Mongo DB  
mongo atlas

Швидка, не потрібно впорядкованості, менш секюрна

Use mydb  
вставляєм дані з репозиторія  
дивимся текстом, bson об’єкт (схоже на json)

db.users.find() – все  
db.users.inserOne({car:’bmv’, model:’s3’}) –добавив філди  
db.users.inserMany({})  
db.users.find({gender:’mail’}) – шукаєм жінок  
db.users.find({\_id:ObjectId(’стрінгу ID’)}) – шукаєм по ІД  
db.users.find({gender:’mail’, rating:2}) - and  
db.users.find(  
 {gender:’mail’, rating:2} виключає щось  
 {id:false, })

db.users.find({  
 $or:[  
 { rating:2}  
 { rating:5}  
 ]  
})

db.users.find({  
 $and:[  
 { rating:2}  
 { age:34}  
 ]  
})

db.users.find({age:{$in:[25,12,34]}})

db.users.find({age:{$nin:[25,12,34]}})

db.users.find({age:{$not:{$in: [25, 12]}}})

db.users.find({name:/^a/i}) – починається з а, i виключає регістр

db.users.find({name:/ a$/i}) – закінчується на а

db.users.find({name:/ ^a.\*x$/i}) - починається з а закінчується х, .\* пропускає все

db.users.find({name:/ ^.{2}t/i}) - третя буква t

db.users.find({name:/ .\*v.\*/i}) – шукаємо десь в середині v

db.users.find().sort({rating:1}) –від меншого до більшого, -1 навпаки

db.users.find().sort({rating:1, age:-1}) – якщо є однакові, то сортує їх по age

db.users.find().sort({rating:1, age:-1}).limit(2) – взяти перші 2

db.users.find().sort({rating:1, age:-1}).limit(2).skip(2) – припустити 2 з лімітом 2

db.users.find(rating:{$gt:2}) – >

db.users.find(rating:{$lt:2}) – <

db.users.find(rating:{$lte:2}) - <=

db.users.find(rating:{$gte:2}) - >=

db.users.find(rating:{$ne:2}) - !=

db.users.find(rating:{$eq:2}) - ==

db.users.updateOne

db.users.updateMany(

{gender:’male’},

{$set{gender:’animal’}}

)

db.users.find({parents:{size:1}}) – шукаєм у кого масив 1 елемент

db.users.find({parents:{$exists:0}}) – всі у кого немає поля parents , 1 в кого є

db.users.updateMany(

{parents:{$exists:1}}

{$set:{‘parents.$[]’ : ’dashto’}} -цу кого є батьки переіменували

)

db.users.updateOne({\_id:ObjectId(‘id\_user’)}, - додаєм в масив noda

{$push:{studing: ‘noda’}}  
)

db.users.updateOne({\_id:ObjectId(‘id\_user’)}, -видаляєм з масива джава

{$pull:{studing: ‘java}}  
)

Db.users.find({‘address.city’ : ‘Lviv’}) – достукатись до об’єктів

Db.users.find({parents:null}) – находить всіх у кого немає parants – але не вивеле у кого так вставлено

Db.users.remove

Db.users.deleteOne(\_id:ObjectId(‘User\_id’)})

Db.users.deleteMany

Агрегації

Db.teacher.aggregate(  
[  
{},  
{},  
{}  
])

Db.teacher.aggregate(  
[  
 {  
 $group:{\_id:’’},  
 minmin:’$payment’}   
 max:{$min:’$payment’} - поле в дужках і спереді $  
 avg:{$avg:’$payment’}  
 sum:{$sum:’$payment’}  
 }  
])

Db.teacher.aggregate(  
[  
 {  
 $group:{\_id:’$lesson’},  
 minmin:’$payment’}   
 max:{$min:’$payment’} - поле в дужках і спереді $  
 avg:{$avg:’$payment’}  
 sum:{$sum:’$payment’}  
 count:{$sum:1} – кількість по групованого(множник) – lesson  
   
 } ,  
 {$sort:{avg: -1}} – сюди приходить попереднє значення  
])

Db.teacher.aggregate(  
[  
 {  
 $group:{  
 \_id:’$lesson’,  
 count:{$sum:1}  
 }  
 } ,   
{$sort:{\_id:1}},  
{  
 $project:{  
 lesson:$\_id’’,  
 count:1,  
 \_id:0  
 }  
}  
  
])

Зв’язки

Db.teacher.aggregate([  
 {  
 $match:{  
 class\_curator:{$gt:3}  
 },  
 $lookup:{  
from:’students’,  
localField:’clas\_curator’,  
foreignField:’class’  
as:’my\_students’  
 } – підєднує студентів вчителю в масиві

},  
 {  
 {$skip:2}  
 },  
 {  
 $limit:2  
 },  
   
])

Db.teacher.aggregate([  
 {  
 $match:{  
 class\_curator:{$gt:3}  
 },

$lookup:{  
from:’students’,  
localField:’clas\_curator’,  
foreignField:’class’  
as:’my\_students’  
 },  
{  
 $match:{  
 ‘my\_students.parents.profession’: ‘programmer’  
 }  
  
}  
}

])

Db.teacher.aggregate([  
{  
 $match:{  
 class\_curator:{$gt:5}  
},

{$lookup:{  
from:’students’,  
localField:’clas\_curator’,  
foreignField:’class’  
as:’my\_students’  
 }  
},  
{  
 {$inwind:’$my\_students ‘}, – по 1 студенту  
 {$inwind:’$my\_students.parents ‘},  
 { $match:{’ my\_students.parents’:’doctor’}},  
 {$count:’counter’}  
}

])