How can we use the $substr function in MongoDB?

Authored by

stats writer

June 30, 2024

The $substr function in MongoDB is a powerful tool that allows users to extract a substring from a given string or text field. This function takes in three parameters: the target string, the starting index, and the length of the desired substring. By specifying these parameters, users can easily manipulate and extract specific parts of a string, making it a valuable tool for data analysis and manipulation in MongoDB. This function can be used in various scenarios, such as retrieving specific characters from a name or extracting a portion of a URL. Its flexibility and ease of use make it an essential function for efficient data processing in MongoDB.

MongoDB: Use the $susbtr Function

You can use the $substr function in MongoDB to extract a substring from a string.

This function uses the following basic syntax:

db.myCollection.aggregate(}}}
])

This particular example extracts the four characters from the field titled "fullString" starting from position 0.

The following example shows how to use this syntax in practice with a collection sales with the following documents:

db.sales.insertOne({yearMonth: 201702, amount: 40})
db.sales.insertOne({yearMonth: 201802, amount: 32})
db.sales.insertOne({yearMonth: 201806, amount: 19})

db.sales.insertOne({yearMonth: 201910, amount: 29})
db.sales.insertOne({yearMonth: 201907, amount: 35})

Example: How to Use the $susbtr Function in MongoDB

We can use the following code to extract the first four characters from the "yearMonth" field and display it in a new field titled "year":

db.sales.aggregate(}}}
])

This code produces the following output:

{ _id: ObjectId("620145544cb04b772fd7a929"), year: '2017' }
{ _id: ObjectId("620145544cb04b772fd7a92a"), year: '2018' }
{ _id: ObjectId("620145544cb04b772fd7a92b"), year: '2018' }
{ _id: ObjectId("620145544cb04b772fd7a92c"), year: '2019' }
{ _id: ObjectId("620145544cb04b772fd7a92d"), year: '2019' }

Notice that the first four characters from the "monthYear" field in each document are displayed in a new field titled "year."

It's important to note that this code only *displays* the substring.

To actually add a new field to the collection that contains this substring, we must use the $merge function as follows:

```
db.sales.aggregate( }}},
{ $merge: "sales" }
])
```

Here's what the updated collection now looks like:

```
{ _id: ObjectId("620145544cb04b772fd7a929"),
yearMonth: 201702,
amount: 40,
year: '2017' }
{ _id: ObjectId("620145544cb04b772fd7a92a"),
yearMonth: 201802,
amount: 32,
year: '2018' }
```

{ _id: ObjectId("620145544cb04b772fd7a92b"),
yearMonth: 201806,
amount: 19,
year: '2018' }
{ _id: ObjectId("620145544cb04b772fd7a92c"),
yearMonth: 201910,
amount: 29,
year: '2019' }
{ _id: ObjectId("620145544cb04b772fd7a92d"),
yearMonth: 201907,
amount: 35,
year: '2019' }

Note: You can find the complete documentation for the $substr function .

Additional Resources

The following tutorials explain how to perform other common operations in MongoDB: