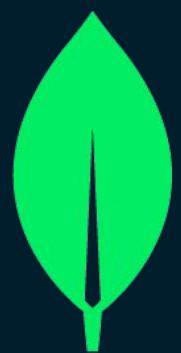




MongoDB®



MongoDB®

Curso de Modelado de Datos con MongoDB





Nicolas Molina
@nicobytes



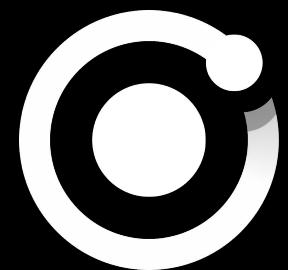
Experts



Microsoft®
Most Valuable
Professional



Platzi



ionic

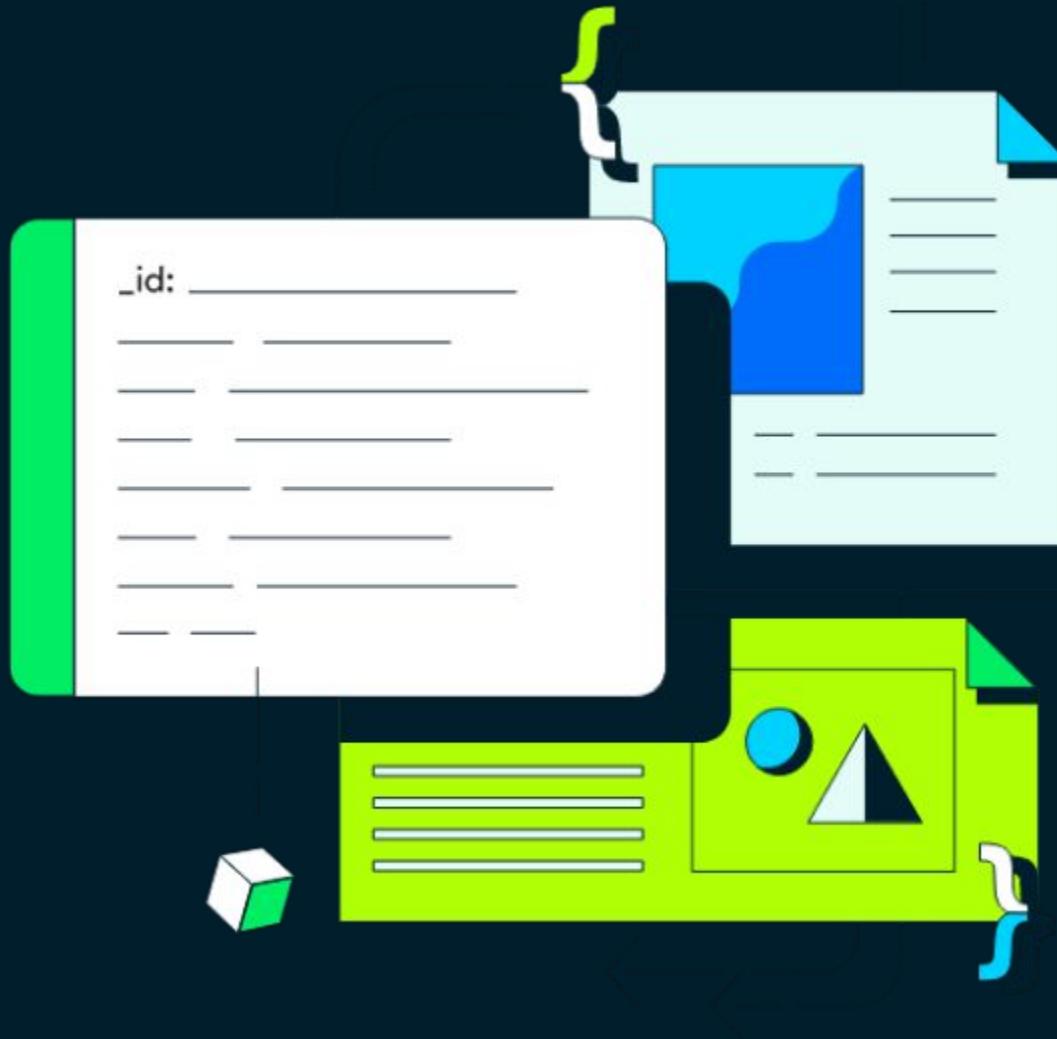


 MongoDB®

Beneficios

- Buen rendimiento
- Productividad en desarrollo
- Minimizar costos



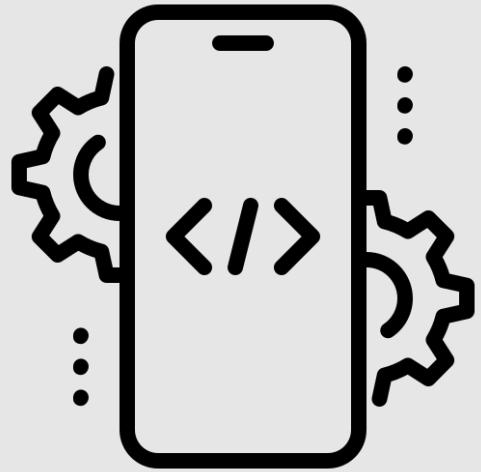


 MongoDB®

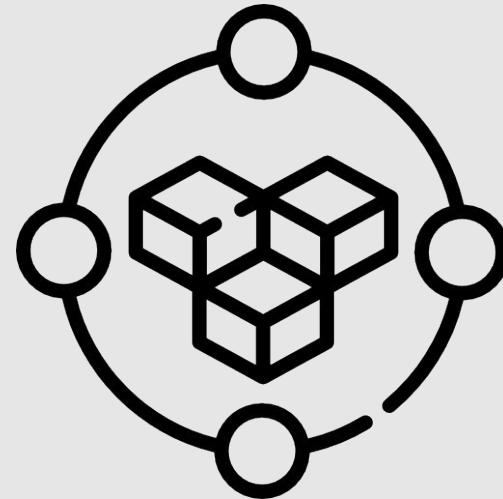
MongoDB

Que una base de datos NoSQL sea
muy flexible no significa que no
tenga una estructura.

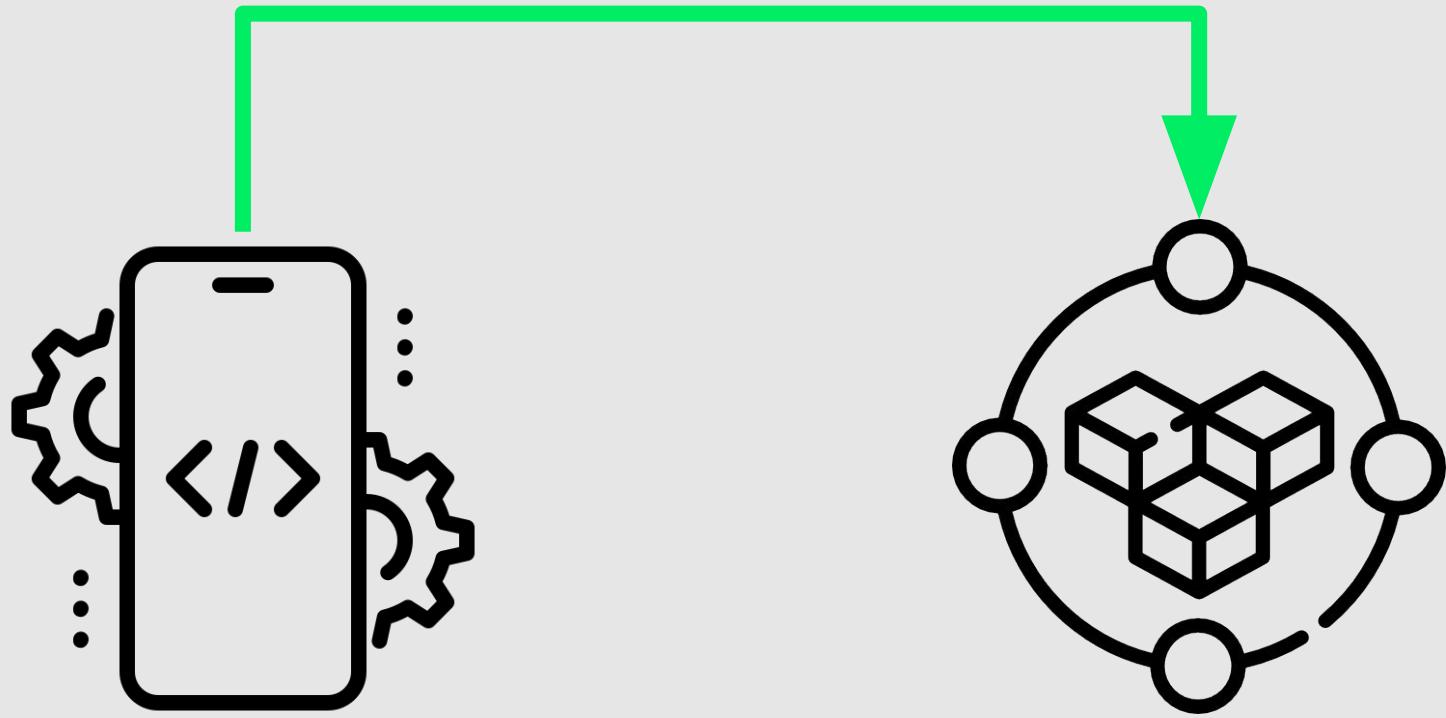




App

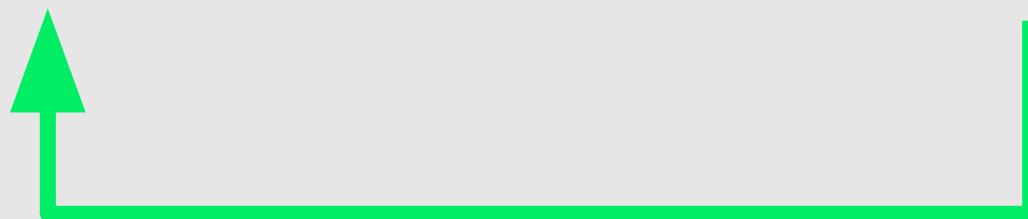


Data



App

Data



 MongoDB®

Proyecto



Conceptos para NoSQL



```
{  
  name: "sue",           ← field: value  
  age: 26,              ← field: value  
  status: "A",           ← field: value  
  groups: [ "news", "sports" ] ← field: value  
}  
          ↑  
        document
```



```
{  
  "_id": "5c8ecc1caa187d17ca6ed16",  
  "city": "ALPINE",  
  "zip": "35014",  
  "loc": {  
    "y": 33.331165,  
    "x": 86.208934  
  },  
  "pop": 3062,  
  "state": "AL"  
}
```



```
{  
  "_id": "5c8ecc1caa187d17ca6ed16",  
  "city": "ALPINE",  
  "zip": "35014",  
  "loc": {  
    "y": 33.331165,  
    "x": 86.208934  
  },  
  "pop": 3062,  
  "state": "AL"  
}
```

```
{  
  "_id": "5c8ecc1caa187d17ca6ed16",  
  "city": "ALPINE",  
  "zip": "35014",  
  "loc": {  
    "y": 33.331165,  
    "x": 86.208934  
  },  
  "pop": 3062,  
  "state": "AL"  
}
```



```
{  
  name: "sue",           ← field: value  
  age: 26,              ← field: value  
  status: "A",           ← field: value  
  groups: [ "news", "sports" ] ← field: value  
}  
          ↑  
        document
```



```
{  
  name: "al",  
  age: 18,  
  status: "D",  
  groups: [ "politics", "news" ]  
}
```



```
{  
  name: "al",  
  age: 18,  
  status: "D",  
  groups: [ "politics", "news" ]  
}
```

Collection

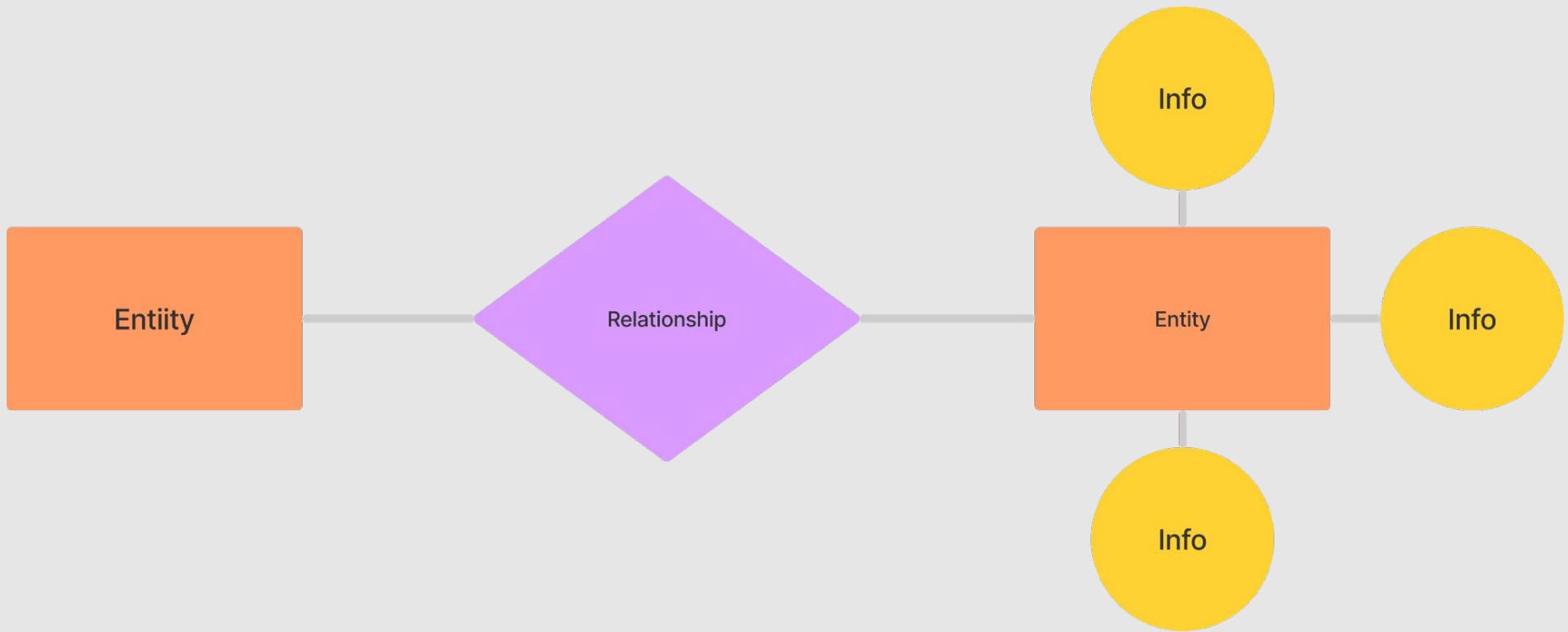


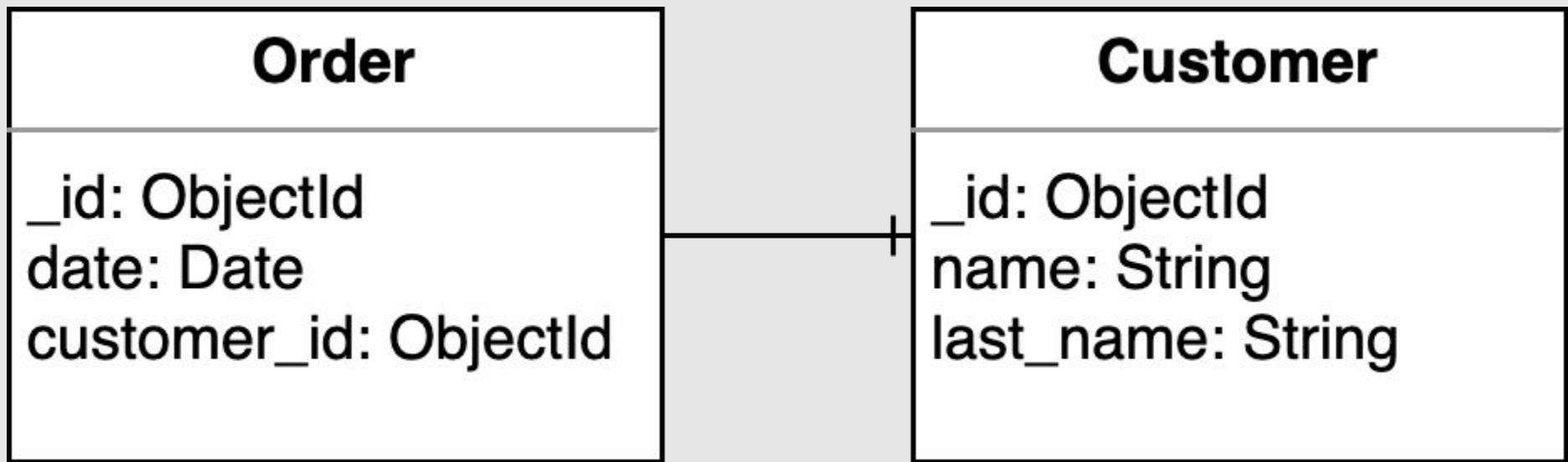


 MongoDB®

Diagrama ER









One



Many



One (and only one)



Zero or one



One or many



Zero or many



Metodología para el modelado de datos

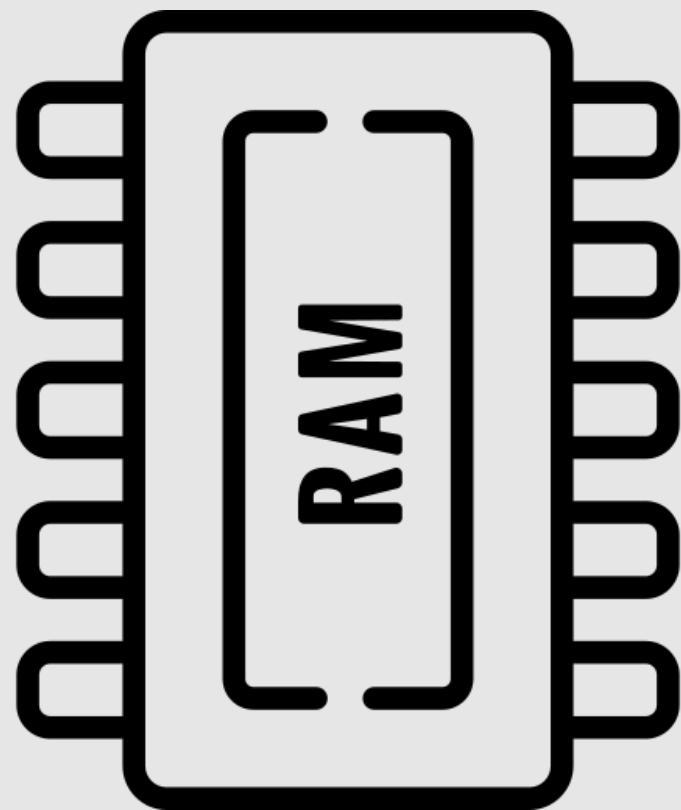


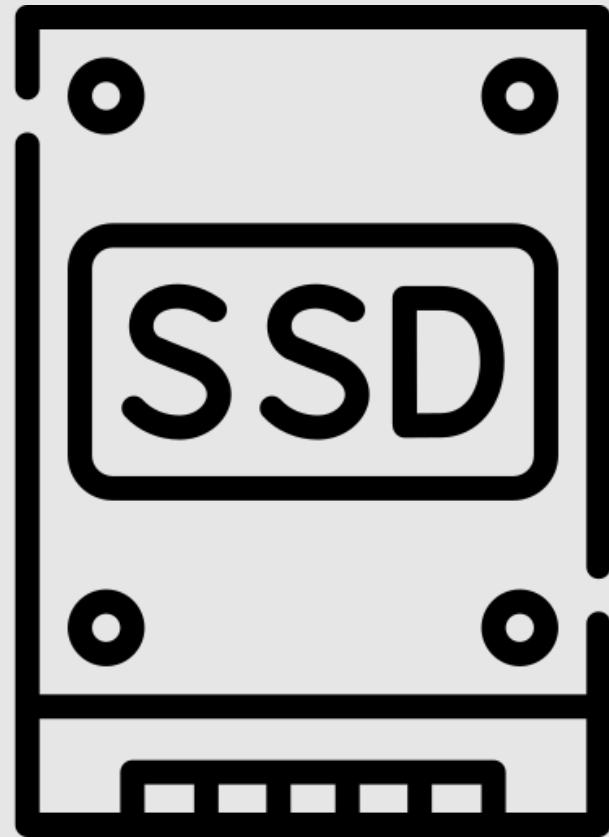
Restricciones









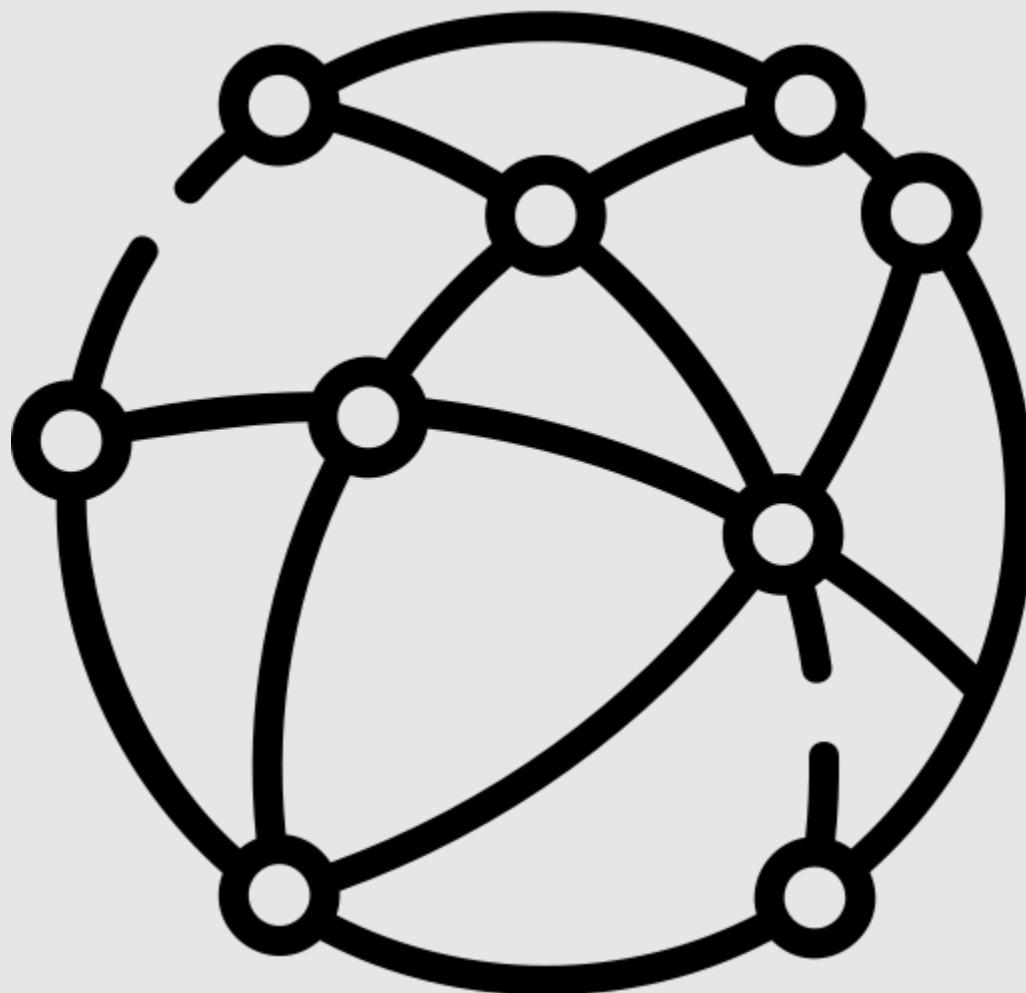


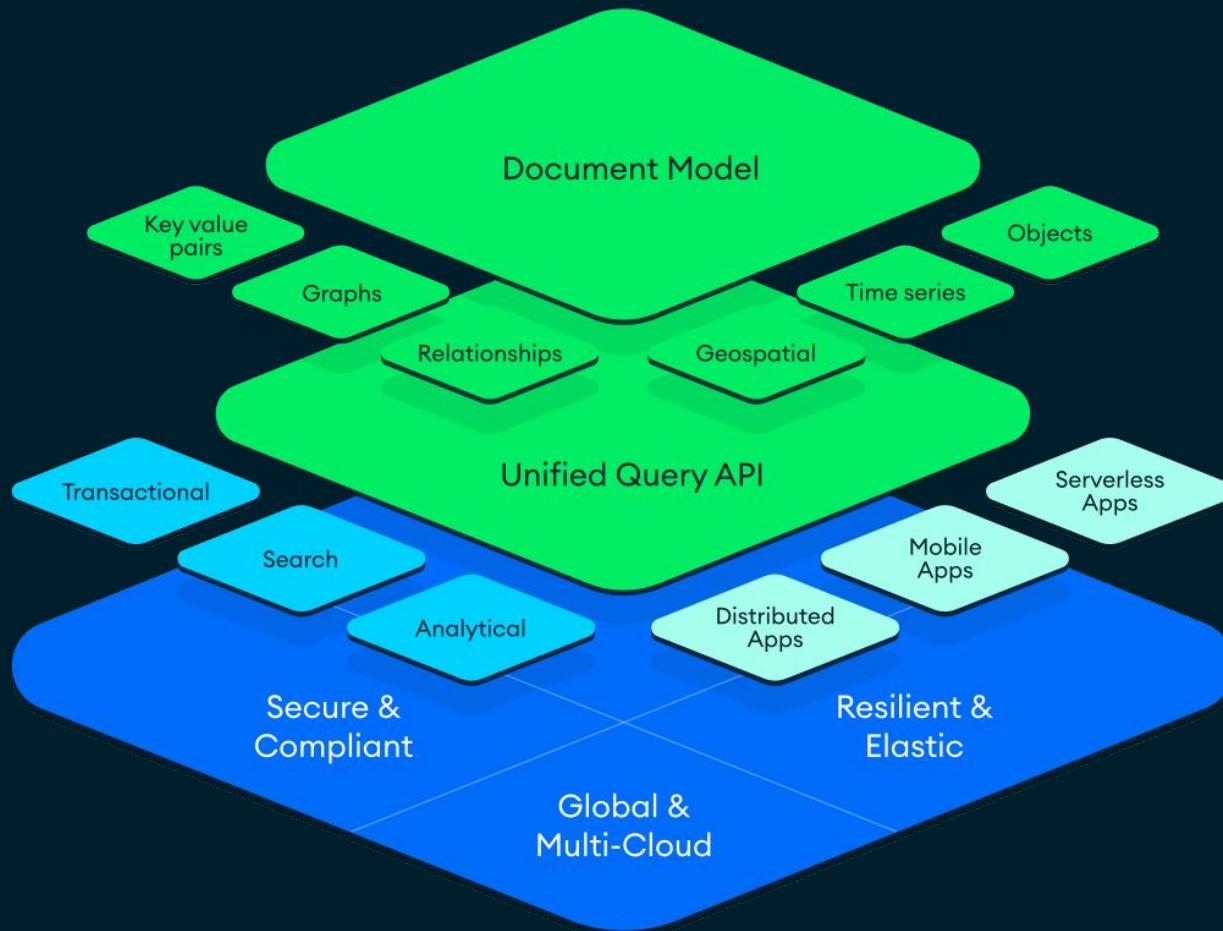




16 MB







Metodología



Metodología

1. Requerimientos
2. Identificar ER
3. Aplicar patrones

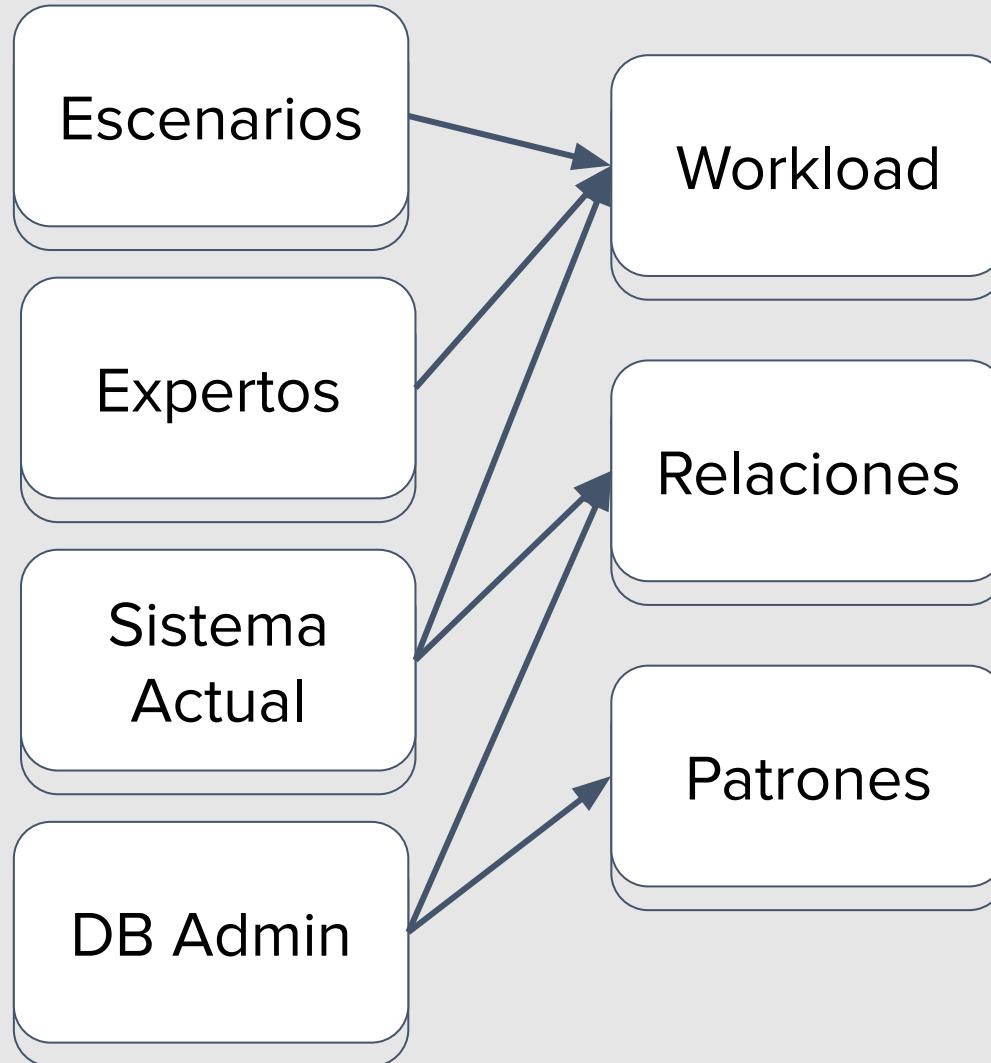
Escenarios

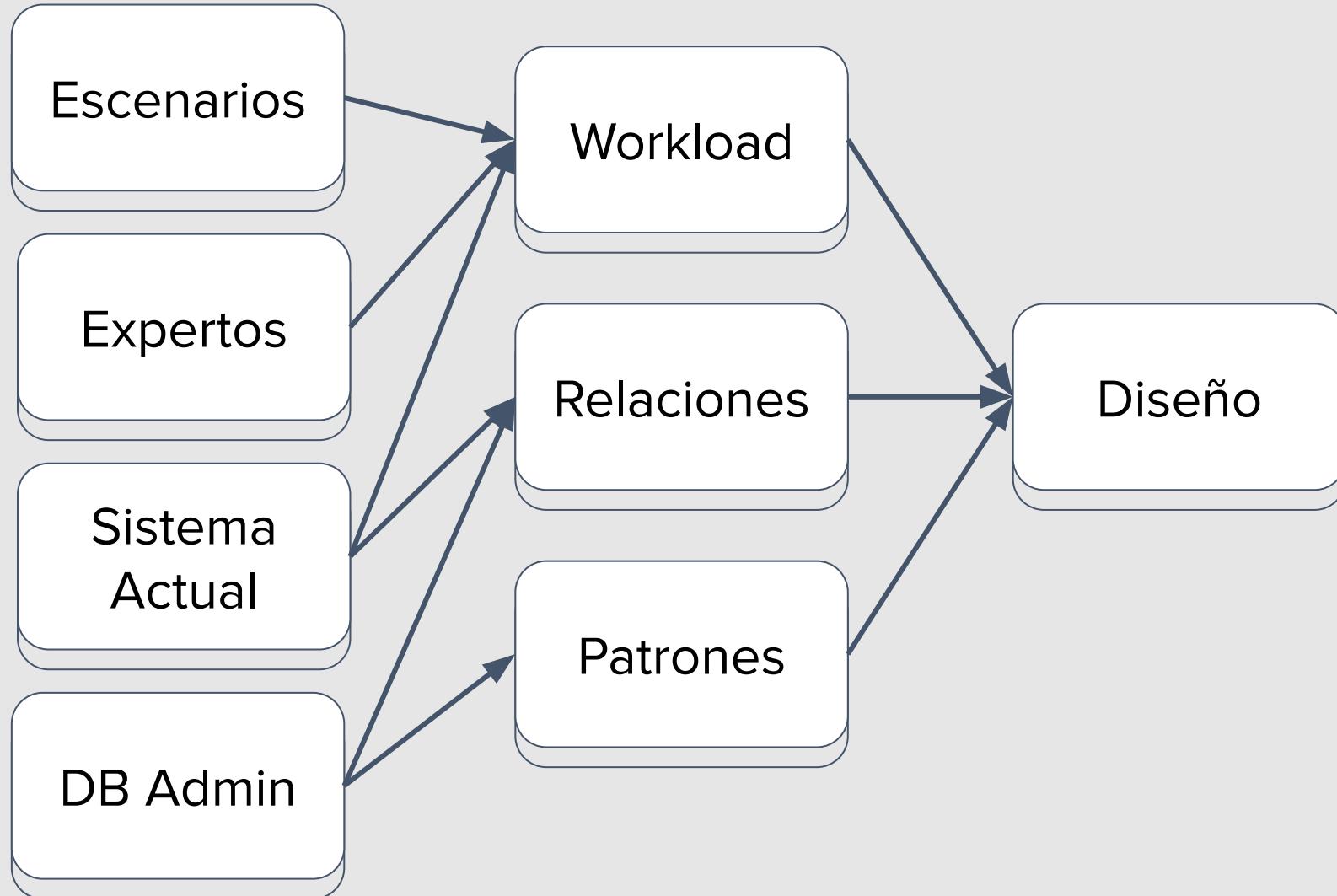
Expertos

Sistema
Actual

DB Admin







Workload

Tamaño de los datos
Consultas y indices
Operaciones importantes
Suposiciones



Relaciones

Identificar entidades

Identificar atributos

Identificar restricciones

Identificar relaciones

Embeber vs referenciar



Patrones

Identificar y aplicar



Crear la DB



Mongo Atlas

- > Cluster
- > Databases
- > Collections
- > Documents

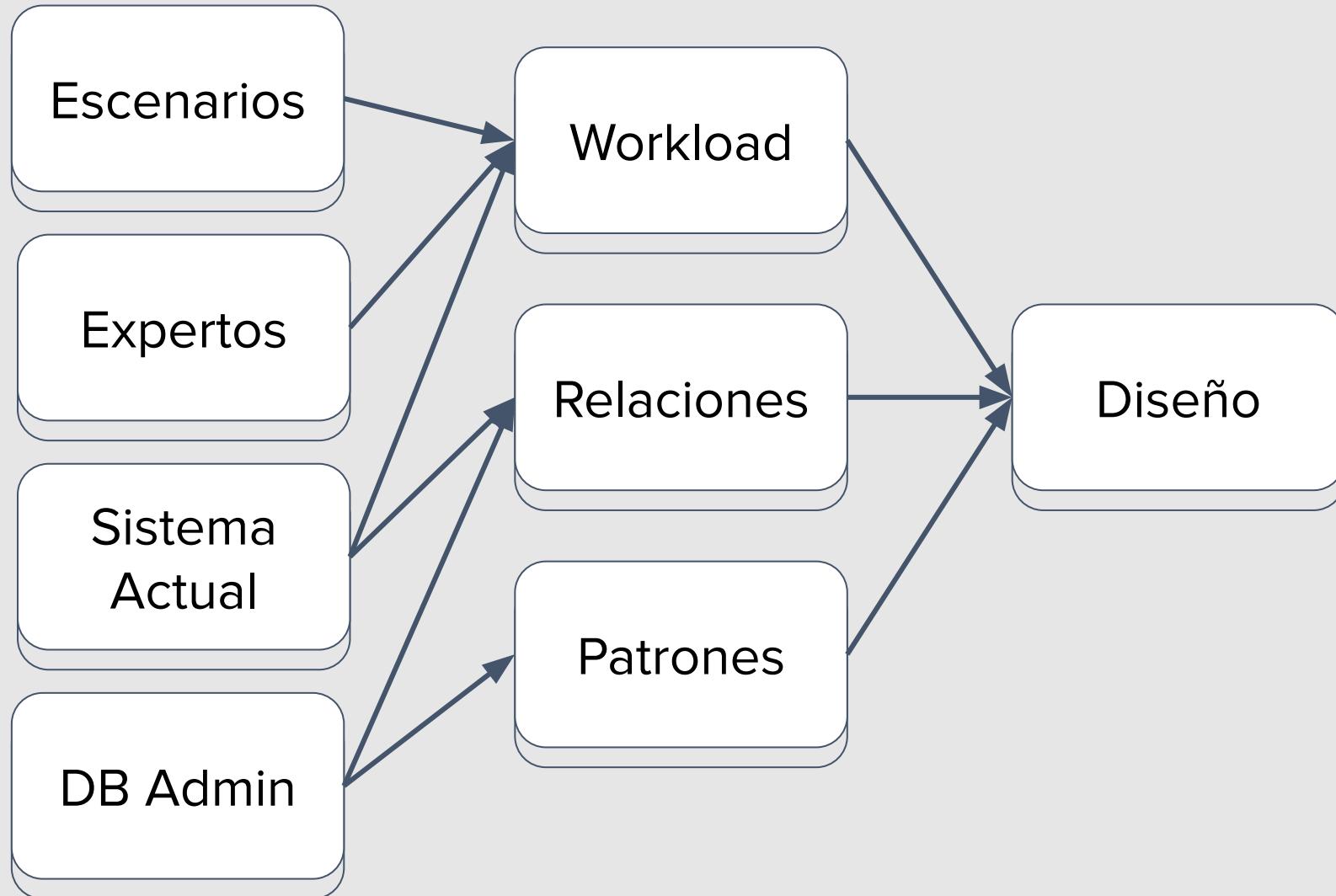


Ejemplo de **Workload**



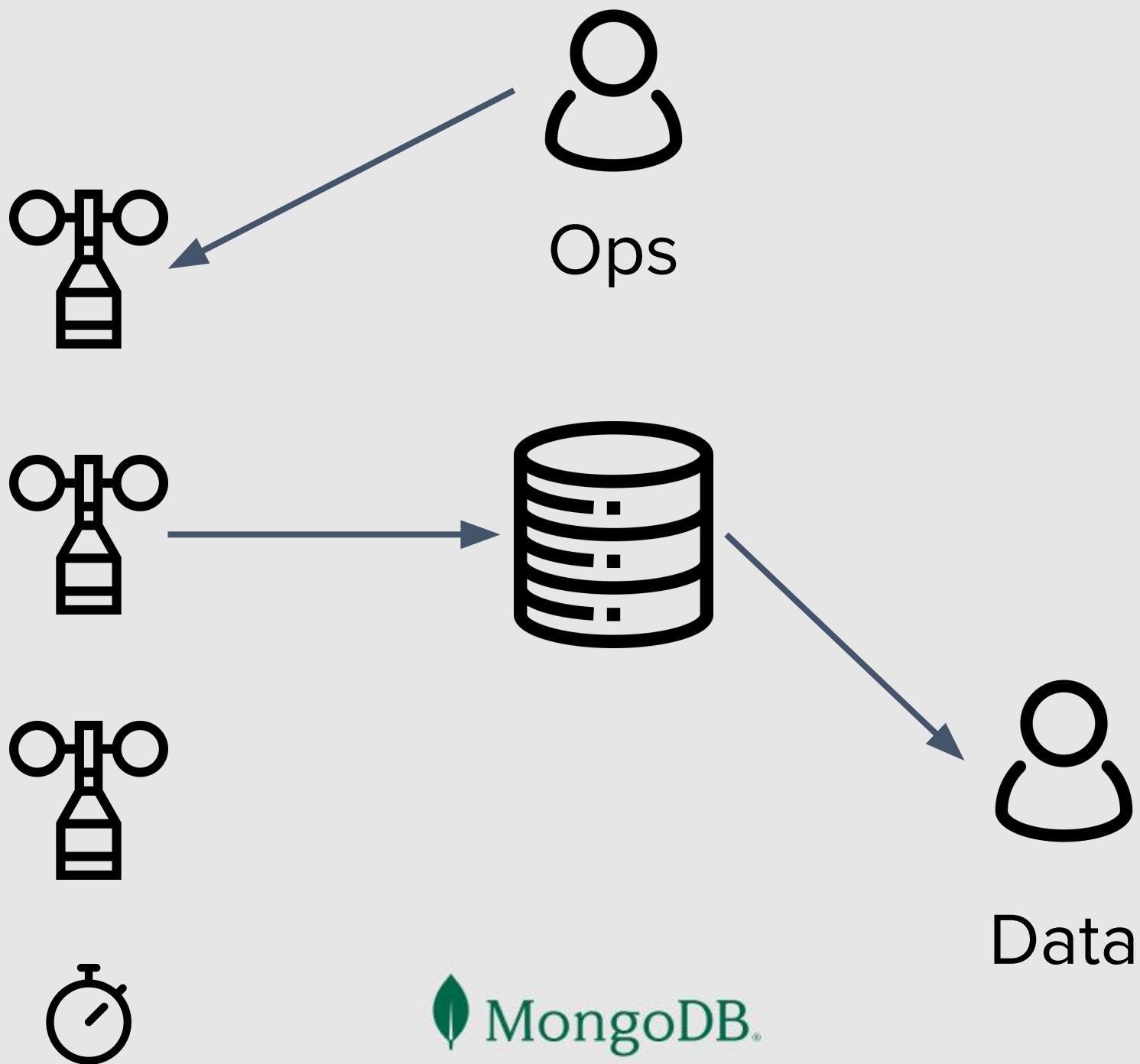
Metodología

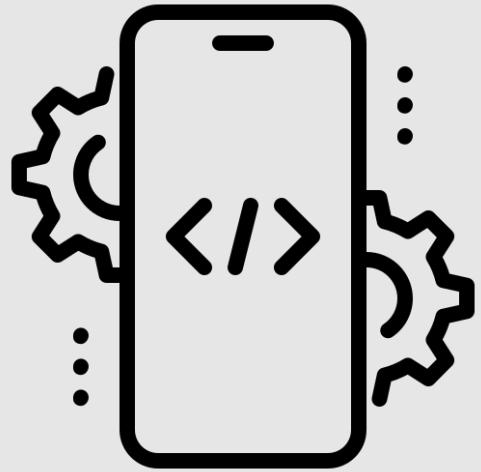
1. Requerimientos
2. Identificar ER
3. Aplicar patrones



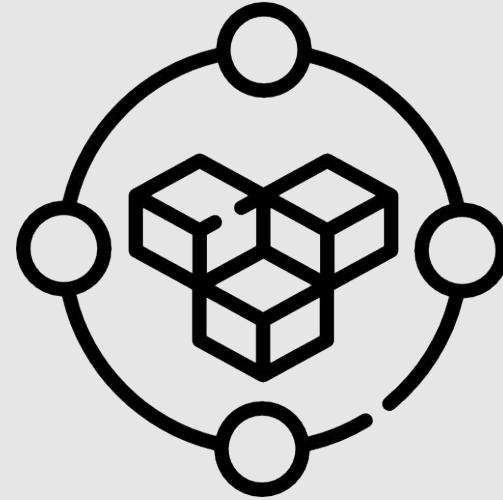


 MongoDB®

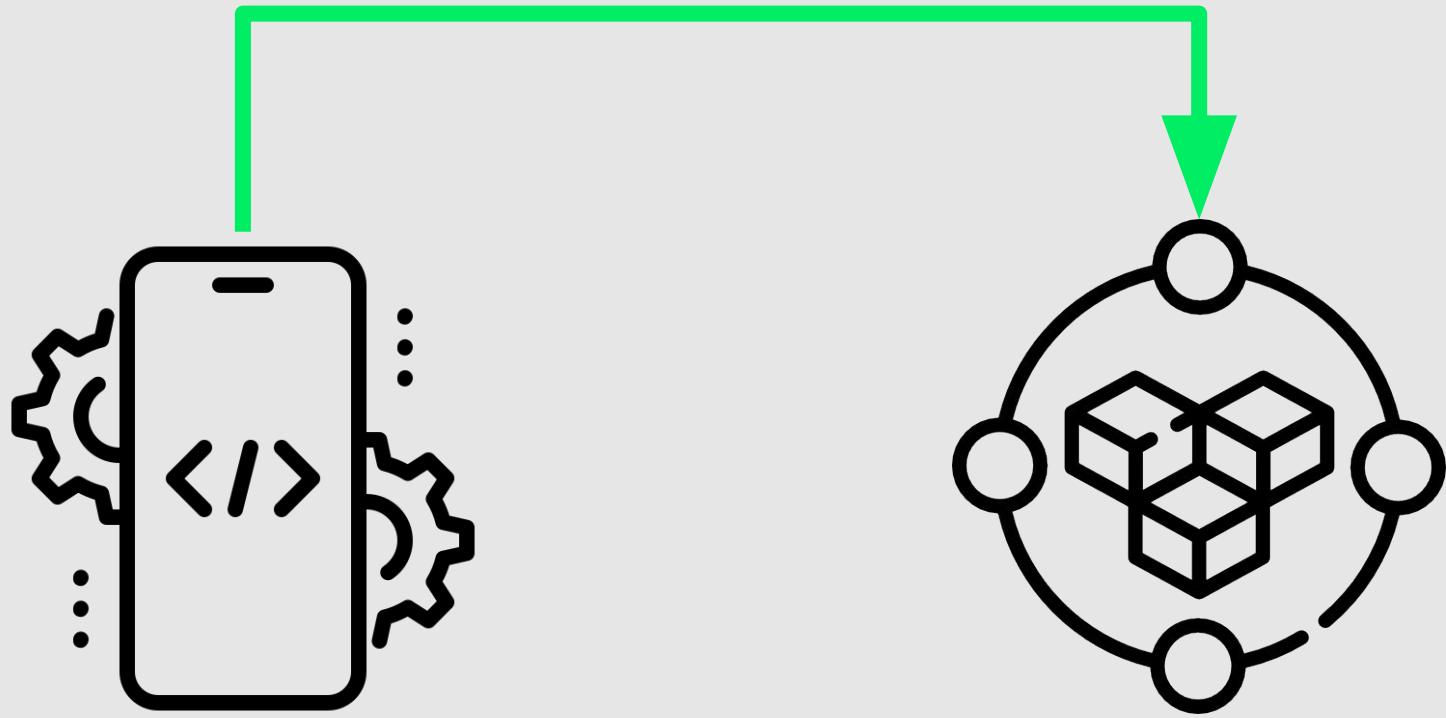




App

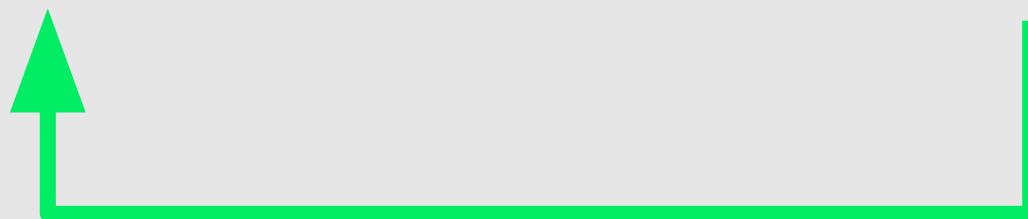


Data



App

Data



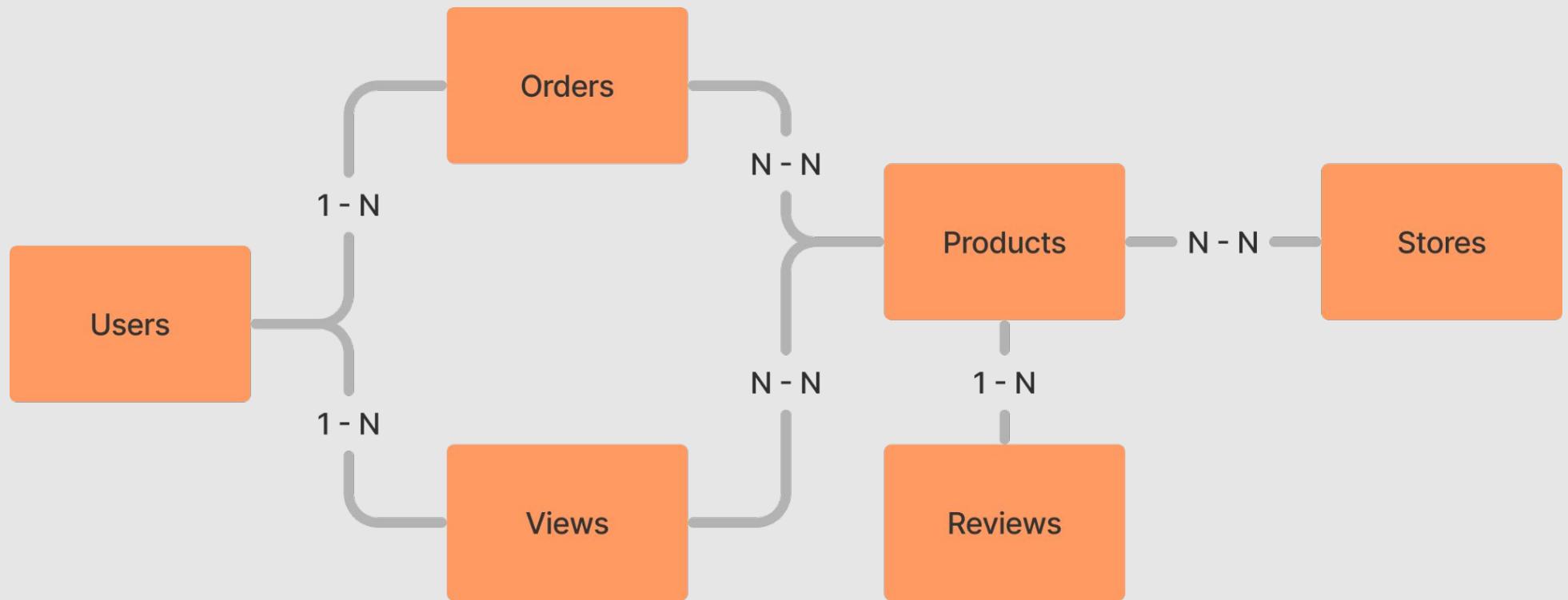
 MongoDB®

Preparando Proyecto



Workload

- Navigate and view products under 1ms
- Filter products by categories
- Add products in order by users



Validación de datos



Relaciones



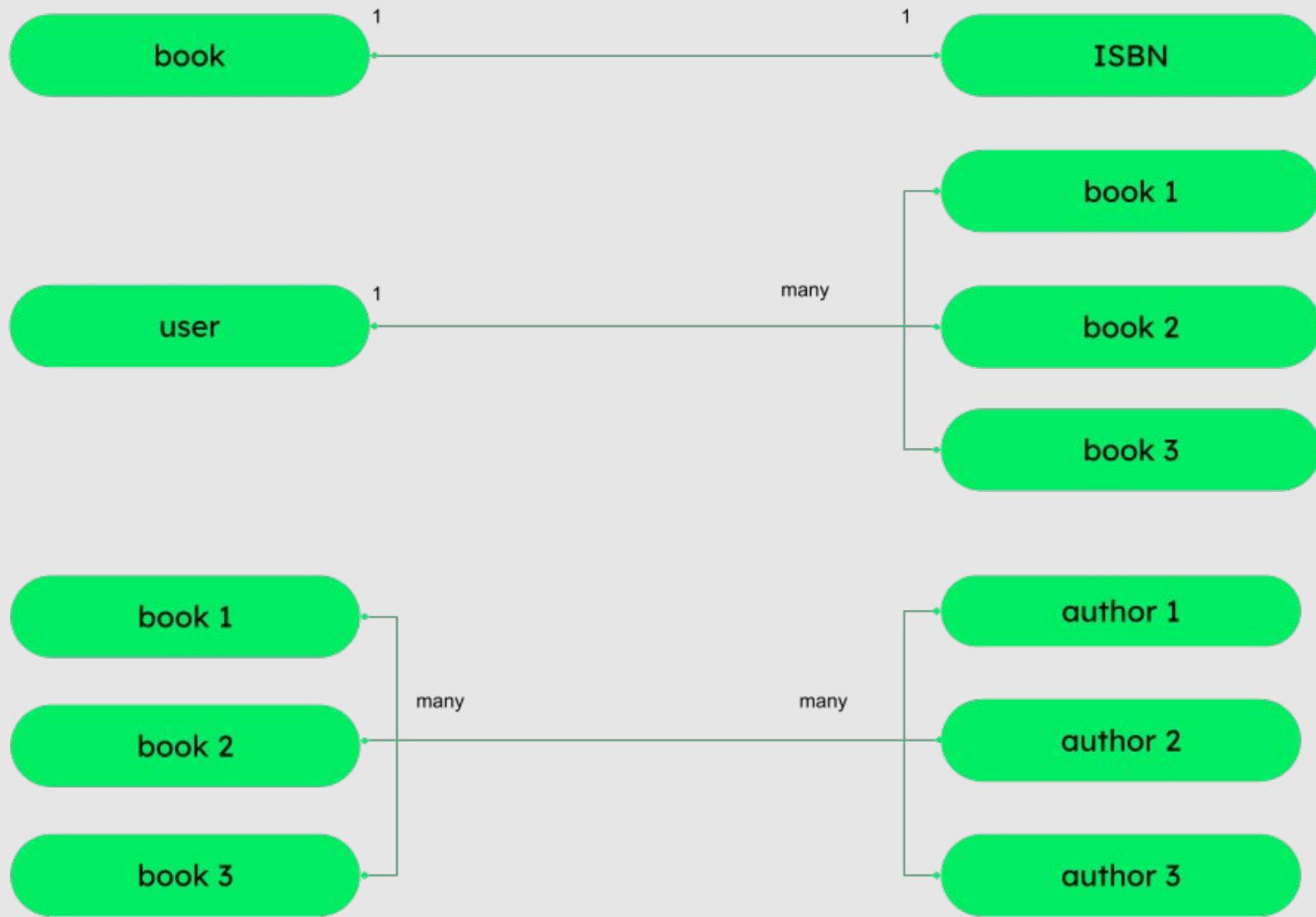
Tipos

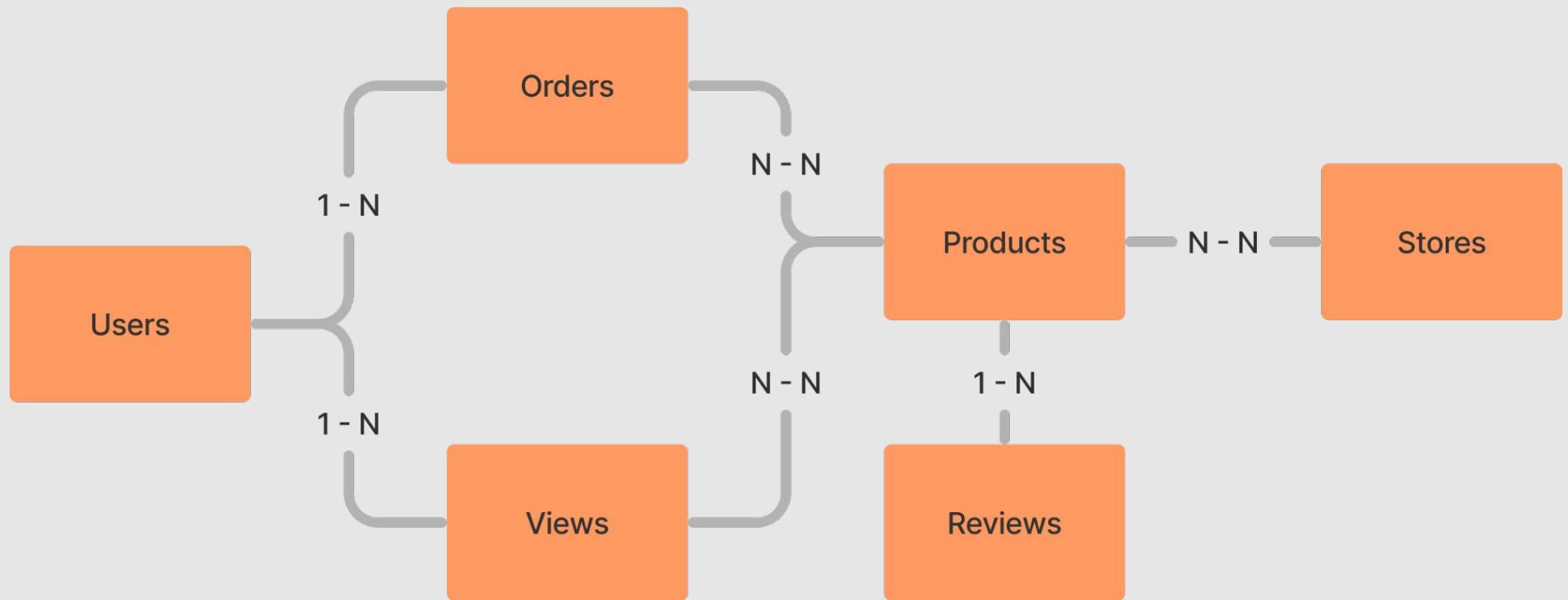
One to one (1-1)

One to many (1-N)

Many to many (N-N)







Embeber vs. Referenciar



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



Embedded sub-document

Embedded sub-document



user document

```
{  
  _id: <ObjectId1>,  
  username: "123xyz"  
}
```

contact document

```
{  
  _id: <ObjectId2>,  
  user_id: <ObjectId1>,  
  phone: "123-456-7890",  
  email: "xyz@example.com"  
}
```

access document

```
{  
  _id: <ObjectId3>,  
  user_id: <ObjectId1>,  
  level: 5,  
  group: "dev"  
}
```



```
{  
  "_id": "5c8ecc1caa187d17ca6ed16",  
  "city": "ALPINE",  
  "zip": "35014",  
  "loc": {  
    "y": 33.331165,  
    "x": 86.208934  
  },  
  "pop": 3062,  
  "state": "AL"  
}
```



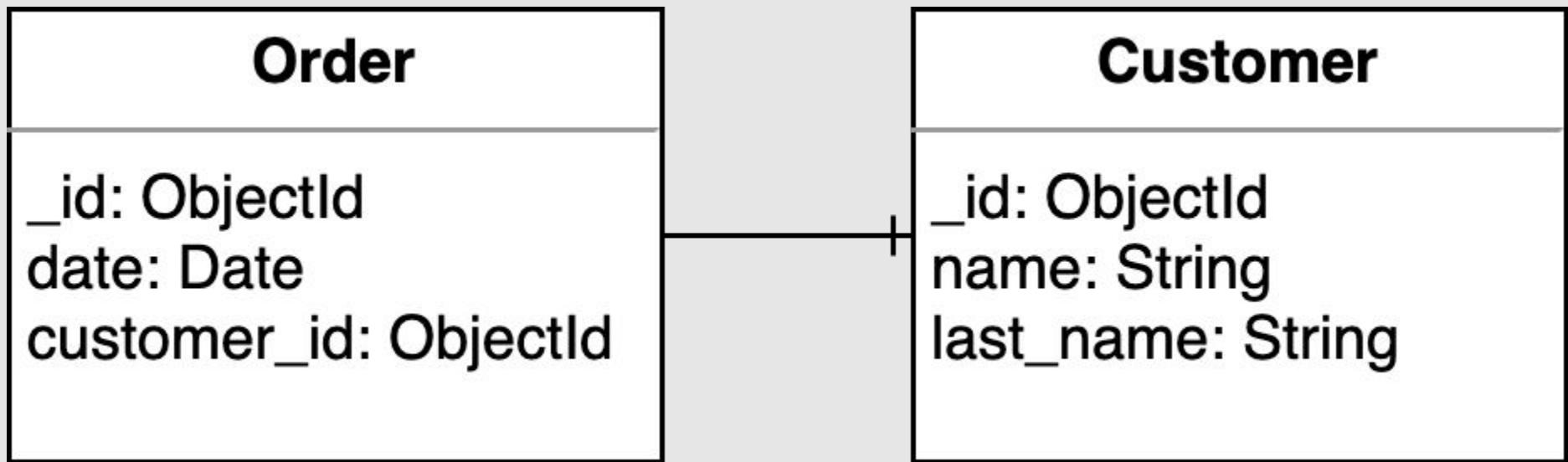
Entity

`_id: <ObjectId>`
`city: <string>`
`zip: <string>`
`pop: <number>`
`state: <string>`

Loc

`y: <number>`
`x: <number>`





¿Qué tan frecuente
es **consultada**
la información?



¿Qué tan frecuente
se **actualiza**
la información?



¿La información
se consulta en
conjunto o por partes?









Relaciones

1 - 1



User

```
_id: <ObjectId>
name: <string>
address_street: <string>
address_city: <string>
address_zip: <string>
shipping_street: <string>
shipping_city: <string>
shipping_zip: <string>
```



User

_id: <ObjectId>
name: <string>

Address

street: <string>
city: <string>
zip: <string>

Shipping Address

street: <string>
city: <string>
zip: <string>



MongoDB®

Cuando la información es consultada en conjunto

```
{  
  "_id": "5c8ecc1caa187d17ca6ed16",  
  "city": "ALPINE",  
  "zip": "35014",  
  "loc": {  
    "y": 33.331165,  
    "x": 86.208934  
  },  
  "pop": 3062,  
  "state": "AL"  
}
```



Cuando los sub docs son una dependencia

```
{  
  "name": "Nicolas",  
  "email": "nico@gmail.com",  
  "address": {  
    "city": "Bogotá",  
    "zip": "111011",  
    "street": "Cr 324"  
  }  
}
```



En el 90% de los casos, cuando hay una **relación 1:1**,
esta suele estar embebida.

```
{  
  "name": "Nicolas",  
  "email": "nico@gmail.com",  
  "address": {  
    "city": "Bogotá",  
    "zip": "111011",  
    "street": "Cr 324"  
  }  
}
```



Relaciones

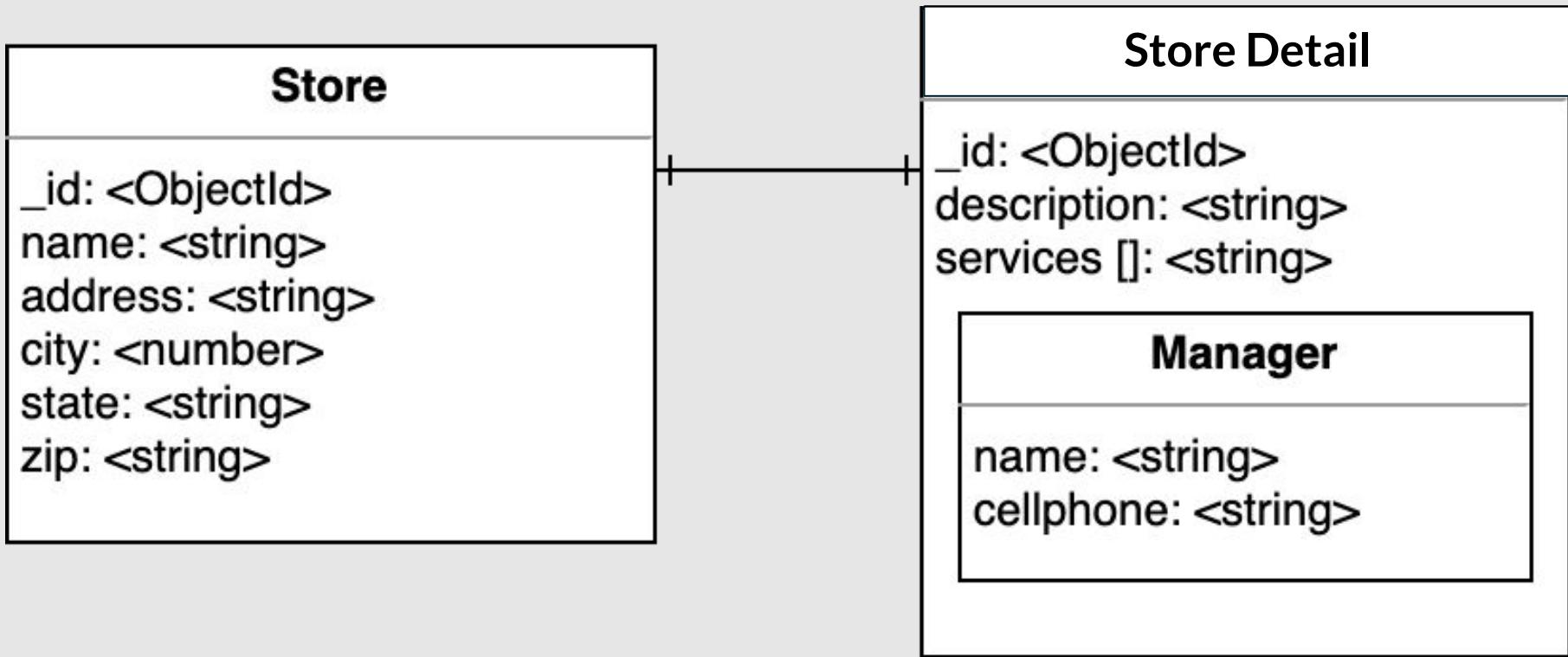
1 - 1





16 MB





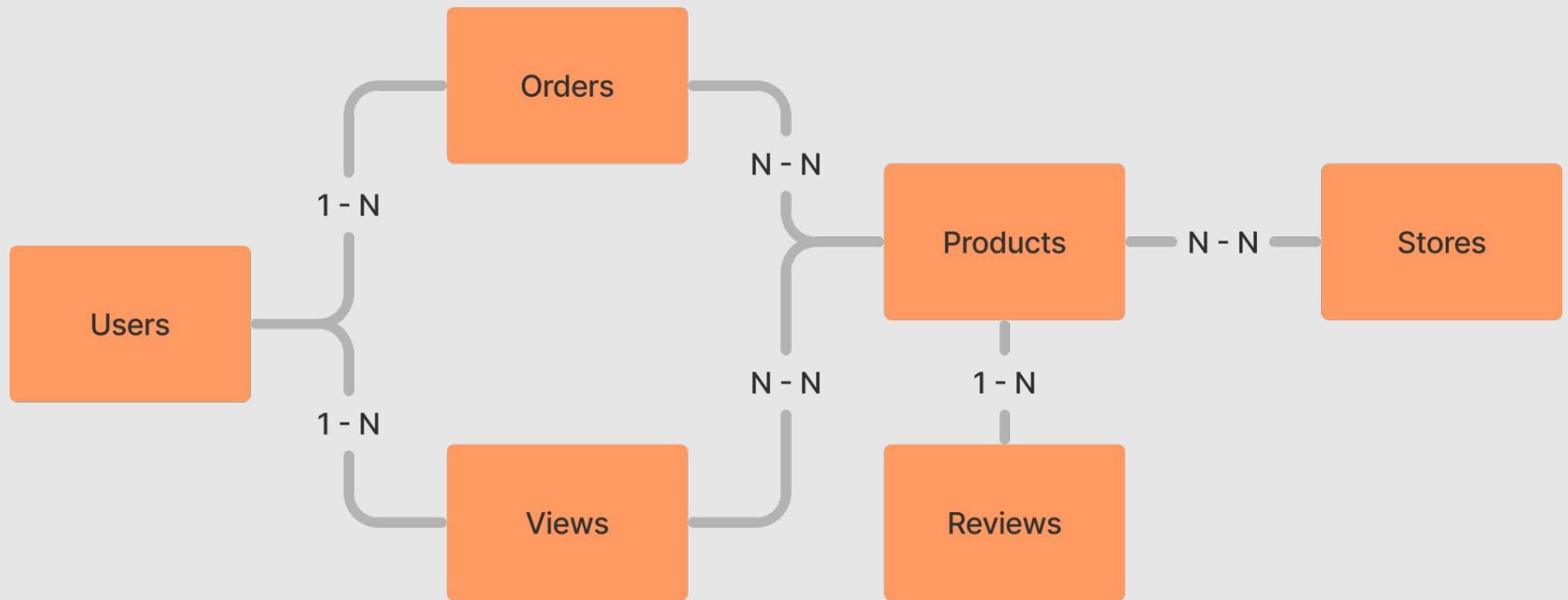
Cuando el doc es **muy pesado** y hace que la consulta demore

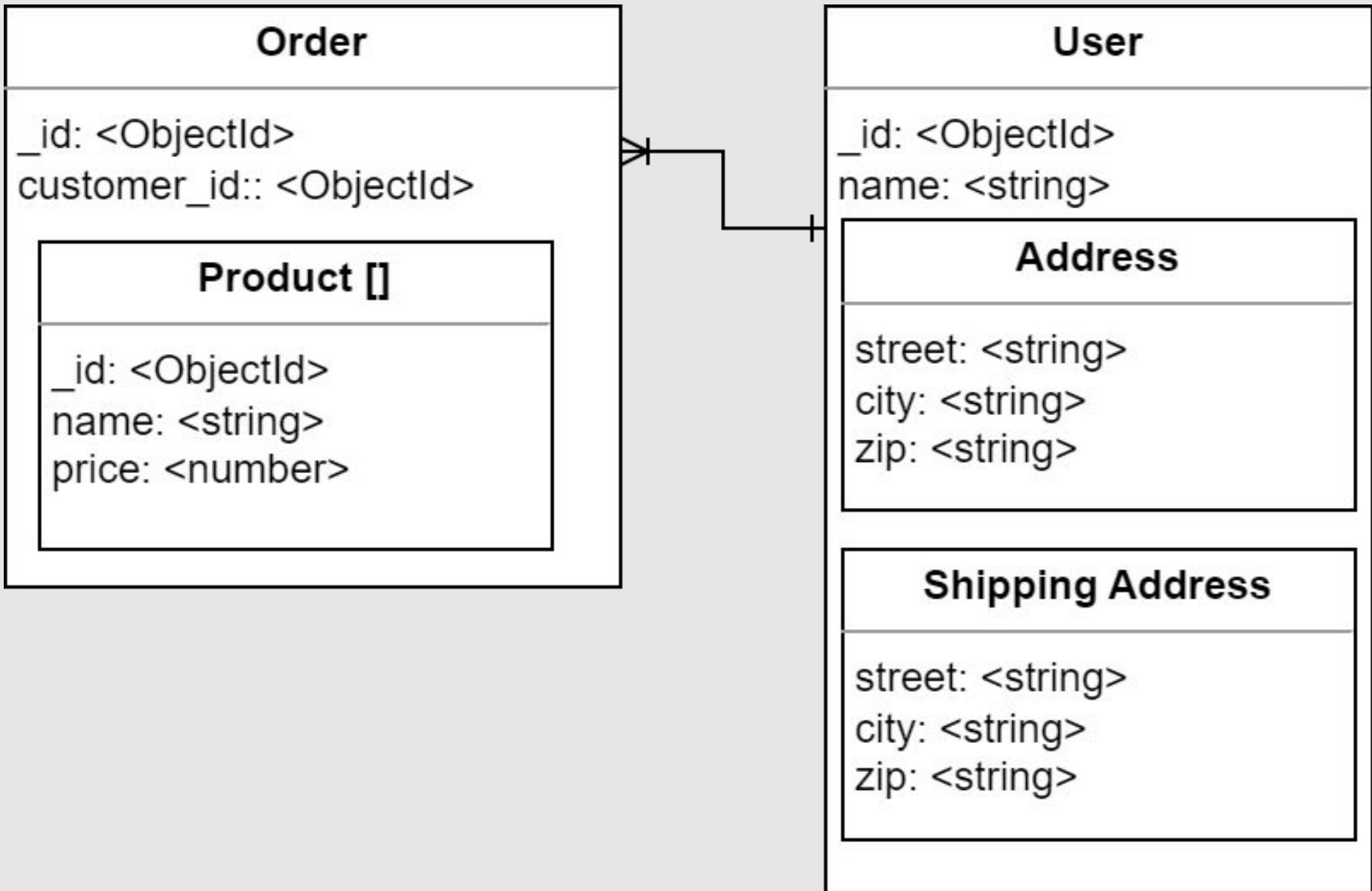
```
{  
  "name": "Nicolas",  
  "email": "nico@gmail.com",  
  "address": {  
    "city": "Bogotá",  
    "zip": "111011",  
    "street": "Cr 324"  
    ... //(16MB)  
  }  
}
```



Relaciones 1 - N







Cuando la información es consultada en conjunto

```
{  
  "bookId": "book01",  
  "name": "Harry Potter",  
  "covers": [  
    { "type": "front", "image": "url" },  
    { "type": "back", "image": "url" }  
  ]  
}
```

Cuando los sub docs son una dependencia

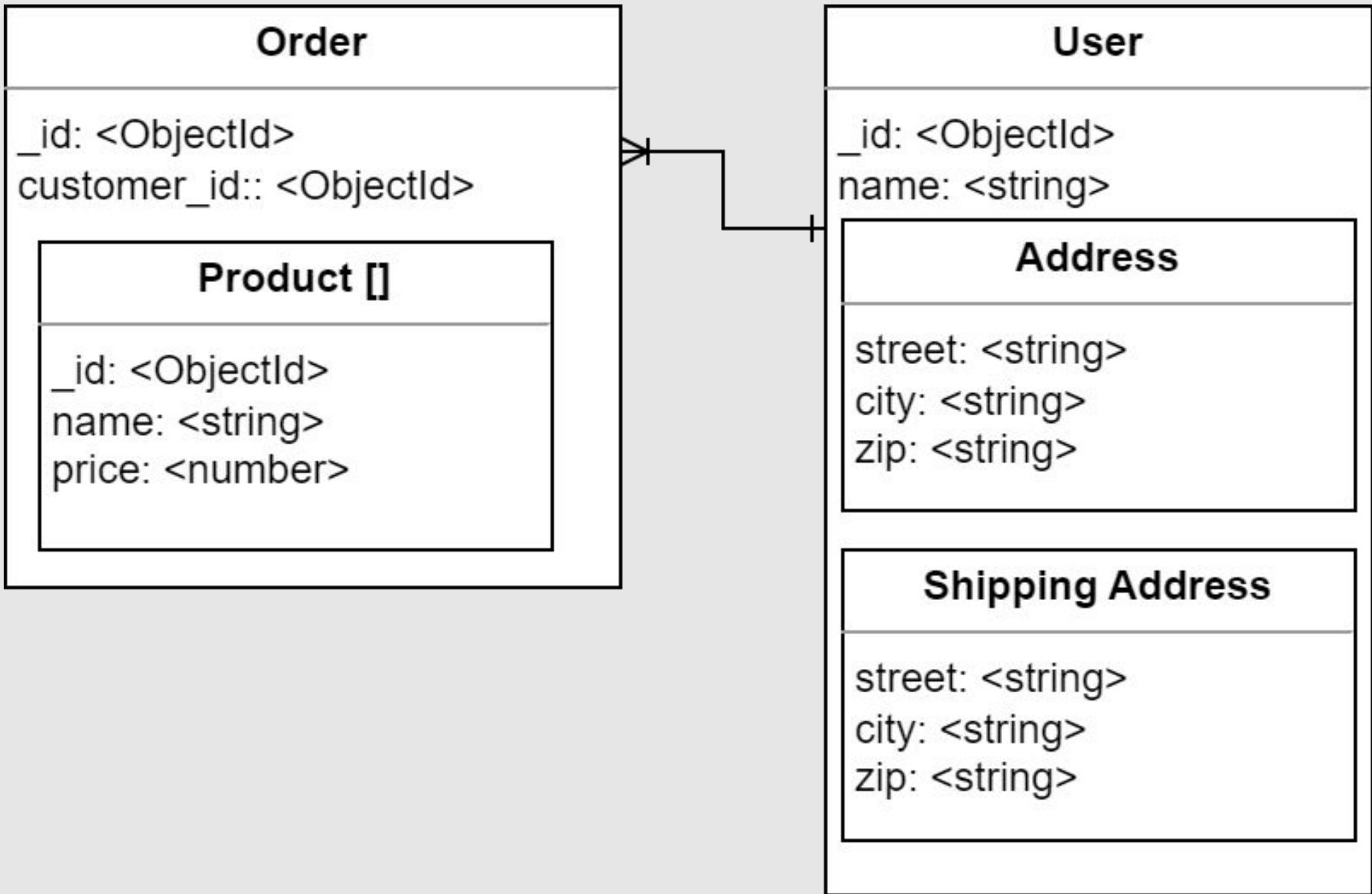
```
{  
  "bookId": "book01",  
  "name": "Harry Potter",  
  "covers": [  
    { "type": "front", "image": "url" },  
    { "type": "back", "image": "url" }  
  ]  
}
```

Cuando los sub docs son 1:few

```
{  
  "bookId": "book01",  
  "name": "Harry Potter",  
  "covers": [  
    { "type": "front", "image": "url" },  
    { "type": "back", "image": "url" }  
  ]  
}
```

Relaciones 1 - N





Usa referencia cuando la relación puede incrementar considerablemente

```
{  
  "bookId": "book01",  
  "name": "Harry Potter",  
  "comments": [  
    { "date": "2020-20-12", "comment": "wow genial!!" },  
    { "date": "2020-20-32", "comment": "mejor las pelis" },  
    { "date": "2020-20-22", "comment": "Harry es increible" },  
    ...  
    { "date": "2020-20-22", "comment": "F por Severus Snape" },  
  ]  
}
```



Usa referencia cuando la entidad se actualiza de forma constante

```
[  
 {  
   "orderId": "ORDER-01",  
   "customer": {  
     "id": "12",  
     "address": "Cr 23 sd"  
   }  
,  
 {  
   "orderId": "ORDER-01",  
   "customer": {  
     "id": "12",  
     "address": "Cr 23 sd"  
   }  
 }  
]
```

Usa referencia cuando la entidad es usada por muchos documentos

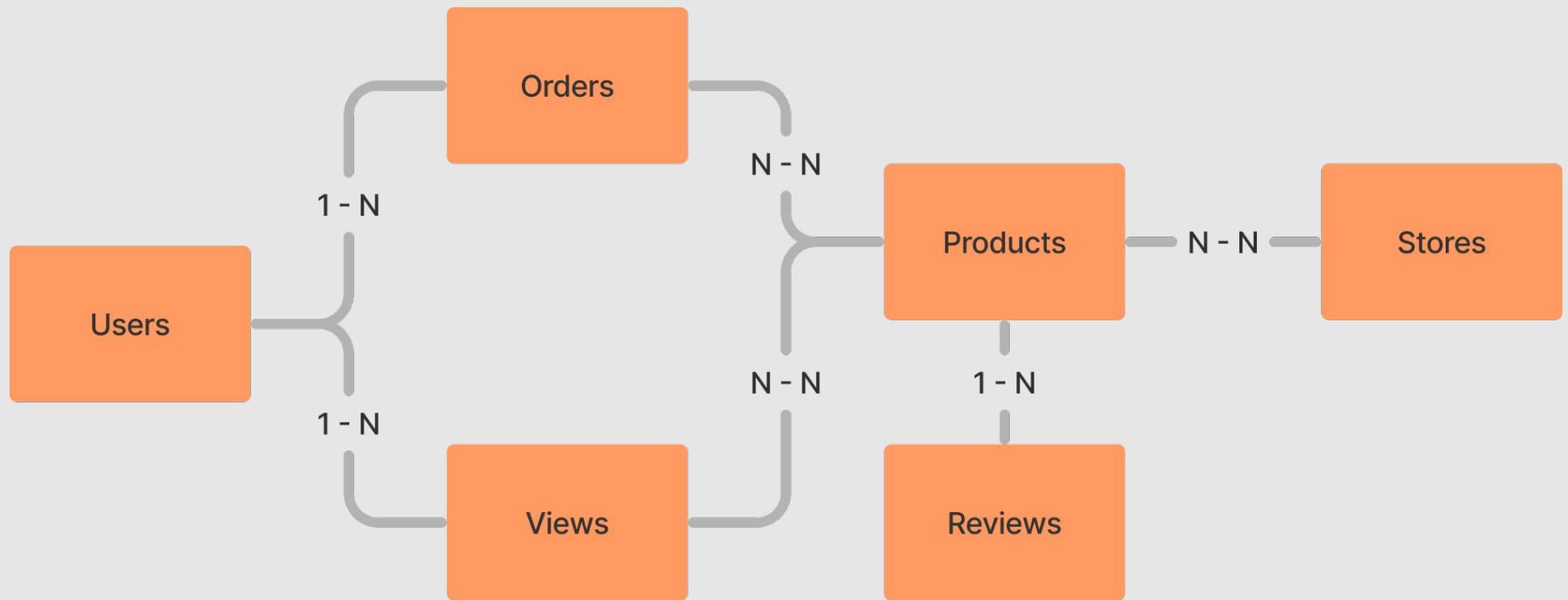
```
[  
 {  
   "orderId": "ORDER-01",  
   "customer": {  
     "id": "12",  
     "address": "Cr 23 sd"  
   }  
,  
 {  
   "orderId": "ORDER-01",  
   "customer": {  
     "id": "12",  
     "address": "Cr 23 sd"  
   }  
 }  
]
```

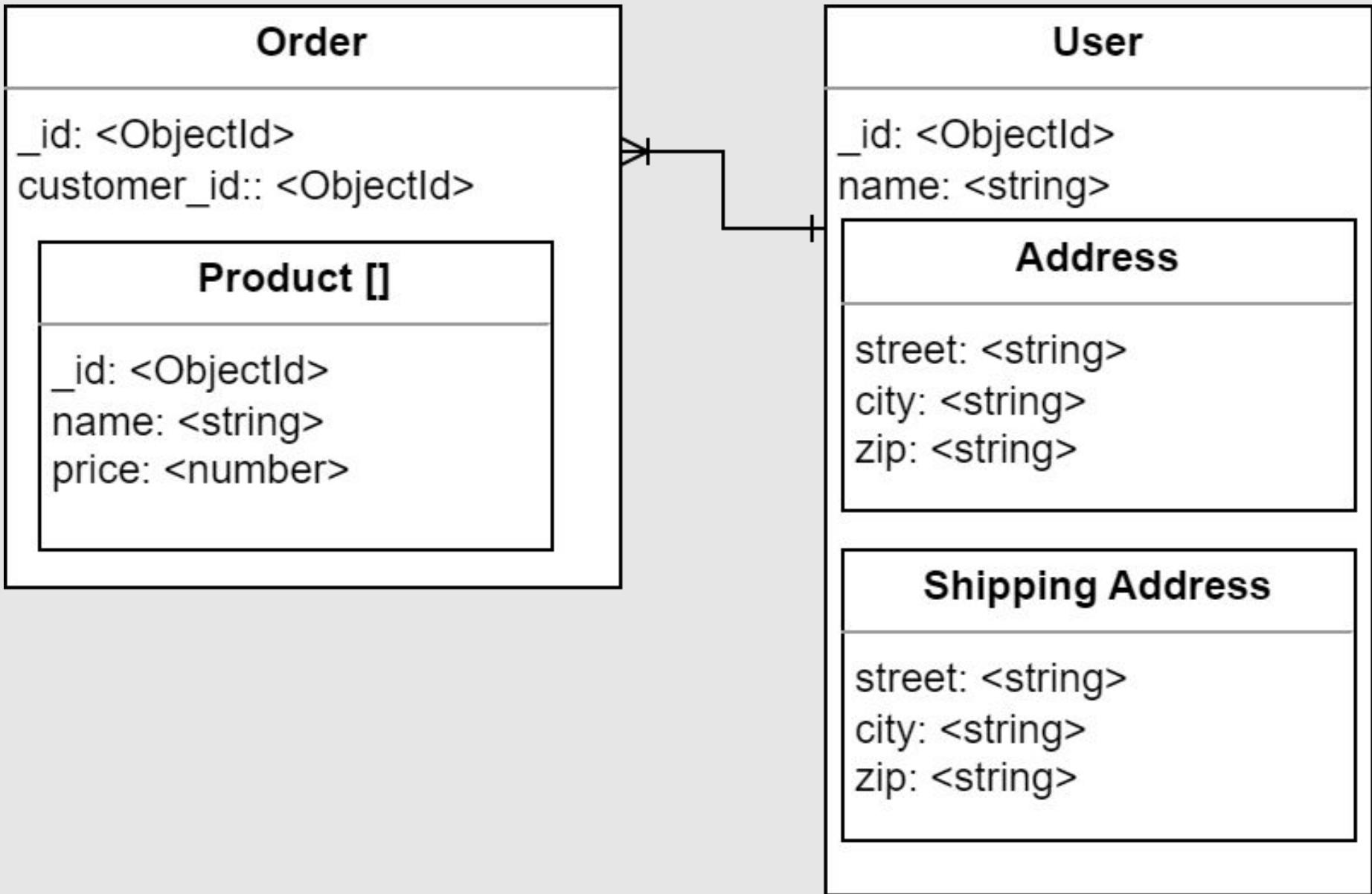
Usa referencia cuando la entidad se actualiza de forma constante (también aplica para 1:1)

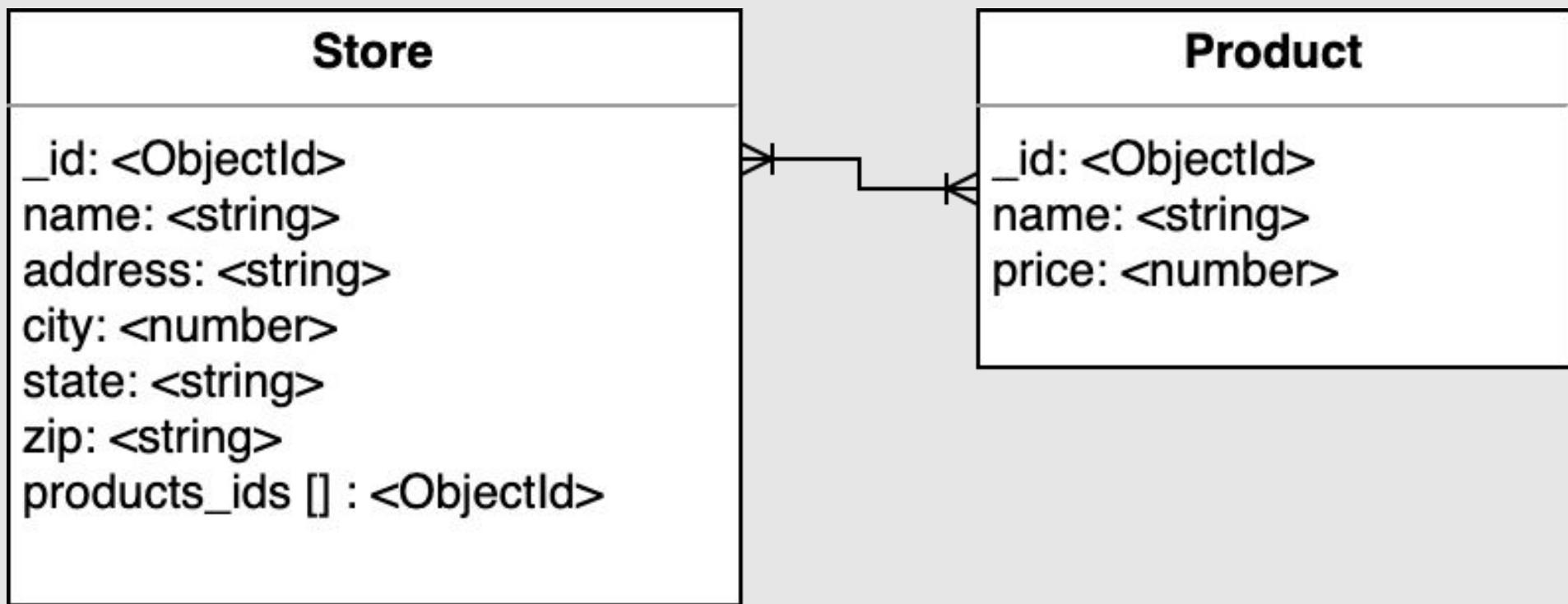
```
[  
  {  
    "talk": "WebAssembly es el futuro!!",  
    "room": { "id": 201, "cap": 200, "recording": true }  
  },  
  {  
    "talk": "WebAssembly es el futuro!!",  
    "room": { "id": 201, "cap": 200, "recording": true }  
  }  
]
```

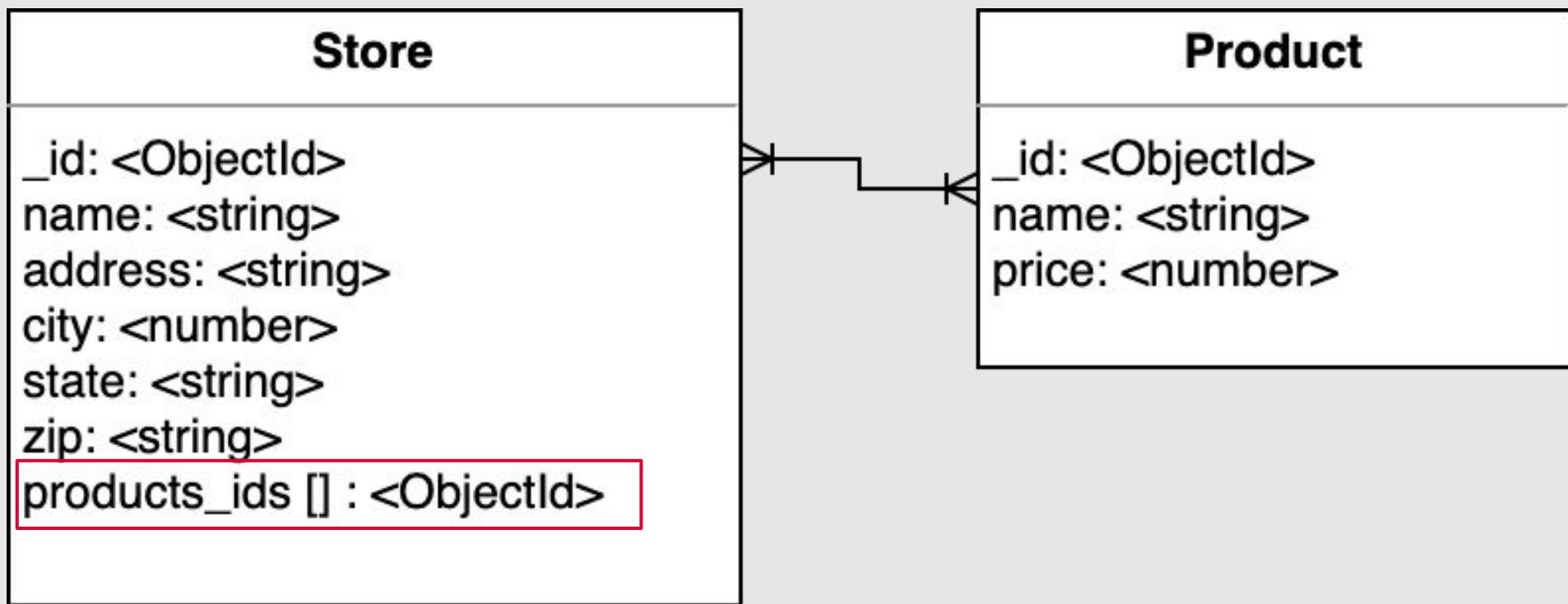
Relaciones N - N

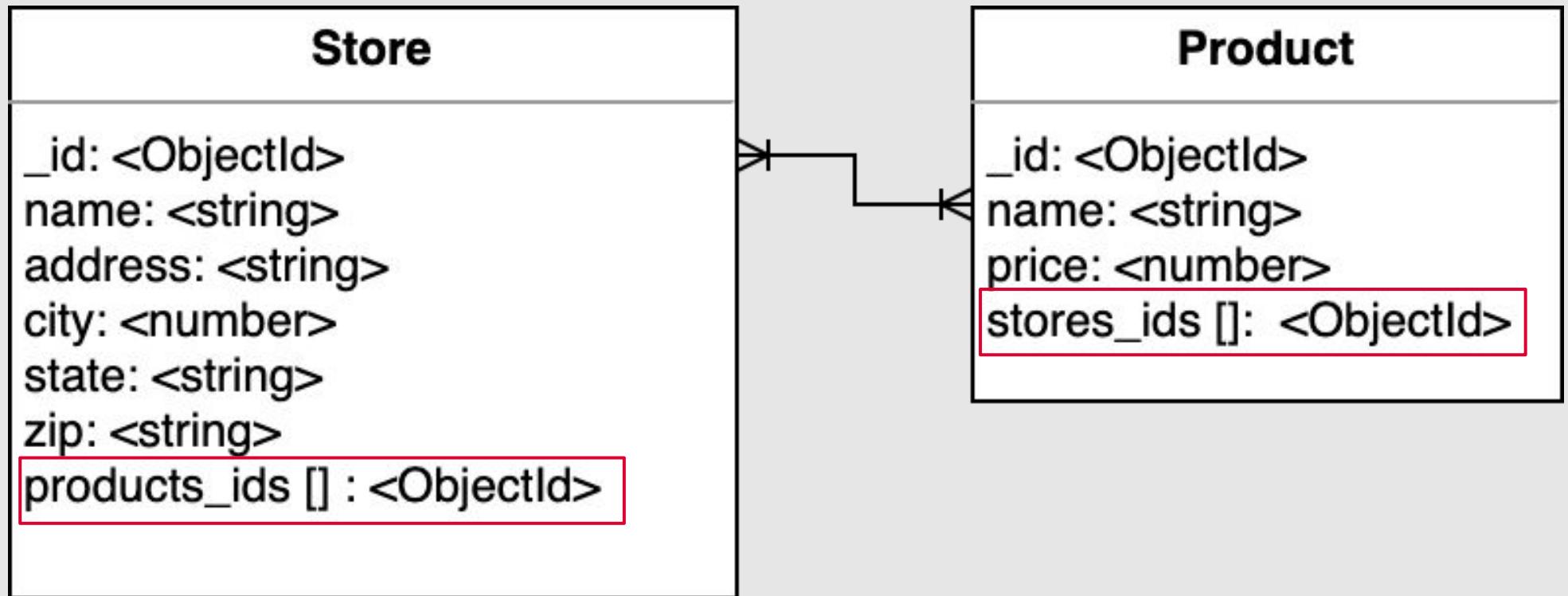










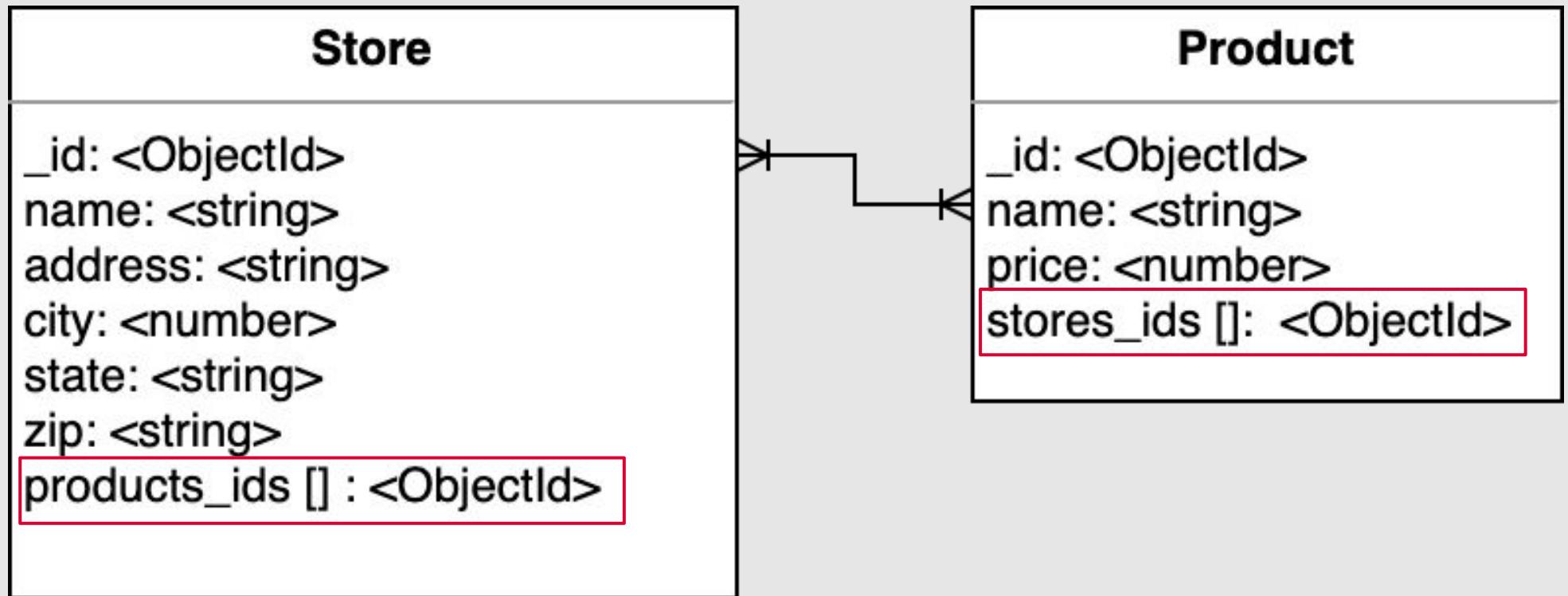


U^sa referencia cuando la **relación es n -n**

```
{  
  "title": "La Web 3.0",  
  "speakers_ids": [...],  
},
```

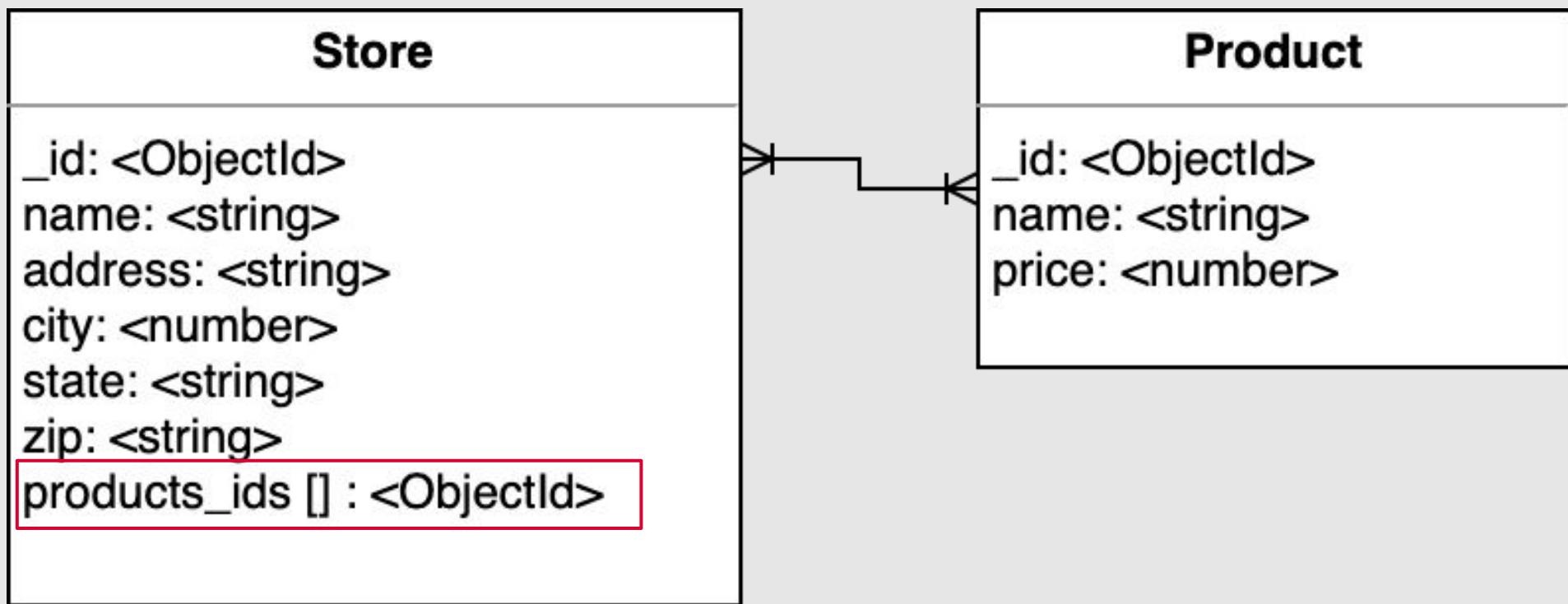
```
{  
  "name": "Nicolas Molina",  
  "sessions_ids": [...],  
}
```

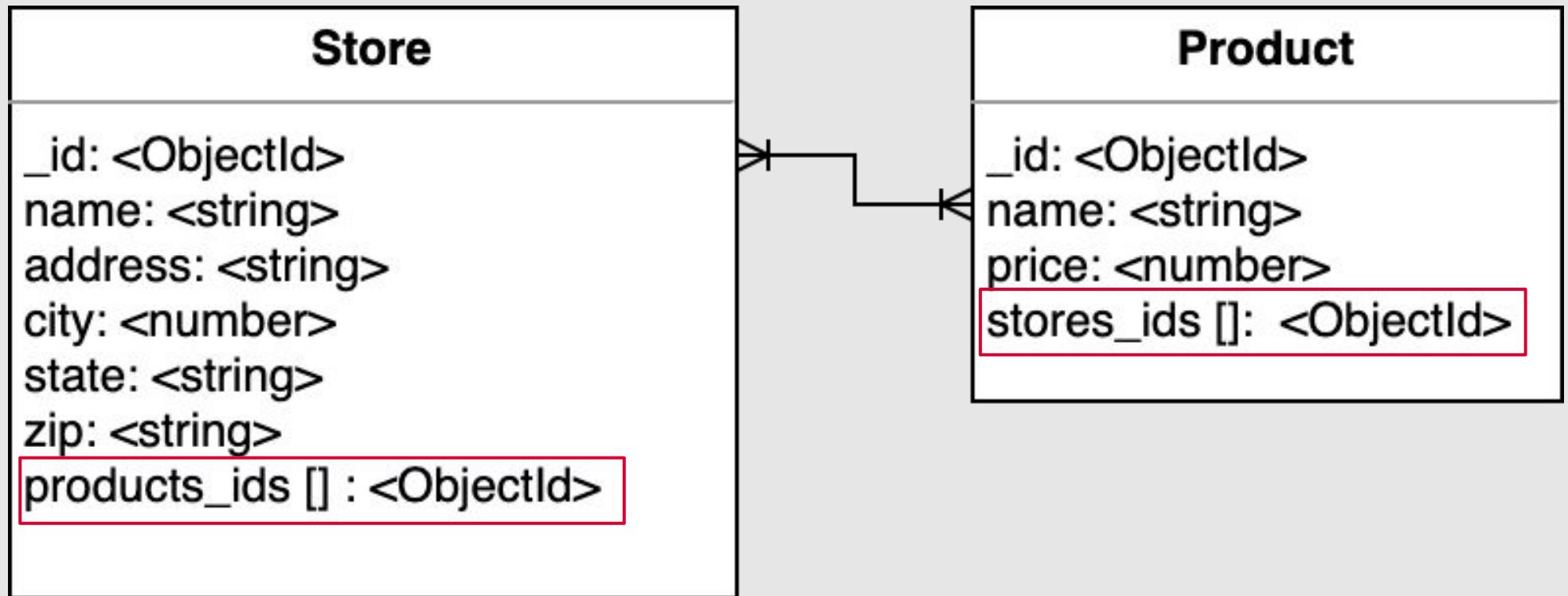




Relaciones N - N







U^sa referencia cuando la **relación es n -n**

```
{  
  "title": "La Web 3.0",  
  "speakers_ids": [...],  
},
```

```
{  
  "name": "Nicolas Molina",  
  "sessions_ids": [...],  
}
```



Cuando la información es consultada en conjunto

```
{  
  "title": "La Web 3.0",  
  "speakers_ids": [...],  
},  
  
{  
  "name": "Nicolas Molina",  
  "sessions_ids": [...],  
}
```



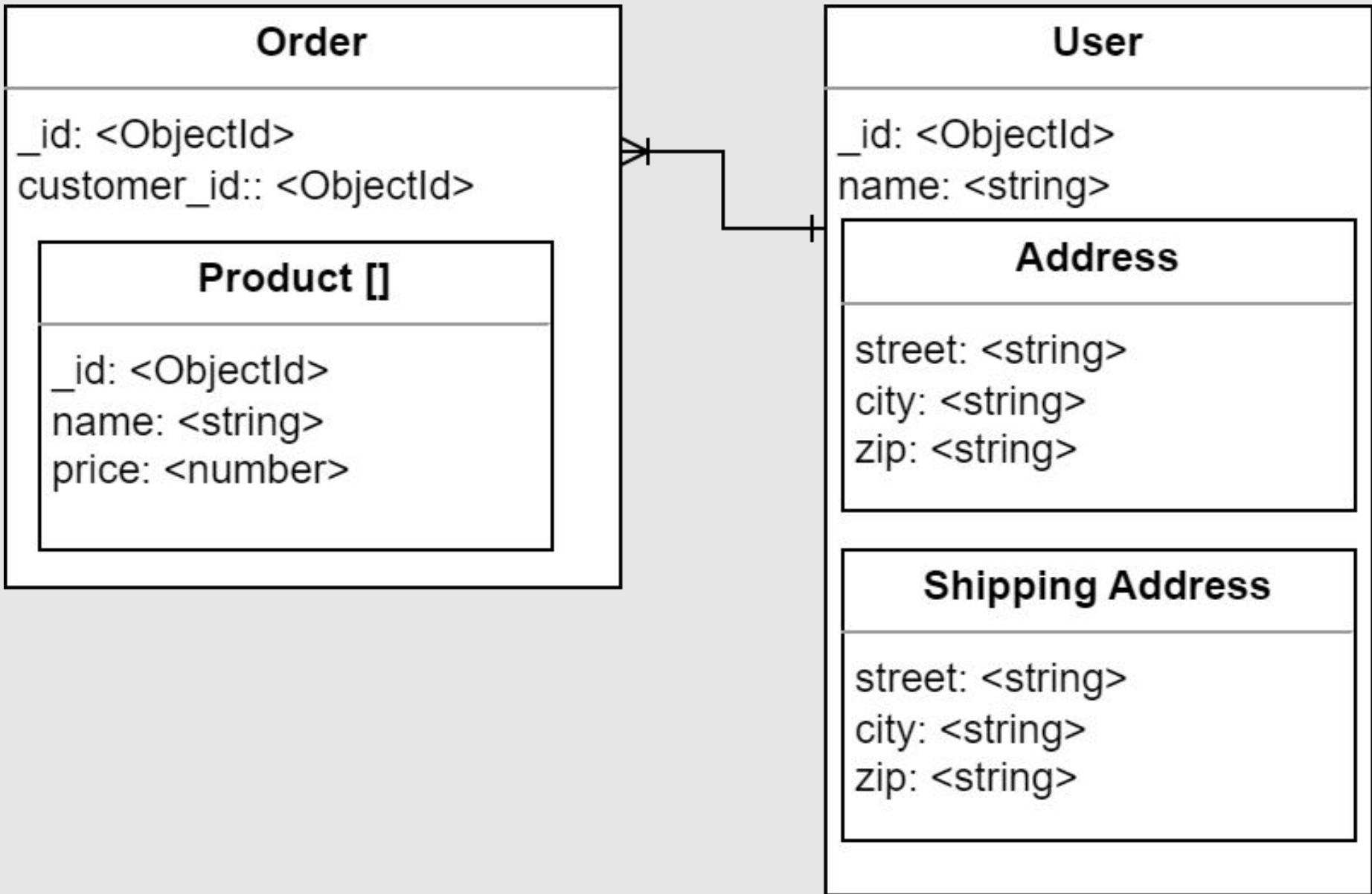
Aplicar Desnormalización

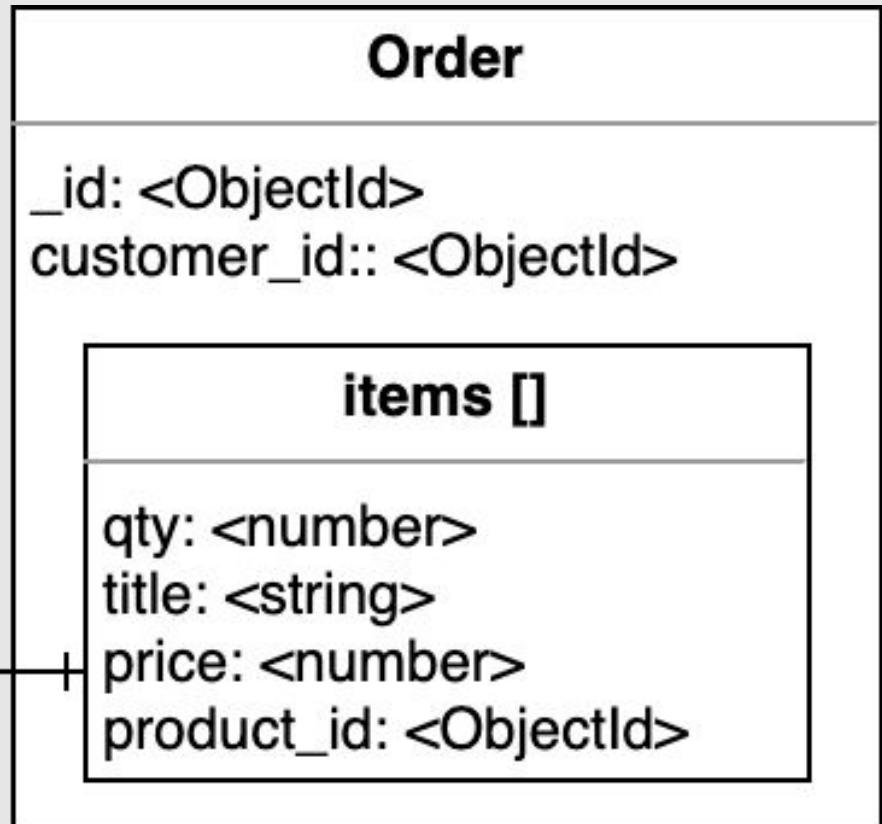
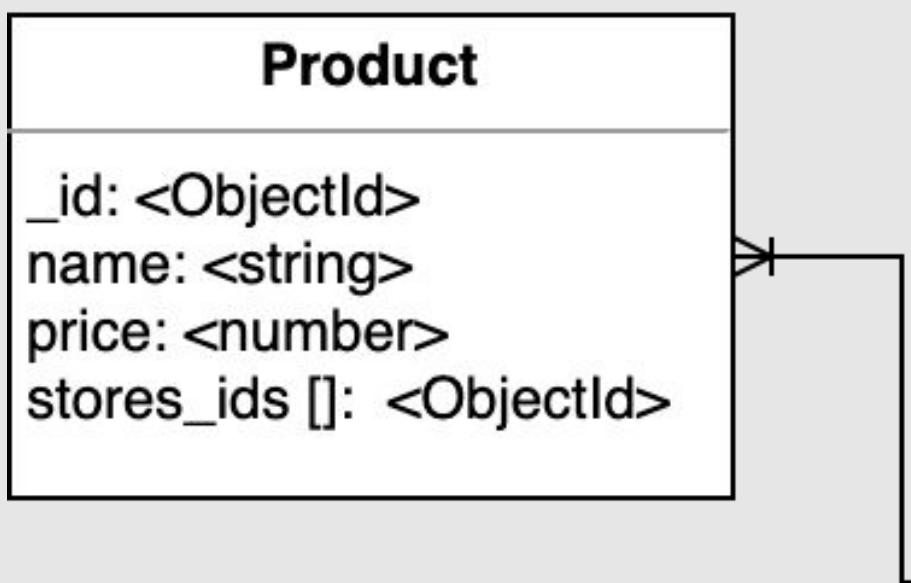


Desnormalización

Es el proceso de optimizar el funcionamiento de una DB agregando datos redundantes.







Computed Pattern



Metodología

1. Requerimientos (Workload)
2. Identificar ER
3. Aplicar patrones

Computed Pattern



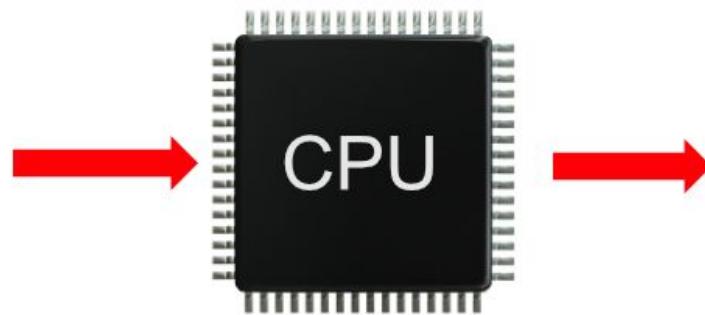
Screening Information

```
{  
    "ts": DateTime(XXX),  
    "theater": "Alger Cinema",  
    "location": "Lakeview, OR",  
    "movie_title": "Jack Ryan: Shadow Recruit",  
    "num_viewers": 344,  
    "revenue": 3440  
}
```

```
{  
    "ts": DateTime(XXX),  
    "theater": "City Cinema",  
    "location": "New York, NY",  
    "movie_title": "Jack Ryan: Shadow Recruit",  
    "num_viewers": 1496,  
    "revenue": 22440  
}
```

```
{  
    "ts": DateTime(XXX),  
    "theater": "Overland Park Cinema",  
    "location": "Boise, ID",  
    "movie_title": "Jack Ryan: Shadow Recruit",  
    "num_viewers": 760,  
    "revenue": 7600  
}
```

Movie Information



```
{  
    "ts": DateTime(XXX),  
    "title": "Jack Ryan: Shadow Recruit",  
    "viewers": 2600,  
    "revenue": 33480  
}
```

Computed Pattern

- Cuando necesitas calcular un valor a partir de otros campos en un documento y mostrar el resultado en una consulta o informe.
- Cuando es más frecuente la lectura que la escritura.
- Calcular el valor por cada lectura es costoso, es mejor pre calcular en cada escritura.

Use Case Categories

Patterns

- Approximation
- Attribute
- Bucket
- Computed
- Document Versioning
- Extended Reference
- Outlier
- Preallocated
- Polymorphic
- Schema Versioning
- Subset
- Tree and Graph

Catalog	Content Management	Internet of Things	Mobile	Personalization	Real-Time Analytics	Single View
✓		✓	✓		✓	
✓	✓					✓
		✓			✓	
✓		✓	✓	✓	✓	✓
✓	✓			✓		✓
✓			✓		✓	
		✓	✓	✓		
		✓			✓	
✓	✓		✓			✓
✓	✓	✓	✓	✓	✓	✓
✓	✓		✓	✓		
✓	✓			✓		

Simplicidad vs. Rendimiento



Simplicidad

Rendimiento



Simplicidad

Rendimiento



Simplicidad

Rendimiento



Small Team



Embed > Reference



Frequent Queries

Simplicidad

Rendimiento



Large Team



Embed & Reference



Frequent Queries & Patterns



Metodología

1. Requerimientos (Workload)
2. Identificar ER
3. Aplicar patrones

Curso de Modelado de Datos con MongoDB





MongoDB®



 MongoDB®

```
from pymongo import MongoClient  
  
client = MongoClient(port=27017)  
  
db=client.business  
  
fivestarcount = db.reviews.find({'rating': 5}).count()  
  
result = db.reviews.update_one({...})
```



```
const { MongoClient } = require("mongodb");

const client = new MongoClient("uri");

const database = client.db('sample_mflix');

const movies = database.collection('movies');

const query = { title: 'Back to the Future' };
const movie = await movies.findOne(query);
```





 MongoDB®



Nicolas Molina
@nicobytes