1=>

MongoDB:

->Didtributed Database(Horizontal Scaling-> dividng into multiple servers)

whereas MYSQL is Vertically Scaled.

-> NOSQL Database

-> Schemaless

-> Non Relational

-> No concept of Joins

-> Takes data in the form of JSON.

2=>

JSON:

{

"name":"Aditya"

//comma is invalid.

}

OBJECT:

{

name:'Aditya',

or

'name':'Aditya'

or

name:"Aditya"

//comma is valid.

}

3=>

MYSQL NoSQL

----------------------------------------------------

table -> Collection

row -> Document

column -> field

id -> object id (\_id)

4=>

object id is in the form of BJSON(Binary).

BJSON - 12 Byte

4Byte 4Byte 4Byte

Timestamp Server ID/ random number

(Current Time/ Process ID

Date)

//site: medium

4Byte 3Byte 2Byte

Timestamp Server ID

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

DAY 2:

5=>

DB-> Colllection-> Document

=> step 1: copy bin location, set path in environment settings.

Open cmd-> mongod-> server gets on

Step 2: open another cmd file-> mongo

=> show dbs

Show default databases present.

=> use test

creates new database test. //Its not displayed in databases until its empty

=> db.createCollection('aptech')

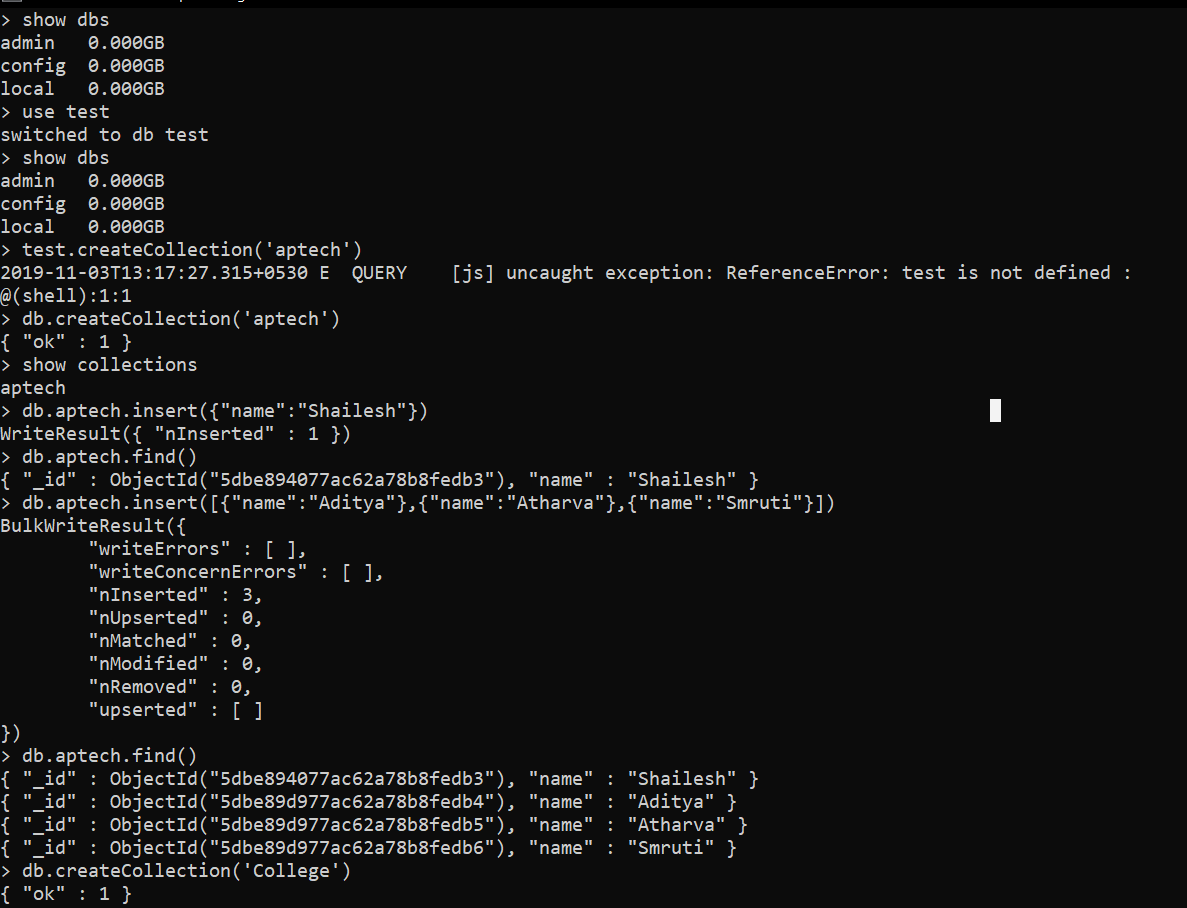
creates new collection(table). //{ "ok" : 1 } ensures that collection is created

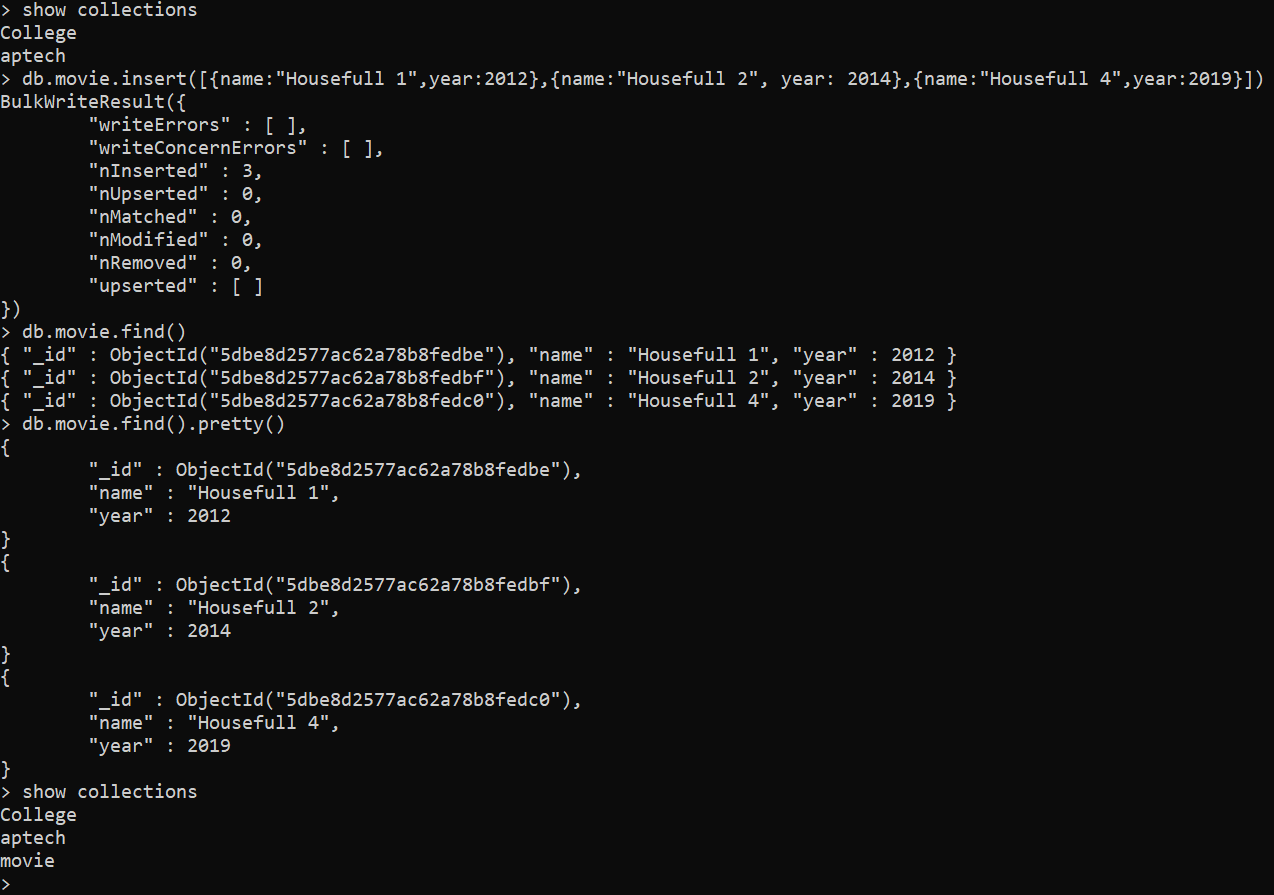
//collection can also be created by inserting .

=> show collection

shows collection made

=>





**insert()**

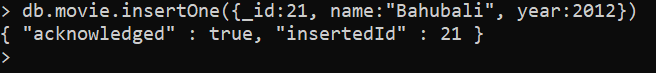
* Single
* Multiple

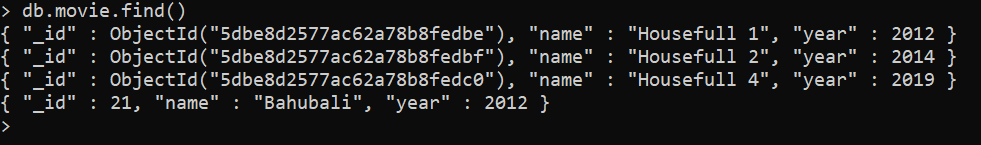
**insetOne()**

* Single, returns object id

**insertMany()**

* Many, returns object ids.

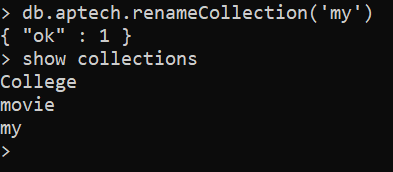




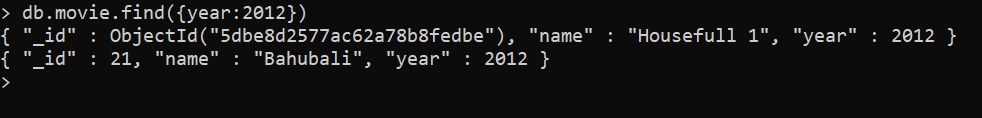
**db.dropDatabase** – drops the current database

**db.movie.drop() –** drops the collection

**rename:**



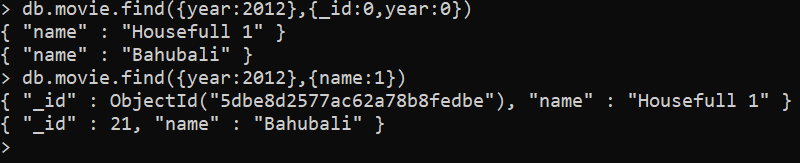
**Queries:**

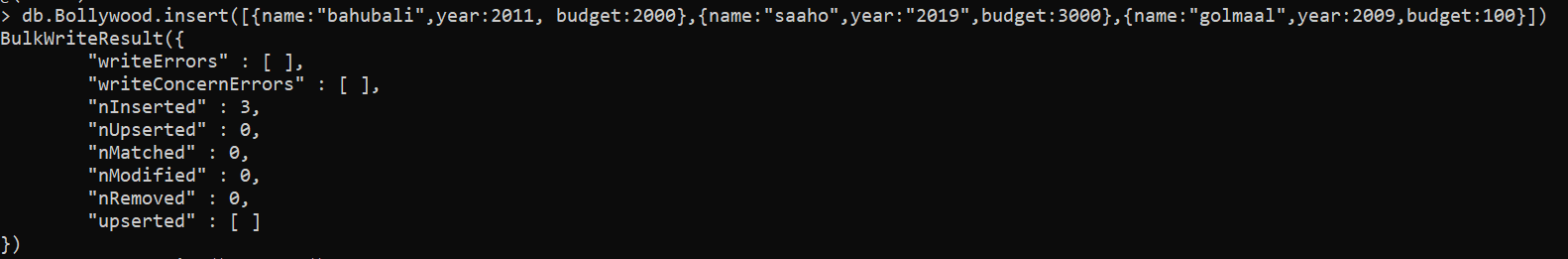


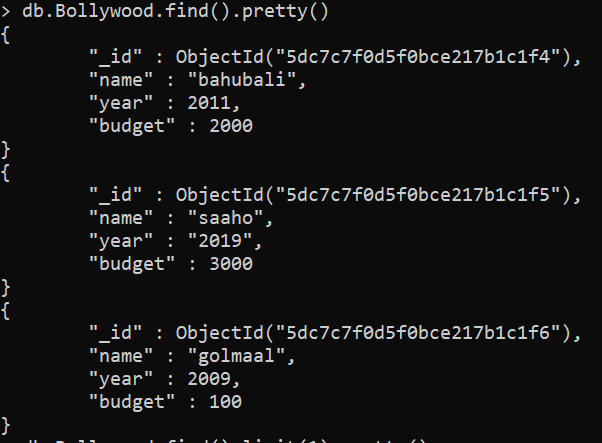
**Projections:**

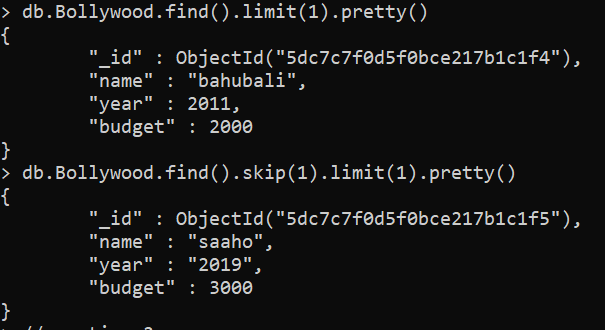
**Name:0 -> name will not be represented.**

**Name: 1 -> name will be displayed**

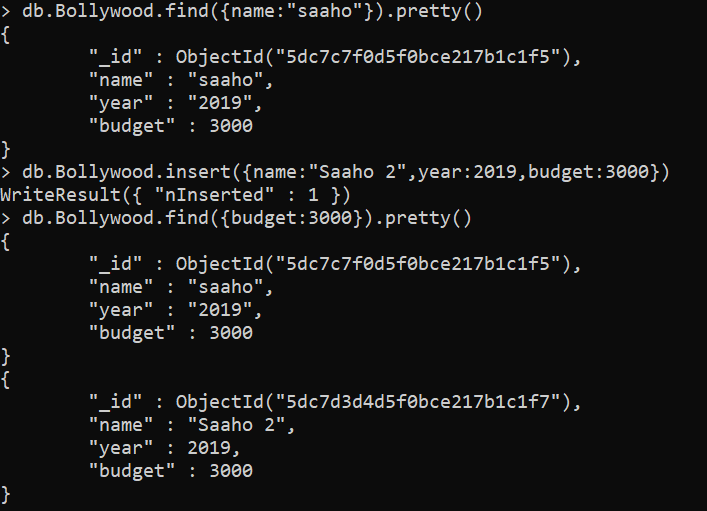


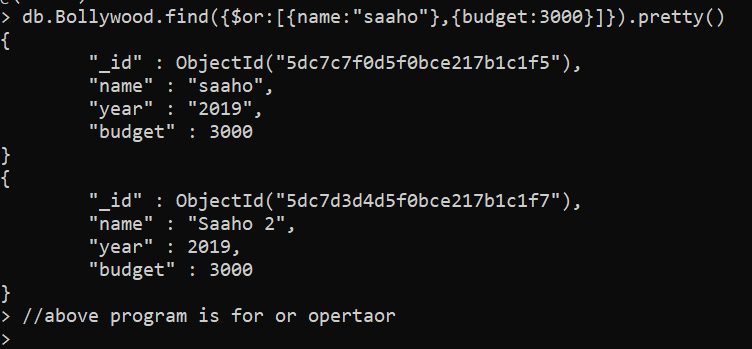








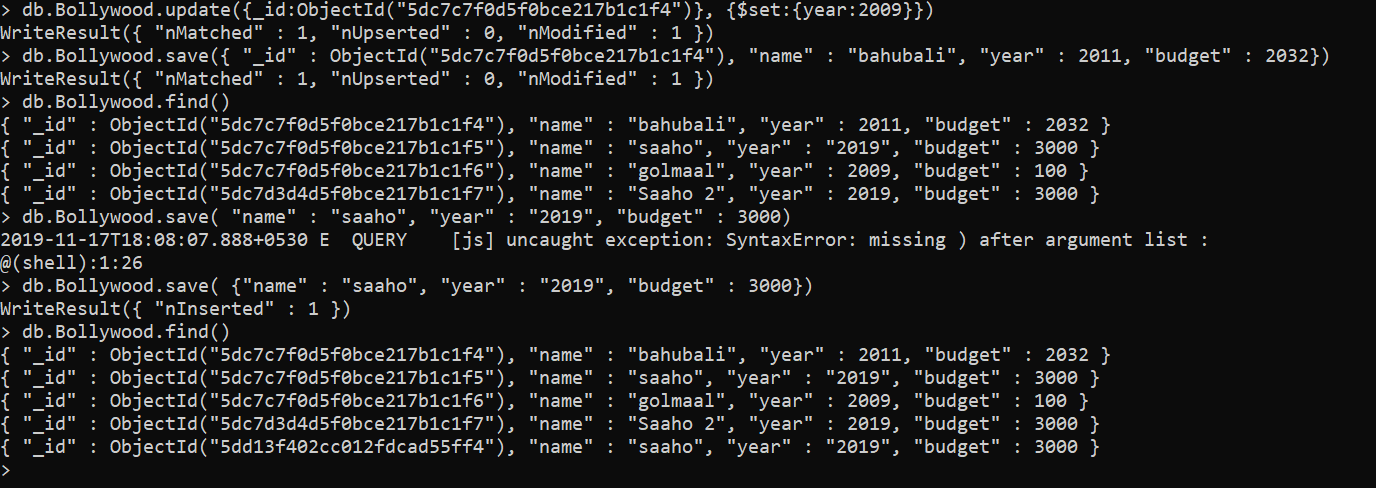




db.Bollywood.find({budget:{$in:[3000,2000]}}).pretty

above query displays movies having budget of 2000 and 3000. (Not Range).

* Difference between save and set is that if save is called without \_id, then it creates a new record.

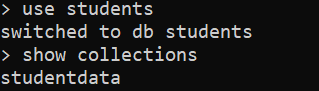


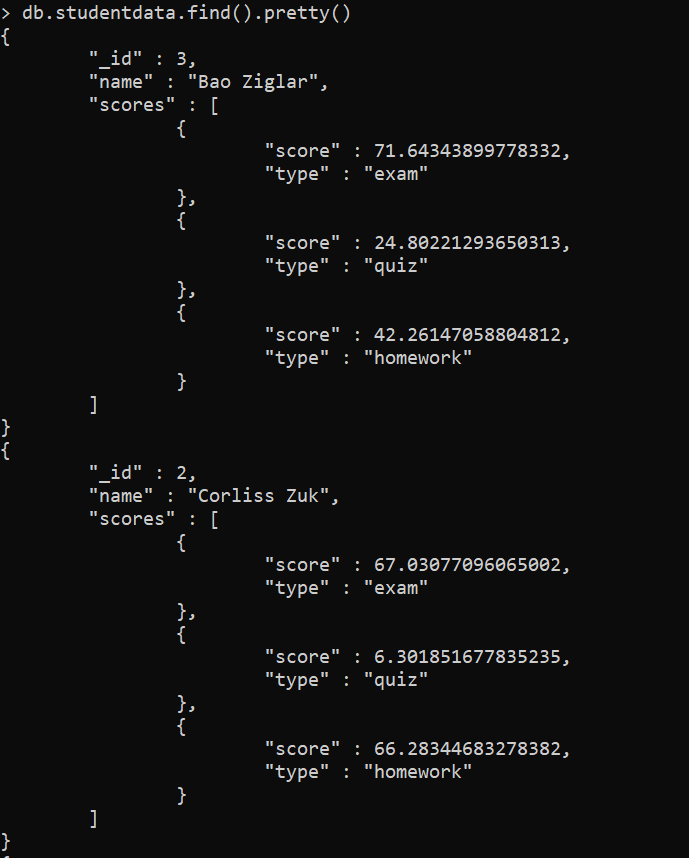
Json file was imported using command mongoimport:

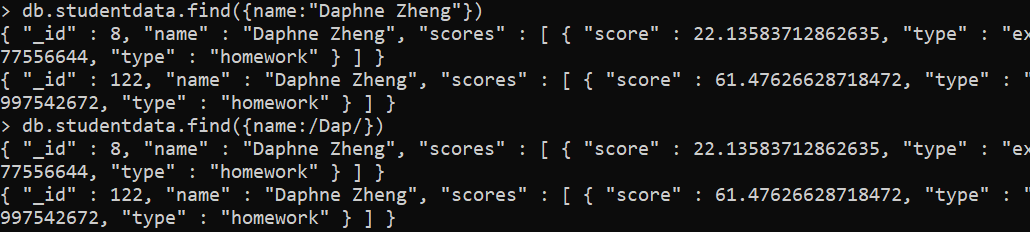
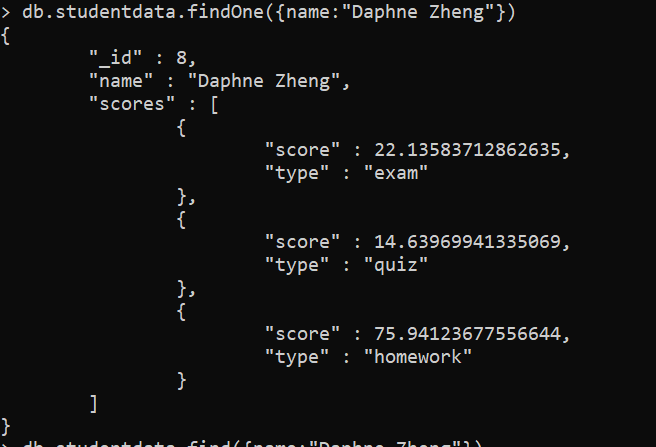
mongoimport jsonfilename -d=db\_name -c=collection\_name --jsonArray

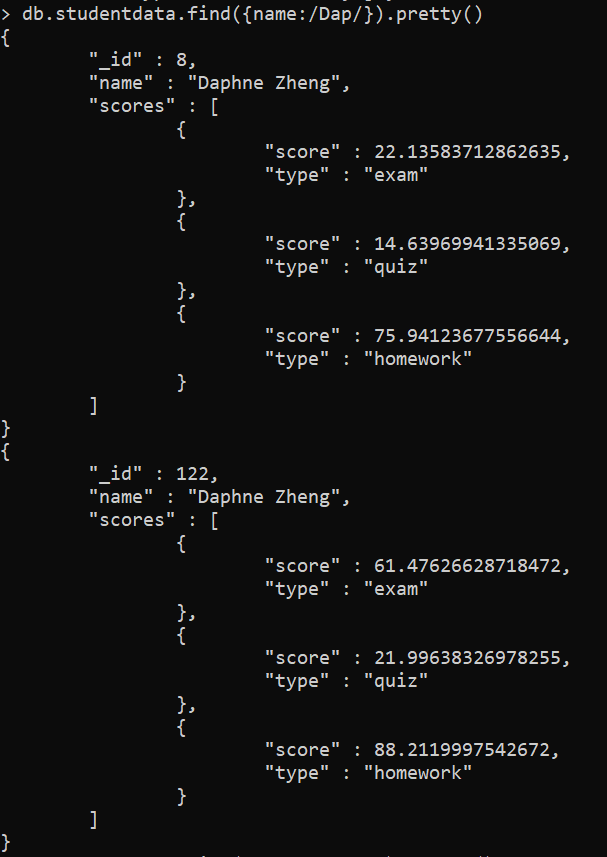
We used database Students

With collection studentdata

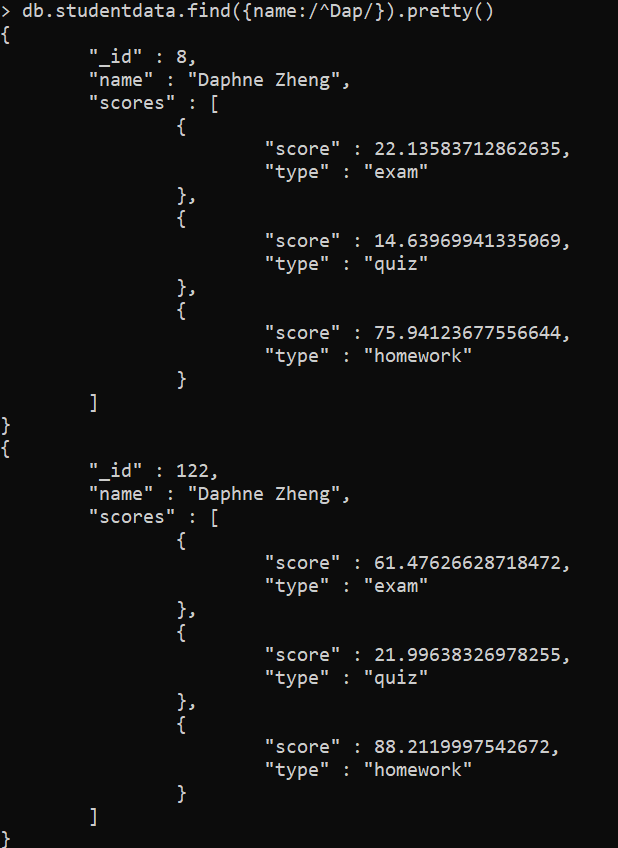




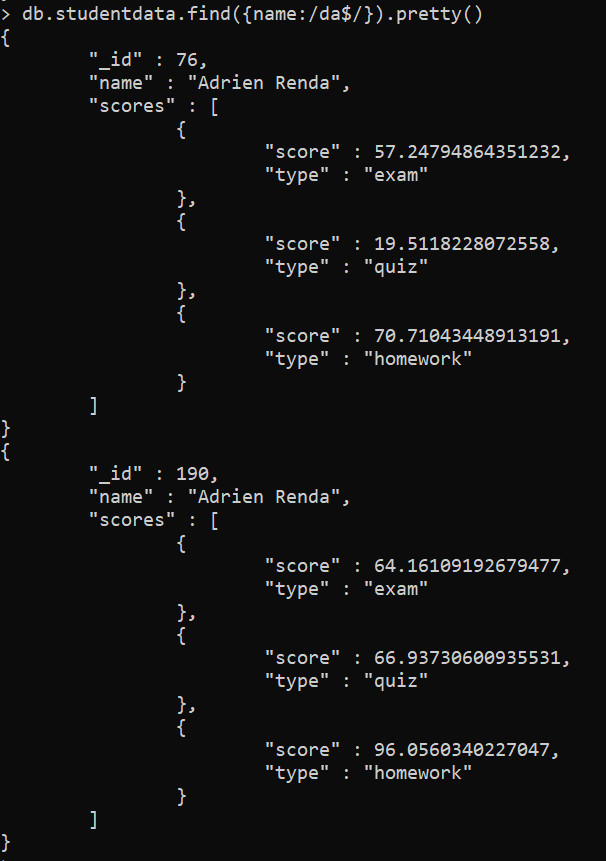




Any word having Dap

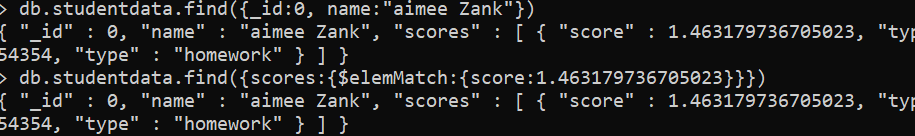


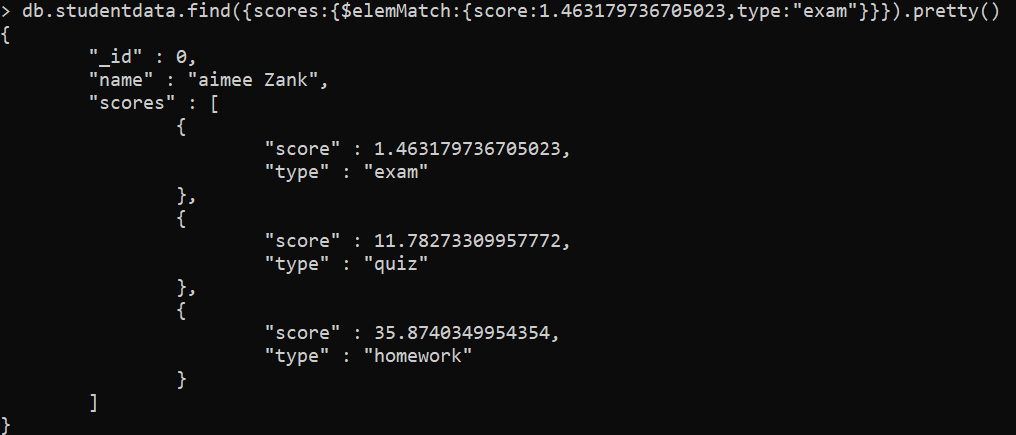
At the beginning Dap is present



At the end da is present







Operators:

$gte = greater than equal to,

$gt = greater than,

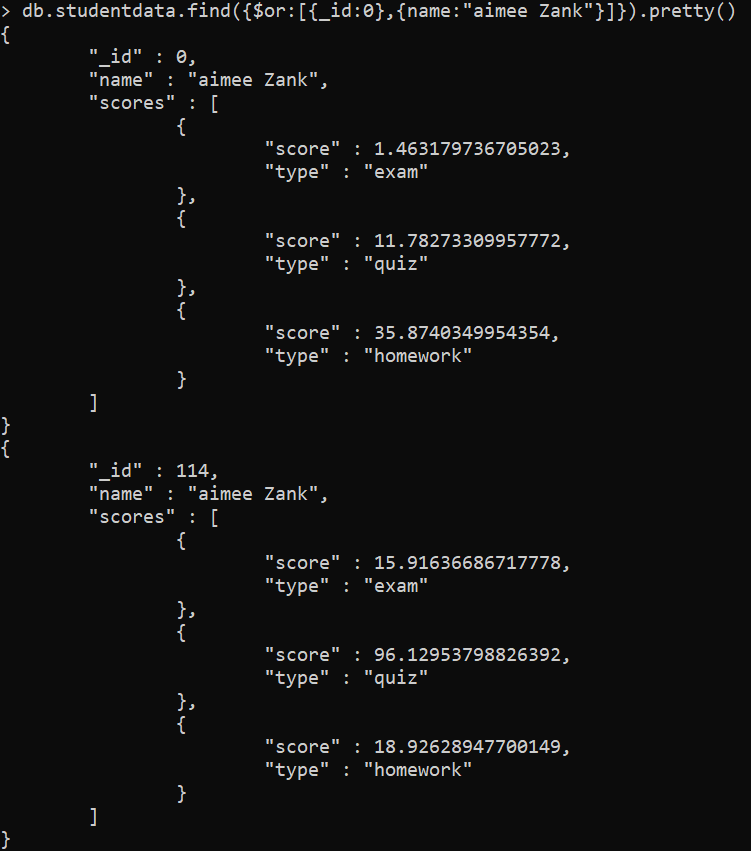
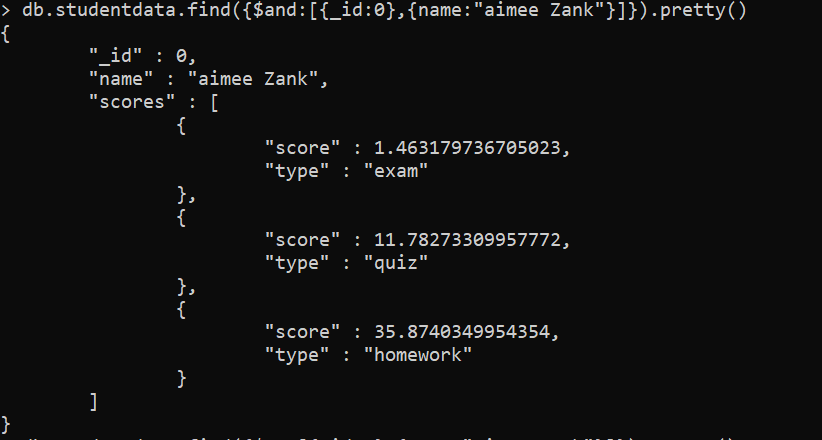
$lte = less than equal to,

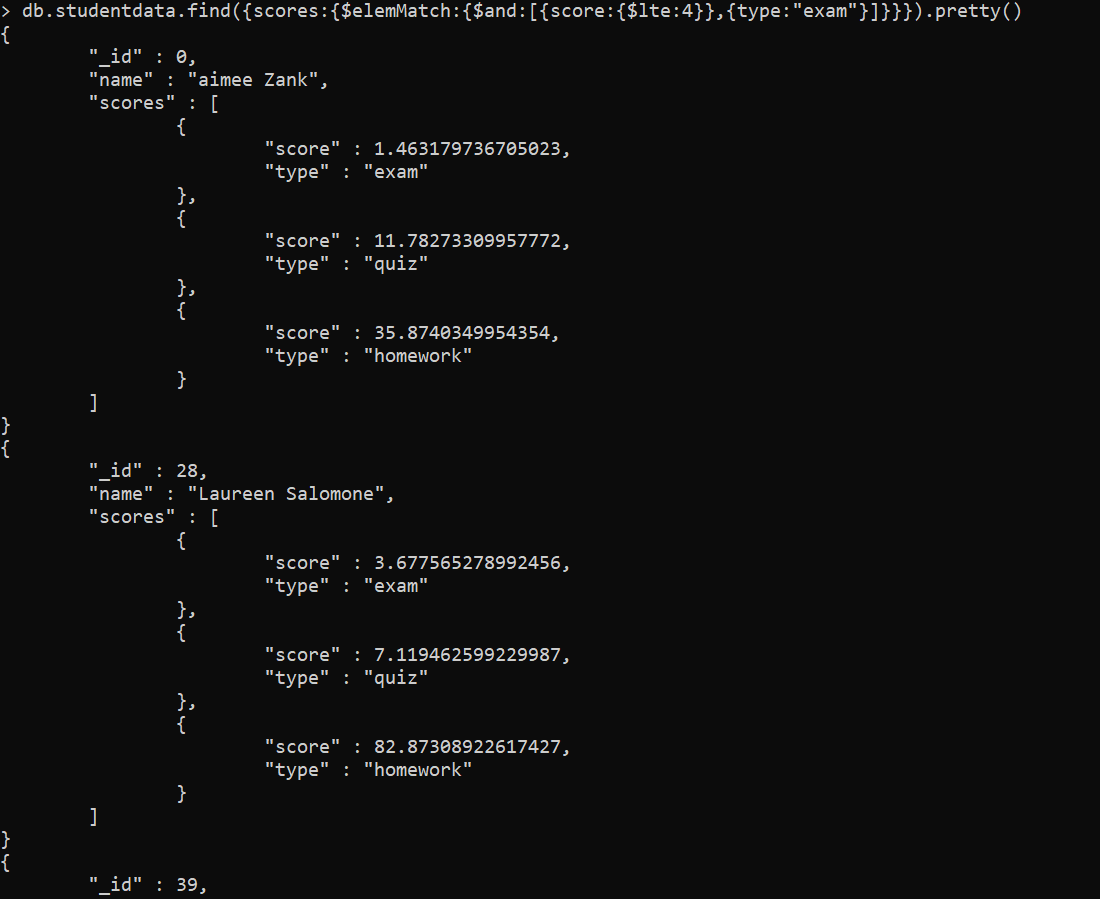
$lt = less than,

$eq = equal to,

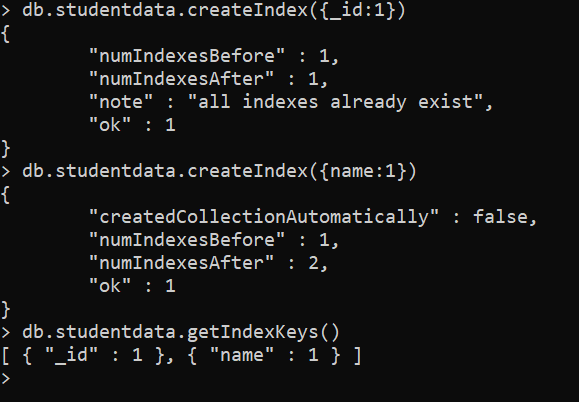
$in = in,

$nin = not in





Indexing:

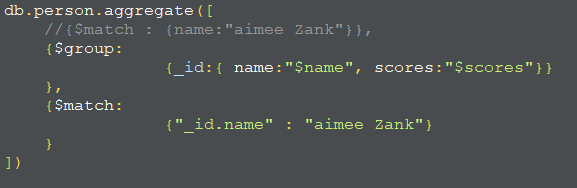


Data modelling

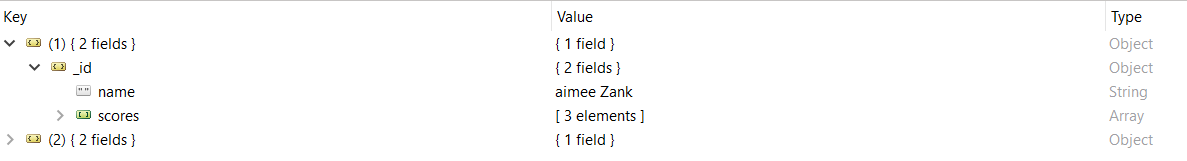
Aggregation



Installed ROBO 3T

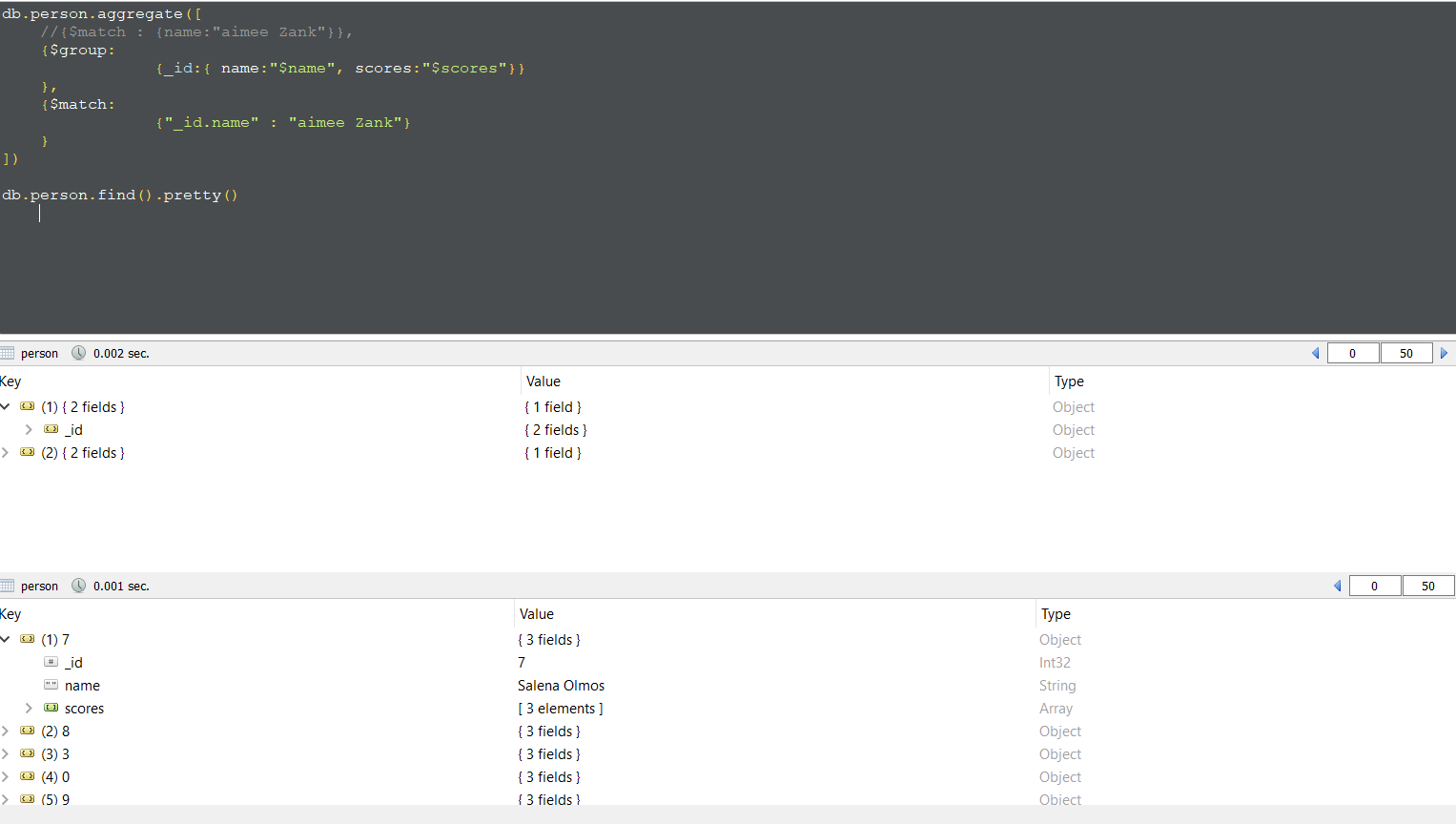


Output:



Similar to SQL: groupby having

If performing two commands:



Mongoose(ODM) – NODE Js

Converts js file(mongodb) to mongo compatible file.