How can greater than and less than operators be used in MongoDB queries?

Authored by

stats writer

June 30, 2024

The greater than and less than operators in MongoDB are used to perform comparison operations in queries. These operators allow the user to specify a range of values to be included in the query results. The greater than operator (>) is used to find documents with values greater than a specified value, while the less than operator (

MongoDB: Use Greater Than & Less Than in Queries

You can use the following operators in MongoDB to perform greater than or less than queries:

$lt: Less than$lte: Less than or equal$gt: Greater than$gte: Greater than or equal

The following methods show common ways to use these operators:

Method 1: Greater Than Query

db.myCollection.find({field1: {$gt:25}})

Method 2: Less Than Query

db.myCollection.find({field1: {$lt:25}})

Method 3: Greater Than *and* Less Than Query

db.myCollection.find({field1: {$gt:25, $lt:32}})

Method 4: Greater Than or Less Than Query

db.myCollection.find({ "$or": })

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: 31})
db.teams.insertOne({team: "Spurs", points: 22})
db.teams.insertOne({team: "Rockets", points: 19})
db.teams.insertOne({team: "Warriors", points: 26})
db.teams.insertOne({team: "Cavs", points: 33})

Example 1: Greater Than Query

The following code shows how to query for all documents where the value in the "points" field is greater than 25:

db.teams.find({points: {$gt:25}})

This query returns the following documents:

{ _id: ObjectId("6203e4a91e95a9885e1e764f"),

team: 'Mavs',
points: 31 }
{ _id: ObjectId("6203e4a91e95a9885e1e7652"),
team: 'Warriors',
points: 26 }
{ _id: ObjectId("6203e4a91e95a9885e1e7653"),
team: 'Cavs',
points: 33 }

Notice that each of the three documents in the output have a value in the "points" field greater than 25.

Example 2: Less Than Query

The following code shows how to query for all documents where the value in the "points" field is less than 25:

```
db.teams.find({points: {$lt:25}})
```

This query returns the following documents:

{ _id: ObjectId("6203e4a91e95a9885e1e7650"),
team: 'Spurs',
points: 22 }

{ _id: ObjectId("6203e4a91e95a9885e1e7651"),
team: 'Rockets',
points: 19 }

Notice that both of the documents in the output have a value in the "points" field less than 25.

Example 3: Greater Than *and* Less Than

The following code shows how to query for all documents where the value in the "points" field is greater than 25 *and* less than 32:

```
db.teams.find({points: {$gt:25, $lt:32}})
```

This query returns the following documents:

{ _id: ObjectId("6203e4a91e95a9885e1e764f"),
team: 'Mavs',
points: 31 }
{ _id: ObjectId("6203e4a91e95a9885e1e7652"),
team: 'Warriors',
points: 26 }

Notice that both of the documents in the output have a

value in the "points" field greater than 25 *and* less than 32.

Example 4: Greater Than or Less Than

The following code shows how to query for all documents where the value in the "points" field is greater than 30 *or* less than 20:

```
db.teams.find({ "$or": })
```

This query returns the following documents:

```
{ _id: ObjectId("6203e4a91e95a9885e1e764f"),
team: 'Mavs',
points: 31 }
{ _id: ObjectId("6203e4a91e95a9885e1e7651"),
team: 'Rockets',
points: 19 }
{ _id: ObjectId("6203e4a91e95a9885e1e7653"),
team: 'Cavs',
points: 33 }
```

Notice that each of the documents in the output have a value in the "points" field greater than 30 *or* less than

20.

Additional Resources

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