

# Question 1

**In MongoDB, what is a collection most similar to in MySQL?**

- 1) A column
- 2) A table
- 3) A row
- 4) A database server

# Answer 1

**In MongoDB, what is a collection most similar to in MySQL?**

- 1) A column
- 2) A table**
- 3) A row
- 4) A database server

**Correct answer: 2**

Why: A collection stores many documents, similar to how a table stores many rows.

## Question 2

**What is MongoDB Compass mainly used for?**

- 1) A command to insert documents
- 2) A cloud hosting service for MongoDB
- 3) A GUI tool to view and work with databases and collections
- 4) Writing Java code to connect to MongoDB

## Answer 2

### What is MongoDB Compass mainly used for?

- 1) A command to insert documents
- 2) A cloud hosting service for MongoDB
- 3) A GUI tool to view and work with databases and collections**
- 4) Writing Java code to connect to MongoDB

### Correct answer: 3

Why: Compass is MongoDB's graphical interface for browsing data and running queries.

# Question 3

**What does MQL mean in MongoDB?**

- 1) Mongo Quick Language
- 2) MySQL Query Language
- 3) Mongo Query Language
- 4) Machine Query Language

# Answer 3

**What does MQL mean in MongoDB?**

- 1) Mongo Quick Language
- 2) MySQL Query Language
- 3) Mongo Query Language**
- 4) Machine Query Language

**Correct answer: 3**

Why: MQL stands for Mongo Query Language, the query syntax used to filter and work with documents.

# Question 4

**Which operation is part of CRUD?**

- 1) Update
- 2) Debug
- 3) Deploy
- 4) Compile

# Answer 4

**Which operation is part of CRUD?**

- 1) Update
- 2) Debug
- 3) Deploy
- 4) Compile

**Correct answer: 1**

Why: CRUD stands for Create, Read, Update, and Delete.

# Question 5

**If you run `use company\_db` in MongoDB Compass (shell), what happens?**

- 1) It deletes the database
- 2) It switches your current database context to company\_db
- 3) It changes the MongoDB server port
- 4) It creates a collection called company\_db

# Answer 5

**If you run `use company\_db` in MongoDB Compass (shell), what happens?**

- 1) It deletes the database
- 2) It switches your current database context to company\_db**
- 3) It changes the MongoDB server port
- 4) It creates a collection called company\_db

**Correct answer: 2**

Why: The `use` command changes which database your next commands apply to.

## Question 6

**Which query filters employees where department is exactly "Engineering"?**

- 1) db.employees.find({ department: { \$ne: "Engineering" } })
- 2) db.employees.find({ department: { \$gt: "Engineering" } })
- 3) db.employees.find({ department: { \$lt: "Engineering" } })
- 4) db.employees.find({ department: "Engineering" })

# Answer 6

**Which query filters employees where department is exactly "Engineering"?**

- 1) db.employees.find({ department: { \$ne: "Engineering" } })
- 2) db.employees.find({ department: { \$gt: "Engineering" } })
- 3) db.employees.find({ department: { \$lt: "Engineering" } })
- 4) **db.employees.find({ department: "Engineering" })**

**Correct answer: 4**

Why: An equality filter matches documents where the field value exactly equals the given value.

# Question 7

**Which operator matches values greater than a number?**

- 1) \$eq
- 2) \$ne
- 3) \$gt
- 4) \$lt

# Answer 7

**Which operator matches values greater than a number?**

- 1) \$eq
- 2) \$ne
- 3) \$gt**
- 4) \$lt

**Correct answer: 3**

Why: \$gt means greater than, so it matches numbers above the given value.

# Question 8

**What does a projection do in a find() query?**

- 1) Limits the number of documents returned
- 2) Inserts new fields into documents
- 3) Sorts the results
- 4) Chooses which fields to include or exclude in the output

# Answer 8

**What does a projection do in a find() query?**

- 1) Limits the number of documents returned
- 2) Inserts new fields into documents
- 3) Sorts the results
- 4) Chooses which fields to include or exclude in the output**

**Correct answer: 4**

Why: Projection controls which fields are returned in the result documents.

# Question 9

**Which projection returns only name and department and hides \_id?**

- 1) db.employees.find({}, { name: 0, department: 1, \_id: 0 })
- 2) db.employees.find({}, { name: 1, department: 0 })
- 3) db.employees.find({}, { name: 0, department: 0, \_id: 1 })
- 4) db.employees.find({}, { name: 1, department: 1, \_id: 0 })

# Answer 9

**Which projection returns only name and department and hides \_id?**

- 1) db.employees.find({}, { name: 0, department: 1, \_id: 0 })
- 2) db.employees.find({}, { name: 1, department: 0 })
- 3) db.employees.find({}, { name: 0, department: 0, \_id: 1 })
- 4) **db.employees.find({}, { name: 1, department: 1, \_id: 0 })**

**Correct answer: 4**

Why: Setting a field to 1 includes it, and setting \_id to 0 removes it from the output.

# Question 10

**In MongoDB, what is a document most similar to in MySQL?**

- 1) A row
- 2) A database
- 3) A server
- 4) An index

# Answer 10

**In MongoDB, what is a document most similar to in MySQL?**

- 1) A row
- 2) A database
- 3) A server
- 4) An index

**Correct answer: 1**

Why: A document represents one record, similar to a row in a table.