name” : 1}
    }
).skip(5)
```

Q10. What are the objectives of MapReduce?

Ans. MapReduce is one of the most powerful tools in the Aggregation Framework. It is used to run any type of JavaScript as a query language so that we can perform some advanced business logic. But as per the performance point of view, it is too much slow and always advisable not to run on the active database.

Q11. How to use count()?

Ans. The count is the simplest aggregation tool in MongoDB. These aggregation tools return the total number of documents in the collections

```
db.<CollectionName>.count()
```

Q12. How to use distinct()?

Ans. The distinct command is used to finds all of the distinct values for a given key. For this, you must provide a collection and key:

```
db.runCommand({"distinct" : "people", "key" : "age"})
```

Q13. How to use group()?

Ans. The group allows you to perform more complex aggregation. You choose a key to group by, and MongoDB divides the collection into separate groups for each value of the chosen key.

5
Replication

Q1. What is Replication?

Ans. Replication is the process which is responsible for keeping identical copies of our data in multiple servers and is always a recommended process for any type of production server. Replication always keeps our database safe even if database server crashed or data corrupted. With the additional copies, MongoDB maintains the copy of the data for disaster recovery, reporting or backup purpose.

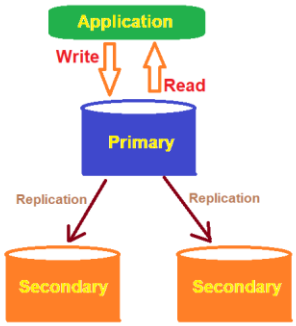
Q2. Why Replication is required in MongoDB?

Ans. In MongoDB, replication is required due to the following reason –

a. To provide always the availability of data
b. To secure our application data
c. Recover the data from any type of disaster recovery
d. In the Replication process, no downtime requires for maintenance like backup, index rebuilds etc.
e. Replication can provide us with a read scaling means it will provide us with a copy of data only for read purpose.

Q3. What is a replica set?

Ans. A replica set is a group of MongoDB instances or servers which will maintain the same data set. Every replica set must have a primary node for all types of write operations and one or multiple secondary nodes which will always be updated from the primary replica set so that data can be identical with the primary node. For creating replication in MongoDB, we need at least 3 replica set node. For any type of failover of the primary node, an election process has been initialized among all available secondary node and one secondary will be selected as a primary node. Once the failed primary node recovered, then it will join the flow as a secondary node and continue the working process. A typical diagram of replication is shown below –



Q4. What is the primary replica set?

Ans. The primary replica set is the replica set which accepts all the write operations from the user.

Q5. What is the secondary replica set?

Ans. The secondary replica set is the replica set which is used to replicate the oplog of the primary replicate set and also apply the operation to their datasets so that secondary replica set always looks similar like primary replica set.

Q6. How to setup a replica set?

Ans. In MongoDB, we can convert a standalone MongoDB server to a replication server. For that purpose, we need to start the MongoDB server with the below command –

```
mongod --port <port> --dbpath <db_path_url> --replSet <replicaset_instance_name>
```

Q7. How to initialize the replica set?

Ans. To initialize the replication set, run the following command

```
rs.initiate()
```

Here rs is the global variable which contains the replication helper functions.

Q8. What is syncing?

Ans. Replication means keeping an identical copy of data in multiple servers. For doing this operation, MongoDB maintains an operations log file for maintaining all write operations. The secondary replica set query this operation log file known as oplog for replicate data. Earn secondary replica set maintains its own log which contains the details of replication from the primary node. This process is known as the sync process among the replication servers.

Q9. What is Heartbeat request?

Ans. In the replication process, a member needs to know about who is the primary node, who is down and from whom they will sync the data. For this purpose, every member in the replication server sends a request to other members in every two seconds which is known as heartbeat request. A heartbeat request is a small message which checks everyone's state.

Q10. What is the election process?

Ans. When a member does receive a response from the primary node against the heartbeat request, then that member send a notice to all member for that primary node is not reachable. Then all the other member will accept that request and start the process make one secondary node as a primary node. This process is known as the election process in the replication server.

Q11. What is replica set oplog?

Ans. In MongoDB, the oplog records details information of all the operations which are responsible for any type of data modification in the replica set.

Q12. What is the purpose of the replication?

Ans. In MongoDB, replication always provides the data redundancy and increase the data availability.

Q13. How to add a new member in the Replica Set?

Ans. In MongoDB, we can add a new member in the replica set by using the below commands –

```
rs.add(Server_Name:Port_No)
```

Q14. How to remove a member from the Replica Set?

Ans. In MongoDB, we can remove any replica set server by using the below command –

```
rs.remove(Server_Name:Port_No)
```

Q15. How to stop a replica set from running?

Ans. In MongoDB, we can stop the replica set by using this command –

```
replicaSet.stopSet()
```

6
Sharding

Q1. What is Sharding?

Ans. Sharding is actually indicated a process which is responsible for splitting data across different machines. Sometimes we used the term partitioning to explain this concept. This process is used to meet the requirement of data growth. Since for a large volume of the database, when the database size increases, then it is always not possible to store the data in a single server. Sharding also solved the problem of horizontal scaling.

Q2. Why Sharding is required?

Ans. Sharding is required as per the below reason –

1. In the Replication server, all database writes normally goes to master node
2. In the case of Replication, Single Replica set can contain maximum 12 nodes
3. High latency queries normally go to the master node
4. In case of a large dataset or data, memory can't be a large enough to operate into the database
5. When the local storage system is not a high capacity
6. When we want to implement vertical scrolling

Q3. What are the components of Sharding?

Ans. In the Sharding process, there are 3 components related –

1. Shards
2. Config Servers
3. Query Routers

Q4. What are Shards?

Ans. In the Sharding Process, Shards is one of the key components which is responsible for storing data and it will provide high availability and data consistency. In the production environment, each shard normally maintains a replica set of the database.

Q5. What are Config Servers?

Ans. Config Servers is normally used to maintain or store the cluster's metadata. This metadata basically contains a mapping information related to the cluster's data set with the shards. This metadata is used by the Query Routers for any target operations for a specific shard. In a production environment, every cluster must have exactly 3 config serves.

Q6. What is Query Routers?

Ans. Query Routers are normally mongo server instances which directly interact with the client application and then perform direct operations with the appropriate shards in Sharding system. Query routers will process the data and the target operations to shards and then return the result to the clients. In the sharding process, a sharded cluster always contains more than one query routers as per the client request load. A client always communicates with a single query router always. Basically, the sharded cluster can contain many query routers.

Q7. What time do we need to implement Sharding?

Ans. Deciding when to perform sharding is a balancing act. In general, we do not want to use sharding at the very early stage of the mongo database and also it is not recommended to use sharding when database volume is too big. Since at that time, we need downtime to implement sharding. So, normally sharding is used when we required

a. To increase available RAM
b. To increase available disk space
c. Want to reduce the load on the server
d. Want to be read or write data with a large volume of throughput in compared to a single mongod instances

Q8. How to configure Sharding?

Ans. To create a sharded cluster, we need to use below command –

```
cluster = new ShardingTest({“shards”:3, “chunksize”:1})
```

Executing this command, it creates a new cluster with 3 shards (mongod processes) which running on the port 30000,30001 and 30002.

Q9. How to add a Shard from a ReplicaSet?

Ans. To add a shard from a replica set, we need to execute the below command –

```
sh.addShard("replica-server:27017")
```

Q10. How to Shard data of a Collection?

Ans. In a normal scenario, MongoDB would not distribute the data automatically until we tell it to do. For this purpose, we need to tell database name and collection name which need to be sharded. For enable sharding in a collection, we need to use the command –

```
db.enableSharding(“employee”)
```

If we enable sharding in the database level, then we can shard a collection by using the command –

```
sh.shardCollection("employee ", {"name" : 1})
```

Since in the above command, we use sharding in the name field, that means employee collection must have an index on the name field. Otherwise, it will return an error. In that case, we need to run the create index and retry the shardCollection.

Q11. What is Shard Key?

Ans. When we shard a collection, need to choose one or more field of that collection to split the data. This field (or fields) are called Shard Key.

Q12. What is the different type of Sharding?

Ans. There are 3 possible ways in which we can shard the database –

1. Ascending Shard Key
2. Randomly Distributed Shard Key
3. Location Based Shard Key

Q13. What are the limitations of Shard Key?

Ans. Shard key contains some limitations as below –

a. Shard Key can't be an array of objects
b. After inserted data into the document, shard key data can't be modified.
c. Geospatial Index can't be used as a shard key

Q14. What is the default port number of ShardingTest?

Ans. By Default, ShardingTest starts a mongos on the port 30999.

References

This book has been written by referring to the following sites:

1. https://docs.mongodb.com/ - MongoDB Docs
2. https://stackoverflow.com/questions/tagged/mongodb - Stack Overflow - MongoDB
3. https://www.dotnettricks.com/learn/mongodb - Dot Net Tricks - MongoDB