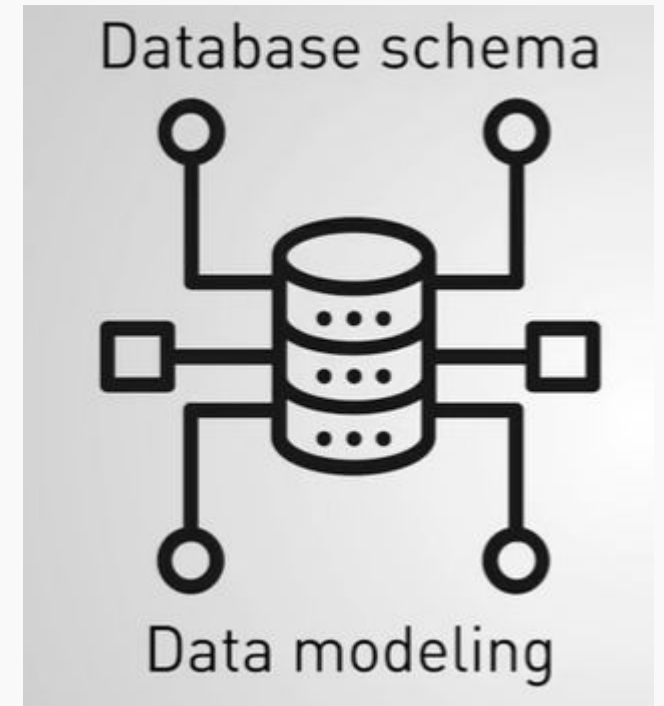




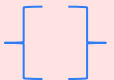
MODELADO EN MONGODB



REGLA

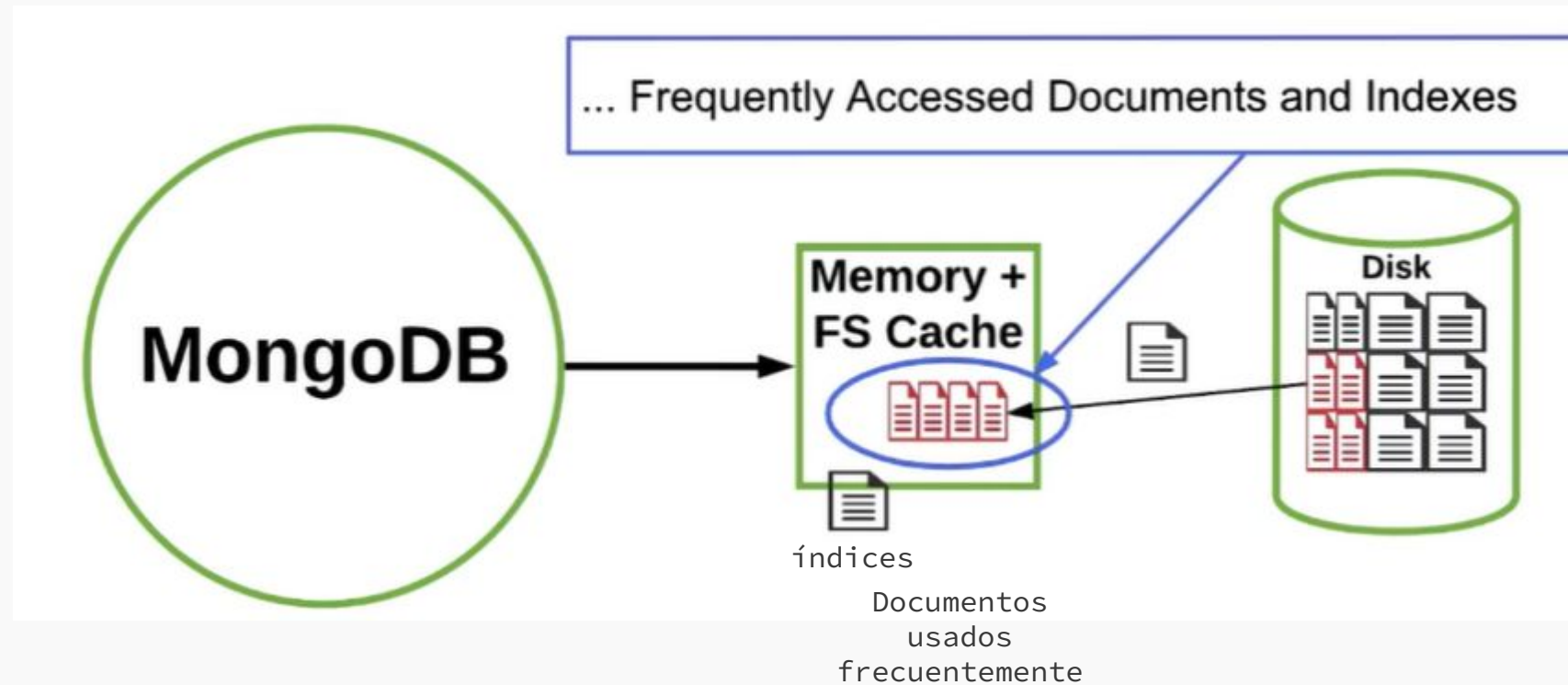
{ } Los datos se almacenan cómo serán usados

Si la información se requiere consultar junta,
entonces se debe almacenar junta.



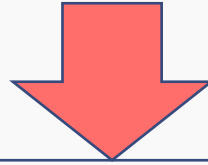
WORKING SET

El conjunto de los datos que las aplicaciones utilizan en el curso normal de su operación.



METODOLOGÍA

1. Describir el workload (cantidad de datos, lecturas y escrituras importantes)
2. Identificar las relaciones (1:1, 1:N, N:M, embebido o referenciado)
3. Aplicar patrones de diseño



SCHEMA

Colecciones

Consultas

Suposiciones

Campos

Índices

Modeling for Simplicity

Simplicity

Performance



- A. Most frequent queries
- B. Mostly embed entities in a single collection



small team
small project



Pocas colecciones
Documentos con mucha información
Documentos embebidos

Modeling for Performance

Simplicity

Performance



Gran cantidad de operaciones
Operaciones de lectura y escritura
Documentos embebidos y referenciados

large team
lots of reads and/or writes
high resources usage

- A. Data Size
Quantify Ops
Qualify Ops
- B. Embed & Link
- C. Patterns A,B,C



A Mix of Simplicity and Performance

Simplicity

Performance



- A. Data Size
Quantify Ops
- B. Embed & Link
- C. Patterns A



1) DEFINIR EL WORKLOAD

Detalles de una operación

Actor	
Descripción	
Tipo de operación (R/W)	
Datos en la operación (campos)	
Frecuencia	
Tamaño de los datos	
Vida útil de los datos	

EJEMPLO DE OPERACIÓN DE ESCRITURA

Actor:	sensor device
Description:	send weather data to the server
Operation Type:	write (insert new information)
Data in Operation:	device ID, time stamp, device metrics
Frequency:	1.6 M/sec (100 000 000/60)
Data Size:	1000 Bytes
Data Life:	10 years
Data Durability:	one node, no need to wait for majority

EJEMPLO OPERACIÓN DE LECTURA

Actor:	Data Scientist
Description:	run analytic/trend query on temperature
Operation Type:	read
Data in Operation:	temperature metrics
Frequency:	100/hour (10 scientists * 10 op/hr)
Read Pattern:	collection scan
Data Freshness:	up to the last hour

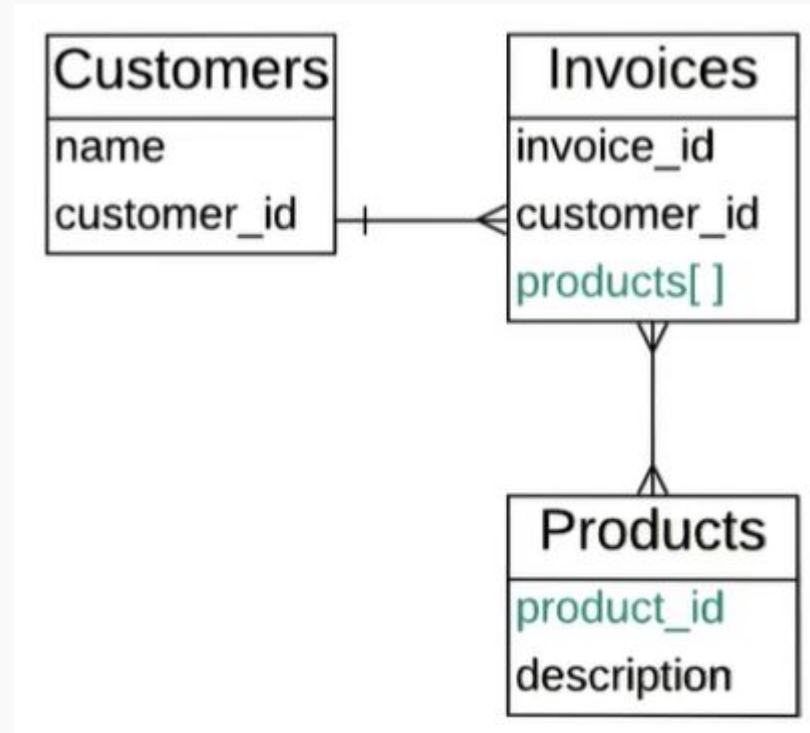
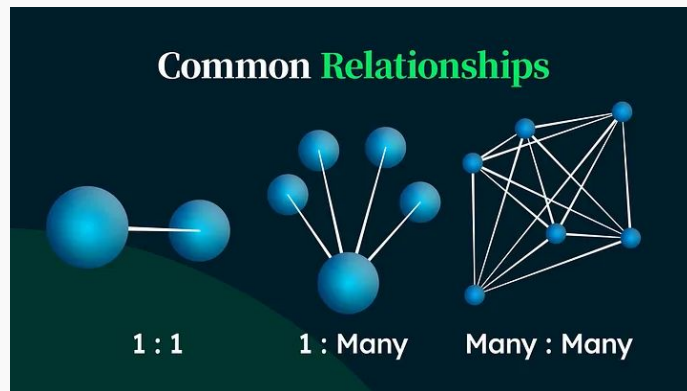
Flexible Methodology

<i>Goal</i>	<i>Simplicity</i>	<i>Simplicity & Performance</i>	<i>Performance</i>
1. Identify Workload	Most Frequent Op	Data Size Quantify Ops	Data Size Quantify Ops Qualify Ops
2. Entities and Relationships	Mostly Embed	Embed & Link	Embed & Link
3. Transformation Patterns	Pattern A	Patterns A, B	Patterns A,B,C

2) IDENTIFICAR LAS RELACIONES

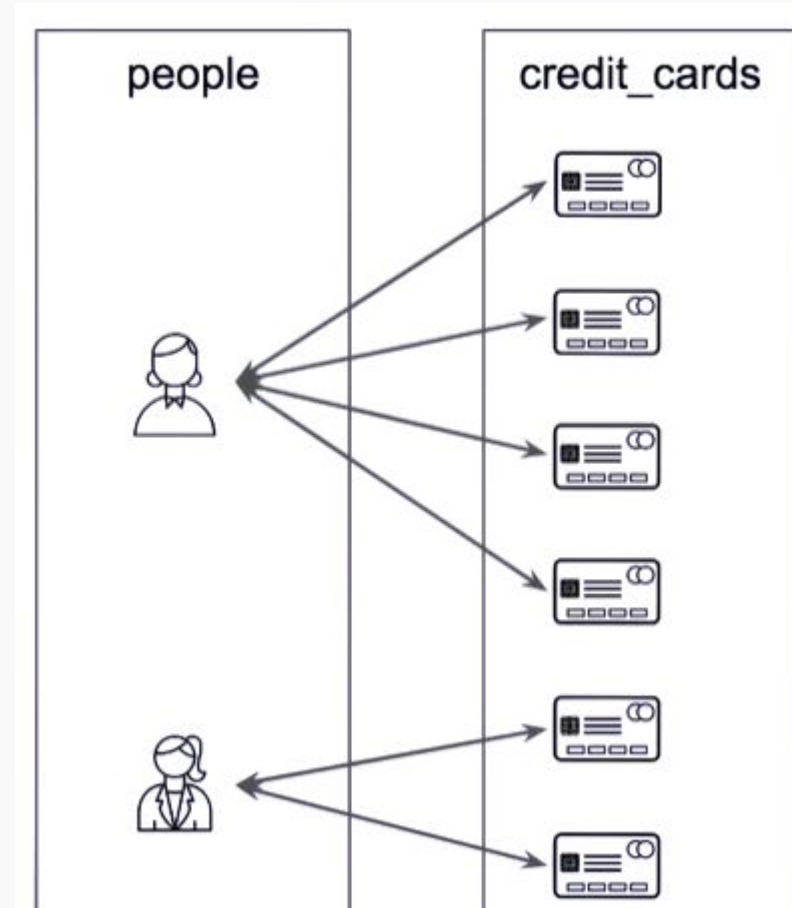
Tipos

- (1:1) Uno a uno
- (1:N) Uno a muchos
- (N:M) Muchos a muchos



UNO A MUCHOS

- Un cliente tiene muchas tarjetas de crédito



RELACIONES UNO A MUCHOS

- Una madre tiene varios hijos
Cuántos? (0,1,2... 80)? Por lo general 2.

Documentos de los hijos embebidos en la madre



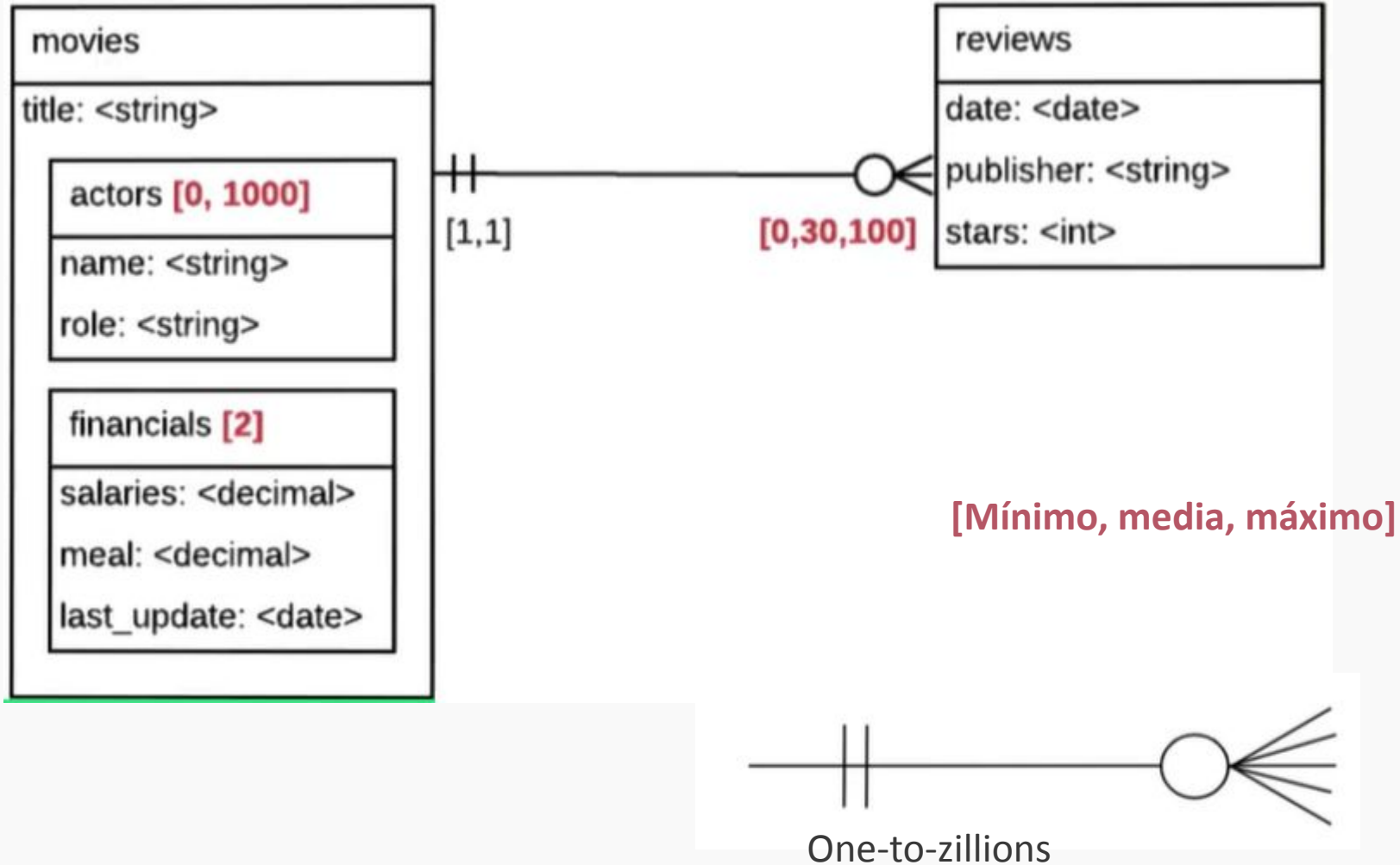
- Una persona tiene muchos seguidores.
Cuántos? Muchos 0,1...20...1'000.. Varios millones

Documentos referenciados



ESTRATEGIA DIFERENTE

NOTACIÓN MODELAMIENTO



UNO A UNO



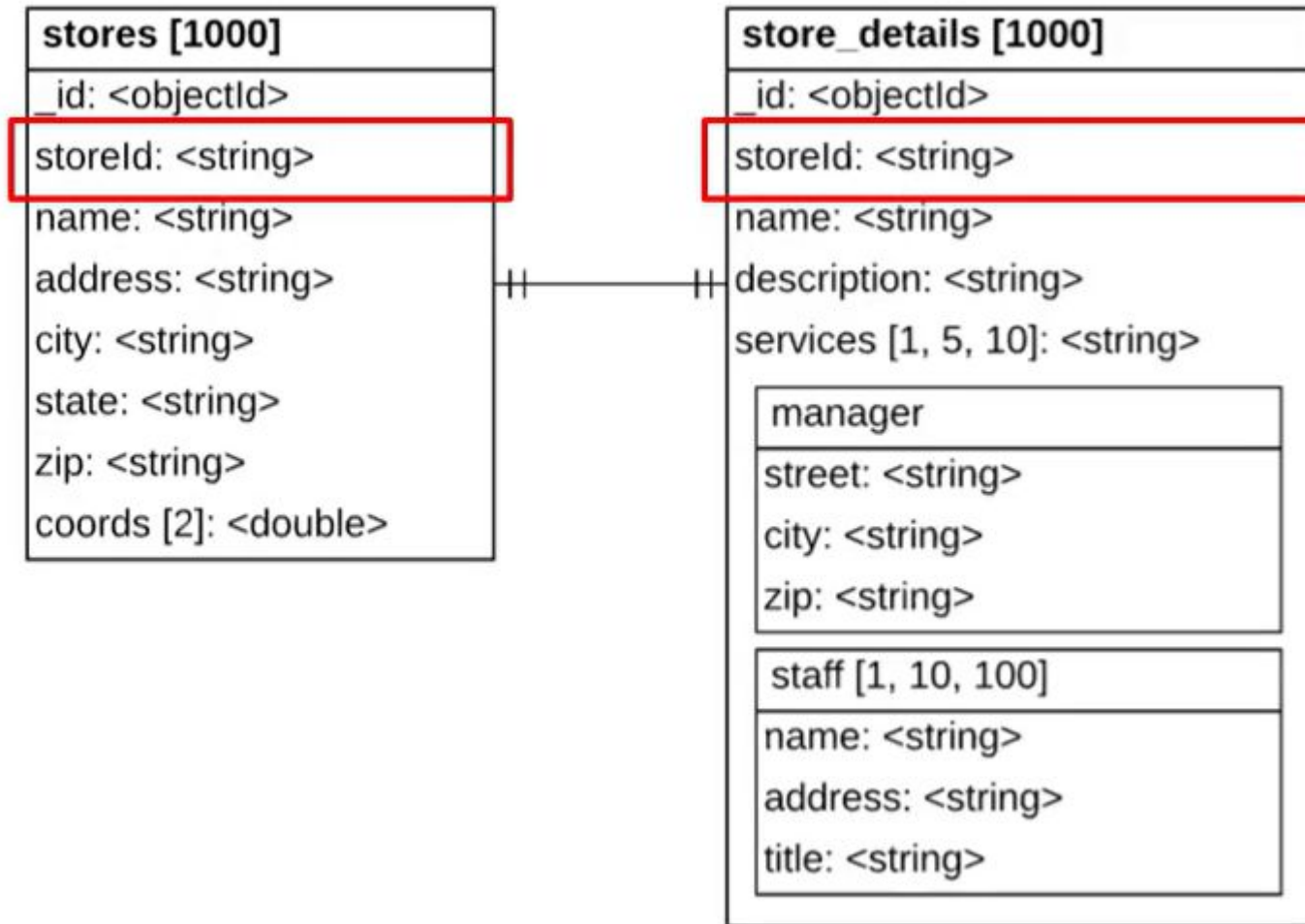
Los datos por lo general van en la misma entidad o documento.

UNO A UNO: EMBEBIDO

users [10M]
<div><div>_id: <objectId></div><div>name: <string></div><div><div>address</div><div><div>street: <string></div><div>city: <string></div><div>zip: <string></div></div></div><div><div>shipping_address</div><div><div>street: <string></div><div>city: <string></div><div>zip: <string></div></div></div></div>

- Modelarlo como documentos independientes, preserva la simplicidad.
- Los documentos son más claros.

UNO A UNO: REFERENCIADO

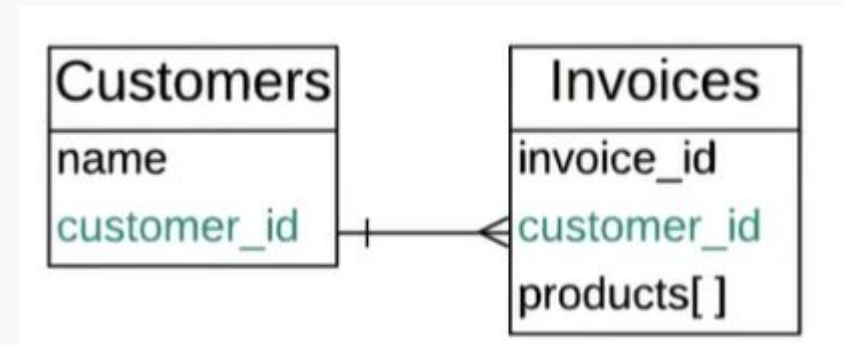


- Adiciona complejidad.
- Posibles mejoras en:
 - Discos pequeños
 - Poca cantidad de RAM

UNO A UNO

- ¿Embebido o referenciado?
- Preferible escoger embebidos a referenciados por simplicidad.
- Uso de subdocumentos para organizar campos.
- Uso de referencias para propósitos de optimización.

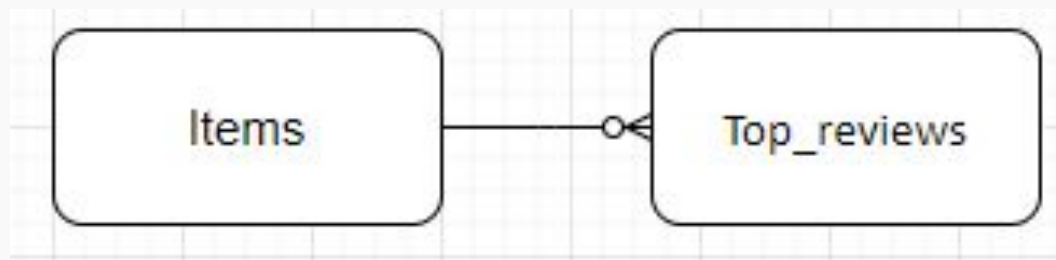
UNO A MUCHOS



Puede el documento estar embebido o referenciado.

UNO A MUCHOS: EMBEBIDO – EN EL UNO

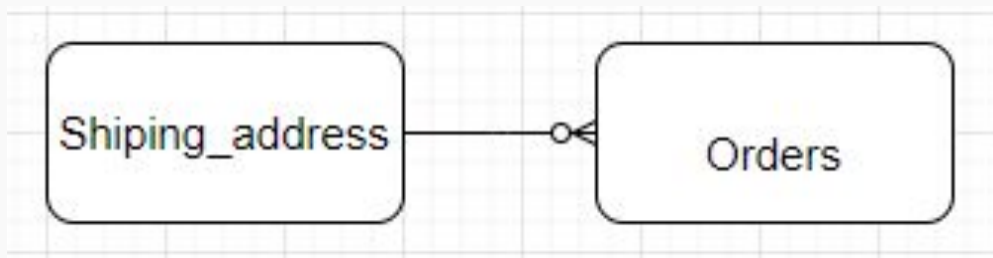
- El documento del lado del muchos, queda embebido.
- Utilizado en aplicaciones simples, o pocos documentos embebidos.
- Se procesa el objeto principal junto con los documentos relacionados.
- Se realizaría indexación sobre el array



items [100K]					
<code>_id: <int></code>					
<code>title: <string></code>					
<code>slogan: <string></code>					
<code>description: <string></code>					
<code>stars: <int></code>					
<code>category: <string></code>					
<code>img_url: <string></code>					
<code>price: <decimal></code>					
<code>sold_at [1, 1000]: <objectId></code>					
<table border="1"><thead><tr><th>top_reviews [0, 20]</th></tr></thead><tbody><tr><td><code>user_id: <string></code></td></tr><tr><td><code>user_name: <string></code></td></tr><tr><td><code>date: <date></code></td></tr><tr><td><code>body: <string></code></td></tr></tbody></table>	top_reviews [0, 20]	<code>user_id: <string></code>	<code>user_name: <string></code>	<code>date: <date></code>	<code>body: <string></code>
top_reviews [0, 20]					
<code>user_id: <string></code>					
<code>user_name: <string></code>					
<code>date: <date></code>					
<code>body: <string></code>					

UNO A MUCHOS: EMBEBIDO – EN EL MUCHOS

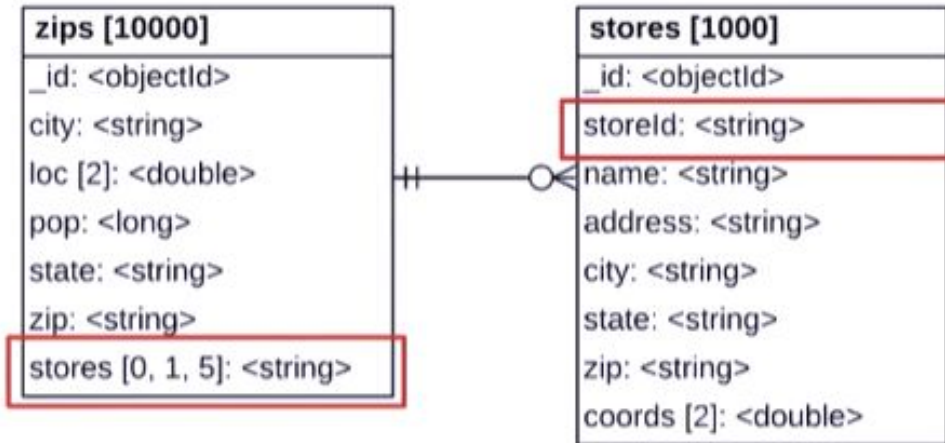
- Menos usado
- Útil cuando es más a menudo consultado el lado del muchos.
- El documento embebido se duplica en cada documento (la aplicación debe garantizar consistencia si se requiere).



orders [10M]	
_id: <objectId>	{
date: <date>	date: "2010/06/24",
customer_id: <int>	address: {
	"300 University",
	"Palo Alto, CA, USA",
	}
	}
	{
	date: "2019/06/24",
	address: {
	"100 Forest",
	"Palo Alto, CA, USA",
	}
	}

UNO A MUCHOS: REFERENCIADO – EN EL UNO

{ }



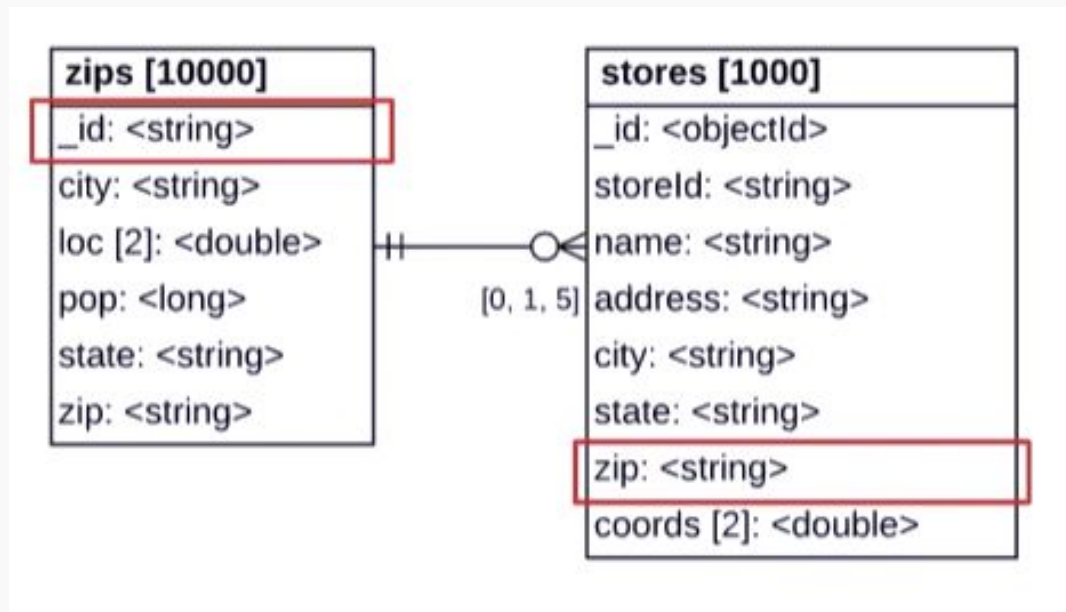
```
{  
  ...  
  stores:  
    [ "store1", "store408", "store441" ],  
  ...  
}
```

- Se utiliza un array para las referencias.
- Útil en grandes documentos. Para no tener toda la información embebida.

{ }

{ }

UNO A MUCHOS: REFERENCIADO – EN EL MUCHOS



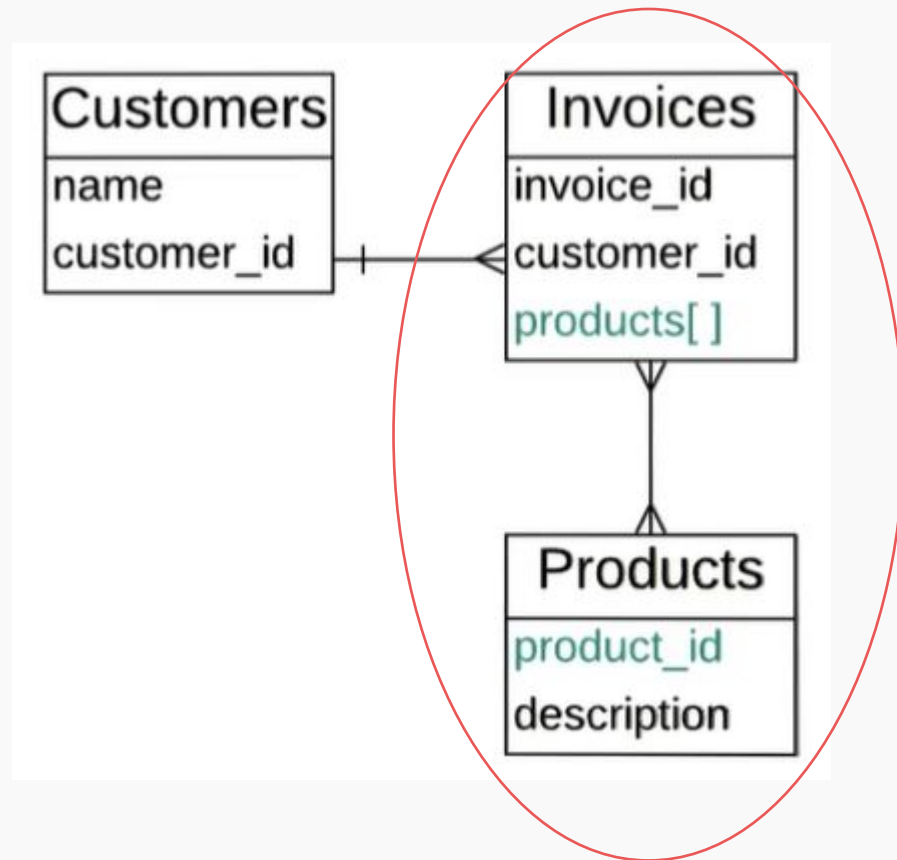
- Se utiliza un arreglo para las referencias
- Útil en grandes documentos.
- Cuando es importante que si hay una actualización aplique para todos.

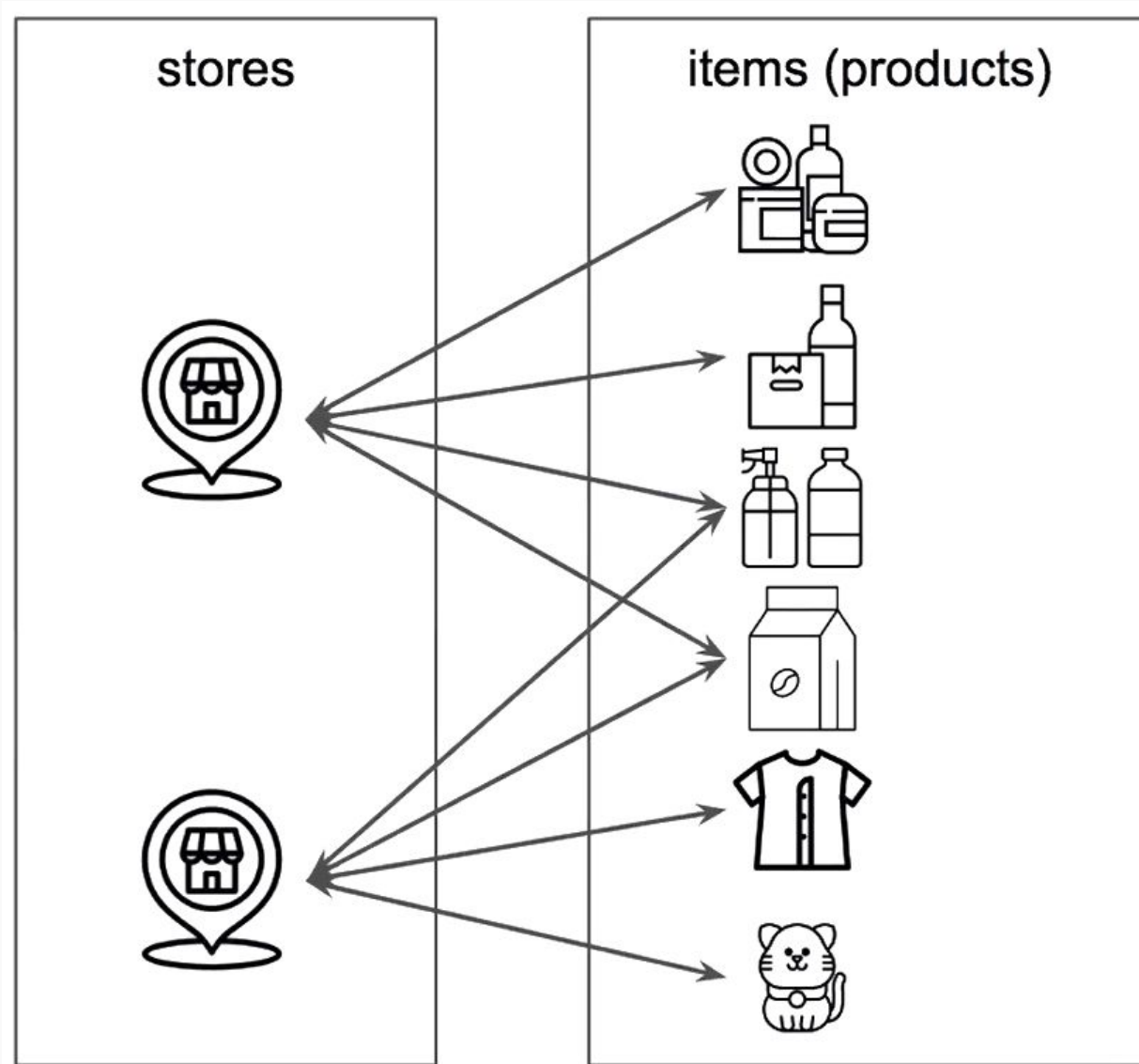
UNO A MUCHOS

- **Embebido o referenciado y seleccionar si en el uno o en el muchos.**
- Si se embeben documentos en el muchos, podría haber documentos duplicados, pero está bien, se debe garantizar desde la aplicación.
- Preferible escoger embebidos por simplicidad o cuando hay pocos documentos asociados, o es importante que la información esté junta (consultas).
- Embeber en el lado de la colección más consultada.
- Preferir referenciar cuando los documentos asociados no siempre sean requeridos en las consultas.

MUCHOS A MUCHOS

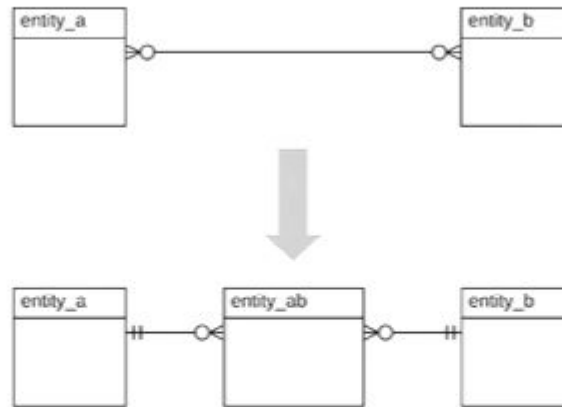
()





EJEMPLO

Many-to-Many \Rightarrow 2 x One-to-Many



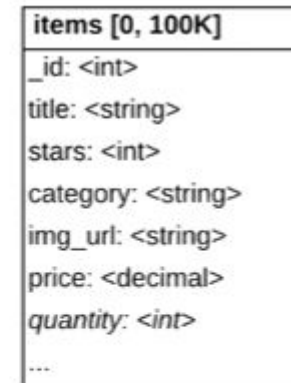
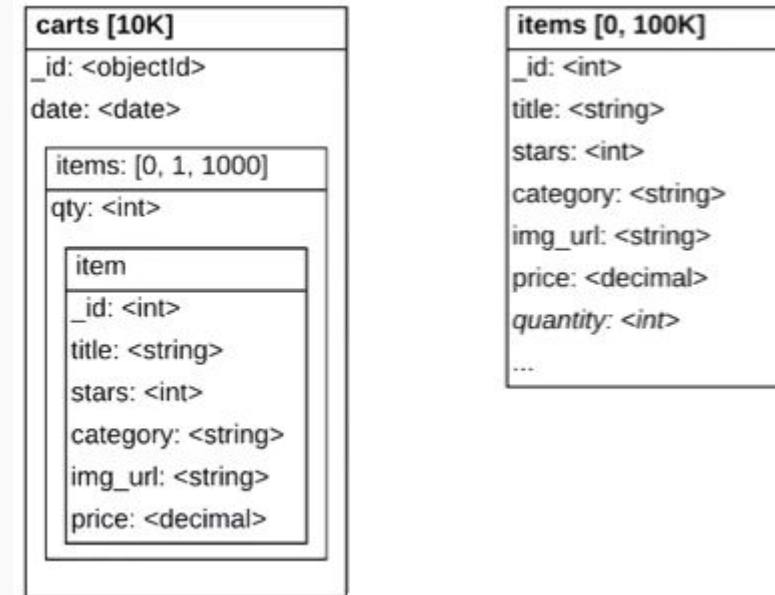
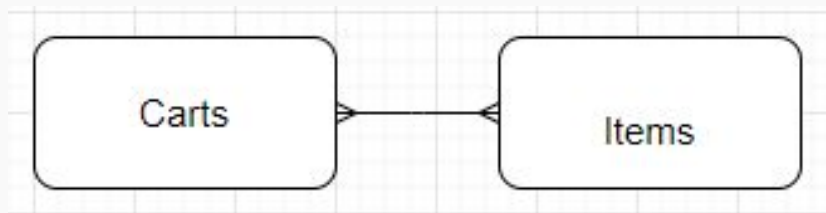
actor_id	actor_name
actor1	Marlon Brando
actor2	Al Pacino
actor3	Robert DeNiro

movie_id	actor_id
movie1	actor1
movie1	actor2
movie2	actor2

movie_id	movie_title
movie1	The Godfather
movie2	The Godfather 2

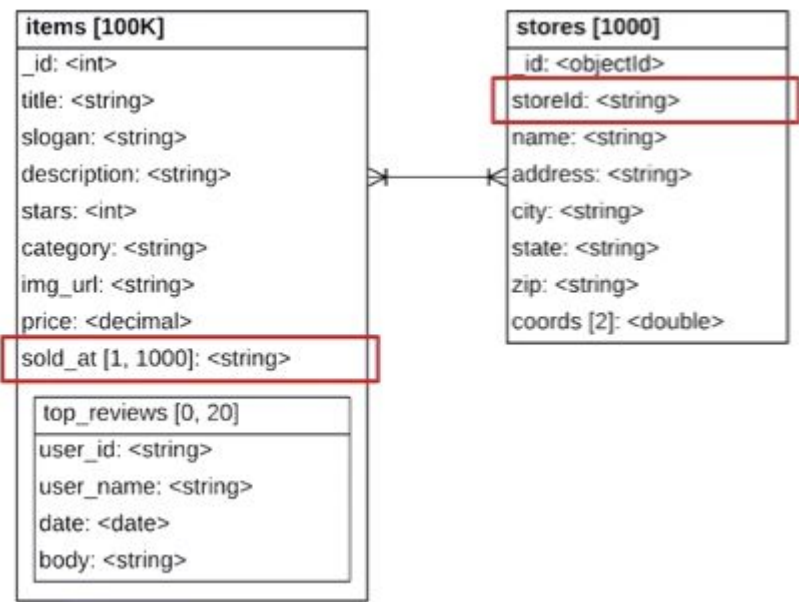
MUCHOS A MUCHOS: EMBEBIDO EN EL PPAL

- Los documentos del lado menos consultado son embebidos.
- Interesa los datos que tenía en el momento de la operación.
- Se mantiene los documentos “fuente” en otra colección.
- Indexación en el array.

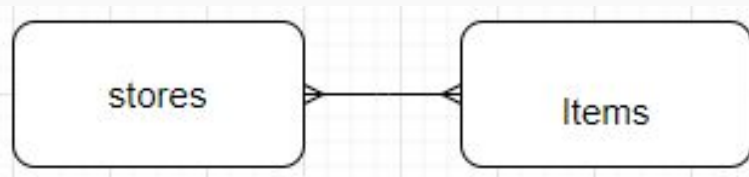


MUCHOS A MUCHOS: REFERENCIADO EN EL PPAL

{ }



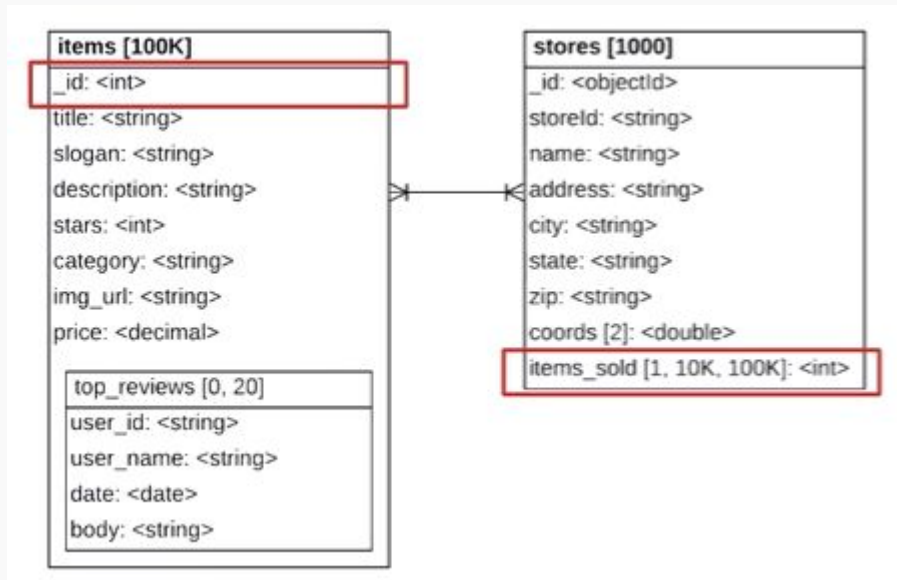
- Arreglo de referencias a los documentos de la otra colección.
- referencias disponibles tras la primera consulta en la colección principal.



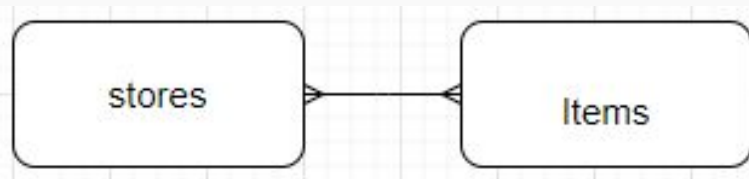
{ }

{ }

MUCHOS A MUCHOS: REFERENCIADO EN LA SECUNDARIA



- Arreglo de referencias a los documentos de la otra colección.
- Necesita hacer una segunda consulta para obtener información adicional.

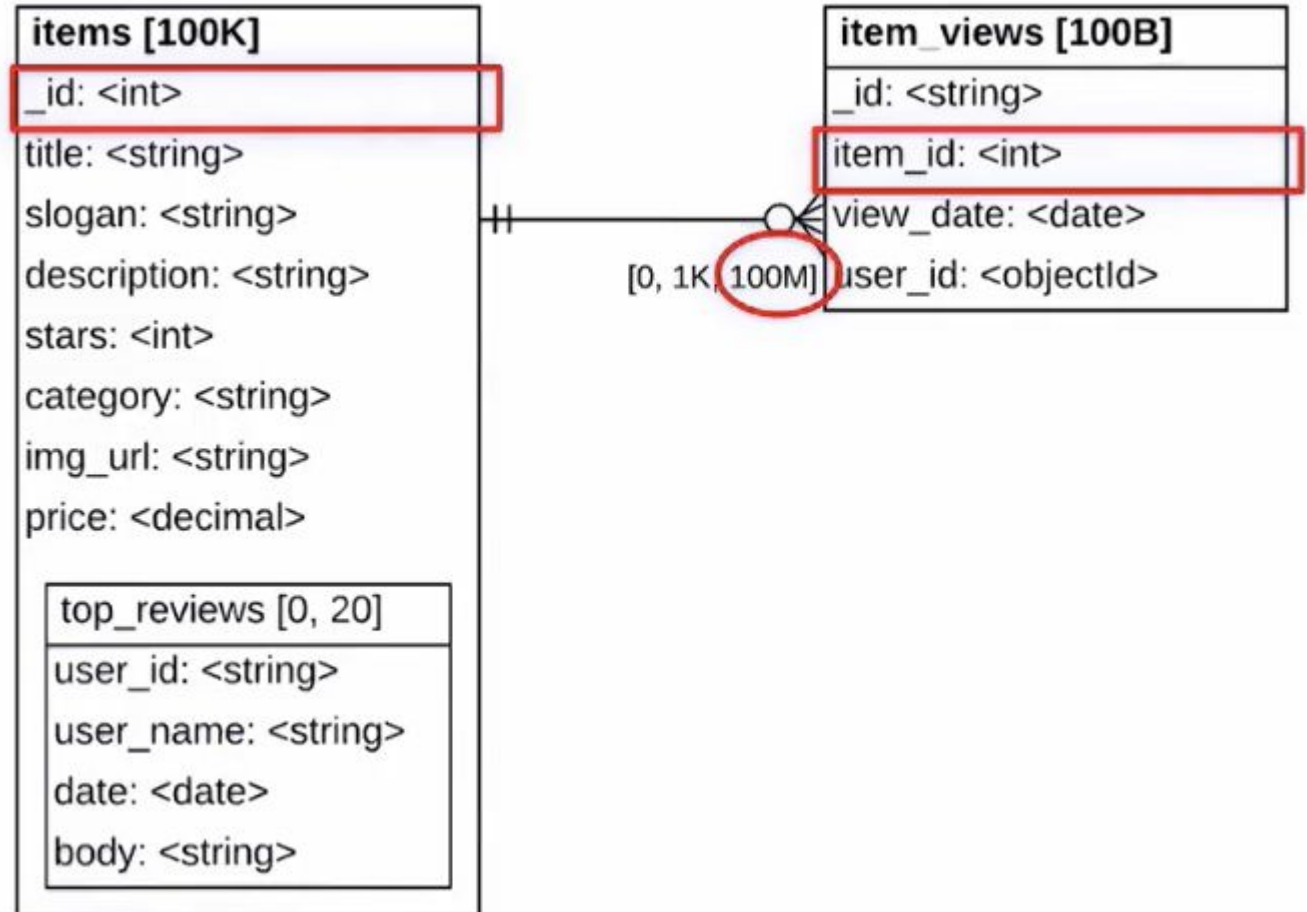


MUCHOS A MUCHOS

- Asegurarse de que no puede ser simplificada.
- Una relación muchos a muchos, puede ser reemplazada por dos uno a muchos.
- Preferible embeber en el lado de la colección más consultada.
- Preferible la incrustación para la información que es principalmente estática en el tiempo y puede beneficiarse de la duplicación.
- Preferir referenciar a embeber para evitar el manejo de la duplicación.

ONE-TO-ZILLION

- Solo se puede hacer referenciado- y se referencia el documento del lado del uno en el del “zillion”.

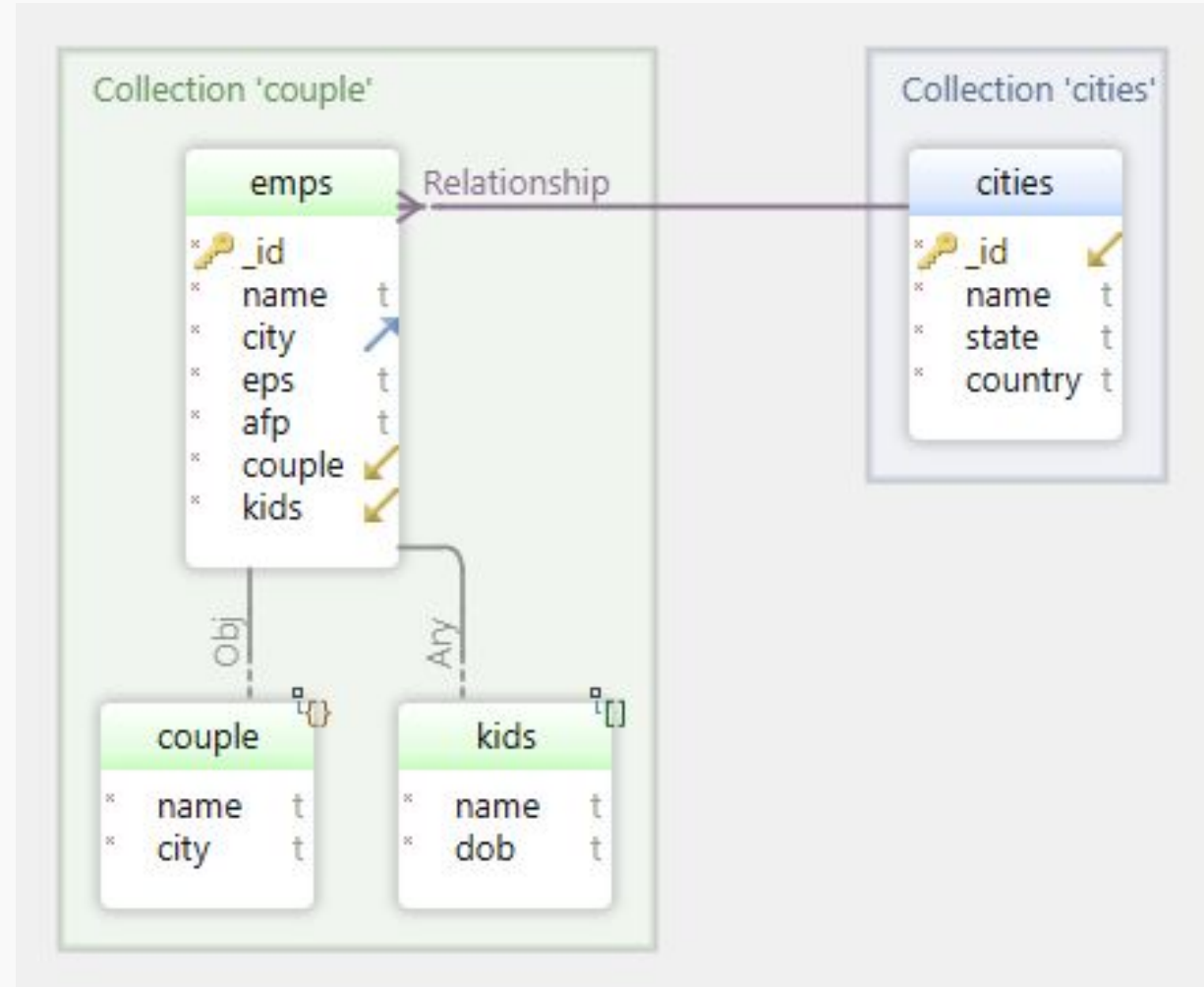


3) PATRONES

	Use Case Categories						
	Catalog	Content Management	Internet of Things	Mobile	Personalization	Real-Time Analytics	Single View
Patterns	✓		✓	✓		✓	
	✓	✓					✓
			✓			✓	
	✓		✓	✓	✓	✓	✓
	✓	✓			✓		✓
	✓			✓		✓	
			✓	✓	✓		
			✓			✓	
	✓	✓		✓			✓
	✓	✓	✓	✓	✓	✓	✓
	✓	✓		✓	✓		
	✓	✓					
	✓	✓					

<https://www.mongodb.com/blog/post/building-with-patterns-a-summary>

PRÁCTICA



Para la siguiente actividad, cada uno va a crear su propia BD...
Ej. USE COMPANY

DOCUMENTO EMBEBIDO

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

Embedded sub-document

Embedded sub-document

Implica desnormalización de los datos

Embedding

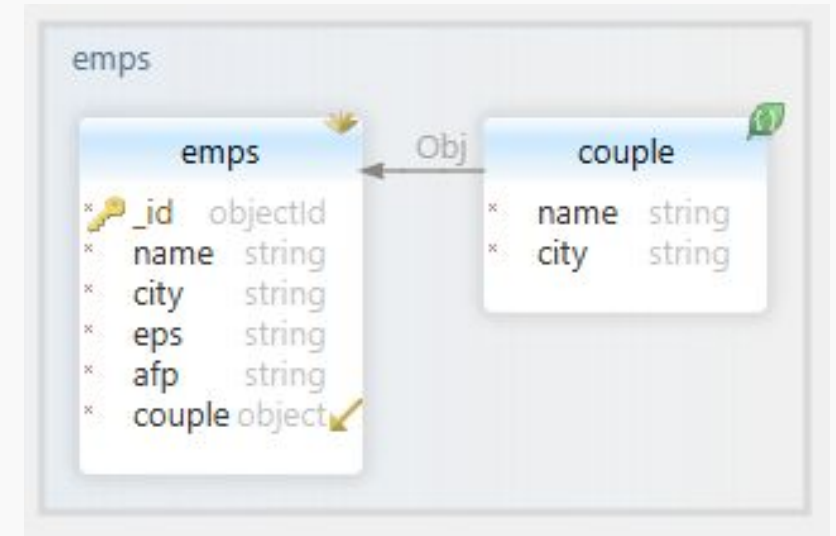
```
{
  "_id": ObjectId("573a1390f29313caabcd413b"),
  "title": "Star Wars: Episode IV - A New Hope",
  "cast": [
    {"actor": "Mark Hamill", "character": "Luke Skywalker"},
    {"actor": "Harrison Ford", "character": "Han Solo"},
    {"actor": "Carrie Fisher", "character": "Princess Leia Organa"},
    ...
  ]
}
```

DOCUMENTOS EMBEBIDOS: UNO A UNO

- Insertar en la colección employees, utilizando sus propios datos e incluir los datos de la pareja como un documento embebido:

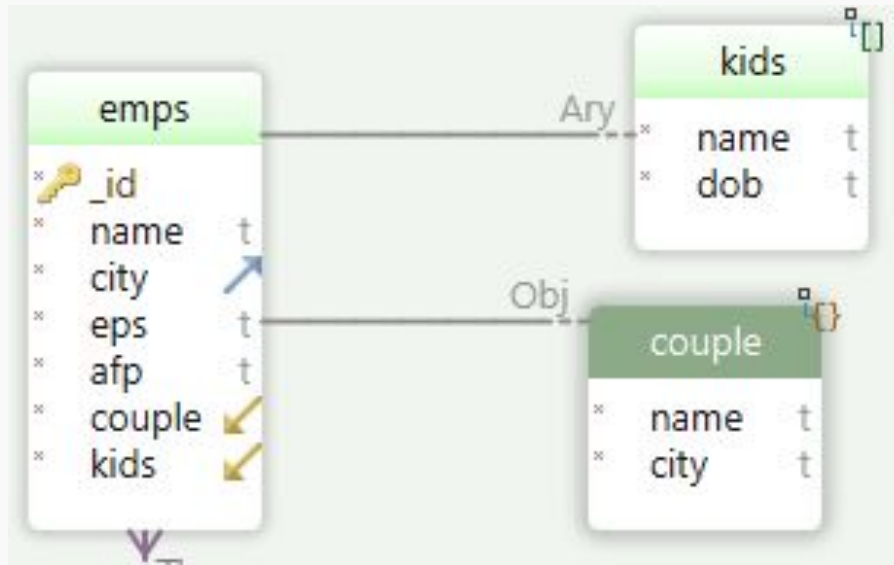
```
{
  "name": "Monica Rojas",
  "city": "Cali",
  "eps": "sura",
  "afp": "colpensiones",
  "couple": {
    "name": "Felipe Sanchez",
    "city": "Medellin"
  }
}
```

Documento
embebido



DOCUMENTOS EMBEBIDOS: UNO A MUCHOS

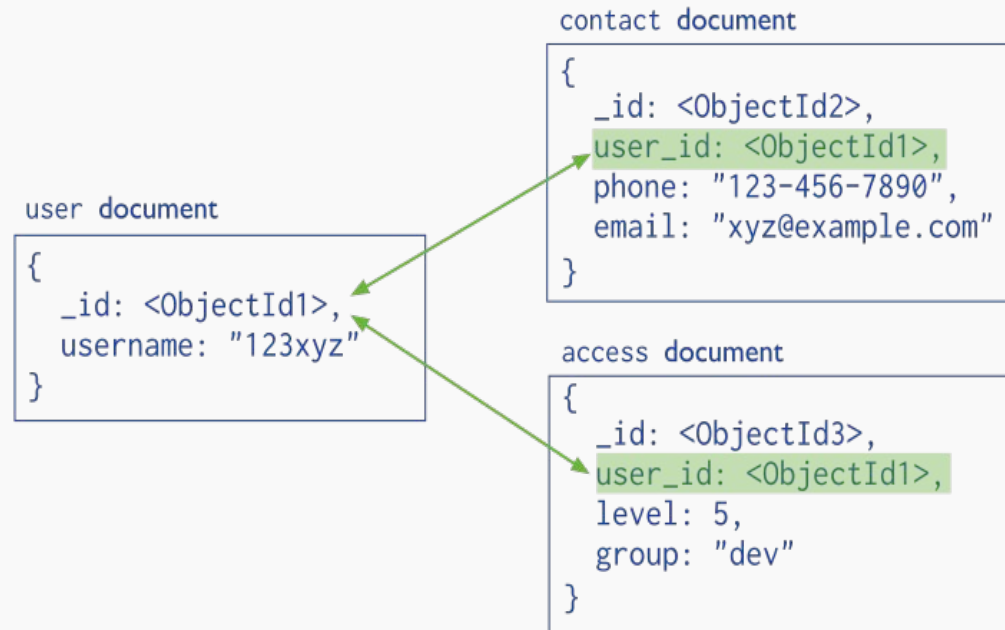
- Modificar el documento ingresado anteriormente e incluir hijos o sobrinos, en el campo kids (name, dob):



Documento
embebido

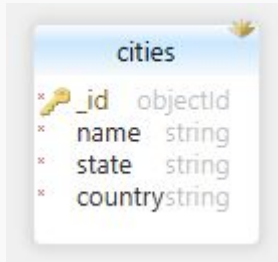
MODELO NORMALIZADO

- Referencia manual
- En el ejemplo, se ven tres tipos de documentos diferentes, relacionados por el Id del documento.



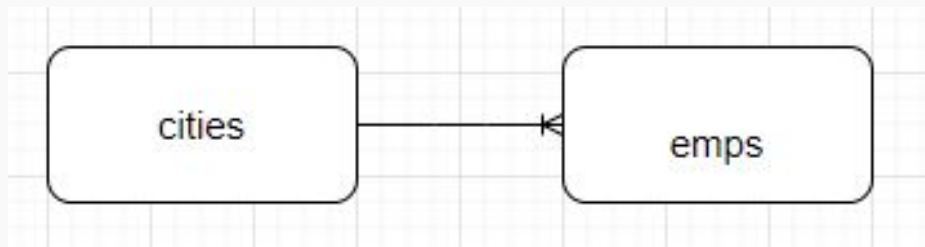
DOCUMENTO REFERENCIADO: UNO A MUCHOS

- Vamos a crear una colección cities, e insertar documentos:



```
{  
  "name": "Cali",  
  "state": "Valle",  
  "country": "Colombia"  
},  
{  
  "name": "Medellín",  
  "state": "Antioquia",  
  "country": "Colombia"  
}
```

Modificar- el documento de empleado ingresado previamente y colocar la referencia manual a la ciudad.



LOOKUP

- Ejecuta un left outer join entre colecciones de la misma BD

```
db.couple.aggregate( [  
  {  
    $lookup:  
    {  
      from: "cities",  
      localField: "city",  
      foreignField: "_id",  
      as: "city_birth"  
    }  
  }  
])
```

```
[  
  {  
    _id: ObjectId("6375aaf4fa7a85f391739f5e"),  
    name: 'Monica Rojas',  
    city: ObjectId("6375af37fa7a85f391739f5f"),  
    eps: 'sura',  
    afp: 'colpensiones',  
    couple: { name: 'Felipe Sanchez', city: 'Medellin' },  
    kids: [  
      { name: 'Camila Sanchez', bod: '03/09' },  
      { name: 'Alejandro Sanchez', bod: '07/01' }  
    ],  
    city_birth: [  
      {  
        _id: ObjectId("6375af37fa7a85f391739f5f"),  
        name: 'Cali',  
        state: 'Valle',  
        country: 'Colombia'  
      }  
    ]  
  }  
]
```

DOCUMENTO REFERENCIADO: DBREFS

- Se utiliza DBRefs para referenciar el documento.
- Tiene los siguientes campos
 - \$ref : nombre de la colección donde reside el documento
 - \$id : contiene el campo `_id` del documento referenciado
 - \$db : nombre de su base de datos (opcional)
- Los DBRefs permiten referenciar más fácilmente los documentos almacenados en diferentes colecciones o bases de datos.

```
{
  "_id" : ObjectId("5126bbf64aed4daf9e2ab771"),
  // .. application fields
  "creator" : {
    "$ref" : "creators",
    "$id" : ObjectId("5126bc054aed4daf9e2ab772"),
    "$db" : "users",
    "extraField" : "anything"
  }
}
```

Referencing

```
{
  "_id": ObjectId("573a1390f29313caabcd413b"),
  "title": "Star Wars: Episode IV - A New Hope",
  "director": "George Lucas",
  "runtime": 121,
  "filming_locations": [
    ObjectId("654a1420f29313fggbcd718"),
    ObjectId("654a1420f29313fggbcd719"),
    ...
  ]
}
```

Driver Support for DBRefs

Driver	DBRef Support	Notes
C	Not Supported	You can traverse references manually.
C++	Not Supported	You can traverse references manually.
C#	Supported	Please see the C# driver page for more information.
Go	Not Supported	You can traverse references manually.
Haskell	Not Supported	You can traverse references manually.
Java	Supported	Please see the Java driver page for more information.
Node.js	Supported	Please see the Node.js driver page for more information.
Perl	Supported	Please see the Perl driver page for more information.
PHP	Not Supported	You can traverse references manually.
Python	Supported	Please see the PyMongo driver page for more information.
Ruby	Supported	Please see the Ruby driver page for more information.
Scala	Not Supported	You can traverse references manually.

REFERENCIAS

- MongoDB – balanceo de carga
<http://juanroy.es/es/how-mongodb-balance-your-data/>
 - MongoDB tiene soporte variado de lenguajes y librerías cliente:
<http://docs.mongodb.org/ecosystem/drivers/java/>
 - Para programar en Java mirar:
<http://docs.mongodb.org/ecosystem/drivers/java/>
<http://docs.mongodb.org/ecosystem/tutorial/getting-started-with-java-driver/>
 - Patrones
<https://www.mongodb.com/blog/post/building-with-patterns-a-summary-es>
 - Manual MongoDB
<https://docs.mongodb.com/manual/replication/>
- Herramienta Modelar :<https://dbdiagram.io/>
<https://hackolade.com/>
- Modelado, qué tener en cuenta:
<https://barcochrist.medium.com/modelado-de-datos-en-mongodb-aa19f963b479>