Sistemas de Gestión de Bases de Datos
$1^{er.}$ cuatrimestre 2019

Guía Práctica 4
NoSQL

Ingeniería Informática
Universidad Nacional de Avellaneda

1. Introducción

Los objetivos de esta práctica son:

- Crear una base de datos NoSQL (orientada a documentos)
- Realizar consultas de documentos
- Comprender nociones básicas de Map-Reduce
- Integrar python y MongoDB

2. Actividades preliminares

Para empezar a trabajar con MongoDB es necesario entender las nociones de "collection" y "document" para eso la sugerencia es leer estas referencias:

- mongodb_quick_reference_cards.pdf (pagina 2)
- Collections: `https://docs.mongodb.com/manual/core/databases-and-collections/`
- Documents: `https://docs.mongodb.com/manual/core/document/`

2.1. Utiles

El manual con todas las referencias es: `https://docs.mongodb.com/manual/contents/`

La referencia del cliente Mongo Shell es: `https://docs.mongodb.com/manual/reference/mongo-shell/`

La referencia básica del lenguaje de consulta es: `https://docs.mongodb.com/manual/tutorial/query-documents/`

Relación entre SQL y el lenguaje de consulta de MongoDB. Referencias:

- mongodb_quick_reference_cards.pdf (sección "Mapping SQL to MongoDB")
- `https://docs.mongodb.com/manual/reference/sql-comparison/`

En MongoDB no hay esquema, para ver las "keys" hay que examinar un documento de una colección. Por ejemplo, para ver la estructura de la collection "tweets" deberíamos hacer los siguiente:

```
Object.keys(db.tweets.findOne())
```

2.2. Instalación

1. Instalar MongoDB
2. Ejecutar el servidor de MongoDB (mongod)
3. Ejecutar el cliente de MongoDB (mongo)

Para instalar MongoDB (Community Edition) ver: `https://docs.mongodb.com/manual/installation/`

2.3. Importacion de datos

Seleccionar un subconjunto de tweets del archivo crisis.20190410.json:

```
head -10000 crisis.20190410.json > tweets.json (selecciona 5000 tweets)
```

Importar el conjunto de tweets en MongoDB:

```
mongoimport --db test --collection tweets --drop --file tweets.json
```

3. Ejercicios

3.1. Ejercicio 1

Realizar las siguientes consultas:

1. Seleccionar el id y el texto de 10 "documents"
2. Seleccionar los lenguajes distintos de los tweets
3. Seleccionar el id, el nombre, la descripción y la cantidad de followers de aquellos usuarios que tengan más de 100000 followers
4. Seleccionar el id, el nombre y la cantidad de followers de los 10 usuarios con más followers ordenado de manera descendente

3.2. Ejercicio 2

MongoDB provee soporte para el modelo de programación map-reduce. Analizar estas referencias:

- `https://docs.mongodb.com/manual/core/map-reduce/`
- `https://docs.mongodb.com/manual/tutorial/map-reduce-examples/`

Mediante la técnica de Map-Reduce implementar las siguientes consultas:

1. En base al campo "source", determinar la cantidad de usuarios que hay por cada canal.
2. Determinar la cantidad de tweets por cada lenguaje
3. Clasificar los textos de los tweets en corto (< 10 palabras), mediano (>= 10, < 20 palabras) y largo (>= 20 palabras). Devolver la cantidad de tweets en cada una de las clases. Referencia: `https://stackoverflow.com/questions/3559883/javascript-split-regex-question`

3.3. Ejercicio 3

MongoDB provee funcionalidades de agregación (similar a SQL):

- `https://docs.mongodb.com/manual/aggregation/`
- Ordenamiento: `https://docs.mongodb.com/manual/reference/operator/aggregation/sort/`
- Comparación *Map-Reduce/aggregate*: `https://docs.mongodb.com/manual/reference/aggregation-commands-comparison`
- Máximo: `https://docs.mongodb.com/manual/reference/operator/aggregation/max/`

Mediante la técnica de agregación implementar las siguientes consultas:

1. Listar los 10 usuarios que publicaron más tweets (ordenarlos de manera descendente por cantidad de tweets).
2. Listar por lenguaje la cantidad de followers del usuario con mayor cantidad de followers que publica en ese lenguaje

3.4. Ejercicio 4

Ahora se trabajará con todo el conjunto de tweets, para importarlo se puede ejecutar:

```
mongoimport --db test --collection tweets --drop --file crisis.20190410.json
```

Instalar el módulo para conectarse a MongoDB desde Python y revisar los artículos de referencia para consultar la base:

- `https://docs.mongodb.com/getting-started/python/client/`
- Consultas simples: `https://docs.mongodb.com/getting-started/python/query/`
- Consultas con agregación: `https://docs.mongodb.com/getting-started/python/aggregation/`

Diseñar estrategias en Python para realizar las siguientes tareas:

1. Determinar el origen de los usuarios basándose en: el campo "user.location" y la base de datos "world" de la práctica 3. Agregar un campo nuevo en cada "document" que explicite el país de origen del usuario. Una idea sería hacer chequeos incrementales que se ejecuten secuencialmente, por ejemplo:

- Si el campo “user.location” contiene un nombre de país, considero ese como país del usuario
- Los usuarios de EEUU muchas veces escriben su ubicación de forma “Ciudad,Estado” (ej.: “Washington, DC”). En ese caso hay que matchear el estado y si es válido hay que colocar “us” como país
- etc.

2. En base al punto anterior, realizar un Mapa Choropleth que refleje por cada país la cantidad total de tweets vinculados a usuarios de ese país
3. Basado en el punto 1, elegir 2 países y realizar una nube de palabras (una para cada país) de las 20 palabras más usadas en los campos “text” de los tweets de esos dos 2 países