Для користування даними командами необхідно на диску «С:\» створити папку «data», в якій створити дерикторію «db»

1. "C:\Program Files\MongoDB\Server\4.4\bin\mongod.exe" --dbpath="c:\data\db"
2. "C:\Program Files\MongoDB\Server\4.4\bin\mongo.exe"

Для зручного спостерігання за базами можна використовувати «MongoDB Compass».

MongoDB Compass – це прикладна програма, яка дозволяє не лише спостерігати за змінами в базах даних, а й вносити зміни в них: створювати, редагувати, видаляти як складові баз даних так і самі бази.

**Для створення бази даних достатньо використати команди:**

* “use database.name” – база даних з назвою
* “db.createCollection(“collection.name”)” – колекція з назвою

**Для модифікації колекцій можна використовувати:**

* “db.insertOne({some fields})” – додати поле до колекції.
* “db.insertMany{[{}, {}, {}]}” – додати множинні поля до колекції.

**Для фільтрації об’єктів можна використовувати такі команди:**

* “db.collection.name.find()” – віддає всі поля з колекції.
* “db.collection.name.find().limit(number)” – віддає всі об’єкти з колекції «collection.name».
* “db.collection.name.find({}, {\_id: 0})” – віддає всі об’єкти з колекції «collection.name» без поля id.
* “db.collection.name.find().sort({age: 1, name: 1}).limit(number)” – сортування лімітованого результату виводу по age та name.
* “db.collection.name.find({$or: [{age: 20}, {status: “is studying”}]})” – знайти та вивести об’єкти, які мають поле age = 20 або status = is studying
* “db.collection.name.find($or: [{age: {$lt: 40}}, {status: “is studying”}])” - знайти та вивести об’єкти, які мають поле age < 40 або status = is studying

На місці $lt могли стояти наступні модифікатори:

1. $lte (less than equal: “<=”)
2. $gt (greater than: “>”)
3. $gte (greater than equal: “>=”)
4. $eq (equal: “==”)
5. $ne (not equal: “!=”)

* “db.collection.name.find({name: {$in: [“name\_1”, “name\_2”]}, {\_id: 0}}” – знайти об’єкти в яких присутнє ім’я “name\_1” або “name\_2” та вивести їх без поля ідентифікатора.
* На місці $in міг стояти модифікатор $nin (не присутнє ім’я):
* “db.collection.name.find({birthday: {$exists: true}}, {\_id: 0})” – всі об’єкти які мають поле birthday
* “db.collection.name.find({friends: {$elemMatch: {$lte: "Z"}} }, {\_id: 0})” – знайти об’єкти, що містять у полі friends елементи, які починаються з “Z”(код символу) або молодших символів.

**Для проведення операцій над об’єктами (модифікація, видалення) можна використовувати наступні команди:**

* “db.collection.name.updateOne({age: 21}, {$set: {course: 3, status: “is studying”}})” – знаходить **перший!** об’єкт з age: 21 і модифікує поля course: 3, status: “is studying”
* “db.collection.name.updateMany({age: 21}, {$set: {course: 3, status: “is studying”}})” – знаходить **всі!** об’єкт з age: 21 і модифікує поля course: 3, status: “is studying”
* “db.collection.name.replaceOne({age: 23}, {name: “Harry”, “last name”: ”Potter”, course: 6, “special skills”: “magic”, age: 23})” – заміна першого об’єкта (в базі) із age: 23, якому надають поля: name: “Harry”, “last name”: ”Potter”, course: 6, “special skills”: “magic”, age: 23.
* “db.collection.name.deleteOne({age: {$gt: 23}, age: {$lt: 35}})” – видаляє **перший** об’єкт у базі даних в якому вік входить в задані рамки, а саме: 23 < age < 35

**Для проведення множини операцій над об’єктами (модифікація, видалення) можна використовувати наступні команди:**

* “db.collection.name.bulkWrite([{insertOne: {“document”: {name: “David”, age: 25, birthday: “20.03.1999”}}}, {deleteOne: {filter: {name: “Yura”}}}, {updateOne: {filter: {name: “Roman”}, update: {$set: {name: “Romashka”}}}}, {replaceOne: {filter: {name: “”}}}])