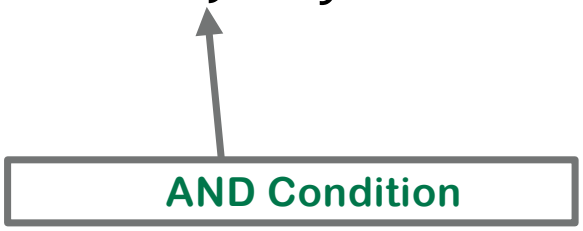
Equality Query

- **Match by field name and it's exact value**

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

- **Examples**

```
{"name" : "Kitty Snow"}
{"age" : 38}
{"gender" : "female" , "eyeColor" : "blue"}
```



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}} ] }
```

$or

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

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

```
db.getCollection('restaurants').count({$and:[{"borou
gh" : "Queens"}, {"cuisine" : "American "}]})
```

- 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 of Nested Documents

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

- Examples

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

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})
```

In MongoDB, projection means selecting only the necessary data rather than selecting whole of the data of a document. If a document has 5 fields and you need to show only 3, then select only 3 fields from them.

>db.mycol.find({},{"title":1,_id:0})