

How do I query for “not null” in a specific field in MongoDB?

Authored by
stats writer

July 2, 2024

RECOMMENDED CITATION

stats writer (2024). *How do I query for “not null” in a specific field in MongoDB?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165623>

To query for "not null" in a specific field in MongoDB, the `$exists` operator can be used. This operator allows for the search of documents that contain a given field, regardless of its value. By using this operator with a value of `true`, the query will return all documents that have the specified field and its value is not null. This can be useful in filtering out documents that do not have a particular field or have a null value in that field. The syntax for this query would be:

```
db.collection.find({field: {$exists: true}})
```

This will return all documents from the collection that have a non-null value in the specified field. The `$exists` operator can also be used in combination with other operators to further refine the query. It is a powerful tool for retrieving specific data from a MongoDB database.

MongoDB: Query for "not null" in Specific Field

You can use the following syntax to query for all documents where a specific field is not null in MongoDB:

```
db.collection.find({"field_name":{$ne:null}})
```

The following examples show how to use this syntax in practice.

Example 1: Query for "not null" in Specific Field

Suppose we have a collection `teams` with the following documents:

```
db.teams.insertOne({team: "Mavs", position: null, points: 31})db.teams.insertOne({team: "Spurs",
```

```
position:          "Guard",           points:
22})db.teams.insertOne({team: "Rockets", position:
null, points: 19})db.teams.insertOne({team: "Warriors",
position:          "Forward",           points:
26})db.teams.insertOne({team: "Cavs", position:
"Guard", points: 33})
```

We can use the following code to find all documents where the "position" field is not null:

```
db.teams.find({"position":{$ne:null}})
```

This query returns the following documents:

```
{_id: ObjectId("618bf18f35d8a762d3c28717"),
team: 'Spurs',
position: 'Guard',
points: 22 }
```

```
{_id: ObjectId("618bf18f35d8a762d3c28719"),
team: 'Warriors',
position: 'Forward',
points: 26 }
```

```
{_id: ObjectId("618bf18f35d8a762d3c2871a"),
```

```
team: 'Cavs',  
position: 'Guard',  
points: 33 }
```

Notice that the only documents returned are the ones where the "position" field is not null.

Example 2: Query for "not null" (When Not Every Document Contains the Field)

Suppose we have a collection teams with the following documents:

```
db.teams.insertOne({team: "Mavs", position: null,  
points: 31})db.teams.insertOne({team: "Spurs", points:  
22})db.teams.insertOne({team: "Rockets", position:  
null, points: 19})db.teams.insertOne({team: "Warriors",  
position: "Forward", points:  
26})db.teams.insertOne({team: "Cavs", position:  
"Guard", points: 33})
```

Note that the second document in the collection doesn't even have a "position" field.

We can use the following code to find all documents where the "position" field is not null:

```
db.teams.find({"position":{$ne:null}})
```

```
{_id: ObjectId("618bf18f35d8a762d3c28719"),  
team: 'Warriors',  
position: 'Forward',  
points: 26 }
```

```
{_id: ObjectId("618bf18f35d8a762d3c2871a"),  
team: 'Cavs',  
position: 'Guard',  
points: 33 }
```

Since the second document doesn't even have a "position" field, it is not returned.

Also note that the other two documents that have a null value in the "position" field are not returned either.

Summary: By using the `$ne:null` syntax, we only return the documents where a specific field exists *and* is not null.

Additional Resources

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