|  |
| --- |
|  |
| **INFORME LAB 13** |
| Big Data |

|  |
| --- |
| Integrantes: Jorge González – Cristian Álvarez  15-6-2022 |

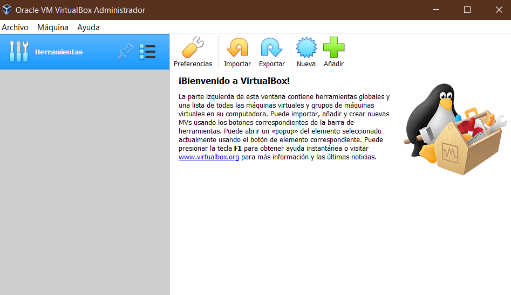
# Introduccion

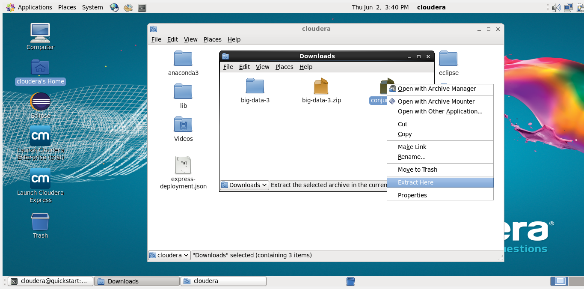
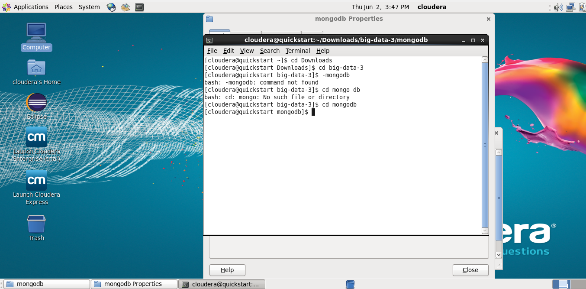
En este laboratorio, se nos solicita realizar una secuencia de actividades relacionadas con MongoDB utilizando una MV que se nos entrega en este laboratorio y entregar evidencia de haberlos realizados, además de realizar una investigación por nuestra parte para extraer información específica de la base de datos en la que estamos trabajando. Para finalizar el laboratorio debemos responder unas preguntas marcando la alternativa correcta.

# Parte1 del Lab (Operaciones MongoDB)

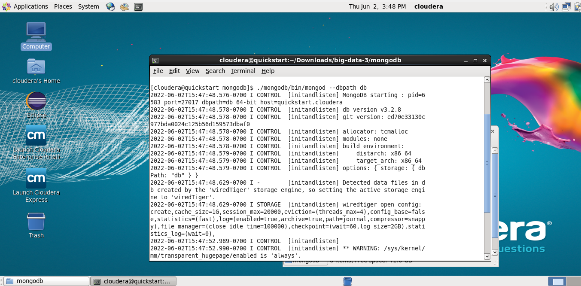
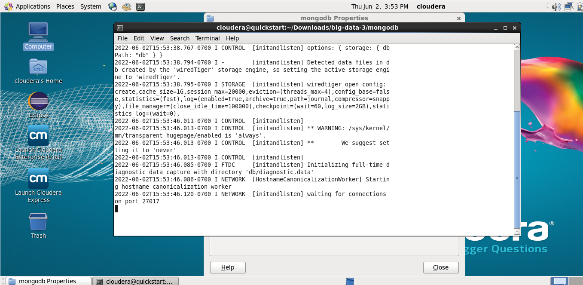
Rapuzar una secuencia de operaciones en MongoDB, utilizando la MV entregada en este laboratorio.

***Paso 1.*** Utilizaremos Oracle VM VirtualBox para la MV entregada, donde agregamos los parámetros que se piden e iniciamos la MV, seguido de ello hacemos los preparativos para iniciar el servidor MongoDB y a través del terminal cambiamos el directorio del mongoDB.

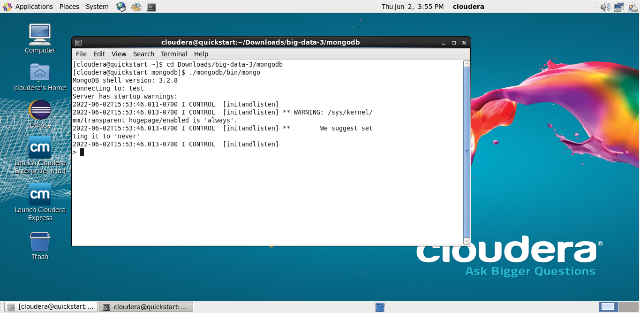
 

Comando --dbpath db

Ejecutamos cd Downloads/big-data-3/mongodb



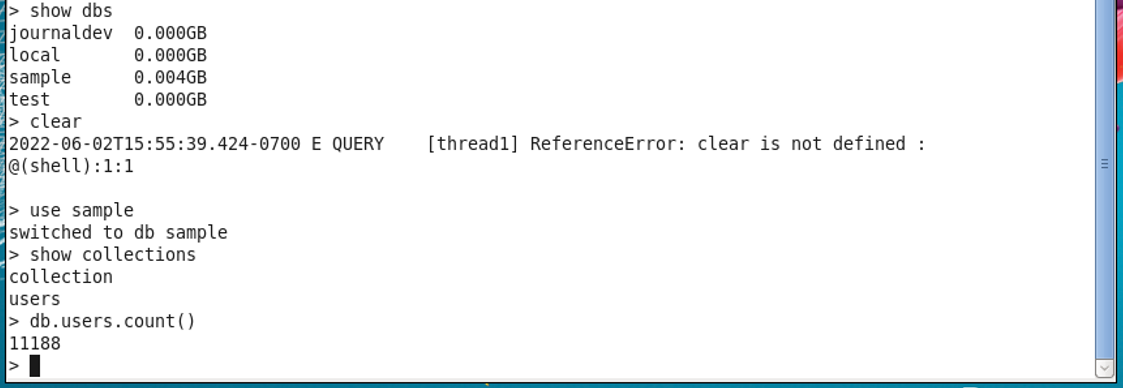
***Paso 2.*** En este siguiente paso veremos las DB y Colletions existentes

Ejecutamos show dbs

Ingresamos a DB “sample” con use sample

Vemos los collections existentes con show collections

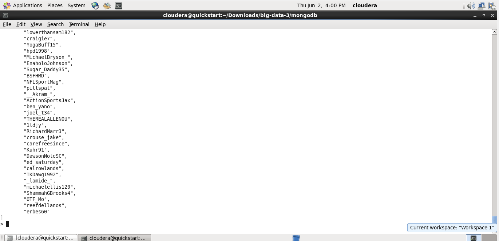
Finalmente contamos la cantidad de documentos Twitteados con db.users.count()



***Paso 3.*** Ahora nos adentraremos más en la información que podemos encontrar en los documentos de la DB y sus distintos valores

Ejecutamos db.users.findOne():

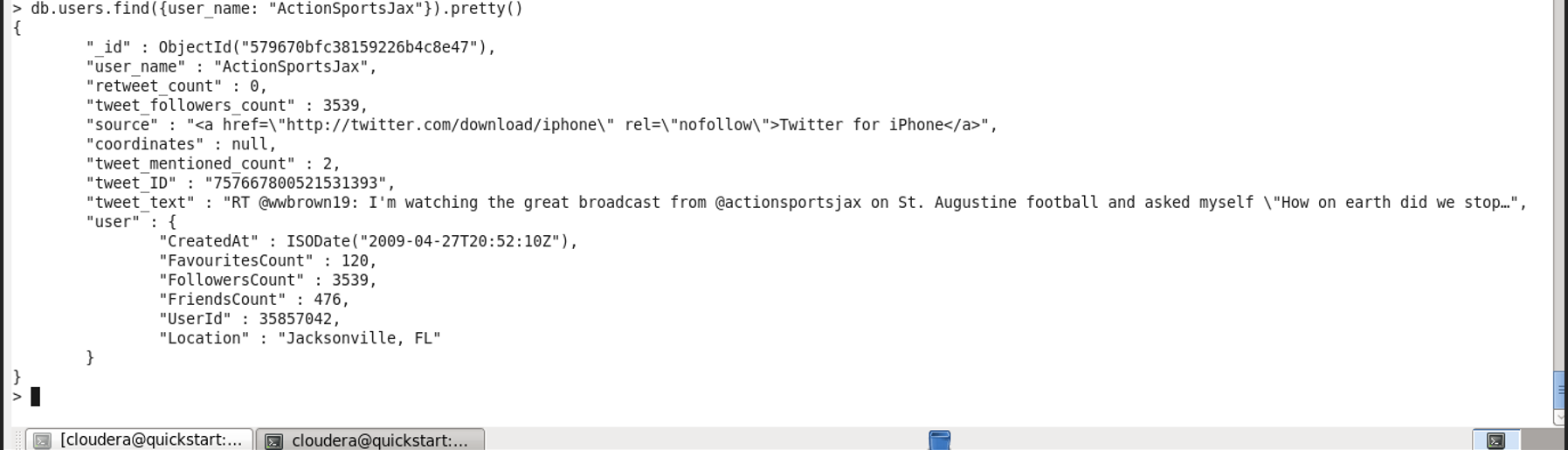
Buscamos los valores en nombre de usuario con db.users.distinct(“user\_name”)

***Paso 4.*** En este paso hacemos una buscada especifica en los campos que tenemos disponibles

Para buscar ejecutamos db.users.find({user\_name : “ActionSportsJax”})

Para formatear datos db.users.find({user\_name : “ActionSportsJax”}).pretty()

***Paso 5.*** Ahora procedemos a realizar filtros y obtener campos mas específicos

Ejecutamos para especificar un campo:

db.users.find({user\_name : “ActionSportsJax”}, { tweet\_ID: 1})

Además de poder eliminar un campo por ejemplo el ID:

db.users.find({user\_name : “ActionSportsJax”}, { tweet\_ID: 1, \_id: 0})



***Paso 6.*** Hacemos búsqueda con expresiones regulares como lo es “FIFA”

Usamos la expresión “FIFA”

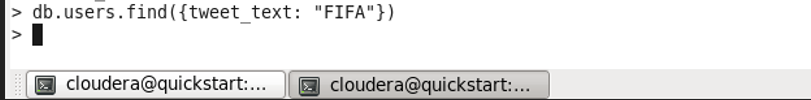
db.users.find({tweet\_text: “FIFA”})

Usamos la expresión regular /FIFA/:

db.users.find({tweet\_text: “FIFA”})

Contamos los resultados existentes:

db.users.find({tweet\_text: “FIFA”}).count()

***Paso 7.*** En este paso veremos que se puede acelerar las búsquedas y realizar búsquedas avanzadas.

Usamos el comando:

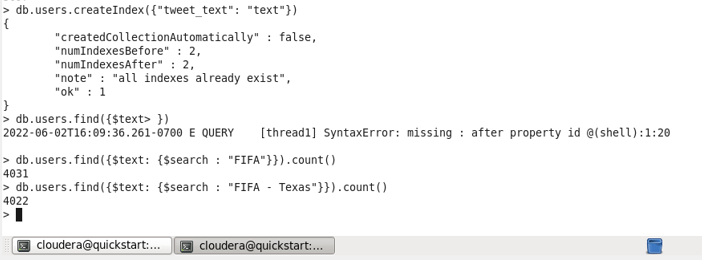
db.users.createIndex({tweet\_text: “text”})

Podemos buscar una collection especifica como lo es FIFA:

db.users.find({$text : {$search : “FIFA”}}).count()

También Podemos eliminar algún carácter o valor de la búsqueda:

db.users.find({$text : {$search : “FIFA -Texas”}}).count()



***Paso 8.*** En este paso hacemos búsquedas usando operadores

Por ejemplo, el siguiente código de búsqueda:

db.users.find({tweet\_mentioned\_count: {$gt : 6}})

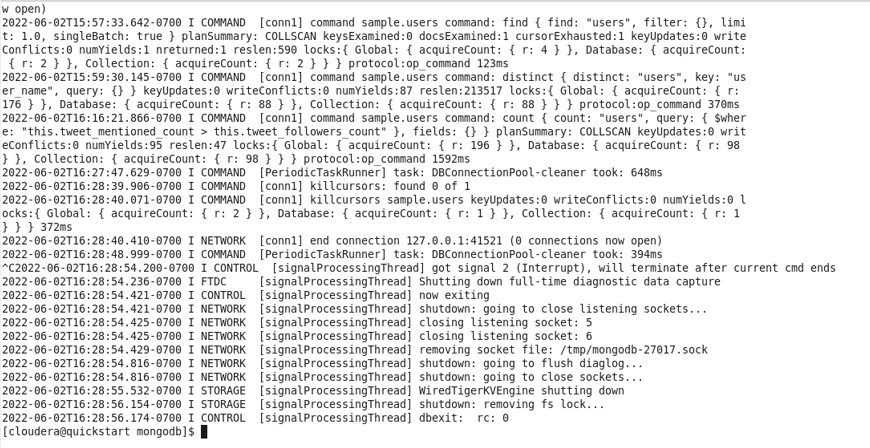
También se puede comparar entre campos pertenecientes al mismo documento:

db.users.find({$where : “this.tweet\_mentioned\_count > this.tweet\_followers\_count”})count()

Podemos combiner multiples busquedas:

db.users.find({$and : [ {tweet\_text : /FIFA/}, {tweet\_mentioned\_count: {$gt : 4}}]})count()

Finalmente cerramos MongoDB y finalizamos la ejecución en proceso

# Parte2 del Lab (Investigación)

En esta segunda parte nos solicitan hacer una investigación propia para exportar la collection users a un archivo json y entregarlo junto al informe

En nuestra investigación nos encontramos con un problema al ejecutar el comando correspondido por lo que ejecutamos uno antes que resuelve el problema particular que tuvimos, a esto nos referimos la descarga e instalación de MongoDB-Tools.



El comando que nos permite realizar la extracción es el siguiente:



El cual obtuvimos el archivo:



# Parte3 del Lab (Test)

**Pregunta 1**

What does it mean for a query language to be declarative?

1. The language specifies both the process of how to obtain the data and specifies what data

to obtain.

2. A language specific declaration of data types in order to define the method of data

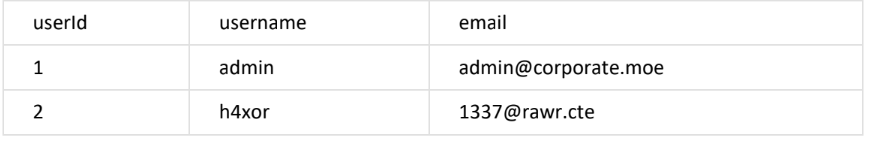
retrieval.

3. The language specifies the process of how to obtain the data.

4. The language specifies what data to obtain.

**Pregunta 2**

Use the following table named "user\_table" to answer the next 2 problems.



How would you go about querying the entire username column (however many)?

1. SELECT username FROM userId WHERE \*

2. SELECT user\_table FROM username

3. SELECT username FROM user\_table

4. SELECT username FROM user\_table WHERE userId=1

**Pregunta 3**

How would you go about querying the entire database table (please refer to question 2's table)?

1. SELECT user\_table FROM \*

2. SELECT \* FROM user\_table

3. SELECT \* FROM \* WHERE user\_table

4. SELECT username, email FROM userId

**Pregunta 4**

What is the global indexing table?

1. A global table that uses a specific technique called indexing and the table uses an index as

the primary key.

2. An index table in order to keep track of data records within one machine.

3. An index table in order to keep track of a given data type that might exist within one

machine.

4. An index table in order to keep track of a given data type that might exist within multiple

machines.

**Pregunta 5**

What are the three computing steps of a semi-join?

1. Project, Ship, Reduce

2. Project, Decompose, Send

3. Index, Join, Display

4. Query, Join, Display

5. None Applicable

**Pregunta 6**

What is the purpose of a semi-join?

1. Increase the efficiency of sending data across multiple machines.

2. Increase the speed of the join for trade-off of increased data transmission cost.

3. Another name for join: an operation to combine two tables by column.

**Pregunta 7**

What is a subquery?

1. A query statement within another query.

2. A short query than normal.

3. An alternative query that acts as a substitute for another query.

**Pregunta 8**

What is the purpose of GROUP BY queries?

1. Enables calculations based on specific columns of the table.

2. Enables queries within queries.

3. Required before you can use functions like AVG, SUM, MIN, MAX, COUNT.

**Pregunta 9**

Consider the following generic statement for questions 10-12:

db.<collection>.find(<query filter>, <projection>).<cursor modifier>

Which part of the statement would reflect that of the FROM statement in SQL?

1. <query filter>

2. <projection>

3. <collection>

4. <cursor modifier>

**Pregunta 10**

Which part of the statement would reflect that of the SELECT statement in SQL?

1. <cursor modifier>

2. <collection>

3. <query filter>

4. <projection>

**Pregunta 11**

Which part of the statement would reflect that of the WHERE statement in SQL?

1. <collection>

2. <query filter>

3. <projection>

4. <cursor modifier>

**Pregunta 12**

A sample part of the data structure is as follows:

{ \_id:1, userIndex: 10, email: “arealeamil@notreallu.asd", retainRate:2}

What would be the most likely statement that we would need to list email info for user indexes

greater than 24?

1. db.email.find({userIndex:{$gt:24}}, {email:1, \_id:0})

2. db.userIndex.find({email:{$lte:24}}, {\_id:0})

3. db.email.find({userIndex:{$lte:24}}, {email:1, \_id:0})

4. db.userIndex.find({email:{$gt:24}}, {\_id:0})

**Pregunta 13**

What does it mean to have a \_id:0 within our query statement?

1. Grab as many objects as possible.

2. Tell MongoDB not to return a document id.

3. Does not have an effect, simple convention left for compatibility issues.

4. Grab the first object in the results.