Discussion 5.1 Aggregate Operations

Shane Hingtgen

4/10/23

Bellevue University

Aggregate Operations are one way to filter a lot of documents within the collection. (Studio 3T, 2023) These are stages that can filter, sort, group, reshape or even modify the documents, Also known as the pipeline. (Studio 3T, 2023). According to studio3t.com, the typical pipeline for MongoDB is input -> $match -> $group -> $sort -> output.

Diagram

Description automatically generated

(Studio 3T, 2023)

Different types of aggregation include $match, $group, $sort and $project. According to MongoDB $project “passes along the documents with the requested fields to the next stage in the pipeline.” With this you can include certain fields with out the \_id field. Also if you try to include a field that isn’t in the document it will ignore the field and it will not be added to the document (*$Project (Aggregation) — MongoDB Manual*, n.d.). A good use of this aggregate is for searching in a collection of books. An example of $project:

//This is an example of a book in the DB

{

"\_id" : 1,

title: "Jurassic Park",

isbn: "0345538986",

author: { last: "Crichton", first: "Michael" },

copies: 4

}

//Here we are looking a book in the document

db.books.aggregate( [ { $project : { \_id: 1, title : 1 , author : 1 } } ] )

//This is the output

{ "title" : " Jurassic Park ", "author" : { "last" : " Crichton ", "first" : " Michael " } }

For the second aggregate $match its purpose is filter and use only the conditions that match. With $match it will not be able to use raw aggregation expressions. (*$Match (Aggregation) — MongoDB Manual*, n.d.) Some restrictions include not being able to use $where with $match, not being able to use $near or $nearSphere, but you can use $geoNear, $geoWithin along side $center and $centerSphere. (*$Match (Aggregation) — MongoDB Manual*, n.d.) An example of code:

{ $match: { $expr: { <aggregation expression> } } }

Lastly the $sort aggregation will organize through the input documents and will return the data to the pipeline in the order you select ascending or descending. (*$Sort (Aggregation) — MongoDB Manual*, n.d.). This is useful since MongoDB doesn’t store the documents inside the collection in any order. (*$Sort (Aggregation) — MongoDB Manual*, n.d.)

{ $sort: { <field1>: <sort order>, <field2>: <sort order> ... } }

References:

*Aggregation Operations — MongoDB Manual*. (n.d.). MongoDB. Retrieved April 10, 2023, from https://www.mongodb.com/docs/manual/aggregation/

Studio 3T. (2023, February 27). *MongoDB Aggregation: tutorial with examples and exercises | Studio 3T*. Retrieved April 10, 2023, from https://studio3t.com/knowledge-base/articles/mongodb-aggregation-framework/

*$project (aggregation) — MongoDB Manual*. (n.d.). MongoDB. Retrieved April 10, 2023, from https://www.mongodb.com/docs/manual/reference/operator/aggregation/project/

*$match (aggregation) — MongoDB Manual*. (n.d.). MongoDB. Retrieved April 10, 2023, from https://www.mongodb.com/docs/manual/reference/operator/aggregation/match/

*$sort (aggregation) — MongoDB Manual*. (n.d.). MongoDB. Retrieved April 10, 2023, from https://www.mongodb.com/docs/manual/reference/operator/aggregation/sort/