

MongoDB → Node.js.

* Installing MongoDB driver:
npm install mongodb

* Node.js can use ~~node~~ this module to manipulate MongoDB DB:

```
var mongo = require('mongodb');
```

* Creating Database:

To create DB, start by creating a MongoClient object, then specify a connection URL with the correct IP address & the name of the DB you want to create.



~~var~~ ~~Mon~~

In MongoDB, a database is not created until it gets content.

MongoDB waits until you have created a collection, with at least one doc before it actually create DB & collection.

* Collection Creation:

A collection in MongoDB is the same as a table in MySQL.
In MongoDB, a collection is not created until it gets content.

* Insert:

The 1st para. of the `insertOne()` method is an object containing the name & value of each field in the doc. you want to insert

eg: insert a doc. in "school" collection

```
db.collection("school").insertOne({myobj: { "Cerritos" } })  
if (err) throw err;  
console.log("1 doc inserted");  
db.close();  
}
```

* Find one:

~~Find one~~ We use the 'find' & 'findOne' methods to find data in a collection.

Just like 'select' statm. is used to find data in SQL.

To select data ^{from collection} we use `findOne()` method.
It returns first occurrence in selection.

1st para. is always a query object.

* Find All:

To select data from table we use `find()` method.
It returns all occurrences in the collection.

No para. in the find() method gives you the same result as SELECT * in MySQL.

* Find

The 2nd para. of find() method is the 'projection' obj. that describes which field to include in result.

It's optional, but if omitted then all field will be included.

eg.

db.collection("School").find({}, {projection: {id: 0}})
↳ Array. (for curr, result)

* Query

When finding doc. in collection, we can filter result by using a query obj.

1st argument of find() method is a query obj & is used to limit the search.

* Sort

sort() method to sort the result in ascending order / descending

It takes 1 para, an obj. defining the sorting order

* Delete:

To delete record, or doc. as it's called in MongoDB, we use the `deleteOne()` method.

1st para. of `deleteOne()` is a query obj. defining which doc. to delete.

If query is more than one then 1st occurrence will be deleted.

* Drop:

You can delete a table (collection), by using `drop()` method.

The `drop()` method takes a callback fun. containing the error obj & the result para which returns true if the collection was dropped successfully, otherwise false.

* db.dropCollection() method:

It's used to delete a table (collection).

It takes 2 para: the name of the collection & a callback fun.

* Update Doc:

To update a record, doc. it's done by using `updateOne()` method. 1st para. of '`updateOne()`' method is a query object which define doc. to update.

* Limit:

To limit result, we use `limit()` method.

The `limit()` method takes one para, a no. defining how many doc. to return.

* Join:

→ MongoDB is not a relational database, but you can perform a left outer join by using the '\$lookup' stage.

→ The '\$lookup' stage lets you specify which collection you want to join with current collection & which fields that should match.