



Marco Fernandez <marcofer.fernandez@gmail.com>

clase 5/5/2019

1 message

manuel canon <manuelhcama1@gmail.com>

Thu, Jun 6, 2019 at 12:35 AM

To: fqe2309@gmail.com, leoquintero7@gmail.com, marcofer.fernandez@gmail.com, jhonatan.pregonero@gmail.com

mongo db
base de datos no relacional que entiende js
basada en c++
su formato es netamente
sintaxis json se guarda todo en BSON (Binary Json)

es una base de datos no relacional
-- aquella que en su estructura no procesa ni tablas ni filas ni columnas
como le hace una BD relacional ej: MySQL. SQL, oracle, postgresQL

al no tener filas de datos, mongo maneja documentos

comparacion
BD relacional mysql excel
BD no relacional mongoDB word

mongodb reemplaza el demolo entidad relacion mer por medio de designacion de json
a esto se le conoce como sub subdocumento

el ID
en MONGO, el id se genera automaticamente, este id generado contiene 24 caracteres
12 byts
timespam
proceso unico de la maquina
hash procedimiento hash descripcion de procesos
numero autouncrementable en el rango alfanumerico que promueve mong

quien usa mongo y porque

ebay, googley, usuario

utilizan por la extencion de los documentos y las colecciones.

//verificar mongo debe tener 2 terminales.
waiting for connections on port 2770
cliente de mongo

//ya dentro de mongo
show data bases

use thorPM
switched to db thorDB

documento = (nombre: "pepe", edad:16)

db.curso.insert(documento)

mongo writeResult("Inserted":1)

db.curso.find()

//el id de 24 caracteres o los 12 bites.



```
db.curso.find().pretty()
```

```
"id"  
"nombre"  
"edad"
```

```
db.curso.insert(dato)  
writeResult({})
```

```
uno = {nombre:"juan"}  
dos = {nombre:"pepe"}  
tres = {nombre:"laura"}
```

```
db.curso.insert([uno,dos,tres])
```

```
db.curso.find({edad:{$gt:10}{})
```

```
//cursores de busqueda $gt greater than / mayor que  
// $gte greater than equals / mayor o igual que  
// $lt /menor que  
// $lte / menor o igual que
```

operadores logicos

```
db.curso.find({id:0, nombre:1{}
```

```
db.curso.find( (edad:{$ne:20}) )
```

```
db.curso.update({nombre:"clase"},{$set:{nombre:"fullstack"}})  
writeResult({"nMatched":1,"nUoserted":0,"nModified":1{}
```

```
db.curso.update({nombre:"clase"},{$set:{nombre:"fullstack"}{,(multi:true))  
writeResult({"nMatched":2,"nUoserted":0,"nModified":2{}
```

```
documento = {nombre:"prueba"}  
{"nombre":"prueba"}
```

```
db.ejemplo.insert(documento)  
writeResult({"nInserted":1{}
```

```
db.ejemplo.find()  
("_id":ObjectId("Sckjsakjhsjdajaskaksadh"),"nombre":"prueba")
```

```
db.curso.update( {nombre:"Fullstack"}, {$set:(instituto:"bictia")}, (multi:true))  
writeResult({"nMatched":3,"nUoserted":0,"nModified":3{}
```

```
db.curso.remove((nombre:"pepe"))  
WriteResult(("nRemoved":1))
```

```
uno = {nombre : "fabian",nota: 4}
```

```
dos = {nombre : "mateo",nota: 3}
```

```
tres = {nombre : "fabian",nota: 4}
```

```
cuatro = {nombre : "juan", nota: 3}
```

```
db.nota.insert([uno,dos,tres,cuatro])
```



db.notas.find()

