MongoDB / NoSQL Interview Questions and Answers

Q: What is MongoDB? How is it different from SQL databases?

A: MongoDB is a NoSQL database storing data in JSON-like documents. Unlike SQL databases, MongoDB has dynamic schemas, uses collections and documents instead of tables and rows, and scales horizontally.

Q: What is a document and a collection in MongoDB?

A: A document is a JSON-like object, and a collection is a group of documents. Documents are stored in BSON format.

Q: What is BSON in MongoDB?

A: BSON (Binary JSON) is a binary-encoded serialization of JSON-like documents. It supports additional data types like ObjectId and Date.

Q: What are the advantages of MongoDB?

A: Schema-less, high availability via replication, horizontal scalability via sharding, and native support for JSON and rich queries.

Q: How do you perform CRUD operations in MongoDB?

A: Create: db.users.insertOne({ name: 'Alice' })

Read: db.users.find({ name: 'Alice' })

Update: db.users.updateOne({ name: 'Alice' }, { $set: { age: 30 } })

Delete: db.users.deleteOne({ name: 'Alice' })

Q: What is indexing in MongoDB?

A: Indexes improve query performance. Example: db.users.createIndex({ email: 1 }) creates an ascending index on the email field.

Q: What is aggregation in MongoDB?

A: Aggregation processes data and returns computed results. Example: db.orders.aggregate([{ $group: { _id: '$customerId', total: { $sum: '$amount' } } }])

Q: What is the difference between find() and aggregate()?

A: find() is for simple queries, while aggregate() is for advanced data processing, grouping, and transformation.

Q: How does MongoDB handle relationships between documents?

A: MongoDB supports embedded documents (denormalization) and references using ObjectId (normalization). Use based on access patterns.

Q: What are some common MongoDB data types?

A: String, NumberInt, NumberLong, Boolean, Date, Array, Object, ObjectId, Null