# MongoDB - Query Document

In this chapter, we will learn how to query document from MongoDB collection.

## The find() Method

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

### Syntax

The basic syntax of **find()** method is as follows −

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

**find()** method will display all the documents in a non-structured way.

### Example

Assume we have created a collection named mycol as −

```
> use sampleDB
switched to db sampleDB
> db.createCollection("mycol")
{ "ok" : 1 }
>
```

And inserted 3 documents in it using the insert() method as shown below −

```
> db.mycol.insert([
    {
        title: "MongoDB Overview",
        description: "MongoDB is no SQL database",
        by: "tutorials point",
        url: "http://www.tutorialspoint.com",
```

```
        tags: ["mongodb", "database", "NoSQL"],
        likes: 100
    },
    {
        title: "NoSQL Database",
        description: "NoSQL database doesn't have tables",
        by: "tutorials point",
        url: "http://www.tutorialspoint.com",
        tags: ["mongodb", "database", "NoSQL"],
        likes: 20,
        comments: [
            {
                user:"user1",
                message: "My first comment",
                dateCreated: new Date(2013,11,10,2,35),
                like: 0
            }
        ]
    }
])
```

Following method retrieves all the documents in the collection –

```
> db.mycol.find()
{ "_id" : ObjectId("5dd4e2cc0821d3b44607534c"), "title" : "MongoDB
Overview", "description" : "MongoDB is no SQL database", "by" :
"tutorials point", "url" : "http://www.tutorialspoint.com", "tags" : [
"mongodb", "database", "NoSQL" ], "likes" : 100 }
{ "_id" : ObjectId("5dd4e2cc0821d3b44607534d"), "title" : "NoSQL
Database", "description" : "NoSQL database doesn't have tables", "by" :
"tutorials point", "url" : "http://www.tutorialspoint.com", "tags" : [
"mongodb", "database", "NoSQL" ], "likes" : 20, "comments" : [ { "user"
: "user1", "message" : "My first comment", "dateCreated" :
ISODate("2013-12-09T21:05:00Z"), "like" : 0 } ] }
>
```

## The pretty() Method

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

## Syntax

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

## Example

Following example retrieves all the documents from the collection named mycol and arranges them in an easy-to-read format.

```
> db.mycol.find().pretty()
{
    "_id" : ObjectId("5dd4e2cc0821d3b44607534c"),
    "title" : "MongoDB Overview",
    "description" : "MongoDB is no SQL database",
    "by" : "tutorials point",
    "url" : "http://www.tutorialspoint.com",
    "tags" : [
        "mongodb",
        "database",
        "NoSQL"
    ],
    "likes" : 100
}
{
    "_id" : ObjectId("5dd4e2cc0821d3b44607534d"),
    "title" : "NoSQL Database",
    "description" : "NoSQL database doesn't have tables",
    "by" : "tutorials point",
    "url" : "http://www.tutorialspoint.com",
    "tags" : [
        "mongodb",
        "database",
        "NoSQL"
    ],
    "likes" : 20,
    "comments" : [
        {
            "user" : "user1",
            "message" : "My first comment",
            "dateCreated" : ISODate("2013-12-09T21:05:00Z"),
            "like" : 0
        }
```

```
    ]
}
```

## The findOne() method

Apart from the find() method, there is **findOne()** method, that returns only one document.

### Syntax

```
>db.COLLECTIONNAME.findOne()
```

### Example

Following example retrieves the document with title MongoDB Overview.

```
> db.mycol.findOne({title: "MongoDB Overview"})
{
    "_id" : ObjectId("5dd6542170fb13eec3963bf0"),
    "title" : "MongoDB Overview",
    "description" : "MongoDB is no SQL database",
    "by" : "tutorials point",
    "url" : "http://www.tutorialspoint.com",
    "tags" : [
        "mongodb",
        "database",
        "NoSQL"
    ],
    "likes" : 100
}
```

## 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>:{$eg;<value>}} | db.mycol.find({"by":"tutorials point"}).pretty() | where by = 'tutorials point' |
| Less Than | {<key>:{$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":{$ne:50}}).pretty() | where likes != 50 |
| Values in an array | {<key>:{$in:[<value1>, <value2>, <valueN>]}} | db.mycol.find({"name":{$in:["Raj", "Ram", "Raghu"]}}).pretty() | Where name matches any of the value in : ["Raj", "Ram", "Raghu"] |
| Values not in an array | {<key>:{$nin:<value>}} | db.mycol.find({"name":{$nin:["Ramu", "Raghav"]}}).pretty() | Where name values is not in the array : ["Ramu", "Raghav"] or, doesnt exist at all |

# AND in MongoDB

## Syntax

To query documents based on the AND condition, you need to use $and keyword. Following is the basic syntax of AND −

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

## Example

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

```
> db.mycol.find({$and:[{"by":"tutorials point"},{"title": "MongoDB
Overview"}]}).pretty()
{
    "_id" : ObjectId("5dd4e2cc0821d3b44607534c"),
    "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. Following is the basic syntax of **OR** −

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

### Example

Following 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

The following example will show the documents that have likes greater than 10 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"
}
>
```

# NOR in MongoDB

## Syntax

To query documents based on the NOT condition, you need to use $not keyword. Following is the basic syntax of **NOT**



```
        $not: [
            {key1: value1}, {key2:value2}
        ]
    }
)
```

## Example

Assume we have inserted 3 documents in the collection **empDetails** as shown below −

```
db.empDetails.insertMany(
    [
        {
            First_Name: "Radhika",
            Last_Name: "Sharma",
            Age: "26",
            e_mail: "radhika_sharma.123@gmail.com",
            phone: "9000012345"
        },
        {
            First_Name: "Rachel",
            Last_Name: "Christopher",
            Age: "27",
            e_mail: "Rachel_Christopher.123@gmail.com",
            phone: "9000054321"
        },
        {
            First_Name: "Fathima",
            Last_Name: "Sheik",
            Age: "24",
            e_mail: "Fathima_Sheik.123@gmail.com",
            phone: "9000054321"
```

```
        }
    ]
)
```

Following example will retrieve the document(s) whose first name is not "Radhika" and last name is not "Christopher"

```
> db.empDetails.find(
    {
        $nor:[
            40
            {"First_Name": "Radhika"},
            {"Last_Name": "Christopher"}
        ]
    }
).pretty()
{
    "_id" : ObjectId("5dd631f270fb13eec3963bef"),
    "First_Name" : "Fathima",
    "Last_Name" : "Sheik",
    "Age" : "24",
    "e_mail" : "Fathima_Sheik.123@gmail.com",
    "phone" : "9000054321"
}
```

## NOT in MongoDB

### Syntax

To query documents based on the NOT condition, you need to use $not keyword following is the basic syntax of **NOT**

```
>db.COLLECTION_NAME.find(
    {
        $NOT: [
            {key1: value1}, {key2:value2}
        ]
    }
).pretty()
```

## Example

Following example will retrieve the document(s) whose age is not greater than 25

```
> db.empDetails.find( { "Age": { $not: { $gt: "25" } } } )
{
    "_id" : ObjectId("5dd6636870fb13eec3963bf7"),
    "First_Name" : "Fathima",
    "Last_Name" : "Sheik",
    "Age" : "24",
    "e_mail" : "Fathima_Sheik.123@gmail.com",
    "phone" : "9000054321"
}
```

DSA Tutorial

Spring Boot Tutorial

SDLC Tutorial

Unix Tutorial

## CERTIFICATIONS

Business Analytics Certification

Java & Spring Boot Advanced Certification

Data Science Advanced Certification

Cloud Computing And DevOps

Advanced Certification In Business Analytics

Artificial Intelligence And Machine Learning

DevOps Certification

Game Development Certification

Front-End Developer Certification

AWS Certification Training

Python Programming Certification

## COMPILERS & EDITORS

Online Java Compiler

Online Python Compiler

Online Golang Compiler

Online C Compiler

Online C++ Compiler

Online C# Compiler

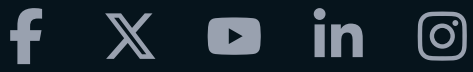
Online PHP Compiler

Online MATLAB Compiler

Online Bash Compiler

Online SQL Compiler

Online Html Editor



Tutorials Point is a leading Ed Tech company striving to provide the best learning material on technical and non-technical subjects.