Universidad Autónoma de Madrid

Departamento de Ingeniería Informática

Modelado, Almacenamiento y Gestión de Datos

Práctica - 4

Bases de datos NoSQL

Álvaro del Val

Índice

1. Objetivos

El objetivo de esta práctica es entender las similitudes y diferencias entre bases de datos SQL y NoSQL, así como crear una base de datos NoSQL en MongoDB y utilizarla para resolver consultas.

Seguiremos utilizando el conjunto de datos `airbnb`. En las prácticas anteriores trabajamos para diseñar y construir una base de datos relacional en PostgreSQL y resolver consultas sobre ella. Ahora, el objetivo es generar una base de datos NoSQL en MongoDB a partir del mismo conjunto de datos.

Para esta segunda práctica, se requiere:

- Familiarizarse con las herramientas gráficas y de línea de comandos para operar con MongoDB.
- Código python para obtener ficheros `json` a partir de los `csv` de `airbnb`.
- Creación de una base de datos NoSQL `airbnb` e importación de los ficheros generados en el paso anterior.
- Resolución de consultas sobre esta base de datos.
- Memoria en pdf.

2. Material

Se proporciona:

- El dataset `airbnb` de las prácticas anteriores
- Un fichero python que ilustra el código necesario para convertir los `csv` de `airbnb` en ficheros `json` de fácil importación
- Scripts de bash (se pueden usar tanto en Linux como en WSL y git bash); y Makefile que une todos los scripts y permite realizar las consultas.

3. Preprocesamiento de datos

MongoDB permite importar datos tanto desde ficheros `csv` como `json`. En la práctica, con csv funciona bien mientras no haya entradas complejas, como por ejemplo en formato json, por razones que puedes intentar descubrir por ti mismo (básicamente, las comillas son a la vez necesarias y un estorbo). Por lo tanto, lo que haremos será preprocesar todos los ficheros csv de nuestro dataset para convertirlo a formato json.

Os proporcionamos un fichero de ejemplo `csv2json.py` con código python para generar un fichero `keywords.json` a partir de los datos de la tabla json. Debéis generalizar este código para que genere ficheros json a partir de cada uno de los csvs del dataset `airbnb`. El programa deberá aceptar de 0 a 2 argumentos, con el primer y segundo argumentos representado el directorio con los csv de entrada, y el segundo argumento el directorio para los json de salida. Ambos argumentos son opcionales; si no se proporcionan, el directorio de entrada deberá llamarse `csv` y el de salida `json`.

Debéis generar colecciones para las ciudades de Menorca, Mallorca y Málaga.

Podéis consultar el dataset de aribnb de muestra para hacerse una idea del formato que queremos. Se pide generar ficheros json para listings, reseñas y calendarios, con los siguientes requisitos:

1. Las columnas que tienen formato json (amenities e identity_verified) en los csv originales están almacenadas como cadenas. Deben almacenarse como json, para lo que puedes usar ast.literal_eval.

2. Las claves que aparecen como "objetos" en los datos de muestra (images, hosts, availability, address) deben ser documentos embebidos dentro de los listados. Para eso tendrás que procesar las columnas correspondientes para generar el json anidado necesario.

4. Importación de los datos

Una vez generado un fichero json, este se puede importar a la base de datos airbnb con el comando `mongoimport`. Como ejemplo, se proporciona el fichero

`airbnb_sample.json`, que se puede importar de la siguiente manera.

```
mongoimport -d airbnb --collection=airbnb_sample --jsonArray mongodb://
    localhost:27017/ airbnb_sample.json
```

También podéis probar a importar cvs, como el `airbnb_sample.csv` o alguno de nuestro dataset, como *barcelona_listings_gz_small.csv*, etc.:

```
  mongoimport -d airbnb --collection sample_airbnb_csv --type=csv --
      headerline mongodb://localhost:27017 sample_airbnb
```

Se proporcionan scripts de bash (db_*.sh) y un Makefile que permiten automatizar estos procesos.

5. Consultas

1. Lista todos los alojamientos que son casas enteras o habitaciones privadas de Menorca disponibles entre el 2024-11-20 y el 2024-11-25. El resultado debe mostrar los siguientes campos: listing_id, name, neighbourhood, room_type, date, available, price, y estar ordenado por precio ascendente, date y listing_id. ordenadas de más reciente a más antigua.
2. Reseñas que incluyen alguna de las subcadenas “excelent” o “fantastic” y cuyo precio no supere 100€.
3. Precio medio de los alojamientos por ciudad. Mostrar ciudad, precio medio, y precio medio como porcentaje del máximo precio medio, ordenado por precio medio.
4. Alojamientos con al menos 10 reseñas y con una puntuación de al menos 4,5. Mostrar su nombre,d escripción y amenities, ordenando los resultados por número de reseñas.
5. La misma consulta, pero comprobando que las 10 reseñas están en la colección de reviews (no en el campo number_of_reviews)
6. Valoración media de los alojamientos por barrios

6. Memoria

La memoria es un documento en formato pdf que debe incluir, de forma **concisa pero informativa**, al menos los siguientes apartados:

- Descripción de las consultas.
- Pruebas realizadas para comprobar el resultado correcto de las mismas.

- Comentarios sobre el código python en `csv2json.py`.
- Comparación con las bases de datos relacionales: ¿qué ventajas y desventajas has encontrado en la gestión de este conjunto de datos con el modelo relacional y con NoSQL?
- Cualquier otro aspecto que consideréis relevante.

7. Entregable

Debéis entregar un único fichero zip llamado `P4_magd_XXXX_pYY.zip`, donde XXXX es vuestro grupo de prácticas (7261 grupo del martes, 7262 grupo del viernes), e YY es vuestro número de pareja, por ejemplo 03 si sois la pareja 3 (los números de parejas están publicados en la pèstaña de presentación de las prácticas).

El contenido de este fichero debe ser un **único directorio** de nombre `magd_XXXX_pYY` (es decir, igual que el zip pero sin la extensión).

Tiene que estar todo contenido dentro de un directorio porque, si no, al descomprimir todas las entregas se sobrescribirían los ficheros de unas parejas con los de otras.

Este directorio debe contener los siguientes ficheros:

- Fichero `csv2json.py` para preprocesar los datos
- Fichero Makefile para ejecutar el código python e importar los datos a la base de datos, y para ejecutar las queries (puede ser el mismo Makefile que os proporcionamos, no hay necesidad de cambiarlo).
- Scripts de bash necesarios invocados por el Makefile (no es necesario cambiar los que os proporcionan)
- Ficheros con las consultas
- Memoria en formato pdf.

8. Criterios de corrección

Generación de ficheros json: 3 puntos; consultas: 1 punto cada una; memoria 1 punto. También se valorarán la presentación y la redacción de la memoria, y que se comenten otros aspectos relevantes del contenido de la práctica.