

Sentencias III

Más comandos del find()...

- `db.nombreColeccion.findOne()`
- `db.nombreColeccion.findOneAndUpdate()`
- `db.nombreColeccion.findOneAndReplace()`
- `db.nombreColeccion.findAndModify()`
- `db.nombreColeccion.findOneAndDelete()`

db.collection.findOne ()

Devuelve un solo documento que satisface los criterios de consulta especificados en la colección. Si varios documentos satisfacen la consulta, este método devuelve el primer documento según el orden natural que refleja el orden de los documentos en el disco

Si ningún documento satisface la consulta, el método devuelve un valor nulo.

```
db.collection.findOne( consulta , campos_a_mostrar )
```

db.collection.findOne ().- Ejemplos

```
db.gerentes.findOne ()
```

```
db .gerentes. findOne (  
  {},  
  { nombre : 1 , Svariable : 1 }  
)
```

```
db .gerentes. findOne (  
  {  
    $ or : [  
      { nombre.nombre : "Pepe" },  
      { fechaIngreso : { $ lt : new Date ( '01 / 01/1945 ' ) } }  
    ]  
  }  
)
```

```
db.gerentes.findOne (  
  { contribuciones : 'POO' },  
  { _id : 0 , 'nombre.nombre' : 0 , fechaIngreso : 0 }  
)
```

db.collection.findOneAndUpdate() v3.2

```
db.collection.findOneAndUpdate(  
  <filter>,  
  <update>,  
  {  
    sort: <document>,  
    upsert: <boolean>,  
    returnNewDocument: <boolean>,  
  }  
)
```

Buscará documentos que coincidan con los criterios de búsqueda, y actualizará el primero de ellos, según lo que se especifique en Update

db.collection.findOneAndUpdate() .- Ejemplo

```
db.clasificacion.findOneAndUpdate(  
  { "nombre" : "C. Menendez" },  
  { $inc : { "puntos" : 5 } },  
  { sort : { "puntos" : 1 }, upsert : true }  
)
```

Se ordenan aquellos documentos con el campo “nombre” con valor “C. Menéndez”, ascendentemente según el campo “Puntos”, y se actualiza (con un incremento de cinco unidades), por tanto el documento con menos cantidad de puntos.

Se devuelve el documento PREVIO a la actualización.

db.collection.findOneAndUpdate().- Ejemplo II

```
db.clasificacion.findOneAndUpdate(  
  { "nombre" : "Luco" },  
  { $set: { "nombre" : "Lucio Constantino", "prueba" : 5}, $inc : { "puntos" : 5 } },  
  { sort: { "puntos" : 1 }, upsert:true, returnNewDocument : true }  
)
```

Busca los documentos que coincidan con ese nombre, los ordena ascendentemente por su campo PUNTOS, actualiza según los datos expuestos ("nombre" y "prueba") e incrementa cinco unidades sus puntos.

Si no existiera el documento LO CREA, y devuelve, como fin último de la operación, el nuevo documento

db.collection.findOneAndReplace() v3.2

```
db.collection.findOneAndReplace(  
  <filter>,  
  <replacement>,  
  {  
    projection: <document>,  
    sort: <document>,  
    upsert: <boolean>,  
    returnNewDocument: <boolean>,  
  }  
)
```

Busca un documento (según unos criterios especificados) y lo reemplaza por otro.

No se podrá especificar un campo `_id`, que sea diferente al del documento reemplazado.

No se pueden utilizar operadores de actualización (`$inc`, `$rename`, `$set`, `$unset`, ...)

db.collection.findOneAndReplace().- Ejemplo

```
db.clasificacion.findOneAndReplace(  
  { "calificacion" : { $lt : 100 } },  
  { "nombre" : "Lau Zam", "puntuacion" : 100 })
```

Encontrará el primer documento
con “puntuación” menor que 100
y lo cambiará por lo
especificado en la siguiente
línea.

Se devolverá el documento ORIGINAL que ha sido reemplazado...

```
{ "_id" : 2512, "nombre" : "NurLop", "puntuacion" : 92 }
```

Si, returnNewDocument está configurado como TRUE, también se devolverá el documento reemplazado,

db.collection.findOneAndReplace().- Ejemplo II

```
db.clasificacion.findOneAndReplace(  
  { calificacion : { $lt : 200 } },  
  { nombre : "PaoMay", calificacion : 200 },  
  { sort: { calificacion : 1 },  
    returnNewDocument : 1,  
    project : {nombre:1,calificacion:1,_id:0}  
  }  
)
```

primero ordena todos los documentos por el campo "Calificacion", luego selecciona el primero que tenga un score menor que 200, y lo reemplaza por esos valores...

Devolverá el documento anterior al reemplazado, pero sólo los campos "nombre" y "calificacion"

db.collection.findAndModify() v3.6

```
db.collection.findAndModify({  
  query: <document>,  
  sort: <document>,  
  remove: <boolean>,  
  update: <document>,  
  new: <boolean>,  
  fields: <document>,  
  upsert: <boolean>,  
});
```

- Se modificará el primero de los encontrados.
- Se indica cuales son los criterios para identificar documentos, y si, el resultado de esa búsqueda, requiere alguna ordenación.
- O Remove o Update tiene que especificarse.
- Opcionalmente, se indica si se quiere devolver el nuevo documento (en un Update), o los campos que se quieren devolver, o si se quiere insertar si no existiera coincidencia con los criterios de búsqueda.
- En un Remove, devolverá el documento eliminado

db.collection.findAndModify().- Ejemplo

```
db.alumnos.findAndModify({  
  query: { nombre: "Helena", estado: "activa", calificacion: { $gt: 10 } },  
  sort: { calificacion: 1 },  
  update: { $inc: { puntos: -6 } }  
})
```

Ordena, todos los documentos que cumplan esas tres condiciones, por el campo “calificacion”; luego selecciona el primero y lo decrementa en seis unidades.

db.collection.findAndModify().- Ejemplo II

```
db.alumnos.findAndModify({  
  query: { nombre: "Andrés" },  
  sort: { calificacion: 1 },  
  update: { $mul: { puntos: 2 } },  
  upsert: true,  
  new: true,  
  fields: { nombre: 1, calificacion: 1 }  
})
```

buscará todos aquellos documentos cuyo valor para el campo “nombre”, coincida con Andrés, los ordenará ascendentemente por su valor de campo “Rating”, actualizará el documento incrementando el campo “score” por el doble de su valor actual; si no existiera ningún documento con “nombre = Andrés”, lo crearía, y en todos los casos devolvería el nombre y el rating del documento actualizado

diferencias

`findOneAndUpdate()`: Buscará documentos que coincidan con los criterios de búsqueda, y actualizará el primero de ellos, según lo que se especifique en Update

`findOneAndReplace()`: Busca un documento (según unos criterios especificados) y lo reemplaza por otro. No se podrán utilizar operadores de Actualización (`$mul`, `$inc`, `$min`, `$set`, `$rename`,...)

```
db.trabajadores.updateMany({},{$rename:{'apodo':'alias','teléfono móvil':'Móvil'}})
```

`findAndModify()`: Modifica el primero de los encontrados tanto para update como para remove.

db.collection.findOneAndDelete()

```
db.collection.findOneAndDelete(  
  <filter>,  
  {  
    projection: <document>,  
    sort: <document>,  
  }  
)
```

```
db.puntuaciones.findOneAndDelete(  
  { "nombre" : "G.Banderas" },  
  {projection:{nombre:1, puntos:1}}  
)
```

Se devolverá el documento que ha sido borrado...

```
{ _id: 71, nombre: "G.Banderas", "puntos" : 8 }
```

db.collection.findOneAndDelete()

La diferencia con `db.collection.deleteOne()` → se borra directamente por coincidencia de un campo. Y sólo se muestra que la operación se realizó con éxito: `{ "acknowledged" : true, "deletedCount" : 1 }`

Y en este puedes ordenar (por si coincidieran varios) y eliminar el primero. Y además puedes elegir qué campos del documento eliminado mostrar para informar al usuario.