

# Taller de MongoDB

## Sesión 2

# Contenido

- Introducción ✓
- Configuración ATLAS y Compass ✓
- JSON/BSON ✓
- Operaciones CRUD
- Tipos de data
- Relaciones y Modelado
- Validación de esquema
- Introducción al aggregation framework
- Sigüientes pasos



# Query documents

## Basic Find

```
db.collection.find( { attribute: "value" } )
```

## Using In Operator

```
db.collection.find( { attribute: { $in: [ "value1", "value2" ] } } )
```

## Using And Operator

```
db.collection.find( { attribute1: "value1" , attribute2 : "value2"} )
```

## Using Or Operator

```
db.collection.find({ $or: [ { attribute: "value" }, { attribute : "value"} ] } )
```

# Query Embedded Documents

## Exact Match Find

```
db.collection.find( { attribute: { "innerAttribute1": "value", "innerAttribute2": "value2" } } )
```

## Dot Notation

```
db.collection.find( { "attribute.innerAttribute": "value" } )
```

# Query Arrays

## Exact Match Array Find

```
db.collection.find( { attribute: [ "value1", "value2" ] } )
```

## All elements must be in Array

```
db.collection.find( { attribute: { $all: [ "value1", "value2" ] } } )
```

## Multiple criteria in Array Find

```
db.collection.find( { attribute: { $elemMatch: { $gt: 25, $lt: 30 } } } )
```

## Especific position in Array Find

```
db.collection.find( { "attribute.0": "value" } )
```

## Array length Find

```
db.collection.find( { attribute: { $size: 10 } } )
```

# Query Arrays of Embedded Documents

## Exact Match Array Embedded Document Find

```
db.collection.find( { attribute: { "innerAttribute1": "value", "innerAttribute2": "value2" } })
```

## Dot Notation for Condition

```
db.collection.find( { "attribute.innerAttribute" : { $gte: 10 } })
```

## Especific position in Array with Embedded Attribute Find

```
db.collection.find( { "attribute.0.innerAttribute": "value" })
```

## Multiple criteria for One Embedded

```
db.collection.find( { attribute: { $elemMatch: { "innerAttribute1": "value", "innerAttribute2": "value2" } } })
```

# Projection

## Return specific attributes

```
db.collection.find( { attribute1: "value1"},{attribute1: 1, attribute2: 1} )
```

## Avoid \_id

```
db.collection.find( { attribute1: "value1"},{attribute1: 1, attribute2: 1, _id: 0} )
```

## All but avoid fields

```
db.collection.find( { attribute1: "value1"},{attribute1: 0, attribute2: 0 } )
```

# Update

## Operations

```
db.collection.updateOne({/**Some query**/},{$set{attribute: "value"}})
```

```
db.collection.updateMany({/**Some query**/},{$set{attribute: "value"}})
```

## Aggregation Pipeline usage

- **\$set**
- **\$addField**



# Delete

## Operations

```
db.collection.deleteOne({/**Some query**/})
```

```
db.collection.deleteMany({/**Some query**/})
```

# Data types

- String
- Integer
- Boolean
- Double
- Arrays
- Timestamp
- Object
- Null
- Date
- Object ID
- Binary data
- Code
- Regular expression
- 32-bit integer
- 64-bit integer
- Decimal128
- Min key
- Max key

Revisar

# Modelado

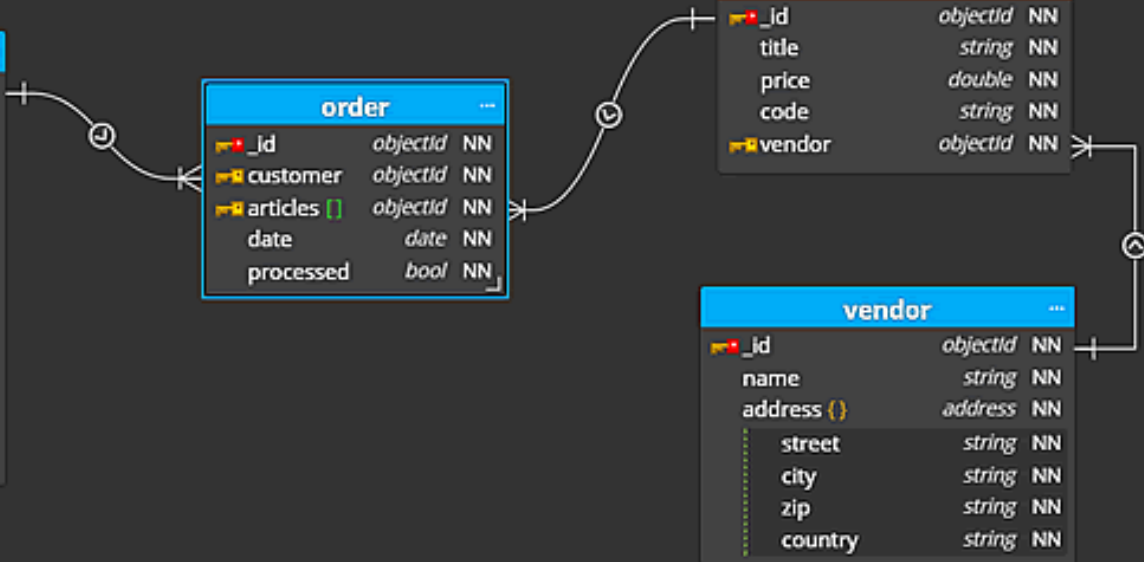
## MongoDB Collections

customer			
...			
🚩 _id	objectId	NN	
name	string	NN	
surname	string	NN	
contact {}	contact	NN	
addresses [{}]	address	NN	
street	string	NN	
city	string	NN	
zip	string	NN	
country	string	NN	
phones [{}]	phone	NN	
mobile	string	NN	
active	int	NN	

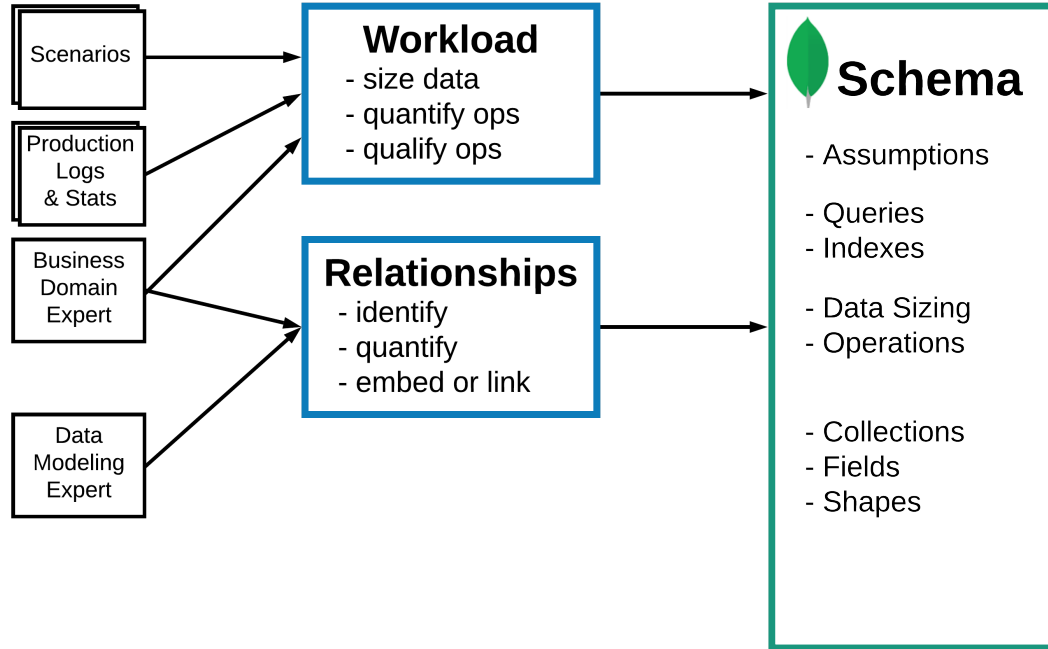
order			
...			
🚩 _id	objectId	NN	
customer	objectId	NN	
articles []	objectId	NN	
date	date	NN	
processed	bool	NN	

article			
...			
🚩 _id	objectId	NN	
title	string	NN	
price	double	NN	
code	string	NN	
vendor	objectId	NN	

vendor			
...			
🚩 _id	objectId	NN	
name	string	NN	
address {}	address	NN	
street	string	NN	
city	string	NN	
zip	string	NN	
country	string	NN	



# Metodología



# Relations

## Common relations

- One-to-One
- One-to-Many
- Many-to-Many

## Special relation

- One-to-Zillions

# Patrones

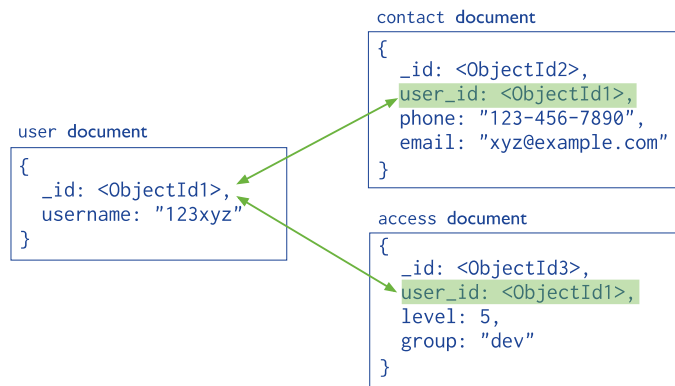
- MongoDB Manual

## Reference vs Embedded

### EMBEBIDO



### REFERENCIADO



- MongoDB Manual