

# Chapter 3: Deeper Dive on the MongoDB Query Language

## comparison query operators

### extras

- <https://docs.mongodb.com/manual/reference/operator/query/>
- <https://docs.mongodb.com/manual/reference/operator/query-comparison/>
  - m001\_comparisonoperators.zip

### operators

- \$eq, \$ne : igual y no-igual
- \$gt, \$gte : mayor y mayor-igual
- \$lt, \$lte : menor y menor-igual
- \$in, \$nin : en array y no-en array

### ejemplos

películas con **runtime** superior a 90 (minutos):

```
db.movieDetails.find({runtime: {$gt: 90}}, {_id: 0, title: 1, runtime: 1})
```

películas con **runtime** superior a 90 (minutos) e inferior a 120:

```
db.movieDetails.find({runtime: {$gt: 90, $lt: 120}}, {_id: 0, title: 1, runtime: 1})
```

películas con **runtime** mayor o igual a 180(minutos) y índice tomato igual a 100

```
db.movieDetails.find({runtime: {$gte: 180}, "tomato.meter": 100}, {_id: 0, title: 1, runtime: 1})
```

películas que **rated** diferente de «UNRATED» (incluso las que no tienen nada definido)

```
db.movieDetails.find({rated: {$ne: "UNRATED"}}, {_id: 0, title: 1, rated: 1})
```

películas que **rated** igual a los valores del array

```
db.movieDetails.find({rated: {$in: ["G", "PG"]}}, {_id: 0, title: 1, rated: 1})
```

## Element Operators

### extras

- <https://docs.mongodb.com/manual/reference/operator/query/type/>

- m001\_element\_operators.zip

## operators

- `$exists` : verifica la existencia o no de un campo en un documento
- `$type` : podemos filtrar usando el tipo del campo (ver enlace anterior)
- `null` : pueden existir campos a NULL o no existir, ambos serán tratados igual

## ejemplos

películas que tienen el campo **mpaaRating** (se podría hacer lo contrario cambiando a **false**)

```
db.moviesDetails.find({mpaaRating: {$exists: true}})
```

películas que no tienen el campo **mpaaRating** o que lo tienen = null

```
db.movieDetails.find({mpaaRating: null})
```

películas que tienen el campo **viewerRating** como un **int32**

```
db.movies.find({viewerRating: {$type: "int"}}).pretty()
```

## logical operators

### extras

- m001\_logical\_operators.zip

## operators

- `$or`
- `$and`
- `$not`
- `$nor`

## ejemplos

películas por dos campos (cualquier de ellos)

```
db.movieDetails.find({$or: [{"tomato.meter": {$gt: 95}}, {"metacritic": {$gt: 88}}]},
  {_id: 0, title: 1, "tomato.meter": 1, "metacritic": 1})
```

películas que cumplan los dos criterios a la vez

```
db.movieDetails.find({$and: [{"tomato.meter": {$gt: 95}}, {"metacritic": {$gt: 88}}]},
```

```
{_id: 0, title: 1, "tomato.meter": 1, "metacritic": 1})
```

de hecho, la instrucción anterior es equivalente a esta otra (por defecto se usa un AND en las búsquedas):

```
db.movieDetails.find({"tomato.meter": {$gt: 95},  
  "metacritic": {$gt: 88}},  
  {_id: 0, title: 1, "tomato.meter": 1, "metacritic": 1})
```

el uso del **\$and** tiene sentido cuando el campo es el mismo y ha de cumplir más de un criterio

```
db.movieDetails.find({$and: [{"metacritic": {$ne: null}},  
  {"metacritic": {$exists: true}}]},  
  {_id: 0, title: 1, "metacritic": 1})
```

## array operators

### extras

- m001\_all\_operator.zip

### operators

### ejemplos

## regex operator

### extras

### operators

### ejemplos

From:  
<https://miguelangel.torresegea.es/wiki/> - miguel angel torres egea

Permanent link:  
<https://miguelangel.torresegea.es/wiki/info:cursos:mongodbuniversity:m001:cap3?rev=1544889965>

Last update: 15/12/2018 08:06

