**SCHEMA & RELATIONS**

**(How to structure your documents)**

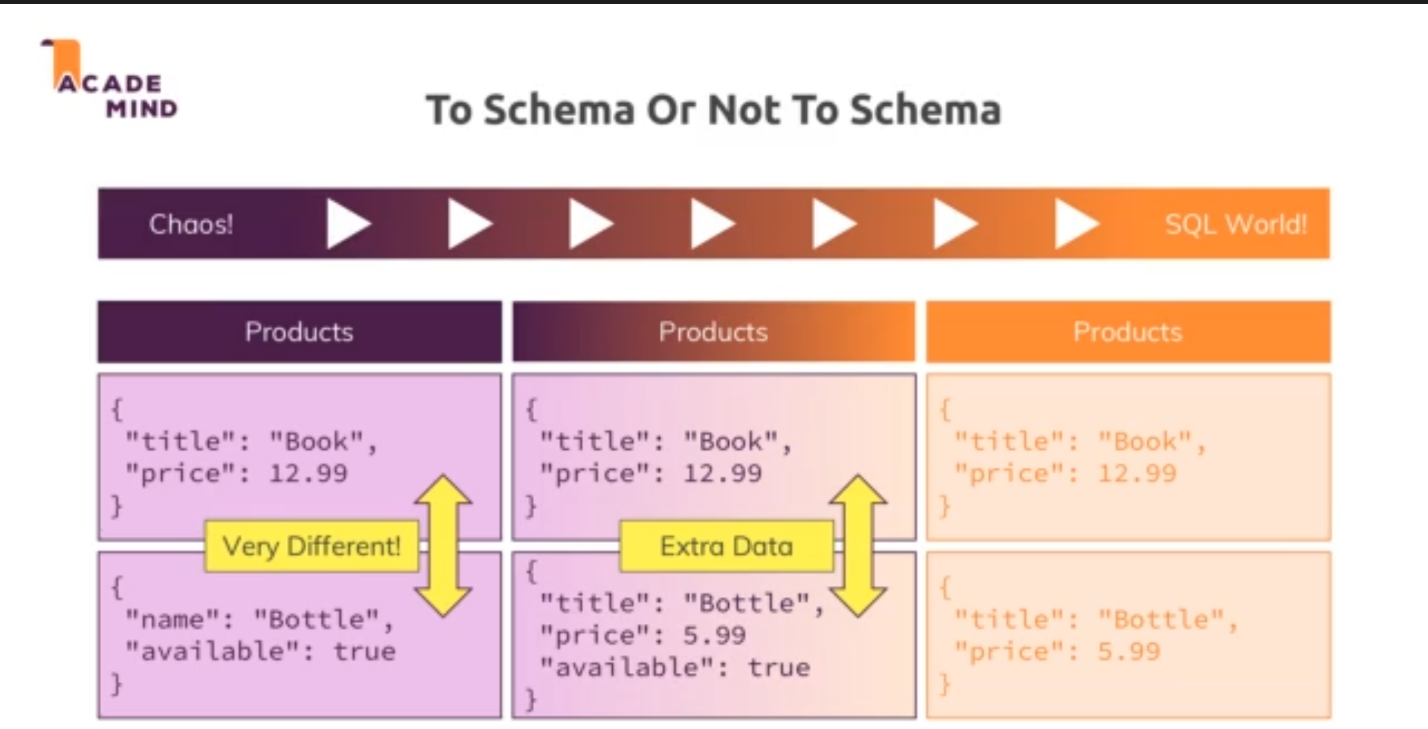
As we know that mongo-db is a schema-less and non-relational database, we can store any types of data in the documents, it can also be possible that there are two documents in a collections that looks totally different from each other. But I real life scenerios we have to have some kind of structure to store our data.

**Data storage Hard-Limit for Mongodb:**

* The overall document (one document) size must not be greater than **16mb**.
* These must not be greater than **100 level** of nesting/embedded documents.

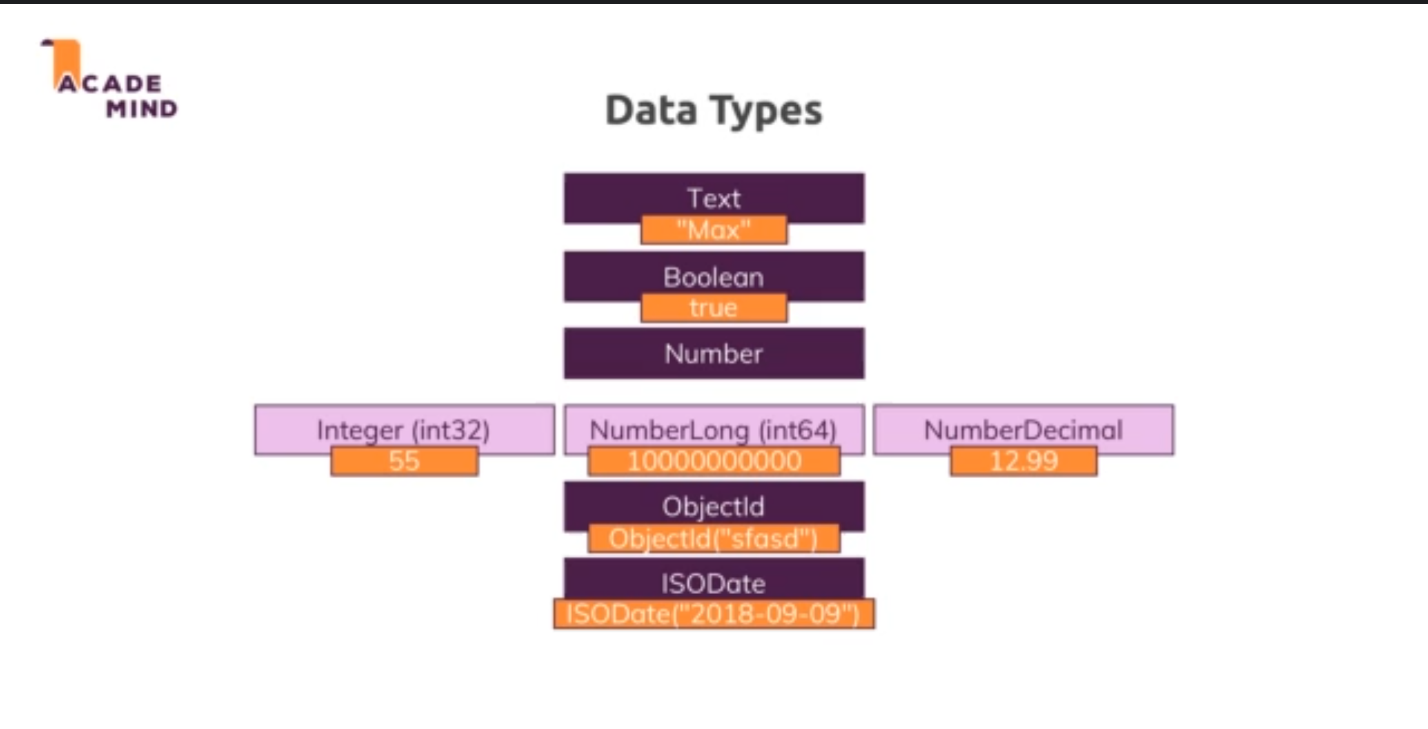
**Different levels of structured Data:**

Given below the example of the documents stored in collection named products, that goes from highly unstructured, semi-structured to highly structured(SQLise) documents.

****

**Major Data Types in Mongodb:**

Out of All available data types in the mongodb, here are some of the most widely used data types :

****

**Relations**

Even though mongodb is a non-relational database relations in mongodb can be established using two methodologies:

* **Embedded/Nested documents:** When we store an Object or many Objects in our mongodb document than it is termed as Embedded/Nested documents. E.g. address is a embedded document in the mongodb document given below.

**{**

**\_id: ObjectId(‘3874893rhe2rhe289349’),**

**name: “Harshit”,**

**address: {**

**city: “Indore”,**

**state:”Madhya Pradesh”**

**}**

**}**

* **References :**  When we store the ObjectId of the document from some another collection into our document than it is termed as references.

E.g. We have a Driver collection and a Truck collection each driver is assigned a truck, so we keep reference of the truck doc in the driver’s document as truckId.

**{**

**\_id: ObjectId(“agfsd2786eui23yer3uyr”),**

**name: ‘Suresh Rana’,**

**liceneseNumber: “7423894n2k4n”,**

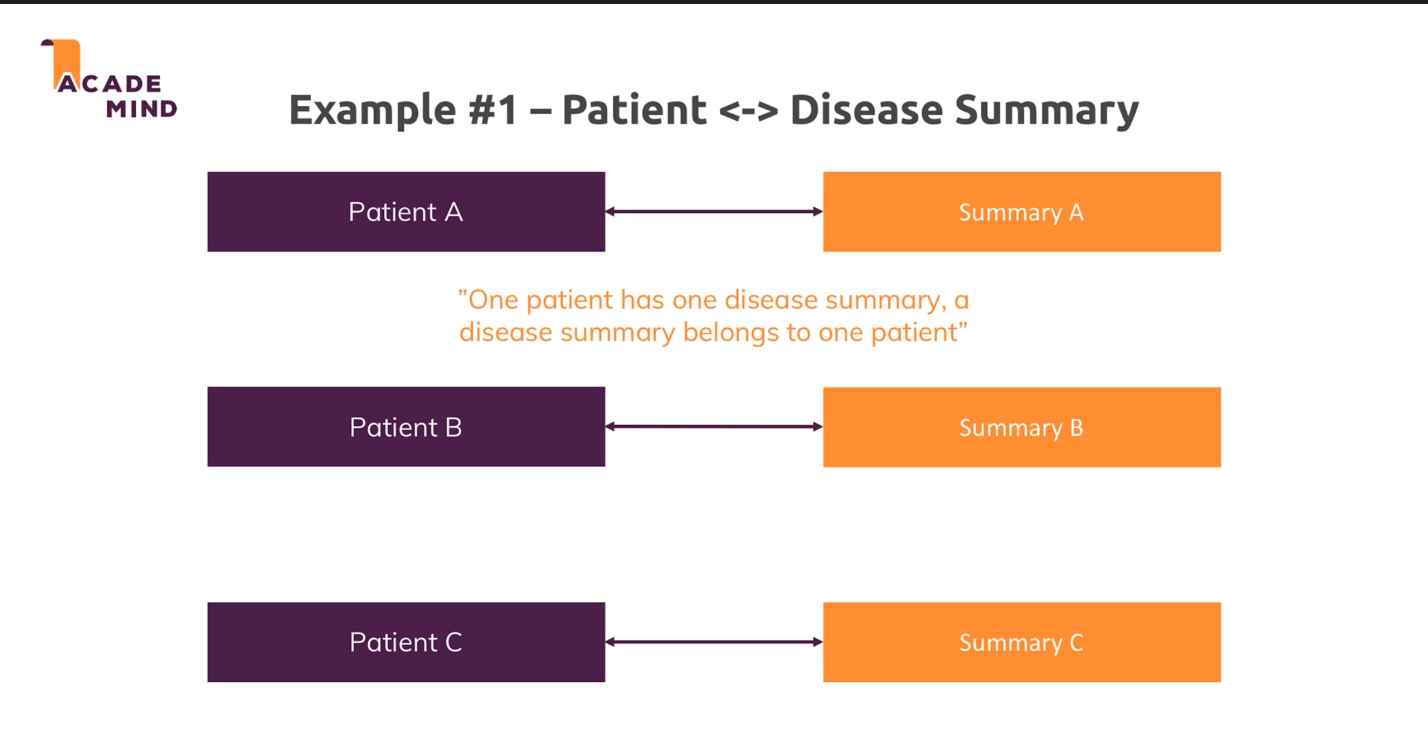
**isActive: true,**

**truckId: ObjectId(“sjkdfhjkshdhskjdklsdklsj”)**

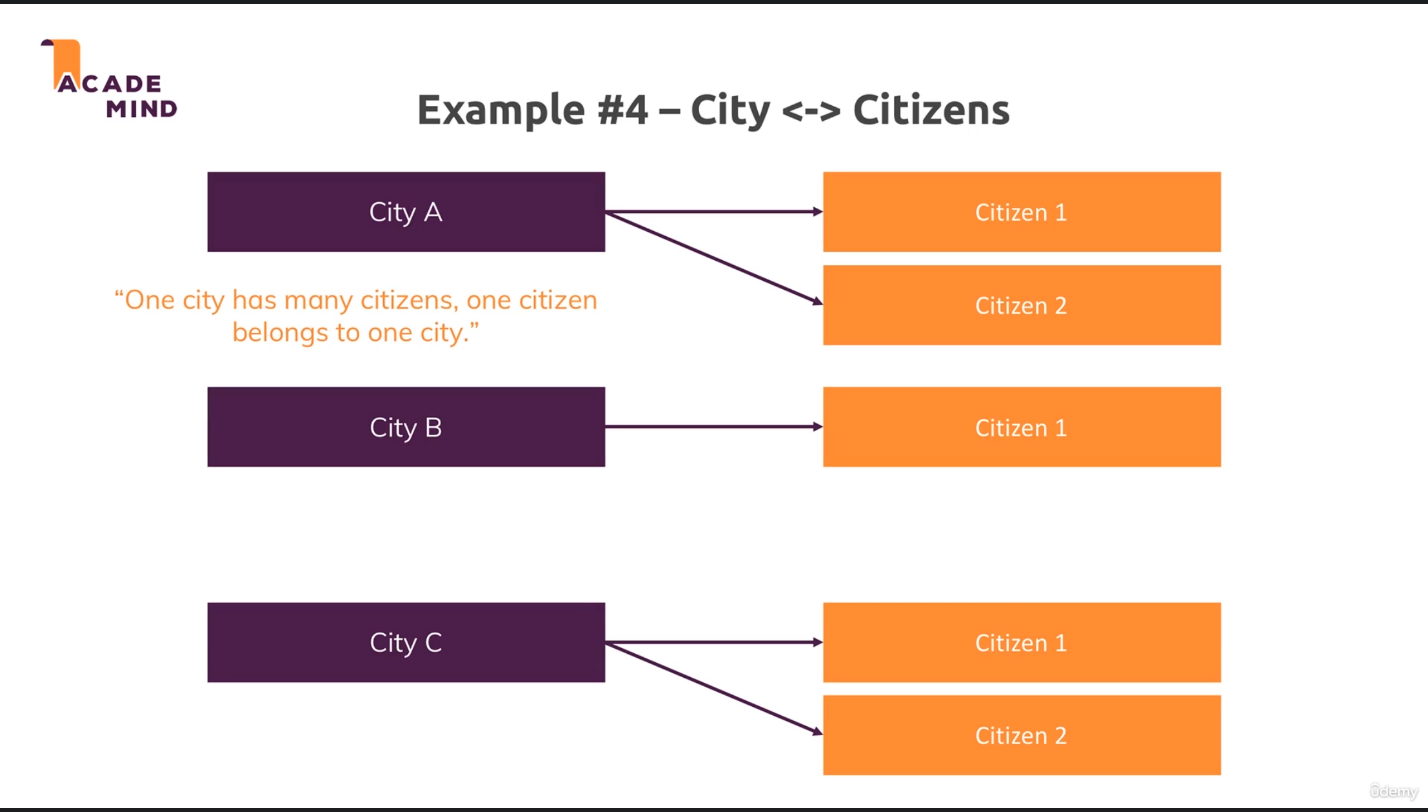
**}**

**Types of Relations:**

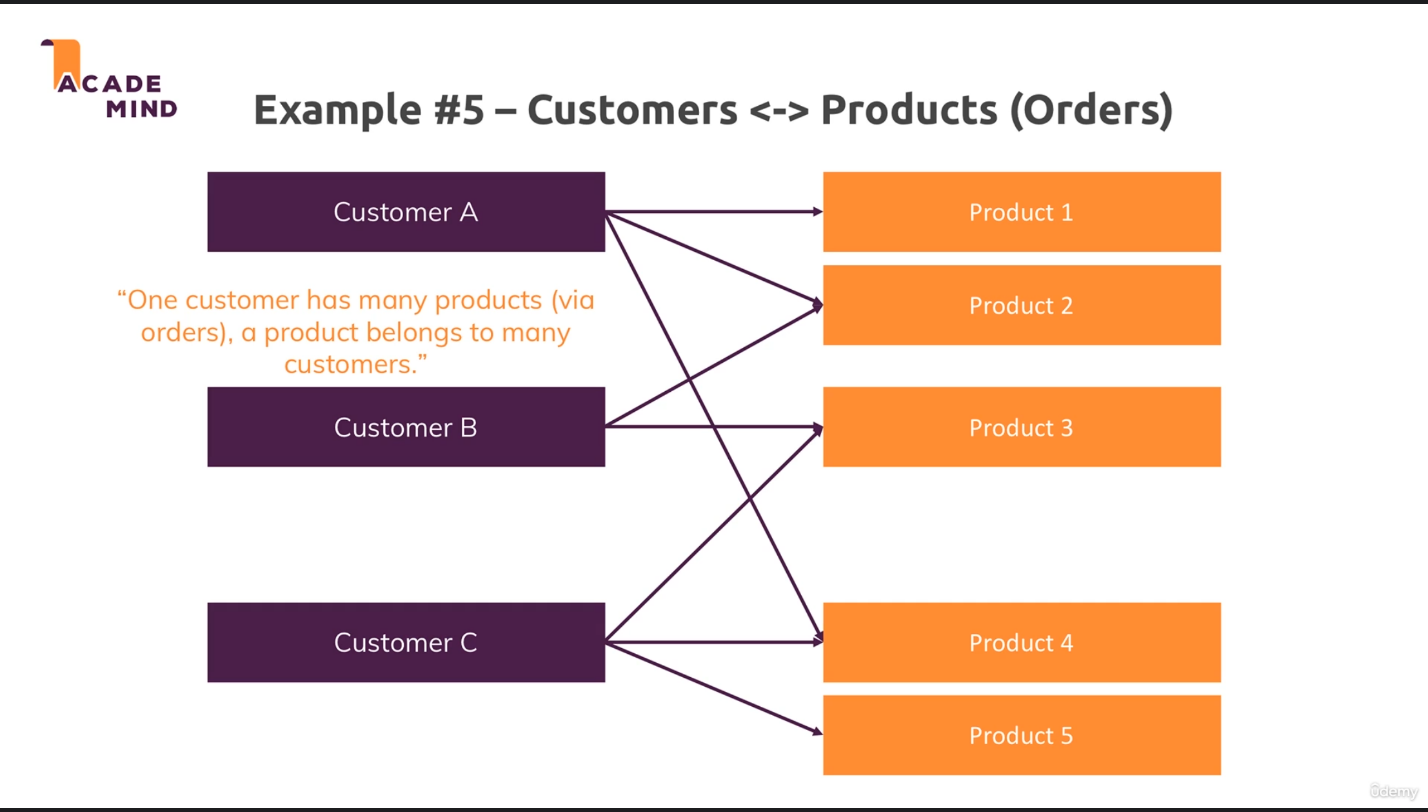
* **One To One:**  In this type of relations one document From Collection A is related to One Document in Collection B. ref:Patient Collection and Disease Collection.



* **One To many:** In this type of relations one Document from collection A is related to two or more documents from the collection B. ref: City Collection and Citizens collection.



* **Many to Many:** In this type of relations many documents from Collection A is related to many documents of collection B. ref: Customers Collections and Products Collection.



**Note:** Its our call as per our requirement and amount of data that we implement these relations using embedded documents or references approach.