

MongoDB Associate Developer

Practice Quiz — 30 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. Good luck tomorrow!

1. MongoDB Basics & BSON

Q1. What is the size of an ObjectId in bytes?

- A) 8
- B) 12
- C) 16
- D) 24

Q2. Which BSON type should you use for a monetary field like \$19.99 that needs exact decimal precision (no floating-point rounding)?

- A) Double
- B) Int32
- C) Decimal128
- D) String

Q3. Which statement about MongoDB collections is true?

- A) A single collection can contain documents with different sets of fields, since MongoDB is schema-flexible by default
- B) All documents in a collection must have identical fields
- C) Schema flexibility only applies to arrays
- D) A collection can only ever hold one schema, enforced automatically

2. CRUD Operations

Q4. Which method updates only specific fields of a single matching document without replacing the whole document?

- A) replaceOne()
- B) updateOne()
- C) findOneAndDelete()
- D) insertOne()

Q5. What happens if you call updateOne() with a plain document that contains no update operators (no \$set, \$inc, etc.)?

- A) It silently replaces the whole document
- B) It throws an error
- C) It merges fields automatically
- D) It inserts a new document

Q6. What does the projection {name: 1, _id: 0} return in a find() query?

- A) Only the name field, with _id excluded
- B) All fields except name

- C) Only the `_id` field
- D) This is always an invalid projection

Q7. Which update operator adds a value to an array only if it does not already exist in that array?

- A) `$push`
- B) `$addToSet`
- C) `$set`
- D) `$pull`

3. Aggregation Framework

Q8. Which pipeline stage should generally be placed as early as possible to improve performance?

- A) `$project`
- B) `$match`
- C) `$sort`
- D) `$limit`

Q9. What does the `_id` field represent inside a `$group` stage?

- A) An auto-generated identifier for the output document
- B) The grouping key expression
- C) The first document's `_id` in each group
- D) It is always null

Q10. Given `{ scores: [80, 90, 100] }`, what does `$unwind: "$scores"` produce?

- A) A single averaged number
- B) One output document per array element
- C) The array field removed entirely
- D) The array sorted in place

Q11. By default, what happens to a document during `$unwind` if its array field is missing, null, or empty (`[]`)?

- A) It's kept with a null value
- B) It throws an error
- C) It is dropped from the output
- D) It converts the field to an empty string

Q12. What is the key difference between `$project` and `$addFields` when reshaping documents?

- A) `$project` can only exclude fields; `$addFields` can only include
- B) `$addFields` keeps all existing fields and adds new ones; `$project` (inclusion mode) drops any field not explicitly listed
- C) They are functionally identical
- D) `$addFields` requires a preceding `$group` stage

4. Indexes

Q13. What index exists on every MongoDB collection by default?

- A) A text index on all string fields

- B) A unique index on `_id`
- C) A compound index across all fields
- D) No index exists by default

Q14. What does the ESR rule recommend for ordering fields in a compound index?

- A) Equality, Sort, Range
- B) Exclusion, Sort, Range
- C) Equality, Selectivity, Range
- D) Exclusion, Selectivity, Reference

Q15. Given a compound index `{status: 1, date: -1}`, which query uses it most efficiently?

- A) `db.col.find({date: {$gte: ISODate(...)}})`
- B) `db.col.find({status: "active"}).sort({date: -1})`
- C) `db.col.find({}).sort({date: -1})`
- D) `db.col.find({randomField: "x"})`

Q16. What does `explain()` showing `COLLSCAN` indicate?

- A) The query used an index efficiently
- B) The query performed a full collection scan without using an index
- C) The query failed to execute
- D) The query used a covered index

5. Data Modeling

Q17. For a one-to-few relationship where child data is always accessed together with the parent and won't grow unbounded, what's generally recommended?

- A) Reference the children in a separate collection
- B) Embed the child data directly in the parent document
- C) Use a separate join/junction collection
- D) Store the children in a different database

Q18. For a relationship where children could grow to thousands of records (e.g., comments on a popular post) and need independent access, what's recommended?

- A) Embed all children in an array inside the parent
- B) Reference: store the parent's `_id` on each child document
- C) Duplicate the parent's data into every child document instead of referencing
- D) Store all children as one large concatenated string field

Q19. What is the hard limit on the size of a single BSON document in MongoDB?

- A) 4MB
- B) 8MB
- C) 16MB
- D) No limit

6. Replication

Q20. Why does MongoDB recommend an odd number of voting members in a replica set?

- A) To reduce network latency
- B) To avoid tie votes and ensure a majority can be reached during elections

- C) Odd numbers use less disk space
- D) It's a storage engine requirement

Q21. What happens to write operations during the window when a replica set temporarily has no primary (mid-election)?

- A) Writes succeed but are queued for later
- B) Writes fail until a new primary is elected
- C) Writes are automatically redirected to a secondary
- D) Writes succeed with eventual consistency

Q22. By default, which replica set member(s) can accept write operations?

- A) Any secondary
- B) Only the primary
- C) All members simultaneously
- D) Only the member with the most storage

7. Sharding

Q23. What is the main risk of using a monotonically increasing field (e.g., a timestamp or default ObjectId) alone as a shard key with range-based sharding?

- A) Data becomes unreadable
- B) It creates a write hotspot on a single shard
- C) It causes permanent data loss
- D) It disables replication

Q24. Which sharding strategy distributes writes evenly across shards by hashing the shard key value, at the cost of inefficient range queries?

- A) Range-based sharding
- B) Hashed sharding
- C) Zone sharding
- D) Manual sharding

Q25. What component do client applications connect to in a sharded cluster to route queries to the correct shard(s)?

- A) Config server
- B) mongos
- C) The primary shard directly
- D) Arbiter

8. Security

Q26. Which built-in role provides full read and write access to data (including deletes), but not user/role administration?

- A) dbAdmin
- B) readWrite
- C) userAdmin
- D) clusterAdmin

Q27. Which role combines readWriteAnyDatabase + dbAdminAnyDatabase + userAdminAnyDatabase + clusterAdmin into one superuser role?

- A) dbOwner
- B) root
- C) clusterManager
- D) backup

Q28. A user's credentials live in the "admin" database, but they're granted a role scoped to the "sales" database. Is this valid?

- A) No, users can only be granted roles in their home database
- B) Yes, a user's roles array can include roles scoped to other databases
- C) Only if the user is a root user
- D) Only during replica set elections

9. Drivers & Tooling (pymongo)

Q29. In pymongo, calling `collection.find_one({"_id": "507f1f77bcf86cd799439011"})` with a plain string instead of `ObjectId(...)` will:

- A) Successfully find the document, since MongoDB auto-converts strings
- B) Return None / no match, because BSON String and BSON ObjectId are different types
- C) Raise a `TypeError` immediately
- D) Permanently convert the field to a string

Q30. What does `collection.find()` return in pymongo, and how should a large result set be consumed?

- A) A single dictionary; access fields directly
- B) A cursor; iterate lazily (e.g. with a for loop) instead of loading everything into memory at once
- C) A JSON string; parse with `json.loads()`
- D) A pandas DataFrame automatically

Answer Key & Explanations

1. MongoDB Basics & BSON

Q1. Correct answer: **B) 12**

ObjectId = 4-byte timestamp + 5-byte random value + 3-byte counter = 12 bytes total.

Q2. Correct answer: **C) Decimal128**

Decimal128 avoids the rounding errors of binary floating point (Double), making it correct for currency.

Q3. Correct answer: **A) A single collection can contain documents with different sets of fields, since MongoDB is schema-flexible by default**

MongoDB collections are schema-flexible by default; documents can vary in structure unless schema validation rules are added.

2. CRUD Operations

Q4. Correct answer: **B) updateOne()**

updateOne() applies update operators (like \$set) to modify specific fields; replaceOne() swaps the entire document.

Q5. Correct answer: **B) It throws an error**

updateOne() requires update operators; passing a plain replacement-style document raises an error.

Q6. Correct answer: **A) Only the name field, with _id excluded**

Inclusion (1) and exclusion (0) can't normally be mixed, except _id, which can always be explicitly excluded alongside inclusion projections.

Q7. Correct answer: **B) \$addToSet**

\$addToSet prevents duplicate values; \$push always appends, even if the value is already present.

3. Aggregation Framework

Q8. Correct answer: **B) \$match**

Placing \$match early reduces the number of documents flowing through the rest of the pipeline.

Q9. Correct answer: **B) The grouping key expression**

_id in \$group defines what you're grouping by — it can be a field, an expression, or an embedded document combining multiple fields.

Q10. Correct answer: **B) One output document per array element**

\$unwind deconstructs the array, emitting one document per element while copying the rest of the document's fields.

Q11. Correct answer: **C) It is dropped from the output**

Without `preserveNullAndEmptyArrays: true`, `$unwind` silently drops documents whose array field is empty/missing/null.

Q12. Correct answer: **B) \$addField keeps all existing fields and adds new ones; \$project (inclusion mode) drops any field not explicitly listed**

`$addField` (alias `$set`) is additive and preserves the rest of the document; `$project` in inclusion mode reshapes the document down to only the listed fields.

4. Indexes

Q13. Correct answer: **B) A unique index on _id**

MongoDB automatically creates a unique index on `_id` for every collection.

Q14. Correct answer: **A) Equality, Sort, Range**

ESR = Equality fields first, then Sort fields, then Range fields, for the most efficient compound index usage.

Q15. Correct answer: **B) `db.col.find({status: "active"}).sort({date: -1})`**

This matches the ESR pattern exactly: equality on status, then sort on date, both covered by the compound index's field order.

Q16. Correct answer: **B) The query performed a full collection scan without using an index**

`COLLSCAN` means MongoDB scanned every document in the collection; `IXSCAN` would indicate index usage instead.

5. Data Modeling

Q17. Correct answer: **B) Embed the child data directly in the parent document**

Embedding is ideal when data is small, bounded, and typically read together with the parent.

Q18. Correct answer: **B) Reference: store the parent's _id on each child document**

Unbounded, independently-accessed children should be referenced (child holds parent's `_id`), avoiding unbounded array growth and write contention on the parent.

Q19. Correct answer: **C) 16MB**

A single BSON document cannot exceed 16MB, which is one of the key reasons to avoid unbounded embedded arrays.

6. Replication

Q20. Correct answer: **B) To avoid tie votes and ensure a majority can be reached during elections**

An odd number of voters prevents a 50/50 tie, guaranteeing a majority can always be established during a primary election.

Q21. Correct answer: **B) Writes fail until a new primary is elected**

Only the primary accepts writes; with no primary present, all write operations fail until the election completes.

Q22. Correct answer: **B) Only the primary**

Writes go to the primary only by default; secondaries are read-only replicas unless configured otherwise for reads.

7. Sharding

Q23. Correct answer: **B) It creates a write hotspot on a single shard**

New writes with ever-increasing values all land in the same chunk/shard, creating a hotspot while other shards sit idle.

Q24. Correct answer: **B) Hashed sharding**

Hashed sharding scatters values evenly (avoiding hotspots) but scatters related range values across many shards, hurting range query performance.

Q25. Correct answer: **B) mongos**

mongos is the query router that clients connect to; it consults the config servers to route operations to the correct shard(s).

8. Security

Q26. Correct answer: **B) readWrite**

readWrite covers read, insert, update, and delete (remove) operations on data; it does not grant user or role management.

Q27. Correct answer: **B) root**

root is the cluster-wide superuser role, granting full control over all data, all admin tasks, all users, and cluster operations.

Q28. Correct answer: **B) Yes, a user's roles array can include roles scoped to other databases**

A user's home database only determines where their credentials are stored; their roles array can reference roles on any database.

9. Drivers & Tooling (pymongo)

Q29. Correct answer: **B) Return None / no match, because BSON String and BSON ObjectId are different types**

_id is stored as BSON ObjectId; a Python string of the same characters is BSON type String, so it won't match without wrapping in ObjectId(...).

Q30. Correct answer: **B) A cursor; iterate lazily (e.g. with a for loop) instead of loading everything into memory at once**

find() returns a cursor that fetches matching documents from the server lazily in batches as you iterate, which scales to large result sets.