docker ps -a = check docker jalan atau engga

docker exec -it mongodb bash = run mongodb bash

mongosh = masuk ke mongoshell

root@bd08fa1fb4b6:/# mongosh -u reynard -p(enter) lalu masukin password = ini buat masukin authetification

**QUERY MONGOSHELL**

* show dbs
* use example(nama database)
* db.example(nama collection).find
* console.clear();
* use perpusdb; (ngebuat db baru)
* db.buku.insertOne({

judul:"Waktu yang salah",

tahun:2019

})

* db.buku.updateOne(

{ \_id: ObjectId("6655858026db51077cabfa79") },

{ $set: { tahun: 2000, judul: "bebek" } }

)

* db.buku.deleteOne({\_id:ObjectId('6655858026db51077cabfa79')})
* show collections
* db.dropDatabase()
* db.buku.findOne({\_id:1}, {name: 1, \_id: 0}) = parameter kiri(find name = 1), parameter kanan(show berdasarkan name saja, dan \_id jangan dimunculin),

kalo mau 0 berarti 0 semua, kalo 1 berarti 1 semua. kalo 1 semua berarti munculin itu doang, kalo 0 berarti kita ignore dia doang. untuk \_id itu pengecualian boleh 1 atau 0

* db.buku.find({}, {

name: 1,

"inventory.waktu": 1

})

* db.buku.find({

price: {$lt: 799}

})less than

* db.buku.find({

price: {$lte: 799}

})less than equal

* db.buku.find({

price: {$gt: 799}

})greater than

* db.buku.find({

price: {$gte: 799}

})greater than equal

* db.buku.find({

price: {$eq: 799}

})equal

* db.buku.find({

price: {$ne: 799}

})not equal

* db.buku.find({

price: {$in: [799, 299]}

})in

* db.buku.find({

&and:[

{price: 899},

{color: {$in: ["white", "black"]}}

]

})and

* db.buku.find({

&or:[

{price: 899},

{price: 10},

{color: {$in: ["white", "black"]}}

]

})or

* db.buku.find({

price: {$not: {$gt: 10}}

})not

kalo dia engga ada price nya, yahh dia tetep masuk, karena not tapi kalo dia lte dia cumin bandingin price nya doang

* db.buku.find({

$nor:[

{price: 899},

{color: "gold"}

]

})nor

bedanya not = mengembalikan nilai tunggal maksudnya cuman 1 validasi

kalo nor = validasinya bisa banyak

* db.buku.find({

age: {$nin: 25, 30}

})nin

sama kayak in tapi ini not

* db.buku.find({

{price: {$exists: true, $gt : 200}}

})exists + greater than

dia cek, kalo dia true, berarti tampilin meskipun dia null bakal tetep di tampilin, sedangkan kalo engga ada tulisan price samsek dia baru ga munculin

**TYPE**

https://www.mongodb.com/docs/manual/reference/operator/query/type/

check type

* db.buku.find({

price: {$type: "number"}

})

* db.buku.find({

price: {$type: ["number", "string"]}

})

bisa tambahin array

* db.buku.find({

color: {$size: 2}

})size

check kalo di document yg ada arraynya, dia cek isinya itu harus 2

* db.buku.find({

price:{$exists: 1}

}).sort({price:1})

sorting price dari pertama null, paling kecil ke paling besar, terakhir string

1. .sort({price: 1}) kecil-besar
2. .sort({price: -1}) besar-kecil
3. .sort({price: -1, name: 1}) kombinasi

* db.buku.find({

price:{$exists: 1}

}).sort({price:1}).skip(2).limit(1)

limmitasi hanya return 1 document, kalo misalnya dia ada 10 data, dia skip 2 lalu limitasi 1 doang

* db.buku.updadteOne({

id: 4

}, {

$set:{"spec.ram": 16, "spec.screen": 10.5}

})

cara update untuk object

* db.buku.updadteOne({

id: 4

}, {

$set:{"storage.0": 16, "storage.1": 32}

})

cara update untuk array

* db.buku.updadteOne({

id: 4

}, {

$inc:{storage.0 : 50}

})

increment, nilai sebelum diupdate dan ditambah 50

**min, mul(multiple), max, unset(hapus), rename(ganti nama field)**

**AGREGATION**

AVG, SUM, MIN, MAX, COUNT

* db.orders.aggregate([

{ $match: { status: "A" } },

{

$group: {

\_id: "$cust\_id",

totalAmount: { $sum: "$amount" },

avgAmount: { $avg: "$amount" }

}

},

{ $sort: { totalAmount: -1 } },

{ $limit: 10 } // Mengambil 10 dokumen teratas

])

**Indexes**

* db.book.getIndexes()
* db.book.createIndex({title:1})
* db.book.createIndex({title:1, unique:true})
* db.book.createIndex({title:1, Waktu:1, unique:true})
* db.book.dropIndex("status\_1") -> dapet dari name dari getIndexes