

How can “Not Equal” be used in queries in MongoDB?

Authored by
stats writer

June 30, 2024

RECOMMENDED CITATION

stats writer (2024). *How can “Not Equal” be used in queries in MongoDB?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=162090>

The "Not Equal" operator is a key component in MongoDB queries, allowing users to filter and retrieve data based on specific criteria. This operator is used to check for values that are not equal to a specified value, resulting in a more precise and targeted query. By using the "Not Equal" operator, users can easily narrow down their search results and obtain the desired data from a MongoDB database. This feature makes querying in MongoDB more efficient and effective, providing users with the ability to customize and refine their data retrieval process.

MongoDB: Use "Not Equal" in Queries

You can use the `$ne` operator (which stands for "not equal") in MongoDB to query for documents where a field is not equal to a certain value.

This operator uses the following basic syntax:

```
db.myCollection.find({'team': {$ne : "Mavs"}})
```

This particular example finds all documents in the collection titled `myCollection` where the `team` field is not equal to "Mavs."

You can also use the `$nin` operator (which stands for "not in") to query for documents where a field is not equal to any value in a list.

This operator uses the following basic syntax:

```
db.myCollection.find({'team': {$nin : }})
```

This particular example finds all documents in the collection titled `myCollection` where the `team` field is not equal to "Mavs", "Cavs", or "Spurs."

The following examples show how to use each method in practice with a collection `teams` with the following documents:

```
db.teams.insertOne({team: "Mavs", points: 30, rebounds: 8})
```

```
db.teams.insertOne({team: "Spurs", points: 35, rebounds: 12})
```

```
db.teams.insertOne({team: "Rockets", points: 20, rebounds: 7})
```

```
db.teams.insertOne({team: "Warriors", points: 25, rebounds: 5})
```

```
db.teams.insertOne({team: "Cavs", points: 23, rebounds: 9})
```

Example 1: "Not Equal" Query

The following code shows how to find all documents in the `teams` collection where the `team` field is not equal to "Mavs":

```
db.teams.find({'team': {$ne : "Mavs"}})
```

This query returns the following documents:

```
{ _id: ObjectId("6203ec0e1e95a9885e1e7658"),  
team: 'Cavs',  
points: 23,  
rebounds: 9 }  
{ _id: ObjectId("6203ec0e1e95a9885e1e7656"),  
team: 'Rockets',  
points: 20,  
rebounds: 7 }  
{ _id: ObjectId("6203ec0e1e95a9885e1e7655"),  
team: 'Spurs',  
points: 35,  
rebounds: 12 }  
{ _id: ObjectId("6203ec0e1e95a9885e1e7657"),  
team: 'Warriors',  
points: 25,  
rebounds: 5 }
```

Notice that every document in the teams collection is returned where the team field is not equal to "Mavs."

Note: The \$ne operator is case-sensitive.

Example 2: "Not In" Query

The following code shows how to find all documents in the teams collection where the team field is not equal to "Mavs", "Cavs", or "Spurs":

```
db.teams.find({'team': {$nin : }})
```

This query returns the following documents:

```
{ _id: ObjectId("6203ec0e1e95a9885e1e7656"),  
team: 'Rockets',  
points: 20,  
rebounds: 7 }  
{ _id: ObjectId("6203ec0e1e95a9885e1e7657"),  
team: 'Warriors',  
points: 25,  
rebounds: 5 }
```

Notice that every document in the teams collection is returned where the team field is not equal to "Mavs", "Cavs", or "Spurs."

Note #1: You can find the complete documentation for

the \$ne function .

Note #2: You can find the complete documentation for the \$nin function .

Additional Resources

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

[MongoDB: How to Query for "not null" in Specific Field](#)