

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

14-11-2021

Proyecto 1^a Evaluación

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Francisco Buiza Pérez

Contenido

1. Introducción 2

2. Inserciones 2

3. Consultas 2

4. Operadores y sentencias usadas no vistos en clase..... 8

1. Introducción

En este proyecto he realizado la base de datos de un concesionario de coches de segunda mano.

En las inserciones he empleado una gran variedad de tipos de datos, como enteros, booleanos, strings, fechas, arrays de documentos, arrays de fechas, arrays de strings, etc.

En las consultas he empleado gran variedad de operadores, como \$and, \$or, \$nor, \$in, \$all, \$not, \$eq, \$neq, etc.

2. Inserciones

Hay 2 colecciones en este proyecto. Una de coches y otra de neumáticos que son usados por los coches de la primera inserción.

Respecto a los coches, tenemos información relevante como marca, modelo, matricula, fechas de matriculación y fabricación, kilómetros, características del motor, medidas de los neumáticos, si tiene abs o no, los extras, el color y el precio.

Respecto a los neumáticos, tenemos la marca, las medidas, el precio, las ofertas y los modelos que hay según sus fechas.

3. Consultas

```
db.coches.distinct("marca")
```

Resultado: nos devuelve todas las marcas que hay en la colección.

```
db.coches.find({
  $and:[
    {extras: { $all: [ /pantalla t.ctil/i , /techo solar/i ] }},
    {fechaFabricacion: {$gte: new Date("2015-01-01")}},
    {"potencia.cv":{$gte:140,$lt:190}},
    {marca:{$not:{$eq:"Renault"}}}
  ]
},
{
  marca:1,
  modelo:1,
  matricula:1,
  kilometros:1,
  extras:1,
  "precioInicial.Eur":1,
  fechaFabricacion:1
}).pretty()
```

Resultado: Muestra los coches que hayan sido fabricados después del 2015 (este incluido), que tengan pantalla táctil y techo solar. Además quiero que el coche tenga al menos 140cv, pero menos de 190cv, y que no sea de la marca Renault.

```
db.coches.find({
  extras: {$exists: true},
  fechaMatriculacion: {$gte: new Date("2015-01-01"), $lte: new
Date("2018-01-01")}
},{
  marca:1,
  modelo:1,
  extras:{$slice:1},
  fechaMatriculacion:1
})
```

Resultado: Muestra el primer extra de los coches que tengan extras y que hayan sido matriculados entre 2015 y 2018. Para saber el ultimo extra deberemos cambiar el \$slice a -1

```
db.coches.find({
  $and:[
    {extras: {$nin:[/techo solar/i,/frenada autom.tica/i]}},
    {"motor.combustible": {$eq: "Gasolina"}},
    {color: "rojo"},
    {"potencia.cv":{$gte: 180}},
    {kilometros:{$lte: 33000}}
  ]
},{
  marca:1,
  modelo:1,
  extras:1,
  "potencia.cv":1,
  kilometros:1,
  "precioInicial.Eur":1
})
```

Resultado: Muestra los coches que tenga no tenga techo solar o frenada automatica, que sea gasolina, de color rojo, con al menos 180 caballos y menos de 33000km

```
db.coches.find({
  $or:[
    {$and:[
      {abs: false},
      {"potencia.cv":{$gte: 190}},
      {"precioInicial.Eur":{$lt: 8000}}
    ]},
    {$and:[
      {abs: {$ne: true}},
      {"potencia.cv": {$gte: 190}},
      {kilometros: {$lte: 50000}},
      {"precioInicial.Eur":{$lte: 13000}}
    ]}
  ]
}).pretty()
```

Resultado: Muestra coches sin ABS, con mas de 190 caballos y que cueste menos de 8000€. Si tiene menos de 50000KM, muestra también lo que cuesten hasta 13000€

```
db.coches.find({
  $nor: [
    {color: "blanco"},
    {extras: {$elemMatch:{$regex:/pantalla t.ctil/i}}}
  ]},
  {
    marca:1,
    modelo:1,
    color:1,
    extras:1,
    fechaFabricacion:1
  }
).sort(
  {
    fechaFabricacion: -1.0
  }
)
```

Resultado: Muestra los nombres de todos los coches que no sean blancos ni que tengan pantalla táctil por orden de fecha de fabricación (de más nuevo a más antiguo).

El resultado solo muestra marca, modelo, color, extras, fechaFabricacion e ID.

```
db.coches.find({
  extras: {$regex:/..?pilot/i}
}).pretty()
```

Resultado: Muestra los coches que tengan un extra que acaba en "pilot"

```
db.coches.find(
  {
    color: "blanco",
    "precioInicial.Eur": {$lt: 16000}
  }
).sort(
  {
    "kilometros" : 1.0
  }
).limit(1);
```

Resultado: Muestra el coche con menos kilómetros que cueste menos de 16000€

```
db.coches.find({
  $or:[
    {"precioInicial.Eur": {$lte: 12000}},
    { $and: [
      {color: {$eq: "blanco"}},
      {"precioInicial.Eur": {$lte: 15000}}
    ]}
  ],
  {
    marca:1,
    modelo:1,
    matricula:1,
    color:1,
    kilometros:1,
    extras:1,
    "precioInicial.Eur":1
  })
}).pretty()
```

Resultado: Muestra la marca, modelo, matricula, color, kilómetros y extras de los coches que no valgan más de 12000€, a no ser de que sea blanco, que lo muestra si cuesta menos de 15000€.

```
db.coches.find({
  $or:[
    {
      "precioInicial.Eur": {$lt: 12000},
      "motor.cilindrada": {$regex: /.?TDI/i}
    },
    {
      "precioInicial.Eur": {$lte: 16000},
      "motor.cilindrada": {$regex: /.?TDI/i},
      extras: {$regex: /asientos calefactables/i},
      kilometros: {$lte: 150000}
    }
  ],
  {
    marca:1,
    modelo:1,
    matricula:1,
    kilometros:1,
    extras:1,
    "precioInicial.Eur":1
  })
}).pretty()
```

Resultado: Muestra todos los coches con motor TDI que cuesten menos de 12000€, y los coches que tienen menos de 150000km, tienen asientos calefactables y valgan menos de 15000€.

```
db.coches.find({
  marca: "Volkswagen",
  modelo: "Golf",
  "motor.cilindrada": {$regex: /1.6 TDI/i}
},
{
  _id:0,
  medidasNeumatico:1,
})
```

Muestra las medidas de los neumáticos de un Golf.

```
db.neumaticos.aggregate([
  {$match:{medidasNeumatico: [
    { ancho: 195 },
    { perfil: 65 },
    { interior: 15 }
  ]},
  marca: "Michelin"}},
  {$project:
    {valor:
      {$multiply:["$precioEur", 4]}
    }
  }
]).pretty()
```

Calcula el precio de 4 ruedas para un Golf.

```
db.coches.aggregate([
  {$project:
    {diasDesmatriculado:
      {$divide:[{$subtract:["$fechaMatriculacion",
"$fechaFabricacion"]},1000*60*60*24]}
    }
  }
]).pretty()
```

Resultado: Calcula los días que se han pasado los coches sin matrícula desde que se fabricaron (Muestra una lista de todos los coches).

```
db.coches.find({
  $and: [
    {marca: {$not:
      {$eq: "Toyota"}}},
    {medidasNeumatico:
      {$elemMatch:
        {perfil:
          {$lt: 50},
        }
      }
    }
  ]
})
```

```
    }]  
  },  
  {  
    marca:1,  
    modelo:1,  
    matricula:1,  
    kilometros:1,  
    extras:1,  
    "precioInicial.Eur":1,  
    medidasNeumatico:1  
  }).pretty()
```

Resultado: Muestra coches que no sean Toyota y con perfil de neumáticos menor a 50. Se muestra la marca, el modelo, la matricula, los kilómetros, los extras, el precio y las medidas de los neumáticos.

```
db.neumaticos.find({  
  modelo: {$in:[new Date("2018-01-01"),new Date("2019-01-01")]},  
  medidasNeumatico: [  
    { ancho: 205 },  
    { perfil: 60 },  
    { interior: 16 }  
  ],  
  oferta: {$in:["3x2"]}  
})
```

Resultado: Muestra ruedas de tamaño 205/60/16 con ofertas "3x2".

```
db.neumaticos.find({  
  $and:[  
    {modelo: {$all:[new Date("2019-01-01"), new Date("2020-01-01")]}},  
    {precioEur: {$lt:58}}  
  ]  
}).sort(  
  {  
    "precioEur" : 1.0  
  }  
)
```

Muestra neumáticos que solo tienen modelos del 2018 y del 2019 que valgan menos de 58€, mostrándolos de menor a mayor precio.

```
db.neumaticos.updateOne(  
  {_id: 7},  
  {  
    $addToSet: {modelo: new Date("2021-01-01")}  
  }  
)
```

Añade la fecha 2021 al array de modelos de la colección neumáticos al neumático con la id 7.

4. Operadores y sentencias usadas no vistos en clase

- **aggregate():** nos permite manipular datos de una manera mucho más compleja que el find. También nos permite realizar operaciones matemáticas.
- **Match:** nos permite definir los datos que vamos a tratar dentro del aggregate().
- **Project:** Nos permite elegir unos campos y realizar las operaciones dentro del aggregate().
- **Update():** Nos permite actualizar campos de la colección sin tener que volver a ejecutar el insert() completo.
- Operador **.?** en regex: Nos permite sustituir un numero de caracteres que no conocemos.

Operaciones realizadas:

- **Multiply:** Multiplica los campos pasados.
- **Substract:** Resta dos campos pasados.
- **Divide:** Divide dos campos pasados.