

Introducción

MongoDB es un sistema de gestión de bases de datos (DBMS) de código abierto que utiliza un modelo de base de datos orientado a documentos que admite numerosos tipos de datos. En lugar de usar tablas y filas como en las bases de datos relacionales, la arquitectura MongoDB está formada por colecciones y documentos. MongoDB almacena datos en documentos similares a JSON que pueden variar en estructura.



# ¿Cuáles son los principales objetivos de este proyecto?

* Comprender qué es una base de datos no relacional.
* Comprender cómo se realizan las consultas a una base de datos no relacional.
* Analizar y diseñar bases de datos con sus correspondientes colecciones.
* Comprender cómo interactuar con los datos almacenados en la base de datos.

1.1. Desarrollar el Proyecto

A continuación, creará su propia base de datos de red social con las siguientes colecciones:

* Users
  + db.createCollection("users")
* Posts
  + db.createCollection("posts")

1.2. Ejecute las siguientes consulta

A continuación tendrás que realizar las siguientes consultas MongoDB:

#### 1.2.1 INSERTAR DATOS

* Insertar al menos 15 publicaciones nuevas con almenos 2 comentarios por publicación:
  + title
  + body
  + username
  + comments
    - username
    - body

db.posts.insertOne({title:"post one", body:"this is the first post", username:"Jane Doe", comments:[{username:"Jane Doe", body:"this is the first comment"}]})

db.posts.insertMany([{title:'post two', body:'this is the second post', username:'Lidya', comments:[{username:'Lidya', body:'this issocialNetwork> db.posts.insertMany(

title:'post 3',

... [

... {

... title:'post two',

... body:'this is the second post',

... username:'Lidya',

... comments:[

... {

... username:'Lidya',

... body:'this is the second comment'

... }

... ]

... },

... {

... body:'this is the 3 post',

... username:'Joseph',

... comments:[

... {

... username:'Joseph',

... body:'this is the 3 comment'

... }

... ]

... },

... {

... title:'post 4',

... body:'this is the 4 post',

... username:'Mary',

... comments:[

... {

... username:'Mary',

... body:'this is the 4 comment'

... }

... ]

... },

... {

... title:'post 5',

... body:'this is the 5 post',

... username:'Carl',

... comments:[

... {

... username:'Carl',

... body:'this is the 5 comment'

... }

... ]

... },

... {

... title:'post 6',

... body:'this is the 6 post',

... username:'Mya',

... comments:[

... {

... username:'Mya',

... body:'this is the 6 comment'

... }

... ]

... },

... {

... title:'post 7',

... body:'this is the 7 post',

... username:'James',

... comments:[

... {

... username:'James',

... body:'this is the 7 comment'

... }

... ]

... },

... {

... title:'post 8',

... body:'this is the 8 post',

... username:'Julia',

... comments:[

... {

... username:'Julia',

... body:'this is the 8 comment'

... }

... ]

... },

... {

... title:'post 9',

... body:'this is the 9 post',

... username:'David',

... comments:[

... {

... username:'David',

... body:'this is the 9 comment'

... }

... ]

... },

... {

... title:'post 10',

... body:'this is the 10 post',

... username:'Lily',

... comments:[

... {

... username:'Lily',

... body:'this is the 10 comment'

... }

... ]

... },

...

... ]

... )

* Insertar al menos 10 nuevos usuarios:
  + username
  + email
  + age

db.users.insertMany(

... [

... {

... username: 'Mara',

... email: 'Mara@example.com',

... age: 52,

... },

... {

... username: 'Marc',

... email: 'marc@example.com',

... age: 74,

... }

... ]

... )

#### 1.2.2 ACTUALIZAR DATOS

* Actualizar publicaciones:
  + Actualiza todos los campos de una publicación

db.posts.updateMany({title: 'post one'},{$set:{title:'post 1', body:'this is the 1st post', username:'Jane', comments: [{ username: 'Jane', body: 'this is the 1st comment'}]}})

* + Cambiar el body de una publicación.

db.posts.updateOne({title: 'post 1'},{$set:{body:'updated'}})

* + Actualizar comentarios:
    - Actualiza el comentario de una publicación.

db.posts.updateOne({title:'post 15'},{$set:{comments: [ {username: 'Carl', body: 'updated comment'}]}})

db.posts.updateOne({title:'post 15'},{$set:{comments: [ {username: 'Carl', body: 'updated comment' },{username: 'Carl', body: 'updated comment 2' }]}})

Actualizar usuarios:

* + Actualiza todos los campos de un usuario

db.users.updateMany({username:'Lily'},{$set:{username:'Lilo', email:'lilo@example.com', age: 10}})

* + Cambiar el email de dos usuarios es decir hacer la query dos veces.

db.users.updateOne({username:'Marc'},{$set:{email:'marc@marc.com'}})

db.users.updateOne({username:'Mara'},{$set:{email:'mara@mara.com'}})

* + Aumenta en 5 años la edad de un usuario

db.users.updateOne({username:'Mara'}, {$inc:{age:5}})

#### 1.2.3 OBTENER DATOS

* Seleccionar todas las publicaciones
  + db.posts.find()
* Selecciona las publicaciones que coincidan con el username indicado
  + db.posts.findOne({username:'Carl'})
* Seleccione todos los usuarios con una edad mayor a 20
  + db.users.find({age:{$gt:20}})
* Seleccione todos los usuarios con una edad inferior a 23
  + db.users.find({age:{$lt:23}})
* Seleccione todos los usuarios que tengan una edad entre 25 y 30 años
  + db.users.find({ age: { $gte: 25, $lte: 30 } })
  + db.users.find({ $and:[{ age: { $gte: 25}}, {age: { $lte: 30 } }]})
* Muestra los usuarios de edad menor a mayor y viceversa
  + db.users.find().sort({age:1})
  + db.users.find().sort({age:-1})
* Seleccione el número total de usuarios
  + db.users.find().count()
* Seleccione todas las publicaciones y haz que se muestren con la siguiente estructura: Título de la publicación: "title one"

db.posts.find().forEach((doc) => {print('Post Title: ' + doc.title)})

* Selecciona solo 2 usuarios

db.users.find().limit(2)

* Busca por title 2 publicaciones

db.posts.find({ title: { $in: ['post 5', 'post 7'] } })

#### 

#### 1.2.4 BORRAR DATOS

* Elimina a todos los usuarios con una edad mayor a 30

db.users.deleteMany({age:{$gt:30}})

1.3 Extra

* Seleccione el número total de publicaciones que tienen más de un comentario

db.posts.find({ "comments.1": { $exists: true } }).count()

* Seleccione la última publicación creada

db.posts.find().sort({$natural:-1}).limit(1)

* Selecciona 5 publicaciones y que sean las últimas creadas

db.posts.find().sort({$natural:-1}).limit(5)

* Elimina todas las publicaciones que tengan más de un comentario

db.posts.deleteMany({ "comments.1": { $exists: true } })

# 1.4. Entregables

* Crear un documento (Word, Readme, etc.) con las consultas que se piden en el ejercicio
* Dejar el enlace o adjuntar el archivo en Classroom