

Proyecto 1ª evaluación

Explicación de las consultas:

```
db.peliculas.find( { $and: [ { date: { $regex: /^1922/ } }, { precioentrada: { $lte: 3.33 } } ] } )
```

El fin de esta consulta es encontrar las películas que cumplan que el año de estreno sea 1922 y que cumplan que el precio de la entrada sea menor o igual que 3,33€.

En primer lugar se utiliza \$and (y) para que el resultado cumpla todos los requerimientos que nosotros escribamos.

Seguidamente \$regex actúa en esta ocasión para seleccionar las películas que sean del año 1922 independientemente del mes y el día que sea. (Campo tipo fecha).

Para acabar se usa \$lte (igual o menor) para seleccionar las películas cuyo precio de entrada sea menor o igual que 3,33€.

```
db.peliculas.find( { pais: { $not: { $eq: "España" } } } )
```

```
db.peliculas.find( { pais: { $ne: "España" } } )
```

Esta consulta pretende hallar las películas que no sean españolas.

Con ambas consultas se llega al mismo resultado.

Usando \$not (no cumplir) + \$eq (igual) estamos seleccionando las películas que no sean españolas.

Con \$ne (no es igual) hacemos exactamente lo mismo pero con un solo método.

```
db.peliculas.find( { $and: [ { color:true }, { genero: "Superhéroes" }, { date: { $lt: "2005-01-01" } } ] } )
```

El objetivo de esta consulta es encontrar las películas que cumplan que sea a color, que trate de superhéroes y que se estrenase antes del 2005.

En esta consulta introduzco un campo tipo boolean, que es el color.

\$and (y) es utilizado para que el resultado cumpla todos los requerimientos que nosotros escribamos.

\$lt (menor) sirve, en este caso, para seleccionar las películas que se estrenan antes del año 2005.(Campo tipo fecha).

```
db.peliculas.find( { $and: [ { duracion: { $lt: 115 } }, { genero: "Animación" } ] } )
```

Esta consulta tiene como fin encontrar la película que cumpla que dure menos de 115 minutos y que sea de animación.

\$and (y) es utilizado para que el resultado cumpla todos los requerimientos que nosotros queramos determinar.

\$lt (menor) sirve, en este caso, para seleccionar las películas que duren menos de 115 minutos.

```
db.peliculas.find( { "personal.direccion": { $regex: /Spiel.erg$/ } } )
```

Con esta consulta conseguimos mostrar las películas que están dirigidas por alguien que se apellida algo parecido a Spielverg o Spielberg.

En este caso accedemos al campo dirección mediante ese punto entre personal y dirección entre paréntesis.

Posteriormente uso \$regex para concretar un poco más cómo se llama el director de la película que busco.

```
db.peliculas.find( { $and: [ { date: { $gte: "2000-01-01" } }, { precioentrada: 8 } ] } )
```

```
db.peliculas.find( { $and: [ { date: { $gte: "2000-01-01" } }, { precioentrada: 5 } ] } )
```

El objetivo de esta consulta es hallar las películas que cumplen que su precio de entrada es 8€ y que fue estrenada en el 2000 o posteriormente.

Por otra parte el objetivo de la segunda consulta es el mismo pero cambiando el precio de entrada de 8€ a 5€.

En esta consulta se usa \$and (y) anteriormente explicado varias veces y \$gte (mayor o igual) para que la fecha de estreno sea el 2000 o posterior. (Campo tipo fecha).

```
db.peliculas.find( { $and: [ { $or: [ { pegi: "TP" }, { pegi: { $lte: 7 } } ] }, { $or: [ { genero: "Aventura" }, { genero: "Animación" } ] } ] } )
```

Esta consulta pretende encontrar las películas que cumplen que son para todos los públicos o para niños de 7 años o menos y sea de aventuras o de animación.

En esta consulta aplico una estructura un poco más compleja ya que el \$and envuelve a toda la consulta, mientras que el primer \$or (o) se aplica sólo a los dos pegi y el segundo \$or (o) se aplica a los dos géneros. Así, el resultado de la búsqueda podría ser una película para todos los públicos o recomendada para los niños de 7 años o menos y tendrá que ser de aventuras o animación.

```
db.peliculas.find( { $and: [ { "personal.direccion": "Juan Antonio Bayona" }, { "personal.protagonista": "Belén Rueda" } ] } )
```

El fin de esta consulta es encontrar la o las películas que estén dirigidas por Juan Antonio Bayona y protagonizadas por Belén Rueda.

Para esto vuelvo a usar el \$and que envuelve a toda la consulta y vuelvo a concretar dentro de un subdocumento, en este caso concreto en el subdocumento de dirección y en el subdocumento de los protagonistas, que esté a su vez es un campo tipo array.

Explicación de los nuevos operadores empleados:

```
db.peliculas.updateMany( { titulo: "Tenet" }, { $set: { precioentrada: 5 } } )
```

Este operador sirve para actualizar uno o varios documentos para cambiar sus valores.

La primera llave es un filtro para seleccionar los documentos que se desean actualizar. En este caso el documento que quiero cambiar es el que tiene como título Tenet, por lo que solo se cambiará ese documento.

La segunda llave, con \$set, se indica el cambio que se va a ejecutar. En este caso se va a rebajar el precio de la entrada, de 8€ a 5€.

```
db.peliculas.deleteMany( { titulo: "Nosferatu, eine Symphonie des Grauens" } )
```

La función de este operador es borrar el o los documentos deseados. En este caso solo quiero borrar el documento que tiene como título "Nosferatu, eine Symphonie des Grauens".