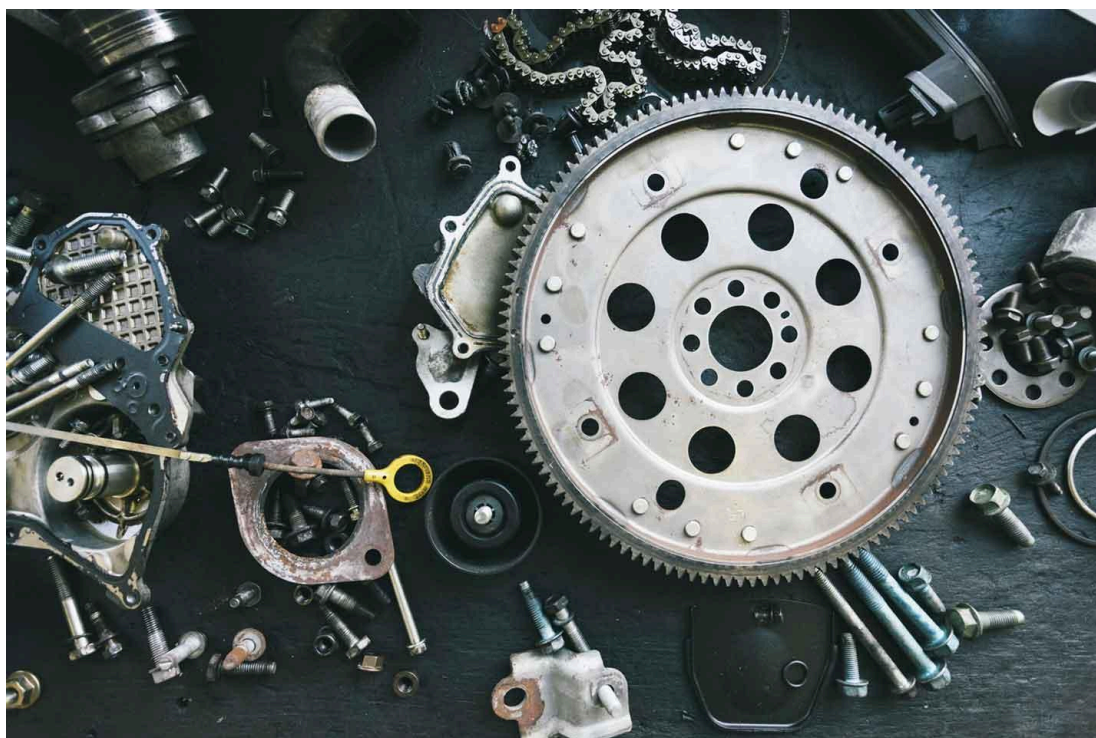


TEMA 4

Bases de datos NO SQL



Progresa
Una formación de futuro

ÍNDICE

CONSULTAS.....	3
Buscar todo.....	3
Igual.....	3
Consultas numéricas.....	3
And.....	4
Or.....	4
Búsquedas en arrays.....	4
Operadores sobre cadenas de texto.....	5
Operadores sobre la respuesta.....	6

CONSULTAS

A continuación veremos los operadores que podemos usar en las consultas a una base de datos de MongoDB con el operador find.

Para realizar las consultas desde el Shell es con **db.nombreColección.find({Consulta})**

Buscar todo

Para devolver todos los documentos de una colección se utilizan las llaves vacías { }

Ejemplo:

```
db.misPelículas.find({})
```

Igual

Se utiliza cuando queremos encontrar un valor concreto:

Ejemplos:

```
db.misPelículas.find({director: "George Lucas"})
```

```
db.misPelículas.find({year: 2000})
```

Consultas numéricas

Mayor >\$gt

Mayor o igual >=\$gte

Menor <\$lt

Menor o igual <=\$lte

Se utiliza cuando queremos valores numéricos por encima o por debajo de un valor deseado.

Ejemplos:

```
db.misPelículas.find({year:{$gt: 2000}})
```

```
db.misPelículas.find({year:{$lt: 2000}})
```

And

El operador and se puede utilizar concatenando con comas las consultas o con \$and. Devuelve los documentos que cumplen todas las condiciones del and.

Ejemplo:

Películas cuyo año es mayor que 2000 **y** su runtime es menor de 120

```
db.misPeliculas.find({year:{$gt: 2000}, runtime:{$lt:120}})
```

Películas cuyo año es mayor de 2000 y menor de 2005

```
db.misPeliculas.find({year:{$gt: 2000, $lt:2005}})
```

Or

El operador or se expresa utilizando \$or:[{consulta1},{consulta2}]

Devuelve los documentos que cumplen todas las condiciones del and.

Ejemplo:

Películas cuyo año es mayor de 2000 **o** su runtime es menor de 120

```
{$or:[{year:{$gt: 2000}}, {runtime:{$lt:120}}]}
```

Búsquedas en arrays

\$in: [Elementos del array] Si queremos la aparición de algún elemento.

\$all: [Elementos del array] Si queremos la aparición de todos los elementos.

Ejemplos:

Películas en las que el país aparezca USA o Spain

```
{countries: {$in:["USA","Spain"]}}
```

Películas en las que el país aparezca USA y Spain

```
{countries: {$all:["USA","Spain"]}}
```

También podemos buscar por el tamaño del array con size

Ejemplo:

```
db.misPelículas.find({countries: {$size: 3}})
```

O por la posición de un elemento del array

Ejemplo:

```
db.misPelículas.find({"countries.0": "Spain"})
```

Operadores sobre cadenas de texto

Podemos buscar cadenas de texto de 2 maneras diferentes, con el operador \$regex o con //.

Tenemos 3 operadores más usados:

- ^ para el comienzo de una cadena de texto.
- \$ para el fin de una cadena de texto.
- nada para el texto en cualquier posición.

Ejemplos:

Películas cuyo título empieza por Star Wars

```
db.misPelículas.find({title: /^Star Wars/})
```

```
db.misPelículas.find({title: {$regex: "^Star Wars"}})
```

Películas cuyo título termina por cowboy

```
db.misPelículas.find({title: /cowboy$/})
```

```
db.misPelículas.find({title: {$regex: "cowboy$"}})
```

Películas cuyo título contenga Star Wars

```
db.misPelículas.find({title: /Star Wars/})
```

```
db.misPelículas.find({title: {$regex: "Star Wars"}})
```

Operadores sobre la respuesta

Podemos operar sobre los documentos recibidos tras la consulta con diversos operadores:

- **count**: para contar los elementos que componen la respuesta.
- **Distinct**: Para sacar los resultados distintos de una consulta.
- **Limit**: Para la cantidad de documentos en la respuesta.
- **Skip**: Para saltar documentos en la respuesta (Junto con el limit definen la paginación).

Ejemplos:

Número de películas de antes del año 2000

```
db.misPeliculas.find({year:{$lt: 2000}}).count()
```

Géneros diferentes de las películas del año 2000

```
db.misPeliculas.distinct("genres",{year:2000})
```