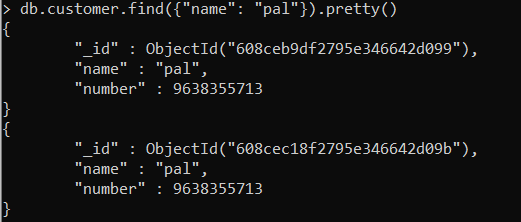
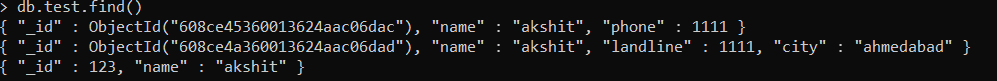
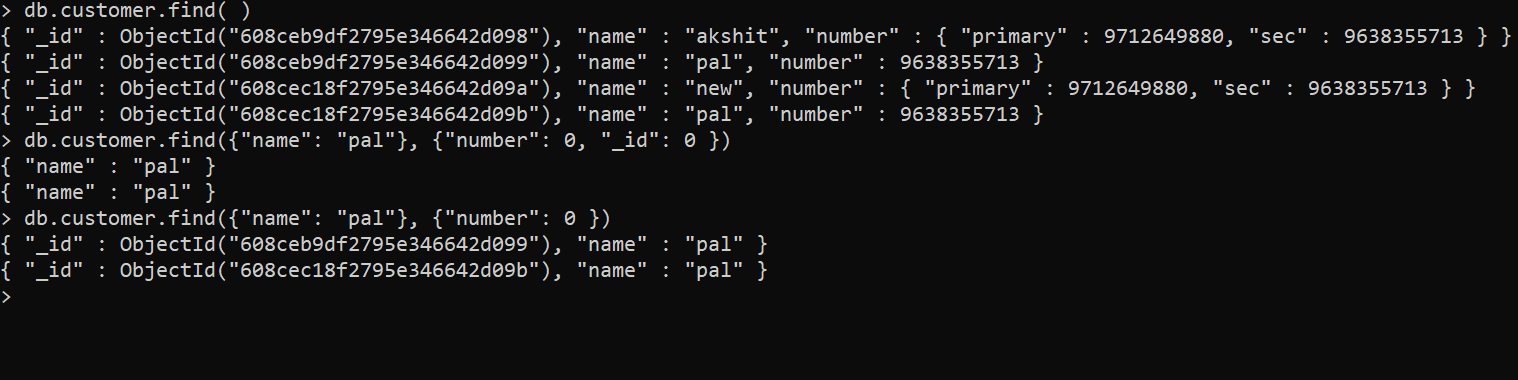
* In database world most important thing is management of data.
* By structuring data into some another format we can increase our I/O speed with same amount of data.
* In database world file known as table & folder is known as database.
* Sometimes we don’t stick with our schema.
* Eg, we are collecting name no & city of customer.
  + But one customer don’t has mobile number instead of this he has land line number.
  + But we have to manage this also somehow.
  + So here on the fly we can create a column for this thing.
  + Here we are not going with out default schema.
  + It is known as **schema less**.
* In mongodb world record is known as document.
* Mongodb is **flexible document oriented database**.
* Mongodb server command name is mongod & client command is mongo.
* Add this in the path environment.
  + C:\Program Files\MongoDB\Server\4.4\bin
* Commands
  + show dbs
    - it will show you all tables.
  + use ak
    - they will create a database and switch to that database.
  + Show collection
  + Db.createCollection(“test”)
    - It will create collection with name test.
    - Now you can see db is showing inside show dbs.
  + Db.test.insert("name" : "akshit", "phone" : 1111)
    - It will create a document for you.
    - Record
  + db.test.find()
    - it is like select \* from table.
* Discussion in JSON format.
  + In mongo db we most of time store document inside document.
  + This is known as nested or embedded document.
  + db.customer.insert( [ { "name": "akshit", "number": { "primary": 9712649880, "sec": 9638355713} }, { "name": "pal", "number": 9638355713 } ] )
  + db.customer.find({"name": "pal"})
* There are some pre defined variable.
  + Like gt, lt and many more.
  + For using pre defined variable we have to use this with $.
  + db.customer.find({"name": "pal"}).pretty()
    - It will format your output.



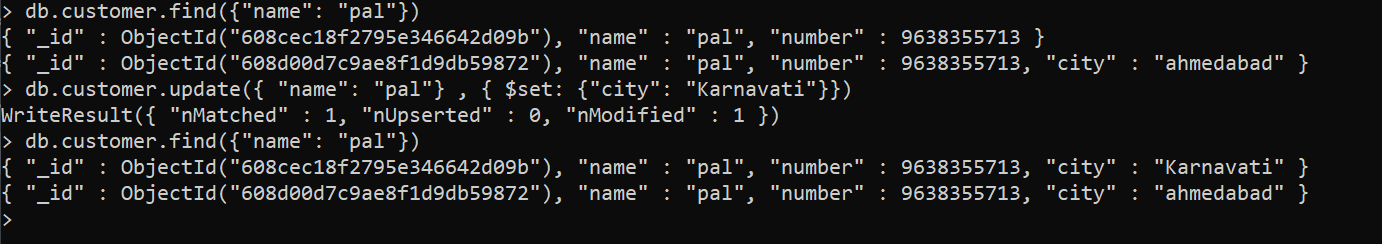
* + Sometimes we want to retrieve only some column from document.
  + For this we can use this thing.
    - db.customer.find({"name": "pal"}, {“city”: 0} )
      * In output we don’t have city column.
      * It will give you all details apart from city.
    - db.customer.find({"name": "pal"}, {“city”: 1} )
      * It will give you only city column.
  + Mongodb automatically assign unique for record.
  + If you want assign by yourself then you can use thus.
    - db.customer.insert({"name": "pal", “\_id”: 123} )



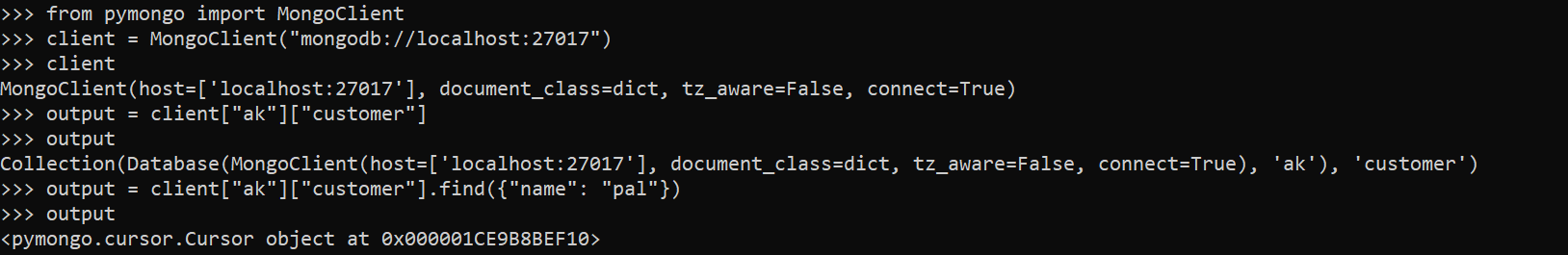
* + if you don’t want id to come you have to manually remove it.



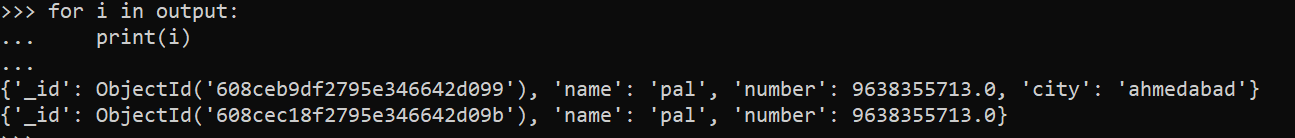
* You can update it using this syntax.
  + If you don’t use $set, it will update (replace) complete document with this document.



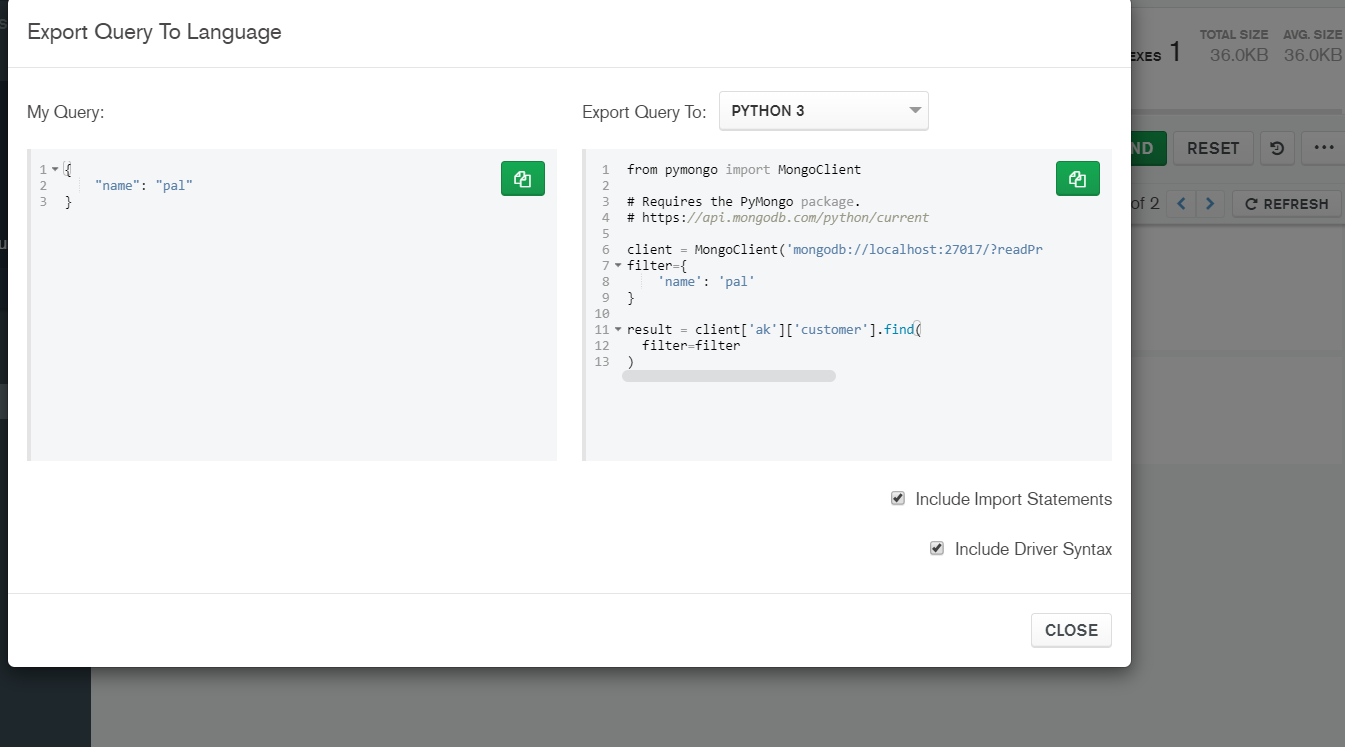
* for connecting to mongoDB we have 3 ways.
  + CLI
  + GUI
    - compass
  + API
* We can connect GUI using compass.
* For connecting to API we are using python language.
  + We have to install pymongo library.
    - Pip install pymongo



* + Here you can see find is basically finding cursor for that requirement.
  + It will giving cursor for you.
  + Now we can print this using for loop.



* You can also get this python code from query using compass.
  + On compass select export to language.



* Instead of creating json code manually, we can get json from somewhere and import in out system.
  + For this you have to download mongodb database tools.
  + And after this you have to also set path for this.
* Check for how to show data without id (18:00)
* Custom fetch from database.