
mongoDB


Apache
CASSANDRA™


APACHE
HBASE


amazon
DynamoDB


neo4j


RAVENDB


CouchDB
relax


redis


HYPERTABLE INC


OrientDB®

ACCIÓN FORMATIVA

DURACIÓN: 20 horas

MODALIDAD: Online

FECHAS y HORARIO: 8 de Julio a 12 de Julio de 9:30 a 13:30

CONTENIDOS

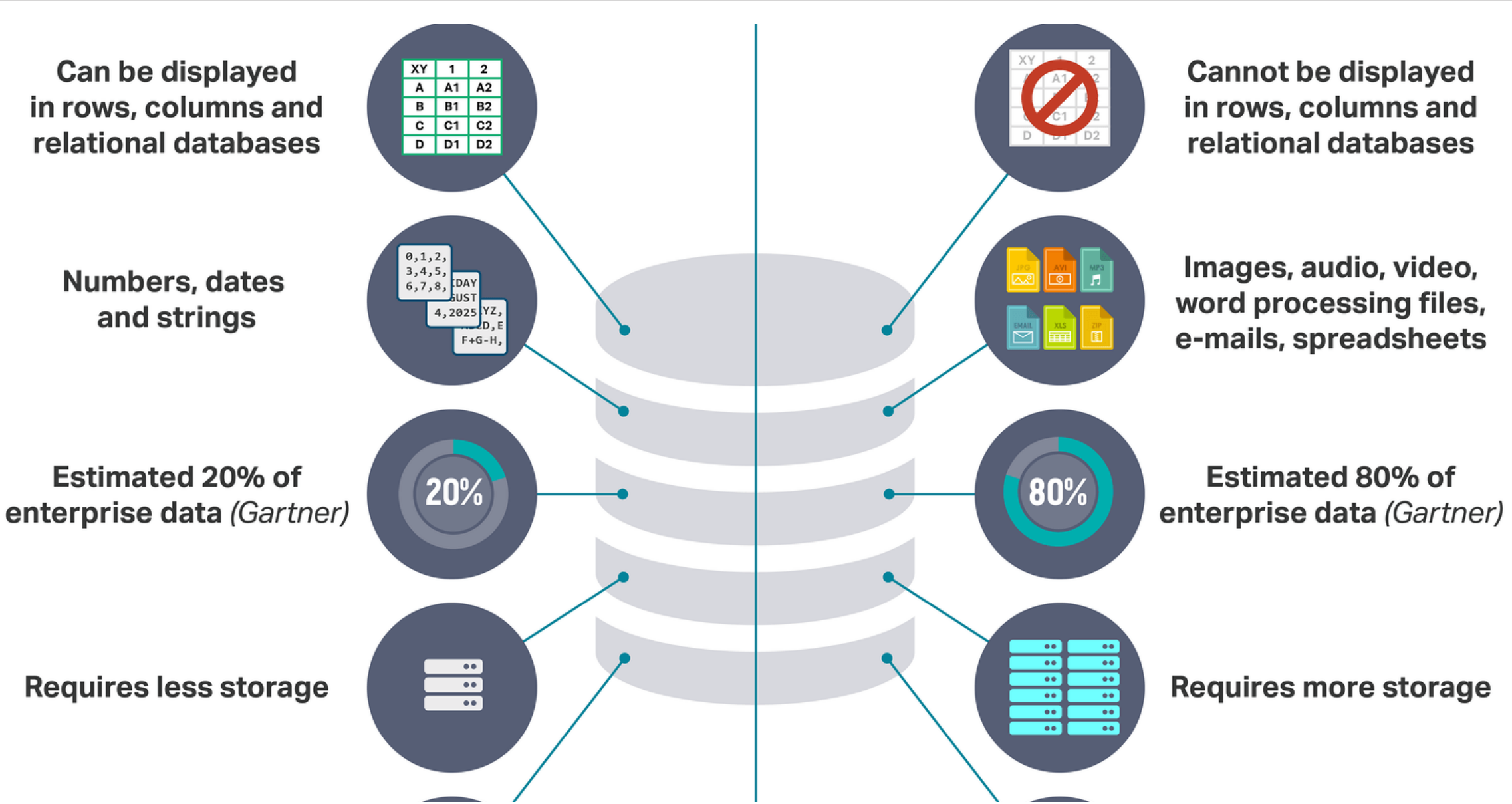
- Introducción.
- Bases de datos Relacionales.
- NoSQL.
- Redis.
- MongoDB.
- Apache Cassandra.
- Neo4J

PRESENTACIÓN

- NOMBRE Y UBICACIÓN
- EXPERIENCIA PREVIA CON BASES DE DATOS
- EXPERIENCIA PREVIA CON MONGODB
- EXPECTATIVAS DEL CURSO

STRUCTURED DATA

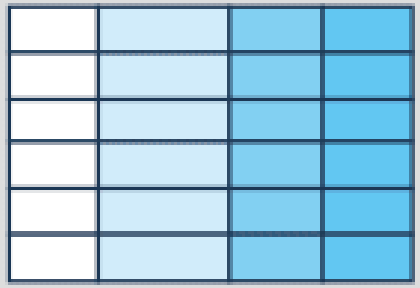
UNSTRUCTURED DATA


Can be displayed in rows, columns and relational databases
XY 1 2
A A1 A2
B B1 B2
C C1 C2
D D1 D2
Numbers, dates and strings
0,1,2,
3,4,5,
6,7,8,
4,2025
F+G-H,
Estimated 20% of enterprise data (Gartner)
20%
Requires less storage
Cannot be displayed in rows, columns and relational databases
Images, audio, video, word processing files, e-mails, spreadsheets
JPG
AVI
MP3
EMAIL
XLS
ZIP
Estimated 80% of enterprise data (Gartner)
80%
Requires more storage

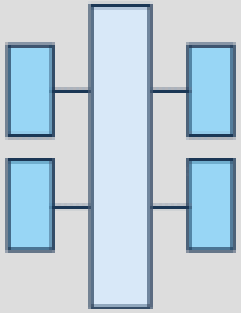

Cost reduction
BIG DATA ANALYTICS
Faster, better decision making
New products & services

SQL

Relational

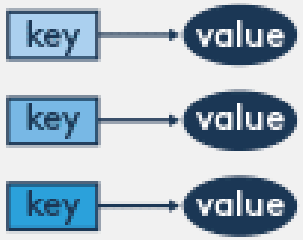


Analytical (OLAP)

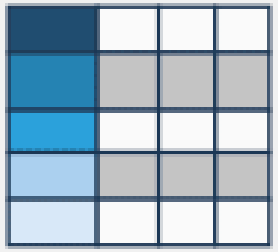


NoSQL

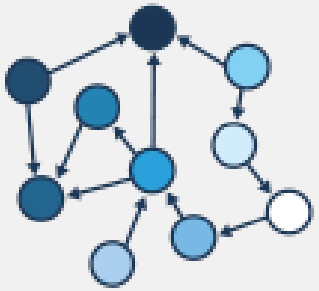
Key-Value



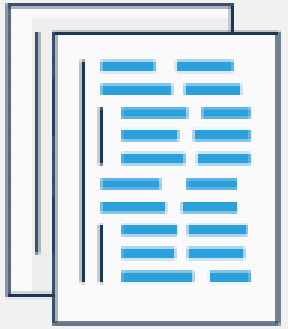
Column-Family

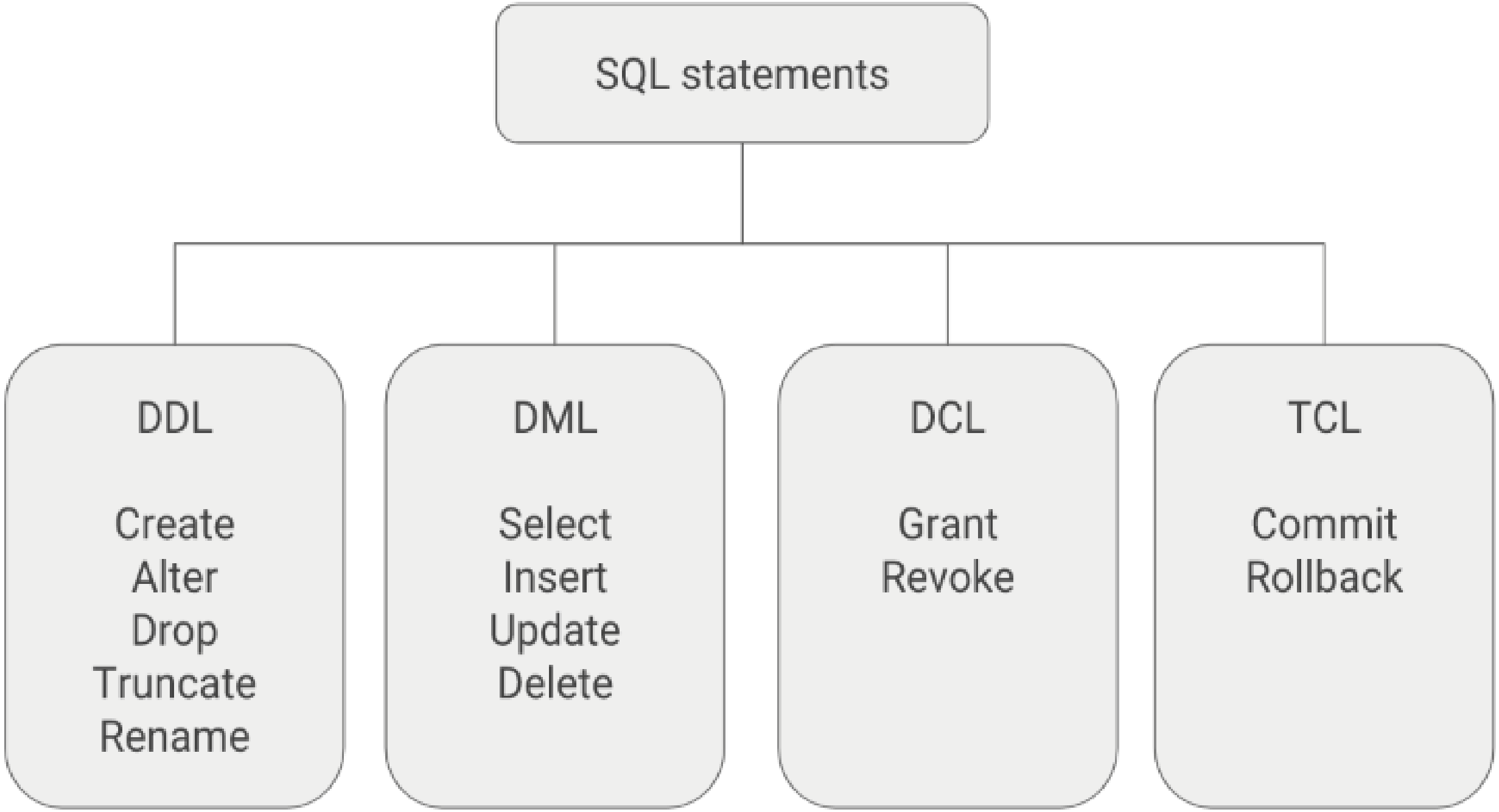


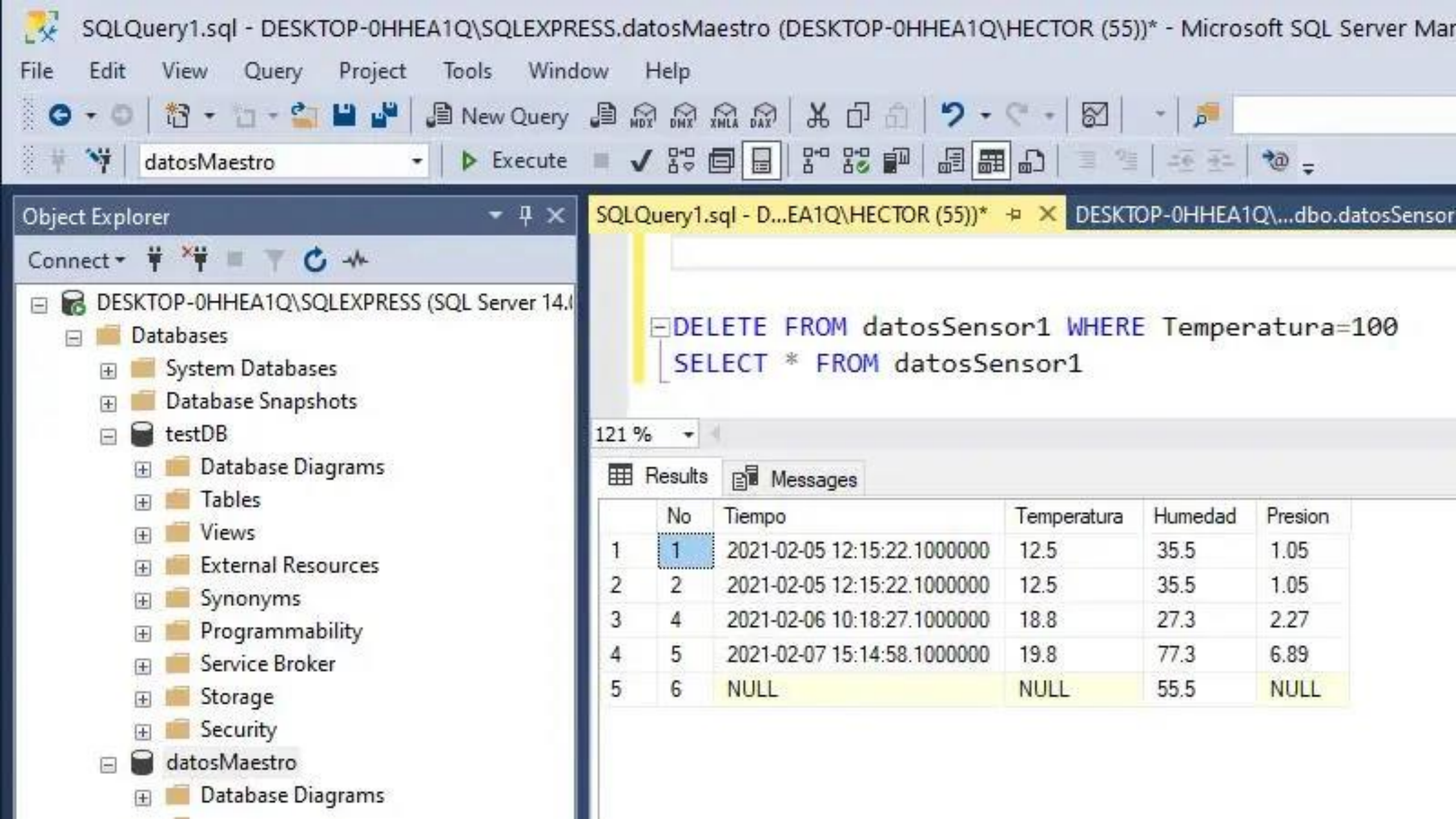
Graph



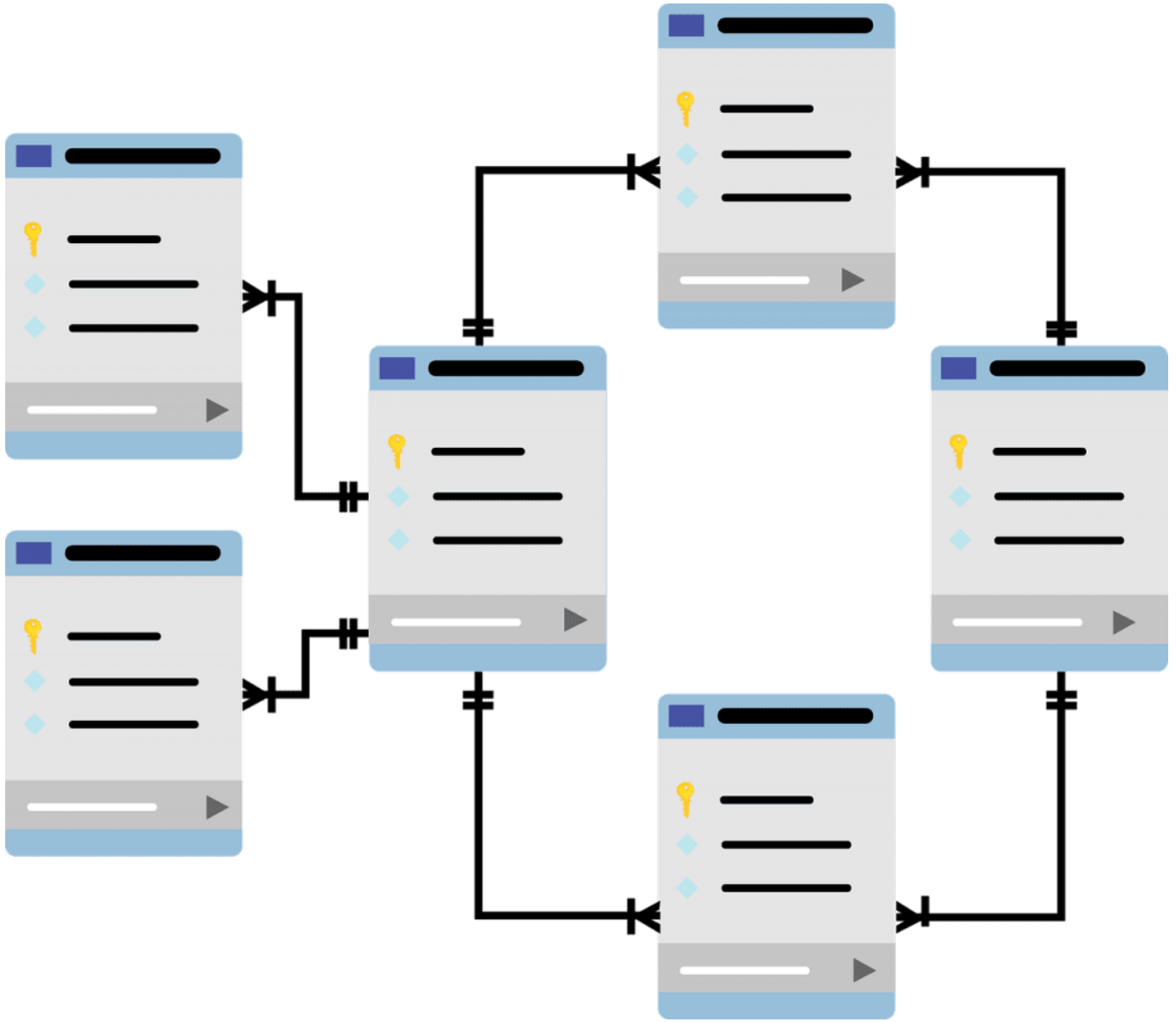
Document




SQL statements
DDL
Create
Alter
Drop
Truncate
Rename
DML
Select
Insert
Update
Delete
DCL
Grant
Revoke
TCL
Commit
Rollback


SQLQuery1.sql - DESKTOP-0HHEA1Q\SQLEXPRESS.datosMaestro (DESKTOP-0HHEA1Q\HECTOR (55))* - Microsoft SQL Server Mar
File Edit View Query Project Tools Window Help
New Query
datosMaestro
Execute
Object Explorer
Connect
DESKTOP-0HHEA1Q\SQLEXPRESS (SQL Server 14.
Databases
System Databases
Database Snapshots
testDB
Database Diagrams
Tables
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
datosMaestro
Database Diagrams
SQLQuery1.sql - D...EA1Q\HECTOR (55))*
DESKTOP-0HHEA1Q\...dbo.datosSensor
DELETE FROM datosSensor1 WHERE Temperatura=100
SELECT * FROM datosSensor1
121 %
Results
Messages
No Tiempo Temperatura Humedad Presion
1 1 2021-02-05 12:15:22.1000000 12.5 35.5 1.05
2 2 2021-02-05 12:15:22.1000000 12.5 35.5 1.05
3 4 2021-02-06 10:18:27.1000000 18.8 27.3 2.27
4 5 2021-02-07 15:14:58.1000000 19.8 77.3 6.89
5 6 NULL NULL 55.5 NULL

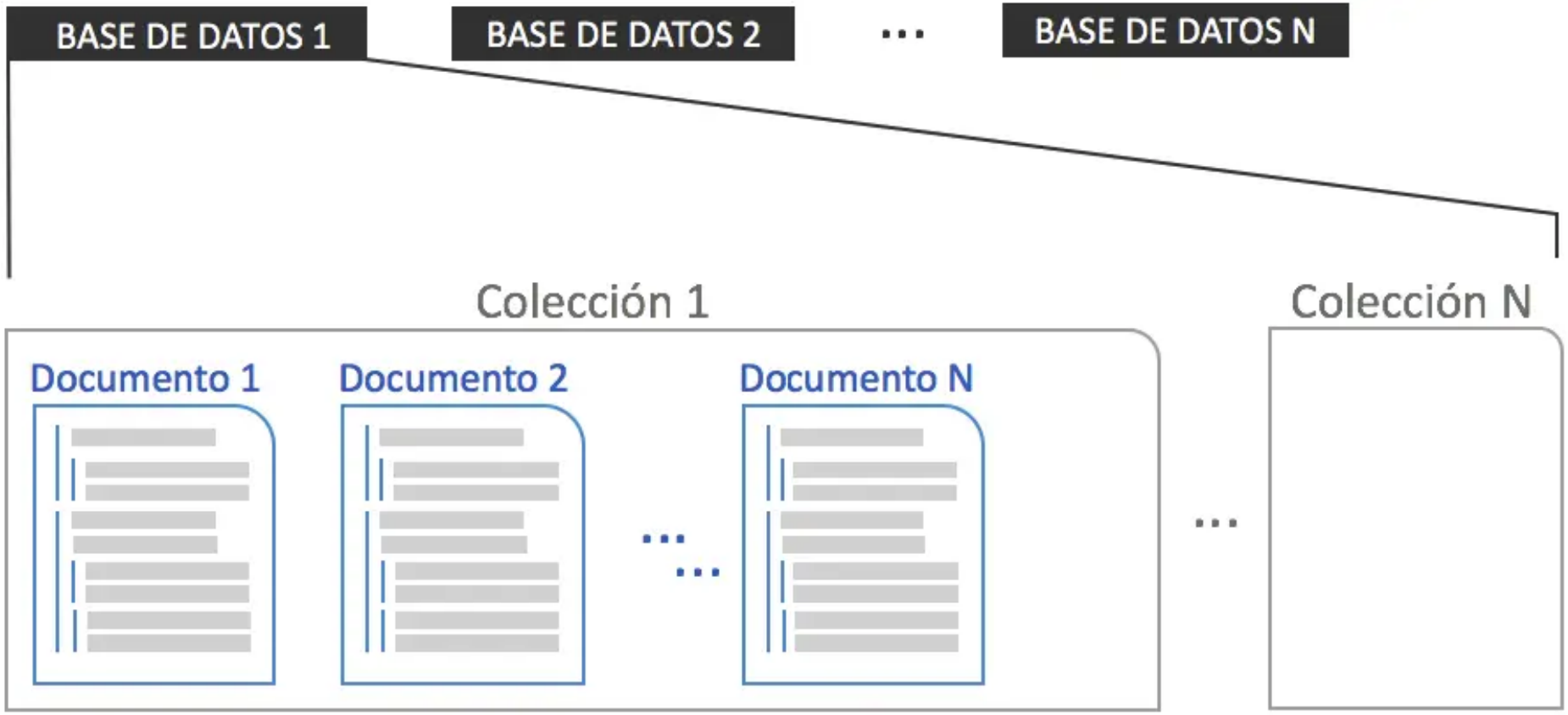
MODELO RELACIONAL

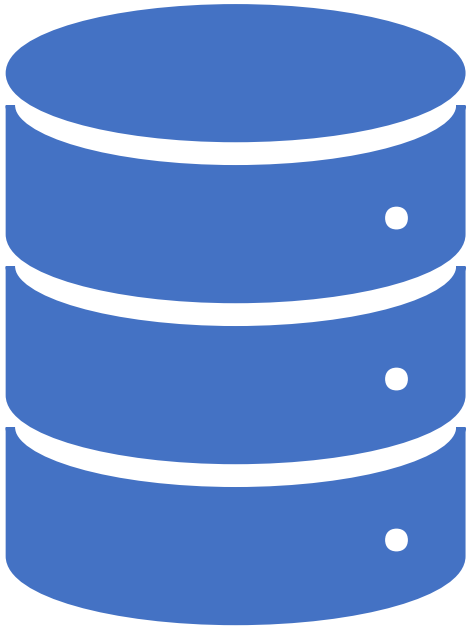



Microsoft Learn

MOTORES NOSQL

BASE DE DATOS DOCUMENTAL





MONGODB

- El gestor de base de datos MongoDB se lo puede asociar a un conjunto de gestores de bases de datos que no tienen como lenguaje principal el SQL para su manipulación.
- Los gestores de bases de datos NoSQL no requieren estructuras fijas como tablas, normalmente no soportan operaciones join y presentan como gran ventaja que pueden escalar en forma sencilla.

CARACTERÍSTICAS MONGODB

- Indexación
- Replicación
- Balanceo de carga
- Almacenamiento de archivos
- Agregación

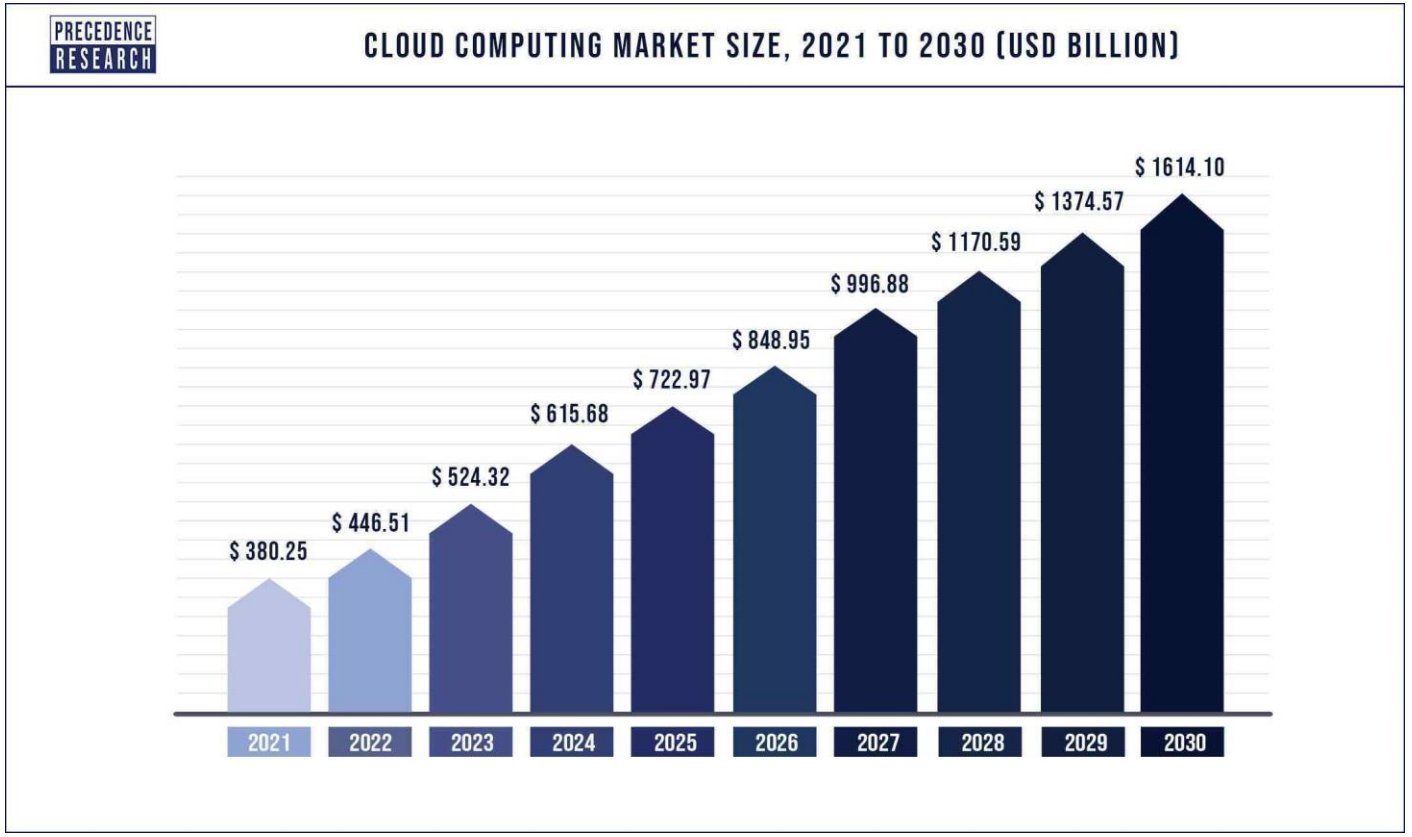
JSON

```
{
  codigo: 1,
  nombre: 'El aleph',
  autor: 'Borges',
  editoriales: ['Planeta','Siglo XXI']
}
```

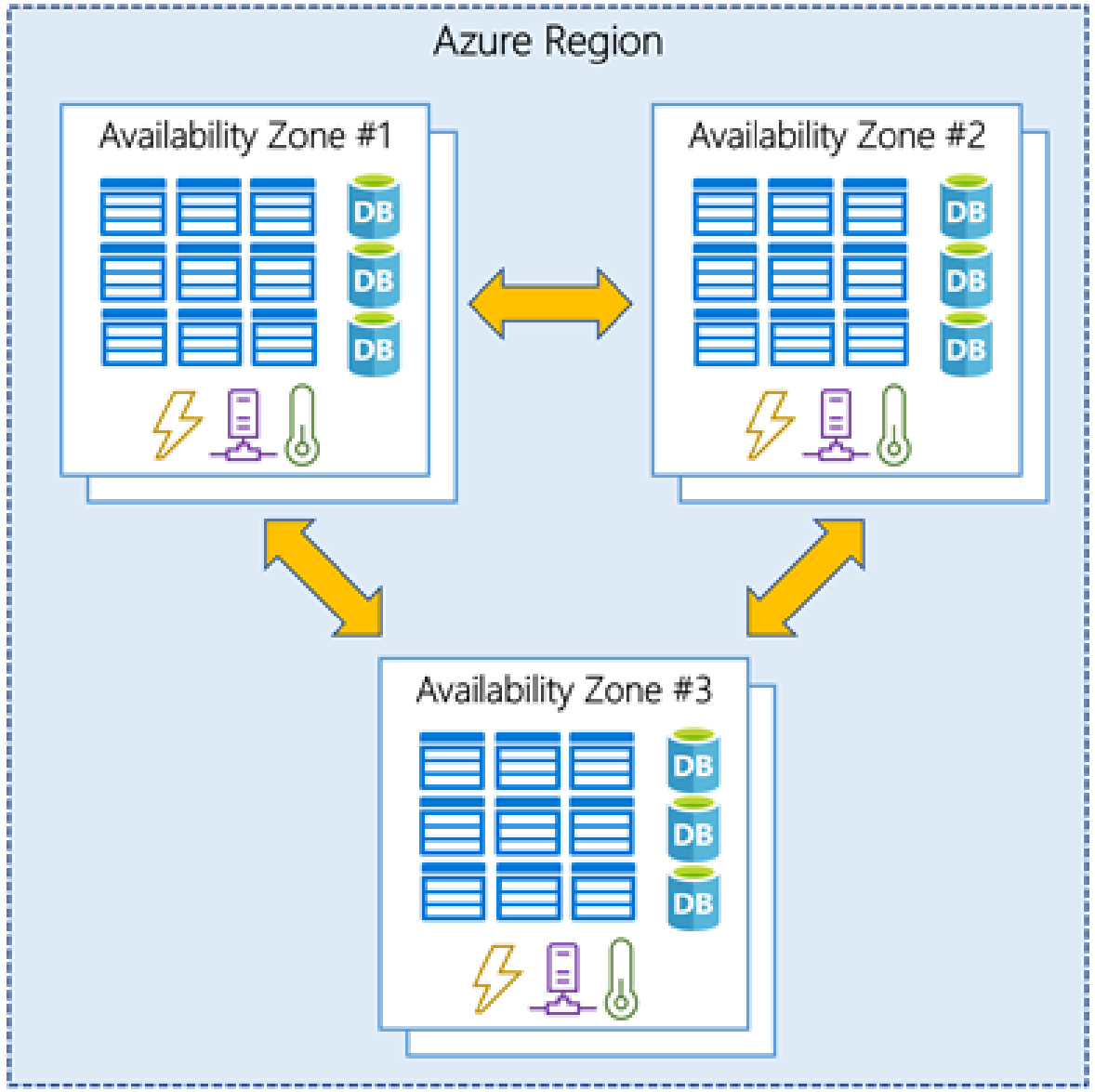
mongoDB® Atlas

Amazon DynamoDB

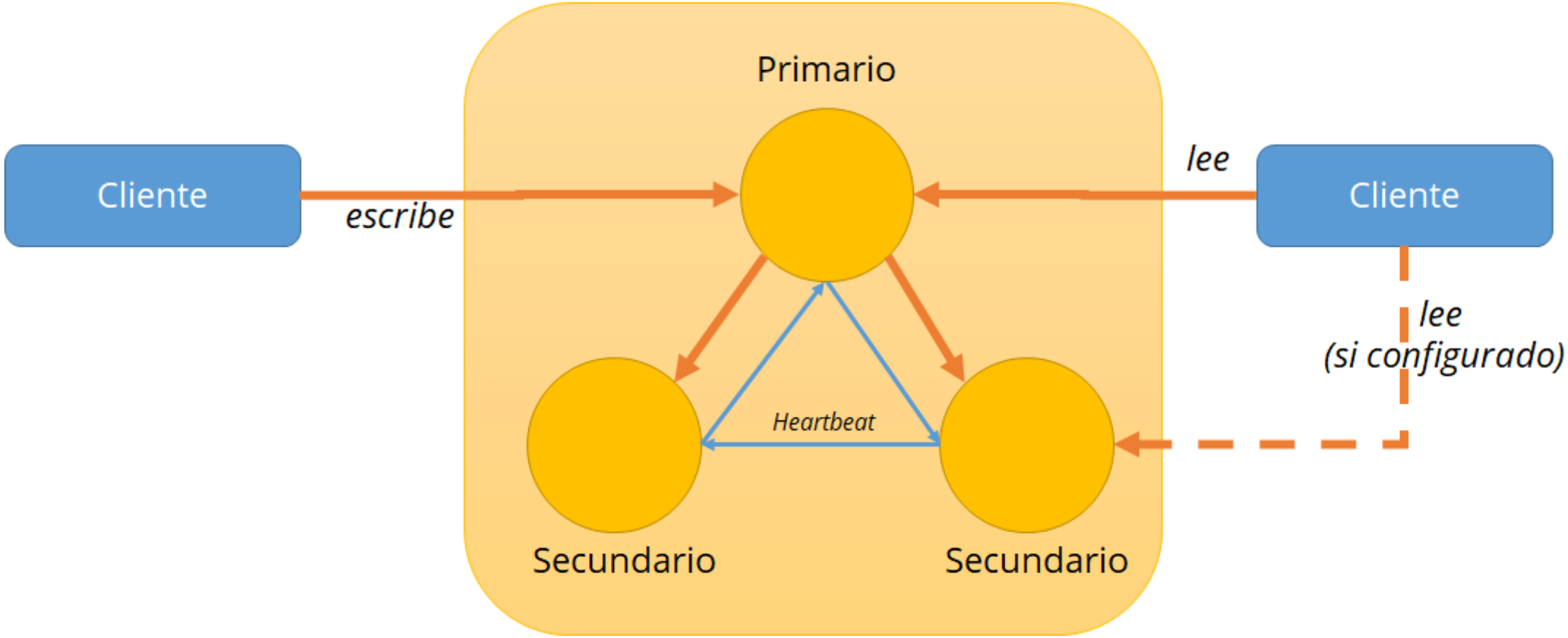
CLOUD COMPUTING

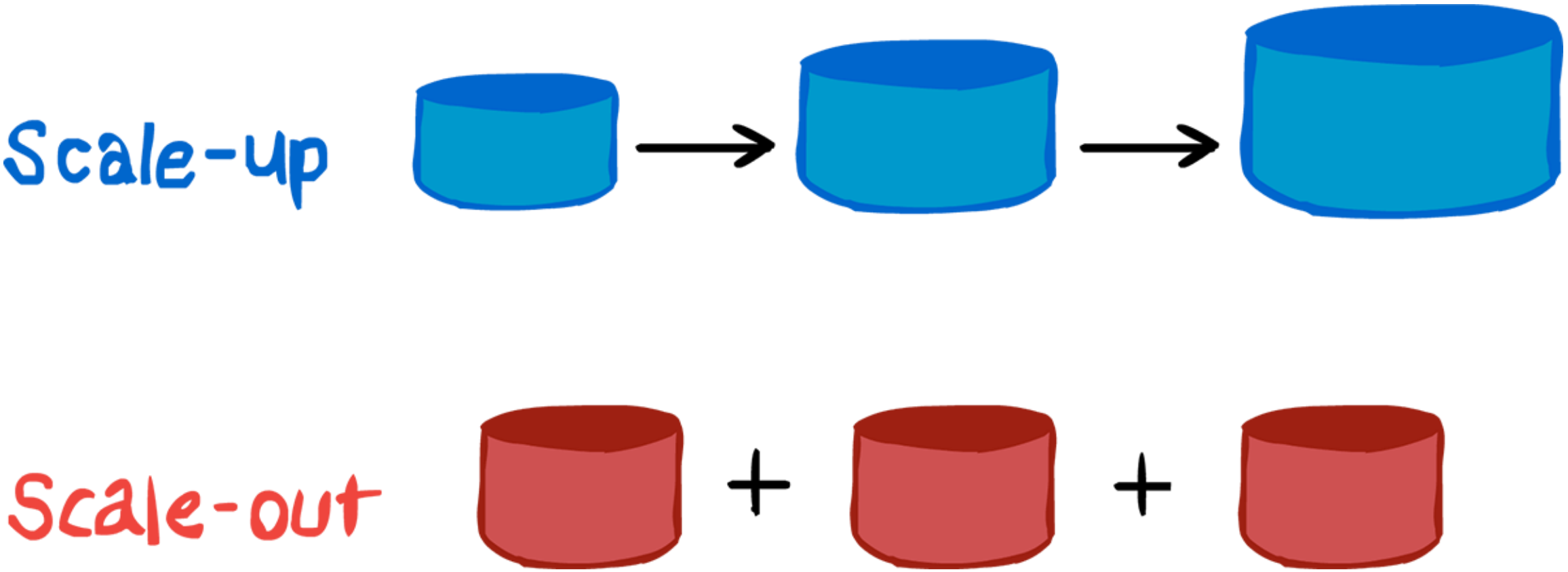


ALTA DISPONIBILIDAD



REPLICACIÓN

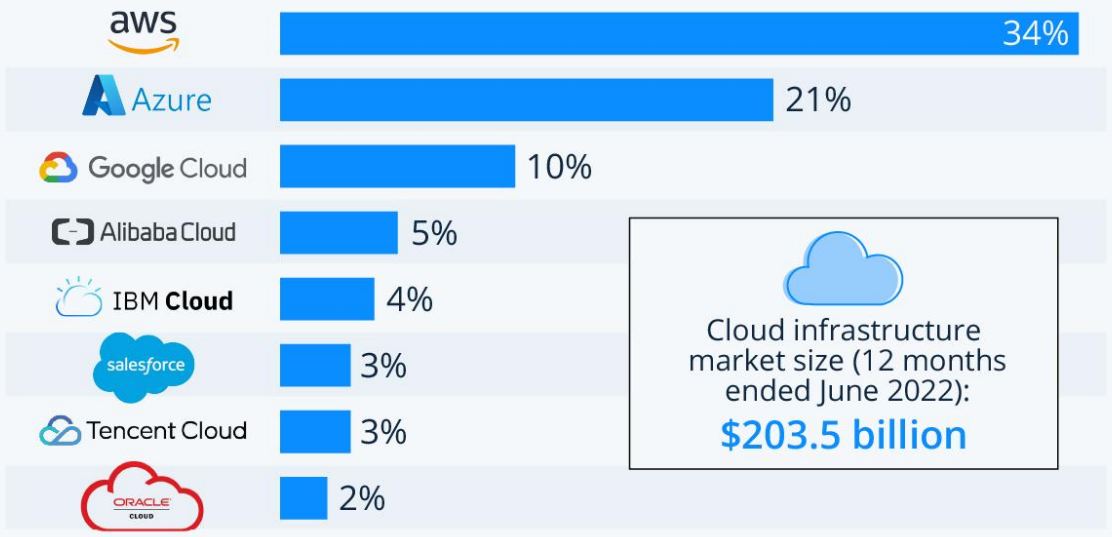




ESCALABILIDAD

Amazon Leads $200-Billion Cloud Market

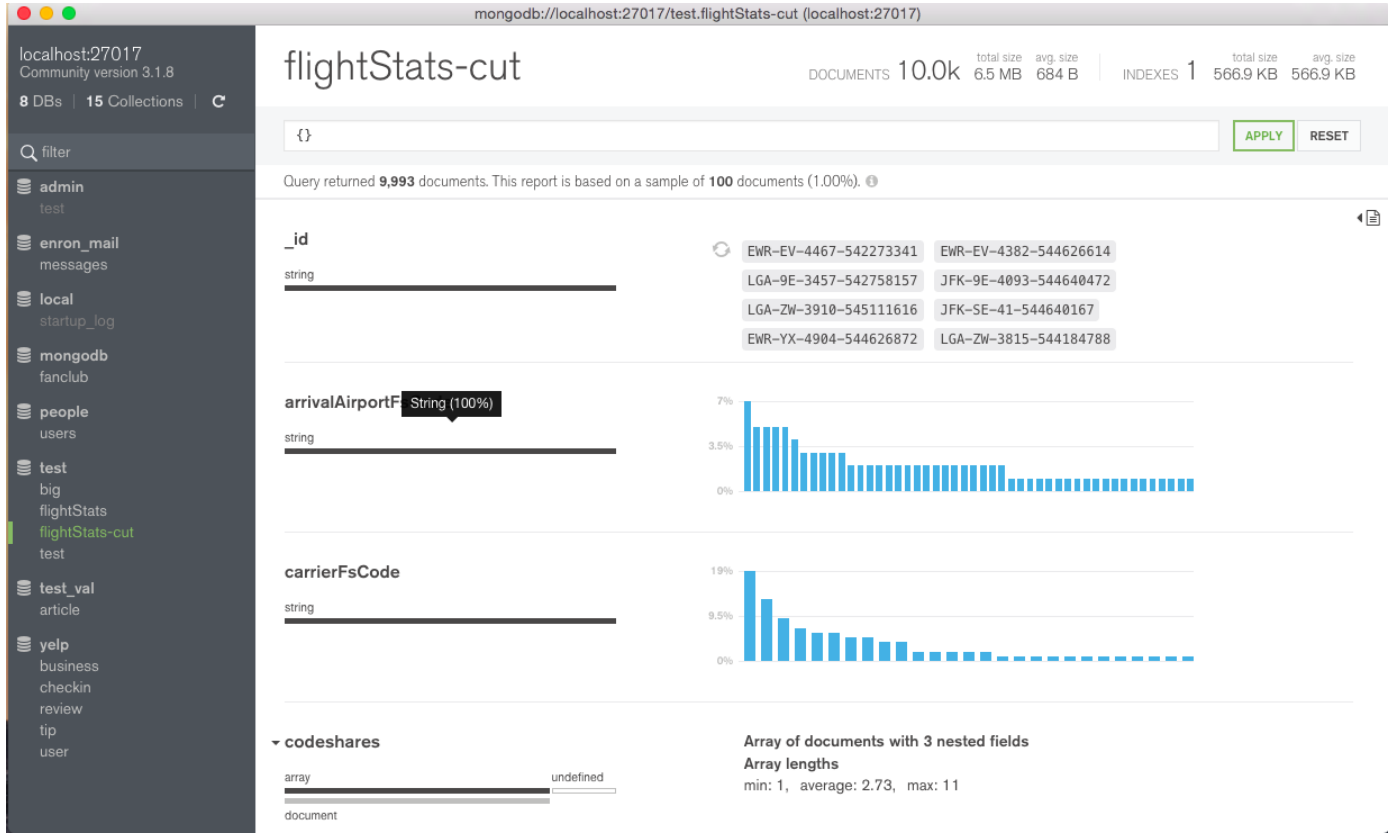
Worldwide market share of leading cloud infrastructure service providers in Q2 2022*



* includes platform as a service (PaaS) and infrastructure as a service (IaaS) as well as hosted private cloud services

Source: Synergy Research Group

MONGODB COMPASS




MongoDB
University
APK4Fun

myCompiler

Español | Reciente | Iniciar sesión | Inscríbase

Introduce un título...

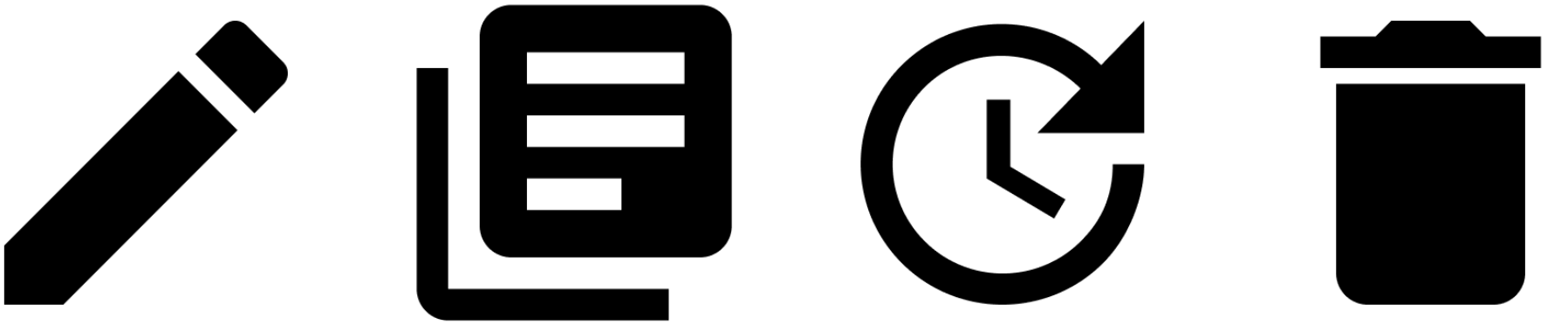
MongoDB

Ejecute | Guardar

```
1 db.students.insertMany([
2   { id: 1, name: 'Ryan', gender: 'M' },
3   { id: 2, name: 'Joanna', gender: 'F' }
4 ]);
5 db.students.find({ gender: 'F' });
```

Salida del programa

```
(Ejecute el programa para ver su salida)
```



Create Read Update Delete

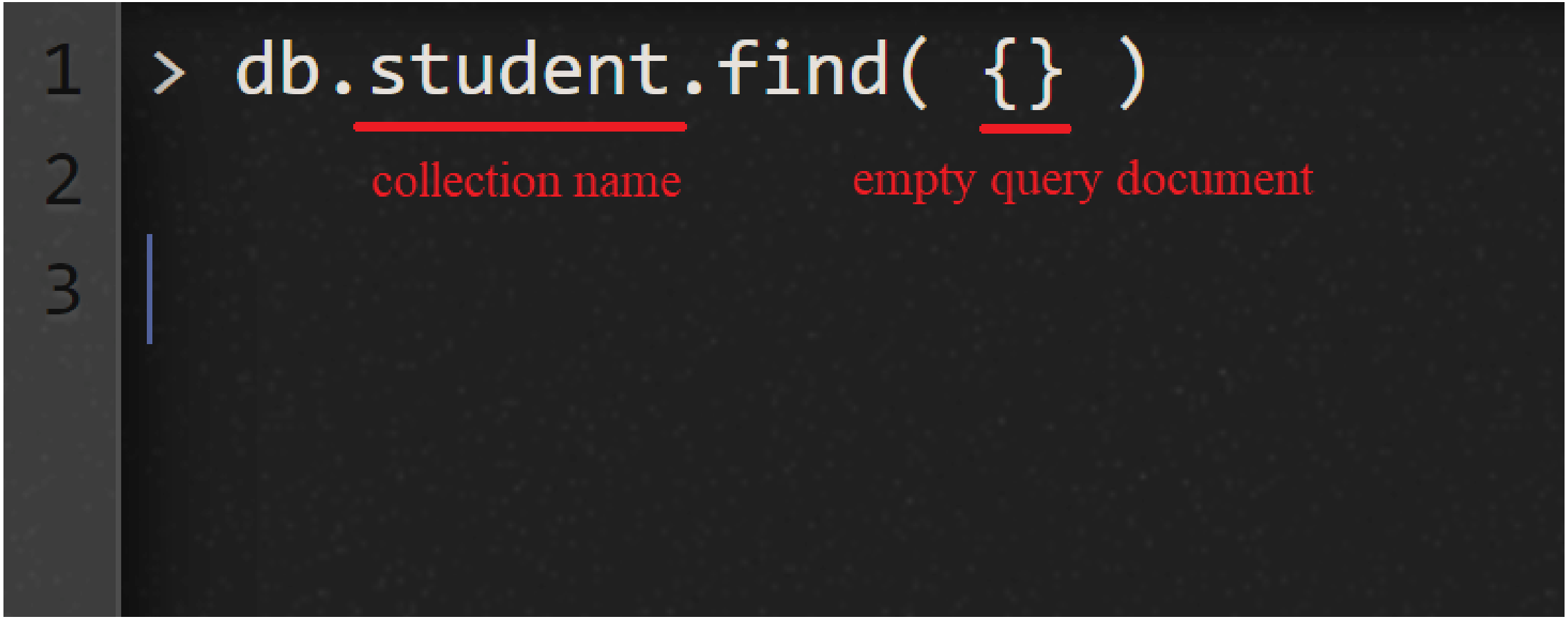
INSERTAR DOCUMENTOS

Para inserta un documento o un conjunto de documentos disponemos de los métodos:

- insertOne: Inserta un documento en una colección.
- insertMany: Inserta múltiples documentos en una colección.

CAMPO OBLIGATORIO _ID

- En MongoDB, cada documento almacenado en una colección requiere un único _id que actúa como clave principal . Si se inserta documento omite el _id, el controlador MongoDB automáticamente genera un ObjectId para el _id.



RECUPERAR DOCUMENTOS

OPERADORES COMPARACIÓN

- $eq - equal - igual
- $lt - low than - menor que
- $lte - low than equal - menor o igual que
- $gt - greater than - mayor que
- $gte - greater than equal - mayor o igual que
- $ne - not equal - distinto
- $in - in - dentro de
- $nin - not in - no dentro de

db.collection.deleteOne()

db.collection.deleteMany()

db.collection.remove()

ELIMINAR DOCUMENTOS Y BBDD

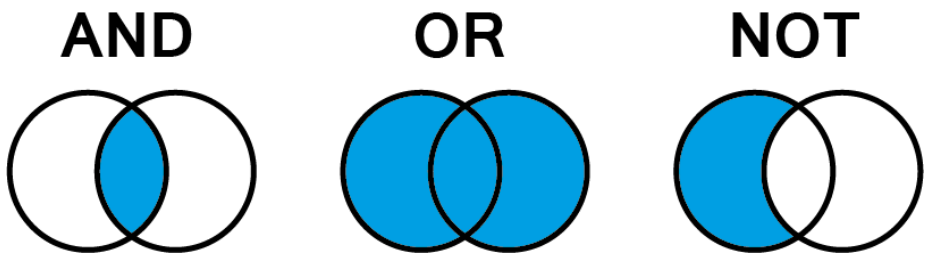
MODIFICAR UN ELEMENTO

```
> db.job.updateOne({salary:5000}, { $set: {firstName:'Morgan'}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.job.find().pretty()
{
        "_id" : ObjectId("6286f50681063f5a55d4775b"),
        "firstName" : "Morgan",
        "lastName" : "Dew",
        "email" : "john.dew@abc.com",
        "salary" : 5000
}
```

MODIFICAR MÚLTIPLES ELEMENTOS

```
db.users.updateMany(
   { age: { $lt: 18 } },
   { $set: { status: "reject" } }
)
```

OPERADORES LÓGICOS



CURSORES Y MÉTODOS

RECUPERAR SOLO ALGUNOS CAMPOS

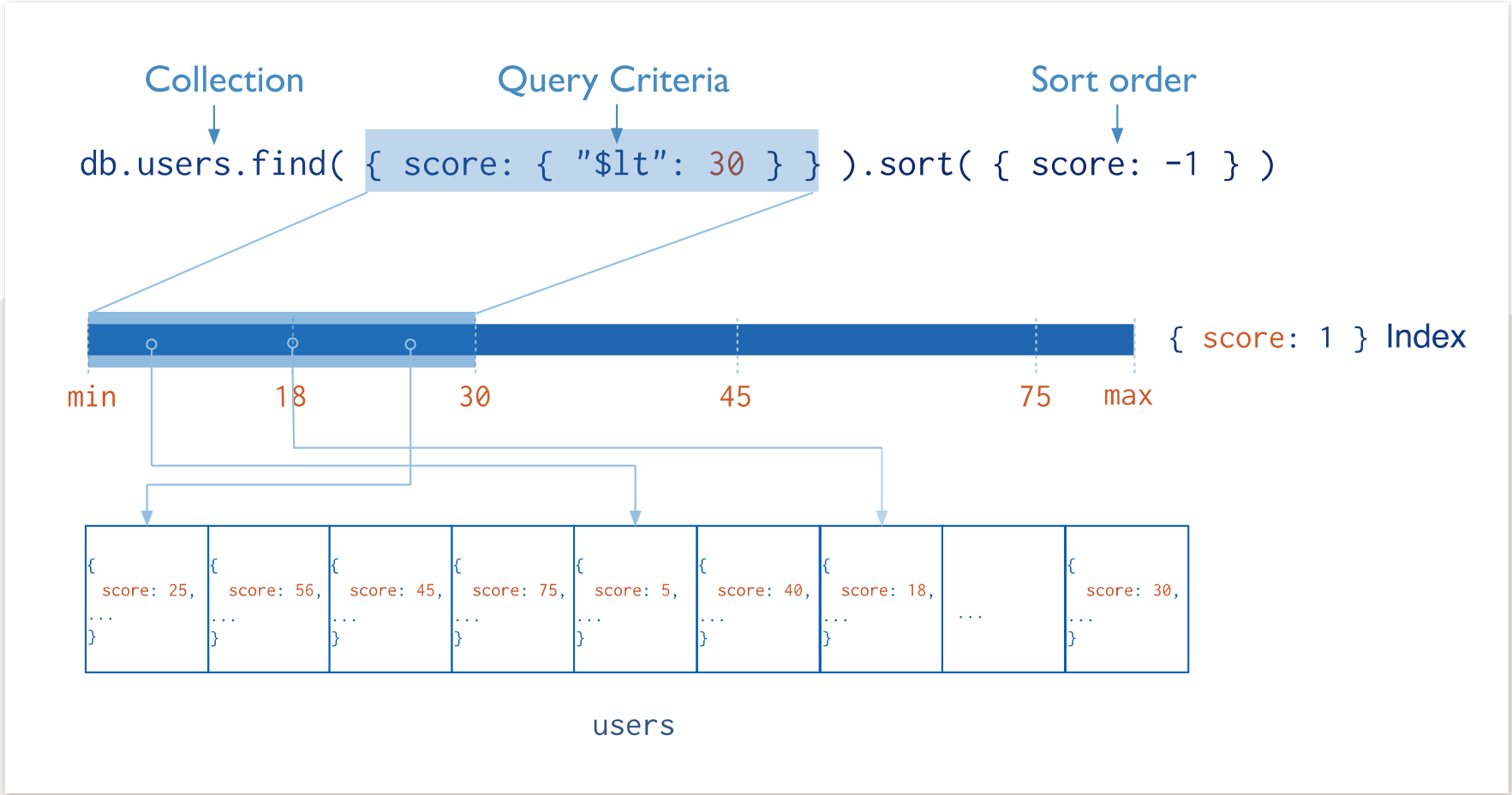
- Hemos visto que el método 'find’:
  - Si no le pasamos parámetros nos retorna todos los documentos de la colección que hace referencia: db.libros.find({precio: 50},{titulo:1,cantidad:1,_id:0})
  - El primer parámetro en el caso que lo indiquemos filtra la colección y recupera los documentos que cumplen la condición
  - En el segundo parámetro del método 'find' debemos especificar cada campo y un valor 1 indicando que se lo quiere recuperar.

DOCUMENTOS EMBEBIDOS

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

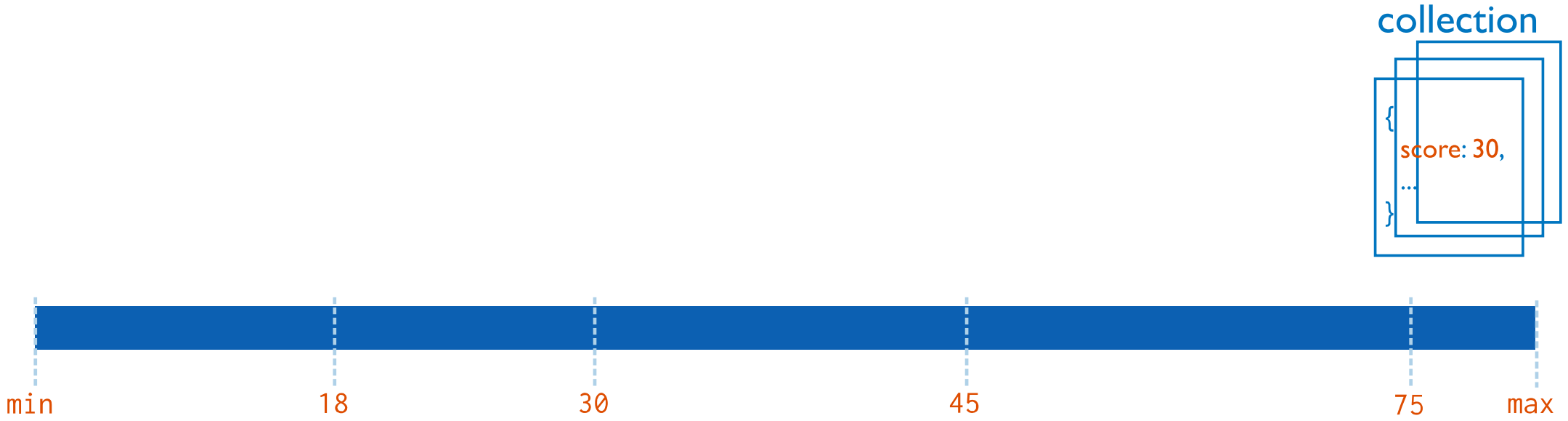
Embedded sub-document

Embedded sub-document

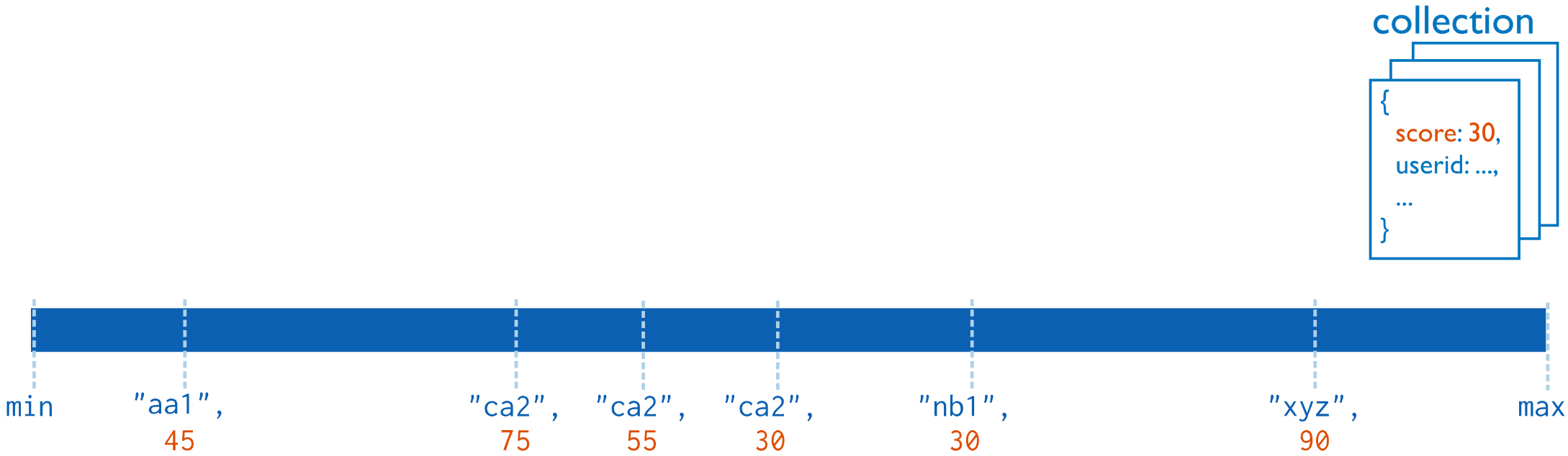

Collection
Query Criteria
Sort order
db.users.find({ score: { "$lt": 30 } }).sort({ score: -1 })
{ score: 1 } Index
min
18
30
45
75
max
{ score: 25, ... }
{ score: 56, ... }
{ score: 45, ... }
{ score: 75, ... }
{ score: 5, ... }
{ score: 40, ... }
{ score: 18, ... }
...
{ score: 30, ... }
users

ÍNDICES

ÍNDICE SIMPLE



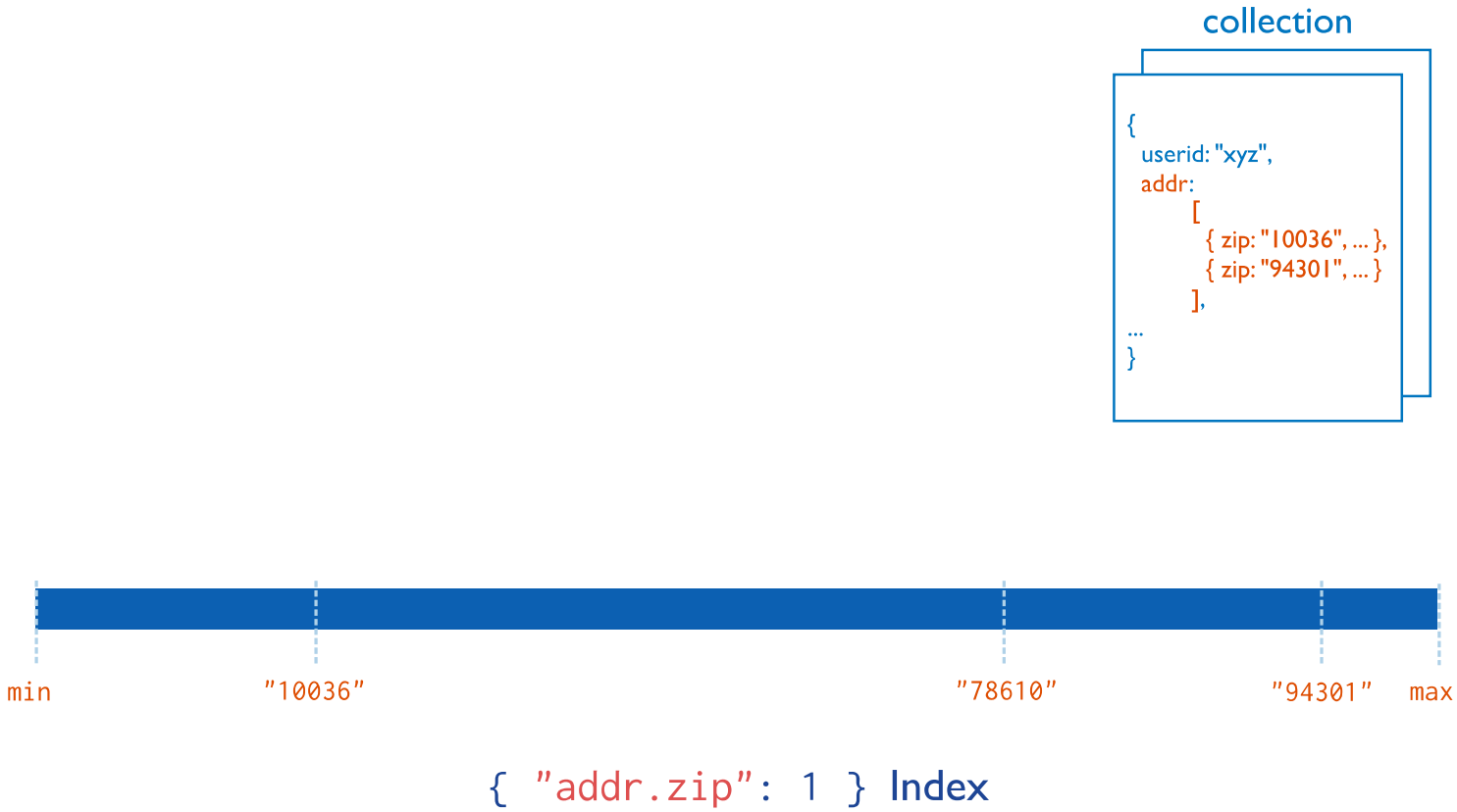
`{ score: 1 }` **Index**

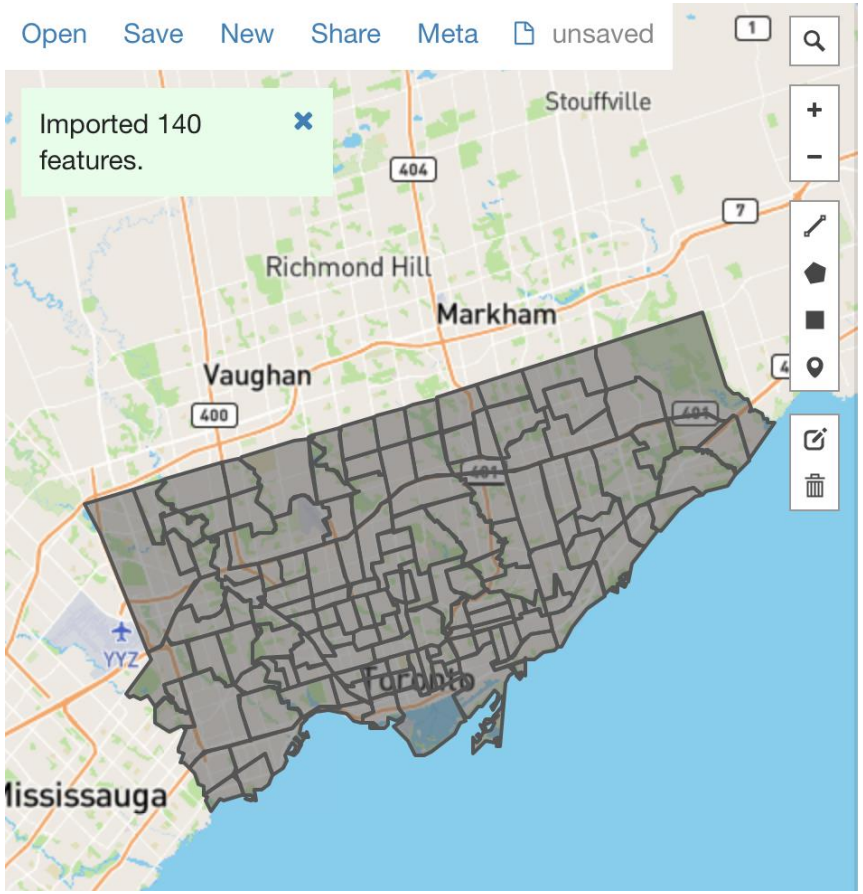


{ userid: 1, score: -1 } **Index**

ÍNDICE COMPUESTO

ÍNDICE MULTIKEY





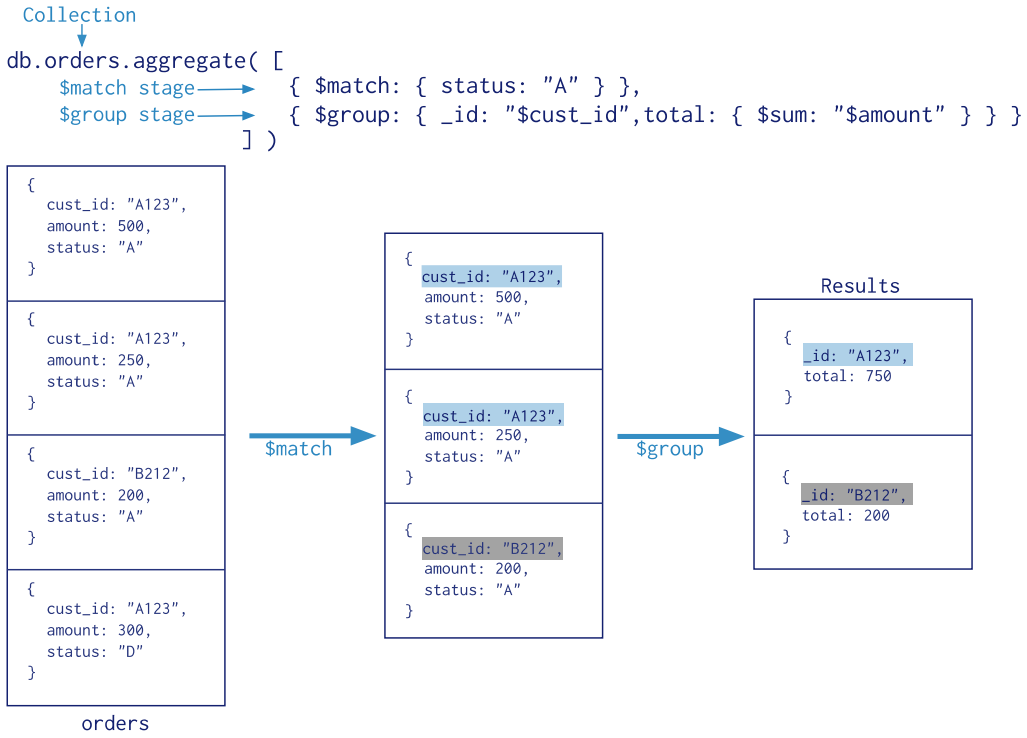

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              -79.4359157087306
              43.6801533947749
            ],
            [
              -79.4359116034227
              43.679547504996
            ],
            [
              -79.4337822305274
              43.6744077713687
```


GEOJSON

ÍNDICE GEOESPACIAL

- db.collection.createIndex({ <location field> : "2dsphere" })
- db.collection.createIndex({ <location field> : "2d" })

AGGREGATION PIPELINE



$match

```
db.universities.aggregate([
  { $match : { country : 'Spain', city : 'Salamanca' } }
]).pretty()
```

$project

```
db.universities.aggregate([
  { $project : { _id : 0, country : 1, city : 1, name : 1 } }
]).pretty()
```

$group

```
db.universities.aggregate([
  { $group : { _id : '$name', totaldocs : { $sum : 1 } } }
]).pretty()
```

$out

```
db.universities.aggregate([
  { $group : { _id : '$name', totaldocs : { $sum : 1 } } },
  { $out : 'aggResults' }
])
```

$unwind

```
db.universities.aggregate([
  { $match : { name : 'USAL' } },
  { $unwind : '$students' }
]).pretty()
```

$sort

```
db.universities.aggregate([
  { $match : { name : 'USAL' } },
  { $unwind : '$students' },
  { $project : { _id : 0, 'students.year' : 1, 'students.number' : 1 } },
  { $sort : { 'students.number' : -1 } }
]).pretty()
```

$limit

```
db.universities.aggregate([
  { $match : { name : 'USAL' } },
  { $unwind : '$students' },
  { $project : { _id : 0, 'students.year' : 1, 'students.number' : 1 } },
  { $sort : { 'students.number' : -1 } },
  { $limit : 2 }
]).pretty()
```

$addFields

```
db.universities.aggregate([
  { $match : { name : 'USAL' } },
  { $addFields : { foundation_year : 1218 } }
]).pretty()
```

$count

```
db.universities.aggregate([
  { $unwind : '$students' },
  { $count : 'total_documents' }
]).pretty()
```

$lookup

```
db.universities.aggregate([
  { $match : { name : 'USAL' } },
  { $project : { _id : 0, name : 1 } },
  { $lookup : {
      from : 'courses',
      localField : 'name',
      foreignField : 'university',
      as : 'courses'
  } }
]).pretty()
```

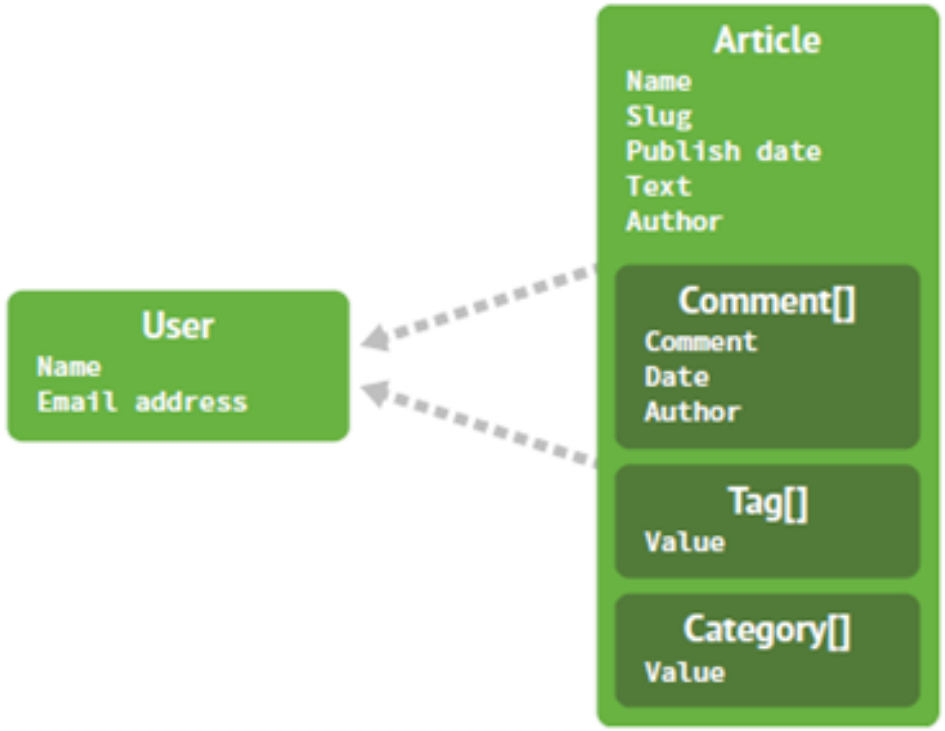
$sortByCount

```
db.courses.aggregate([
  { $sortByCount : '$level' }
]).pretty()
```

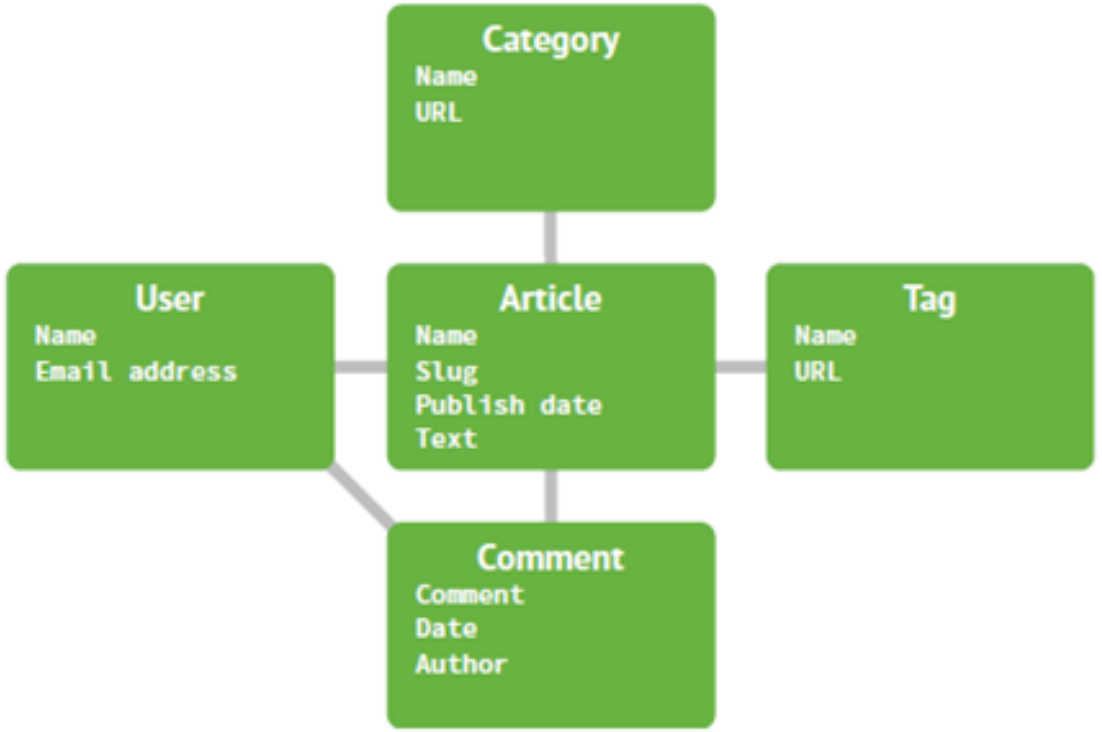
SQL VS MONGODB

- https://www.mongodb.com/docs/manual/reference/sql-aggregation-comparison/

SQL Terms, Functions, and Concepts	MongoDB Aggregation Operators
WHERE	`$match`
GROUP BY	`$group`
HAVING	`$match`
SELECT	`$project`
ORDER BY	`$sort`
LIMIT	`$limit`
SUM()	`$sum`
COUNT()	`$sum` `$sortByCount`
join	`$lookup`
SELECT INTO NEW_TABLE	`$out`
MERGE INTO TABLE	`$merge` (Available starting in MongoDB 4.2)
UNION ALL	`$unionWith` (Available starting in MongoDB 4.4)



MongoDB Data Model



ScaleBase Data Model

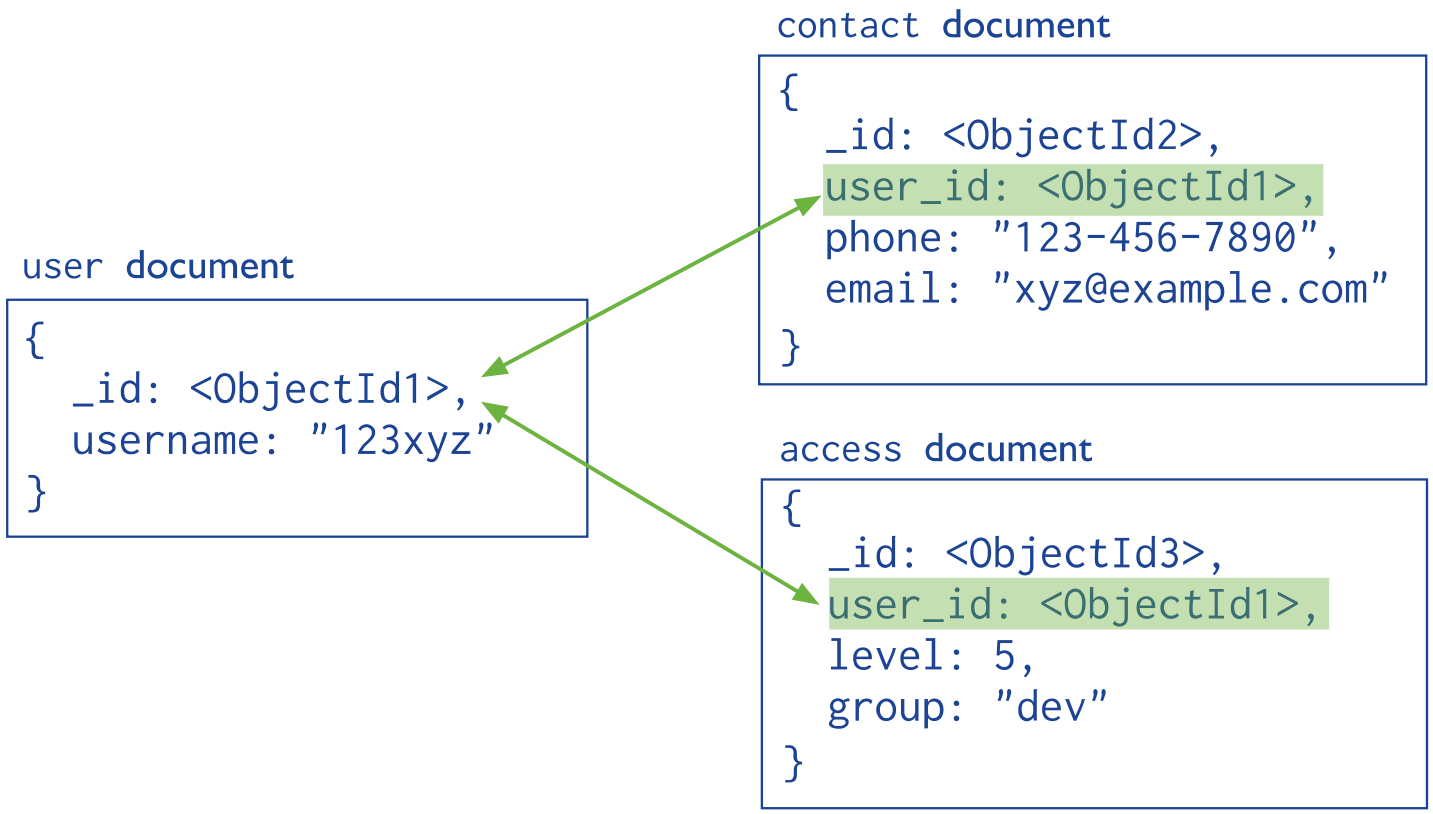
DATA MODEL

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

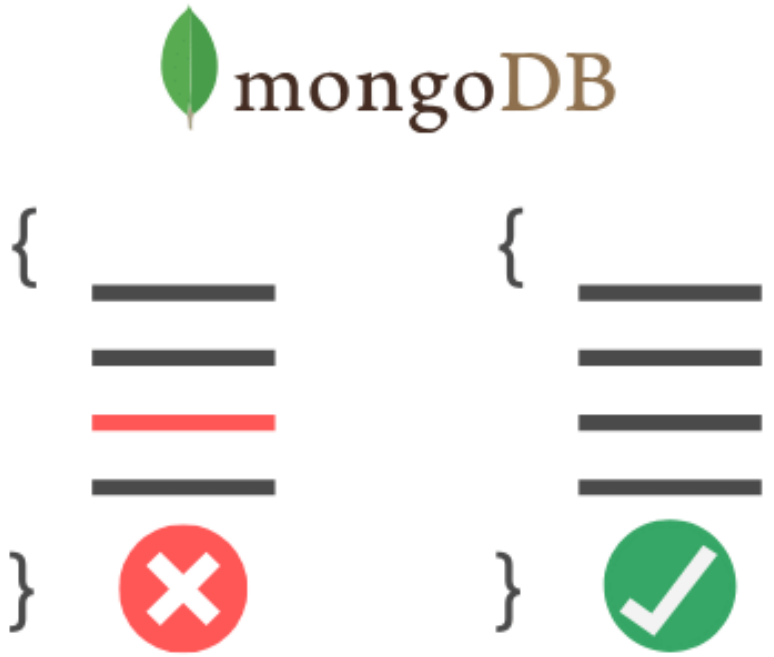
Embedded sub-document

Embedded sub-document

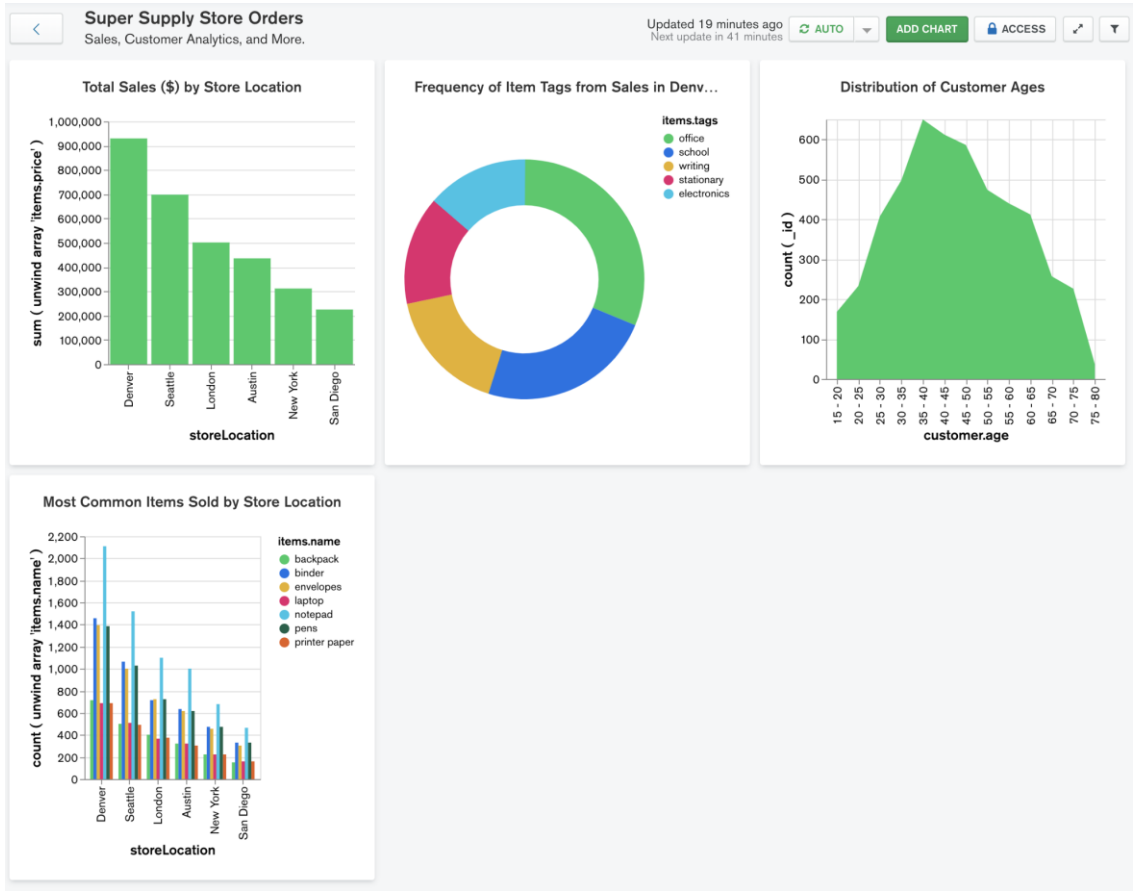
DOCUMENTOS EMBEBIDOS



REFERENCIAS



ESQUEMA DE VALIDACIÓN

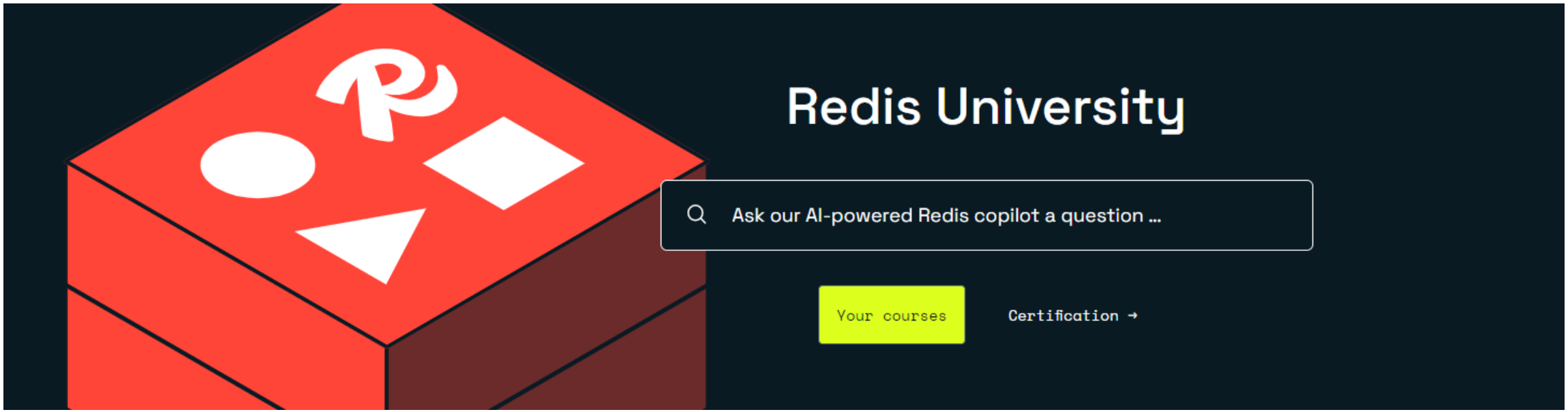

Super Supply Store Orders
Sales, Customer Analytics, and More.
Updated 19 minutes ago
Next update in 41 minutes
AUTO
ADD CHART
ACCESS
Total Sales ($) by Store Location
sum (unwind array 'items.price')
1,000,000
900,000
800,000
700,000
600,000
500,000
400,000
300,000
200,000
100,000
0
Denver
Seattle
London
Austin
New York
San Diego
storeLocation
Frequency of Item Tags from Sales in Denv...
items.tags
office
school
writing
stationary
electronics
Distribution of Customer Ages
count (_id)
600
500
400
300
200
100
0
15 - 20
20 - 25
25 - 30
30 - 35
35 - 40
40 - 45
45 - 50
50 - 55
55 - 60
60 - 65
65 - 70
70 - 75
75 - 80
customer.age
Most Common Items Sold by Store Location
count (unwind array 'items.name')
2,200
2,000
1,800
1,600
1,400
1,200
1,000
800
600
400
200
0
Denver
Seattle
London
Austin
New York
San Diego
storeLocation
items.name
backpack
binder
envelopes
laptop
notepad
pens
printer paper

CHARTS

EJERCICIO

https://www.mongodb.com/docs/charts/tutorial/movie-details/movie-details-tutorial-overview/

- REDIS CLOUD


Redis University
Ask our AI-powered Redis copilot a question ...
Your courses
Certification →