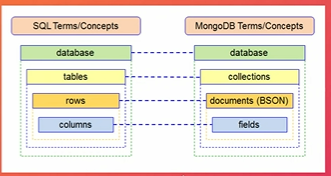
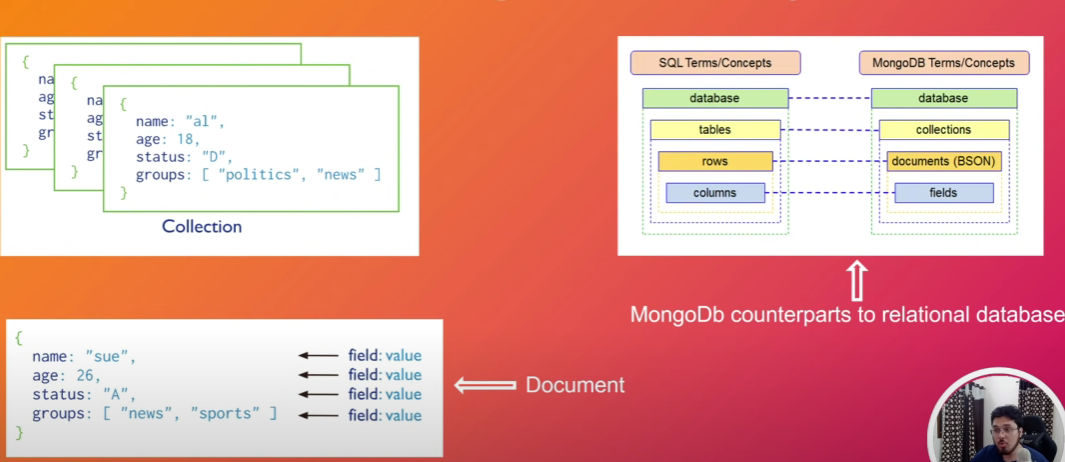
No SQL and MongoDB

NoSQL??

1. No SQL databases are primarily called non-relational or distributed database.
2. No SQL databases are document-based key-value pairs, or graph databases.
3. No SQL databases are horizontality scalable
4. MongoDB, Redis. Cassandra etc





* Difference betwn Mongo and Mongod is, **Mongo** is a command line shell that connects into specific instance of **Mongod**, where **Mongod is a** mongo daemon basically a host process for the database.
* It represents in the form of tables known as **collection** and **field and value.**
* After opening into the console from the file location we need to first check if the server is available or not? By using command **mongod**
* To check mongdb version – **mongo --version**
* To view all database – **show dbs**
* To Create and Swap a database – **use database\_name**
* To check in which database you are in/ current database - **db**
* To delete Database – **db.dropDatabase()**

**MongoDB commands for collections**

* To create collections – **db.createCollection(“collection\_name”)**
* To delete a collection – **db.collection\_name.drop()**
* To show collections – **show collections**

**MongoDB commands for Rows**

* Insert one data into collection **- db.collection\_name.insert ( {"\_id":1, "name": "banty", "age": "23"} )**
* Insert multiple data into collection –
  + **db.collection\_name.insertmany ( [ {"\_id":2, "name":"raja", "class":"10th"}, {"\_id":3, "name":"Dudul", "class" :”11th”] )**
* To show all **ROWS** collections details in database – **db.collection\_name.find() // db.collection\_name.find().pretty()**
* In MongoDB collections **object Id acts as the primary key**.
* **db.programs.insert({ "date": new Date() }) –** by using this new Date we can add current date.
* **db.collection\_name.find({condition1\_ for \_search}) –** It gives us the result for searching any element in the collection eg **db.collection\_name.find( {name:”Raja”} )**

we can pass multiple condition here as well **- db.collection\_name.find({condition1\_ for \_search, condition2\_ for \_search })**

* **db.collection\_name.find().pretty().limit(2) –** It shows the first two like range row of the collections
* **db. collection \_name.find().count() -** It shows the count of the data present in the collections
* **db. collection \_name.find ({ 'class':'11th' }).count() –** It counts number of students having class 11th
* **db. collection \_name .find().sort({“condition\_field':-1}).pretty() –** It shows in descending order
* **db. collection \_name .find().sort({condition\_field': 1}).pretty() -**It shows in ascending order

**Search In MongoDB**

* Search by the field and value : **db. collection \_name.find ({ 'class':'11th' })**
* Search only one data : **db. collecrtion\_name.findOne({'name':"raja"})**

**Update in Collections**

* **db. collection \_name.update({ name :'banty'},{'name':'banty','class':'13th'}) – changed the condition at uncerline part**
* **db. collection \_name.update({ name :'king'},{'\_id':5,'name':'king','class':'11th'},{upsert:true}) –** It update the collection where name is King but in case name is not present the it will add it to collections
* **db. collection \_name.update({ name :'king'},{'\_id':5,'name':'king','class':'11th','age':20},{upsert:true}) -**updated collections and added age with value 20

if I want to **increase/increment**  the age section with value 2 then: **db. collection \_name.update({ name :'king'},**

**{$inc : { age:2} } )**

* **db. collection \_name.update({ name :'hari'}, {$rename:{ date: 'updated\_date'}}) –**

If you want to update the field section for a particular row then we can use this RENAME

* Remove row from a collections - **db. collection \_name.remove({name : "banty"})**
* Grater than less than matching **: db.collection\_name.find( { age: {$gt: 23}} )**
  + **db.collection\_name.find( { age: {$lt: 23}} )**