

MONGODB - QUERY DOCUMENT

http://www.tutorialspoint.com/mongodb/mongodb_query_document.htm

Copyright © tutorialspoint.com

The find Method

To query data from MongoDB collection, you need to use MongoDB's **find** method.

Syntax

Basic syntax of **find** method is as follows

```
>db.COLLECTION_NAME.find()
```

find method will display all the documents in a non structured way.

The pretty Method

To display the results in a formatted way, you can use **pretty** method.

Syntax:

```
>db.mycol.find().pretty()
```

Example

```
>db.mycol.find().pretty()
{
  "_id": ObjectId(7df78ad8902c),
  "title": "MongoDB Overview",
  "description": "MongoDB is no sql database",
  "by": "tutorials point",
  "url": "http://www.tutorialspoint.com",
  "tags": ["mongodb", "database", "NoSQL"],
  "likes": "100"
}
```

Apart from find method there is **findOne** method, that reruns only one document.

RDBMS Where Clause Equivalents in MongoDB

To query the document on the basis of some condition, you can use following operations

Operation	Syntax	Example	RDBMS Equivalent
Equality	{<key>:<value>}	db.mycol.find " by ":" tutorials point " .pretty	where by = 'tutorials point'
Less Than	{<key>:{\$lt:<value>}}	db.mycol.find " likes ": \$lt:50.pretty	where likes < 50
Less Than Equals	{<key>:{\$lte:<value>}}	db.mycol.find " likes ": \$lte:50 .pretty	where likes <= 50
Greater Than	{<key>:{\$gt:<value>}}	db.mycol.find " likes ": \$gt:50 .pretty	where likes > 50

Greater Than Equals	{<key>:{\$gte:<value>}}	db.mycol.find " likes ": \$gte:50 .pretty	where likes >= 50
Not Equals	{<key>:{\$ne:<value>}}	db.mycol.find " likes ": \$ne:50 .pretty	where likes != 50

AND in MongoDB

Syntax:

In the **find** method if you pass multiple keys by separating them by ',' then MongoDB treats it **AND** condition. Basic syntax of **AND** is shown below:

```
>db.mycol.find({key1:value1, key2:value2}).pretty()
```

Example

Below given example will show all the tutorials written by 'tutorials point' and whose title is 'MongoDB Overview'

```
>db.mycol.find({"by":"tutorials point","title": "MongoDB Overview"}).pretty()
{
  "_id": ObjectId(7df78ad8902c),
  "title": "MongoDB Overview",
  "description": "MongoDB is no sql database",
  "by": "tutorials point",
  "url": "http://www.tutorialspoint.com",
  "tags": ["mongodb", "database", "NoSQL"],
  "likes": "100"
}
```

For the above given example equivalent where clause will be '**where by='tutorials point' AND title='MongoDB Overview'**'. You can pass any number of key, value pairs in find clause.

OR in MongoDB

Syntax:

To query documents based on the OR condition, you need to use **\$or** keyword. Basic syntax of **OR** is shown below:

```
>db.mycol.find(
{
  $or: [
    {key1: value1}, {key2:value2}
  ]
})
).pretty()
```

Example

Below given example will show all the tutorials written by 'tutorials point' or whose title is 'MongoDB Overview'

```
>db.mycol.find({$or:[{"by":"tutorials point"}, {"title": "MongoDB Overview"}]}).pretty()
{
  "_id": ObjectId(7df78ad8902c),
  "title": "MongoDB Overview",
  "description": "MongoDB is no sql database",
```

```
"by": "tutorials point",
"url": "http://www.tutorialspoint.com",
"tags": ["mongodb", "database", "NoSQL"],
"likes": "100"
}
>
```

Using AND and OR together

Example

Below given example will show the documents that have likes greater than 100 and whose title is either 'MongoDB Overview' or by is 'tutorials point'. Equivalent sql where clause is '**where likes>10 AND by = 'tutorials point' OR title = 'MongoDB Overview'**'

```
>db.mycol.find({"likes": {$gt:10}, $or: [{"by": "tutorials point"}, {"title": "MongoDB Overview"}]}).pretty()
{
  "_id": ObjectId(7df78ad8902c),
  "title": "MongoDB Overview",
  "description": "MongoDB is no sql database",
  "by": "tutorials point",
  "url": "http://www.tutorialspoint.com",
  "tags": ["mongodb", "database", "NoSQL"],
  "likes": "100"
}
>
```

Processing math: 100%