

Bases de Datos 2023

Práctico VII: Operaciones CRUD

Juan Cabral, Tadeo Cocucci,
Ramiro Demasi, Guido Ivetta

Parte I

Contexto

La base de datos **mflix** contiene datos sobre películas y salas de cine. Adicionalmente incluye datos de usuarios y comentarios sobre las películas. A continuación se describen todas las colecciones de la base de datos mflix.

Colección	Descripción
movies	Contiene información de las películas, incluyendo título, año de estreno, y director.
theaters	Contiene ubicaciones de salas de cine.
users	Contiene información del usuario.
comments	Contiene comentarios asociados a películas específicas
sessions	Campo de metadatos. Contiene JSON Web Token de los usuarios.

I. Descargar el archivo mflix.tar.gz que contiene el dump de la base datos mflix

II. Cargar los datos desde el dump

```
$ tar xzvf mflix.tar.gz  
$ mongorestore --host <host> --drop --gzip --db mflix mflix/
```

Desde los labs:

```
$ tar xzvf mflix.tar.gz  
$ mongorestore --host 172.18.7.140 --drop --gzip --db mongo_gUSERID_db1 mflix/ -u  
MONGO_USER
```

III. Para ver el esquema de un documento usar el método `findOne()`, por ejemplo:

```
$ mongo --host <host>

> use mflix
> db.comments.findOne()
{
  "_id" : ObjectId("5a9427648b0beebeeb69579cc"),
  "name" : "Andrea Le",
  "email" : "andrea_le@fakegmail.com",
  "movie_id" : ObjectId("573a1390f29313caabcd418c"),
  "text" : "Rem officiis eaque repellendus amet eos doloribus...",
  "date" : ISODate("2012-03-26T23:20:16Z")
}
```

Desde los labs:

```
$ mongo --host 172.18.7.140 --authenticationDatabase mongo_gUSERID_db1 -u
MONGO_USER -pMONGO_PASSWORD
```

```
> use mongo_gUSERID_db1
> db.comments.findOne()
```

Tareas

1. Insertar 5 nuevos usuarios en la colección `users`. Para cada nuevo usuario creado, insertar al menos un comentario realizado por el usuario en la colección `comments`.
2. Listar el título, año, actores (`cast`), directores y `rating` de las 10 películas con mayor `rating` (`"imdb.rating"`) de la década del 90. ¿Cuál es el valor del `rating` de la película que tiene mayor `rating`? (Hint: Chequear que el valor de `"imdb.rating"` sea de tipo `"double"`).
3. Listar el nombre, email, texto y fecha de los comentarios que la película con `id` (`movie_id`) `ObjectId("573a1399f29313caabcee886")` recibió entre los años 2014 y 2016 inclusive. Listar ordenados por fecha. Escribir una nueva consulta (modificando la anterior) para responder ¿Cuántos comentarios recibió?
4. Listar el nombre, `id` de la película, texto y fecha de los 3 comentarios más recientes realizados por el usuario con email `patricia_good@fakegmail.com`.
5. Listar el título, idiomas (`languages`), géneros, fecha de lanzamiento (`released`) y número de votos (`"imdb.votes"`) de las películas de géneros Drama y Action (la película puede tener otros géneros adicionales), que solo están disponibles en un único idioma y por último tengan un `rating` (`"imdb.rating"`) mayor a 9 o bien tengan una duración (`runtime`) de al menos 180 minutos. Listar ordenados por fecha de lanzamiento y número de votos.
6. Listar el `id` del teatro (`theaterId`), estado (`"location.address.state"`), ciudad (`"location.address.city"`), y coordenadas (`"location.geo.coordinates"`) de los teatros que

se encuentran en algunos de los estados "CA", "NY", "TX" y el nombre de la ciudades comienza con una 'F'. Listar ordenados por estado y ciudad.

7. Actualizar los valores de los campos texto (text) y fecha (date) del comentario cuyo id es ObjectId("5b72236520a3277c015b3b73") a "mi mejor comentario" y fecha actual respectivamente.
8. Actualizar el valor de la contraseña del usuario cuyo email es joel.macdonel@fakegmail.com a "some password". La misma consulta debe poder insertar un nuevo usuario en caso que el usuario no exista. Ejecute la consulta dos veces. ¿Qué operación se realiza en cada caso? (Hint: usar upserts).
9. Remover todos los comentarios realizados por el usuario cuyo email es victor_patel@fakegmail.com durante el año 1980.

Parte II

Contexto

La base de datos **restaurantdb** contiene datos sobre restaurantes. A continuación se describe la colección que contiene.

Colección	Descripción
restaurants	Contiene información de restaurantes

I. Descargar el archivo restaurantdb.tar.gz

II. Importar los datos de un archivo json

```
$ tar xzvf restaurantdb.tar.gz
```

```
$ mongoimport --host <host> --db restaurantdb --collection restaurants --drop --file restaurantdb/restaurants.json
```

III. Para ver el esquema de un documento usar el método findOne(), por ejemplo:

```
$ mongo --host <host>
> use restaurantdb
> db.restaurants.findOne()
{
  "_id" : ObjectId("5eb3d669b31de5d588f48b94"),
  "address" : {
```

```

        "building" : "11002",
        "coord" : [
            -73.8318829,
            40.6842261
        ],
        "street" : "Liberty Ave",
        "zipcode" : "11419"
    },
    "borough" : "Queens",
    "cuisine" : "Caribbean",
    "grades" : [
        {
            "date" : ISODate("2015-01-15T00:00:00Z"),
            "grade" : "A",
            "score" : 12
        }
    ],
    "name" : "Trini Delite Roti Shop",
    "restaurant_id" : "50018608"
}

```

Tareas

10. Listar el id del restaurante (restaurant_id) y las calificaciones de los restaurantes donde al menos una de sus calificaciones haya sido realizada entre 2014 y 2015 inclusive, y que tenga una puntuación (score) mayor a 70 y menor o igual a 90.
11. Agregar dos nuevas calificaciones al restaurante cuyo id es "50018608". A continuación se especifican las calificaciones a agregar en una sola consulta.

```

{
    "date" : ISODate("2019-10-10T00:00:00Z"),
    "grade" : "A",
    "score" : 18
}

{
    "date" : ISODate("2020-02-25T00:00:00Z"),
    "grade" : "A",
    "score" : 21
}

```