

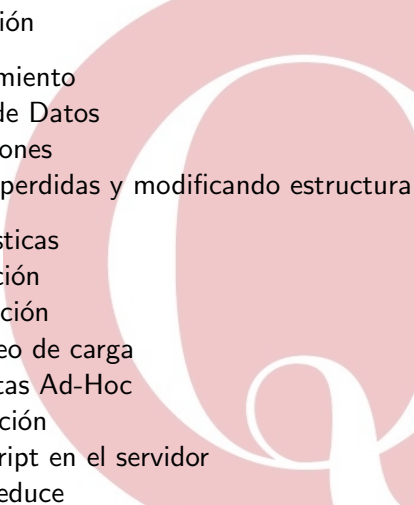


Estrategias de Persistencia

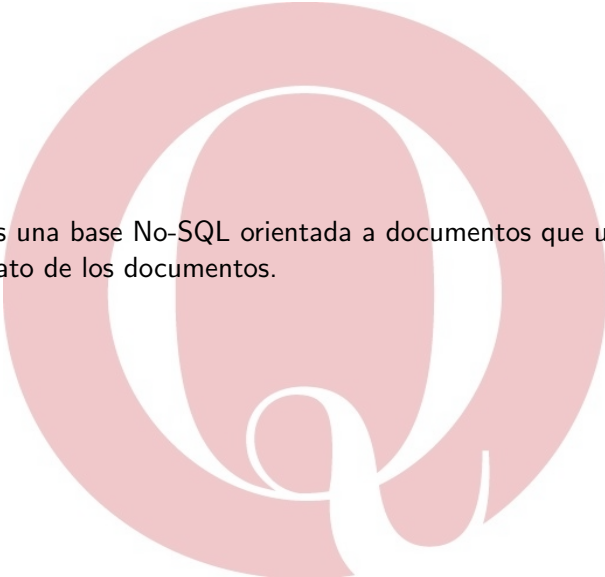
MongoDB

Universidad Nacional de Quilmes

Contenido


- 
- 1 Introducción
 - 2 Funcionamiento
 - Tipos de Datos
 - Colecciones
 - Claves perdidas y modificando estructura
 - 3 Características
 - Indexación
 - Replicación
 - Balanceo de carga
 - Consultas Ad-Hoc
 - Agregación
 - JavaScript en el servidor
 - Map Reduce

Introducción



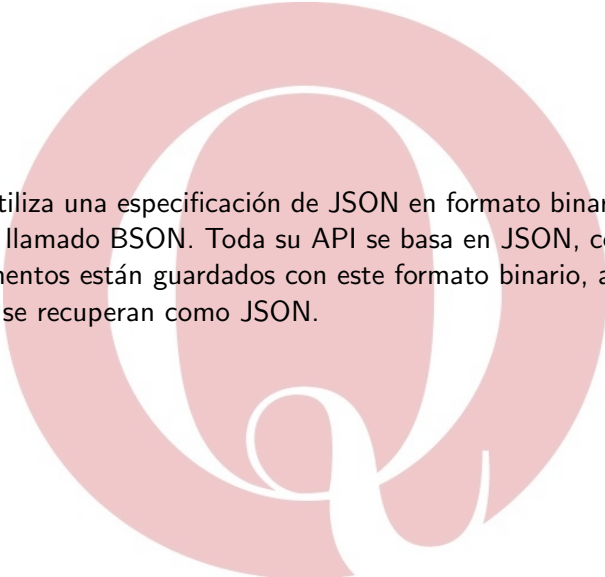
MongoDB es una base No-SQL orientada a documentos que utiliza JSON para el formato de los documentos.

Historia



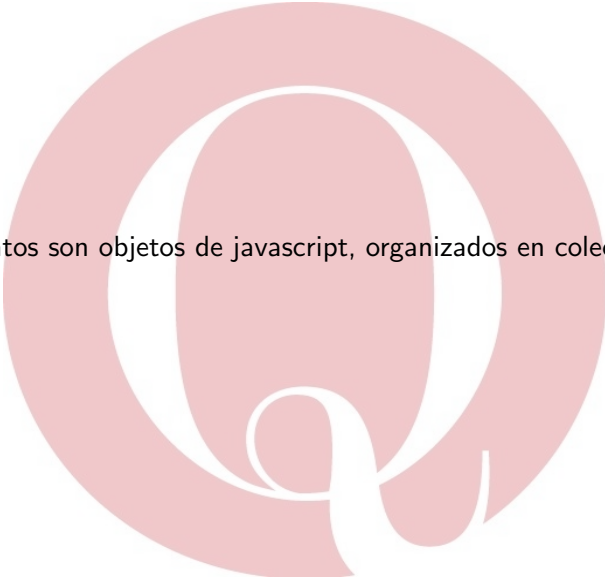
Desarrollada en 2007 por 10gen Inc. (Actualmente MongoDB Inc.). La compañía se encontraba desarrollando una PaaS, y eventualmente dándose cuenta del potencial de MongoDB, decide liberarlo como software libre bajo licencia AGPL. La compañía se dedica hoy exclusivamente al desarrollo y evolución de este producto.

JSON - BSON



MongoDB utiliza una especificación de JSON en formato binario (JSON es texto plano) llamado BSON. Toda su API se basa en JSON, con lo cual, si bien los elementos están guardados con este formato binario, al recuperarlos se recuperan como JSON.

Funcionamiento



Todos los datos son objetos de javascript, organizados en colecciones.

Outline

1 Introducción

2 Funcionamiento

- Tipos de Datos

- Colecciones

- Claves perdidas y modificando estructura

3 Características

- Indexación

- Replicación

- Balanceo de carga

- Consultas Ad-Hoc

- Agregación

- JavaScript en el servidor

- Map Reduce

Tipo de Datos

Básicos de JSON:

- Numeros
- Strings
- Boolean
- Objects
- Arrays
- Null
- Undefined

Adicionales:

- Object Id
- Date
- Timestamp
- Regex

Outline

1 Introducción

2 Funcionamiento

- Tipos de Datos
- **Colecciones**
- Claves perdidas y modificando estructura

3 Características

- Indexación
- Replicación
- Balanceo de carga
- Consultas Ad-Hoc
- Agregación
- JavaScript en el servidor
- Map Reduce

Colecciones

MongoDB organiza los datos en colecciones, una colección es un array de objetos. Las colecciones son el punto de entrada de los datos y es sobre las mismas donde se realizan las operaciones CRUD.

```
db.Personas.insert(  
  {  
    name: "John",  
    age: 21,  
    male: true,  
    groups: ["news", "sports"]  
  }  
)
```

Anidamiento

Los objetos pueden anidarse tanto como se quiera: `db.Personas.insert(`

```
{  
  name: "John",  
  age: 21,  
  male: true,  
  friends: [  
    {name: "Paul", age: 21, male: true},  
    {name: "Ringo", age: 25, male: true},  
    {name: "George", age: 23, male: true, friends: [  
      {name: "Ringo"}  
    ]}  
  ]  
}
```

Outline

1 Introducción

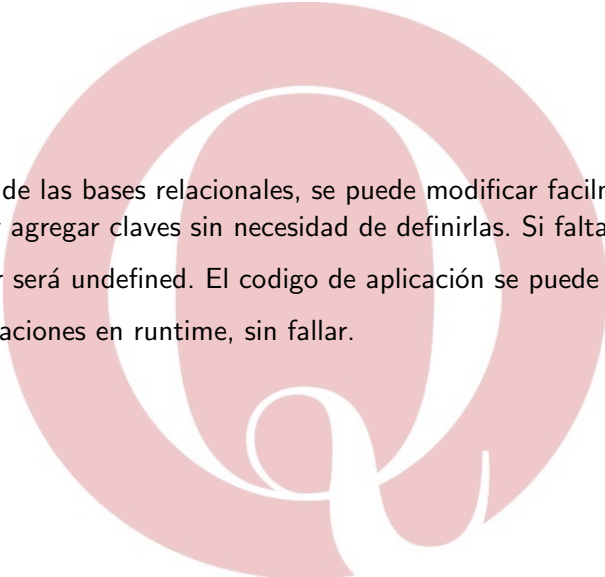
2 Funcionamiento

- Tipos de Datos
- Colecciones
- Claves perdidas y modificando estructura

3 Características

- Indexación
- Replicación
- Balanceo de carga
- Consultas Ad-Hoc
- Agregación
- JavaScript en el servidor
- Map Reduce

Claves perdidas y modificando estructura



A diferencia de las bases relacionales, se puede modificar fácilmente la estructura, y agregar claves sin necesidad de definirlas. Si faltan claves de algo, el valor será undefined. El código de aplicación se puede encargar de realizar migraciones en runtime, sin fallar.

Características

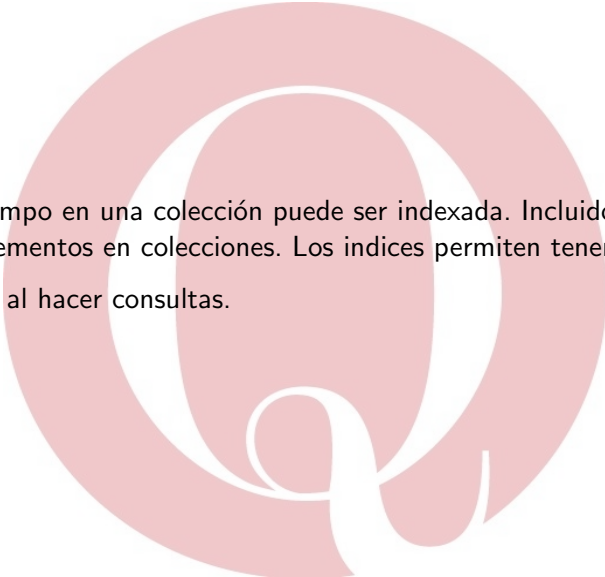
Dependiendo de que forma se almacene la información, nos podemos encontrar con distintas características en una base de datos NoSQL:

- Indexación
- Replicación
- Balanceo de carga
- Consultas Ad-Hoc
- Agregación
- Ejecución de JavaScript en el servidor

Outline

- 1 Introducción
- 2 Funcionamiento
 - Tipos de Datos
 - Colecciones
 - Claves perdidas y modificando estructura
- 3 Características**
 - Indexación**
 - Replicación
 - Balanceo de carga
 - Consultas Ad-Hoc
 - Agregación
 - JavaScript en el servidor
 - Map Reduce

Indexación

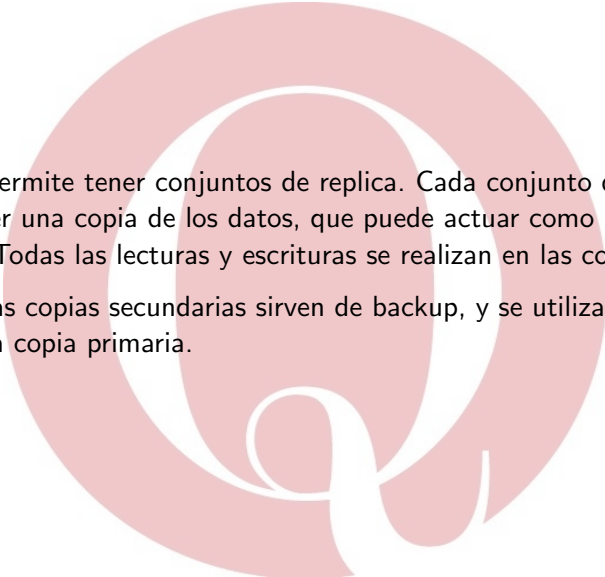


Cualquier campo en una colección puede ser indexada. Incluidos campos dentro de elementos en colecciones. Los índices permiten tener mejor performance al hacer consultas.

Outline

- 
- 1 Introducción
 - 2 Funcionamiento
 - Tipos de Datos
 - Colecciones
 - Claves perdidas y modificando estructura
 - 3 Características**
 - Indexación
 - Replicación**
 - Balanceo de carga
 - Consultas Ad-Hoc
 - Agregación
 - JavaScript en el servidor
 - Map Reduce

Replicación

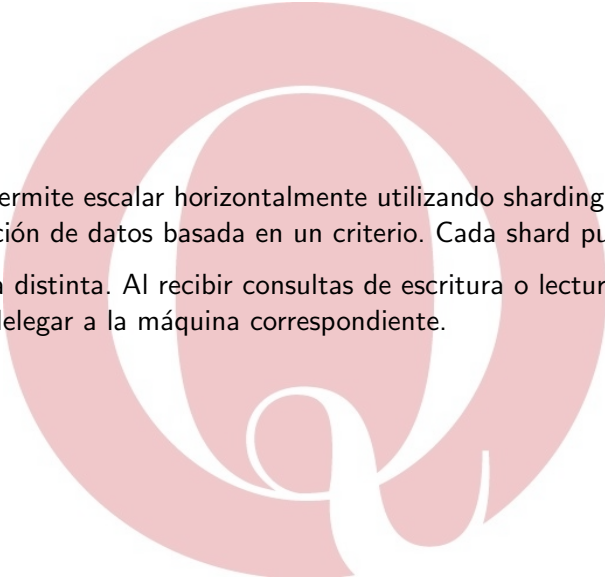


MongoDB permite tener conjuntos de replica. Cada conjunto de replica permite tener una copia de los datos, que puede actuar como primaria o secundaria. Todas las lecturas y escrituras se realizan en las copias primarias. Las copias secundarias sirven de backup, y se utilizan en caso de que falle una copia primaria.

Outline

- 
- 1 Introducción
 - 2 Funcionamiento
 - Tipos de Datos
 - Colecciones
 - Claves perdidas y modificando estructura
 - 3 Características**
 - Indexación
 - Replicación
 - Balanceo de carga**
 - Consultas Ad-Hoc
 - Agregación
 - JavaScript en el servidor
 - Map Reduce

Balanceo de carga



MongoDB permite escalar horizontalmente utilizando sharding. Cada shard es una partición de datos basada en un criterio. Cada shard puede estar en una máquina distinta. Al recibir consultas de escritura o lectura, Mongo se encarga de delegar a la máquina correspondiente.

Outline

1 Introducción

2 Funcionamiento

- Tipos de Datos
- Colecciones
- Claves perdidas y modificando estructura

3 Características

- Indexación
- Replicación
- Balanceo de carga
- **Consultas Ad-Hoc**
- Agregación
- JavaScript en el servidor
- Map Reduce

Consultas Ad-Hoc

MongoDB permite realizar consultas sobre los datos a través de las colecciones:

```
db.users.find()  
db.users.find({ status: "A" })  
db.users.find({ status: { $ne: "A" } })  
db.users.find({ status: "A", age: { $gt: 25, $lte: 50 } })  
db.users.find({ status: "A" }).sort({ user_id: -1 })  
db.users.find({ status: "A" }).limit(5).skip(10)
```

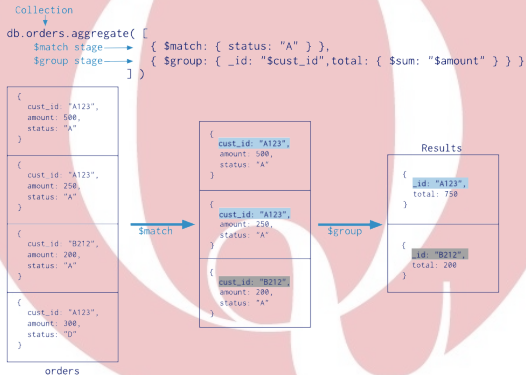
Outline

- 
- 1 Introducción
 - 2 Funcionamiento
 - Tipos de Datos
 - Colecciones
 - Claves perdidas y modificando estructura
 - 3 Características**
 - Indexación
 - Replicación
 - Balanceo de carga
 - Consultas Ad-Hoc
 - Agregación**
 - JavaScript en el servidor
 - Map Reduce

Agregación

También se puede hacer agregación, como hacer sumas de campos o sacar promedios. `db.users.distinct("status")`

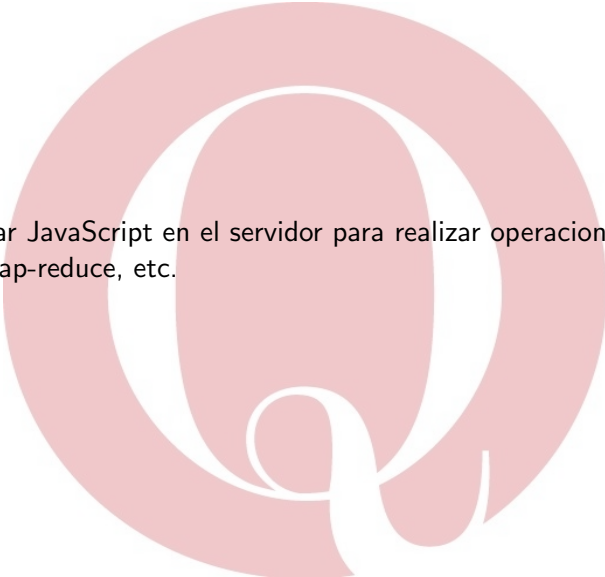
`db.users.count()`



Outline

- 
- 1 Introducción
 - 2 Funcionamiento
 - Tipos de Datos
 - Colecciones
 - Claves perdidas y modificando estructura
 - 3 Características**
 - Indexación
 - Replicación
 - Balanceo de carga
 - Consultas Ad-Hoc
 - Agregación
 - JavaScript en el servidor**
 - Map Reduce

JavaScript en el servidor



Se puede usar JavaScript en el servidor para realizar operaciones de agregado, map-reduce, etc.

Outline

- 
- 1 Introducción
 - 2 Funcionamiento
 - Tipos de Datos
 - Colecciones
 - Claves perdidas y modificando estructura
 - 3 Características**
 - Indexación
 - Replicación
 - Balanceo de carga
 - Consultas Ad-Hoc
 - Agregación
 - JavaScript en el servidor
 - Map Reduce**

Map Reduce

Map-reduce es una técnica que permite procesar y generar grandes volúmenes de datos. El formato del algoritmo permite ejecutarse de forma distribuida y paralelizada, lo cual permite obtener una buena performance al realizar este tipo de consultas. El nombre de map y reduce proviene de las funciones características de funcional sobre colecciones, aunque no se corresponden exactamente con las mismas. Pseudocódigo:

$$\text{map}(k1, v1) = \{k2, v2\}$$
$$\text{reduce}(k2, \text{list}(v2)) = \text{list}(v3)$$

Map Reduce

El algoritmo consiste en tres pasos.

- Map: Cada nodo aplica la función `map()` sobre cada elemento de los datos locales y guarda el resultado de forma temporal.
- Shuffle: Cada nodo redistribuye los datos de forma tal que los datos que produjeron una misma key sean agrupados.
- Reduce: Cada nodo procesa ahora el producto de shuffle de forma paralela para producir el resultado esperado.

Map Reduce

MongoDB soporta map-reduce usando javascript y algunas funciones built-in.

