**Logical Operator - $and**

The MongoDB $and operator performs a logical AND operation on an array of two or more expressions and retrieves the documents which satisfy all the expressions in the array. The $and operator uses short-circuit evaluation. If the first expression (e.g. <expression1>) evaluates to false, MongoDB will not evaluate the remaining expressions.

**Syntax:**

{ $and: [ { <exp1> }, { <exp2> } , ... , { <expN> } ] }

**Our database name is 'myinfo' and our collection name is 'student'. Here, is the collection bellow.**

**Sample collection "student"**

[

{

"f\_name" : "Zenny",

"sex" : "Female",

"class" : "VI",

"age" : 12,

"grd\_point" : 32.6342

},

{

"f\_name" : "Paul",

"sex" : "Male",

"class" : "VII",

"age" : 13,

"grd\_point" : 29.5904

},

{

"f\_name" : "Tom",

"sex" : "Male",

"class" : "VI",

"age" : 11,

"grd\_point" : 30.1257

},

{

"f\_name" : "Lassy",

"sex" : "Female",

"class" : "VIII",

"age" : 13,

"grd\_point" : 28.2514

},

{

"f\_name" : "Peter",

"sex" : "Male",

"class" : "VI",

"age" : 11,

"grd\_point" : 31.5201

}

]

Copy

**Example of MongoDB Logical Operator - $and**

If we want to select all documents from the collection "student" which satisfying the condition -

1. *sex* of student is Female and

2. *class* of the student is VI and

3. *grd\_point* of the student is greater than equal to 31

the following mongodb command can be used :

>db.student.find({$and:[{"sex":"Male"},{"grd\_point":{ $gte: 31 }},{"class":"VI"}]}).pretty();

**N.B.** find() method displays the documents in a non structured format but to display the results in a formatted way, the pretty() method can be used.

## SQL equivalent command is

SELECT \*

FROM student

WHERE sex="Male" AND grd\_point>=31 AND class="VI";

Copy

Output:

{

"\_id" : ObjectId("527b3cc65ceafed9b2254a98"),

"f\_name" : "Peter",

"sex" : "Male",

"class" : "VI",

"age" : 11,

"grd\_point" : 31.5201

}

The above AND operation can be written inplicitly in such a manner, by specifying a comma separated list of expressions. Here is the example below:

db.student.find({"sex":"Male","grd\_point":{ $gte: 31},"class":"VI"}).pretty();

## Logical Operator - $not

The MongoDB $not operator performs a logical NOT operation on the given expression and fetches selected documents that do not match the expression and the document that do not contain the field as well, specified in the expression.

**Syntax:**

{ field: { $not: { <expression> } } }

## Example of MongoDB Logical Operator - $not

If we want to select all documents from the collection "student" which satisfying the condition -

*age* of the student is at least 12

the following mongodb command can be used :

>db.student.find( {"age": { $not: {$lt : 12}}}).pretty();

Copy

## SQL equivalent command is

SELECT \*

FROM student

WHERE age>=12;

Copy

Output:

{

"\_id" : ObjectId("527b3cc65ceafed9b2254a94"),

"f\_name" : "Zenny",

"sex" : "Female",

"class" : "VI",

"age" : 12,

"grd\_point" : 32.6342

}

{

"\_id" : ObjectId("527b3cc65ceafed9b2254a95"),

"f\_name" : "Paul",

"sex" : "Male",

"class" : "VII",

"age" : 13,

"grd\_point" : 29.5904

}

{

"\_id" : ObjectId("527b3cc65ceafed9b2254a97"),

"f\_name" : "Lassy",

"sex" : "Female",

"class" : "VIII",

"age" : 13,

"grd\_point" : 28.2514

}

## Example of MongoDB Logical Operator - $not with pattern matching

If we want to select all documents from the collection "student" which satisfying the condition -

*sex* of student must not Male.

the following mongodb command can be used :

>db.student.find( {"sex": { $not: /^M.\*/}}).pretty();

Copy

## SQL equivalent command is

SELECT \*

FROM student

WHERE sex NOT LIKE 'M%';

Copy

Output:

{

"\_id" : ObjectId("527b3cc65ceafed9b2254a94"),

"f\_name" : "Zenny",

"sex" : "Female",

"class" : "VI",

"age" : 12,

"grd\_point" : 32.6342

}

{

"\_id" : ObjectId("527b3cc65ceafed9b2254a97"),

"f\_name" : "Lassy",

"sex" : "Female",

"class" : "VIII",

"age" : 13,

"grd\_point" : 28.2514

}

**The logical operator $or and $nor.**

The $or operator is used to search multiple expression in a single query with only one matching criterion to be needed in a document. More than one keys and values can be used with the $or operator.

The $nor operator is used to search multiple expression in a single query which does not match any of the values specified with the $nor.

**Our database name is 'myinfo' and our collection name is 'testtable'. Here, is the collection bellow.**

## Sample collection "testtable"

{

"\_id" : ObjectId("528f4e630fe5e6467e58ae7b"),

"user\_id" : "user1",

"password" : "1a2b3c",

"sex" : "Male",

"age" : 17,

"date\_of\_join" : "16/10/2010",

"education" : "M.C.A.",

"profession" : "CONSULTANT",

"interest" : "MUSIC",

"extra" : {

"community\_name" : [

"MODERN MUSIC",

"CLASSICAL MUSIC",

"WESTERN MUSIC"

],

"community\_moder\_id" : [

"MR. Alex",

"MR. Dang",

"MR Haris"

],

"community\_members" : [

700,

200,

1500

],

"friends" : {

"valued\_friends\_id" : [

"kumar",

"harry",

"anand"

],

"ban\_friends\_id" : [

"Amir",

"Raja",

"mont"

]

}

}

}

{

"\_id" : ObjectId("528f4e720fe5e6467e58ae7c"),

"user\_id" : "user2",

"password" : "11aa1a",

"sex" : "Male",

"age" : 24,

"date\_of\_join" : "17/10/2009",

"education" : "M.B.A.",

"profession" : "MARKETING",

"interest" : " MUSIC",

"extra" : {

"community\_name" : [

"MODERN MUSIC",

"CLASSICAL MUSIC",

"WESTERN MUSIC"

],

"co mmunity\_moder\_id" : [

"MR. Roy",

"MR. Das",

"MR Doglus"

],

"community\_members" : [

500,

300,

1400

],

"friends" : {

"valued\_friends\_id" : [

"pal",

"viki",

"john"

],

"ban\_friends\_id" : [

"jalan",

"mono j",

"evan"

]

}

}

}

{

"\_id" : ObjectId("528f4e7e0fe5e6467e58ae7d"),

"user\_id" : "user3",

"password" : "b1c1d1",

"sex" : "Female",

"age" : 19,

"date\_of\_join" : "16/10/2010",

"education" : "M.C.A.",

"profession" : "IT COR.",

"interest" : "AR T",

"extra" : {

"community\_name" : [

"MODERN ART",

"CLASSICAL ART",

"WESTERN ART"

],

"community\_mo der\_id" : [

"MR. Rifel",

"MR. Sarma",

"MR Bhatia"

],

"community\_members" : [

5000,

2000,

1500

],

"friends" : {

"valued\_friends\_id" : [

"philip",

"anant",

"alan"

],

"ban\_friends\_id" : [

"Amir",

"Raja",

"mont"

]

}

}

}

{

"\_id" : ObjectId("528f4e910fe5e6467e58ae7e"),

"user\_id" : "user4",

"password" : "abczyx",

"sex" : "Female",

"age" : 22,

"date\_of\_join" : "17/8/2009",

"education" : "M.B.B.S.",

"profession" : "DOCTOR",

"interest" : "SPORTS",

"extra" : {

"community\_name" : [

"ATHELATIC",

"GAMES FAN GYES",

"FAVOURIT GAMES"

],

"community\_moder\_id" : [

"MR. Paul",

"MR. Das",

"MR Doglus"

],

"community\_members" : [

2500,

2200,

3500

],

"friends" : {

"valued\_friends\_id" : [

"vinod",

"viki",

"john"

],

"ban\_friends\_id" : [

"jalan",

"monoj",

"evan"

]

}

}

}

[Document written in command prompt](https://www.w3resource.com/mongodb/mongodb-testtable-dot-notation-sample.gif)

## MongoDB conditional operator - $or example

If we want to fetch documents from the collection "testtable" which containing the value of "age " either 19 or 22 or 23, the following mongodb command can be used :

>db.testtable.find({$or : [{"age" : 19},{"age" : 22},{"age":23}]})

N.B. find() method displays the documents in a non structured format but to display the results in a formatted way, the pretty() method can be used.

## Sql equivalent command is

SELECT \*

FROM testtable

WHERE age=19 OR age=22 OR age=23;

Output of the command

{

"\_id" : ObjectId("528f35450fe5e6467e58ae79"),

"user\_id" : "user3",

"password" : "b1c1d1",

"sex" : "Female",

"age" : 19,

"date\_of\_join" : "16/10/2010",

"education" : "M.C.A.",

"profession" : "IT COR.",

"interest" : "AR T",

"extra" : {

"community\_name" : [

"MODERN ART",

"CLASSICAL ART",

"WESTERN ART"

],

"community\_mo der\_id" : [

"MR. Rifel",

"MR. Sarma",

"MR Bhatia"

],

"community\_members" : [

5000,

2000,

1500

],

"friends" : {

"valued\_friends\_id" : [

"philip",

"anant",

"alan"

],

"ban\_friends\_id" : [

"Amir",

"Raja",

"mont"

]

}

}

}

{

"\_id" : ObjectId("528f35860fe5e6467e58ae7a"),

"user\_id" : "user4",

"password" : "abczyx",

"sex" : "Female",

"age" : 22,

"date\_of\_join" : "17/8/2009",

"education" : "M.B.B.S.",

"profession" : "DOCTOR",

"interest" : "SPORTS",

"extra" : {

"community\_name" : [

"ATHELATIC",

"GAMES FAN GYES",

"FAVOURIT GAMES"

],

"community\_moder\_id" : [

"MR. Paul",

"MR. Das",

"MR Doglus"

],

"community\_members" : [

2500,

2200,

3500

],

"friends" : {

"valued\_friends\_id" : [

"vinod",

"viki",

"john"

],

"ban\_friends\_id" : [

"jalan",

"monoj",

"evan"

]

}

}

}

[Document written in command prompt](https://www.w3resource.com/mongodb/mongodb-or-operators-example1.gif)

## MongoDB $or operator with another field

If we want to fetch documents from the collection "testtable" which containing the value of "date\_of\_join" is "16/10/2010" and the value of "age " either 19 or 22 or 23, the following mongodb command can be used :

>db.testtable.find( { "date\_of\_join" : "16/10/2010" , $or : [{"age" : 19},{"age" : 22},{"age":23}] } )

N.B. find() method displays the documents in a non structured format but to display the results in a formatted way, the pretty() method can be used.

Output of the command:

{

"\_id" : ObjectId("528f35450fe5e6467e58ae79"),

"user\_id" : "user3",

"password" : "b1c1d1",

"sex" : "Female",

"age" : 19,

"date\_of\_join" : "16/10/2010",

"education" : "M.C.A.",

"profession" : "IT COR.",

"interest" : "AR T",

"extra" : {

"community\_name" : [

"MODERN ART",

"CLASSICAL ART",

"WESTERN ART"

],

"community\_mo der\_id" : [

"MR. Rifel",

"MR. Sarma",

"MR Bhatia"

],

"community\_members" : [

5000,

2000,

1500

],

"friends" : {

"valued\_friends\_id" : [

"philip",

"anant",

"alan"

],

"ban\_friends\_id" : [

"Amir",

"Raja",

"mont"

]

}

}

}

[Document written in command prompt](https://www.w3resource.com/mongodb/mongodb-or-operators-example2.gif)

## MongoDB $nor(not or ) operator

If we want to fetch documents from the collection "testtable" which containing the value of "date\_of\_join" is "16/10/2010" and not containing the value of "age " either 19 or 22 or 23, the following mongodb command can be used :

>db.testtable.find( { "date\_of\_join" : "16/10/2010" , $nor : [{"age" : 19},{"age" : 22},{"age":23}] } )

N.B. find() method displays the documents in a non structured format but to display the results in a formatted way, the pretty() method can be used.

Output of the command:

{

"\_id" : ObjectId("528f34950fe5e6467e58ae77"),

"user\_id" : "user1",

"password" : "1a2b3c",

"sex" : "Male",

"age" : 17,

"date\_of\_join" : "16/10/2010",

"education" : "M.C.A.",

"profession" : "CONSULTANT",

"interest" : "MUSIC",

"extra" : {

"community\_name" : [

"MODERN MUSIC",

"CLASSICAL MUSIC",

"WESTERN MUSIC"

],

"community\_moder\_id" : [

"MR. Alex",

"MR. Dang",

"MR Haris"

],

"community\_members" : [

700,

200,

1500

],

"friends" : {

"valued\_friends\_id" : [

"kumar",

"harry",

"anand"

],

"ban\_friends\_id" : [

"Amir",

"Raja",

"mont"

]

}

}

}