**What kind of database MongoDB is?**

A - Graph Oriented

B - Document Oriented

C - Key Value Pair

D - Column Based

**Which of the following command can be used in mongo shell to show all the databases in your MongoDB instance?**

A - show dbs

B - show databases

C - show dbs -all

D - ls dbs

**How many data types are there in JSON?**

1 (just strings)

2

4

6

**What is the corresponding JSON for the following XML document?**

**<person>**

**<name>John</name>**

**<age>25</age>**

**<address>**

**<city>New York</city>**

**<postalCode>10021</postalCode>**

**</address>**

**<phones>**

**<phone type="home">212-555-1234</phone>**

**<phone type="mobile">646-555-1234</phone>**

**</phones>**

**</person>**

**For the following XML, Is the corresponding JSON example legal json?**

**<things>**

**<hat>one</hat>**

**<coat>z</coat>**

**<hat>two</hat>**

**</things>**

{

"hat" : "one",

"coat" : "z",

"hat" : "two"

}

**The mongoimport utility can import what types of data?**

CSV (comma separated values)

TSV (tab separated values)

XML

JSON

BSON

plain text

**Which of the following is a valid MongoDB JSON document?**

1. {}

{

"user\_id"=1,

"user\_name"="Joe Sanders",

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

}

{

"user\_id":1;

"user\_name":"Joe Sanders";

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

}

D

{

"user\_id":1,

"user\_name":"Joe Sanders",

"occupation":[

"occupation1":"engineer",

"occupation2":"writer"

]

}

**Which of the following replica sets vote in the election of a primary replica set?**

1. Secondary
2. Hidden
3. Delayed
4. All of the above

**Consider that our posts collection contains an array field called tags that contains tags that the user enters.**

**{**

**\_id: 1,**

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

**post\_text: "This is my first post",**

**//other elements of document**

**}**

**Which of the following commands will find all the posts that have been tagged as tutorial?**

1. db.posts.find( { tags : "tutorial" } );
2. db.posts.find( { tags : ["tutorial"] } );
3. db.posts.find( { $array : {tags: "tutorial"} } );
4. db.posts.findInArray( { tags : "tutorial" } );

**Which option can be used with update command so that a new document gets created if no matching document is found based on the query condition?**

1. Specify {upsert : true } as the third parameter of update command
2. upsert command instead of update command
3. {update: true, insert: true} as the third parameter of update command
4. This has to be handled in application code (Node.js, PHP, JAVA, C#, etc.) and cannot be handled in mongo shell query

**What does the totalKeysExamined field returned by the explain method indicate?**

1. Number of documents that match the query condition
2. Number of index entries scanned
3. Number of documents scanned
4. Details the completed execution of the winning plan as a tree of stages

**Update If Correct is an approach for which of the following concepts in MongoDB:**

1. Concurrency Control
2. Transaction Management
3. Atomicity
4. Performance Management

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

1. Manual References
2. DBRefs
3. Both a and b
4. There is no concept of relations in documents

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

1. db.collection.getIndexes()
2. db.collection.showIndexes()
3. db.collection.findIndexes()
4. db.showIndexes()

**Which of the following operators can reverse the effects of a double unwind operation?**

1. $push
2. $wind
3. $wind.$wind
4. Can't be reversed.

**The \_\_\_\_\_\_\_\_ operator limits the contents of an array field from the query results to contain only the first element matching the query condition.**

1. $
2. $elemMatch
3. $slice
4. An array cannot be retrieved element wise in MongoDB.

**Which of the following MongoDB query is equivalent to the following SQL query:**

**UPDATE users SET status = "C" WHERE age > 25**

1. db.users.update(

{ age: { $gt: 25 } },

{ status: "C" })

1. db.users.update(

{ age: { $gt: 25 } },

{ $set: { status: "C" } })

1. db.users.update(

{ age: { $gt: 25 } },

{ $set: { status: "C" } },

{ multi: true })

1. db.users.update(

{ age: { $gt: 25 } },

{ status: "C" },

{ multi: true })

**Within how much time does MongDB writes are written to the journal?**

1. 60 s
2. 100 ms
3. 1 s
4. 100 s

**Which of the following is supported by MongoDB?**

1. ACID Transactions
2. Relationships between Collections (Primary Key - Foreign Key)
3. Journaling
4. Transaction Management

**Consider that our posts collection contains an array field called tags that contains tags that the user enters.**

**{**

**\_id: 1,**

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

**post\_text: "This is my first post",**

**//other elements of document**

**}**

**What does the following command return:**

**db.posts.find( { 'tags.0': "tutorial" } )**

1. All the posts whose tags array contains tutorial
2. All the posts which contains only one tag element in the tag array
3. All the posts having the first element of the tags array as tutorial
4. All the posts which contains 0 or more tags named tutorial

**Which are the ONLY ways to project portions of an array?**

1. $elemMatch
2. $slice
3. $
4. All of the above

**Update If Correct is an approach for which of the following concepts in MongoDB:**

1. Concurrency Control
2. Transaction Management
3. Atomicity
4. Performance Management

**Consider that you have a collection called population which has fields state and city. Which of the following query will calculate the population grouped by state and city?**

1. db.population.aggregate( [{ $group: { \_id: { state: "$state", city: "$city" },pop: { $sum: "$pop" } } }] )
2. db.population.aggregate( [{ $group: { \_id: { state: "$state", city: "$city" },pop: { $sum: 1 } } }] )
3. db.population.aggregate( [{ $group: { \_id: { state: "$state", city: "$city" },pop: { "$pop": 1 } } }] )
4. db.population.aggregate( [{ $group: { \_id: { city: "$city" },pop: { $sum: "$pop" } } }] )Multi Document Transaction is not supported by MongoDB

**Given a collection posts as shown below having a document array comments, which of the following command will create an index on the comment author descending?**

**{**

**"\_id":1,**

**"post\_text":"This is a sample post",**

**"author":"Tom",**

**"comments":[**

**{**

**"author":"Joe",**

**"comment\_text":"This is comment 1"**

**},**

**{**

**"author":"Leo",**

**"comment\_text":"This is comment 2"**

**}**

**]**

**}**

1. db.posts.createIndex({"comments.$.author":-1});
2. db.posts.createIndex({"comments.author":-1});
3. db.posts.createIndex({"comments.author":1});
4. db.posts.createIndex({"comments.$.author": {"$desc":1}});

**The oplog (operations log) is a special capped collection that keeps a rolling record of all operations that modify the data stored in your databases. All the replica set members contain a copy of the oplog in the following collection:**

1. oplog.rs
2. local.oplog.rs
3. ..oplog.rs
4. .oplog.rs

**Which command can be used to rebuild the indexes on a collection in MongoDB?**

1. db.collection.createIndex({reIndex:1})
2. db.collection.createIndex({author:1}).reIndex()
3. db.collection.reIndex()
4. db.collection.reIndex({author:1})

**mongoimport command is used to:**

1. import all the data from one database to another
2. import all the data from one collection to another
3. imports content from an Extended JSON, CSV, or TSV export created by mongoexport
4. import all the MongoDB data from one format to another

**Which option should be used to update all the documents with the specified condition in the MongoDB query?**

1. updateAll instead of update
2. specify {multi : true} as the third parameter of update command
3. specify {all: true} as the third parameter of update command
4. specify {updateAll: true} as the third parameter of update command

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

1. Compound Indexes
2. Multikey Indexes
3. Geospatial Indexes
4. All of the above

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

1. database.system.indexes
2. database.system.namespaces
3. admin.system.users
4. admin.system.preferences

**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?**

1. 0
2. 1
3. 2
4. 7

**What is a replica set node which does not maintain its own data but exists only for voting purpose called?**

1. Secondary
2. Arbiter
3. Delayed
4. Hidden

**Which index is used to index the content stored in arrays?**

1. Multikey Index
2. Compound Index
3. Text Index
4. Sparse Index

**The MongoDB explain() method does not support which of the following verbosity mode:**

1. queryPlanner
2. executionStats
3. allPlansExecution
4. customExecutionStats

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

1. queryPlanner
2. executionStats
3. allPlansExecution
4. customExecutionStats

**Which of the following operations on a single document will operate atomically?**

1. update
2. $push
3. Both a and b
4. None of the above

**What does the following query do when performed on the posts collection?**

**db.posts.update({\_id:1},{Title:This is post with ID 1"})**

1. Updates the Title of the post
2. Updating a document is possible only with $set
3. Replaces the complete document with \_id as 1 with the document specified in second parameter
4. Syntax error

**Which are the ONLY ways to project portions of an array?**

1. $elemMatch
2. $slice
3. $
4. All of the above

**As per the aggregation pipeline optimization concepts, if you have a $sort followed by a $match:**

1. $match moves before $sort
2. $sort moves before $match
3. MongoDB does not do any movements by default and will use the order provided
4. Providing these parameters in any order does not impact the performance

**Which of the following commands create an unique index on author field of the posts collection?**

1. db.posts.createIndex({"author":1 }, {"unique": true});
2. db.posts.createIndex({"author": unique });
3. db.posts.createIndex({"author": {"$unique":1} });
4. db.posts.createIndexUnique({"author":1 });

**In a sharded replica set environment, w defines the level and kind of write concern. Which of the following values of w specifies to return success only after a majority of voting members have acknowledged?**

1. n
2. majority
3. m
4. major

**If you have created a compound index on (A,B, C) which of the following access pattern will not be able to utilize the index?**

1. A, B, C
2. A, B
3. B, C
4. A

**What is the maximum size of Index Key Limit and Number of Indexes per collection?**

1. 64 bytes and 1024 indexes
2. 12 mega bytes and 64 indexes
3. 1024 bytes and 64 indexes
4. 1024 bytes and unlimited indexes

**Which of the following command can be used to check the size of a collection named posts?**

1. db.posts.stats()
2. db.posts.findStats()
3. db.posts.find({stats:1})
4. db.stats({ collection : posts })

**What does the following query do when performed on the posts collection?**

**db.posts.update({\_id:1},{$set:{Author:"Tom"}})**

1. Sets the complete document with \_id as 1 with the document specified in second parameter by replacing it completely
2. Adds a new field Author in the searched collection if not already present
3. Updates only the Author field of the document with \_id as 1
4. Both b and c

**Which option should be used with findAndModify() command to return the modified document instead of the pre-modification document?**

1. findAndModify by default returns the pre-modification document
2. Set {new : true}
3. Use the POST version of findAndModify called findAndModifyPost
4. Both b and c are valid

**What is the default size of a GridFS chunk?**

1. 16 MB
2. 255 K
3. 1 MB
4. 2 MB

**What will be the equivalent MongoDB command for the following SQL command:**

**SELECT author, count(\*) FROM posts GROUP BY author HAVING count(\*) > 1**

1. db.posts.aggregate( [

{

$group: {

\_id: "$author",

count: { $sum: 1 },

$match: { count: { $gt: 1 } }

}

}

] )

1. db.posts.aggregate( [

{

$group: {

\_id: "$author",

count: { $sum: 1 }

}

},

{

$match: { count: { $gt: 1 } }

}

] )

1. db.posts.aggregate( [

{

$match: { count: { $gt: 1 } } },

{

$group: {

\_id: "$author",

count: { $sum: 1 }

}

}

] )

1. db.posts.aggregate( [

{

$group: {

\_id: "$author",

count: { $sum: "$author" }

}

},

{

$match: { count: { $gt: 1 } }

}

] )

**Which of the tags in a replica set configuration specify the operations to be read from the node with the least network latency?**

1. primaryPreferred
2. secondaryPreferred
3. nearest
4. netLatency

**Consider the following document from the products collection:**

**{**

**\_id: 1,**

**product\_code: "345678",**

**variations: [**

**{ size: "L", price: 1000 },**

**{ size: "M", price: 800 }**

**]**

**}**

**What does the following query using $elemMatch return?**

**db.products.find( { product\_code: "345678" },**

**{ variations: { $elemMatch: { size: "L" } }**

**} )**

A - Returns the complete document since MongoDB does not support partial array retrieval

B - Returns the document but with only one element in the variations array (corresponding to size L)

C - Returns the complete document but retrieves only the size field from the array

D - Returns the complete document but retrieves only the size field from the array and also with only one element in the

variations array (corresponding to size L)

Which of the following commands removes a single document that matches the condition that Author is Joe?

A - db.posts.removeOne( { Author : "Joe" }, 1 )

B - db.posts.remove( { Author : "Joe" }, 1 )

C - db.posts.remove( { Author : "Joe" }, {justOne: true} )

D - Both b and c

Which is the correct order (lowest to highest) in which MongoDB compares the BSON types?

A - Null, Number, String and Object

B - Number, Null, String and Object

C - String, Null, Number and Object

D - Null, Number, Object and String

What is the minimum sensible number of voting nodes to a replica set?

A - 2

B - 3

C - 4

D - 5

Given a collection posts as shown below having a document array comments, which of the following command will create

an index on the comment author descending?

{

"\_id":1,

"post\_text":"This is a sample post",

"author":"Tom",

"comments":[

{

"author":"Joe",

"comment\_text":"This is comment 1"

},

{

"author":"Leo",

"comment\_text":"This is comment 2"

}

]

}

A - db.posts.createIndex({"comments.$.author":-1});

B - db.posts.createIndex({"comments.author":-1});

C - db.posts.createIndex({"comments.author":1});

D - db.posts.createIndex({"comments.$.author": {"$desc":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