1. We had a detailed look at
   1. Database’s collections and documents and what these things are.
   2. DB holds collections and collection holds documents. This is how **mongodb structures data**.
   3. Databases and collections are **created lazily**. Only when a document is inserted.
   4. Document can’t be inserted directly into database. It’s always part of collection. Keep that **layered architecture** in mind.
   5. Documents have a certain **structure** each document needs a **unique ID which has to be called \_id.** You can set that on your own or take the one mongodb creates for you.
   6. You can have embedded documents as well as arrays as well as texts, numbers, all the other data types.
   7. You can perform CRUD Operations.
   8. Mongodb offers multiple CRUD operations for a single document and bulk actions, you can insert one element/document or multiple documents in one go. You can find many documents or one document.

Some methods require arguments 🡺 insertOne({d1}) whereas in find() you can pass or not depending on your need.

* 1. Find() returns cursor which is useful to manually go through the list otherwise all the documents which could be thousands or millions would have to be transefferred over the wire in one go that would not be great for your bandwidth and your connection.
  2. We can use filters on find(), update(), delete() to narrow down which documents you want to work with.
  3. While retrieving data, you can use filters and operators allowing you to narrow down the set of the data found like the greater than operator and the such operators can be identified by the fact that they start with $ and they simply allows you to limit the amount of data you’re fetching. Then you can use project to then take the data you fetched and tell mongodb which fields you actually want to get and which you don’t want to get.  
     **NOTE**: **Filters** allow you to restrict the amount of documents.   
     **NOPE**: **Projects** allow you to restrict the amount of fields per document.

1. 