

Ejercicios

1. Crea un bucle en JavaScript que introduzca 10 elementos con números aleatorios en la colección **numeros**

```
> for(i = 0; i < 10; i++) { db.numeros.insertOne({numero: _rand()})}

> for(i = 0; i < 10; i++) { db.numeros.insertOne({numero: _rand()})}
{
  "acknowledged" : true,
  "insertedId" : ObjectId("5c7da8598ba45928dae715f3")
}
```

Comprobamos que en efecto, el método **_rand()** ha generado números aleatorios para cada uno de los documentos creados en la colección **numeros**.

```
> db.numeros.find()
{ "_id" : ObjectId("5c7da8598ba45928dae715ea"), "numero" : 0.24257829878479242 }
{ "_id" : ObjectId("5c7da8598ba45928dae715eb"), "numero" : 0.013469574507325888 }
{ "_id" : ObjectId("5c7da8598ba45928dae715ec"), "numero" : 0.38313884986564517 }
{ "_id" : ObjectId("5c7da8598ba45928dae715ed"), "numero" : 0.4146526902914047 }
{ "_id" : ObjectId("5c7da8598ba45928dae715ee"), "numero" : 0.06776897283270955 }
{ "_id" : ObjectId("5c7da8598ba45928dae715ef"), "numero" : 0.9931269292719662 }
{ "_id" : ObjectId("5c7da8598ba45928dae715f0"), "numero" : 0.4843080462887883 }
{ "_id" : ObjectId("5c7da8598ba45928dae715f1"), "numero" : 0.7653377656824887 }
{ "_id" : ObjectId("5c7da8598ba45928dae715f2"), "numero" : 0.03183381538838148 }
{ "_id" : ObjectId("5c7da8598ba45928dae715f3"), "numero" : 0.030935481656342745 }
> █
```

2. Inserta 3 elementos definiendo su **_id** a los siguientes campos en la colección **elementos**

```
> db.elementos.insertMany([{_id:"primero"},{_id:"segundo"},{_id:"tercero"}])

[> db.elementos.insertMany([{_id:"primero"},{_id:"segundo"},{_id:"tercero"}])
{
  "acknowledged" : true,
  "insertedIds" : [
    "primero",
    "segundo",
    "tercero"
  ]
}
```

Si ahora intentáramos añadir más documentos a nuestra colección **elementos** y uno de ellos comparte **_id** veríamos lo siguiente:

```
> db.elementos.insertMany([{_id:"cuarto"},{_id:"segundo"},{_id:"sexto"}])

> db.elementos.insertMany([{_id:"cuarto"},{_id:"segundo"},{_id:"sexto"}])
2019-03-04T23:43:19.531+0100 E QUERY [js] BulkWriteError: write error at item 1 in bulk operation :
BulkWriteError({
  "writeErrors" : [
    {
      "index" : 1,
      "code" : 11000,
      "errmsg" : "E11000 duplicate key error collection: ejercicios.elementos index:
[_id_ dup key: { : \"segundo\" }",
      "op" : {
        "_id" : "segundo"
      }
    }
  ],
  "writeConcernErrors" : [ ],
  "nInserted" : 1,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
BulkWriteError@src/mongo/shell/bulk_api.js:369:48
BulkWriteResult/this.toError@src/mongo/shell/bulk_api.js:333:24
Bulk/this.execute@src/mongo/shell/bulk_api.js:1173:1
DBCollection.prototype.insertMany@src/mongo/shell/crud_api.js:314:5
@(shell):1:1
```

Obtendremos un **error de clave _id duplicada**. Se habrá añadido el elemento anterior al que haya dado el error y acto seguido se habrá detenido la inserción, haciendo que el sexto documento no se introduzca en la colección.

```
[> db.elementos.find()
{ "_id" : "primero" }
{ "_id" : "segundo" }
{ "_id" : "tercero" }
{ "_id" : "cuarto" }
```

Si en cambio, añadimos **ordered** a **false** a la misma sentencia, el error no detendrá las siguientes inserciones:

```
> db.elementos.insertMany([{_id:"quinto"},{_id:"segundo"},{_id:"sexto"}], {ordered: false})

> db.elementos.insertMany([{_id:"quinto"},{_id:"segundo"},{_id:"sexto"}], {ordered: false})
2019-03-04T23:47:46.376+0100 E QUERY [js] BulkWriteError: write error at item 1 in bulk operation :
BulkWriteError({
  "writeErrors" : [
    {
      "index" : 1,
      "code" : 11000,
      "errmsg" : "E11000 duplicate key error collection: ejercicios.elementos index: _id
dup key: { : \"segundo\" }",
      "op" : {
        "_id" : "segundo"
      }
    }
  ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
BulkWriteError@src/mongo/shell/bulk_api.js:369:48
BulkWriteResult/this.toError@src/mongo/shell/bulk_api.js:333:24
Bulk/this.execute@src/mongo/shell/bulk_api.js:1173:1
DBCollection.prototype.insertMany@src/mongo/shell/crud_api.js:314:5
@(shell):1:1
```

Podremos ver que se han insertado (**nInserted**) dos, mientras que el elemento con **_id** repetido no se habrá incluido.

```
[> db.elementos.find()
{ "_id" : "primero" }
{ "_id" : "segundo" }
{ "_id" : "tercero" }
{ "_id" : "cuarto" }
{ "_id" : "quinto" }
{ "_id" : "sexto" }
```

3. Crea 4 o más documentos en la colección **amigos** que contengan las siguientes claves:

- **nombre**
- **apellidos**
- **nacimiento**
 - **lugar**
 - **fecha**
 - **día**
 - **mes**
 - **año**
- **aficiones (array)**
- **mascotas (número entero)**

* El máximo de aficiones es 3. Se puede elegir entre “deporte”, “programar”, “bailar”, “descansar”, “viajar”, “beber” y “saltar”.

Vamos a introducir documentos a través del comando **insertMany**:

```
> db.amigos.insertMany([{\nombre:"Martín", apellidos:"San José", nacimiento:
{\lugar:"Canarias", fecha: {\día: 14, mes: "Septiembre", año: 1991}}, aficiones:
["programar", "viajar", "beber"], mascotas: 6}, {\nombre:"Andrés", apellidos:"San
José", nacimiento: {\lugar:"Ucrania", fecha: {\día: 11, mes: "Septiembre", año: 1997}},
aficiones: ["descansar", "viajar", "deporte"], mascotas: 1}, {\nombre:"Juan Carlos",
apellidos:"Moreno", nacimiento: {\lugar:"Valencia", fecha: {\día: 9, mes: "Enero", año:
1997}}, aficiones: ["descansar"], mascotas: 0}, {\nombre:"Alejandro",
apellidos:"García", nacimiento: {\lugar:"Valencia", fecha: {\día: 2, mes: "Febrero",
año: 1980}}, aficiones: ["descansar", "bailar"], mascotas: 2}])
```

Comprobamos que se han insertado correctamente en la colección **amigos**

```
> db.amigos.find()
{ "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "apellidos" : "San José", "nacimiento" : { "lugar" : "Canarias", "fecha" :
{ "día" : 14, "mes" : "Septiembre", "año" : 1991 } }, "aficiones" : [ "programar", "viajar", "beber" ], "mascotas" : 6 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania", "fecha" : {
"día" : 11, "mes" : "Septiembre", "año" : 1997 } }, "aficiones" : [ "descansar", "viajar", "deporte" ], "mascotas" : 1 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "apellidos" : "Moreno", "nacimiento" : { "lugar" : "Valencia", "fecha"
: { "día" : 9, "mes" : "Enero", "año" : 1997 } }, "aficiones" : [ "descansar" ], "mascotas" : 0 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García", "nacimiento" : { "lugar" : "Valencia", "fecha" :
{ "día" : 2, "mes" : "Febrero", "año" : 1980 } }, "aficiones" : [ "descansar", "bailar" ], "mascotas" : 2 }
```

4. En la colección **amigos** encuentra el primer documento que haya nacido en Valencia

```
> db.amigos.findOne({"nacimiento.lugar": "Valencia"})
```

```
[> db.amigos.findOne({"nacimiento.lugar": "Valencia"})
{
  "_id" : ObjectId("5c7db43c8ba45928dae715f6"),
  "nombre" : "Juan Carlos",
  "apellidos" : "Moreno",
  "nacimiento" : {
    "lugar" : "Valencia",
    "fecha" : {
      "dia" : 9,
      "mes" : "Enero",
      "año" : 1997
    }
  },
  "aficiones" : [
    "descansar"
  ],
  "mascotas" : 0
}
```

5. En la colección **amigos** buscar todos los documentos que hayan nacido en el año 1980

```
> db.amigos.find({"nacimiento.fecha.año": 1980})
```

```
> db.amigos.find({"nacimiento.fecha.año": 1980})
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García",
  "nacimiento" : { "lugar" : "Valencia", "fecha" : { "dia" : 2, "mes" : "Febrero", "año" : 1980 }
}, "_aficiones" : [ "descansar", "bailar" ], "mascotas" : 2 }
```

6. En la colección **amigos** busca todos los documentos que hayan nacido en un día del mes superior al 10

```
> db.amigos.find({"nacimiento.fecha.dia": {$gt: 10}})
```

```
> db.amigos.find({"nacimiento.fecha.dia": {$gt: 10}})
{ "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "apellidos" : "San José", "nacimiento" : { "lugar" : "Canarias", "fecha" : { "dia" : 14, "mes" :
  "Septiembre", "año" : 1991 } }, "aficiones" : [ "programar", "viajar", "beber" ], "mascotas" : 6 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania", "fecha" : { "dia" : 11, "mes" :
  "Septiembre", "año" : 1997 } }, "aficiones" : [ "descansar", "viajar", "deporte" ], "mascotas" : 1 }
```

7. En la colección **amigos** busca todos los documentos que hayan nacido Valencia o Madrid

```
> db.amigos.find({"nacimiento.lugar": {$in: ["Madrid", "Valencia"]}})
```

```
> db.amigos.find({"nacimiento.lugar": {$in: ["Madrid", "Valencia"]}})
{ "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "apellidos" : "Moreno", "nacimiento" : { "lugar" : "Valencia", "fecha" : { "dia" : 9, "mes" : "Enero", "año" : 1997 } }, "aficiones" : [ "descansar" ], "mascotas" : 0 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García", "nacimiento" : { "lugar" : "Valencia", "fecha" : { "dia" : 2, "mes" : "Febrero", "año" : 1980 } }, "aficiones" : [ "descansar", "bailar" ], "mascotas" : 2 }
```

8. En la colección **amigos** busca todos los documentos cuyo lugar de nacimiento sea Valencia y que su año de nacimiento sea mayor o igual a 1980

```
> db.amigos.find({"nacimiento.lugar": "Valencia", "nacimiento.fecha.año": {$gte: 1980}})
```

```
> db.amigos.find({"nacimiento.lugar": "Valencia", "nacimiento.fecha.año": {$gte: 1980}})
{ "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "apellidos" : "Moreno", "nacimiento" : { "lugar" : "Valencia", "fecha" : { "dia" : 9, "mes" : "Enero", "año" : 1997 } }, "aficiones" : [ "descansar" ], "mascotas" : 0 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García", "nacimiento" : { "lugar" : "Valencia", "fecha" : { "dia" : 2, "mes" : "Febrero", "año" : 1980 } }, "aficiones" : [ "descansar", "bailar" ], "mascotas" : 2 }
```

9. En la colección **amigos** busca todos los documentos que tengan “beber” como afición

```
> db.amigos.find({"aficiones": "beber"})
```

```
> db.amigos.find({"aficiones": "beber"})
{ "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "apellidos" : "San José", "nacimiento" : { "lugar" : "Canarias", "fecha" : { "dia" : 14, "mes" : "Septiembre", "año" : 1991 } }, "aficiones" : [ "programar", "viajar", "beber" ], "mascotas" : 6 }
```

10. En la colección **amigos** busca todos los documentos que tengan “beber” como única afición

```
> db.amigos.find({"aficiones": ["beber"]})
```

Dado que no tenemos documentos con una única afición de **beber** no obtendremos ningún documento resultante.

10. En la colección **amigos** busca todos los documentos que solo tengan UNA afición.

```
> db.amigos.find({"aficiones": {$size: 1}})
```

```
{ "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "apellidos" : "Moreno", "nacimiento" : { "lugar" : "Valencia", "fecha" : { "dia" : 9, "mes" : "Enero", "año" : 1997 } }, "aficiones" : [ "descansar" ], "mascotas" : 0 }
```

11. En la colección **amigos** busca todos los documentos que no hayan nacido en Valencia.

```
> db.amigos.find({"nacimiento.lugar": {$ne: "Valencia"}})
```

```
> db.amigos.find({"nacimiento.lugar": {$ne: "Valencia"}})
{ "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "apellidos" : "San José", "nacimiento" : { "lugar" : "Canarias",
"fecha" : { "dia" : 14, "mes" : "Septiembre", "año" : 1991 } }, "aficiones" : [ "programar", "viajar", "beber" ], "mascotas" : 6 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania",
"fecha" : { "dia" : 11, "mes" : "Septiembre", "año" : 1997 } }, "aficiones" : [ "descansar", "viajar", "deporte" ], "mascotas" : 1 }
```

12. En la colección **amigos** devuelve solo los nombres de los documentos.

```
> db.amigos.find({}, {_id: false, nombre: 1})
```

```
[> db.amigos.find({}, {_id: false, nombre: 1})
{ "nombre" : "Martín" }
{ "nombre" : "Andrés" }
{ "nombre" : "Juan Carlos" }
{ "nombre" : "Alejandro" }
```

13. En la colección **amigos** devuelve solo los TRES PRIMEROS elementos.

```
> db.amigos.find().limit(3)
```

```
> db.amigos.find().limit(3)
{ "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "apellidos" : "San José", "nacimiento" : { "lugar" : "Canarias",
"fecha" : { "dia" : 14, "mes" : "Septiembre", "año" : 1991 } }, "aficiones" : [ "programar", "viajar", "beber" ], "mascotas" : 6 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania",
"fecha" : { "dia" : 11, "mes" : "Septiembre", "año" : 1997 } }, "aficiones" : [ "descansar", "viajar", "deporte" ], "mascotas" : 1 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "apellidos" : "Moreno", "nacimiento" : { "lugar" : "Valencia",
"fecha" : { "dia" : 9, "mes" : "Enero", "año" : 1997 } }, "aficiones" : [ "descansar" ], "mascotas" : 0 }
```

14. En la colección **amigos** devuelve solo los DOS PRIMEROS ordenados por nombre (ascendente)

Esto se puede hacer de dos maneras:

```
> db.amigos.find().sort({nombre: 1}).limit(2)
```

```
[> db.amigos.find().sort({nombre: 1}).limit(2)
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García", "nacimiento" : { "lugar" : "Valencia",
"fecha" : { "dia" : 2, "mes" : "Febrero", "año" : 1980 } }, "aficiones" : [ "descansar", "bailar" ], "mascotas" : 2 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania",
"fecha" : { "dia" : 11, "mes" : "Septiembre", "año" : 1997 } }, "aficiones" : [ "descansar", "viajar", "deporte" ], "mascotas" : 1 }
```

```
> db.amigos.find().limit(2).sort({nombre: 1})
```

```
> db.amigos.find().limit(2).sort({nombre: 1})
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García", "nacimiento" : { "lugar" : "Valencia",
"fecha" : { "dia" : 2, "mes" : "Febrero", "año" : 1980 } }, "aficiones" : [ "descansar", "bailar" ], "mascotas" : 2 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania",
"fecha" : { "dia" : 11, "mes" : "Septiembre", "año" : 1997 } }, "aficiones" : [ "descansar", "viajar", "deporte" ], "mascotas" : 1 }
```

15. En la colección **amigos** devuelve todos los documentos ordenados por edad (descendente) y luego nombre (ascendente).

```
> db.amigos.find({}).sort({"nacimiento.fecha.año": 1, "nacimiento.fecha.mes": 1, "nacimiento.fecha.dia": 1, nombre: 1})
```

```
> db.amigos.find({}).sort({"nacimiento.fecha.año": 1, "nacimiento.fecha.mes": 1, "nacimiento.fecha.dia": 1, nombre: 1})
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García", "nacimiento" : { "lugar" : "Valencia", "fecha" : { "dia" : 2, "mes" : "Febrero", "año" : 1980 } }, "aficiones" : [ "descansar", "bailar" ], "mascotas" : 2 }
{ "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "apellidos" : "San José", "nacimiento" : { "lugar" : "Canarias", "fecha" : { "dia" : 14, "mes" : "Septiembre", "año" : 1991 } }, "aficiones" : [ "programar", "viajar", "beber" ], "mascotas" : 6 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "apellidos" : "Moreno", "nacimiento" : { "lugar" : "Valencia", "fecha" : { "dia" : 9, "mes" : "Enero", "año" : 1997 } }, "aficiones" : [ "descansar" ], "mascotas" : 0 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania", "fecha" : { "dia" : 11, "mes" : "Septiembre", "año" : 1997 } }, "aficiones" : [ "descansar", "viajar", "deporte" ], "mascotas" : 1 }
```

16. En la colección **amigos** devuelve todos los elementos pero solo muestra el nombre y su primera afición.

```
> db.amigos.find({}, {nombre: 1, aficiones: {$slice: 1}})
```

```
> db.amigos.find({}, {nombre: 1, aficiones: {$slice: 1}})
{ "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "aficiones" : [ "programar" ] }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "aficiones" : [ "descansar" ] }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "aficiones" : [ "descansar" ] }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "aficiones" : [ "descansar" ] }
```


17. En la colección **amigos** actualiza todas las ocurrencias de Valencia por València.

```
> db.amigos.updateMany({"nacimiento.lugar": "Valencia"}, {$set: {"nacimiento.lugar": "València"}})
```

```
> db.amigos.updateMany({"nacimiento.lugar": "Valencia"}, {$set: {"nacimiento.lugar": "València"}})
{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }
```

Comprobamos que en efecto se ha actualizado correctamente el campo lugar de nacimiento. Además, lo mostramos con **.pretty()** para que se nos muestre de una forma más agradable:

```
> db.amigos.find().pretty()
```

```
> db.amigos.find().pretty()
{
  "_id" : ObjectId("5c7db3318ba45928dae715f4"),
  "nombre" : "Martín",
  "apellidos" : "San José",
  "nacimiento" : {
    "lugar" : "Canarias",
    "fecha" : {
      "dia" : 14,
      "mes" : "Septiembre",
      "año" : 1991
    }
  },
  "aficiones" : [
    "programar",
    "viajar",
    "beber"
  ],
  "mascotas" : 6
}
{
  "_id" : ObjectId("5c7db43c8ba45928dae715f5"),
  "nombre" : "Andrés",
  "apellidos" : "San José",
  "nacimiento" : {
    "lugar" : "Ucrania",
    "fecha" : {
      "dia" : 11,
      "mes" : "Septiembre",
      "año" : 1997
    }
  },
  "aficiones" : [
    "descansar",
    "viajar",
    "deporte"
  ],
  "mascotas" : 1
}
{
  "_id" : ObjectId("5c7db43c8ba45928dae715f6"),
  "nombre" : "Juan Carlos",
  "apellidos" : "Moreno",
  "nacimiento" : {
    "lugar" : "València",
    "fecha" : {
      "dia" : 9,
      "mes" : "Enero",
      "año" : 1997
    }
  },
  "aficiones" : [
    "descansar"
  ],
  "mascotas" : 8
}
{
  "_id" : ObjectId("5c7db43c8ba45928dae715f7"),
  "nombre" : "Alejandro",
  "apellidos" : "García",
  "nacimiento" : {
    "lugar" : "València",
    "fecha" : {
      "dia" : 2,
      "mes" : "Febrero",
      "año" : 1988
    }
  },
  "aficiones" : [
    "descansar",
    "bailar"
  ],
  "mascotas" : 2
}
_
```

18. En la colección **amigos** decrementa todos los años de nacimiento en 1.

```
> db.amigos.updateMany({}, {$inc: {"nacimiento.fecha.año": -1}})
```

```
> db.amigos.updateMany({}, {$inc: {"nacimiento.fecha.año": -1}})
{ "_acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 4 }
```

```
> db.amigos.find()
```

```
> db.amigos.find()
{ "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "apellidos" : "San José", "nacimiento" : { "lugar" : "Canarias", "fecha" : { "dia" : 14, "mes" : "Septiembre", "año" : 1990 } }, "aficiones" : [ "programar", "viajar", "beber" ], "mascotas" : 6 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania", "fecha" : { "dia" : 11, "mes" : "Septiembre", "año" : 1996 } }, "aficiones" : [ "descansar", "viajar", "deporte" ], "mascotas" : 1 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "apellidos" : "Moreno", "nacimiento" : { "lugar" : "València", "fecha" : { "dia" : 9, "mes" : "Enero", "año" : 1996 } }, "aficiones" : [ "descansar" ], "mascotas" : 0 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García", "nacimiento" : { "lugar" : "València", "fecha" : { "dia" : 2, "mes" : "Febrero", "año" : 1979 } }, "aficiones" : [ "descansar", "bailar" ], "mascotas" : 2 }
```

19. En la colección **amigos** aumenta todos los años de nacimiento al año 1990 exceptuando aquellos que son superiores a 1990.

```
> db.amigos.updateMany({}, {$max: {"nacimiento.fecha.año": 1990}})
```

```
> db.amigos.updateMany({}, {$max: {"nacimiento.fecha.año": 1990}})
{ "acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 1 }
```

```
> db.amigos.find()
```

```
> db.amigos.find()
{ "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "apellidos" : "San José", "nacimiento" : { "lugar" : "Canarias", "fecha" : { "dia" : 14, "mes" : "Septiembre", "año" : 1990 } }, "aficiones" : [ "programar", "viajar", "beber" ], "mascotas" : 6 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania", "fecha" : { "dia" : 11, "mes" : "Septiembre", "año" : 1996 } }, "aficiones" : [ "descansar", "viajar", "deporte" ], "mascotas" : 1 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "apellidos" : "Moreno", "nacimiento" : { "lugar" : "València", "fecha" : { "dia" : 9, "mes" : "Enero", "año" : 1996 } }, "aficiones" : [ "descansar" ], "mascotas" : 0 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García", "nacimiento" : { "lugar" : "València", "fecha" : { "dia" : 2, "mes" : "Febrero", "año" : 1990 } }, "aficiones" : [ "descansar", "bailar" ], "mascotas" : 2 }
```

20. En la colección **amigos** multiplica el número de mascotas por 2

```
> db.amigos.updateMany({}, {$mul: {mascotas: 2}})
```

```
> db.amigos.updateMany({}, {$mul: {mascotas: 2}})
{ "_acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 3 }
```

```
> db.amigos.find()
```

```
> db.amigos.find()
{ "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "apellidos" : "San José", "nacimiento" : { "lugar" : "Canarias", "fecha" : { "dia" : 14, "mes" : "Septiembre", "año" : 1990 } }, "aficiones" : [ "programar", "viajar", "beber" ], "mascotas" : 12 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania", "fecha" : { "dia" : 11, "mes" : "Septiembre", "año" : 1996 } }, "aficiones" : [ "descansar", "viajar", "deporte" ], "mascotas" : 2 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "apellidos" : "Moreno", "nacimiento" : { "lugar" : "València", "fecha" : { "dia" : 9, "mes" : "Enero", "año" : 1996 } }, "aficiones" : [ "descansar" ], "mascotas" : 0 }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García", "nacimiento" : { "lugar" : "València", "fecha" : { "dia" : 2, "mes" : "Febrero", "año" : 1990 } }, "aficiones" : [ "descansar", "bailar" ], "mascotas" : 4 }
```

21. En la colección **amigos** renombra el campo aficiones por hobbies.

```
> db.amigos.updateMany({}, {$rename: {aficiones: "hobbies"}})
```

```
> db.amigos.updateMany({}, {$rename: {aficiones: "hobbies"}})
{ "_acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 4 }
```

```
> db.amigos.find()
```

```
> db.amigos.find()
{ "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "apellidos" : "San José", "nacimiento" : { "lugar" : "Canarias", "fecha" : { "dia" : 14, "mes" : "Septiembre", "año" : 1990 } }, "mascotas" : 12, "hobbies" : [ "programar", "viajar", "beber" ] }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania", "fecha" : { "dia" : 11, "mes" : "Septiembre", "año" : 1996 } }, "mascotas" : 2, "hobbies" : [ "descansar", "viajar", "deporte" ] }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "apellidos" : "Moreno", "nacimiento" : { "lugar" : "València", "fecha" : { "dia" : 9, "mes" : "Enero", "año" : 1996 } }, "mascotas" : 0, "hobbies" : [ "descansar" ] }
{ "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García", "nacimiento" : { "lugar" : "València", "fecha" : { "dia" : 2, "mes" : "Febrero", "año" : 1990 } }, "mascotas" : 4, "hobbies" : [ "descansar", "bailar" ] }
```

22. En la colección **amigos** añade un hobby nuevo a cada documento, sin modificar los anteriores

Esto podemos hacerlo de dos formas:

- Con **\$addToSet**

```
> db.amigos.updateMany({}, {$addToSet: {hobbies: "Tennis"}})
```

```
> db.amigos.updateMany({}, {$addToSet: {hobbies: "Tennis"}})
{ "acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 4 }
```

- Con **\$push**

```
> db.amigos.updateMany({}, {$push: {hobbies: "Futbol"}})
```

```
> db.amigos.updateMany({}, {$push: {hobbies: "Futbol"}})
{ "acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 4 }
```

```
> db.amigos.find()
```

```
> db.amigos.find()
[
  { "_id" : ObjectId("5c7db3318ba45928dae715f4"), "nombre" : "Martín", "apellidos" : "San José", "nacimiento" : { "lugar" : "Canarias", "fecha" : { "dia" : 14, "mes" : "Septiembre", "año" : 1990 } }, "mascotas" : 12, "hobbies" : [ "programar", "viajar", "beber", "Tennis", "Futbol" ] },
  { "_id" : ObjectId("5c7db43c8ba45928dae715f5"), "nombre" : "Andrés", "apellidos" : "San José", "nacimiento" : { "lugar" : "Ucrania", "fecha" : { "dia" : 11, "mes" : "Septiembre", "año" : 1996 } }, "mascotas" : 2, "hobbies" : [ "descansar", "viajar", "deporte", "Tennis", "Futbol" ] },
  { "_id" : ObjectId("5c7db43c8ba45928dae715f6"), "nombre" : "Juan Carlos", "apellidos" : "Moreno", "nacimiento" : { "lugar" : "València", "fecha" : { "dia" : 9, "mes" : "Enero", "año" : 1996 } }, "mascotas" : 0, "hobbies" : [ "descansar", "Tennis", "Futbol" ] },
  { "_id" : ObjectId("5c7db43c8ba45928dae715f7"), "nombre" : "Alejandro", "apellidos" : "García", "nacimiento" : { "lugar" : "València", "fecha" : { "dia" : 2, "mes" : "Febrero", "año" : 1990 } }, "mascotas" : 4, "hobbies" : [ "descansar", "bailar", "Tennis", "Futbol" ] }
]
```

23. En la colección **amigos** renombra el campo hobbies por aficiones.

```
> db.amigos.updateMany({}, {$rename: {hobbies: "aficiones"}})
```

```
> db.amigos.updateMany({}, {$rename: {hobbies: "aficiones"}})
{ "acknowledged" : true, "matchedCount" : 4, "modifiedCount" : 4 }
```

24. En la colección **amigos** borra el primer documento con más de 4 mascotas

```
> db.amigos.deleteOne({mascotas: {$gte: 4}})
```

```
> db.amigos.deleteOne({mascotas: {$gte: 4}})
{ "_acknowledged" : true, "deletedCount" : 1 }
```

Podemos ver que se ha eliminado, contando el número de documentos de la colección con **count()**

```
> db.amigos.count()
```

```
> db.amigos.count()
3
```

25. En la colección **amigos** borra todos los documentos cuya ciudad de nacimiento sea Ucrania.

```
> db.amigos.deleteMany({"nacimiento.lugar": "Ucrania"})
```

```
> db.amigos.deleteMany({"nacimiento.lugar": "Ucrania"})
{ "_acknowledged" : true, "deletedCount" : 1 }
```

26. Clona la colección **amigos** a **amigosDesechables**.

// Da error - Está deprecado -> \$out se usa en en cambio

```
> db.amigos.copyTo("amigosDesechables")
```

27. Elimina la colección **numeros** de la base de datos

```
> show collections
```

```
[> show collections
amigos
elementos
numeros
```

```
> db.numeros.drop()
```

```
> db.numeros.drop()
true
```

```
> show collections
amigos
elementos
```

28. Crea una base de datos llamada **baseTemporal** y bórrala

```
> use baseTemporal
switched to db baseTemporal
```

Al crear una nueva base de datos sin haberle insertado nada, no se listará como base de datos con el comando **show dbs**. Para poder listarla, deberíamos introducir al menos una colección con algún documento.

Para eliminar una base de datos haremos uso de **dropDatabase()**

```
> db.dropDatabase()
```

```
> db.dropDatabase()
{ "ok" : 1 }
```

Ejercicios de Agregación

1. Sobre la colección **amigos** realiza una proyección, convirtiendo la fecha a un formato DD-MM-AAAA. La nueva variable será **fechaNacimiento**

```
db.amigos.aggregate([
  { $project: {
    _id: 0,
    fechaNacimiento:
    { $concat: [
      {$toString: "$nacimient.fecha.dia"},
      "-",
      {$toString: "$nacimient.fecha.mes"},
      "-",
      {$toString: "$nacimient.fecha.año"}
    ]
  }
}]
})
```

```
> db.amigos.aggregate([
...   { $project: {
...     _id: 0,
...     fechaNacimiento:
...     { $concat: [
...       {$toString: "$nacimient.fecha.dia"},
...       "-",
...       {$toString: "$nacimient.fecha.mes"},
...       "-",
...       {$toString: "$nacimient.fecha.año"}
...     ]
...   }
... ])
{ "fechaNacimiento" : "9-Enero-1996" }
{ "fechaNacimiento" : "2-Febrero-1990" }
```

2. Sobre la colección **amigos** realiza un match y una proyección.

En el match, seleccionaremos los documentos con más de una mascotas y la proyección nos devolverá el nombre, la última afición y el número de mascotas.

```
> db.amigos.aggregate([
  { $match: {mascotas: {$gt: 1}}},
  { $project: {_id: 0, nombre: 1, mascotas: 1, ultimaAficion: {$slice: ["$aficiones",
-1]}}}
])
```

```
> db.amigos.aggregate([
...   { $match: {mascotas: {$gt: 1}}},
...   { $project: {_id: 0, nombre: 1, mascotas: 1, ultimaAficion: {$slice: ["$aficiones", -1]}}}
... ])
{ "nombre" : "Alejandro", "mascotas" : 4, "ultimaAficion" : [ "Futbol" ] }
```

3. Sobre la colección **amigos** realiza una agrupación por el lugar de nacimiento y el número de personas nacidas en cada lugar. Después, ordena por el mayor número de personas nacidas en los lugares.

```
> db.amigos.aggregate([
  { $group: {_id: "$nacimiento.lugar", cuenta: {$sum: 1}}},
  { $sort: { cuenta: -1 }}
])
```

```
> db.amigos.aggregate([
...   { $group: {_id: "$nacimiento.lugar", cuenta: {$sum: 1}}},
...   { $sort: { cuenta: -1 }}
... ])
{ "_id" : "València", "cuenta" : 2 }
```

Dado que actualmente tenemos únicamente dos amigos en la colección y ambos son de València, obtendremos una cuenta de 2. Si dispusieras de más amigos (y de otros lugares), éstos serían contabilizados dentro de cada grupo generado a partir del lugar de nacimiento.

4. Sobre la colección **amigos** realiza una proyección que devuelva el número de aficiones de cada documento

```
> db.amigos.aggregate([
  { $project: { _id: 0, nombre: 1, cantidadAficiones: { $size: "$aficiones" }}}
])
```

```
> db.amigos.aggregate([
...   { $project: { _id: 0, nombre: 1, cantidadAficiones: { $size: "$aficiones" }}}
... ])
{ "nombre" : "Juan Carlos", "cantidadAficiones" : 3 }
{ "nombre" : "Alejandro", "cantidadAficiones" : 4 }
```

5. En la colección **amigos** realiza una consulta que te devuelva un documento por cada ciudad y que sea único, ordenado por el nombre de la ciudad. Debes usar el operador **\$unwind**

```
> db.amigos.aggregate([
  { $group: { _id: "arrayCiudades", nombreCiudad: { $addToSet: "$nacimiento.lugar" }}}},
  { $unwind: "$nombreCiudad" },
  { $sort: { nombreCiudad: 1 }}
])
```

```
> db.amigos.aggregate([
...   { $group: { _id: "arrayCiudades", nombreCiudad: { $addToSet: "$nacimiento.lugar" }}}},
...   { $unwind: "$nombreCiudad" },
...   { $sort: { nombreCiudad: 1 }}
... ])
{ "_id" : "arrayCiudades", "nombreCiudad" : "València" }
```

6. En la colección **amigos** realiza las siguientes operaciones:

- Un match para todos aquellos documentos con año de nacimiento igual o superior a 1990.
- Una proyección para elegir el nombre y crea un campo derivado llamado fechaNacimiento, que convierta a fecha los campos año, mes y día de nacimiento.fecha.
- Una ordenación de manera ascendente por el campo derivado fechaNacimiento.
- Un límite para quedarnos con los 5 primeros.

La conversión a Date requiere de valores válidos para día, mes y año. Por lo que esta agregación fallará si disponemos de los meses de cada documento como cadenas de texto. Para ello, actualizaremos su valor a un valor numérico y así asegurarnos de que podemos realizar la agrupación correctamente.

Por ejemplo, vamos a poner todos los meses a 9:

```
> db.amigos.updateMany({}, {$set: {"nacimiento.fecha.mes": 9}})
```

```
> db.amigos.updateMany({}, {$set: {"nacimiento.fecha.mes": 9}})
{ "_acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }
```

Ahora ya podremos hacer uso de \$toDate para crear la fecha por la cual después ordenar:

```
> db.amigos.aggregate([
  {
    $match: { "nacimiento.fecha.año": {$gte: 1990}}
  },
  {
    $project: { _id: 0, nombre: 1, "nacimiento.fecha.año":1, fechaNacimiento: {
      $toDate:
      {
        $concat:
        [{$toString: "$nacimiento.fecha.año"},
          "-",
          {$toString: "$nacimiento.fecha.mes"},
          "-",
          {$toString: "$nacimiento.fecha.dia"}]
        }
      }
    }
  },
  { $sort: {fechaNacimiento: 1}},
  { $limit: 5 }
])
```

```

> db.amigos.aggregate([
...   {
...     $match: { "nacimiento.fecha.año": {$gte: 1990}}
...   },
...   {
...     $project: { _id: 0, nombre: 1, "nacimiento.fecha.año":1, fechaNacimiento: {
...       $toDate:
...       {
...         $concat:
...         [{$toString: "$nacimiento.fecha.año"},
...          "-",
...          {$toString: "$nacimiento.fecha.mes"},
...          "-",
...          {$toString: "$nacimiento.fecha.día"}]
...       }
...     }
...   }
... },
... { $sort: {fechaNacimiento: 1}},
... { $limit: 5 }
... ])
{ "nombre" : "Alejandro", "nacimiento" : { "fecha" : { "año" : 1990 } }, "fechaNacimiento" : ISODate("1990-09-02T00:00:00Z") }
{ "nombre" : "Juan Carlos", "nacimiento" : { "fecha" : { "año" : 1996 } }, "fechaNacimiento" : ISODate("1996-09-09T00:00:00Z") }

```

7. En la colección **amigos** y sobre el ejercicio 6, almacena los documentos en una colección llamada **top5**.

```

> db.amigos.aggregate([
  {
    $match: { "nacimiento.fecha.año": {$gte: 1990}}
  },
  {
    $project: { _id: 0, nombre: 1, "nacimiento.fecha.año":1, fechaNacimiento: {
      $toDate:
      {
        $concat:
        [{$toString: "$nacimiento.fecha.año"},
         "-",
         {$toString: "$nacimiento.fecha.mes"},
         "-",
         {$toString: "$nacimiento.fecha.día"}]
      }
    }
  }
},
{ $sort: {fechaNacimiento: 1}},
{ $limit: 5 },
{ $out: "top5" }
])

```

Podemos consultar la colección **top5** y ver cómo se han introducido los documentos resultantes de la agregación:

```
> db.top5.find()
{ "_id" : ObjectId("5c7dc8a4d34efa046f6fde28"), "nombre" : "Alejandro", "nacimiento" : { "fecha" : { "año" : 1990 } }, "fechaNacimiento" : ISODate("1990-09-02T00:00:00Z") }
{ "_id" : ObjectId("5c7dc8a4d34efa046f6fde29"), "nombre" : "Juan Carlos", "nacimiento" : { "fecha" : { "año" : 1996 } }, "fechaNacimiento" : ISODate("1996-09-09T00:00:00Z") }
```

8. En la colección **amigos** muestra todas las aficiones ordenadas de mayor a menor ocurrencias. Se busca un resultado similar al que sigue:

Resultado ejemplo:

```
{ "_id" : "viajar", "count" : 10 }
{ "_id" : "beber", "count" : 10 }
{ "_id" : "bailar", "count" : 9 }
{ "_id" : "correr", "count" : 1 }
{ "_id" : "programar", "count" : 1 }
{ "_id" : "descansar", "count" : 1 }
```

Podemos realizar este ejercicio de dos maneras:

```
> db.amigos.aggregate([
  { $unwind: "$aficiones"},
  { $group: {"_id": "$aficiones", count: {$sum: 1}}},
  { $sort: {count: -1}}
])
```

```
> db.amigos.aggregate([
...   { $unwind: "$aficiones"},
...   { $group: {"_id": "$aficiones", count: {$sum: 1}}},
...   { $sort: {count: -1}}
... ])
{ "_id" : "Tennis", "count" : 2 }
{ "_id" : "Futbol", "count" : 2 }
{ "_id" : "descansar", "count" : 2 }
{ "_id" : "bailar", "count" : 1 }
```

O de una manera más sencilla:

```
> db.amigos.aggregate([
  { $unwind: "$aficiones"},
  { $sortByCount: "$aficiones" }
])

> db.amigos.aggregate([
...   { $unwind: "$aficiones"},
...   { $sortByCount: "$aficiones" }
[... ])
{ "_id" : "Tennis", "count" : 2 }
{ "_id" : "Futbol", "count" : 2 }
{ "_id" : "descansar", "count" : 2 }
{ "_id" : "bailar", "count" : 1 }
```

9. Edita la colección **amigos**, añadiendo una edad a cada documento. Sobre esta colección amigos crea un bucket, agrupando por edad y que separe en 3 segmentos.

Además, debe devolver el número de personas que pertenecen al grupo y las ciudades (valores únicos) a las que pertenecen las personas.

```
> db.amigos.updateMany({}, {$set: {edad:Math.floor(Math.random() * 30) + 12}})
```

```
> db.amigos.find().forEach(function(amigo){db.amigos.update({_id:amigo._id}, {$set:
{edad:Math.floor(Math.random() * 30) + 12}})})
```

```
> db.clientes.aggregate([
  { $bucketAuto: {
    groupBy: '$edad',
    buckets: 4,
    output: {
      numPersonas: { $sum: 1},
      ciudades: { $addToSet: "$direccion.ciudad" }
    }
  }
])
```

Ejercicios Diapositivas:

Creación de la colección Frutas:

```
> db.frutas.insertMany([{"fruta": "platano", cantidad: 12, precio: 3.30}, {"fruta": "manzana", cantidad: 12, precio: 2}, {"fruta": "sandia", cantidad: 9, precio: 4}, {"fruta": "manzana", cantidad: 20, precio: 1.50}, {"fruta": "sandia", cantidad: 16, precio: 2.45}, {"fruta": "sandia", cantidad: 13, precio: 3}])
```

```
> db.frutas.insertMany([{"fruta": "platano", cantidad: 12, precio: 3.30}, {"fruta": "manzana", cantidad: 12, precio: 2}, {"fruta": "sandia", cantidad: 9, precio: 4}, {"fruta": "manzana", cantidad: 20, precio: 1.50}, {"fruta": "sandia", cantidad: 16, precio: 2.45}, {"fruta": "sandia", cantidad: 13, precio: 3}])
{
  "acknowledged" : true,
  "insertedIds" : [
    ObjectId("5c7e77e471219a4e31cd048a"),
    ObjectId("5c7e77e471219a4e31cd048b"),
    ObjectId("5c7e77e471219a4e31cd048c"),
    ObjectId("5c7e77e471219a4e31cd048d"),
    ObjectId("5c7e77e471219a4e31cd048e"),
    ObjectId("5c7e77e471219a4e31cd048f")
  ]
}
```

Comprobamos que se han incluido correctamente:

```
> db.frutas.find()
```

```
> db.frutas.find()
{ "_id" : ObjectId("5c7e77e471219a4e31cd048a"), "fruta" : "platano", "cantidad" : 12, "precio" : 3.3 }
{ "_id" : ObjectId("5c7e77e471219a4e31cd048b"), "fruta" : "manzana", "cantidad" : 12, "precio" : 2 }
{ "_id" : ObjectId("5c7e77e471219a4e31cd048c"), "fruta" : "sandia", "cantidad" : 9, "precio" : 4 }
{ "_id" : ObjectId("5c7e77e471219a4e31cd048d"), "fruta" : "manzana", "cantidad" : 20, "precio" : 1.5 }
{ "_id" : ObjectId("5c7e77e471219a4e31cd048e"), "fruta" : "sandia", "cantidad" : 16, "precio" : 2.45 }
{ "_id" : ObjectId("5c7e77e471219a4e31cd048f"), "fruta" : "sandia", "cantidad" : 13, "precio" : 3 }
```

Vamos ahora a **agrupar** las frutas por tipo de fruta y sacar la cantidad total y la media de sus precios. Haremos uso de **\$sum** para **calcular la cantidad total** y **\$avg** para calcular la **media**.

```
> db.frutas.aggregate([{$group: {_id: "$fruta", cantidad: {$sum: "$cantidad"}, precio: {$avg: "$precio"}}}]])
```

```
> db.frutas.aggregate([{$group: {_id: "$fruta", cantidad: {$sum: "$cantidad"}, precio: {$avg: "$precio"}}}]]  
{ "_id" : "sandia", "cantidad" : 38, "precio" : 3.15 }  
{ "_id" : "manzana", "cantidad" : 32, "precio" : 1.75 }  
{ "_id" : "platano", "cantidad" : 12, "precio" : 3.3 }
```

Ahora queremos listar todas las cantidades y precios de las **sandías**, pero que **no se nos muestre** ni el **_id** ni el **tipo de fruta**. Para ello haremos uso de **\$match** y **\$project**:

```
> db.frutas.aggregate([{$match: {fruta: "sandia"}},{$project: {_id:0, cantidad:1, precio:1}}]])
```

```
> db.frutas.aggregate([{$match: {fruta: "sandia"}},{$project: {_id:0, cantidad:1, precio:1}}]]  
{ "cantidad" : 9, "precio" : 4 }  
{ "cantidad" : 16, "precio" : 2.45 }  
{ "cantidad" : 13, "precio" : 3 }
```

A continuación vamos a trabajar con **Map Reduce**

Debemos agregar todas frutas cuyo tipo sea **manzana a una nueva colección** que solo contenga **manzanas**, cada una con su precio único y la cantidad.

```
> db.Frutas.mapReduce(function(){ emit(this.cantidad, this.precio);}, function(key,values){return $values},  
{query:{fruta: "manzana"}, out : "manzanas" })
```

```
> db.frutas.mapReduce(function(){ emit(this.cantidad, this.precio);}, function(key,values){return $values},  
{query:{fruta: "manzana"}, out : "manzanas" })
```

```
> db.frutas.mapReduce(function(){ emit(this.cantidad, this.precio);}, function(key,values){return $values}, {query:{fruta: "manzana"}, out : "manzanas" })  
{  
  "result" : "manzanas",  
  "timeMillis" : 273,  
  "counts" : {  
    "input" : 2,  
    "emit" : 2,  
    "reduce" : 0,  
    "output" : 2  
  },  
  "ok" : 1  
}
```

Se nos habrá creado una **colección llamada manzanas**:

```
> show collections
```

```
> show collections
amigos
elementos
frutas
manzanas
top5
```

Si analizamos esta colección:

```
> db.manzanas.find().pretty()
```

```
> db.manzanas.find().pretty()
{ "_id" : 12, "value" : 2 }
{ "_id" : 20, "value" : 1.5 }
```

Ahora vamos a mostrar todos los tipos de frutas:

```
> db.frutas.distinct("fruta")
```

```
> db.frutas.distinct("fruta")
[ "platano", "manzana", "sandia" ]
```

Ahora queremos conocer todas las frutas que son **manzanas, sandías o plátanos**:

```
> db.frutas.count({fruta: "manzana"})
```

```
> db.frutas.count({fruta: "manzana"})
2
```

```
> db.frutas.count({fruta: "sandia"})
```

```
> db.frutas.count({fruta: "sandia"})
3
```

```
> db.frutas.count({fruta: "platano"})
```



```
> db.frutas.count({fruta: "platano"})
1 _
```