

## 1. Crear una coleccion llamada AMIGOS y Insertar los siguientes 4 registros

```
db.amigos.insertMany([
  {
    "_id": "5a6f05fbb56b58b447e66801",
    "Nombre": "Luís",
    "Apellidos": "Martinez",
    "Edad": 18,
    "Aficiones": [
      "fútbol",
      "senderismo",
      "tenis"
    ],
    "Amigos": [
      {
        "Nombre": "Monica",
        "Edad": 20
      },
      {
        "Nombre": "Andrés",
        "Edad": 24
      }
    ]
  },
  {
    "_id": "5a6f05fbb56b58b447e66802",
    "Nombre": "Marisa",
```

```
"Apellidos": "Garcia",
"Edad": 18,
"Aficiones": [
  "senderismo",
  "tenis",
  "pintura"
],
"Amigos": [
  {
    "Nombre": "Monica",
    "Edad": 20
  }
]
},
{
  "_id": "5a6f05fbb56b58b447e66803",
  "Nombre": "Joaquin",
  "Apellidos": "Rodriguez",
  "Edad": 21,
  "Aficiones": [
    "fútbol",
    "natación"
  ],
  "Amigos": [
    {
      "Nombre": "Andrés",
      "Edad": 24
```

```
    }  
  ]  
},  
{  
  "_id": "5a6f05fbb56b58b447e66804",  
  "Nombre": "Luís",  
  "Apellidos": "Martinez",  
  "Edad": 20,  
  "Aficiones": [  
    "natación",  
    "culturismo"  
  ],  
  "Amigos": [  
    {  
      "Nombre": "Juan",  
      "Edad": 40  
    },  
    {  
      "Nombre": "Antonio",  
      "Edad": 52  
    }  
  ]  
}  
]);
```

El resultado es de la forma

```
< {
  acknowledged: true,
  insertedIds: {
    '0': '5a6f05fbb56b58b447e66801',
    '1': '5a6f05fbb56b58b447e66802',
    '2': '5a6f05fbb56b58b447e66803',
    '3': '5a6f05fbb56b58b447e66804'
  }
}
```

## 2. Realizar las siguientes consultas:

`db.amigos.find().count()`

1. **Mostrar todos los datos de la colección amigos.** 4 registros
2. **Personas mayores de 18 años.** 2 registros sale todos los datos (Luis martinez y joaquin rodriguez) `db.amigos.find({Edad:{$gt:18}})`
3. **Mostrar nombre y apellidos de las ‘Marisas’ encontradas** (sale automáticamente el id). 1 registro (Marisa Garcia) `db.amigos.find({Nombre:"Marisa"},{_id:1,Nombre:1,Apellidos:1})`
4. **Mostrar nombre y apellidos de las ‘Marisas’ encontradas pero sin el id (para ello ponemos el campo `_id:0`).** 1 registro (Marisa Garcia) `db.amigos.find({Nombre:"Marisa"},{_id:0,Nombre:1,Apellidos:1})`
5. **Mostrar nombre y apellidos de los mayores de 18 años.** 2 registros (Luis martinez y Joaquin Rodriguez) `db.amigos.find({Edad:{$gt:18}}, {_id:0,Nombre:1,Apellidos:1})`
6. **Mostrar nombre y apellidos de los menores de 20 años.** 2 registros (Luis martinez y marisa garcia) `db.amigos.find({Edad:{$lt:20}}, {_id:0,Nombre:1,Apellidos:1})`
7. **Mostrar los amigos que se llamen “Marisa” y que sean mayores de 17 años** `db.amigos.find({Nombre:"Marisa", Edad:{$gt:17}})`
8. **Mostrar los amigos que se llamen “Marisa” o que tengan más de 20 años.** 2 registros (marisa y joaquin) `db.amigos.find({ $or:[{Nombre:"Marisa"}, {Edad:{$gt:20}}] })`
9. **Ordenar descendentemente. (función sort y en el campo poner un 1)** `db.amigos.find({}, {Nombre:1}).sort({Nombre:1})`

```
< {
  _id: '5a6f05fbb56b58b447e66803',
  Nombre: 'Joaquín'
}
{
  _id: '5a6f05fbb56b58b447e66801',
  Nombre: 'Luís'
}
{
  _id: '5a6f05fbb56b58b447e66804',
  Nombre: 'Luís'
}
{
  _id: '5a6f05fbb56b58b447e66802',
  Nombre: 'Marisa'
}
```

10. **Ordenar ascendentemente.** (función sort y en el campo poner un -1)

```
db.amigos.find({}, {Nombre:1}).sort({Nombre:-1})
```

```
< {
  _id: '5a6f05fbb56b58b447e66802',
  Nombre: 'Marisa'
}
{
  _id: '5a6f05fbb56b58b447e66801',
  Nombre: 'Luís'
}
{
  _id: '5a6f05fbb56b58b447e66804',
  Nombre: 'Luís'
}
{
  _id: '5a6f05fbb56b58b447e66803',
  Nombre: 'Joaquín'
}
```

11. **Contar.** Sale 4

```
db.amigos.find().count()
```

12. **Limitar la búsqueda a 2 registros.** (añadir al comando para ordenar del ejercicio 9 limit(2). Salen joaquin y Luis.

```
db.amigos.find({}, {Nombre:1}).sort({Nombre:1}).limit(2)
```

13. **Buscar a partir del 2º registro** (Luis y marisa).

```
db.amigos.find({}, {Nombre:1}).sort({Nombre:1}).skip(2)
```

14. **Buscar a partir del 3º registro** (marisa).

```
db.amigos.find({}, {Nombre:1}).sort({Nombre:1}).skip(3)
```

15. **Buscar a partir del 2º registro** (ninguno porque solo hay 4 registros).

16. **Contar cuantos amigos tiene cada persona.** ES COMO UN GROUPBY HAVING EN SQL, pero aquí se usa AGREGATE del modo siguiente (size es el tamaño total de la colección)

```
db.amigos.aggregate([
  { $project: { Campo a proyectar: 1, Variable que saldra: { $size:
"$amigos" } } }
]);
```

```
db.amigos.aggregate([
  { $project: { Nombre: 1, CantidadAmigos: { $size:"$Amigos" } } }
]);
```

```
< {
  _id: '5a6f05fbb56b58b447e66801',
  Nombre: 'Luís',
  CantidadAmigos: 2
}
{
  _id: '5a6f05fbb56b58b447e66802',
  Nombre: 'Marisa',
  CantidadAmigos: 1
}
{
  _id: '5a6f05fbb56b58b447e66803',
  Nombre: 'Joaquín',
  CantidadAmigos: 1
}
{
  _id: '5a6f05fbb56b58b447e66804',
  Nombre: 'Luís',
  CantidadAmigos: 2
}
```

En el ejemplo hemos supuesto “Variable que saldrá” a “CantidadAmigos”.

## 17. Crear un array con los elementos de salida donde deben aparecer las marisas o los que sean mayores de 20

```
// Ejecutar la consulta y almacenar el resultado en un array
let resultados = db.amigos.find({
  $or: [
    { Nombre: "Marisa" },
    { Edad: { $gt: 20 } }
  ]
}).toArray();

let resultados = db.amigos.find({ $or:[
  {Nombre:"Marisa"},
  {Edad:{$gt:20}}
]
}).toArray();

// Imprimir el array de resultados
printjson(resultados);
```

Comprobar el resultado y ya sabéis como guardaros en un array una parte de la colección.

```
[
  {
    _id: '5a6f05fbb56b58b447e66802',
    Nombre: 'Marisa',
    Apellidos: 'García',
    Edad: 18,
    Aficiones: [ 'senderismo', 'tenis', 'pintura' ],
    Amigos: [ [Object] ]
  },
  {
    _id: '5a6f05fbb56b58b447e66803',
    Nombre: 'Joaquín',
    Apellidos: 'Rodríguez',
    Edad: 21,
    Aficiones: [ 'fútbol', 'natación' ],
    Amigos: [ [Object] ]
  }
]
```

18. Mostrar solo el primer registro de la colección amigos. Bien con `findOne()` o con `limit(1)` `db.amigos.findOne()`

```
{
  _id: '5a6f05fbb56b58b447e66801',
  Nombre: 'Luís',
  Apellidos: 'Martínez',
  Edad: 18,
  Aficiones: [
    'fútbol',
    'senderismo',
    'tenis'
  ],
  Amigos: [
    {
      Nombre: 'Monica',
      Edad: 20
    },
    {
      Nombre: 'Andrés',
      Edad: 24
    }
  ]
}
```

19. Mostrar solo el ultimo registro de la colección amigos. Pista: usar `sort` y `limit`.

`db.amigos.find().sort({_id:-1}).limit(1)`

```
< {
  _id: '5a6f05fbb56b58b447e66804',
  Nombre: 'Luís',
  Apellidos: 'Martínez',
  Edad: 20,
  Aficiones: [
    'natación',
    'culturismo'
  ],
  Amigos: [
    {
      Nombre: 'Juan',
      Edad: 40
    },
    {
      Nombre: 'Antonio',
      Edad: 52
    }
  ]
}
```



## 20. Mostrar las personas que tengan 18 años ordenando sus amigos descendientemente. Usar aggregate.

```
db.amigos.aggregate([
  { $match: { Edad: 18 } },
  { $unwind: "$Amigos" },
  { $sort: { "Amigos.Nombre": 1 } },
  { $group: {
    _id: "$_id",
    Nombre: { $first: "$Nombre" },
    Apellidos: { $first: "$Apellidos" },
    Edad: { $first: "$Edad" },
    Amigos: { $push: "$Amigos" }
  }}
])
```

```
{
  "_id": "5a6f05fbb56b58b447e66801",
  "Nombre": "Luis",
  "Apellidos": "Martinez",
  "Edad": 18,
  "Aficiones": [
    "fútbol",
    "senderismo",
    "tenis"
  ],
  "Amigos": [
    {
      "Nombre": "Andrés",
      "Edad": 24
    },
    {
      "Nombre": "Monica",
      "Edad": 20
    }
  ]
},
{
  "_id": "5a6f05fbb56b58b447e66802",
  "Nombre": "Marisa",
  "Apellidos": "Garcia",
  "Edad": 18,
  "Aficiones": [
    "senderismo",
    "tenis",
    "pintura"
  ],
  "Amigos": [
    {
      "Nombre": "Monica",
      "Edad": 20
    }
  ]
}
```

## 21. Contar cuantas personas tienen cada afición agrupando a las personas por sus aficiones.

Usar en el aggregate \$unwind para descomponer el array de aficiones de modo que se pondría por ejemplo: \$unwind: "Aficiones" y entonces cuando usemos el identificador de la colección le ponemos también \$Aficiones

```
db.amigos.aggregate([
  { $unwind: "$Aficiones" },
  { $group: {
    _id: "$Aficiones",
    total: { $sum: 1 }
  }}
])
```

```
< {
  "_id": "culturismo",
  "Cantidad": 1
},
{
  "_id": "senderismo",
  "Cantidad": 2
},
{
  "_id": "fútbol",
  "Cantidad": 2
},
{
  "_id": "pintura",
  "Cantidad": 1
},
{
  "_id": "tenis",
  "Cantidad": 2
},
{
  "_id": "natación",
  "Cantidad": 2
}
```

## 22. Mostrar el nombre de los amigos de todas las personas de la colección.

```
db.amigos.aggregate([
  { $unwind: "$Amigos" },
  { $group: {
    _id: "$Amigos._id",
    Nombres: { $push: "$Amigos.Nombre" }
  }}
])
```

```
< {
  _id: '5a6f05fbb56b58b447e66801',
  Nombre: 'Monica'
}
{
  _id: '5a6f05fbb56b58b447e66801',
  Nombre: 'Andrés'
}
{
  _id: '5a6f05fbb56b58b447e66802',
  Nombre: 'Monica'
}
{
  _id: '5a6f05fbb56b58b447e66803',
  Nombre: 'Andrés'
}
{
  _id: '5a6f05fbb56b58b447e66804',
  Nombre: 'Juan'
}
{
  _id: '5a6f05fbb56b58b447e66804',
  Nombre: 'Antonio'
}
```

## 23. Insertar otro registro mas

```
db.amigos.insertOne({
  "_id": "5a6f05fbb56b58b447e66805",
  "Nombre": "Carlos",
  "Apellidos": "Pérez",
  "Edad": 22,
  "Aficiones": [
    "baloncesto",
    "videojuegos"
  ],
  "Amigos": [
    {
      "Nombre": "Juan",
      "Edad": 30
    },
    {
      "Nombre": "Ana",
```

```

> db.amigos.insertOne({
  "_id": "5a6f05fbb56b58b447e66805", // Asegúrate de que el _id sea único
  "Nombre": "Carlos",
  "Apellidos": "Pérez",
  "Edad": 22,
  "Aficiones": [
    "baloncesto",
    "videojuegos"
  ],
  "Amigos": [
    {
      "Nombre": "Juan",
      "Edad": 30
    },
    {
      "Nombre": "Ana",
      "Edad": 25
    }
  ]
});
< {
  acknowledged: true,
  insertedId: '5a6f05fbb56b58b447e66805'
}

```

```

> db.amigos.find().count()
< 5

```

Si hacemos count vemos que hay 5

Vamos a hacer operaciones de modificación y borrado sobre este registro creado teniendo en cuenta

Modificar

- `updateOne()` se utiliza para modificar un solo documento.
- `\$set` para cambiar un campo específico.
- `\$addToSet` para añadir elementos a un array sin duplicados.
- `\$pull` para eliminar elementos de un array.

Borrar:

- `deleteOne()` se utiliza para eliminar un documento completo.
- Puedes modificar el contenido de un documento sin eliminarlo usando `updateOne()` con `\$set`.

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

## 24. Modificar el nombre de Carlos por José Carlos.

Si consultamos sale

```
db.amigos.updateOne(
  { "Nombre": "Carlos" },
  { $set: { "Nombre": "José Carlos" } }
)
```

```
{
  _id: '5a6f05fbb56b58b447e66805',
  Nombre: 'Jose Carlos',
  Apellidos: 'Pérez',
  Edad: 22,
  Aficiones: [
    'baloncesto',
    'videojuegos'
  ],
  Amigos: [
    {
      Nombre: 'Juan',
      Edad: 30
    },
    {
      Nombre: 'Ana',
      Edad: 25
    }
  ]
}
```

## 25. Añadir a Jose Carlos una nueva afición por ejemplo futbol

```
db.amigos.updateOne(
  { "Nombre": "José Carlos" },
  { $addToSet: { "Aficiones": "petanca" } }
)
```

```
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Si consultamos sale

```

{
  _id: '5a6f05fbb56b58b447e66805',
  Nombre: 'Jose Carlos',
  Apellidos: 'Pérez',
  Edad: 22,
  Aficiones: [
    'baloncesto',
    'videojuegos',
    'futbol'
  ],
  Amigos: [
    {
      Nombre: 'Juan',
      Edad: 30
    },
    {
      Nombre: 'Ana',
      Edad: 25
    }
  ]
}

```

**26. Comprobar que se ha añadido correctamente mostrando el nombre, apellidos y todas las aficiones de Jose Carlos (el id no nos interesa)**

```

< {
  Nombre: 'Jose Carlos',
  Apellidos: 'Pérez',
  Aficiones: [
    'baloncesto',
    'videojuegos',
    'futbol'
  ]
}

```

**27. Borrar a Ana de los amigos de Jose Carlos.**

```

db.amigos.updateOne(
  { "Nombre": "José Carlos" },
  { $pull: { Amigos: {Nombre:"Monica"} } }
)

```

```

{
  _id: '5a6f05fbb56b58b447e66805',
  Nombre: 'Jose Carlos',
  Apellidos: 'Pérez',
  Edad: 22,
  Aficiones: [
    'baloncesto',
    'videojuegos',
    'futbol'
  ],
  Amigos: [
    {
      Nombre: 'Juan',
      Edad: 30
    }
  ]
}

```

## 28. Eliminar el registro completo de Jose Carlos (deleteOne())

```
{  
  acknowledged: true,  
  deletedCount: 1  
}
```

```
> db.amigos.find().count()  
< 4
```

```
db.amigos.deleteOne(  
  { "Nombre": "José Carlos" }  
)
```