Capacitación MongoDB

mongoDB

sadosky
2009- BEST IT COMPANY
2013- LIFETIME BUSINESS ACHIEVEMENT
2013- BEST IT SOLUTION

EUROPEAN SEAL OF E-EXCELLENCE AWARD

ExportAr Argentina
SERVICES EXPORTER OF THE YEAR 2011

CMMI
LEVEL 3 APPRAISED

GREAT PLACE TO WORK
INSTITUTE ARGENTINA
2011- 2ND PLACE
2013- 7TH PLACE

нехacта

Agenda

- BDs Relacionales VS MongoDB
- ¿Qué es MongoDB?
- Modelado de datos en Mongo
- Mongo Shell
- Operaciones CRUD
- Aggregation
- Leyendo desde una app NodeJS
- Performance

BDs Relacionales VS MongoDB

¿Qué es MongoDB?

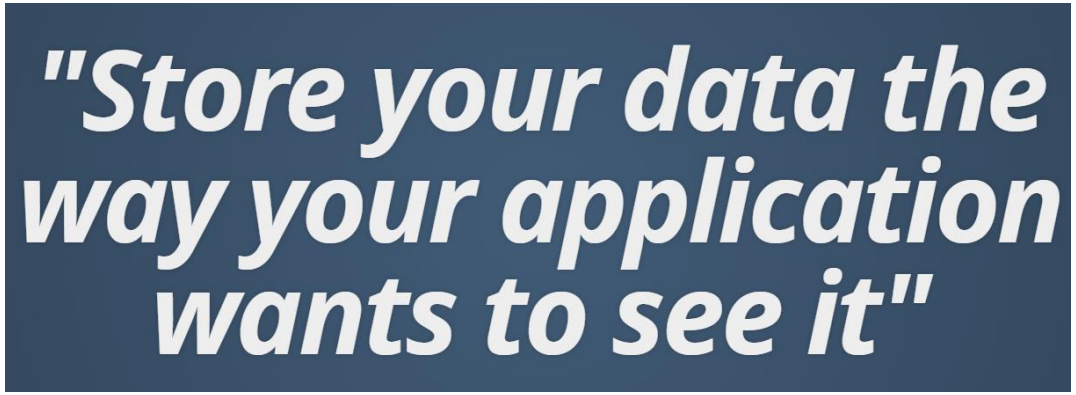
¿Qué es MongoDB?

- MongoDB es una base de datos documental de código abierto.
- Facilita el desarrollo proveyendo alta performance, alta disponibilidad y fácil escalabilidad.
- El motor almacena N cantidad de bases de datos, cada una de las cuales almacena un set de colecciones y cada colección almacena un set de documentos. Un documento es un par clave-valor.
- Los documentos son dinámicos, lo que significa que los documentos almacenados en una colección no necesariamente deben tener el mismo set de campos ni que deben compartir la misma estructura e incluso los campos comunes almacenados en distintos documentos no necesariamente tienen que ser del mismo tipo.
- Un documento en MongoDB es un registro cuya estructura de datos se compone de pares clave-valor. Estos documentos son similares a un objeto JSON. Los valores de los campos pueden incluir otros documentos, listas y/o listas de documentos.
- El siguiente es un ejemplo de un documento:

Modelado de datos en Mongo

Modelado de datos en Mongo

- Los datos en Mongo tienen un *esquema flexible*. Las colecciones en Mongo no deben cumplir con una estructura definida, lo que facilita mapear un documento a una entidad u objeto.
- El desafío a la hora de modelar datos está en balancear las necesidades de la App, las características de performance del motor de base de datos y los patrones de recuperación de datos.



Modelado de datos en Mongo

Estructura de un Documento

> La decisión sobre el diseño de los modelos de datos para MongoDB se resuelven alrededor de la estructura de los documentos y de como la App representa las relaciones entre los datos.

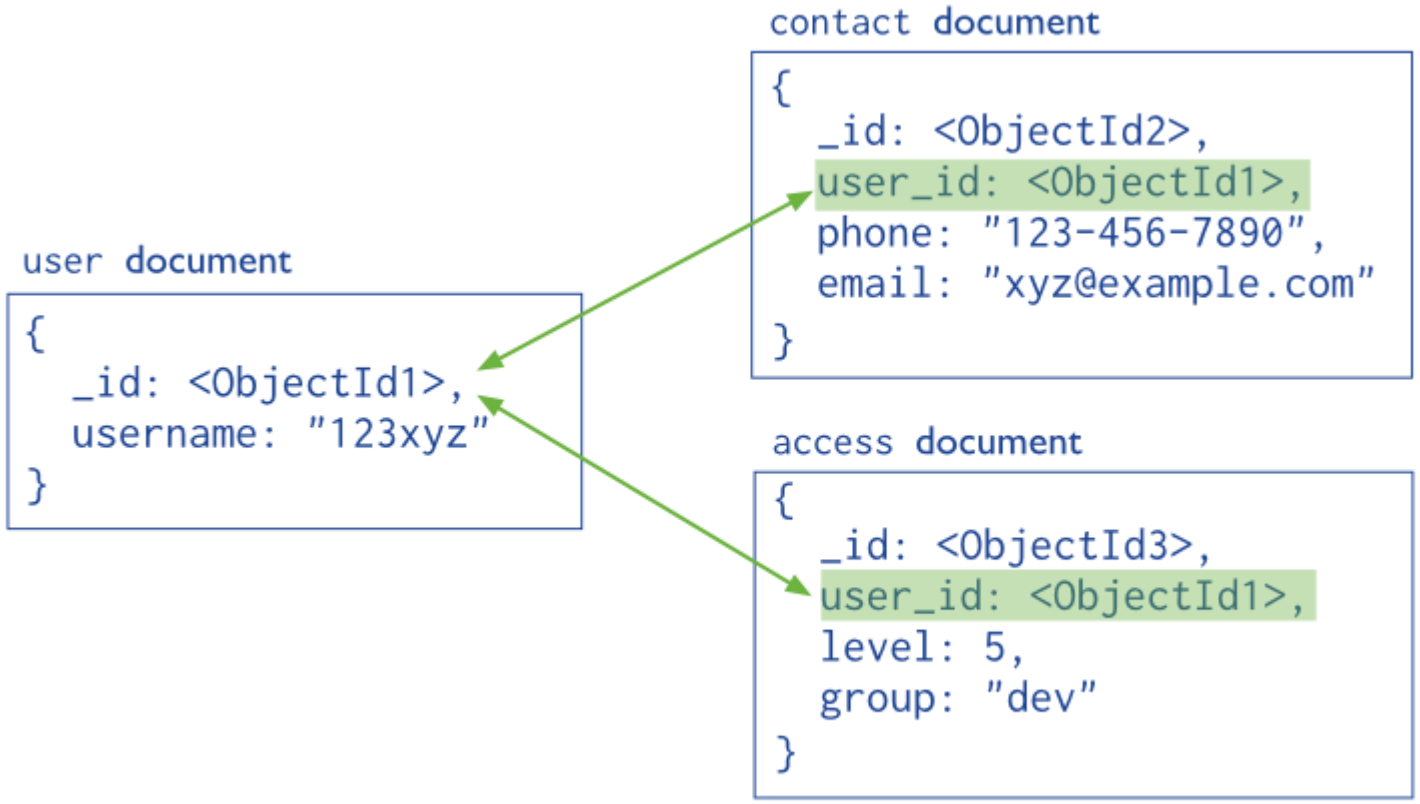
> Existen dos herramientas que permiten a las Apps representar estas relaciones:

- REFERENCIAS
- DUCUMENTOS EMBEBIDOS

Modelado de datos en Mongo

Referencias

- Almacenan las relaciones entre los datos incluyendo *links* desde un documento a otro.
- La Apps pueden resolver esas referencias accediendo a los datos relacionados.
- En términos generales, estos son los modelos de datos normalizados.



Modelado de datos en Mongo

Documentos Embebidos

- Capturan las relaciones entre los datos almacenando la información relacionada en una única estructura documental.
- Los documento en MongoDB hacen posible embeber estructuras en un campo o arrays dentro de un documento.
- Este modelo de datos *desnormalizado* permite a las Apps obtener y manipular datos relacionados en una única operación.

Modelado de datos en Mongo

Factores Operacionales y Modelos de Datos

- Credumiento del Documento: algunas actualizaciones sobre los documentos pueden incrementar el tamaño de los mismos.
- Atomicidad: Las operaciones son *atómicas*. Lo que significa que una operación no puede cambiar mas de un documento. Operaciones que modifican mas de un documento en una colección se ejecutan sobre un documento a la vez.
- Sharding: MongoDB utiliza el *sharding* para proporcionar escalabilidad horizontal. Esto permite particionar una colección dentro de la base de datos para distribuir los documentos de una colección a través de una serie de instancias de Mongo.
- Indices: Los índices son usados para mejorar la performance. Se pueden crear índices a partir de los campos de uso común dentro de las queries que se manejan en la App. Ademas Mongo crea índices únicos para el campo *_id*.
- Las Colecciones contienen un gran número de Documentos: De ser posible, hay que evitar que se dé esta condición por motivos de performance. Es aconsejable juntar estos objetos mas pequeños mediante alguna lógica que los agrupe. De esta manera habrá pocas copias de los campos comunes y habrá pocas entradas claves para el correspondiente índice.
- Optimización de Almacenamiento para pequeños Documentos: Cada documento en MongoDB contiene algún tipo de *sobrecarga* la cual es normalmente insignificante, pero se torna lo contrario si los documentos son pequeños (uno o dos campos máximo).

Mongo Shell

Mongo Shell

Factores Operacionales y Modelos de Datos

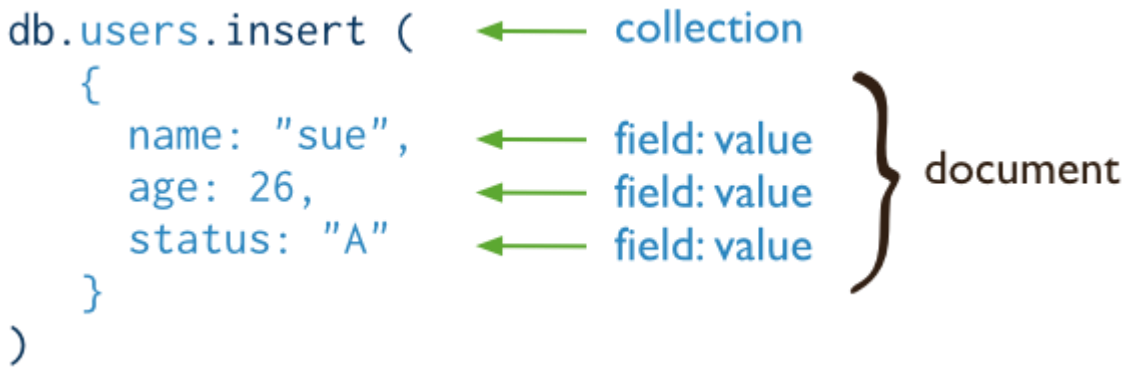
- Es una intuitiva interfaz que se basa en JavaScript y se conecta a MongoDB.
- Se puede usar para correr queries y actualizaciones y aplicar operaciones administrativas.
- Es un componente de las distribuciones MongoDB. Se instala junto con mongo y se puede usar para correr una instancia de MongoDB.

Operaciones CRUD

Operaciones CRUD

Operaciones CREATE

- Representan las operaciones de creación o inserción de un nuevo documento a una colección.
- Si la colección no existe al momento de la inserción, la misma es creada y el documento guardado.
- Métodos que existen actualmente:
- Db.collection.insert();
- Db.collection.insertOne();
- Db.collection.insertMany();



Operaciones CRUD

Operaciones READ

- Representan las operaciones de lectura de una coleccion.
- Métodos que existen actualmente:
- Db.collection.find();

```
db.users.find(                          ← collection
   { age: { $gt: 18 } },                ← query criteria
   { name: 1, address: 1 }              ← projection
).limit(5)                              ← cursor modifier
```

Operaciones CRUD

Operaciones UPDATE

- Representan las operaciones de modificación de un documento que ya existe en una colección.
- En Mongo, el *objetivo* de una modificación es una colección
- Las operaciones de escritura son *atómicas* a nivel de documento.
- Métodos que existen actualmente:
- Db.collection.update();
- Db.collection.updateOne();
- Db.collection.updateMany();
- Db.collection.replaceOne();

```
db.users.update(                    <-- collection
   { age: { $gt: 18 } },            <-- update criteria
   { $set: { status: "A" } },       <-- update action
   { multi: true }                  <-- update option
)
```

Operaciones CRUD

Operaciones DELETE

- Representan las operaciones de remoción de documentos contenidos en una coleccion.
- En Mongo, el *objetivo* de una remoción es una colección
- Las operaciones de escritura son *atómicas* a nivel de documento.
- Métodos que existen actualmente:
- Db.collection.remove();
- Db.collection.removeOne();
- Db.collection.removeMany();

```
db.users.remove(        <-- collection
   { status: "D" }      <-- remove criteria
)
```

ARGENTINA
Clay 2954
Buenos Aires (C1426DLD)
tel: 54+11+5299 5400

BRASIL
Cardoso de Melo 1470 – 8, Vila Olimpia
San Pablo (04548004)
tel: 55+11+3045 2193

URUGUAY
Roque Graseras 857
Montevideo (11300)
tel: 598+2+7117879

USA
12105 Sundance Ct.
Reston (20194)
tel:+703 842 9455

HexactaArg

@Hexacta

hexacta

www.hexacta.com