1. How does MongoDB store data?

- A. DOCUMENTS AND COLUMNS
- B. Documents and Tables
- C. Documents and Rows
- D. Documents and Collections

2 . Consider a "grades" collection that contains these documents:

{ _id: 1, student: "Jamie", scores: [72, 90, 87] }

{ _id: 2, student: "Avery", scores: [98, 67, 64] }

{ _id: 3, student: "Madison", scores: [81, 82, 82] }

{ _id: 4, student: "Lennox", scores: [73, 91, 64] }

What documents does the following query return?

db.grades.find(

{ scores: { $elemMatch: { $gte: 70, $lte: 80 } } }

)

- A. [{ _id: 1, student: 'Jamie', scores: [72, 90, 87] }, { _id: 4, student: 'Lennox', scores: [73, 91, 64] }]
- B. { _id: 4, student: 'Lennox', scores: [73, 91, 64] }
- C. [{ _id: 1, student: 'Jamie', scores: [72, 90, 87] }, { _id: 2, student: 'Avery', scores: [98, 67, 64] } { _id: 4, student: 'Lennox', scores: [73, 91, 64] }]
- D. [{ _id: 3, student: "Madison", scores: [81, 82, 82] } { _id: 4, student: "Lennox", scores: [73, 91, 64] }]

3. Consider a "testScores" collection that contains these documents:

{"_id": 1, "student": "ian", "score": 80}

{"_id": 2, "student": "sarah", "score": 79}

{"_id": 3, "student": "nick", "score": 95}

{"_id": 4, "student": "joseph", "score": 80}

{"_id": 5, "student": "ashley", "score": 90}

{"_id": 6, "student": "dave", "score": 82}

{"_id": 7, "student": "jason", "score": 81}

{"_id": 8, "student": "jocelyn", "score": 98}

{"_id": 9, "student": "ali", "score": 95}

What set of documents does this query return?

db.testScores.find({}).sort({"score": 1, "student": 1}).limit(3)

- A. [{ _id: 8, student: 'jocelyn', score: 98 }, { _id: 3, student: 'nick', score: 95 }, { _id: 9, student: 'ali', score: 95 }]
- B. [{ _ID: 8, STUDENT: 'JOCELYN', SCORE: 98 }, { _ID: 9, STUDENT: 'ALI', SCORE: 95 }, { _ID: 3, STUDENT: 'NICK', SCORE: 95 }]
- C. [{ _id: 2, student: 'sarah', score: 79 }, { _id: 1, student: 'ian', score: 80 }, { _id: 4, student: 'joseph', score: 80 }]
- D. [{ _ID: 2, STUDENT: 'SARAH', SCORE: 79 }, { _ID: 4, STUDENT: 'JOSEPH', SCORE: 80 }, { _ID: 1, STUDENT: 'IAN', SCORE: 80 }]

4. Which expression returns a count of documents in an "orders" collection that match {"state": "NY"}?

- A. db.orders.getCount({"state": "NY"})
- B. db.orders.countDocs({"state": "NY"})
- C. db.orders.getDocumentCount({"state": "NY"})
- D. db.orders.countDocuments({"state": "NY"})

5. Consider a "pizzaToppings" collection. All documents in this collection have the same schema.

{

"_id" : 1,

"topping" : "mushrooms",

"unitsInStock" : 15

}

What operation updates the value of the unitsInStock field to 25 for a document with an "onions" topping value and inserts a new document if it does not exist?

- A. db.pizzaToppings.updateOne({ topping: "onions" } , { $set: { unitsInStock: 25 }, upsert: true })
- B. db.pizzaToppings.insertOne({ topping: "onions" } , { $set: { unitsInStock: 25 }, upsert: true })
- C. DB.PIZZATOPPINGS.INSERTONE({ TOPPING: "ONIONS" } , { $SET: { UNITSINSTOCK: 25 } }, { UPSERT: TRUE })
- D. db.pizzaToppings.updateOne({ topping: "onions" } , { $set: { unitsInStock: 25 } }, { upsert: true })

6. A furniture store stopped selling lamps and needs to delete lamps from the "furniture" collection.

What command deletes all documents from the "furniture" collection where the "type" field is equal to "lamp"?

- A. db.furniture.deleteMany({ "type": "lamp" })
- B. DB.FURNITURE.REMOVEMANY({ "TYPE": "LAMP" })
- C. db.furniture.deleteAll({ "type": "lamp" })
- D. db.furniture.delete({ "type": "lamp" }, { "multi": "true" })

7. A museum maintains an "exhibits" collection with these documents:

{ "_id": 1, "exhibit": "Food in New York", "floors": [2, 3] }

{ "_id": 2, "exhibit": "Starlight", "floors": [5, 6] }

{ "_id": 3, "exhibit": "Reginald Marsh", "floors": [3, 4, 5] }

Each document in the "exhibits" collection contains a "floors" field that indicates the floors of the museum where the exhibit takes place.

What is the output of the following query:

db.exhibits.find({ floors: 5 })

- A. [{ _id: 2, exhibit: 'Starlight', floors: [5, 6] }, { _id: 3, exhibit: 'Reginald Marsh', floors: [3, 4, 5] }]
- B. { _ID: 2, EXHIBIT: 'STARLIGHT', FLOORS: [5, 6] }
- C. { "_ID": 3, "EXHIBIT": "REGINALD MARSH", "FLOORS": [3, 4, 5] }
- D. [{ _ID: 1, EXHIBIT: 'FOOD IN NEW YORK', FLOORS: [2, 3] }, { _ID: 2, EXHIBIT: 'STARLIGHT', FLOORS: [5, 6] }, { _ID: 3, EXHIBIT: 'REGINALD MARSH', FLOORS: [3, 4, 5] }]

8. An administrator needs to see what indexes exist on the "orders" collection.

What command lists the indexes in the "orders" collection?

- A. db.orders.getIndexes()
- B. DB.ORDERS.LISTINDEXES()
- C. DB.ORDERS.GETALLINDEXES()
- D. DB.ORDERS.INDEXES()

9. An shard collection command ends with an error:

sh.shardCollection("test.shardedColl", {shardKey: 1})

MongoServerError: couldn't find valid index for shard key

What is the cause of this error?

- A. The field shardKey is indexed with a multikey index.
- B. THE COLLECTION SHARDCOLL CONTAINS NO DATA AND NO INDEXES.
- C. THERE IS NO INDEX SHARDKEY.
- D. THE HELPER SH.SHARDCOLLECTION DOES NOT EXIST.

10. What command shows if a query of db.load.find({ rnd: 1 }) in the profiling collection was supported by an index?

- A. db.system.profile.find({"command.filter": {rnd: 1}, planSummary: /IXSCAN/})
- B. db.system.profile.find({find: {rnd: 1}, planSummary: /IXSCAN/})
- C. db.system.profile.find({find: {rnd: 1}, planSummary: "IXSCAN"})
- D. DB.SYSTEM.PROFILE.FIND({"COMMAND.FILTER": {RND: 1}, PLANSUMMARY: "IXSCAN"})

11. Given the following query:

db.load.find({rnd: 4321, _id: {$gte: 10000, $lte: 11000}, date: {$gt: ISODate("2022-12-31"), $lt: ISODate("2023-02-01")}}).sort({date: -1})

In a compound index that best supports this query, what field should be first field?

- A. rnd: 1
- B. date: 1
- C. _id: 1

- D. NO FIELD

12. Which two mongosh commands can you use to determine if mongod is running on the default port of your system?

- A. ``` mongosh --quiet --port 27017 --eval "db.serverStatus()" ```
- B. ``` mongosh --quiet --port 20717 --eval "db.serverStatus()" ```
- C. ``` mongosh --quiet --port 1433 --eval "db.serverStatus()" ```
- D. ``` MONGOSH --QUIET --EVAL "DB.SERVERSTATUS()" ```

13. Which two mongod command line options affects where data files are stored?

- A. ``` --storageEngine ```
- B. ``` --DIRECTORYPERDB ```
- C. ``` --dbpath ```
- D. ``` --LOGPATH ```

14. Which two mongosh commands connect to node-1.cluster1.mycompany.local on port 28412?

- A. mongosh "node-1.cluster1.mycompany.local:28412"
- B. MONGOSH --HOST "NODE-1.CLUSTER1.MYCOMPANY.LOCAL&PORT=28412"
- C. MONGOSH --HOST "NODE-1.CLUSTER1.MYCOMPANY.LOCAL" --PORT 28412
- D. MONGOSH --HOSTNAME "NODE-1.CLUSTER1.MYCOMPANY.LOCAL&PORT=28412"

15. What command returns in-progress operations that have been running for more than 3 seconds?

- A. db.currentOp({ "active" : true, "secs_running" : { "$gt" : 3 } })
- B. DB.LISTOPERATIONS({ "ACTIVE" : TRUE, "SECS_RUNNING" : { "$GT" : 3 } })
- C. db.top({ "active" : true, "secs_running" : { "$gt" : 3 } })
- D. db.serverStatus({ "active" : true, "secs_running" : { "$gt" : 3 } })

16. What two formats is the audit log written in?

- A. JSON

- B. TEXT

- C. BSON

- D. CSV

17. What is a benefit of enabling TLS?

- A. Enhancing database access control.
- B. REDUCING DATABASE LATENCY.
- C. SIMPLIFYING DATABASE CONFIGURATION.
- D. ENABLING ADVANCED DATA ANALYTICS.

18. Which benefit is due to enabling authentication?

- A. Enhancing database access control.
- B. Enabling advanced data analytics.
- C. ENABLING FASTER DATA PROCESSING.
- D. SIMPLIFYING DATABASE CONFIGURATION.

19. What is a benefit of enabling encryption at rest?

- A. Providing an additional layer of protection against unauthorized access.
- B. To minimize the downtime and disruption of the database maintenance.
- C. TO IMPROVE THE DATABASE PERFORMANCE BY LOWERING THE I/O LATENCY.
- D. TO MAINTAIN THE RELIABILITY AND RESILIENCE OF THE DATABASE SYSTEM.

20. What is a benefit of using field-level encryption?

- A. Field-level encryption provides granular control over data access.
- B. FIELD-LEVEL ENCRYPTION INCREASES THE RISK OF DATA BREACHES CAUSED BY MALWARE ATTACKS.
- C. FIELD-LEVEL ENCRYPTION ENABLES FASTER DATA PROCESSING AND ANALYSIS.
- D. FIELD-LEVEL ENCRYPTION SIMPLIFIES DATA MANAGEMENT BY AUTOMATICALLY CATEGORIZING DATA BASED ON ITS SENSITIVITY.

21. How would a database administrator make their MongoDB system highly available?

- A. Sharding

- B. Replication
- C. INDEXES
- D. CACHE

22. A _____ maintains a copy of the primary's data set.

- A. Secondary
- B. Node
- C. MEMBER
- D. MONGOS

23. What setting specifies the number of replica set members that must complete a write operation before the operation is acknowledged as successful?

- A. Write preference
- B. READ CONCERN
- C. Write concern
- D. READ PREFERENCE

24. What flag does the administrator use with mongodump during the backup process to capture operations for point-in-time recovery in a MongoDB replica set?

- A. mongodump --archive
- B. mongodump --oplog
- C. MONGODUMP --NUMPARALLELCOLLECTIONS
- D. mongodump --readPreference