Bases de datos NOSQL – Fdi - UCM

# Convocatoria extraordinaria 2018-19

Para cargar los datos,

- arrancar el servidor de MongoDB

- bajarse el fichero log.json

- desde otro terminal Linux, teclear

mongoimport --collection log --drop --file log.json

La colección simula registros de log de accesos de diferentes usuarios a un sistema. Cada documento representa uno de estos accesos, e incluye como \_id el nombre del usuario. También se incluye el país de acceso, la IP o IPs utilizadas, y un array *actions* con las acciones realizadas durante la sesión.

Queremos escribir consultas Mongo para conocer:

1. Número de documentos que incluyan dentro de su clave *actions* un *action* de tipo login pero no uno de tipo *consult*.

Solución

> db.log.count({'$and':[{"actions.action":"login"},{"actions.action":{"$ne":"consult"}}]})

1

> db.log.count({"actions.action":{'$in':["login"],'$nin':["consult"]}})

1

> db.log.count({"actions.action":{'$eq':"login",'$ne':"consult"}})

1

1. Para cada valor diferente de la clave *action*, mostrar dicho valor y el número de documentos que lo contienen (si un documento contiene dos veces el mismo valor para *action* contará dos veces), ordenado de mayor a menor. Salida esperada:

{ "\_id" : "login", "count" : 5 }

{ "\_id" : "consult", "count" : 4 }

{ "\_id" : "edit", "count" : 2 }

{ "\_id" : "delete", "count" : 1 }

{ "\_id" : "delete file", "count" : 1 }

**Solución**

db.log.aggregate(

[

{$unwind:"$actions"},

{$sortByCount:"$actions.action"}

])

1. Documentos de log en los que el array actions contenga al menos 3 elementos, y para cada uno de ellos mostrar solo el \_id y la 3ª opinión

**Solución**

db.log.find({"actions.2":{'$exists':true}},{"\_id":1,"actions":{'$slice':[2,1]}})

{ "\_id" : "Sinda", "actions" : [ { "action" : "delete", "timestamp" : "2018-05-24 15:03:21", "duration" : 8, "success" : false, "error" : "access denied" } ] }

{ "\_id" : "Herminia", "actions" : [ { "action" : "consult", "timestamp" : "2018-05-24 12:22:12" } ] }

---

1. Para cada documento de la colección log con \_id ‘Herminia’, mostrar:

* La clave IP
* El valor de la clave ‘action’ para la primera entrada del array ‘actions’ que tenga un valor ‘duration’ mayor de 20 segundos

Salida esperada:

{ "IP" : "2.18.111.255", "actions" : [ { "action" : "delete file", "timestamp" : "2018-05-24 12:19:13", "duration" : 54, "success" : false } ] }

**Solución**

db.log.find({'\_id':"Herminia"},

{'IP$0':1, '\_id':0, 'actions':{'$elemMatch':{'duration':{'$gt':20}}}})

1. Llega un nuevo acción de la usuario Sinda (se puede suponer que \_id: Sinda ya existe),. Se trata de una 'action' con valor 'copy file', timestamp: "2018-05-24 15:03:21",duration:8,success: true. Añadirla al principio de su array acciones ('actions')

Solucion

db.log.update( { '\_id': 'Sinda' },

{ '$push': { 'actions': {

'$each': [{'action': ' copy file ',

'timestamp': "2018-05-24 15:03:21",

'duration':8,'success': true}],

'$position': 0

}

}

}

)