

# QUERIES

# Operators

\$or

\$and

\$not

\$in

\$exists

\$regex

\$lt

\$gt

# Equality Query

- Match by field name and it's exact value

{<fieldName1> : <value1>, <fieldName2> : <value2>, ...}

- Examples

{"name" : "Kitty Snow"}

{"age" : 38}

{"gender" : "female" , "eyeColor" : "blue"}



AND Condition

# Operators

- Operators add conditions

```
{ <fieldName1>: { <operator1>: <value1> }, ... }
```

- Examples

```
{"favoriteFruit" : {"$ne": "apple"}}
```

```
{"age" : {"$gt": 25}}
```

```
{"eyeColor" : {"$in": ["green" , "blue"]}}
```

# Comparison Operators

\$eq

\$gt

\$lt

\$ne

\$gte

\$lte

## ● Examples

```
{"favoriteFruit" : {"$ne": "orange"}}
```

```
{"age" : {"$lt": 35}}
```

```
{"age" : {"$gte": 23}}
```

```
{"age" : {"$gt": 23, "$lt": 27}}
```

# Comparison Operators

**\$in**

**\$nin**

## ● Examples

```
{"eyeColor" : {"$in": ["green" , "blue"]}}
```

```
{"favoriteFruit" : {"$nin": ["banana" , "apple"]}}
```

# \$and

- Logically combines multiple conditions. Resulting documents must match ALL conditions

```
{ $and: [ { <condition1> }, { <condition2> } ... ] }
```

## Note

Explicit \$and MUST be used if conditions contain same field or operator

- Examples

```
{ $and: [ { "gender" : "male" }, { "age" : 25 } ] }
```

```
{ $and: [  
  {"age": {"$ne": 25}},  
  {"age": {"$gte": 20}} ] }
```

# Explicit \$and with same field

```
{ $and: [ {"age": {"$ne": 25}} , {"age": {"$gte": 20}} ] }
```

≠

```
{"age": {"$ne": 25} , "age": {"$gte": 20}}
```



# \$or


- Logically combines multiple conditions. Resulting documents must match **ANY** of the conditions

```
{ $or: [ { <condition1> }, { <condition2> } ... ] }
```

- Examples

```
{ $or: [ { "gender" : "male" }, { "age" : 25 } ] }
```

```
{ $or: [ { "age" : 20 }, { "age" : 27 } ] }
```



*Equal to:* { "age" : { "\$in": [20 , 27] } }

# Query Embedded Documents

```
"company" : {  
  "title" : "SHEPARD",  
  "email" : "carmellamorse@shepard.com",  
  "phone" : "+1 (829) 478-3744",  
  "location" : {  
    "country" : "USA",  
    "address" : "379 Tabor Court"  
  }  
}
```

## Note

**Fields accessed with dot notation (.) MUST be inside quotation marks**

## ● Examples

```
{ "company.title": "SHEPARD" }
```

```
{ "company.location.address": "379 Tabor Court" }
```

# Query Array by element Value

- **Array contains certain value**

`{<fieldName>: <value>}`

- **Specific element in array has certain value**

`{"<fieldName>.<index>": <value>}`

- **Examples**

`{ tags: "ad" }`

`{ "tags.0": "ad" }`

`{ "tags.3": "ab" }`

# Query Array using \$all , \$size

- **Array contains all specified values independent of order**

`{<fieldName>: {"$all": [<value1> , <value2>, ...]}}`

- **Array is of certain size**

`{<fieldName>: {"$size": <number>}}`

- **Examples**

`{ tags: {"$all": ["ad" , "eu"]}}`

`{ tags: {"$size": 4}}`

# Query Array of Nested Documents

```
{  
  "friends": [  
    {  
      "name": "Lora",  
      "age": NumberInt(23)  
    },  
    {  
      "name": "Bob",  
      "age": NumberInt(25)  
    }  
  ]  
}
```

## ● Examples

```
{ "friends.name": "Lora" }
```

```
{ friends: { "name": "Lora", "age": 23 } }
```

# \$elemMatch

- At least one nested document in the Array must match ALL conditions. Order of conditions doesn't matter

```
{ <arrayField>: { $elemMatch: { <condition1> , <condition2> , ...}} }
```

- Examples

```
{ friends: { $elemMatch: { name:  
"Bob", registered: false }}}}
```

```
{ friends: { $elemMatch: { registered:  
false, age: 25 }}}}
```

# Element Operators

**\$exists**      **\$type**

## ● Examples

```
{ tags: { $exists: true }}
```

```
{ name: { $exists: false }}
```

```
{ posts: { $type: "int" }}
```

```
{ name: { $type: 2 }}
```

```
{ quantity: { $type: ["int" , "double"] }}
```

```
{ index: { $type: "int", $eq: 5 }}
```

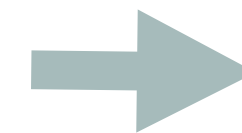
# BSON Type Identifiers

Type	Number ID	String ID
String	2	“string”
Object	3	“object”
Array	4	“array”
Boolean	8	“bool”
32-bit Integer	16	“int”
64-bit Integer	18	“long”
Double	1	“double”
Date	9	“date”
ObjectId	7	“objectId”



# Filter Fields

```
{
  "_id": ObjectId("5ad4cbde2edbf6ddeec71743"),
  "index": 2,
  "name": "Hays Wise",
  "isActive": false,
  "registered": ISODate("2015-02-23T10:22:15.000Z"),
  "age": 24,
  "gender": "male",
  "eyeColor": "green",
  "favoriteFruit": "strawberry",
  "company": {
    "title": "EXIAND",
    "email": "hayswise@exiand.com",
    "phone": "+1 (801) 583-3393",
    "location": {
      "country": "France",
      "address": "795 Borinquen Pl"
    }
  },
  "tags": ["amet", "ad", "elit", "ipsum"]
}
```



```
{
  "index": 2,
  "name": "Hays Wise",
  "isActive": false,
  "company": {
    "location": {
      "country": "France",
      "address": "795 Borinquen Pl"
    }
  }
}
```

# Filter Fields Examples

## ● Examples

```
db.persons.find(<query> , {name: 1, age: 1})
```

```
db.persons.find(<query> , {"company.location": 1, age: 1})
```

```
db.persons.find(<query> , {_id: 0, name: 1, age: 1})
```

```
db.persons.find(<query> , {name: 0, age: 0})
```

# \$regex

## ● Filter using Regular Expression

```
{<fieldName>: {"$regex": /pattern/<options> }}
```

```
{<fieldName>: {"$regex": /pattern/ , $options: "<options>"}}
```

## ● Examples

```
{ city: {"$regex": /ton/i}}
```

```
{ tags: {"$regex": /^ad$/ , $options: "i"}}
```

```
{ state: {"$regex": "ca"}}
```

# SUMMARY

- Query Operators

- \$or

- \$and

- \$regex

- \$lt, \$gt

- Array Query Operators

- \$all

- \$size

- \$elemMatch