MongoDB

→ use mydb
→ db
→ show dbs
→ db.dropDatabase()
→ db.createCollection("mycollection")
→ show collections
→ db.createCollection("mycol", {capped: true, autoIndexID: true, size: 6142800, max: 10000})
→ db.COLLECTION_NAME.drop()
→ db.COLLECTION_NAME.insert({title: 'Hello'})
→ id → 12 byte hexadecimal Numbers

Query → db.COLLECTION_NAME.find()
→ db.COLLECTION_NAME.pretty()

AND → db.mycol.find({key1: value1, key2: value2}).pretty()
OR → db.mycol.find({$or: [{key1: value1}, {key2: value2}]}).pretty()

update → db.COLLECTION_NAME.update(SELECTION_CRITERIA, UPDATED_DATA)

eg. db.mycol.update({'title': 'mongodb'}, {$set: {'title': 'new document'}}, {multi: true})

remove → db.COLLECTION_NAME.remove(DELETION_CRITTERIA)
, 1 → To delete first record

projection → db.mycol.find({}, {"title": 1, _id: 0})
limit → db.COLLECTION_NAME.find().limit(NUMBER)
skip → " " " .skip(NUMBER)
sort → " " " .sort({"title": -1})

Indexing: → db.COLLECTION_NAME.ensureIndex({key:1})

Aggregate: → db.COLLECTION_NAME.aggregate(Aggregate_operation)

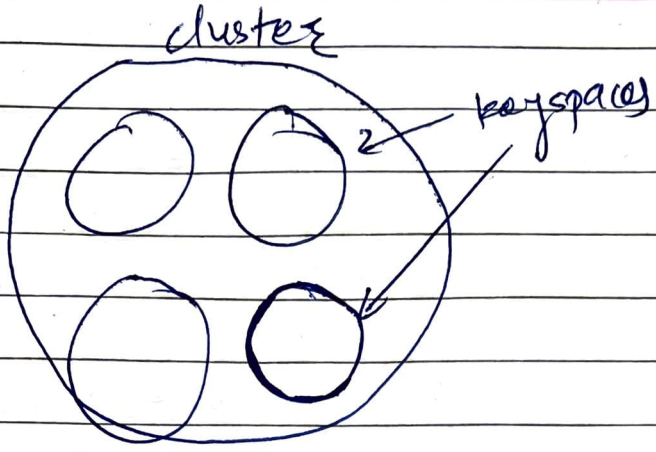
eg: db.mycol.aggregate([{$group: {_id: "$by_user", num_tutorial: {$sum: 1}}}])

Replication: It's a process of synchronizing data across multiple servers.

mongoDB Backup:	mongoDB restore data:
> mongodump	> mongorestore

→ db.mycol.find({"title.name": "_____"});

Cassandra



→ CREATE KEYSPACE test_keyspace WITH replication = {'class': 'simpleStrategy', 'replication_factor': '1'}
AND durable_writes = 'true';

→ ~~disc~~ DISCRIBE KEYSPACES; → excalibur
↳ to show options.

→ DROP KEYSPACE ——;

→ USE test_keyspace; → To use keyspace

→ CREATE TABLE employee_by_id (id int PRIMARY KEY,
name text, position text);

→ DESCRIBE TABLES;

→ DROP TABLE ——;

→ consistency level → responce from node

↳ Quorum = $\frac{RF+1}{2}$

→ CONSISTENCY

→ CONSISTENCY QUORUM; → (ONE, TWO, THREE)

→ INSERT INTO employee_by_id (—, —, —)
VALUES (—, —, —);