

MongoDB

Mikel Egaña Aranguren

mikel-egana-aranguren.github.io

mikel.egana@ehu.eus



MongoDB

<https://github.com/mikel-egana-aranguren/ABD>



MongoDB

<https://www.mongodb.com/>

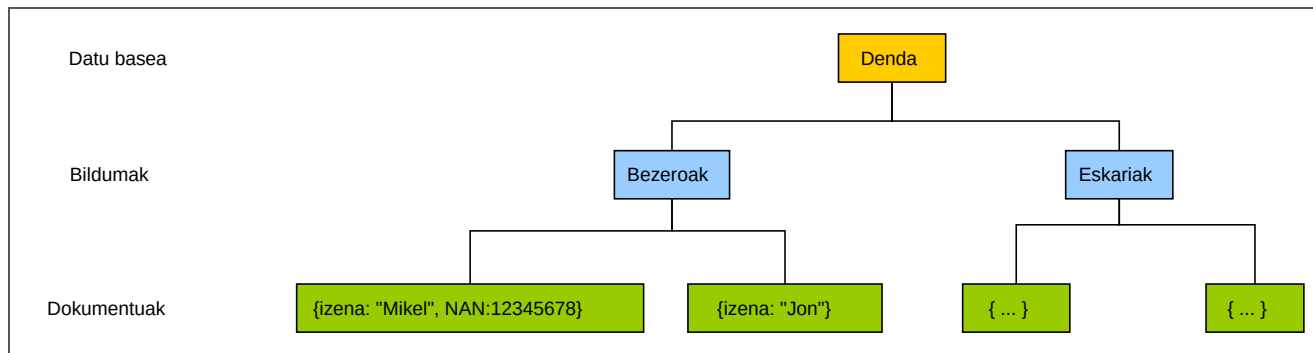
Dokumentuetan oinarritutako DBKS

Bere izena Humongous (“erraldoi”) hitzetik dator

MongoDB

- MongoDB Community Edition: bertsio librea
- MongoDB Enterprise: bertsio komertziala
- Atlas: hodei zerbitzua
- Tresna osagarriak: Adib. Compass (GUI)

Datuen egitura



Datuen egitura

Dokumentu bakoitza [JSON](#) (JavaScript Object Notation) formatuan gordetzen da: barnean [Binary JSON](#) (BSON) formatuan

Datu bakoitza gako:balio moduan gordetzen da

Balio bat beste dokumentu bat izan daiteke

Balio bat beste dokumentuen zerrenda bat izan daiteke

Datuen egitura

```
{  
  izena: "Unai",  
  NAN: "12345678A",  
  helbidea:  
  {  
    hiria: "Bilbao"  
  }  
  telefonoak: [  
    { bulegoa: 946014458 },  
    { mugikorra: 612345678 }  
  ]  
}
```

Balio bat beste dokumentu bat izan daiteke

Balio bat beste dokumentuen zerrenda bat izan daiteke

Datuen egitura

Sistema erlazionalarekiko antzekotasuna:

- Taula - Bilduma
- Lerroa - Dokumentua
- Zelaia - Gakoa

Datuen egitura

Ezberdintasun nagusiak:

- Bilduma bateko dokumentu guztiek ez dituzte gako berdinak eduki behar
- Ez da beharrezkoa dokumentuen arteko erlazioak adieraztea

Komandoak

- Pantaila garbitu:
> `cls`
- Sortutako DBak zerrendatu:
> `show dbs`

Komandoak

- Erabilpenean dagoen DBa aukeratu:

```
> db
```

- DB-ez aldatu:

```
> use DB-izena
```

DB eta bildumak sortu

```
> use denda  
> db.bezeroak.insertOne({ "izena": "Unai" })
```

Bildumaren
izena

Dokumentu bat
sartzeko funtzioa

Dokumentu bat
bezero bat datuekin

DB eta bildumak sortu

Dokumentu bakoitzak identifikadore bakarra du (Sortzen denean automatikoki esleitzen da, baina eskuz adierazi daiteke)

```
{  
  acknowledged: true,  
  insertedId: ObjectId("626169bbae7b857469c4514b")  
}
```

DB eta bildumak sortu

Dokumentua anitzak bilduma batean sartu:

```
> db.[bilduma].insertMany( [array-docs] )
```

Komandoak

- Dokumentu guztiak erakutsi:

```
> db.[bilduma].find()
```
- Patroi batekin bat egiten dituzten dokumentuak erakutsi:

```
> db.[bilduma].find([patroia])
```
- Patroi batekin bat egiten duen lehen dokumentu erakutsi:

```
> db.[bilduma].findOne([patroia])
```

Komandoak

- “unai” balioa “izena” gakoan duten dokumentuak erakutsi:

```
> db.bezeroak.find( { izena: "Unai" } )
```
- “unai” izena eta 12345 NAN zenbakia duten dokumentuak erakutsi:

```
> db.bezeroak.find( { izena: "Unai", NAN: 12345 } )
```


Komandoak

- Konparaketa operazioak (eq,ne,gt,gte,lt,lte):

```
> db.bezeroak.find( { NAN: { $gt: 30000 } } )
```

- Espresio erregularrak:

```
> db.bezeroak.find( { izena: /M./ } )
```

Komandoak

- Erakusteko zelaia:

```
{ erakustekoZelaia: 1, ezErakustekoZelaia: 0}
```

- “bezeroak” bildumako dokumentuen “NAN” zelaia erakutsi:

```
> db.bezeroak.find( {}, {NAN: 1, _id: 0})
```

Komandoak

- Patroi batekin bat egiten duen lehen dokumentua ezabatu:

```
> db.[bilduma].deleteOne([patroia])
```
- Patroi batekin bat egiten duten dokumentu guztiak ezabatu:

```
> db.[bilduma].deleteMany([patroia])
```
- Bilduma bateko dokumentu guztiak ezabatu:

```
> db.[bilduma].deleteMany( {} )
```

Komandoak

- Bilduma bateko dokumentuak aldatu:

```
> db.[bilduma].updateMany([patroia], {$set:  
[aldaketak]}))
```
- 22233 NAN zenbakia duten bezeroen izena aldatu, "Nagore" jarri:

```
> db.bezeroak.updateMany({ NAN: 22233 }, {$set:{  
izena: "Nagore" } })
```

Komandoak

- Dokumentu bat beste batekin ordezkatu:

```
> db.[bılduma].replaceOne([patroia], [dok-berria] )
```

Komandoak

- DB bateko bildumak erakutsi:

```
> show collections
```

- Bilduma bat ezabatu:

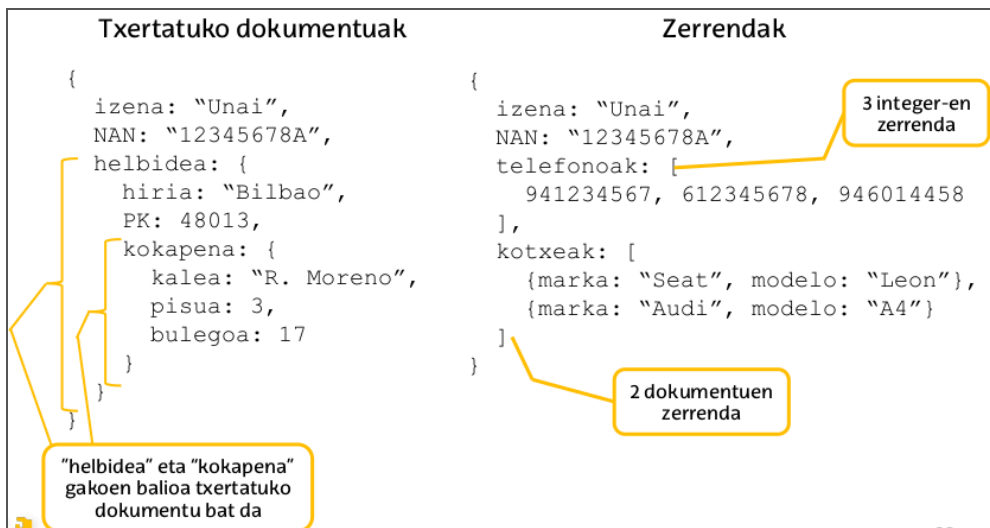
```
> db.[bilduma].drop()
```

- DB bat ezabatu:

```
> db.dropDatabase()
```

Erlazioak

Dokumentu batek txertatuko dokumentuak edo zerrendak eduki ditzazke



Erlazioak

Normalean:

- Bilduma anitz, bakoitzak entitate bat adieraziko du: “bezeroak”, “produktuak” eta “eskariak”
- Erlazioak entitate desberdineko datuen artean

MongoDB-n ez dago teknika konkreturik bildumen arteko erlazioak adierazteko: Eskuz sortu beharko ditugu

Erlazioak

Txertatuako dokumentuak

Zelai batzuk erabili erreferentzia bezala

Txertatuako dokumentuak

"bezeroak" bilduma

```
{  
  izena: "Unai",  
  NAN: "12345678A",  
  helbidea:  
  {  
    hiria: "Bilbao",  
    PK: 48013,  
    ...  
  }  
}
```

Txertatuako dokumentuak

Errepikatzen ez diren datuentzako proposa

(+) Datuak logikoki multzokatzen dira

(-) Bikoiztasunak sortu daitezke

Erreferentziak

"bezeroak" bilduma

```
{  
  izena: "Unai",  
  NAN: "12345678A",  
  helbidea: "001",  
  ...  
}
```

"helbideak" bilduma

```
{  
  _id: "001",  
  hiria: "Bilbao",  
  PK: 48013,  
  ...  
}
```



Erreferentziak

Aproposa hainbat bildumetan erreferenziatzen diren datuentzat

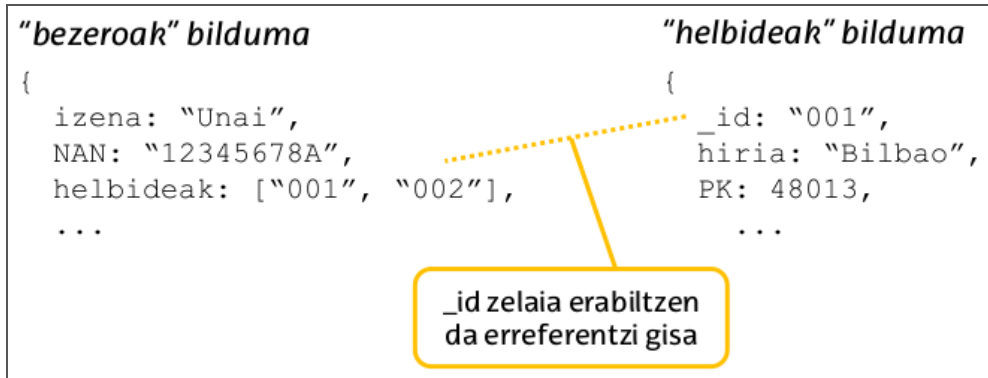
(+) Ez dago datuen bikoiztasunik

(-) Kudeaketa konplexuagoa

(-) Agregazio operazioak erabili behar dira erlazionatutako datuak lortzeko

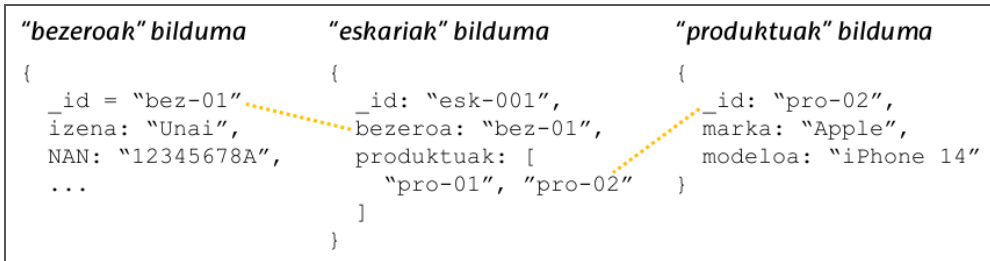
Erreferentziak

1 - N datuen arteko erlazioa: bezero batek hainbat helbide izan ditzake, helbide bat soilik bezero batena da



Erreferentziak

N - N datuen arteko erlazioa: bezero batek hainbat produktu erosi ditzake,
produktu bat hainbat bezerok erosi dezakete



Erreferentziak edo txertatutako dokumentuak

Txertatutako dokumentuak:

- Erlazionatuta baina bikoiztu gabe dauden datuak
- 1 - 1 erlazioak: bezero bati soilik helbide bat esleitzen zaio
- Bikoiztasunik ez duten 1 - N erlazioak

Erreferentziak edo txertatutako dokumentuak

Erreferentziak

- Entitate desberdinen datuak, baina haien artean erlazionatuta daudenak
- 1 - N erlazioak
- N - N erlazioak

Erlazioak

Bilduma desberdinetako datuak konbinatu daitezke agregazio operazioarekin ("Join")

"eskariak" bilduma

```
{ _id: "esk-001",  
  bezeroa: "cli-01",  
  produktuak: ["pro-01"]  
}
```

"bezeroak" bilduma

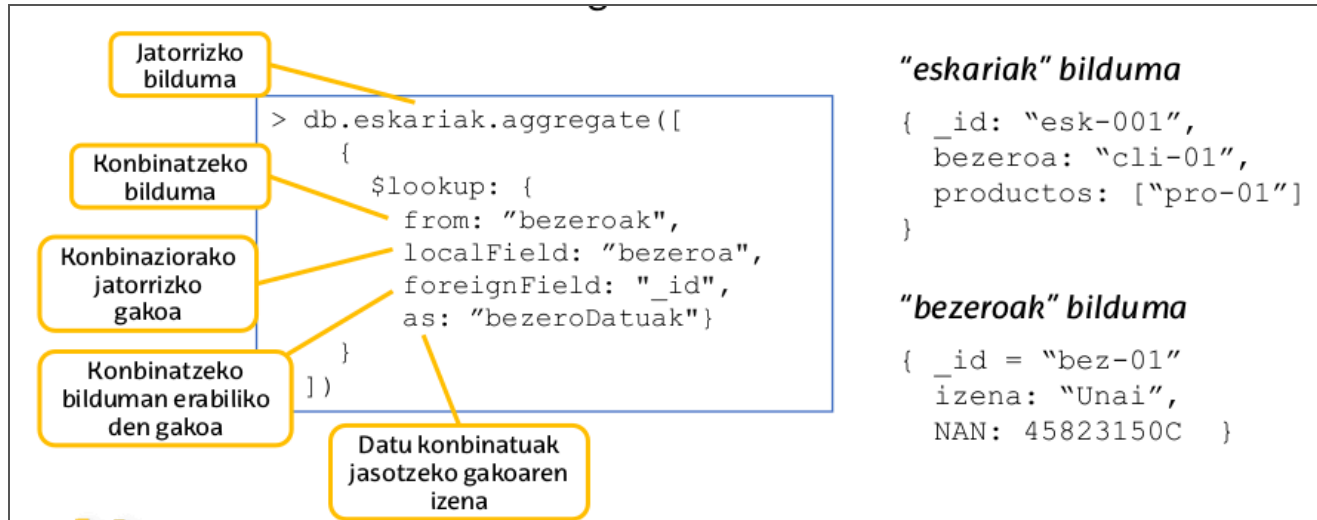
```
{ _id = "bez-01"  
  izena: "Unai",  
  NAN: 45823150C }  
}
```

Konbinazioa

```
{ _id: "ped-001",  
  bezeroa: "bez-01",  
  produktuak: ["pro-01"]  
  bezeroDatuak: {  
    _id = "bez-01"  
    izena: "Unai",  
    NAN: 45823150C  
  }  
}
```

\$lookup bidez bildumak konbinatu

Adibidea: Eskarien eta eskariak egin dituzten bezeroen datuak konbinatzen duen datu egitura bat sortu



Eskemak

MongoDB-n datuak kudeatu daitezke inolako datu egitura ezarri gabe

Baina egoera batzuetan datuen kontrol automatiko bat ezartzea aproposa izan daiteke: Adibidez, “produktuak” bilduma bateko dokumentu guztiek zenbaki den “prezioa” gako bat edukitzeko

Eskemak (M. Schwarzmüller)

Kaos totala			Erdibidea			SQL era		
Bilduma: Produktuak			Bilduma: Produktuak			Bilduma: Produktuak		
{ izena: "Liburua", prezioa: 5.99 }			{ izena: "Liburua", prezioa: 5.99 }			{ izena: "Liburua", prezioa: 5.99 }		
{ titulua: "Botila", eskuragarri: true }			{ izena: "Botila", prezioa: 2.55, eskuragarri: true }			{ izena: "Botila", prezioa: 2.55 }		

Eskemak

Bilduma bateko dokumentuek bete behar duten egitura definitu daiteke

Eskema batek bilduman sortu nahi den dokumentu bakoitza egiaztatzen du

Eskemak

validationLevel: Controla cómo de estricta es la validación

- strict (valor por defecto): Egiaztapena zein zorrotza den kontrolatzen du
- moderate: Bilduman dauden dokumentuei egiten diren aldaketak ez dira kontrolatzen

Eskemak

validationAction: Dokumentu batek eskema betetzen ez duenean zer egin adierazten du

- error (onetsia): Errore bat igortzen da eta ez da bilduman sartzen
- warn: Dokumentua bilduman sartzen da eta abisu bat idazten da MongoDB-ko log fitxategian

Eskemak

```
> db.createCollection("markak", {  
  validator: {  
    $jsonSchema: {  
      bsonType: "object",  
      required: [ "izena", "aurrekontua" ],  
      properties: {  
        izena: { bsonType: "string", description: "markaren izena"},  
        aurrekontua: { bsonType: "int", minimum: 2000 }},  
      },  
    },  
    validationLevel: 'moderate',  
    validationAction: 'error'  
  })
```

Bildumaren
zelai zerrenda

Hautazkoak

Eskemak

Bilduma baten eskema erakutsi:

```
> db.getCollectionInfos({name: "markak"})
```

Bilduma baten eskema aldatu:

```
> db.runCommand( {collMod: "markak",  
  validator: {  
    $jsonSchema: {  
      ...
```

Eskemak

validationAction “warn” bada, abisua MongoDB-ko log fitxategian idazten da
(Onetsia /var/log/mongodb/mongod.log)

Eskemak: data types

Izena	Deskribapena	Adibidea
string	Testu arrunta	"Unai"
Boolean	Balio boolearra	true
int	Zenbaki osoa (int32)	55
NumberLong	Zenbaki oso luzea (int64)	1000000000
NumberDecimal	Zenbaki dezimala	12.99
ObjectId	Identifikadore bakarra	ObjectId("74121...")
ISODate	Data UUUU-HH-EE formatuan	2022-01-09
Timestamp	Data Unix formatuan	11348822
EmbeddedDocument	Txertatutako dokumentua	{ "a": { ... } }
Array	Zerrenda	{ "b": [...] }