**DBVersity MongoDB Assessment**



1. **MongoDB manages collections of \_\_\_\_\_\_\_\_\_ like documents which are internally stored in binary format referred to as BSON?**
2. XML
3. JavaScript
4. HTML
5. JSON
6. All of above
7. **Purpose of MongoDB Replication and Sharding respectively?**
8. Horizontal Scalability & High Availability
9. High Availability & Horizontal Scalability
10. Vertical Scalability & Manual Failover
11. None of above
12. **Which of the following is correct explanation of MongoDB processes?**

A) mongod.exe is the shell process and mongo.exe is the actual database process

B) mongo.exe is the shell process and mongod.exe is the actual database process

C) mongos.exe is the MongoDB server process needed to run database

D) mongodump.exe can be used to import database backup dump

1. **Which of the following collections are used by MongoDB to store GridFS data?**

A) fs.files and fs.chunks

B) fs.grid and fs.chunks

C) fs.parts and fs.files

D) fs.chunks and fs.parts

1. **Which of the following SQL terminology is same as $match in MongoDB?**

A) WHERE

B) HAVING

C) Both WHERE and HAVING

D) GROUP BY

1. **In a replica set, a \_\_\_\_\_\_\_\_ number of members ensures that the replica set is always able to select a primary.**

A) Odd

B) Even

C) Depends on the application architecture

D) 2

1. **Which of the following is a valid MongoDB JSON document:**

A {}

B {

"user\_id"=1,

"user\_name"="Joe ",

"occupation"=["engineer","writer"]

}

C {

"user\_id":1;

"user\_name":"Joe ";

"occupation":["engineer","writer"]

}

D {

"user\_id":1,

"user\_name":"Joe ",

"occupation":[

"occupation1":"engineer",

"occupation2":"writer"

]

}

1. **What is the output of following two commands in MongoDB:**

**db.posts.insert({"\_id":1})**

**db.posts.insert({"\_id":1})**

A) Two documents will be inserted with \_id as 1

B) MongoDB will automatically increment the \_id of the second document as 2

C) This will throw a duplicate key error

D) It will insert two documents and throw a warning to the user

1. **Which of the following command is used to get all the indexes on a collection?**

A) db.collection.getIndexes()

B) db.collection.showIndexes()

C) db.collection.findIndexes()

D) db.showIndexes()

1. **Which of the following methods can be used in MongoDB for relation documents?**

A) SQL Joins

B) References/DBRefs

C) Aggregations

D) There is no concept of relations in documents

1. **Which type of indexes does MongoDB support?**

A) Compound Indexes

B) Text Indexes

C) Geospatial Indexes

D) All of the above

1. **Consider a collection posts which has fields: \_id, post\_text, post\_author, post\_timestamp, post\_tags etc. Which of the following query retrieves ONLY the key named post\_text from the first document retrieved?**

A) db.posts.find({},{\_id:0, post\_text:1})

B) db.posts.findOne({post\_text:1})

C) db.posts.findOne({},{post\_text:1})

D) db.posts.findOne({},{\_id:0, post\_text:1})

1. **What kind of database MongoDB is?**

A)Graph Oriented

B)Document Oriented

C)Key Value Pair

D)Column Based

1. **Which is the default mode in which the explain() command runs?**

A)queryPlanner

B)executionStats

C)allPlansExecution

D)customExecutionStats

1. **In a sharded replica set environment, the w Option provides ability for write concern and j Option provides ability for the data to be written on disk journal. Consider that we have a seven member replica set and we want to assure that the writes are committed to journal. What should be the value of j?**

A)0

B)1

C)2

D)7

1. **Given the following posts document:**

**{**

**"\_id" : 1,**

**"post\_text" : "This post does not matter",**

**"tags": [ "tutorial", "fun", "learning"],**

**// rest of the document**

**}**

**What will be the output of following query:**

**db.posts.aggregate( [ { $unwind : "$tags" } ] )**

A)Return three separate documents for three separate tags

B)Arranges the tags (wind) in ascending order

C)Arranges the tags (wind) in descending order

D)Returns one document but converts the tags array in an object

1. **In a posts collection, which command can be used to find all the posts whose author names begin lie between "A" and "C" in dictionary order?**

A)db.posts.find( { post\_author : { $gte : "A" , $lte : "C" } } );

B)db.posts.find( { post\_author : { $gte : "C" , $lte : "A" } } );

C)db.posts.find( { post\_author : { $gt : "A" , $lt : "C" } } );

D)This type of search is not supported by MongoDB. $lt and $gt operators are applicable only for numeric values.

1. **What does the output x of the following MongoDB aggregation query result into:**

**db.posts.aggregate( [ { $group: { \_id: "$author", x: { $sum: "$likes" } } } ] )**

A)Average of likes on all the posts of an author, grouped by author

B)Number of posts by an author

C)Sum of likes on all the posts by an author, grouped by author

D)Sum of likes on all the posts by all the authors

1. **Suppose that you have the following three documents in your system:**

**{ \_id: 1, product\_code: "123456", description: "mongo db tutorial" }**

**{ \_id: 2, product\_code: "345567", description: "this is mongo tutorial" }**

**{ \_id: 3, product\_code: "123431", description: "my mongo" }**

**If you create a text index on the description field and then apply a text search on term "mongo", which all documents will be fetched.**

A)1 and 2

B)2 and 3

C)1 and 3

D)All of the above

1. **Which of the following methods can be used on a cursor object?**

A)cursor.next()

B)cursor.hasNext()

C)cursor.forEach()

D)All of the above

1. **Which of the following is not a system collection in MongoDB?**

A)database.system.indexes

B)database.system.namespaces

C)admin.system.users

D)admin.system.preferences

1. **What does the following aggregate query perform?**

**db.posts.aggregate( [**

**{ $match : { likes : { $gt : 100, $lte : 200 } } },**

**{ $group: { \_id: null, count: { $sum: 1 } } }**

**] );**

A)Calculates the number of posts with likes between 100 and 200

B)Groups the posts by number of likes (101, 102, 103) by adding 1 every time

C)Fetches the posts with likes between 100 and 200 and sets their \_id as null

D)Fetches the posts with likes between 100 and 200, sets the \_id of the first document as null and then increments it 1 every time

1. **What happens if an index does not fit into RAM?**
2. **How is MongoDB better than other SQL databases?**
3. **What does pretty() in MongoDB query will do ?**