

Mongo DB:

This table will list the commands most commonly used when working with the mongo dB as a whole.

Type	Command	Description
Create/Connect	<code>use <db></code>	Connects to a specific database. If none exists then one will automatically be created with that name.
List All	<code>show dbs</code>	Lists all Databases. DBs with no data are not shown.
List Current	<code>db.getName()</code>	Lists the name of the currently selected database.
Return	<code>db</code>	Returns the currently selected Database.
Drop	<code>db.dropDatabase()</code>	Drops the currently selected Database.
Stats	<code>db.stats()</code>	Lists the stats about the current Database.
Create	<code>db.createCollection('<collection>')</code>	Creates a new empty collection in the database.
List	<code>db.getCollectionNames()</code>	Lists all collections for a current Database.
Return	<code>db.getCollection('<collection>')</code>	Returns a collection.
Return/Create	<code>db.<collection></code>	Similar to <code>db.getCollection()</code> , but if the collection doesn't exist, then the collection will be return, but not

		added unless data is also added to that collection at the same time.
Drop	<code>db.<collection>.drop()</code>	Drops the collection from the database.
Rename	<code>db.<collectn>.renameCollection('collection')</code>	Renames a collection.
Create One	<code>.insertOne({<doc>})</code>	Inserts a document into the collection.
Create One/Many	<code>.insert([{<doc>},{<doc>},{<doc>}])</code>	Inserts One or more documents into the collection. If an array is passed in it will make a record of each document in the array. Otherwise it will accept a single document.
Count	<code>.count()</code>	Returns the number of items in the collection.
Return All	<code>.find({})</code>	Returns an array of all items in a collection. <code>.find()</code> Must be passed <code>{}</code> in order to return all documents.
Return All Filtered	<code>.find({query})</code>	Returns an array of items in a collection that match the query passed into <code>.find()</code> .
Return One Filtered	<code>.findOne({query})</code>	Returns a document (not an array) of the first item found, filtering based off what was passed

into `.findOne()`. Useful when searching by unique fields .

Update/Replace `.update({query}, { $set:
{query} }, options)`

The first argument is used as the query to find the document. The second argument specifies which field which to update.

Exclude `$set:` and the entire document will be Replaced.

Common options: `upsert:` `<boolean>` to keep it unique, and `multi:` `<boolean>` will update multiple documents if set to true.

Update
One/Many `.updateOne()` and `.updateMany()`

Basically the same as the above function, except the `multi:` `<boolean>` option is basically defaulted to false and true, and isn't an allowed option that can be passed in.

Limit `db.COLLECTION_NAME.find().limit(NUMBER)`

The number of documents that you want to be displayed.

Delete
One `.deleteOne({query})`

Deletes the first document that matches the query. Recommend to search by `_id` or another unique field.

Limit	<code>db.COLLECTION_NAME.find().limit(NUMBER)</code>	The number of documents that you want to be displayed.
Sort	<code>db.COLLECTION_NAME.find().sort({KEY:1})</code>	Accepts a document containing a list of fields along with their sorting order. To specify sorting order 1 and -1 are used. 1 is used for ascending order while -1 is used for descending order.
Delete Many/All	<code>.deleteMany({query})</code>	Deletes all records that match the query. Leave the query blank to delete all documents.
	<code>.findOneAndDelete({query})</code>	Finds the first document that matches the query and deletes it.
	<code>.findOneAndUpdate({query}, {<update>}, {<options>})</code>	Finds the first document that matches the query in the first argument, and updates it using the second arguments. Has optional options as well.
	<code>.findOneAndReplace({query}, {<replacement>}, {<options>})</code>	Finds and replaces the document that matches the query. <replacement> cannot