Create your account

Have an account? Log in now

G Google

⊙ GitHub

— Or with email and password —

Email Address
We recommend using your work email

bilal@smartaos.com ✓

First Name

bilal ✓

Last Name

namir ✓

Password

•••••••• ∅

✔ Must be at least 8 characters
✔ Does not contain your email address

Company Name

bilaanaam ✓

☐ I accept the Privacy Policy and
the Terms of Service

Sign up

Pantalla de registre amb el formulari de dades.

ORGANIZATION
Bilal's Org - 2025-12-18 ▼

PROJECT
Project 0 ▼

We are deploying your changes (current action: creating a plan)

Clusters

🔍 Find a database deployment...

🗄 Edit Config ▼   ＋ Create

⚬ **Cluster0**   Connect   View Monitoring   Browse Collections   ...

ORGANIZATION
Bilal's Org - 2025-12-18 ▼

PROJECT
Project 0 ▼

Configurations

Quick setup

☐ Preload sample dataset ⓘ

Name
You cannot change the name once the cluster is created.

Cluster1

Provider

aws   Google Cloud   Azure

Region

🇪🇸 Spain (eu-south-2) ★ 🌿   ▼

★ Recommended ⓘ   🌿 Low carbon emissions ⓘ

Tag (optional)
Create your first tag to categorize and label your resources; more tags can be added later. Learn more.

environment   :   test

Included features

Auto-scale
Atlas enables auto-scaling for cluster storage and cluster tier. Learn more.

✔ Storage Scaling ⓘ   ✔ Cluster Tier Scaling ⓘ

Cloud Backup
Snapshots are taken automatically and stored according to your backup and retention policy. Learn more.
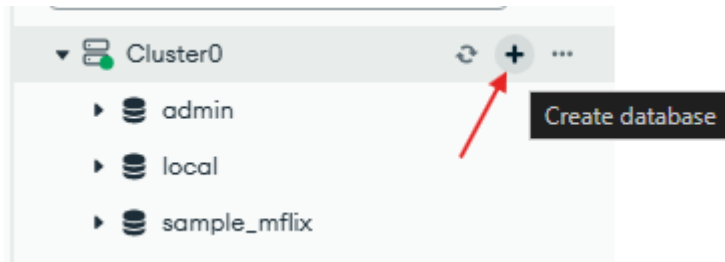
✔ Continuous Cloud Backup ⓘ

I'll do this later   Go to Advanced Configuration   Create Deploy

Un cop creat, cliquem **collection.**

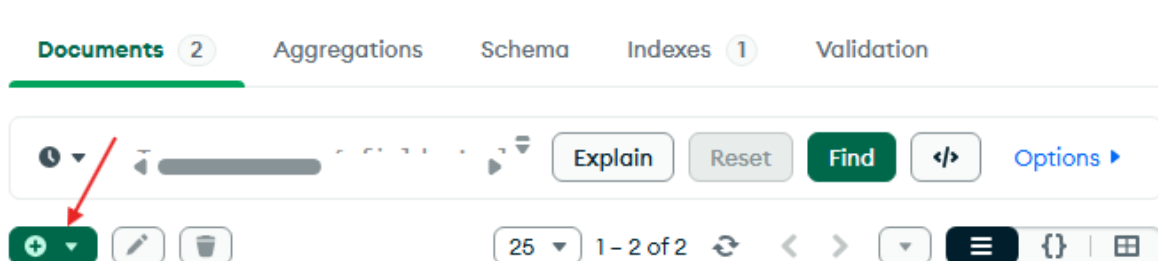**Clica Create Database**
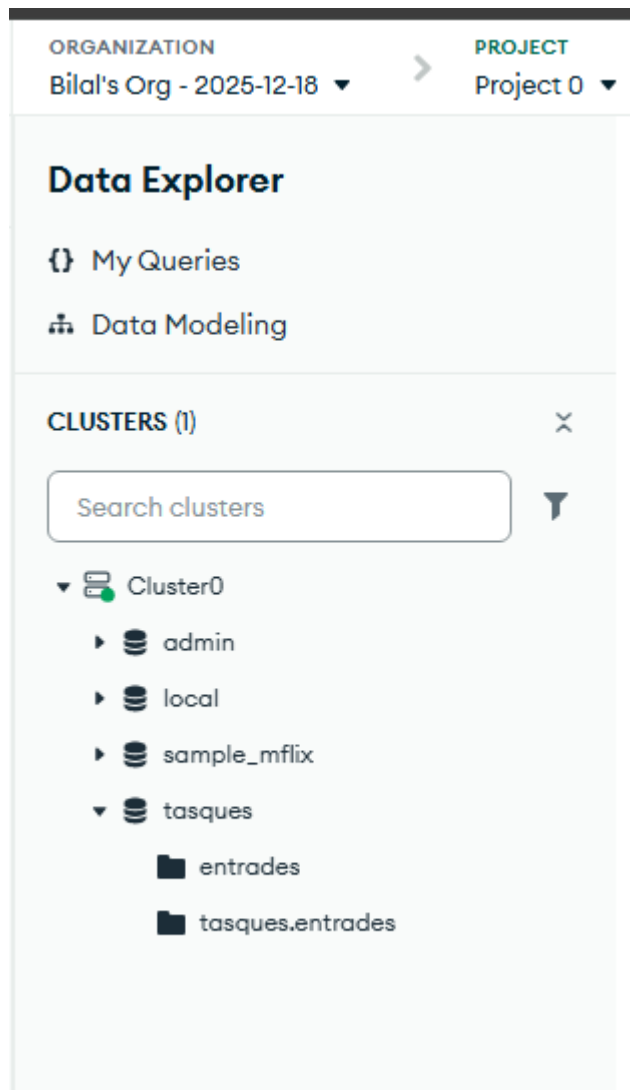
Clica la col·leccio entrades.

Clica Insert Document.

Copia i enganxa el següent codi JSON:

Insert Document

To collection tasques.entrades

VIEW {} ≡

```
 1 ▾ [
 2 ▾   {
 3         "nomAlumne": "Laura",
 4         "cognom1": "Martínez",
 5         "cognom2": "Soler",
 6         "dataEntradaTasca": "2025-12-10T09:30:00Z",
 7         "completa": true,
 8         "observacions": "Tasca entregada abans de la data límit"
 9       },
10 ▾   {
11         "nomAlumne": "Pol",
12         "cognom1": "García",
13         "cognom2": "López",
14         "dataEntradaTasca": "2025-12-12T14:15:00Z",
15         "completa": false,
16         "observacions": "Falten alguns exercicis"
17       },
18 ▾   {
19         "nomAlumne": "Anna",
20         "cognom1": "Vidal",
21         "cognom2": "Riera",
22         "dataEntradaTasca": "2025-12-14T18:45:00Z",
23         "completa": true,
24         "observacions": "Molt bona presentació i contingut"
25       }
26     ]
27     |
```

Cancel   Insert

Cluster0 > tasques > tasques.entrades

View monitoring   Visualize Your Data

Documents 3   Aggregations   Schema   Indexes 1   Validation
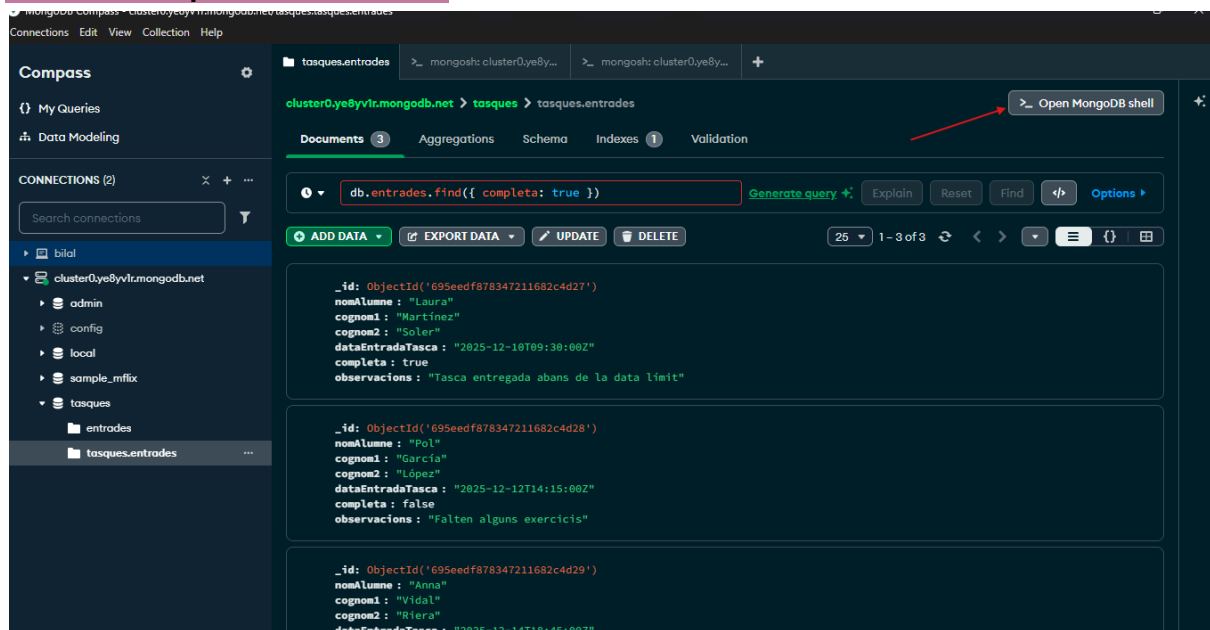
Explain   Reset   Find   </>   Options ▶

25 ▼   1 – 3 of 3

```
_id: ObjectId('695eedf878347211682c4d27')
nomAlumne : "Laura"
cognom1 : "Martínez"
cognom2 : "Soler"
dataEntradaTasca : "2025-12-10T09:30:00Z"
completa : true
observacions : "Tasca entregada abans de la data límit"
```

```
_id: ObjectId('695eedf878347211682c4d28')
nomAlumne : "Pol"
cognom1 : "García"
cognom2 : "López"
dataEntradaTasca : "2025-12-12T14:15:00Z"
completa : false
observacions : "Falten alguns exercicis"
```

```
_id: ObjectId('695eedf878347211682c4d29')
nomAlumne : "Anna"
cognom1 : "Vidal"
cognom2 : "Riera"
dataEntradaTasca : "2025-12-14T18:45:00Z"
completa : true
observacions : "Molt bona presentació i contingut"
```

Realitzar cerques a la col·lecció

Llistar totes les tasques

Llistar les tasques completades

```
db.tasques.entrades.find({ completa: true })
{
  _id: ObjectId('695eedf878347211682c4d27'),
  nomAlumne: 'Laura',
  cognom1: 'Martínez',
  cognom2: 'Soler',
  dataEntradaTasca: '2025-12-10T09:30:00Z',
  completa: true,
  observacions: 'Tasca entregada abans de la data límit'
}
{
  _id: ObjectId('695eedf878347211682c4d29'),
  nomAlumne: 'Anna',
  cognom1: 'Vidal',
  cognom2: 'Riera',
  dataEntradaTasca: '2025-12-14T18:45:00Z',
  completa: true,
  observacions: 'Molt bona presentació i contingut'
}
db.tasques.entrades.find()
{
  _id: ObjectId('695eedf878347211682c4d27'),
  nomAlumne: 'Laura',
  cognom1: 'Martínez',
```

// Llistar totes les tasques.
db.tasques.entrades.find()

// Llistar les tasques completades.
db.tasques.entrades.find({ completa: true })

// Llistar les tasques d'un alumne específic.
db.tasques.entrades.find({ nomAlumne: "Laura" })

// Cerca les tasques de determinat alumne i completades.
db.tasques.entrades.find({ nomAlumne: "Anna", completa: true })

// Llista les tasques completades i amb observacions.
db.tasques.entrades.find({ completa: true, observacions: { $exists: true, $ne: "" } })

// Llista les tasques completades després d'una certa data.
db.tasques.entrades.find({

```
  completa: true,
  dataEntradaTasca: { $gt: ISODate("2025-12-11T00:00:00Z") }
})
```

// Llista les tasques amb observacions que continguin una paraula específica.
```
db.tasques.entrades.find({ observacions: /exercicis/ })
```

// Llista les tasques no completades ordenades alfabèticament.
```
db.tasques.entrades.find({ completa: false }).sort({ nomAlumne: 1 })
```

// Llista les tasques entre dues dates específiques.
```
db.tasques.entrades.find({
  dataEntradaTasca: {
    $gte: ISODate("2025-12-10T00:00:00Z"),
    $lte: ISODate("2025-12-14T23:59:59Z")
  }
})
```

// Llista les tasques amb observacions que comencin amb una determinada paraula ordenades per data més nova a més antiga.
```
db.tasques.entrades.find({ observacions: /^Tasca/ }).sort({ dataEntradaTasca: -1 })
```

Activitat 2

Nom de l'aplicació:

EventBooker

Descripció
Aquesta aplicació és per gestionar esdeveniments i reserves.
Permet crear esdeveniments, buscar-los per data, categoria i preu, i fer operacions CRUD.

Justificació NoSQL
MongoDB és bo perquè:
- Les dades són flexibles
- No tots els esdeveniments tenen els mateixos camps
- Fem moltes consultes per filtres
- És escalable

Tipus de dades
- Text
- Números
- Dates
- Categories

Disseny de la Base de Dades

Col·lecció principal: events

Aquesta col·lecció té tots els esdeveniments de l'aplicació.

Camps:
- title (string): nom de l'esdeveniment
- category (string): tipus d'esdeveniment
- price (double): preu
- eventDate (date): data de l'esdeveniment (es obligatori)
- location (string): lloc
- available (boolean): si esta disponible

Justificació
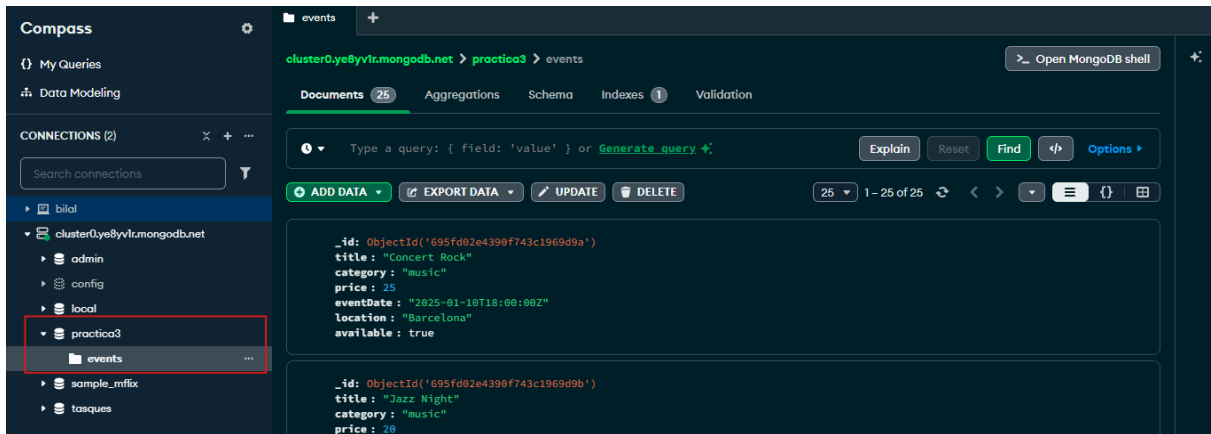**Hem fet una sola col·lecció principal perquè:**
- Les dades són molt relacionades
- Evitem unions innecessàries
- Fa facil les consultes

Esquema JSON Schema

```json
json:
{
  "bsonType": "object",
  "required": ["title", "category", "eventDate"],
  "properties": {
    "title": { "bsonType": "string" },
    "category": { "bsonType": "string" },
    "price": { "bsonType": "double" },
    "eventDate": { "bsonType": "date" },
    "location": { "bsonType": "string" },
    "available": { "bsonType": "bool" }
  }
}
```
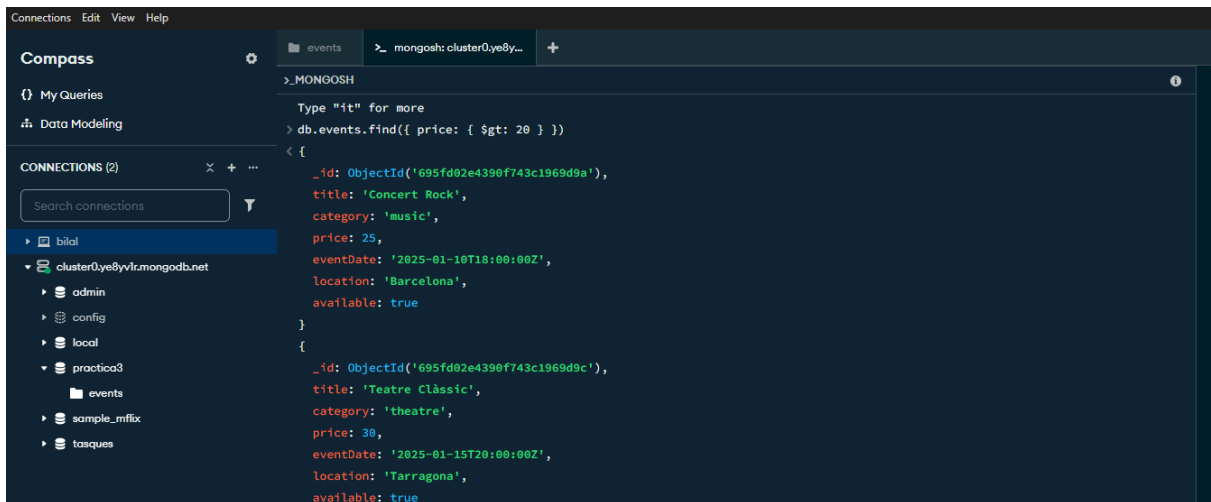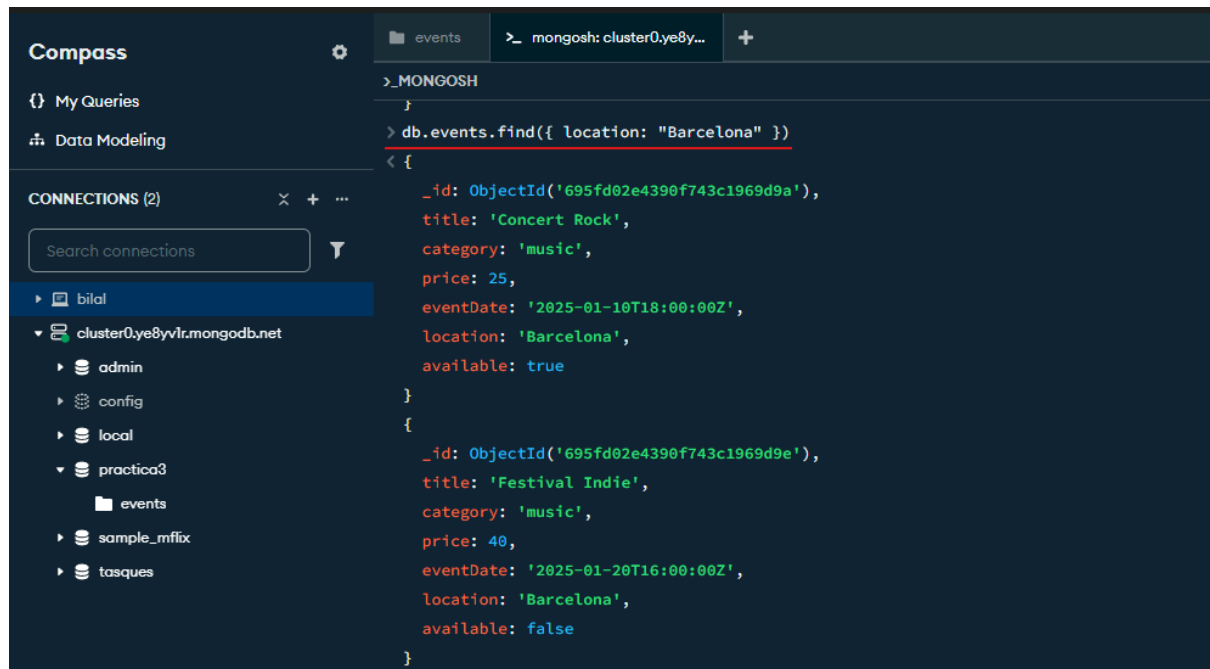

mongo compas

Consultes:

Filtrar per preu més gran que 20:
db.events.find({ price: { $gt: 20 } })

Filtrar per ciutat
db.events.find({ location: "Barcelona" })

Connexió Java-MongoDB i CRUD:
link github:

https://github.com/BILALNAMIR/practica3-nosql-mongodb/blob/main/mongoEvents/src/main/java/dam/uf3/mongo/App_practica3.java