Practice Data: <https://github.com/Apollo-Level2-Web-Dev/mongodb-practice>

**6-0 Intro the powerful aggregation framework**

**6-1 $match , $project aggregation stage**

Google -> Aggregation Pipeline Operators

-> mongodb aggregation tutorial

Like: <https://studio3t.com/knowledge-base/articles/mongodb-aggregation-framework/>

<https://www.mongodb.com/docs/manual/aggregation/>

db.test.aggregate([

// stage-1

{ $match: { gender: "Male", age: { $lt: 35 } } },

// stage-2

{$project: {name: 1, age: 1, gender:1}}

])

**6-2 $addFields , $out , $merge aggregation stage**

db.test.aggregate([

// stage-1

{ $match: { gender: "Male", age: { $lt: 35 } } },

// stage-2

{ $addFields: { course: "level-2" } },

// stage-3

{ $project: { name: 1, gender: 1, course: 1 } }

])

db.test.aggregate([

// stage-1

{ $match: { gender: "Male", age: { $lt: 35 } } },

// stage-2

{ $addFields: { course: "level-2", eduTech: "Programming Hero" } },

// stage-3

{ $project: { name: 1, gender: 1, course: 1, eduTech: 1 } }

])

db.test.aggregate([

// stage-1

{ $match: { gender: "Male", age: { $lt: 35 } } },

// stage-2

{ $addFields: { course: "level-2", eduTech: "Programming Hero", monerMoto: "Moner Iccha" } },

// stage-3

{ $project: { name: 1, gender: 1, course: 1, eduTech: 1 } },

// stage-4

{ $out: "course-students" }

])

db.test.aggregate([

// stage-1

// { $match: { gender: "Male", age: { $lt: 35 } } },

// stage-2

{ $addFields: { course: "level-2", eduTech: "Programming Hero", monerMoto: "Moner Iccha" } },

// stage-3

// { $project: { name: 1, gender: 1, course: 1, eduTech: 1 } },

// stage-4

// { $out: "course-students" }

{ $merge: "test" }

])

db.test.find({}).project({ course: 1, eduTech: 1, monerMoto: 1 })

**6-3 $group , $sum , $push aggregation stage**

Find more documentation on google , try to read many documentation of different important topics ( according to your choice )

db.test.aggregate([

// stage-1

{ $group: { \_id: "$gender" } }

])

db.test.aggregate([

// stage-1

// { $group: { \_id: "$gender" } }

// { $group: { \_id: "$course" } }

{ $group: { \_id: "$address.country" } }

])

db.test.aggregate([

// stage-1

// { $group: { \_id: "$age", count: {$sum: 1} } }

{ $group: { \_id: "$gender", count: {$sum: 1} } }

])

db.test.aggregate([

// stage-1

{ $group: { \_id: "$address.country", count: { $sum: 1 }, amakeDekhaoName: { $push: "$name" } } }

])

db.test.aggregate([

// stage-1

{

$group:

{

\_id: "$address.country",

count: { $sum: 1 }, fullDoc: { $push: "$$ROOT" }

}

},

// stage-2

{$project: {

"fullDoc.name": 1,

"fullDoc.email": 1,

"fullDoc.phone": 1,

}}

])

**6-4 explore more about $group & $project**

db.test.aggregate([

// stage-1

{

$group:

{

\_id: null,

totalSalary: { $sum: 1 }

}

},

])

db.test.aggregate([

// stage-1

{

$group:

{

\_id: null,

totalSalary: { $sum: "$salary" }

}

},

])

db.test.aggregate([

// stage-1

{

$group:

{

\_id: null,

totalSalary: { $sum: "$salary" },

maxSalary: { $max: "$salary" },

minSalary: { $min: "$salary" },

avgSalary: { $avg: "$salary" },

}

},

// stage-2

{

$project: {

totalSalary: 1,

maxSalary: 1,

minSalary: 1,

// avgSalary: 1,

avgSalary: "$avgSalary",

rangeBetweenMaxAndMin: {$subtract: ["$maxSalary","$minSalary"]}

}

}

])

**6-5 Explore $group with $unwind aggregation stage**

db.test.aggregate([

// stage-1

{ $unwind: "$friends" },

// stage-2

{

$group: { \_id: "$friends", count: { $sum: 1 } }

}

])

db.test.aggregate([

// stage-1

{ $unwind: "$interests" },

// stage-2

{ $group: { \_id: "$age", interestsPerAge: {$push: "$interests"} } }

])

Docs: <https://studio3t.com/knowledge-base/articles/mongodb-aggregation-framework/#mongodb-group>

**6-6 $bucket, $sort, and $limit aggregation stage**

Google -> $bucket in mongodb

Necessary doc: <https://www.mongodb.com/docs/manual/reference/operator/aggregation/bucket/>

db.test.aggregate([

// stage-1

{

$bucket: {

groupBy: "$age",

boundaries: [20, 40, 60, 80],

default: "greater than 80",

output: {

count: { $sum: 1 },

// karakaraAse: {$push: "$name"}

karakaraAse: {$push: "$$ROOT"}

}

}

}

])

db.test.aggregate([

// stage-1

{

$bucket: {

groupBy: "$age",

boundaries: [20, 40, 60, 80],

default: "greater than 80",

output: {

count: { $sum: 1 },

// karakaraAse: {$push: "$name"}

karakaraAse: { $push: "$$ROOT" }

}

}

},

// stage-2

{

$sort: { count: -1 }

},

// stage-3

{

$limit: 2

},

// stage-4

{

$project: { count: 1 }

}

])

**6-7 $facet, multiple pipeline aggregation stage**

db.test.aggregate([

{

$facet: {

// pipeline-1

"friendsCount": [

// stage-1

{ $unwind: "$friends" },

{ $group: { \_id: "$friends", count: { $sum: 1 } } }

],

// pipline-2

"educationCount": [

// stage-1

{ $unwind: "$education" },

// stage-2

{ $group: { \_id: "$education", count: { $sum: 1 } } }

]

}

}

])

db.test.aggregate([

{

$facet: {

// pipeline-1

"friendsCount": [

// stage-1

{ $unwind: "$friends" },

{ $group: { \_id: "$friends", count: { $sum: 1 } } }

],

// pipline-2

"educationCount": [

// stage-1

{ $unwind: "$education" },

// stage-2

{ $group: { \_id: "$education", count: { $sum: 1 } } }

],

// pipeline-3

"skillsCount": [

// stage-1

{ $unwind: "$skills" },

// stage-2

{ $group: { \_id: "$skills", count: { $sum: 1 } } }

]

}

}

])

**mongodb-practice**

<https://github.com/Apollo-Level2-Web-Dev/mongodb-practice/tree/main>

**6-8 $lookup stage, embedding vs referencing**

db.orders.aggregate([

{

$lookup: {

from: "test",

localField: "userId",

foreignField: "\_id",

as: "user"

}

}

])

**6-9 What is indexing, COLLSCAN vs IXSCAN**

db.test.find({ \_id: ObjectId("6406ad63fc13ae5a40000066") }).explain()

db.test.find({ \_id: ObjectId("6406ad63fc13ae5a40000066") }).explain("executionStats")

db.test.find({ "email": "omirfin2@i2i.jp", }).explain("executionStats")

**{ email** :  "gofffrye@flexigen.com" } -> paste > mongodb compass -> search bar : > find > explain

**6-10 Explore compound index and text index**

db.getCollection("massive-data").find({ $text: { $search: "dolor" } }).project({ about: 1 })

Practice Data: <https://raw.githubusercontent.com/Apollo-Level2-Web-Dev/mongodb-practice/main/massive-data.json>

Task Link: <https://drive.google.com/file/d/14Bl2h_ctiAmVB-9Xvb0Z9DAuELfDhXv0/view?usp=drive_link>

Solution: <https://github.com/Apollo-Level2-Web-Dev/practice-tasks-2-solutions>