EdX and its Members use cookies and other tracking technologies for performance, analytics, and marketing purposes. By using this website, you accept this use. Learn more about these technologies in the Privacy Policy.   ✕

edX

MongoDB Shell (mongo)

To launch the MongoDB shell (mongo), open a new terminal window, and run the following command:

```
mongo
```

Your command prompt should change from default $ (if you had it), to Mongo's >. This means your client was able to connect to the local database instance and you are ready to start manipulating the data in that database instance.

MongoDB Shell (mongo)

To test the database, use the JavaScript-like interface and commands such as save and find:

```
> db.test.save({a:1})
> db.test.find()
```

MongoDB uses JavaScript so all JavaScript and Node commands, classes and methods are a fair game in the Mongo console!

Some useful MongoDB Shell commands to know:

- `> help`: List of available Mongo shell commands

- `> show dbs`: List all the databases in this DB server/instance

- `> use board`: Work on a specific database named `board`

- `> show collections`: List all collections in this database

- `> db.messages.remove();`: Remove all documents from `messages` collection

- `> var a=db.messages.findOne();`

- `> printjson(a);`

- `> a.message="hi";`

- `> db.messages.save(a);`: Save method

- `> db.messages.find({});`: A read query

- `> db.messages.update({name:"John"},{$set:{message:"bye"}});`: An update documents query

- `> db.messages.find({name:"John"});`: A read query with a specific condition/query which matches only documents with property `name` which equals to value `John`

- `> db.messages.remove({name:"John"});`: A remove query with a condition

There are no schemas in MongoDB. Each document can have a completely different structure. This allows for de-normalization for the purposes of performance optimization and creation of distributed systems at a large scale.

Indexes are used to quickly perform search queries. Without indexes, MongoDB has to scan every document in the collection and find the ones that match the query which takes a long time. `_id` is the unique index that is created default, but you can create a custom index with one or more properties/fields. Each new index field will slow down the insertion operation but will quicken a search/read query on that index. Thus, having the right indexes is a best practice and a balance.