



Clase 24 - Practica bases de datos documentales
Profesora: Erika Gutiérrez Beltrán

Tema 3: Bases de datos NoSQL



Objetivo

Construir bases de datos para datos no estructurados, comprender el funcionamiento y aplicación de estos tipos de bases de datos



Temario

- Arquitecturas de bases de datos manejadoras de grandes volúmenes de datos ✓
- Bases de datos bajo modelos Key - Value ✓
- Bases de datos orientadas a documentos ✓
- Bases de datos columnares
- Bases de datos orientadas a grafos



Logros

Interactuar con grandes volúmenes de datos, realizar consultas e interactuar con los datos



Modelado de datos NoSQL

Para recordar:

MongoDB es una base de datos NoSQL orientada a documentos que se enfoca en almacenar datos no estructurados.

Los datos que se almacenan en MongoDB son documentos que se agrupan en colecciones.

Formatos de los documentos

- JSON
- BSON
- GridFS



Modelado de datos NoSQL

Antes de construir una base de datos NoSQL es necesario llevar a cabo el modelamiento de los datos, que, a diferencia de una base de datos relacional, este presenta un esquema flexible el cual no obliga a que todos los ítems de una colección o tabla cumplan con el mismo conjunto de características.

Para determinar que base de datos debe usar una aplicación se debe identificar si:

1. Es necesario escribir mucho sobre la base de datos, pero poca lectura de la información (tendencia relacional)
2. Es necesario leer más datos que escribir (NoSQL)

Además, se debe tener en cuenta las características de rendimiento del motor de base de datos y patrones de recuperación.



Modelado de datos NoSQL

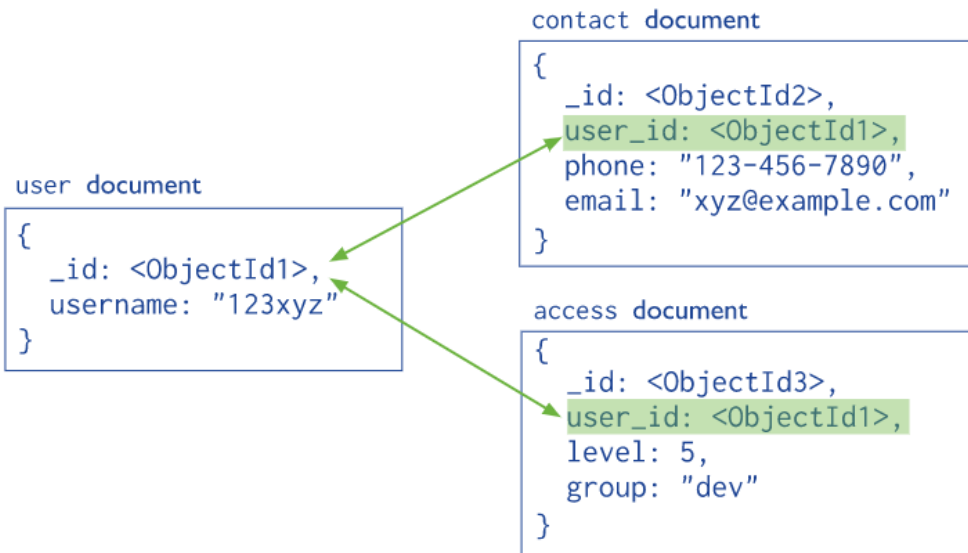
En el caso de MongoDB un documento atómico es aquel que tiene varios documentos relacionados entre sí en uno solo, ya que la escritura atómica es posible que se de a nivel de un solo documento.

Solo se “normaliza” un documento el cual contiene toda la información que se necesita desde una aplicación para la interacción con los datos.

En Mongo es posible modificar varios documentos que se encuentran relacionados entre sí y aunque un documento es atómico a nivel de modificaciones, la operación en conjunto no.



Modelado de datos NoSQL



Modelo de datos normalizado



Modelado de datos NoSQL

Patrones para el modelado de datos:

* **Modelo relacional entre documentos**

- Modelo uno a uno con documentos embebidos
- Modelos uno a muchos con documentos embebidos
- Modelo uno a muchos con documentos referenciados

* **Modelo de estructura en árbol**

- Modelo de estructura de árbol con referencia de los padres
- Modelo de estructura de árbol con referencia de los hijos
- Modelo de estructura de árbol con rutas materializadas
- Modelo de estructura de árbol con conjuntos anidados



Modelado de datos NoSQL

Modelo uno a uno con documentos embebidos

Para lograr una relación uno a uno a nivel de documentos, se incrusta los datos que estén conectados entre sí para reducir la cantidad de operaciones de lectura.

A continuación, se muestra un documento sin embebidos

```
// patron document
{
  _id: "joe",
  name: "Joe Bookreader"
}

// address document
{
  patron_id: "joe", // reference to patron document
  street: "123 Fake Street",
  city: "Faketon",
  state: "MA",
  zip: "12345"
}
```




Modelado de datos NoSQL

Modelo uno a uno con documentos embebidos

Documento con el modelo de uno a uno, ya que se unieron los datos asociados al usuario y se volvieron un solo documento

```
{
  _id: "joe",
  name: "Joe Bookreader",
  address: {
    street: "123 Fake Street",
    city: "Faketon",
    state: "MA",
    zip: "12345"
  }
}
```



Modelado de datos NoSQL

Modelos uno a muchos con documentos embebidos

Este tipo de modelo consiste en incrustar varios documentos que están relacionados en uno solo, lo que facilitaría y reduciría las consultas

```
{
  _id: "joe",
  name: "Joe Bookreader"
}

// address documents
{
  patron_id: "joe", // reference to patron document
  street: "123 Fake Street",
  city: "Faketon",
  state: "MA",
  zip: "12345"
}

{
  patron_id: "joe",
  street: "1 Some Other Street",
  city: "Boston",
  state: "MA",
  zip: "12345"
}
```



Modelado de datos NoSQL

Documento con el modelo uno a muchos embebido, ya que se evidencia la unión de la información en un solo documento

```
{
  "_id": "joe",
  "name": "Joe Bookreader",
  "addresses": [
    {
      "street": "123 Fake Street",
      "city": "Faketon",
      "state": "MA",
      "zip": "12345"
    },
    {
      "street": "1 Some Other Street",
      "city": "Boston",
      "state": "MA",
      "zip": "12345"
    }
  ]
}
```



Modelado de datos NoSQL

Patrón de subconjunto

El modelo por documento embebido tiene problemas y es que puede generar documentos de gran tamaño, ocasionando que el acceso a la información sea complejo y requiera de mucha lógica para obtener alguna cantidad de información .

El patrón de subconjunto indica que se puede limitar la cantidad de información que tengo de uno a muchos solo con lo más relevante o lo que voy a ver en pantalla en primera instancia, luego a nivel detallado, se iría al documento encargado de tener el resto de información complementaria.

Vamos a pensar en el ejemplo de Producto y sus Calificaciones, ¿cómo haríamos su estructura teniendo en cuenta esto?



Modelado de datos NoSQL

Modelos uno a muchos con documentos referenciados

Este modelo permite crear vínculos o referencias entre documentos que tendrán gran tamaño. Por ejemplo

```
{
  title: "MongoDB: The Definitive Guide",
  author: [ "Kristina Chodorow", "Mike Dirolf" ],
  published_date: ISODate("2010-09-24"),
  pages: 216,
  language: "English",
  publisher: {
    name: "O'Reilly Media",
    founded: 1980,
    location: "CA"
  }
}

{
  title: "50 Tips and Tricks for MongoDB Developer",
  author: "Kristina Chodorow",
  published_date: ISODate("2011-05-06"),
  pages: 68,
  language: "English",
  publisher: {
    name: "O'Reilly Media",
    founded: 1980,
    location: "CA"
  }
}
```



Modelado de datos NoSQL

A continuación, se muestra el patrón aplicado sobre el documento

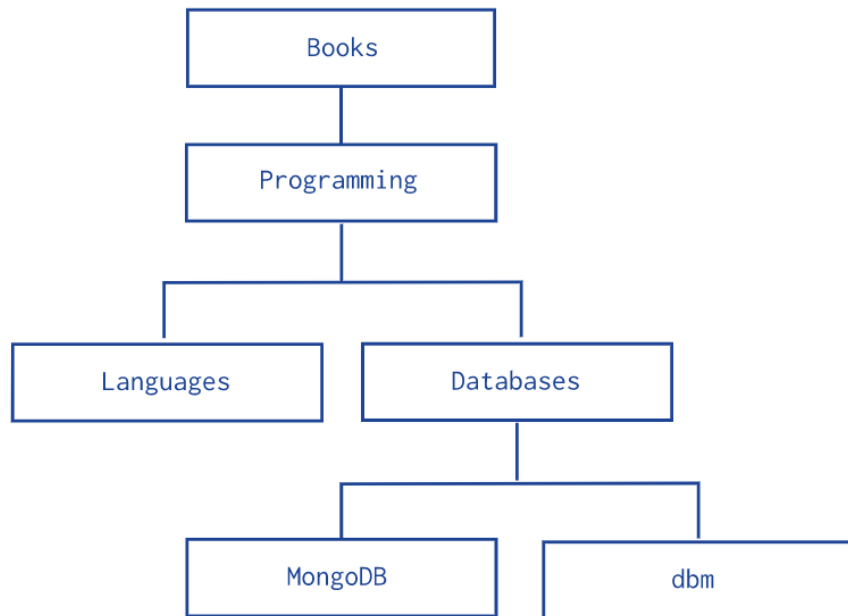
```
{
  name: "O'Reilly Media",
  founded: 1980,
  location: "CA",
  books: [123456789, 234567890, ...]
}

{
  _id: 123456789,
  title: "MongoDB: The Definitive Guide",
  author: [ "Kristina Chodorow", "Mike Dirolf" ],
  published_date: ISODate("2010-09-24"),
  pages: 216,
  language: "English"
}

{
  _id: 234567890,
  title: "50 Tips and Tricks for MongoDB Developer",
  author: "Kristina Chodorow",
  published_date: ISODate("2011-05-06"),
  pages: 68,
  language: "English"
}
```



Modelado de datos NoSQL

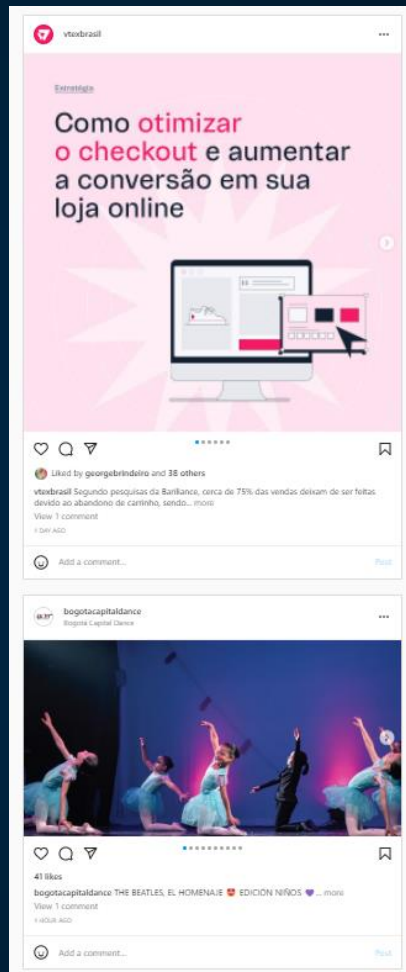


Modelo de estructura en árbol



Practica MongoDB

Ahora, teniendo en cuenta lo anterior, cómo modelarías la base de datos para Instagram y por qué. ¿Utilizarías NoSQL?, en que casos usuarios NoSQL y en que otros relacional





Practica MongoDB

Para validar que tienes instalado no solo el cliente de MongoDB (Compass) con el cual se hará la interacción con la base de datos, sino que tenemos el servidor instalado en nuestro computador ejecutamos en consola el comando

```
mongod --version
```

Para esto es necesario tener configurado MongoDB en las variables de entorno:

C:\Program Files\MongoDB\Server\5.0\bin → agrega esta ruta en el path. Recuerda modificar según la ubicación de tu pc. Cierra la consola o terminal y ejecuta el comando previo.



Practica MongoDB

Cuando quede configurado, al ejecutar el comando en consola se deberá visualizar algo como lo que se muestra a continuación:

```
C:\Users\erika\OneDrive\Desktop\Clases-Eafit\Bases de Datos>mongod --version
db version v5.0.8
Build Info: {
  "version": "5.0.8",
  "gitVersion": "c87e1c23421bf79614baf500fda6622bd90f674e",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}
```



Practica MongoDB

- Para iniciar mongo en la terminal ejecuta el comando **mongo**
- Para ver información de la base de datos ejecuta el comando **db.stats()**. Con este comando podrás ver las colecciones creadas para la base de datos, tamaño y demás.
- Si tienes varias bases de datos o para usar una especifica utiliza el comando **use nombre_db**

```
> use test  
switched to db test
```



Practica MongoDB

- **Show dbs**, muestra todas las bases de datos que se tienen creadas
- Ejecute **db** si desea ver en que base de datos está posicionado
- Para crear una nueva colección ejecute **db.createCollection("Envios")** o **db.createCollection("Envios", {capped: true, max:1, size:200 })** , esta última agrega una colección limitada en espacio
- **Show collections** para ver todas las colecciones creadas en esa base de datos



Practica MongoDB

- `db.nameDB.insertOne({ id_envio: "123", nombre_repartidor: "pepe", origen: "xyz", destino: "xyz" })` Insertar un solo registro

- `db.nameDB.insertMany([
 { id_envio: "123", nombre_repartidor: "pepe", origen: "xyz", destino: "xyz"},
 { id_envio: "124", nombre_repartidor: "violeta", origen: "xyz", destino: "xyz"}
])` Insertar múltiples registros al tiempo

- Registro uno a uno:

```
Var bulk = db.nameDB.initializeUnorderedBulkOp();  
bulk.insert({ id_envio: "123", nombre_repartidor: "pepe", origen: "xyz", destino: "xyz"});  
bulk.execute();
```



Practica MongoDB

- **db.nameDB.find()** Busca los documentos de la colección
- **db.nameDB.find({ idenvio: { \$eq: "123" } })** Busca los documentos de la colección con parámetros de búsqueda
- **db.nameDB.findOne()** Busca los documentos de la colección con salida formateada
- Para dar formato a la consulta de datos agregar al final `.pretty()`



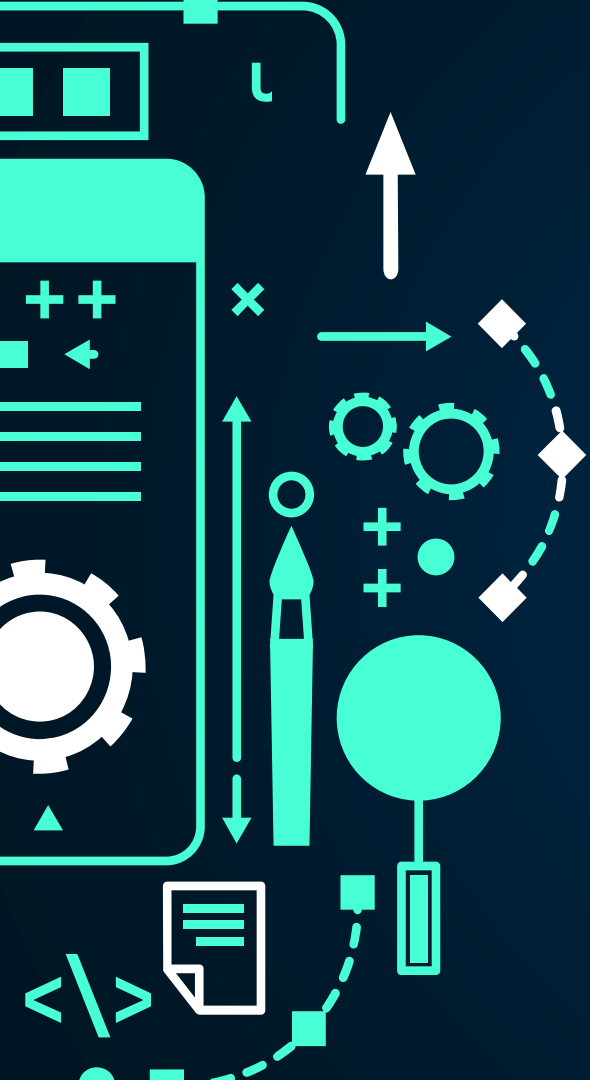
Practica MongoDB

- **db.nameDB.update**({“idenvio”: “123”}, { \$set: {“origen”: “AAA”}}) Actualizar datos de un documento
- **db.nameDB.updateMany**({ “idenvio”: { \$eq: “123” }, { \$set: { “idenvio”: “Inactive”}}) Actualizar múltiples datos
- **db.nameDB.deleteOne**({“idenvio”:”123”}) Elimina un elemento
- **db.nameDB.deleteMany**({“origen”:”xyz”}) Elimina varios elementos que coinciden con el criterio de búsqueda



REFERENCIAS

- Documentación de MongoDB: <https://www.mongodb.com/docs/manual/core/capped-collections/#:~:text=Create%20a%20Capped%20Collection,pre%2Dallocate%20for%20the%20collection.>
- ¿Qué es una base de datos de documentos? <https://aws.amazon.com/nosql/document/>



Gracias!