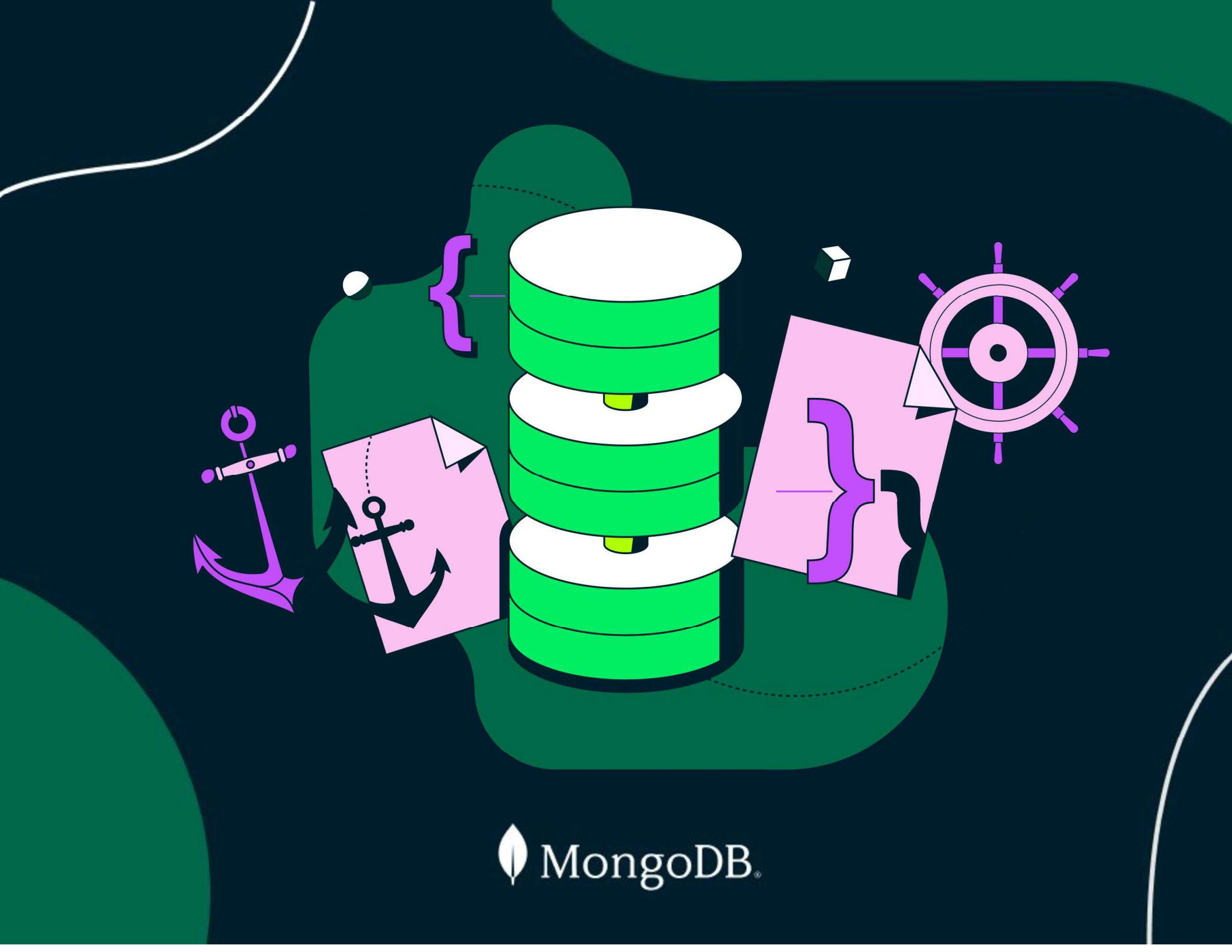
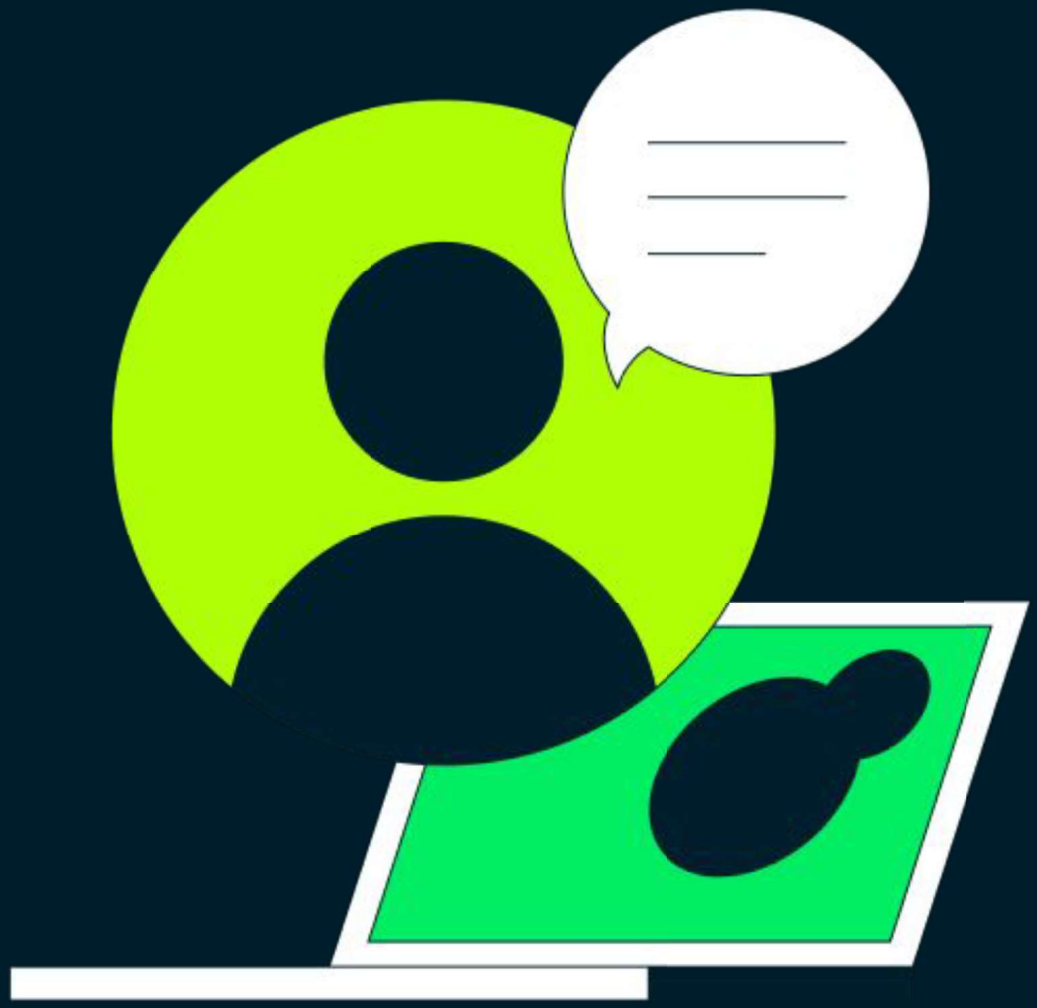

MongoDB®

¿Qué es MongoDB?


MongoDB.

Curso de Modelado de Datos con MongoDB

MongoDB.



MongoDB.

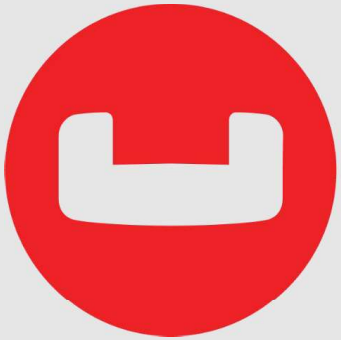
Bases de datos NoSQL

MongoDB.

Documentales

En estas bases de datos se empareja cada clave con una estructura de datos compleja que se denomina 'documento'.

Cloud Firestore

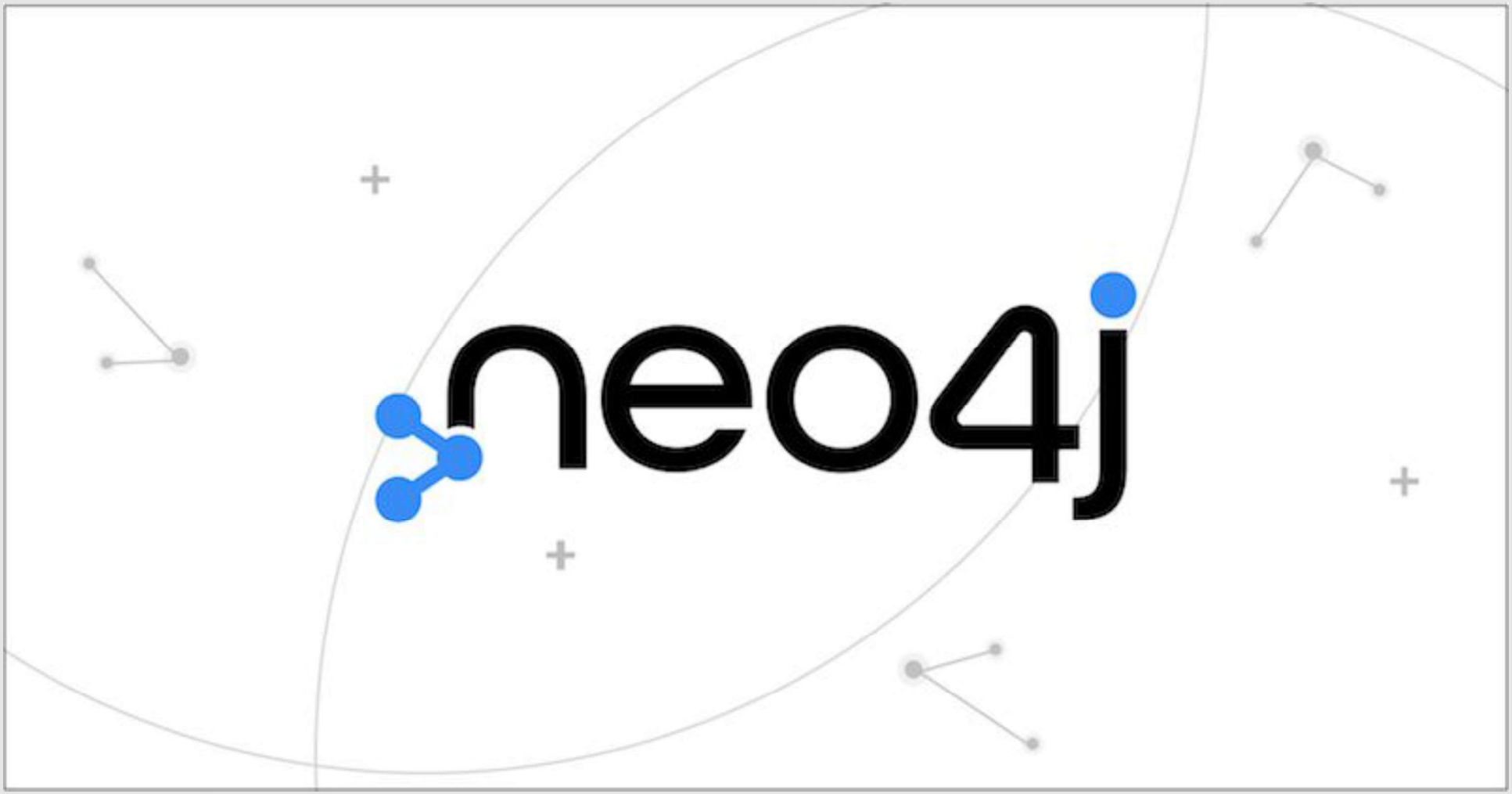


Couchbase

Grafos

Se utilizan para almacenar información sobre redes de datos, como las conexiones sociales.

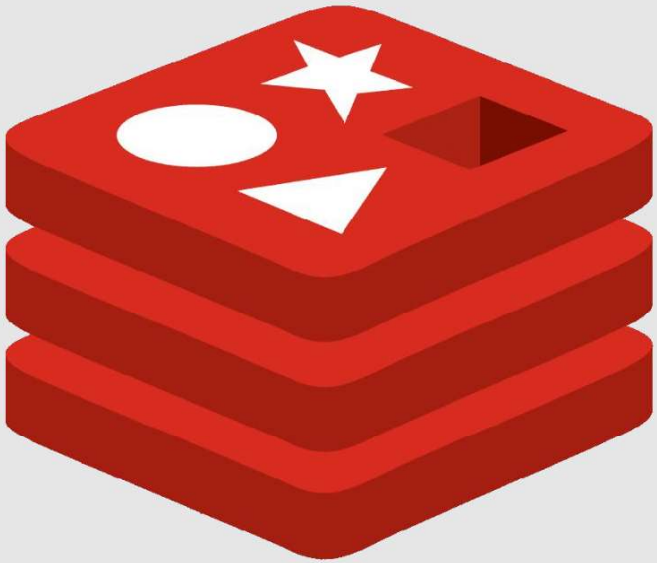
MongoDB.



MongoDB.

Clave - Valor

Son las bases de datos **NoSQL** más simples. Cada elemento de la base de datos se almacena como un nombre de atributo (o «clave»), junto con su valor.

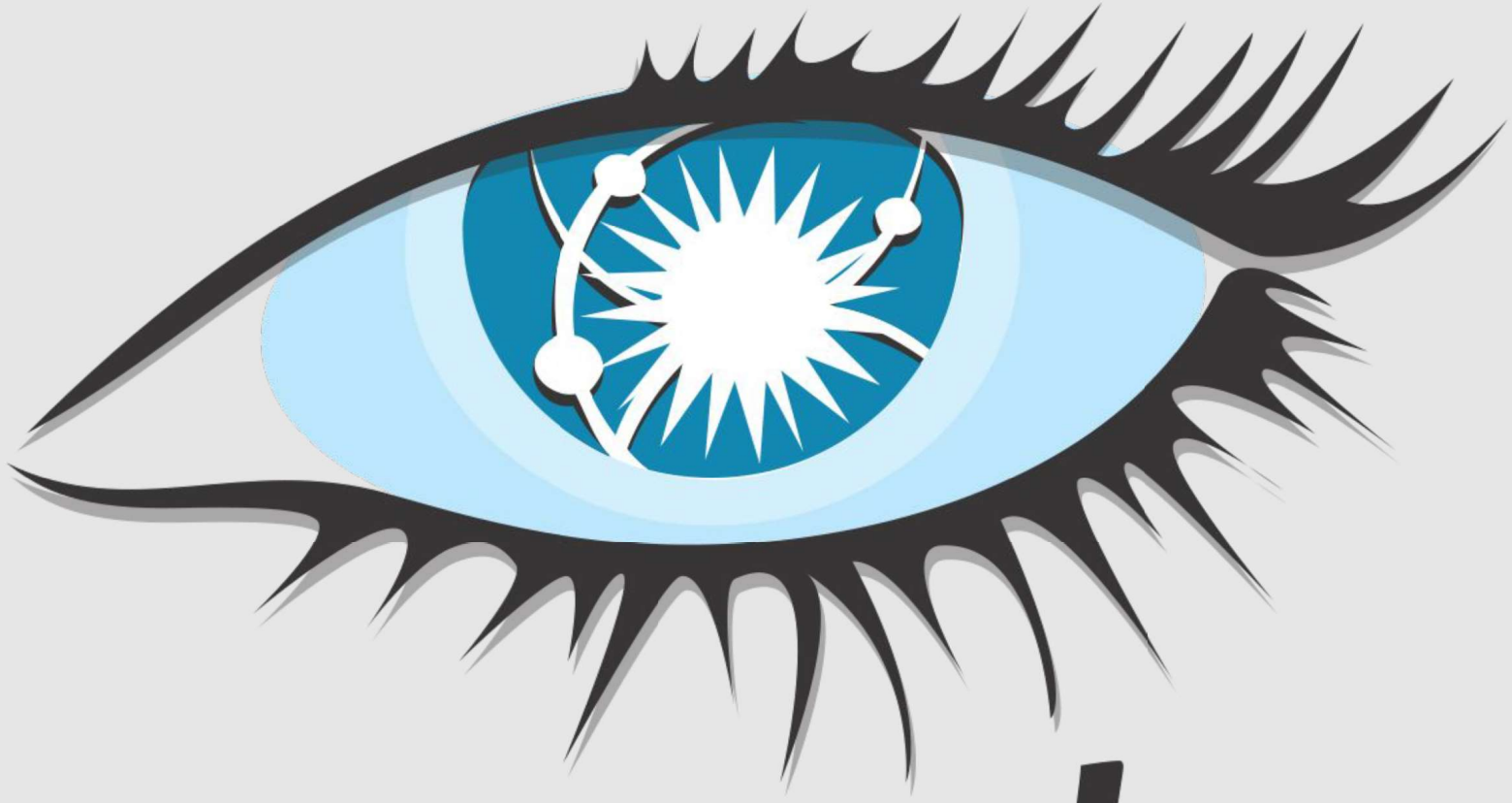


redis

MongoDB.

Orientadas a Columnas

Estas bases de datos, como Cassandra o HBase, permiten realizar consultas en grandes conjuntos de datos y almacenan los datos en columnas, en lugar de filas.

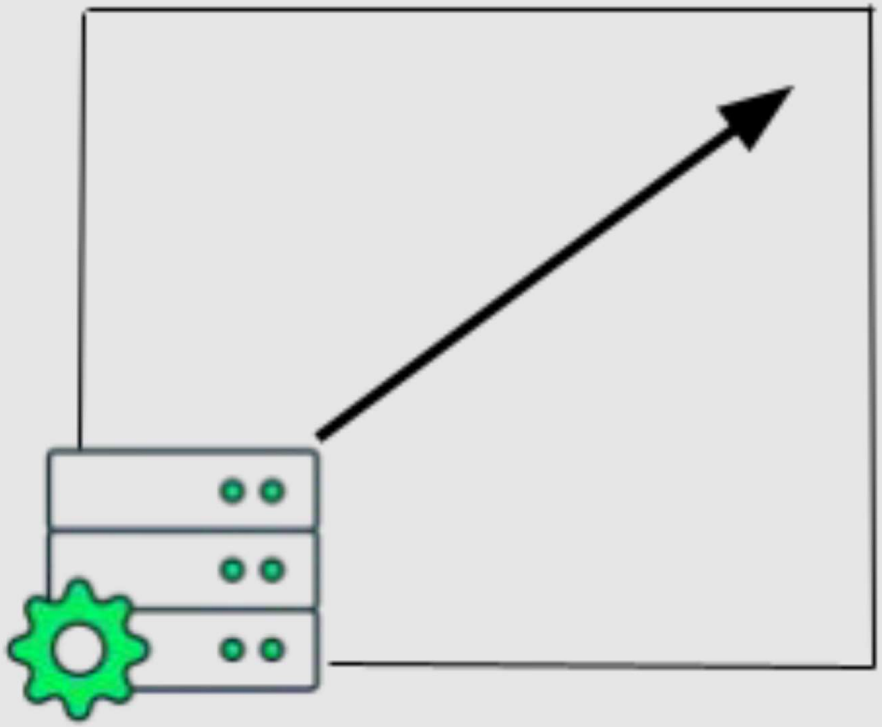


cassandra

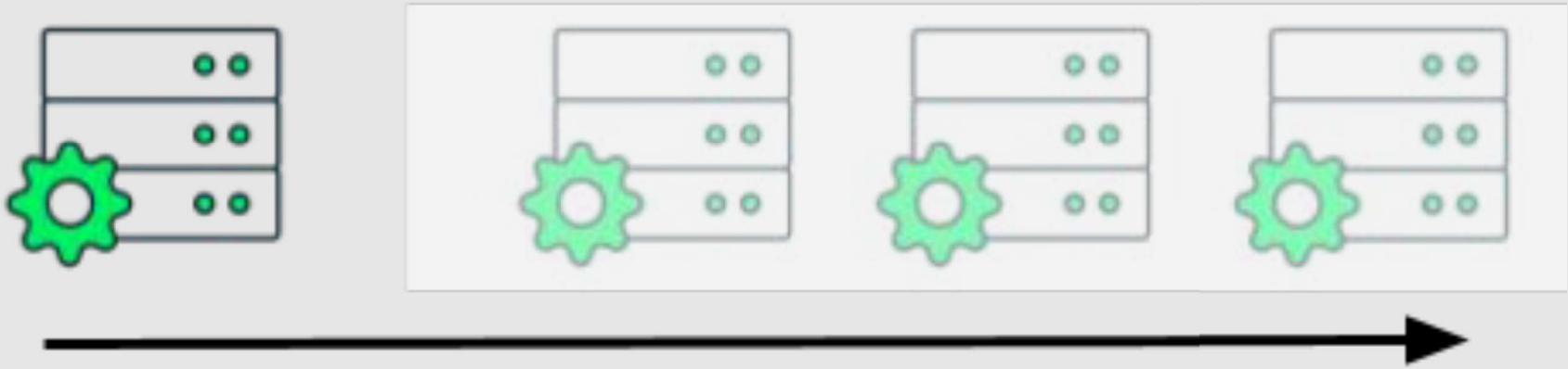
MongoDB.

Escalamiento horizontal

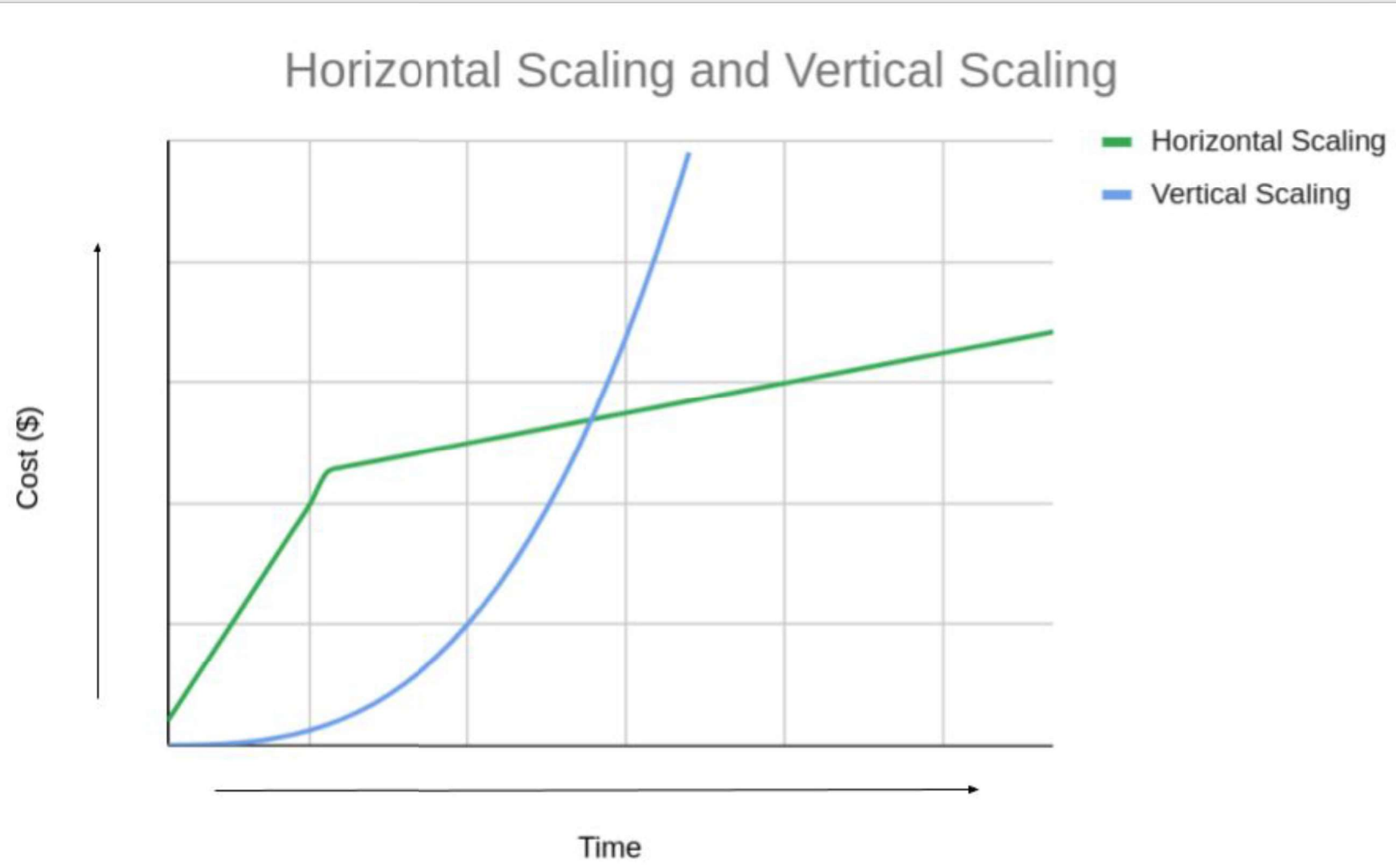
MongoDB.



MongoDB.



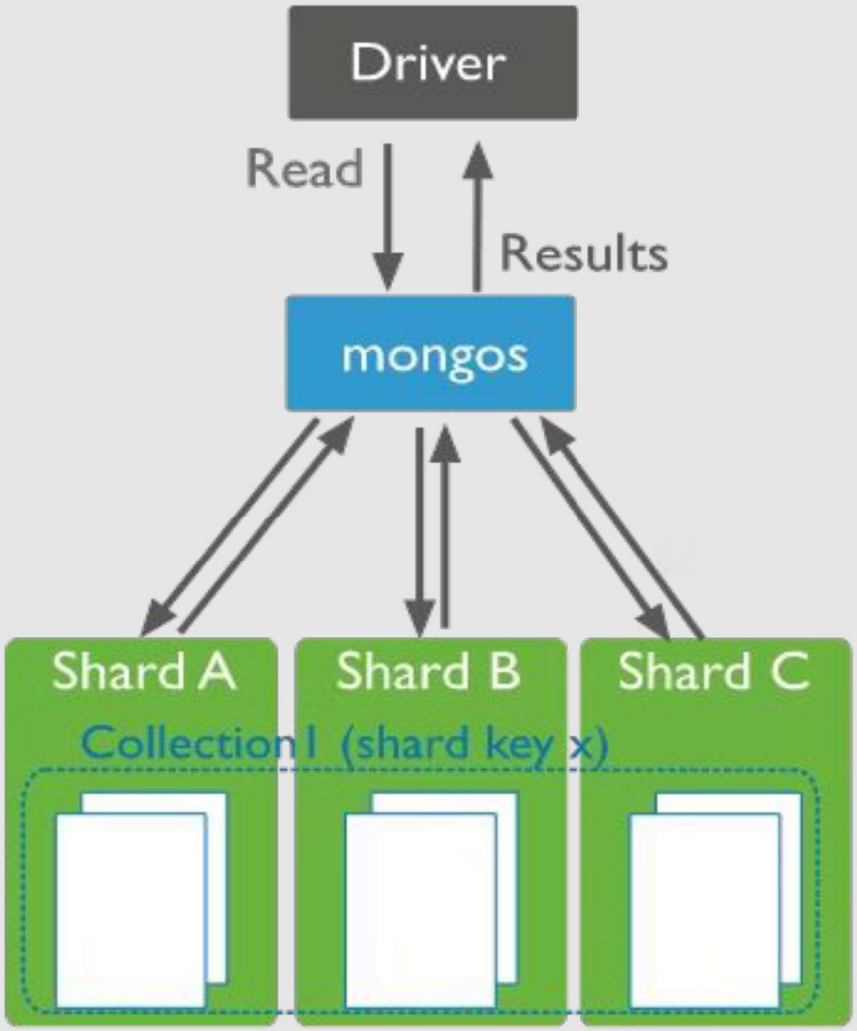
MongoDB.


Horizontal Scaling and Vertical Scaling
Horizontal Scaling
Vertical Scaling
Cost ($)
Time

MongoDB.

¿Qué es la Replicación?

MongoDB.


Driver
Read
Results
mongos
Shard A
Shard B
Shard C
Collection1 (shard key x)

MongoDB.

¿Qué son los documentos?

MongoDB.

MongoDB.

Document

Una forma de organizar y almacenar información con un conjunto de pares **clave-valor**.

MongoDB.


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

MongoDB.

```
{
  "_id": "5c8eccc1caa187d17ca6ed16",
  "city": "ALPINE",
  "zip": "35014",
  "loc": {
    "y": 33.331165,
    "x": 86.208934
  },
  "pop": 3062,
  "state": "AL"
}
```


MongoDB.

```
{
  "_id": "5c8eccc1caa187d17ca6ed16",
  "city": "ALPINE",
  "zip": "35014",
  "loc": {
    "y": 33.331165,
    "x": 86.208934
  },
  "pop": 3062,
  "state": "AL"
}
```


MongoDB.

```
{
  "_id": "5c8eccc1caa187d17ca6ed16",
  "city": "ALPINE",
  "zip": "35014",
  "loc": {
    "y": 33.331165,
    "x": 86.208934
  },
  "pop": 3062,
  "state": "AL"
}
```


MongoDB.

¿Qué son las colecciones?

MongoDB.

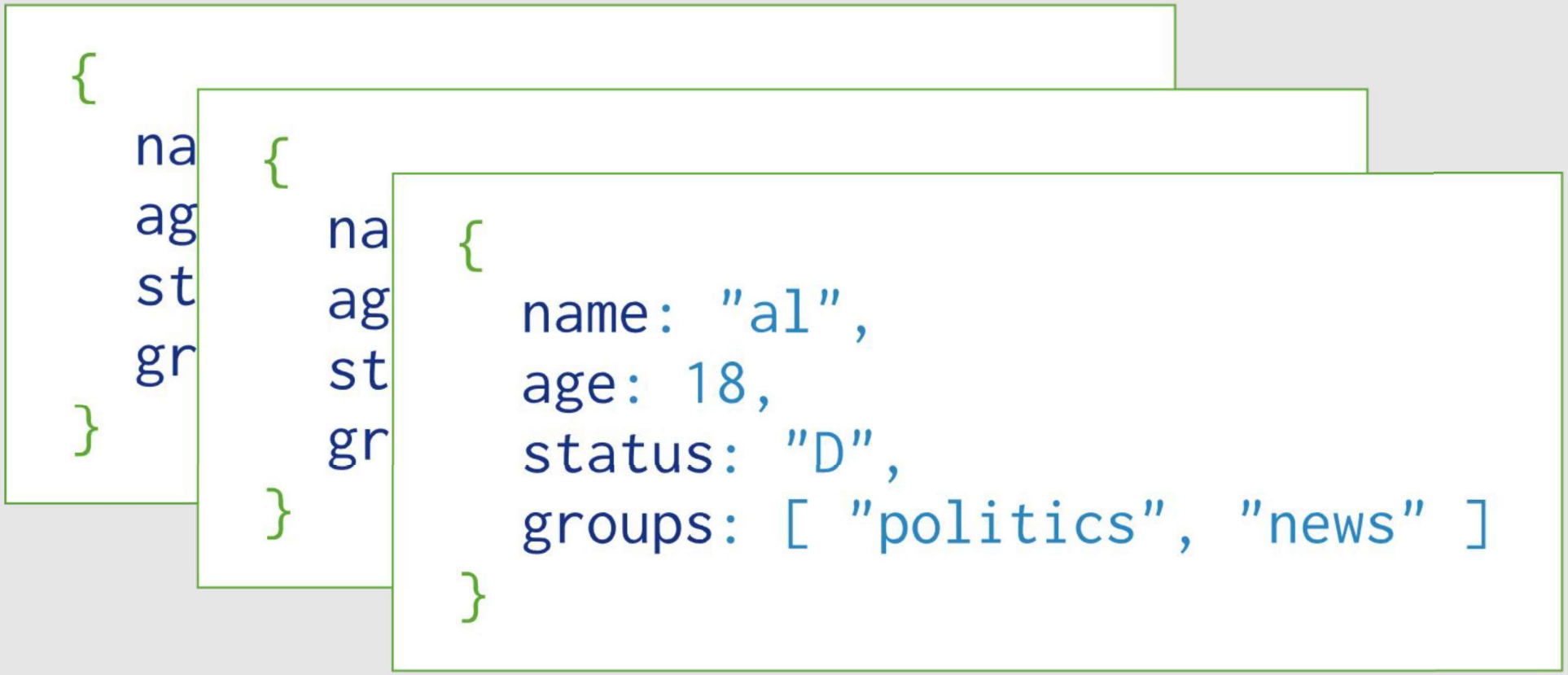
Collection

MongoDB almacena documentos en una colección, usualmente con campos comunes entre sí.

MongoDB.

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

MongoDB.



Collection

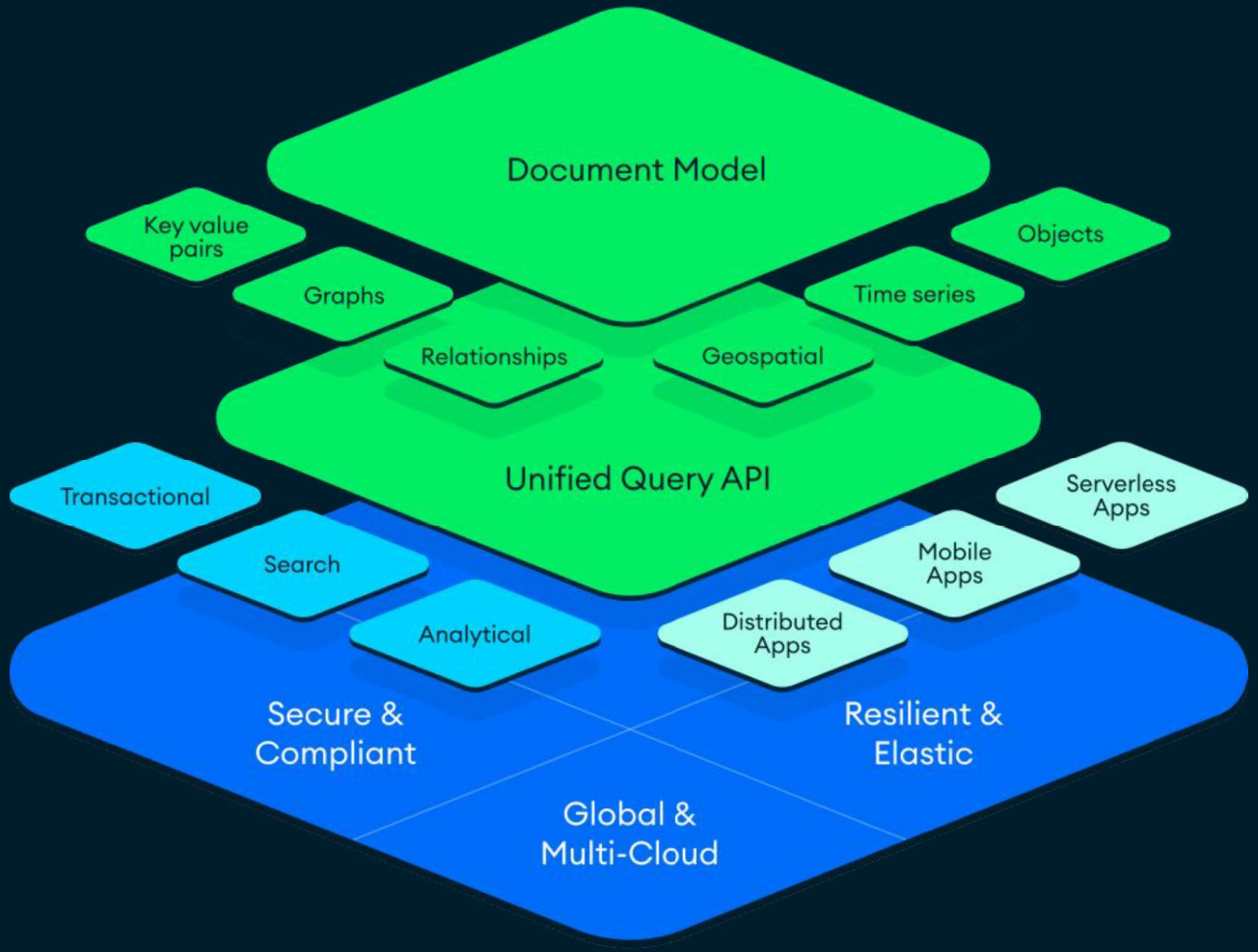
MongoDB.

Crea tu primer DB NoSQL

MongoDB.

¿Qué es Mongo Atlas?

MongoDB.


Document Model
Key value pairs
Graphs
Relationships
Geospatial
Time series
Objects
Unified Query API
Transactional
Search
Analytical
Distributed Apps
Mobile Apps
Serverless Apps
Secure & Compliant
Resilient & Elastic
Global & Multi-Cloud

MongoDB.

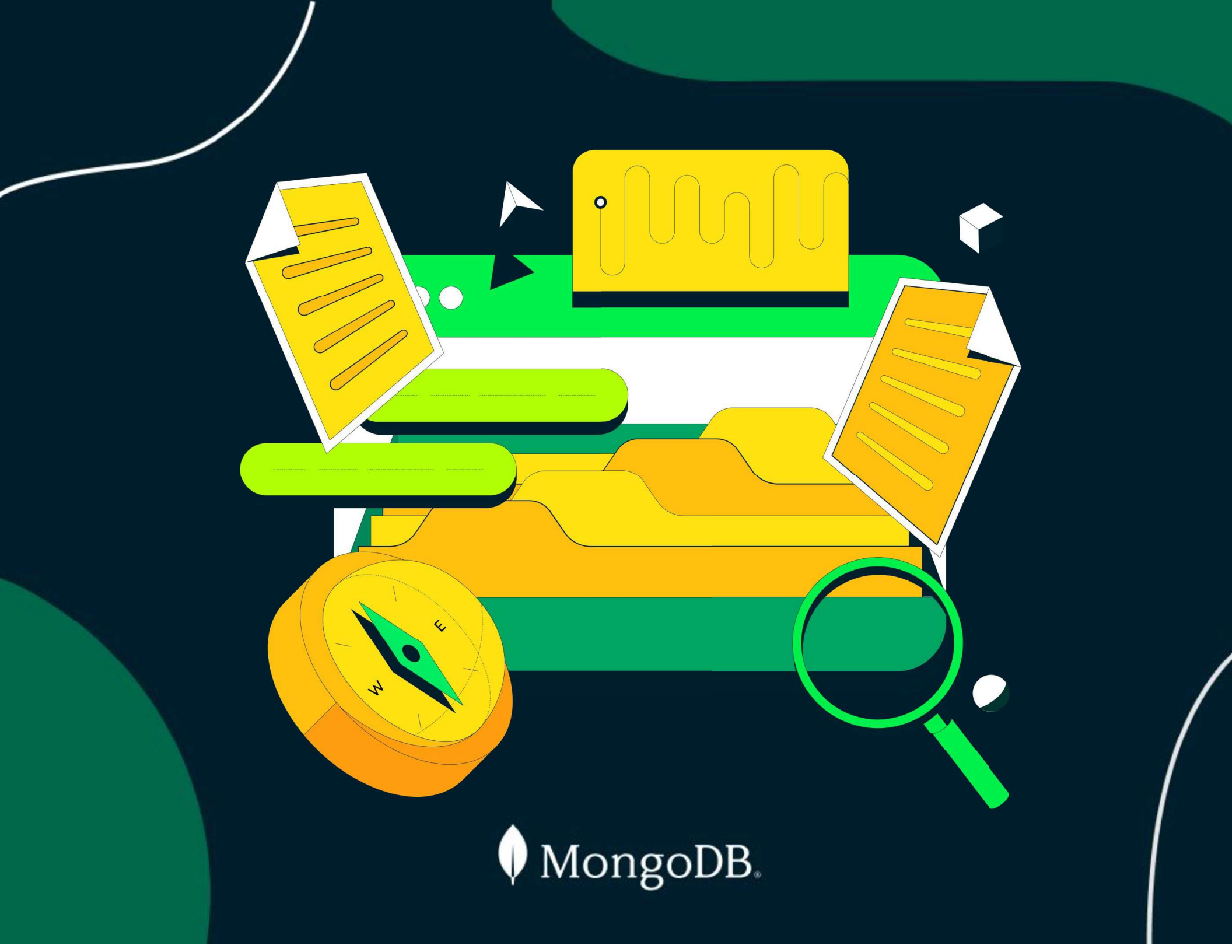
Free Cluster

- 512MB to 5GB of storage.
- Shared RAM.
- Upgrade to dedicated clusters for full functionality.
- No credit card required to start.

MongoDB.

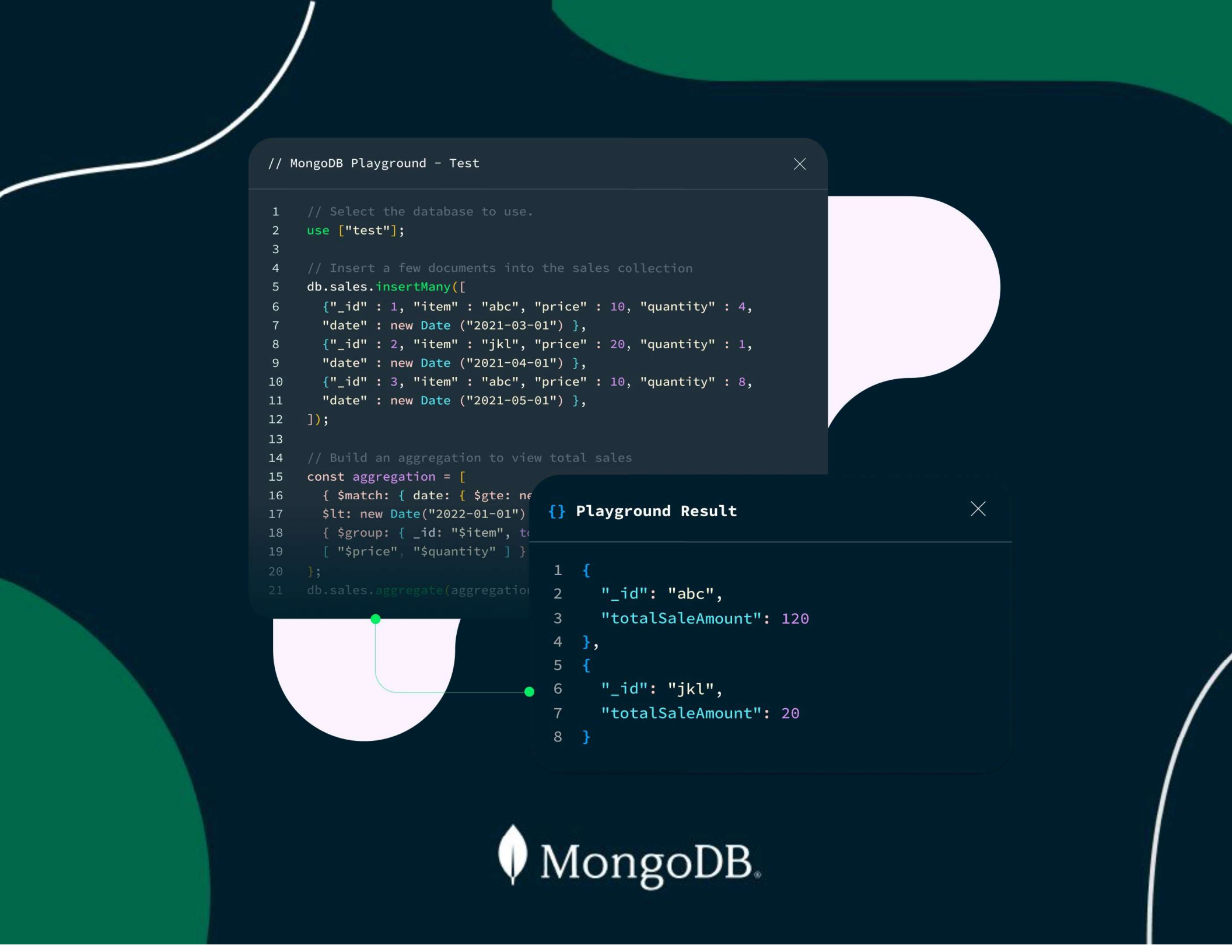
Usando Mongo Compass

MongoDB.


E
W
MongoDB.

MongoDB en VSCode

MongoDB


// MongoDB Playground - Test
// Select the database to use.
use ["test"];
// Insert a few documents into the sales collection
db.sales.insertMany([
{"_id" : 1, "item" : "abc", "price" : 10, "quantity" : 4,
"date" : new Date ("2021-03-01") },
{"_id" : 2, "item" : "jkl", "price" : 20, "quantity" : 1,
"date" : new Date ("2021-04-01") },
{"_id" : 3, "item" : "abc", "price" : 10, "quantity" : 8,
"date" : new Date ("2021-05-01") },
]);
// Build an aggregation to view total sales
const aggregation = [
{ $match: { date: { $gte: ne
$lt: new Date("2022-01-01")
{ $group: { _id: "$item", t
["$price", "$quantity"] }
};
db.sales.aggregate(aggregatio
{} Playground Result
{
"_id": "abc",
"totalSaleAmount": 120
},
{
"_id": "jkl",
"totalSaleAmount": 20
}
MongoDB.

MongoDB corriendo en Docker

Conectándonos usando mongosh

MongoDB.

JSON vs. BSON

Ventajas de **JSON**

- Amigable.
- Se puede leer.
- Es un formato muy usado.

MongoDB.

Desventajas de **JSON**

- Basado en texto.
- Consume mucho espacio.
- Es limitado: string, boolean, number, arrays.

MongoDB.

```
{
  "_id": "5c8eccc1caa187d17ca6ed16",
  "city": "ALPINE",
  "zip": "35014",
  "loc": {
    "y": 33.331165,
    "x": 86.208934
  },
  "pop": 3062,
  "state": "AL"
}
```


MongoDB.

Ventajas de **BSON**

- Representación binaria de **JSON**.
- No consume espacio.
- Alto rendimiento.
- Tipos de datos: +, date, raw binary, integer, long, float.

MongoDB.

Desventajas de **BSON**

- No es estándar.
- Es un formato para la máquina.

MongoDB.

```
{
    "hello" : "world"
}
```

```
\x16\x00\x00\x00                  // total document size
\x02                              // 0x02 = type String
hello\x00                         // field name
\x06\x00\x00\x00world\x00         // field value
\x00                              // 0x00 = end of object
```

Insertando un documento

MongoDB.

Insertando varios documentos

MongoDB.

Actualizando un documento

MongoDB.

Operators

- **$set**
  Sets the value of a field in a document.
- **$inc**
  Increments the value of the field by the specified amount.
- **ObjectId**
  Function to find a doc with objID

MongoDB.

Actualizando varios documentos

MongoDB.

Operators

- **$set**
  Sets or create the value of a field in a document.
- **$rename**
  Renames a field.
- **$unset**
  Removes the specified field from a document.

MongoDB.

Array Update Operators

MongoDB.

Operators

- **$push**
  Adds an item to an array.
- **$pull**
  Removes all array elements that match a specified query.
- **$in**
  Matches any of the values specified in an array.

MongoDB.

Insert or Update?

Operators

- **$pop**
  Removes the first or last item of an array.

Eliminando documentos

MongoDB.

Comparison Query Operators

MongoDB.

Usando $eq & $ne

MongoDB.

Operators

- **$eq**
  Matches values that are equal to a specified value.
- **$ne**
  Matches all values that are not equal to a specified value.

MongoDB.

Usando $gt, $gte, $lt y $lte

MongoDB.

Operators

- **$gte**
  Matches values that are greater than or equal to a specified value.
- **$gt**
  Matches values that are greater than a specified value.

MongoDB.

Operators

- **$lte**
  Matches values that are less than or equal to a specified value.
- **$lt**
  Matches values that are less than a specified value.

MongoDB.

Otros ejemplos con $gt, $gte, $lt y $lte

MongoDB.

Usando $regex

MongoDB.

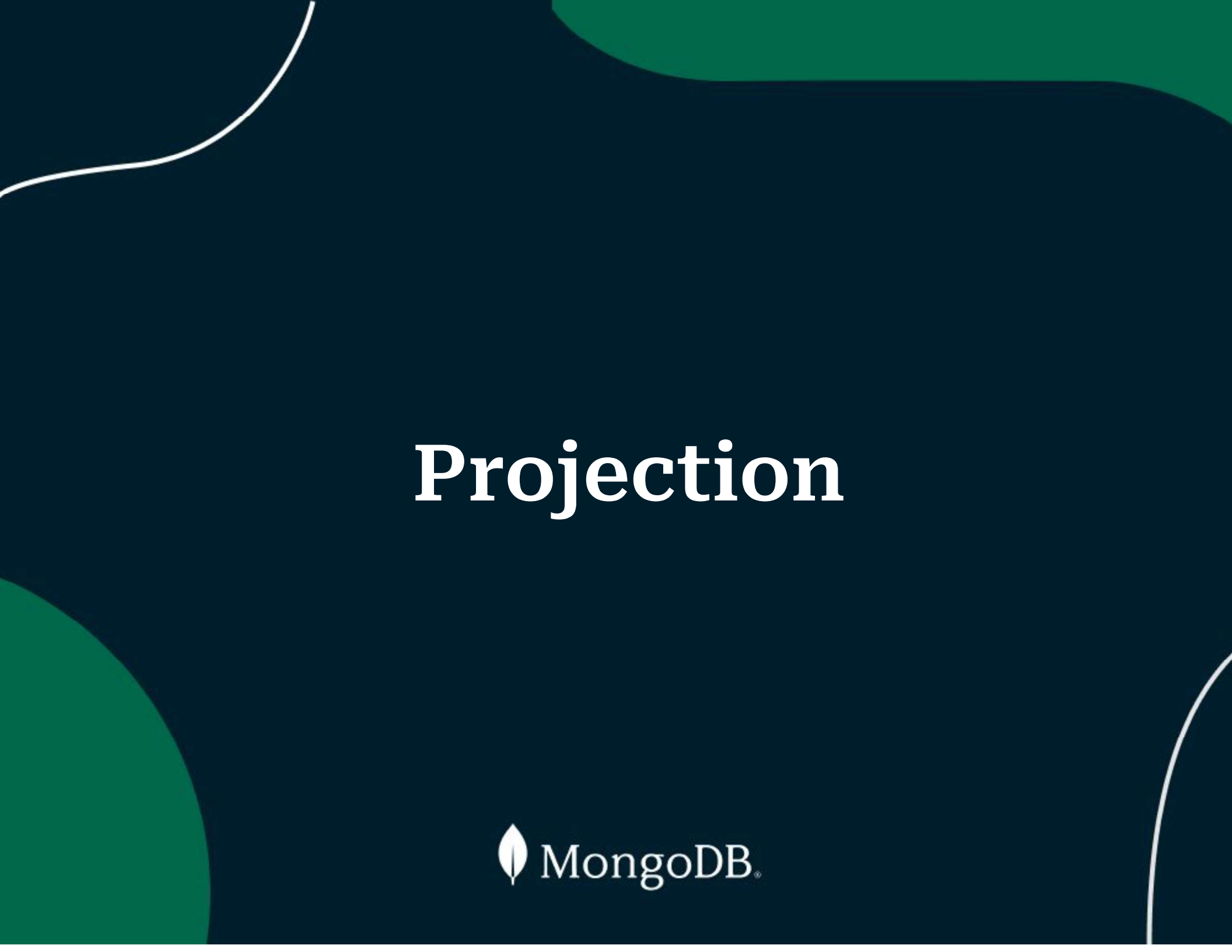
Operators

- **$regex**
  Allows use of aggregation expressions within the query language.

MongoDB Atlas Search

MongoDB.

Projection



Operadores para Arrays

MongoDB.

Operators

- **$in**
  Matches any of the values specified in an array.
- **$nin**
  Matches none of the values specified in an array.

MongoDB.

Operators

- **$all**
  Matches arrays that contain all elements specified in the query.
- **$elemMatch**
  Selects documents if element in the array field matches all the specified $elemMatch conditions.

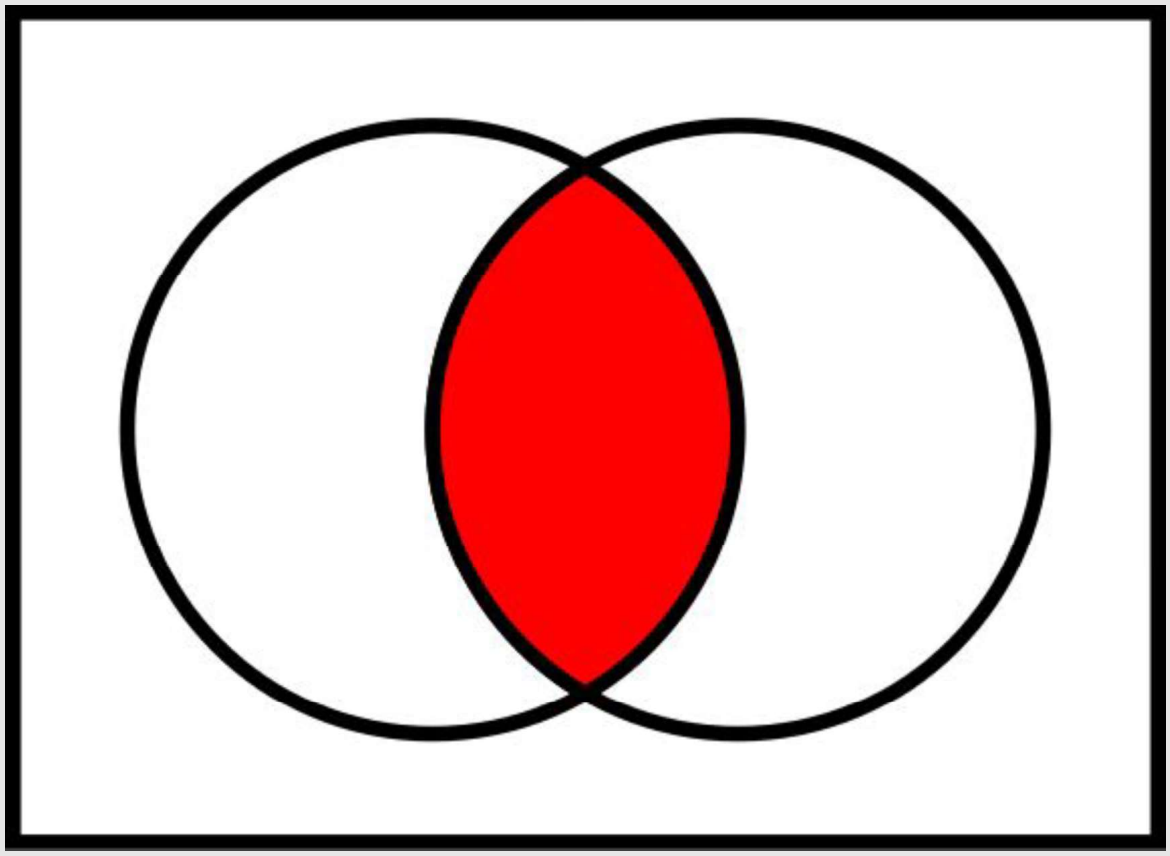
Operators

- **$size**
  Selects documents if the array field is a specified size.

Operadores lógicos

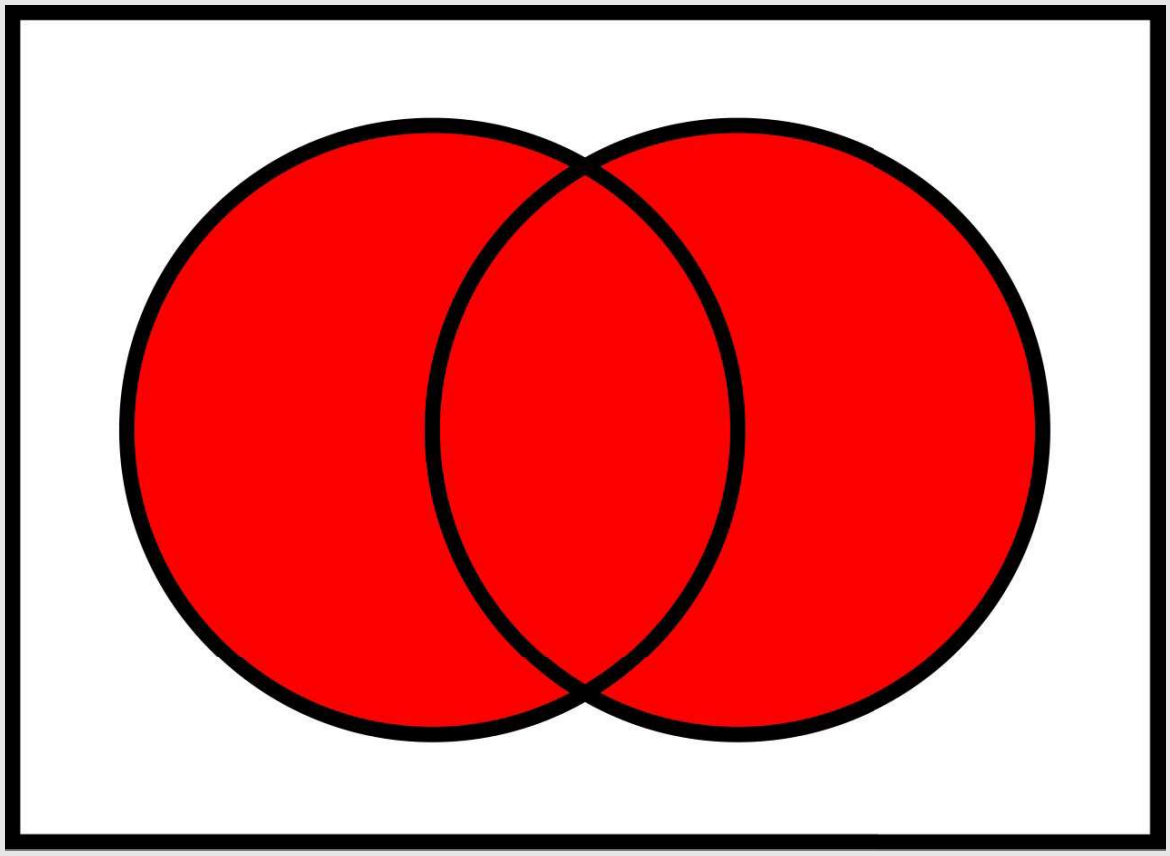
MongoDB.

AND



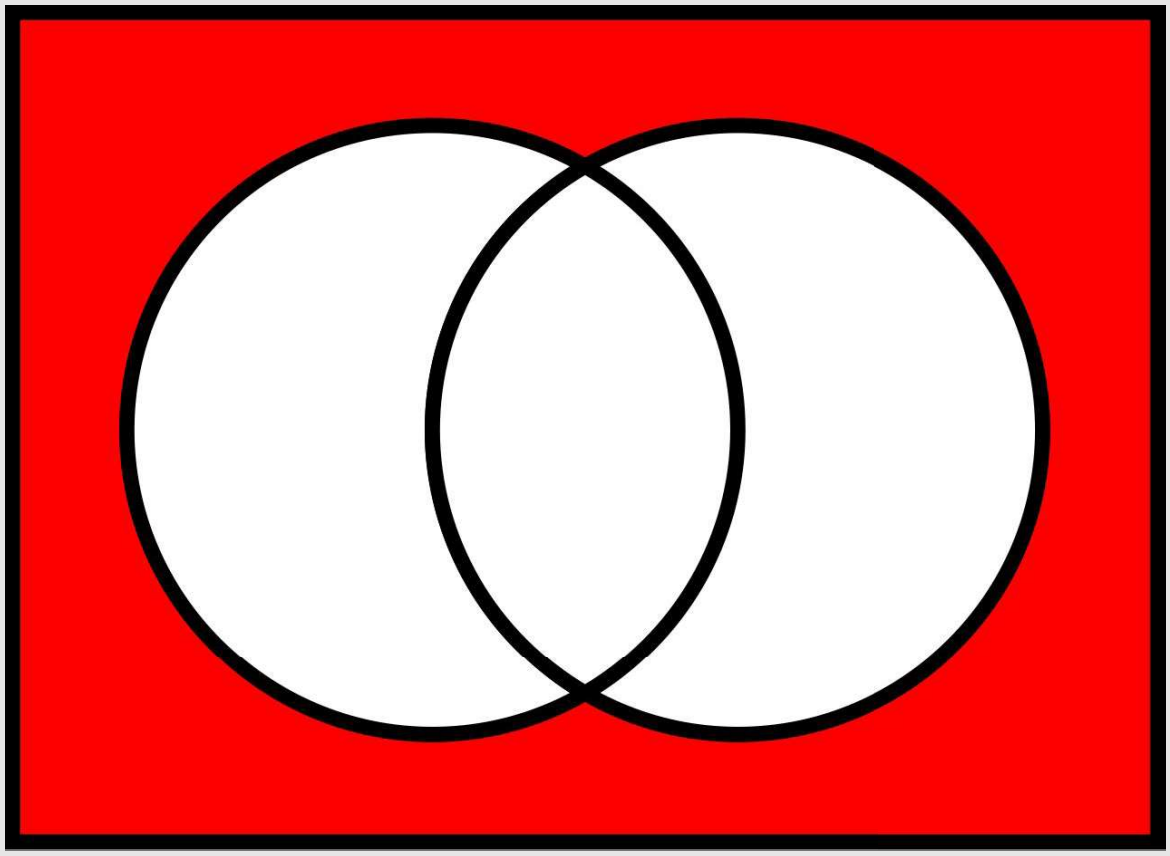
MongoDB.

OR



MongoDB.

NOR



MongoDB.

Operators

- **$and**
  Joins query clauses with a logical AND returns all documents that match the conditions of both clauses.
- **$not**
  Inverts the effect of a query expression and returns documents that do not match the query expression.

MongoDB.

Operators

- **$nor**
  Joins query clauses with a logical NOR returns all documents that fail to match both clauses.
- **$or**
  Joins query clauses with a logical OR returns all documents that match the conditions of either clause.

MongoDB.

Expressive operator

MongoDB.

Operators

- **$expr**
  Allows use of aggregation expressions within the query language.

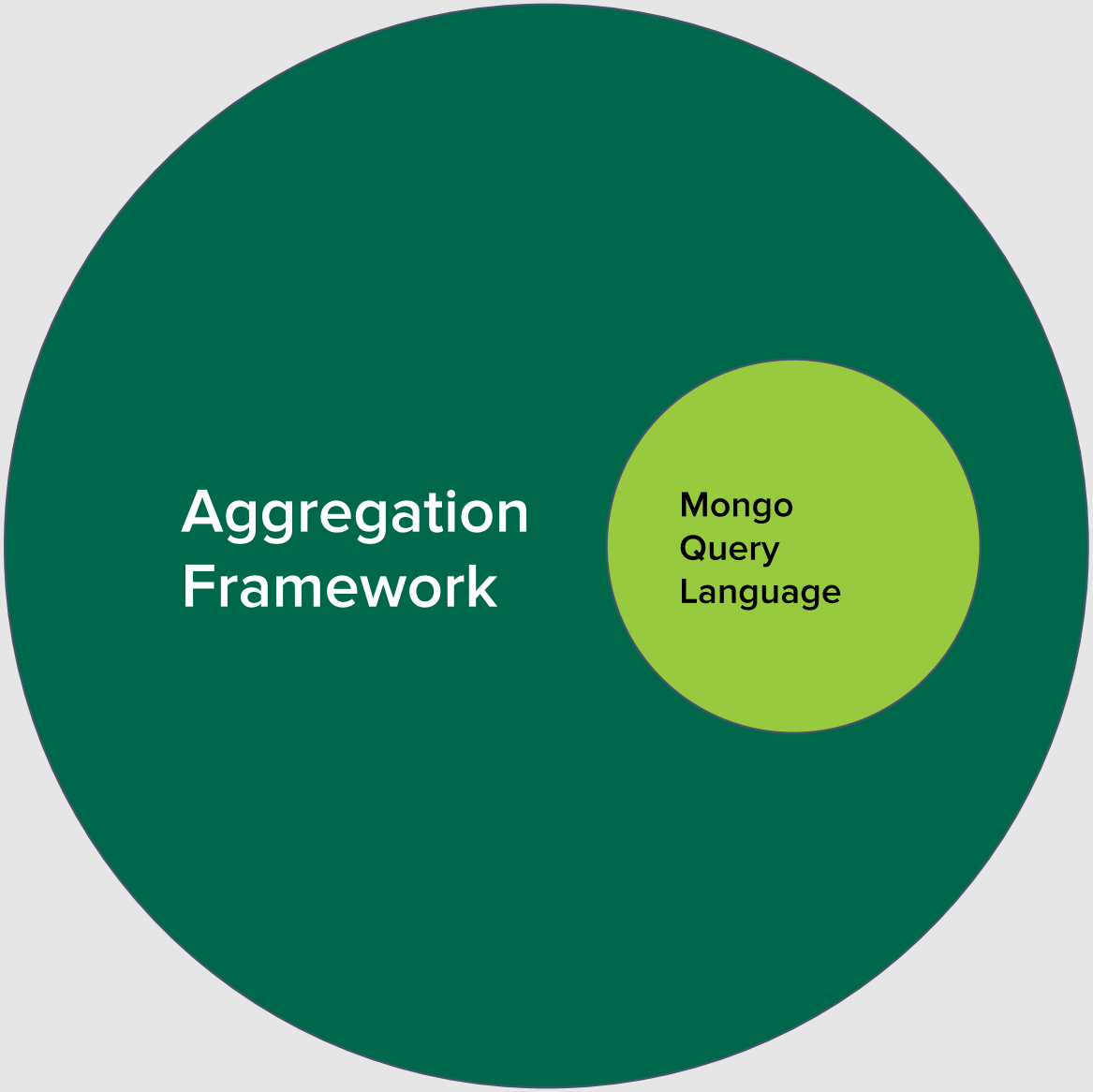
Consultas a Arrays y SubDoc

MongoDB.

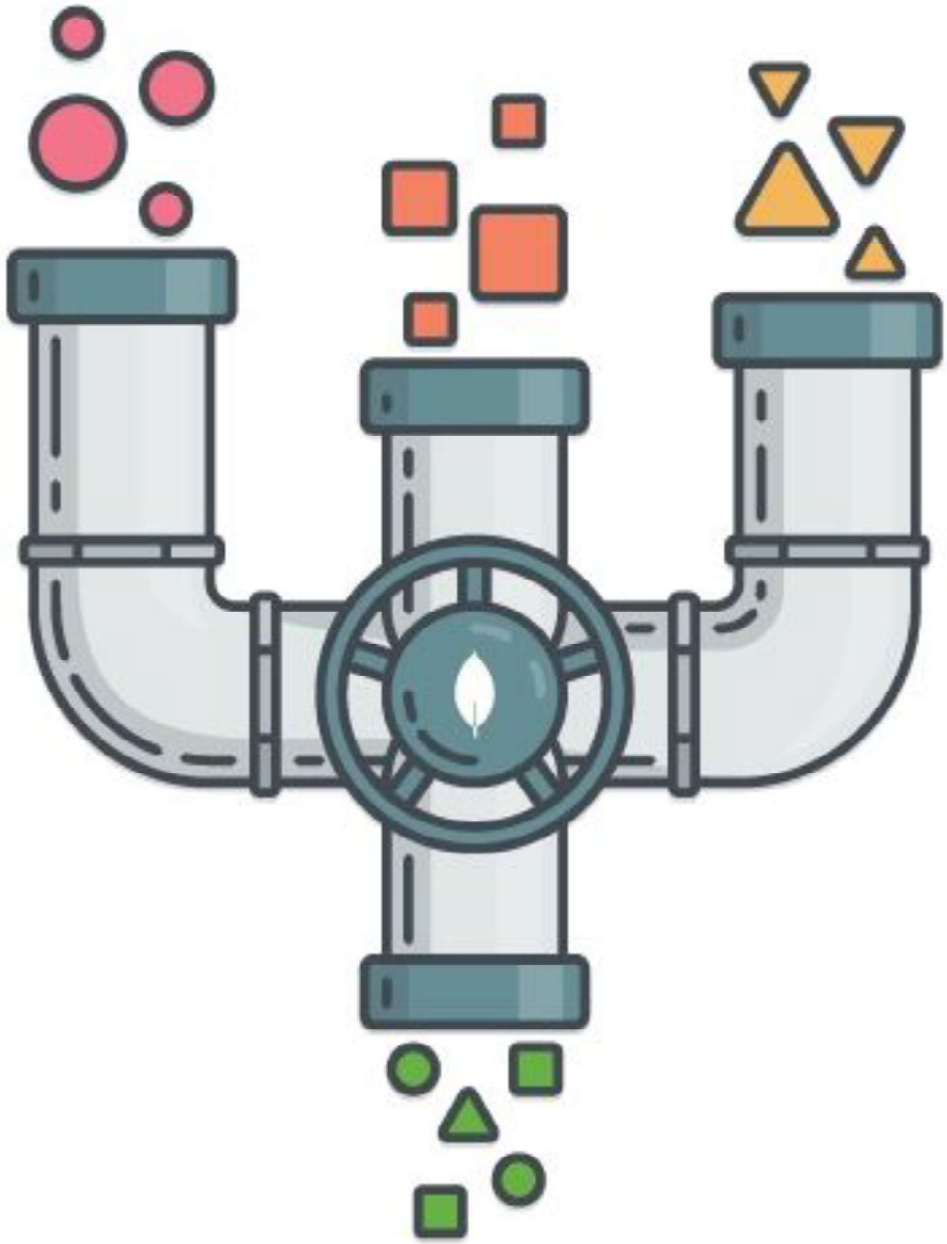
Array Update Operators

MongoDB.

Aggregation Framework



MongoDB.

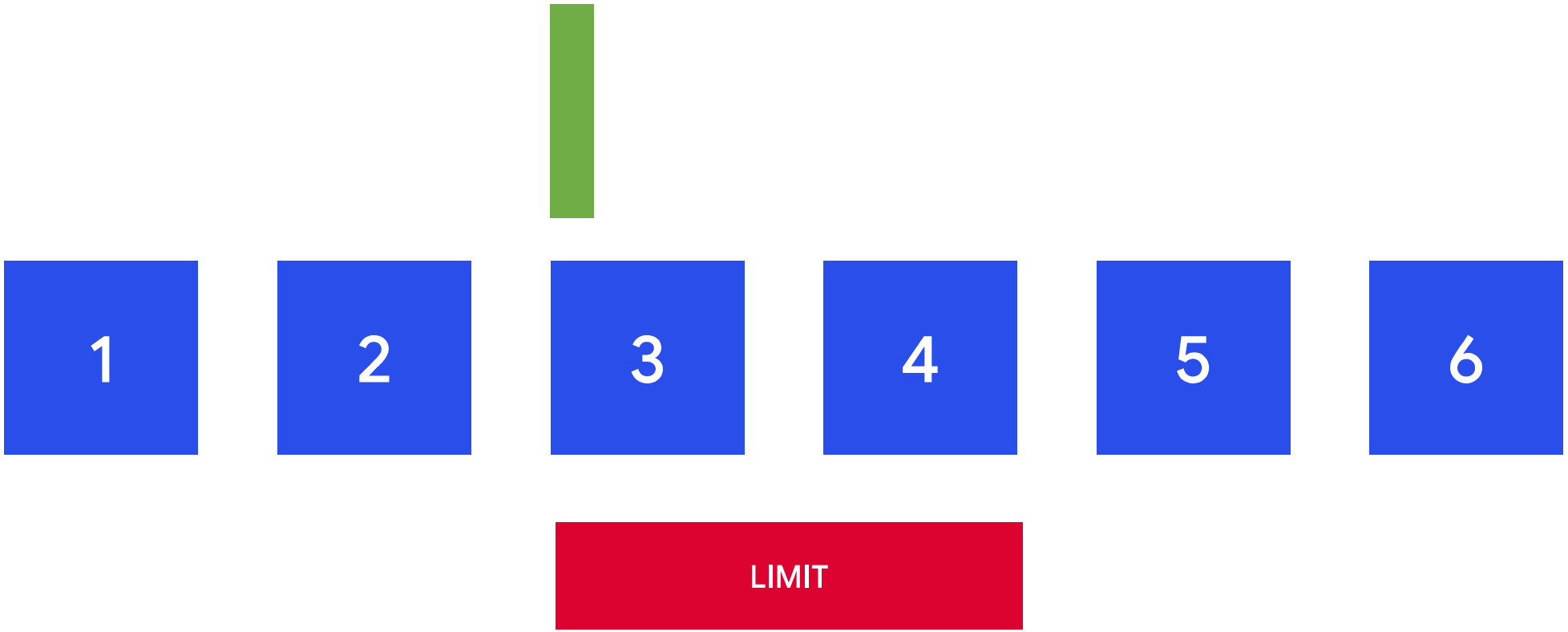



GAZ
911
énergir

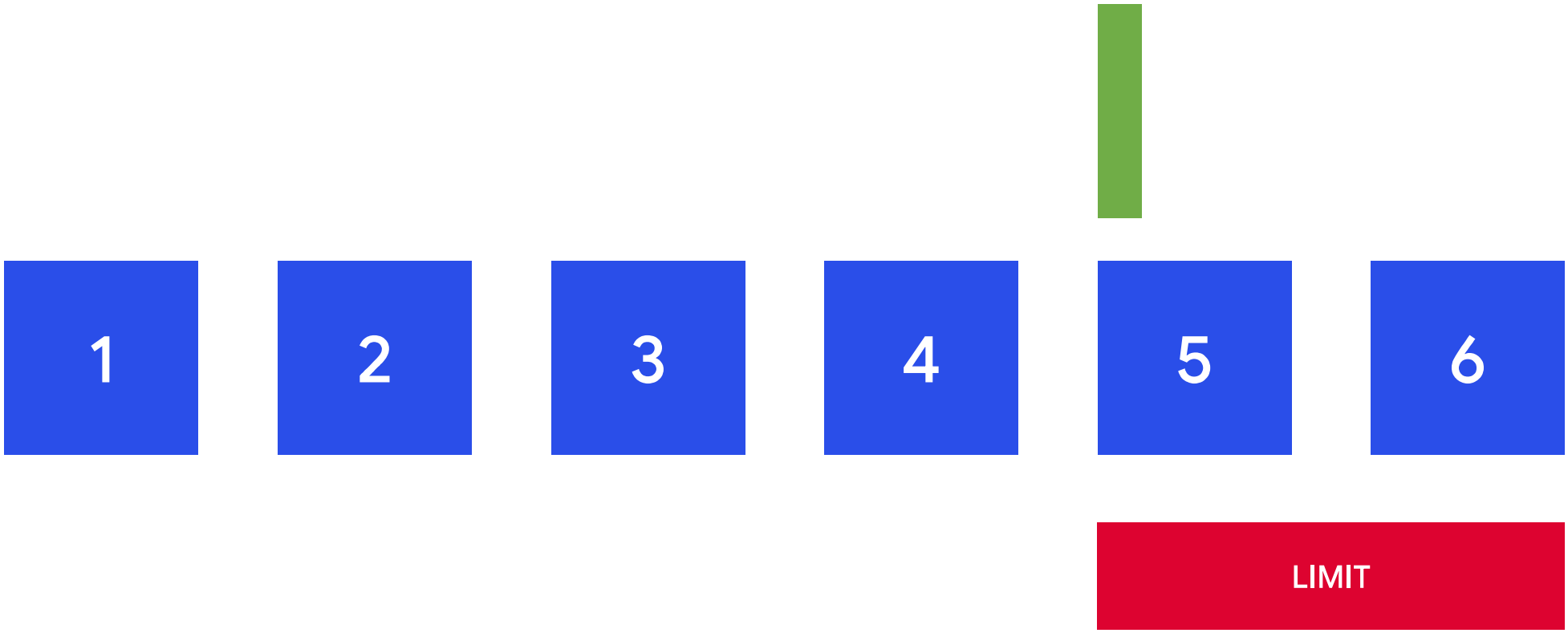
Sort & Limit

Pagination

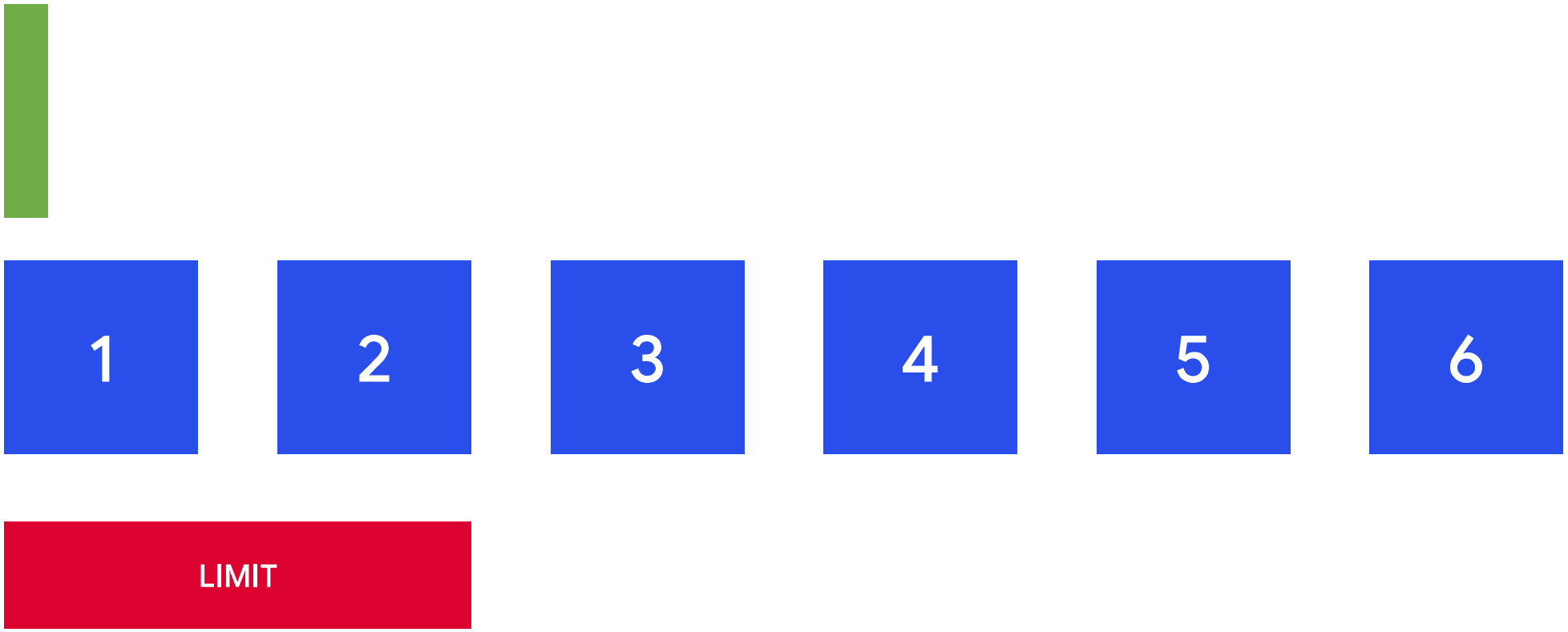
limit = 2
skip / offset = 2



limit = 2
skip / offset = 4



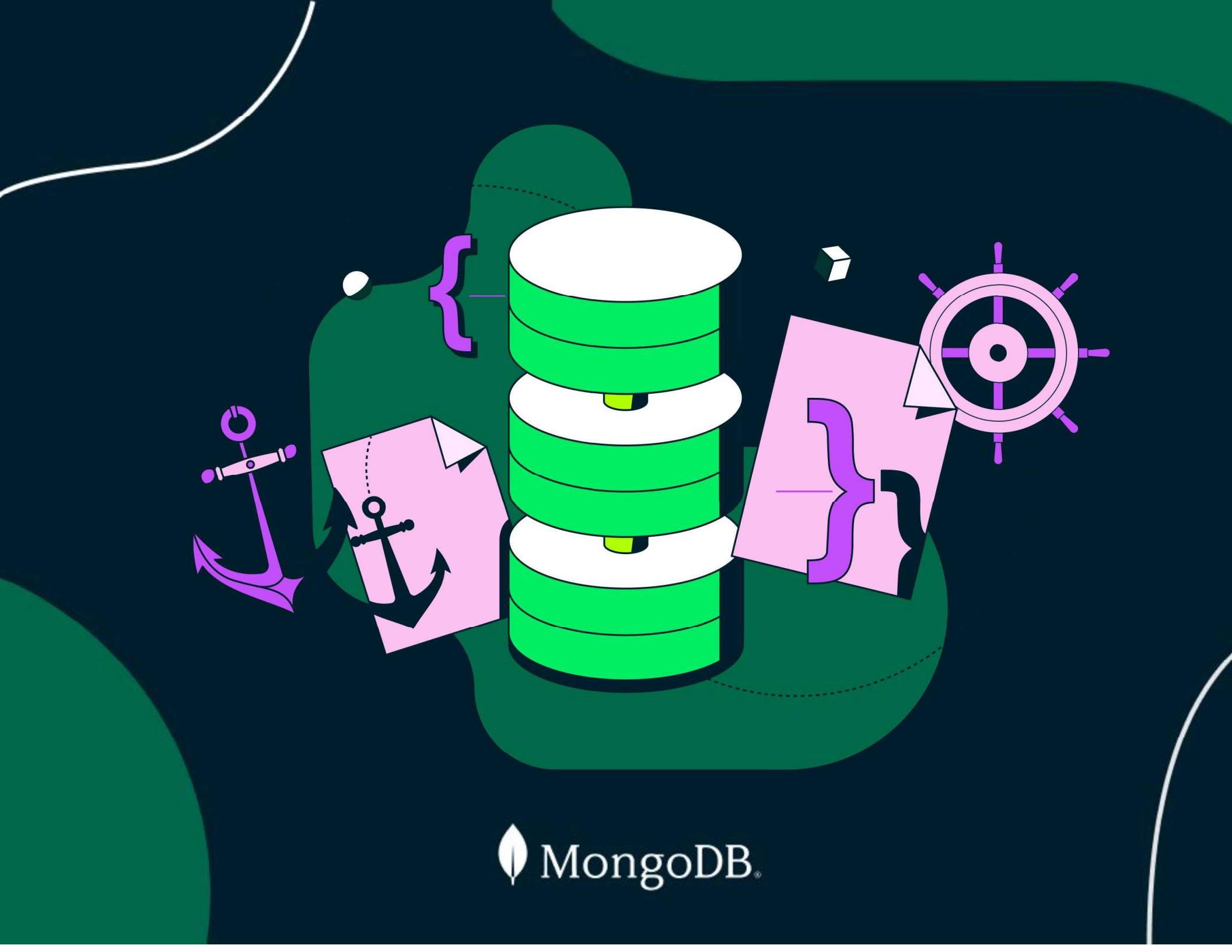
limit = 2
skip / offset = 0



Atlas Features

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);
```

Curso de Modelado de Datos con MongoDB

MongoDB.