**The MongoDB Cheat sheet**

In mongoDB, a document means one record.

Database consists of collections, collections consist of documents, documents consist of key value pairs.

**Databases**

To create a database: Use dbname

To drop a database: db.dropDatabase()

To view databases: show dbs

To create a collection: db.createCollection(‘students’)

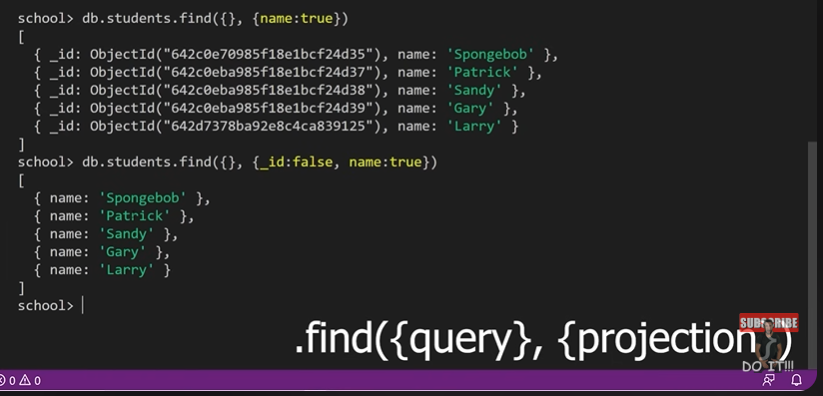
**Read**

To return all documents within a collection: db.students.find()

You can add more than one filters:



Using projection parameter:



**Insert**

To insert a document within a collection: Db.students.insertOne({name:”Tayyib”, age: 30, gpa: 4.0})

To insert more than one documents at a time: db.students.insertManyDocuments([

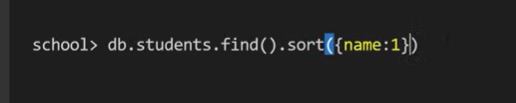
{name: “Aamina”, age: 29, gpa: 4.1}, {name: “Farooq”, age: 100, gpa: 4.0}, …

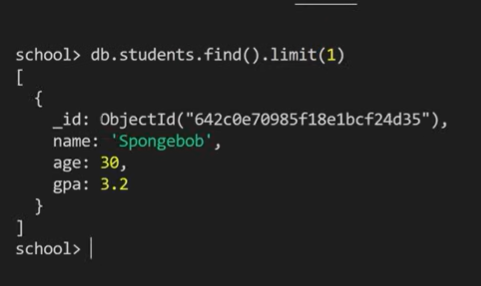
])

**Datatype**

String, numbers, decimals, Booleans, date (e.g., new Date()), null, array (e.g., [comma, separated, values]), nested document (e.g., {})

**Sort and limit**

****

****

**Update**

To add/update a field (One document):

db.students.updateOne(filter, update)

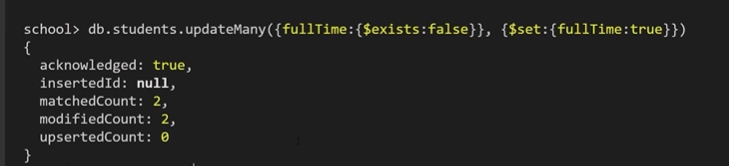
e.g., db.students.updateOne({name:”Tayyib”}, {$set:{fullTime: true}})

Note: the set operator updates the value of a field of document

We can also unset a field:

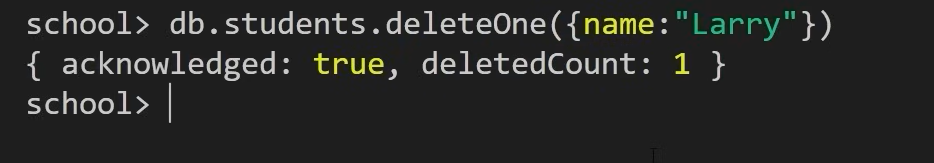
Db.students.updateOne(filter, {$unset: {fullTime: “”}})

For many documents:

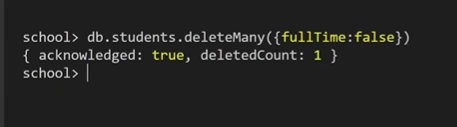


**To delete**

One record:



Multiple records:



**Comparison operators**

Find every name that is not spongebog:

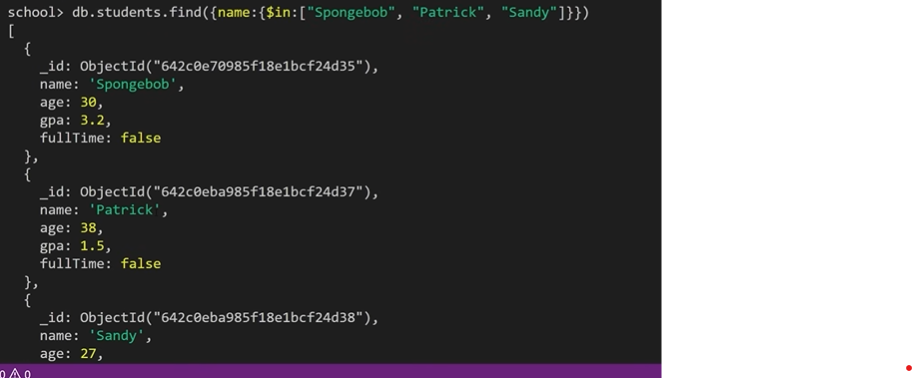
****

In place of $ne, we can use $lt, $lte, $gt, $gte as well.

To define a range with comparison operators:

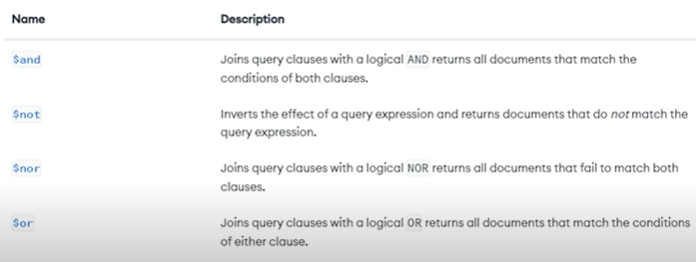


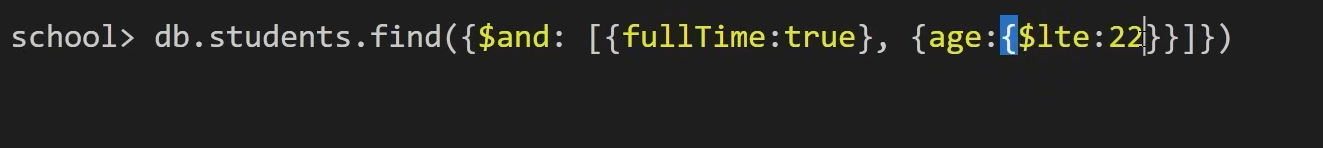
Another fun operator is the **in** operator:



We could also use $nin (Meaning not in)

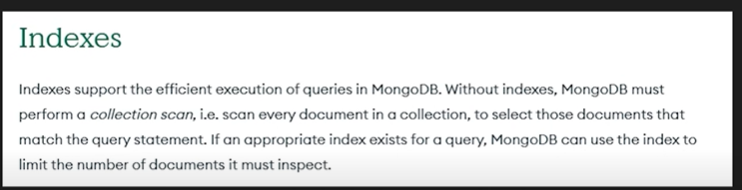
**Logical operators**

****

****

Note suppose you have 3/10 records in your database with age < 20, and 3 records have age = null. When you run the query that age lt 20, you only get 3. The records with age = null are not returned. To solve this problem, you can do: age: {$not: {$gte: 30}}. This will return the 5 required records.

**Index**

****

Pros:

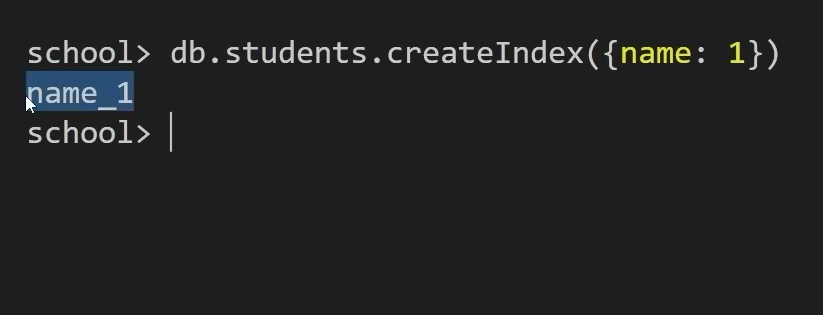
Creating index on a field allows fast look up for that field.

Cons:

Takes up more memory

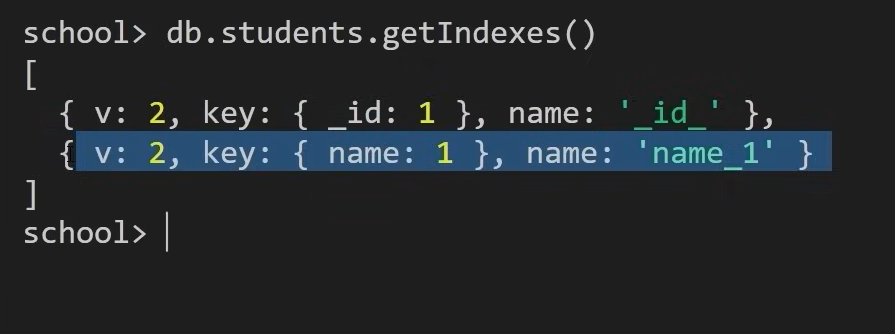
Slows insert, remove and update operations

To create an index:

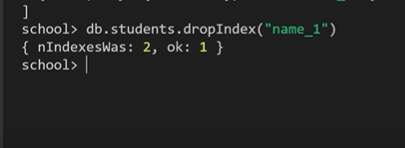


The 1 in front of name means we are creating index based on name column in ascending order.

To view indexes:



To drop an index:



**Creating collection**

****