



mongoDB

Dr. En C. Sergio Galván Cruz

Antecedentes

- El modelo de datos quizás más conocido es el SQL. Dicho modelo realiza la gestión de sus BD basándose en la teoría relacional y teniendo como unidad principal de almacenamiento las tablas (estructuras fijas para almacenar datos).
- **Comerciales:** Oracle y Microsoft SQL Server
- **Software Libre:** Postgre SQL y MySQL.

Antecedentes

- ▶ Los sistemas NoSQL (no sólo SQL) en ocasiones (dependiendo del sistema) pueden soportar consultas tipo SQL.
- ▶ Dentro de las BD NoSQL se encuentran múltiples categorías como: clave-valor, BigTable, BD documentales y BD orientadas a grafos. Dicha clasificación se realiza de acuerdo a su forma de **almacenar** la información. **Existen más clasificaciones.**
- ▶ Actualmente existen más de 150 sistemas de BD NoSQL.

Antecedentes

- ▶ Ejemplos de sistemas NoSQL:
- ▶ Hadoop / Hbase.
- ▶ Cassandra.
- ▶ Amazon SimpleDB.
- ▶ Coudata. (Clon de Big Table parecido a HBase).
- ▶ Big Data.
- ▶ CouchDB.
- ▶ VertexDB
- ▶ Trinity.
- ▶ GraphBase.
- ▶ WhiteDB
- ▶ ZopeDB (ZODB)




Introducción



- ▶ El nombre de **mongoDB** proviene de la palabra en inglés **humongous** lo que significa es enorme.
- ▶ Sistema de Base de datos NoSQL.
- ▶ Desarrollado bajo el concepto de código abierto.
- ▶ Su desarrollo comenzó en 2007 por la compañía 10gen.



Introducción

- ▶ No se utilizan tablas como las BD relacionales.
 - ▶ Se utilizan documentos tipos JSON para guardar los datos con un formato que gestiona MongoDB de nombre BSON
- 




¿Qué es Mongo-DB?

- Es un motor de base de datos open-source el cual suministra un alto desempeño, alta disponibilidad y gran escalamiento.
- 



Documentos Mongo

- ▶ Un registro en MongoDB es un documento el cual se encuentra estructurado de sus respectivos campos y valores.
 - ▶ Dichos documentos son similares a objetos JSON.
 - ▶ Los valores de los campos pueden incluir otros documentos, arreglos y arreglos de documentos.
- 

Documentos Mongo (cont...)

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value



Documentos Mongo

► Ventajas

- El archivo puede almacenar tipos de datos nativos de sistemas o bien de lenguajes de programación.
- Almacenamiento de otros documentos y arreglos lo cual reducen la necesidad de realizar joins costosos.
- Esquema dinámico que soporta polimorfismo.

“


Principales características

”






Principales características: Desempeño

- Reducción de la actividad de I/O para el sistema de BD.
 - Indexación para la realización de queries con mayor velocidad.
 - Soporte para modelos de datos embebidos.
 - Provee mayor consistencia en ambientes de un gran número de transacciones.
- 



Principales características: Disponibilidad

- ▶ Replicación es simple.
 - ▶ Conmutación de errores (failover).
 - ▶ Redundancia de datos.
- 



Principales características: Escalabilidad

- ▶ Provee escalabilidad horizontal como parte principal del núcleo.
- ▶ Distribución de datos a través de un conjunto de máquinas.
- ▶ Provee alto rendimiento a una baja latencia (ambientes replicados)

“

CRUD MongoDB

”





CRUD MongoDB



- **CRUD** es el acrónimo en inglés de (Create, Read, Update and Delete).
- **MongoDB** almacena los datos en forma de documentos los cuales son pares campo-valor. Dichos documentos son análogos con estructuras de programación asociados con claves (llaves), por ejemplo: diccionarios, hashes, mapas y arreglos asociativos).
- Formalmente, los documentos de MongoDB son de tipo BSON, es decir, es una representación Binaria de JSON.

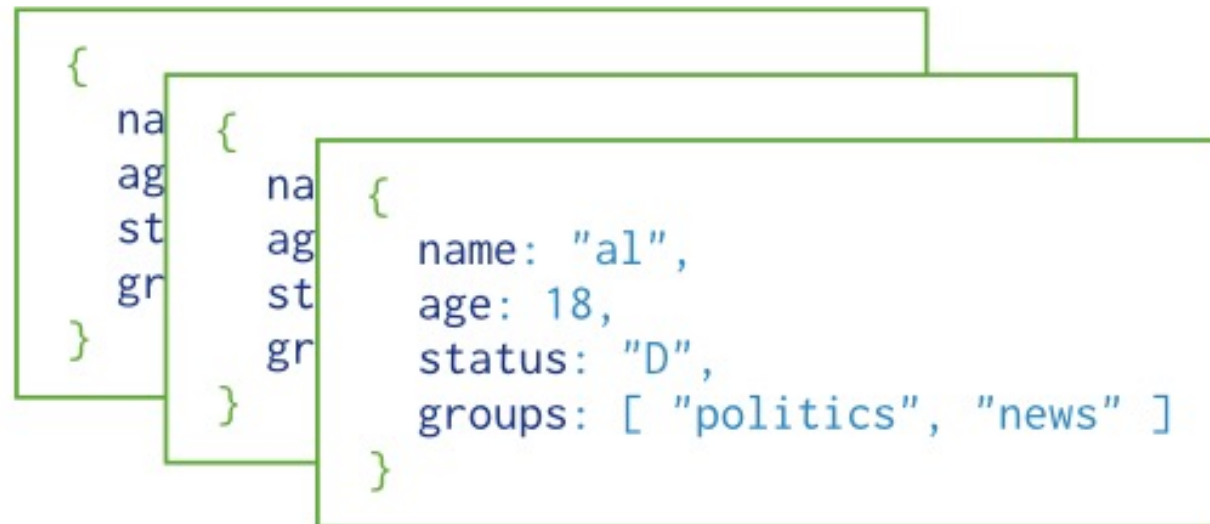
CRUD MongoDB

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value
← field: value
← field: value
← field: value

CRUD MongoDB

- El almacenamiento en MongoDB es en **colecciones**. Una colección es un grupo de documentos relacionados los cuales tienen índices comunes relacionados.



Collection

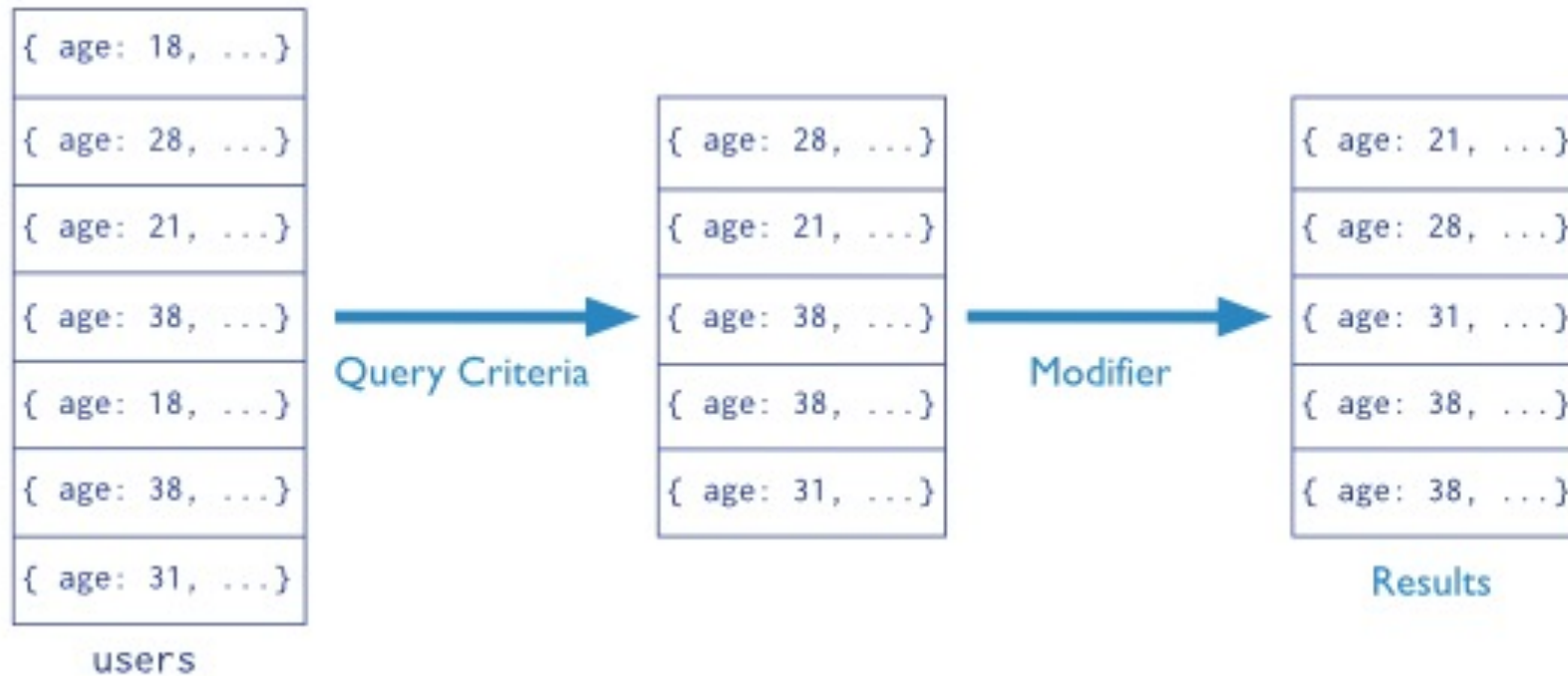
CRUD MongoDB

➤ Query

- En MongoDB un query se especifica sobre una colección de documentos. Los **queries** especifican criterios, condiciones que serán regresadas a los clientes.
- Además, un query debe incluir una **proyección** la cual especifica los campos que concidirán con los documentos que serán regresados.
- Finalmente, un query impone limitaciones, omisiones y ordenaciones.

CRUD MongoDB

Collection Query Criteria Modifier
`db.users.find({ age: { $gt: 18 } }).sort({age: 1 })`





CRUD MongoDB



Modificaciones





CRUD MongoDB

➤ **Modificaciones**

- Se refiere a la creación, actualización y eliminación de los datos.
- Dichas operaciones modifican los datos de una sola colección.
- En el caso de las operaciones de actualización y eliminación, se deberá especificar el criterio para que se realicen.

CRUD MongoDB

Collection
↓
`db.users.insert(`

Document
↓

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

)

Document

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

insert

Collection

{ name: "al", age: 18, ... }
{ name: "lee", age: 28, ... }
{ name: "jan", age: 21, ... }
{ name: "kai", age: 38, ... }
{ name: "sam", age: 18, ... }
{ name: "mel", age: 38, ... }
{ name: "ryan", age: 31, ... }
{ name: "sue", age: 26, ... }

users

“

Características extra

”



Características extra

➤ Índices.

- Su finalidad es incrementar el desempeño de los queries y de las actualizaciones.
- MongoDB tiene soporte para índices secundarios.
- Los índices además solidifican la **unicidad**, ayudan a las representaciones **geoespaciales**, y facilitan las **búsquedas de texto**.



Características extra

- **Réplica. (Replica Set Read Preference)**
- Se realiza para un conjunto de clusters compartidos y aplicaciones que tienen acceso a esos componentes.



Características extra

- **Escritura.**
- Las aplicaciones tiene el control en las escrituras. Particularmente en la utilización de conjuntos de réplicas.
- MongoDB provee reportes sobre el éxito de las operaciones de escritura.



Características extra

- **Agregación.**
- Son operaciones adicionales a los queries básicos.
- Por ejemplo:
 - MongoDB puede regresar el número de documentos que coinciden con una búsqueda o el número de valores distintos para un campo.
 - Se puede utilizar procesos versátiles de procesamiento como los **pipeline** o **map-reduce**.

“


Modelado de Datos

”





Modelado de Datos

- Los datos en MongoDB tienen un esquema flexible.
 - En los modelos SQL se requiere determinar y declarar el esquema de la tabla antes de insertar los datos.
 - La flexibilidad facilita el mapeo de los documentos a una entidad u objeto.
 - Cuando se diseñe el modelo de datos siempre se debe de considerar el uso de la aplicación (queries, updates y procesamiento general).
- 



Modelado de Datos

- ▶ El punto clave en el diseño de los modelos de datos para MongoDB se centra en la estructura de los documentos y las relaciones que se presentan entre los datos.
- ▶ Podemos mencionar dos rubros del modelado:
 1. Referencias
 2. Documentos embebidos.



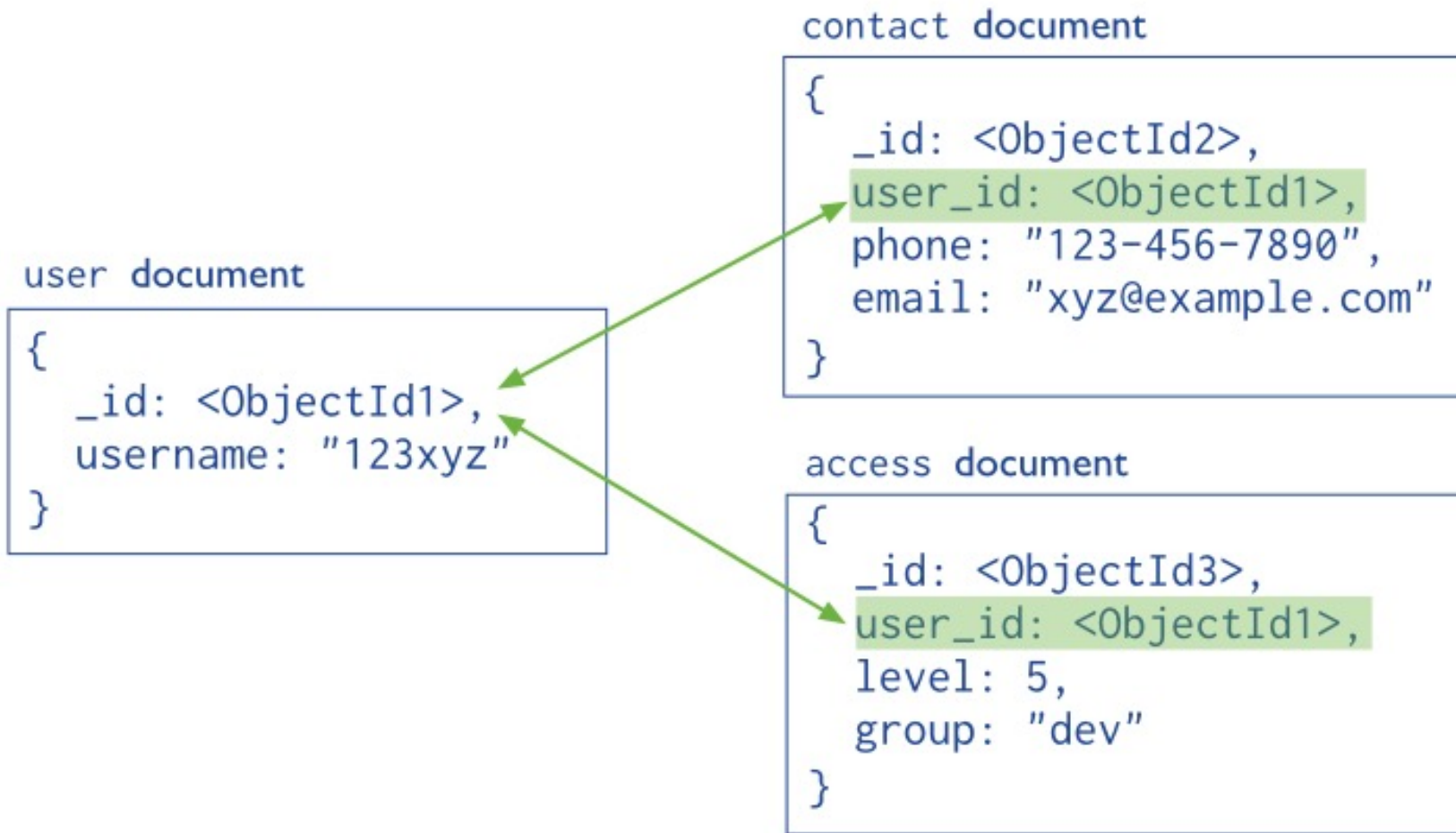
Modelado de Datos

➤ REFERENCIAS

- Almacenan relaciones entre los datos incluyendo vínculos o referencias a otro documento. Son ampliamente utilizados debido a su parecido con los modelos normalizados de las BD.

Modelado de Datos

➤ REFERENCIAS





Modelado de Datos

- ▶ **Datos embebidos**

- ▶ Son documentos los cuales cuentan con relaciones entre datos en un solo archivo. MongoDB permite que la estructura de un archivo tenga datos incrustados.

Modelado de Datos

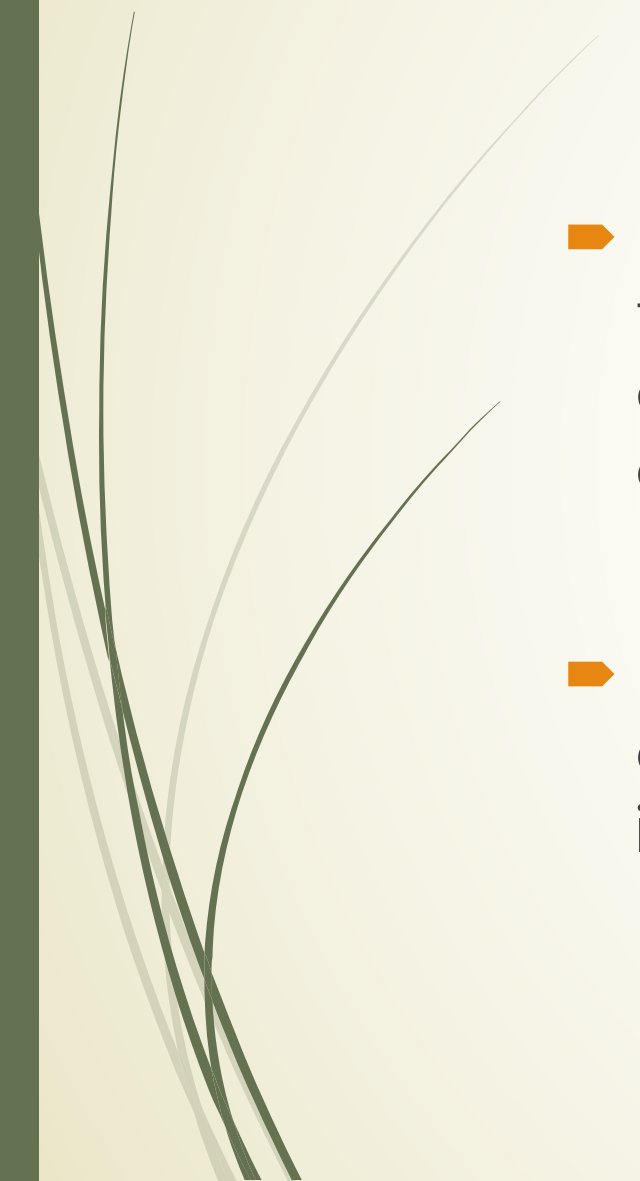
```
{  
  _id: <ObjectId1>,  
  username: "123xyz",  
  contact: {  
    phone: "123-456-7890",  
    email: "xyz@example.com"  
  },  
  access: {  
    level: 5,  
    group: "dev"  
  }  
}
```

Embedded sub-document

Embedded sub-document



Modelado de Datos

- ▶ En MongoDB se debe de asegurar la atomicidad de las transacciones y por consiguiente de la escritura en un archivo con datos incrustados (embebido) en muchas ocasiones es la mejor opción.
 - ▶ Probablemente, cada actualización como agregar elementos podrán añadir nuevos campos e incrementar el tamaño de un documento.
- 



Operadores

Operadores

- **\$gt**: mayor que (o greater than en inglés).
- **\$lt**: menor que (o lower than en inglés).
- **\$gte**: mayor o igual que (o greather or equal than en inglés).
- **\$lte**: menor o igual que (o lower or equal than en inglés).
- **\$not**: no (negación o not en inglés).
- **\$in**: en, para buscar dentro de un array.
- **\$nin**: no en, para buscar algo que no se encuentre en un determinado array (o not in en inglés).
- **\$ne**: no es igual a (o not equal to en inglés).

“

ObjectID

”






ObjectID

- ▶ Esta diseñado para garantizar la unicidad en entornos distribuidos.
- ▶ El campo se compone por 12 bytes.
- ▶ **Timestamp (4 bytes).**
- ▶ **Id Maquina (3 bytes).**
- ▶ **Id del proceso (PID, 2 bytes).**
- ▶ **Campo incremental (3 bytes)**



ObjectID

- ▶ Los primeros nueve bytes garantizan la unicidad.
 - ▶ Los tres últimos bytes, nos garantizan que cada segundo podemos insertar 2^{24} documentos con ID diferente.
 - ▶ Nos sirve para obtener la fecha de creación del documento.
- 

“

REFERENCES

”



- 
- 
- <http://docs.mongodb.org/manual/core/introduction/>
 - <http://docs.mongodb.org/manual/reference/limits/>
 - <http://www.genbetadev.com/bases-de-datos/mongodb-empezando-por-el-principio-insertando-datos>
 - <http://www.genbetadev.com/productos/bases-de-datos/mongodb>
 - <http://nosql-database.org/>
 - <http://es.wikipedia.org/wiki/NoSQL>
 - <http://mviera.io/blog/empezando-con-mongodb-part-ii/>