



## **Bases de Datos Masivas (11088)**

Slides tomadas del material del *Curso Captura y Almacenamiento de la Información* de la Maestría en Inteligencia de Datos Orientada al Big Data de la UNLP

# MongoDB

---

## *Principales características*

- Base de datos NOSQL orientada a documentos.
- Basada en hashing.
- No maneja la noción de esquemas.
- Carece de un lenguaje DDL.
- El formato de almacenamiento es BSON (Binary JSON).
- Ofrece drivers para la mayoría de los lenguajes.

## **BSON**

- Es un formato de codificación para documentos tipo JSON (aunque contiene extensiones propias).
- Fue diseñado con 3 objetivos en mente:
  - Liviano: trata de mantener el espacio ocupado al mínimo.
  - Recorrible: debe ser posible recorrerlo fácilmente. Esto es fundamental ya que el formato primario de representación para MongoDB.
  - Eficiente: la codificación/decodificación puede ser realizada muy rápidamente ya que se basa en tipos de datos de C.

# MongoDB

## *Un ejemplo simple de un documento*

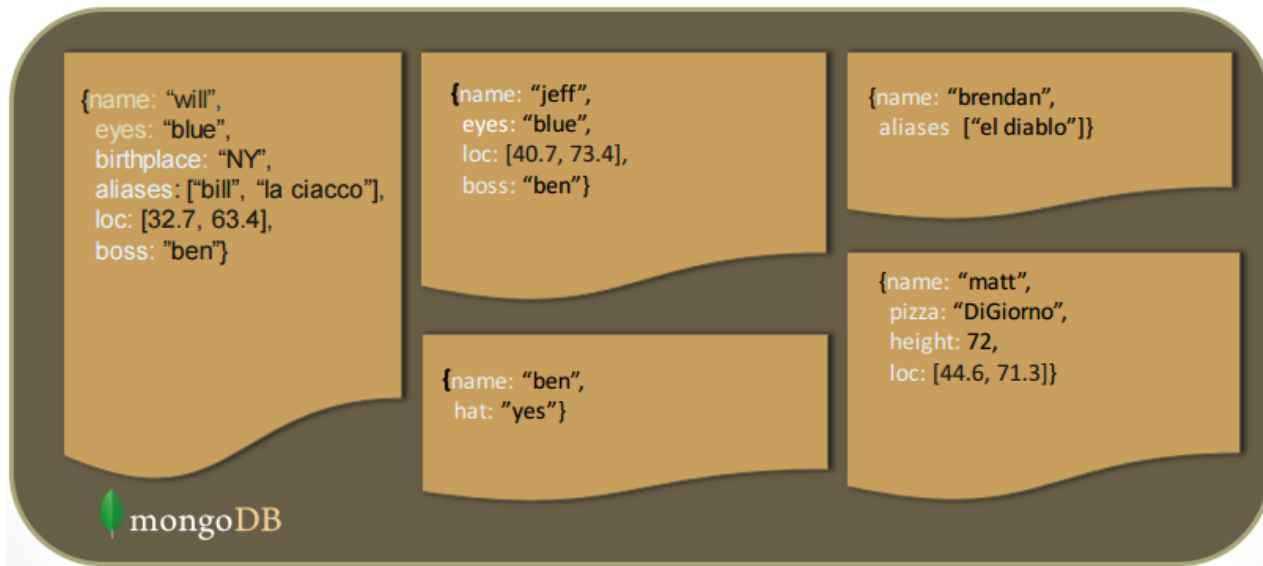
- Formato JSON

```
{  
  "day": [ 2010, 01, 23 ],  
  "products": {  
    "apple": { "price": 10 "quantity": 6 },  
    "kiwi": { "price": 20 "quantity": 2 }  
  },  
  "checkout": 100  
}
```

# MongoDB

## Carencia de esquemas

- Cada documento puede tener su propia “estructura”



# MongoDB

## *Analogías entre bases de datos relacionales y MongoDB*

- Principales conceptos

### **RDBMS**

Base de datos

Tabla

Fila

Columna

Índice

Join

Clave foránea

### **MongoDB**

Base de datos

Colección

Documento

Campo

Índice

Documento embebido

Referencia

# MongoDB

## Organización conceptual

- Una instancia de MongoDB tiene 0 o más bases de datos.
- Una base de datos tiene 0 o más colecciones.
- Una colección tiene 0 o más documentos.
- Un documento tiene 0 o más campos.



# MongoDB

---

## *MongoDB soporta las habituales operaciones C.R.U.D.*

- Create: inserta nuevos documentos
  - Read: permite recuperar documentos de una colección
  - Update: modifica un documento existente
  - Delete: elimina un documento
- 
- Estas operaciones se pueden realizar a través de la consola, de librerías o de herramientas como Robo3T.





# MongoDB

## CREATE

- Inserta nuevos documentos.
- Si la colección no existe, entonces la crea previamente.

- `db.collection.insertOne()`
- `db.collection.insertMany()`

```
db.users.insertOne(  ← collection
{
  name: "sue",        ← field: value
  age: 26,             ← field: value
  status: "pending"   ← field: value
}
)
```

# MongoDB

---

## READ

- Permite recuperar documentos.

- `db.collection.find()`

```
db.users.find(  
  { age: { $gt: 18 } },  
  { name: 1, address: 1 }  
) .limit(5)
```

← collection  
← query criteria  
← projection  
← cursor modifier

# MongoDB

## UPDATE

- Permite modificar documentos.
- Las modificaciones apuntan a una única colección.
- Cada documento se modifica atómicamente.

- `db.collection.updateOne()`
- `db.collection.updateMany()`
- `db.collection.replaceOne()`

```
db.users.updateMany(  
  { age: { $lt: 18 } },  
  { $set: { status: "reject" } }  
)
```

← collection  
← update filter  
← update action

# MongoDB

## DELETE

- Permite eliminar documentos.
- Las eliminaciones afectan a una única colección.
- Cada documento se modifica atómicamente.

- `db.collection.deleteOne()`

- `db.collection.deleteMany()`

```
db.users.deleteMany(  
  { status: "reject" }  
)
```

← collection

← delete filter

## Índices

- Los índices se definen a nivel de las “collections”.
- Se soportan índices por cualquier campo o subcampo del documento.
- MongoDB crea un índice sobre el campo `_id` durante la creación de una colección. Este índice controla la unicidad del valor del `_id`. No se puede eliminar este índice.
- Especificación
  - `db.collection.createIndex( <key and index type specification>, <options> )`
- Ejemplo
  - `db.collection.createIndex( { name: -1 } )`

# MongoDB

---

## *Tipos de índices*

- Single field
- Compound index
- Multikey index
- Geospatial index
- Text index
- Hashed index

# MongoDB

## *Tipos de índices*

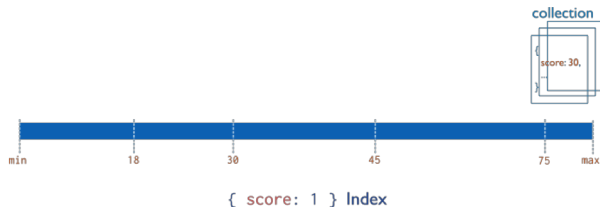
- Single field

- `db.collection.createIndex( <key and index type specification>, <options> )`

- `db.records.createIndex( { score: 1 } )`

- `db.records.createIndex( { "location.state": 1 } )`

```
{
  "_id": ObjectId("570c04a4ad233577f97dc459"),
  "score": 1034,
  "location": { state: "NY", city: "New York" }
}
```



# MongoDB

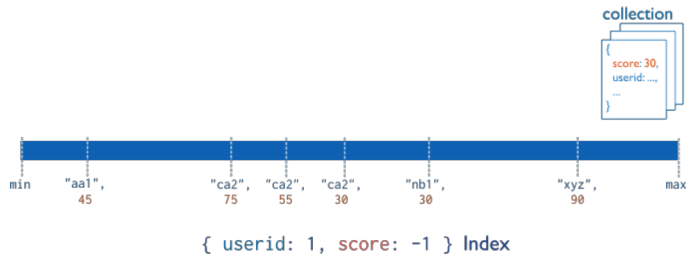
## *Tipos de índices*

- Compound index

- `db.collection.createIndex( { <field1>: <type>, <field2>: <type2>, ... } )`

- `db.products.createIndex( { "item": 1, "stock": 1 } )`

```
{
  "_id": ObjectId(...),
  "item": "Banana",
  "category": ["food", "produce", "grocery"],
  "location": "4th Street Store",
  "stock": 4,
  "type": "cases"
}
```





# MongoDB

## Tipos de índices

- Multikey index
  - `db.coll.createIndex( { <field>: < 1 or -1 > } )`



`{ "addr.zip": 1 } Index`

# MongoDB

## *Tipos de índices*

- GeoSpatial
- Utilizan datos tanto en formato de par de coordenadas como GeoJSON.
  - `db.collection.createIndex( { <location field> : "2dsphere" } )`
  - `db.places.createIndex( { loc : "2dsphere" } )`
  - `db.places.createIndex( { loc : "2dsphere" , category : -1, name: 1 } )`

```
db.places.insert(  
  {  
    loc : { type: "Point", coordinates: [ -73.88, 40.78 ] },  
    name: "La Guardia Airport",  
    category : "Airport"  
  }  
)
```

# MongoDB

---

## *Tipos de índices*

- Text
  - Una colección puede tener a lo sumo un índice de texto.
  - `db.reviews.createIndex( { comments: "text" } )`
  - `db.reviews.createIndex( { subject: "text", comments: "text" } )`
  - `db.collection.createIndex( { "$**": "text" } )`
  - `db.quotes.createIndex( { content : "text" }, { default_language: "spanish" } )`

## *Estrategias de creación de índices*

- Se deben crear ítems con 4 objetivos en mente:
  - Que sirvan para las consultas
    - Es mejor cuando un índice contiene todos los campos escaneados por la consulta.
  - Para ordenar los resultados
  - Que quepan en la memoria
    - `db.collection.totalIndexSize()`
  - Que aseguren selectividad

## Operaciones de agregación

- Procesan datos y devuelven resultados

