**GUIA COMO SACAR RESPALDOS**

Para crear una copia de seguridad se utiliza mongodump, donde genera una carpeta con los archivos

Para crear una copia de seguridad de una base de datos específica, debe usar la opción -d y especificar el nombre de la base de datos. Además, para que mongodump sepa dónde almacenar la copia de seguridad, debe usar la opción -o y especificar una ruta. También utilizamos el método de autenticación como administrador para obtener los permisos de hacer el respaldo utilizando el comando –authenticacionDatabase

|  |
| --- |
| mongodump -d my\_mongo\_db -o /data/db/backup/ --authenticationDatabase "basemongo" -u "jaky" -p "pacomongo16" |

**"CHEAT SHEET" PARA LAS FUNCIONES**

**MongoDB Cheat Sheet**

**Show All Databases**

show dbs

**Show Current Database**

db

**Create Or Switch Database**

use acme

**Drop**

db.dropDatabase()

**Create Collection**

db.createCollection('posts')

**Show Collections**

show collections

**Insert Row**

db.posts.insert({

title: 'Post One',

body: 'Body of post one',

category: 'News',

tags: ['news', 'events'],

user: {

name: 'John Doe',

status: 'author'

},

date: Date()

})

**Insert Multiple Rows**

db.posts.insertMany([

{

title: 'Post Two',

body: 'Body of post two',

category: 'Technology',

date: Date()

},

{

title: 'Post Three',

body: 'Body of post three',

category: 'News',

date: Date()

},

{

title: 'Post Four',

body: 'Body of post three',

category: 'Entertainment',

date: Date()

}

])

**Get All Rows**

db.posts.find()

**Get All Rows Formatted**

db.find().pretty()

**Find Rows**

db.posts.find({ category: 'News' })

**Sort Rows**

# asc

db.posts.find().sort({ title: 1 }).pretty()

# desc

db.posts.find().sort({ title: -1 }).pretty()

**Count Rows**

db.posts.find().count()

db.posts.find({ category: 'news' }).count()

**Limit Rows**

db.posts.find().limit(2).pretty()

**Chaining**

db.posts.find().limit(2).sort({ title: 1 }).pretty()

**Foreach**

db.posts.find().forEach(function(doc) {

print("Blog Post: " + doc.title)

})

**Find One Row**

db.posts.findOne({ category: 'News' })

**Find Specific Fields**

db.posts.find({ title: 'Post One' }, {

title: 1,

author: 1

})

**Update Row**

db.posts.update({ title: 'Post Two' },

{

title: 'Post Two',

body: 'New body for post 2',

date: Date()

},

{

upsert: true

})

**Update Specific Field**

db.posts.update({ title: 'Post Two' },

{

$set: {

body: 'Body for post 2',

category: 'Technology'

}

})

**Increment Field ($inc)**

db.posts.update({ title: 'Post Two' },

{

$inc: {

likes: 5

}

})

**Rename Field**

db.posts.update({ title: 'Post Two' },

{

$rename: {

likes: 'views'

}

})

**Delete Row**

db.posts.remove({ title: 'Post Four' })

**Sub-Documents**

db.posts.update({ title: 'Post One' },

{

$set: {

comments: [

{

body: 'Comment One',

user: 'Mary Williams',

date: Date()

},

{

body: 'Comment Two',

user: 'Harry White',

date: Date()

}

]

}

})

**Find By Element in Array ($elemMatch)**

db.posts.find({

comments: {

$elemMatch: {

user: 'Mary Williams'

}

}

}

)

**Add Index**

db.posts.createIndex({ title: 'text' })

**Text Search**

db.posts.find({

$text: {

$search: "\"Post O\""

}

})

**Greater & Less Than**

db.posts.find({ views: { $gt: 2 } })

db.posts.find({ views: { $gte: 7 } })

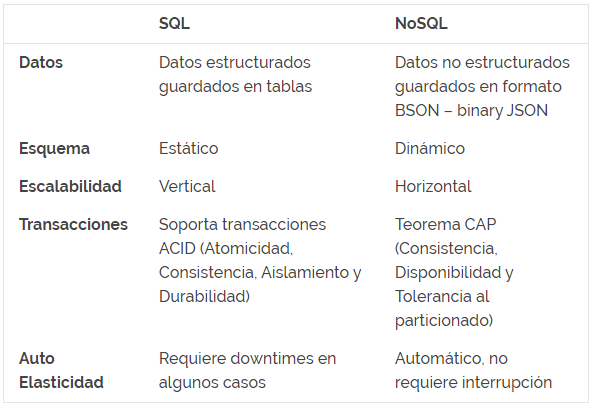
db.posts.find({ views: { $lt: 7 } })

db.posts.find({ views: { $lte: 7 } })

**SQL vs mongoDB**

Antes de seguir debemos aclarar algunos términos de MongoDB para hablar en el mismo lenguaje: [1]

|  |  |
| --- | --- |
| Documentos | Son la unidad básica de MongoDB. Son los registros dentro de las colecciones donde guardamos nuestra información con el formato BSON, una representación binaria y con superpoderes de JSON (Binary JSON). Gracias a este formato tenemos mucha flexibilidad en los tipos de datos que podemos almacenar. |
| Colecciones | Son agrupaciones de documentos. Son equivalentes a las tablas en bases de datos relacionales, pero NO nos imponen un esquema o estructura rígida para guardar nuestra información y tienen muy buena performance. |
| Bases de Datos | Son los contenedores físicos para nuestras colecciones. Cada base de datos tiene un archivo propio en el sistema de archivos de nuestra computadora o servidor |
| Clusters | Cada cluster puede tener múltiples bases de datos. MongoDB es una base de datos distribuida, lo que nos permite tener una gran escalabilidad horizontal. Esto significa que no solo podemos aumentar la capacidad de nuestros servidores (memoria, espacio, núcleos, etc), también podemos aumentar la cantidad de servidores para distribuir la carga de nuestras bases de datos. |



**Referencias**

[1] «MongoDB vs SQL: Di no a la rebeldía. Usa MongoDB con responsabilidad.», *Platzi*. https://platzi.com/blog/mongo-vs-sql/ (accedido jul. 01, 2020).