

MongoDB — HOWTO-2

Inserción de documentos (insert, insertOne, insertMany)		
Operación	Comando	Ejemplo
Insertar un nuevo documento en una colección	<pre>db.collection.insert(<documento>) o db.collection.insertOne(<documento>)</pre>	<pre>db.libros.insert({ isbn:"97888-227-5", titulo:"Don Quijote", autor:"Miguel de Cervantes", stock:10, pvp:25.00 })</pre>
Insertar varios documentos nuevos en una colección	<pre>db.collection.insert([<documento1>, <documento2>, ...]) o db.collection.insertMany([<documento1>, <documento2>, ...])</pre>	<pre>db.libros.insert([{ isbn:"97888-227-5", titulo:"Don Quijote", autor:"Miguel de Cervantes", stock:10, pvp:25.00 }, { isbn:"73858-451-3", titulo:"La Celestina", autor:"Fernando de Rojas", stock:5, pvp:25.00 }])</pre>

MongoDB-HOWTO-2

Actualización de documentos: reemplazar documento completo (update)

Operación	Comando	Ejemplo
Reemplazar un documento de la colección	<pre>db.collection.update(<condición>, <update>)</pre>	<pre>db.libros.update({ _id:ObjectId("5a31...33a76e") }, { isbn:"97888-227-5", titulo:"Don Quijote", autor:"Miguel de Cervantes", stock:5, pvp:85.00 })</pre>

Actualización de documentos: modificar o añadir campos (update, updateMany + \$set)

Operación	Comando	Ejemplo
Actualizar o añadir un campo en el primer documento de una colección	<pre>db.collection.update({}, <update>)</pre>	<pre>db.libros.update({}, { \$set:{estante:""} })</pre>
Actualizar o añadir un campo en todos los documentos de una colección	<pre>db.collection.update({}, <update>, { multi:1 }) o db.collection.updateMany({}, <update>)</pre>	<pre>db.libros.update({}, { \$set:{estante:""} }, { multi:1 }) o db.libros.updateMany({}, { \$set:{estante:""} })</pre>

MongoDB-HOWTO-2

Actualizar o añadir un campo en el primer documento de la colección que cumpla unas condiciones	<pre>db.collection.update(<condición>, <update>)</pre>	<pre>db.libros.update({ categoria:"novela" }, { \$set:{estante:"3"} })</pre>
Actualizar o añadir un campo en todos los documentos de la colección que cumplan unas condiciones	<pre>db.collection.update(<condición>, <update>, { multi:1 }) o db.collection.updateMany(<condición>, <update>)</pre>	<pre>db.libros.update({ categoria:"novela" }, { \$set:{estante:"3"} }, { multi:1 }) o db.libros.updateMany({ categoria:"novela" }, { \$set:{estante:"3"} })</pre>

Actualización documentos: eliminar campos (update, updateMany + \$unset)		
Operación	Comando	Ejemplo
Eliminar un campo del primer documento de una colección	<pre>db.collection.update({}, <update>)</pre>	<pre>db.libros.update({}, { \$unset:{estante:""} })</pre>
Eliminar un campo en todos los documentos de una colección	<pre>db.collection.update({}, <update>, { multi:1 }) o db.collection.updateMany({}, <update>)</pre>	<pre>db.libros.update({}, { \$unset:{estante:""} }, { multi:1 }) o db.libros.updateMany({}, { \$unset:{estante:""} })</pre>

MongoDB-HOWTO-2

Eliminar un campo del primer documento de la colección que cumpla unas condiciones	<pre>db.collection.update(<condición>, <update>)</pre>	<pre>db.libros.update({ categoria:"novela" }, { \$unset:{estante:""} })</pre>
Eliminar un campo en todos los documentos de la colección que cumplan unas condiciones	<pre>db.collection.update(<condición>, <update>, { multi:1 }) o db.collection.updateMany(<condición>, <update>)</pre>	<pre>db.libros.update({ categoria:"novela" }, { \$unset:{estante:""} }, { multi:1 }) o db.libros.updateMany({ categoria:"novela" }, { \$unset:{estante:""} })</pre>

Búsqueda de documentos (find, findOne, pretty)		
Operación	Comando	Ejemplo
Mostrar todos los documentos de una colección	<code>db.collection.find()</code>	<code>db.libros.find()</code>
Mostrar documentos de una colección "en bonito"	<code>db.collection.find().pretty()</code>	<code>db.libros.find().pretty()</code>
Mostrar los documentos filtrados según una condición	<code>db.collection.find(<condición>)</code>	<code>db.libros.find({categoria:"novela"})</code>
Mostrar solo algunos campos de los documentos filtrados según una condición	<pre>db.collection.find(<condición>, <lista>)</pre>	<pre>db.libros.find({}, { titulo:1, pvp:1 }) ; db.libros.find({ isbn:'97888-227-5' }, { titulo:1, stock:1 })</pre>

MongoDB-HOWTO-2

Mostrar el primer documento de una colección filtrada según una condición	<pre>db.collection.findOne(<condición>, <lista>)</pre>	<pre>db.libros.findOne({}, { titulo:1, pvp:1 }) ; db.libros.findOne({ isbn:'97888-227-5' }, { titulo:1, stock:1 })</pre>
Mostrar documentos de una colección ocultando el campo _id	<pre>db.collection.find(<condición>, { _id:0 })</pre>	<pre>db.libros.find({}, { _id:0 })</pre>
Mostrar documentos de una colección ocultando algunos campos	<pre>db.collection.find(<condición>, <lista>)</pre>	<pre>db.libros.find({}, { stock:0, _id:0 })</pre>
Mostrar los valores distintos de un campo en una colección	<pre>db.collection.distinct(campo)</pre>	<pre>db.libros.distinct(categoria)</pre>

Modificadores en operaciones de búsqueda de documentos (limit, skip, sort)		
Operación	Comando	Ejemplo
Mostrar un número máximo N de documentos	<pre>db.collection.find(<condición>, <lista>).limit(<N>)</pre>	<pre>db.libros.find().limit(5)</pre>
Saltar los primeros N documentos	<pre>db.collection.find(<condición>, <lista>).skip(<N>)</pre>	<pre>db.libros.find().skip(100)</pre>
Ordenar los resultados	<pre>db.collection.find(<condición>, <lista>).sort(<orden>)</pre>	<pre>db.libros.find().sort({ categoria:1, pvp:-1 })</pre>

MongoDB-HOWTO-2

Conteo de documentos (count)		
Operación	Comando	Ejemplo
Contar todos los documentos de una colección	<code>db.collection.find().count()</code> o <code>db.collection.count()</code>	<code>db.libros.find().count()</code> o <code>db.libros.count()</code>
Contar todos los documentos de una colección que cumplan con una condición	<code>db.collection.find(<condición>) .count()</code>	<code>db.libros.find({categoria:"novela"}) .count()</code>

Operaciones de borrado de documentos (remove, deleteOne, deleteMany)		
Operación	Comando	Ejemplo
Borrar todos los documentos de una colección	<code>db.collection.remove({})</code>	<code>db.libros.remove({})</code>
Borrar el documento de una colección con un _id determinado	<code>db.collection.remove({ _id:ObjectId("<_id>") })</code>	<code>db.libros.remove({ _id:ObjectId("5ac331...33a76e")})</code>
Borrar todos los documentos de una colección que cumplan una condición	<code>db.collection.remove(<condición>)</code> o <code>db.collection.deleteMany(<condición>)</code>	<code>db.libros.remove({ categoria:"novela" })</code> o <code>db.libros.deleteMany({ categoria:"novela" })</code>
Borrar el primer documento de una colección que cumpla una condición	<code>db.libros.remove(<condición>, { justOne:1 })</code> o <code>db.libros.deleteOne(<condición>)</code>	<code>db.libros.remove({ categoria:"novela" }, { justOne:1 })</code> o <code>db.libros.deleteOne({ categoria:"novela" })</code>

MongoDB-HOWTO-2

Operadores de comparación(\$eq, \$ne, \$lt, \$lte, \$gt, \$gte, \$in, \$nin)		
Operación	Comando	Ejemplo
Igual a	{<campo>: { \$eq: <valor> }}	db.libros.find({ pvp: {\$eq: 15.17} })
Distinto de	{<campo>: { \$ne: <valor> }}	db.libros.find({ pvp: {\$ne: 20.08} })
Menor que	{<campo>: { \$lt: <valor> }}	db.libros.find({ pvp: {\$lt: 18} })
Menor o igual que	{<campo>: { \$lte: <valor> }}	db.libros.find({ pvp: {\$lte: 13.5} })
Mayor que	{<campo>: { \$gt: <valor> }}	db.libros.find({ pvp: {\$gt: 20} })
Mayor o igual que	{<campo>: { \$gte: <valor> }}	db.libros.find({ pvp: {\$gte: 19.5} })
Valor en la lista	{<campo>: { \$in: [<valor1>, <valor2>, ...] }}	db.libros.find({ pvp: {\$in: [14, 15]}})
Valor no en la lista	{<campo>: { \$nin: [<valor1>, <valor2>, ...] }}	db.libros.find({ pvp: {\$nin: [14, 15]} })

Operadores de elementos (\$exists, \$type)		
Operación	Comando	Ejemplo
Buscar los documentos de una colección que contienen un campo determinado	{<campo>: {\$exists: true}}	db.libros.find({ pasillo: {\$exists: true} })
Buscar los documentos de una colección que NO contienen un campo determinado	{<campo>: {\$exists: false}}	db.libros.find({ pasillo: {\$exists: false} })
Buscar los documentos de una colección que contienen un campo cuyo valor del dato es un tipo de datos BSON especificado	{<campo>: {\$type: valor}}	db.libros.find({ estante: {\$type: 2} })

Operadores lógicos (OR, AND, NOT y NOR)		
Operación	Comando	Ejemplo
OR	<pre>{ \$or: [<expresión1>, <expresión2>, ...] }</pre>	<pre>db.libros.find({ \$or: [{pvp: {\$lte: 15}}, {pvp: {\$eq: 23}}] })</pre>
AND	<pre>{ \$and: [<expresión1>, <expresión2>, ...] }</pre>	<pre>db.libros.find({ \$and: [{ pvp: {\$eq: 15} }, { almacen: {\$eq: "A34"} }] })</pre>
NOT	<pre>{ \$not: {<expresión>} }</pre>	<pre>db.libros.find({ \$not: { pvp: {\$eq: 23} } })</pre>
NOR	<pre>{ \$nor: [<expresión1>, <expresión2>, ...] }</pre>	<pre>db.libros.find({ \$nor: [{ pvp: {\$lte: 15} }, { pvp: {\$eq: 23} }] })</pre>

MongoDB-HOWTO-2

Otras operaciones (load, explain)		
Operación	Comando	Ejemplo
Ejecuta un fichero JavaScript	<code>load(<fichero>)</code>	<code>load(myscript.js)</code>
Mostrar el plan de ejecución	<code>db.collection.explain().<método>()</code>	<code>db.libros.explain().find().pretty()</code>

Tipos de datos		
Nombre	Código	
Double		1
String		2
Object		3
Array		4
Binary data		5
Object id		7
Boolean		8
Date		9
Null		10
Regular expression		11
JavaScript code		13
Symbol		14
JavaScript code with scope		15
32-bit integer		16
Timestamp		17
64-bit integer		18
Min key		255
Max key		127