MongoDB Associate Developer

Practice Quiz — Set 2: 30 New Mixed Questions

Instructions: Answer all 30 questions first. The answer key with explanations is on the last pages — don't peek until you're done. This is a fresh set, no overlap with Set 1. Good luck tomorrow!

1. MongoDB Basics & BSON

Q1. Which of these is NOT a valid BSON data type?

A) Decimal128
B) ObjectId
C) Timestamp
D) Set

Q2. What does BSON stand for, and why does MongoDB use it instead of plain JSON internally?

A) Binary JSON; it supports additional data types (e.g. Date, ObjectId, binary data) and is efficient to traverse
B) Basic JSON; it's simpler to parse than JSON
C) Byte-Stream Object Notation; used only for network transport, never storage
D) Big Structured Object Notation; supports only numbers and strings

Q3. A field is defined as { createdAt: ISODate("2026-07-25T10:00:00Z") }. What BSON type is this?

A) String
B) Date
C) Timestamp
D) Int64

2. CRUD Operations

Q4. What does db.collection.deleteMany({}) do?

A) Deletes the entire collection, including its indexes
B) Deletes all documents in the collection but keeps the collection and its indexes
C) Does nothing, since the filter is empty
D) Throws an error requiring a non-empty filter

Q5. Which option, when set to true on updateOne(), inserts a new document if no document matches the filter?

A) multi
B) upsert
C) insertIfMissing
D) create

Q6. What happens when you use $inc on a field that doesn't yet exist on the target document?

A) It throws an error since the field must already exist

B) It creates the field and sets it to the increment value
C) It silently ignores the update
D) It sets the field to null

Q7. Which method atomically finds a matching document and applies an update in one operation, useful for implementing things like counters or queues?

A) updateOne()
B) findOneAndUpdate()
C) find() followed by a separate updateOne()
D) bulkWrite()

3. Aggregation Framework

Q8. Which stage caps the number of documents passed to the rest of the pipeline, e.g. returning only the top 10 after a sort?

A) $skip
B) $limit
C) $sample
D) $facet

Q9. What does the $lookup stage do in an aggregation pipeline?

A) Performs a left outer join against another collection
B) Deletes documents matching a condition
C) Groups documents by a computed key
D) Validates schema before insertion

Q10. Given { $group: { _id: null, total: { $sum: "$amount" } } }, what does _id: null achieve?

A) It causes an error, since _id can't be null
B) It groups all input documents into a single group, producing one aggregate result
C) It groups documents by their own null-valued fields
D) It excludes documents where amount is null

Q11. In a $project stage, what does { total: { $multiply: ["$price", "$qty"] } } do?

A) Creates a new computed field "total" equal to price multiplied by qty
B) Filters out documents where price times qty exceeds a threshold
C) Renames the price field to total
D) Throws an error, since $project can't compute new fields

Q12. Which stage lets you run multiple aggregation sub-pipelines against the same input documents and return their results together, e.g. a count plus a paginated list in one query?

A) $facet
B) $bucket
C) $merge
D) $out

4. Indexes

Q13. What type of index does MongoDB automatically create when you index a field that holds array values?

A) Sparse index
B) Multikey index
C) Partial index
D) Wildcard index

Q14. What does a unique index enforce?

A) That the field must always be a string
B) That no two documents can share the same value for the indexed field(s)
C) That the field must be indexed in ascending order only
D) That the field can never be null

Q15. What is a "covered query"?

A) A query that uses a text index
B) A query where every field it needs is present in the index itself, so MongoDB never has to fetch the full document
C) A query wrapped inside a transaction
D) A query that scans the entire collection

Q16. Which index type automatically removes documents after a set amount of time, useful for things like session data or logs?

A) TTL index
B) Text index
C) Hashed index
D) Geospatial index

5. Data Modeling

Q17. What is the "extended reference" (or "subset") pattern used for?

A) Duplicating a small subset of frequently-needed fields from a referenced document into the referencing document, to avoid extra lookups
B) Storing the entire referenced document twice for redundancy
C) Creating a second database purely for backups
D) Referencing documents across separate MongoDB clusters

Q18. In a many-to-many relationship (e.g. students and courses), what's a common modeling approach?

A) Embed every course inside every student document and vice versa
B) Maintain reference arrays on both sides, or use a separate junction/mapping collection
C) Store all students and courses in a single giant document
D) MongoDB does not support many-to-many relationships

Q19. Why might you deliberately duplicate (denormalize) some data across documents, despite the risk of inconsistency?

A) It's required by MongoDB's storage engine
B) To optimize read performance and avoid frequent joins ($lookup) for data that's accessed together often
C) Duplication is never recommended in MongoDB
D) It reduces the maximum allowed document size

6. Replication

Q20. What is the role of an arbiter in a replica set?

A) It holds a full copy of the data and can become primary
B) It votes in elections but holds no data and can never become primary
C) It exists only to run backups
D) It replaces the need for secondaries

Q21. What does "read preference" control in a MongoDB driver?

A) Which replica set member(s) reads are routed to (e.g. primary, secondary, nearest)
B) How much data is cached client-side
C) Whether documents are returned as JSON or BSON
D) The write concern level

Q22. If a write is acknowledged with write concern { w: "majority" }, what does that guarantee?

A) The write only reached the primary and hasn't replicated yet
B) The write has been propagated to and acknowledged by a majority of voting replica set members
C) The write is guaranteed to eventually reach all members, with no timing guarantee
D) Write concern has no effect on durability

7. Sharding

Q23. What is stored on config servers in a sharded cluster?

A) The actual application data
B) Metadata about the cluster, including chunk ranges and shard locations
C) Only user authentication credentials
D) Backup copies of every shard

Q24. What is a "chunk" in the context of MongoDB sharding?

A) A physical server in the cluster
B) A contiguous range of shard key values, used as the unit of data distribution/migration between shards
C) A single document
D) A backup snapshot

Q25. When would sharding generally NOT be necessary?

A) When your dataset and working set comfortably fit on a single server and you don't need horizontal write scaling
B) When you have more than 1GB of data
C) Whenever you're using a replica set
D) Sharding is always required in production

8. Security

Q26. What authentication mechanism does MongoDB use by default for username/password authentication?

A) SCRAM (Salted Challenge Response Authentication Mechanism)
B) Basic HTTP auth

C) OAuth2
D) Kerberos only

Q27. What is the "principle of least privilege" as applied to MongoDB roles?

A) Granting every user the root role for simplicity
B) Granting users only the specific roles/permissions they need to do their job, nothing more
C) Disabling authentication for internal users
D) Giving all users readWriteAnyDatabase by default

Q28. Which statement is true about network-level security for a MongoDB Atlas cluster?

A) By default, Atlas clusters accept connections from any IP address with no restriction
B) IP access lists (allowlists) restrict which IP addresses are allowed to connect to the cluster
C) Network security isn't configurable on Atlas
D) TLS is optional and disabled by default on Atlas

9. Drivers & Tooling (pymongo)

Q29. In pymongo, what is client.admin.command("ping") typically used for?

A) Deleting the admin database
B) Verifying that the connection to the server is alive
C) Creating a new admin user
D) Listing all databases

Q30. Which pymongo exception is commonly raised when inserting a document whose _id (or a uniquely-indexed field) already exists?

A) pymongo.errors.ConnectionFailure
B) pymongo.errors.DuplicateKeyError
C) pymongo.errors.OperationFailure only
D) ValueError

Answer Key & Explanations

1. MongoDB Basics & BSON

Q1. Correct answer: **D) Set**

MongoDB has no native "Set" BSON type. Decimal128, ObjectId, and (internal) Timestamp are all real BSON types.

Q2. Correct answer: **A) Binary JSON; it supports additional data types (e.g. Date, ObjectId, binary data) and is efficient to traverse**

BSON = Binary JSON. It adds types JSON lacks (Date, ObjectId, Binary, Decimal128, etc.) and is designed for fast encode/decode and traversal.

Q3. Correct answer: **B) Date**

This is BSON type Date. BSON Timestamp is a distinct internal type mainly used for the oplog, not for application-level date fields.

2. CRUD Operations

Q4. Correct answer: **B) Deletes all documents in the collection but keeps the collection and its indexes**

An empty filter {} matches every document, so deleteMany({}) removes all documents while the collection and its index definitions remain.

Q5. Correct answer: **B) upsert**

upsert: true tells MongoDB to insert a new document (built from the filter + update) if no document matches.

Q6. Correct answer: **B) It creates the field and sets it to the increment value**

$inc creates the field if missing, initializing it to the increment amount (e.g. $inc: {count: 1} on a missing field sets count to 1).

Q7. Correct answer: **B) findOneAndUpdate()**

findOneAndUpdate() performs the find-and-modify as a single atomic server-side operation, avoiding race conditions between a separate read and write.

3. Aggregation Framework

Q8. Correct answer: **B) $limit**

$limit restricts the pipeline to a maximum number of documents, commonly used right after $sort for a top-N query.

Q9. Correct answer: **A) Performs a left outer join against another collection**

$lookup joins documents from another collection into the current pipeline, similar to a SQL LEFT OUTER JOIN, based on matching field values.

Q10. Correct answer: **B) It groups all input documents into a single group, producing one aggregate result**

Using _id: null collapses every document into one group, useful for computing an overall total/count/average across the whole collection (or preceding pipeline stage).

Q11. Correct answer: **A) Creates a new computed field "total" equal to price multiplied by qty**

$project (and $addFields) can include computed expressions, not just field selection — here it multiplies two existing fields into a new one.

Q12. Correct answer: **A) $facet**

$facet processes the same set of input documents through several independent sub-pipelines and returns all results in a single document.

4. Indexes

Q13. Correct answer: **B) Multikey index**

Indexing an array field automatically produces a multikey index, with one index entry per array element.

Q14. Correct answer: **B) That no two documents can share the same value for the indexed field(s)**

A unique index rejects inserts/updates that would create a duplicate value for the indexed field(s) (with a caveat: multiple documents can still have a missing field, since MongoDB treats it as one null-value collision only if the field is actually set).

Q15. Correct answer: **B) A query where every field it needs is present in the index itself, so MongoDB never has to fetch the full document**

A covered query is answered entirely from the index (including the projection), avoiding a fetch of the underlying document — this is faster than a normal index scan.

Q16. Correct answer: **A) TTL index**

A TTL (time-to-live) index expires and deletes documents automatically once a date field passes a configured age threshold.

5. Data Modeling

Q17. Correct answer: **A) Duplicating a small subset of frequently-needed fields from a referenced document into the referencing document, to avoid extra lookups**

This pattern copies a few commonly-needed fields (e.g. a product's name and price) into the referencing document so common reads don't require a $lookup, at the cost of needing to keep the duplicated fields in sync.

Q18. Correct answer: **B) Maintain reference arrays on both sides, or use a separate junction/mapping collection**

Many-to-many relationships are typically modeled with reference arrays on each side, or a dedicated junction collection tracking the pairings, similar to a relational join table.

Q19. Correct answer: **B) To optimize read performance and avoid frequent joins ($lookup) for data that's accessed together often**

Denormalization trades some write complexity (keeping duplicated data in sync) for significantly faster, simpler reads — a common and accepted MongoDB modeling tradeoff.

6. Replication

Q20. Correct answer: **B) It votes in elections but holds no data and can never become primary**

An arbiter participates in elections to help reach a majority vote but stores no data itself and is never eligible to become primary.

Q21. Correct answer: **A) Which replica set member(s) reads are routed to (e.g. primary, secondary, nearest)**

Read preference (primary, primaryPreferred, secondary, secondaryPreferred, nearest) determines which replica set member(s) a read operation can be served from.

Q22. Correct answer: **B) The write has been propagated to and acknowledged by a majority of voting replica set members**

w: "majority" waits for acknowledgment from a majority of voting members before confirming the write as successful, providing stronger durability than the default in some configurations.

7. Sharding

Q23. Correct answer: **B) Metadata about the cluster, including chunk ranges and shard locations**

Config servers store the cluster's metadata — which chunks (ranges of shard key values) live on which shards — used by mongos to route operations.

Q24. Correct answer: **B) A contiguous range of shard key values, used as the unit of data distribution/migration between shards**

A chunk is a range of shard key values; MongoDB balances the cluster by migrating chunks between shards to keep data distribution even.

Q25. Correct answer: **A) When your dataset and working set comfortably fit on a single server and you don't need horizontal write scaling**

Sharding solves horizontal scale problems (dataset/throughput beyond one server's capacity); if a single server (or replica set) handles the load fine, sharding just adds operational complexity without benefit.

8. Security

Q26. Correct answer: **A) SCRAM (Salted Challenge Response Authentication Mechanism)**

SCRAM is MongoDB's default challenge-response authentication mechanism for username/password logins.

Q27. Correct answer: **B) Granting users only the specific roles/permissions they need to do their job, nothing more**

Least privilege means scoping each user's roles as narrowly as possible to reduce security risk, rather than defaulting to broad admin/superuser roles.

Q28. Correct answer: **B) IP access lists (allowlists) restrict which IP addresses are allowed to connect to the cluster**

Atlas requires you to configure an IP access list before any client can connect — nothing connects by default, and TLS is enforced, not optional.

9. Drivers & Tooling (pymongo)

Q29. Correct answer: **B) Verifying that the connection to the server is alive**

Running the "ping" command against the admin database is the standard lightweight way to confirm connectivity to a MongoDB deployment.

Q30. Correct answer: **B) pymongo.errors.DuplicateKeyError**

DuplicateKeyError is raised specifically when an insert/update would violate a unique index (including the default unique index on _id).