

MongoDB.

CONEXIÓN Y CONSULTAS EN MONGODB

OBJETIVO

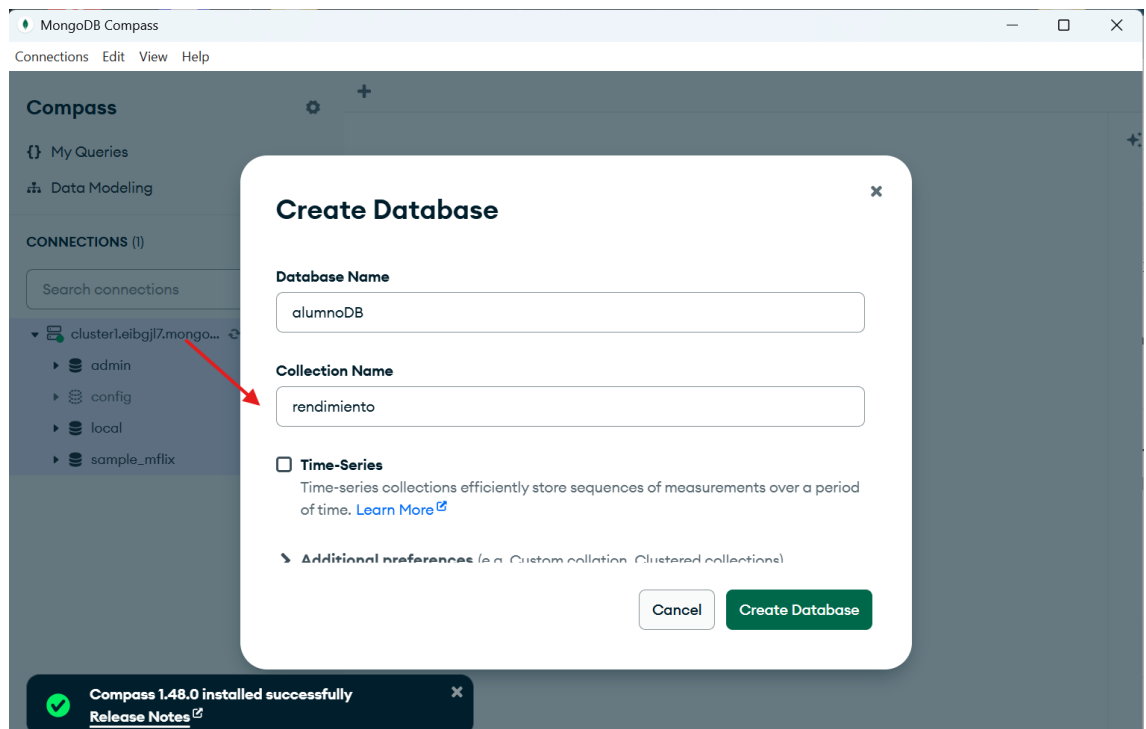
El objetivo de esta práctica es demostrar la capacidad de conectarse a un clúster de MongoDB, importar datos desde un CSV, y realizar consultas y agregaciones usando distintas herramientas: MongoDB Compass y Python.

DESARROLLO DE LA TAREA

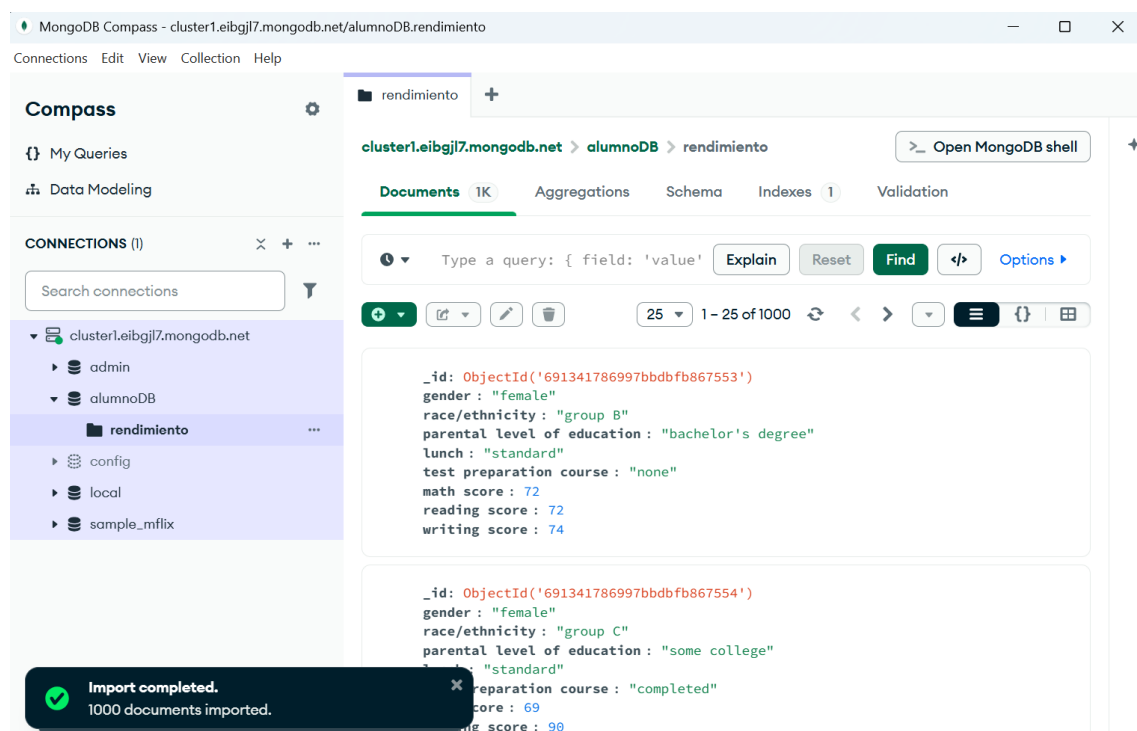
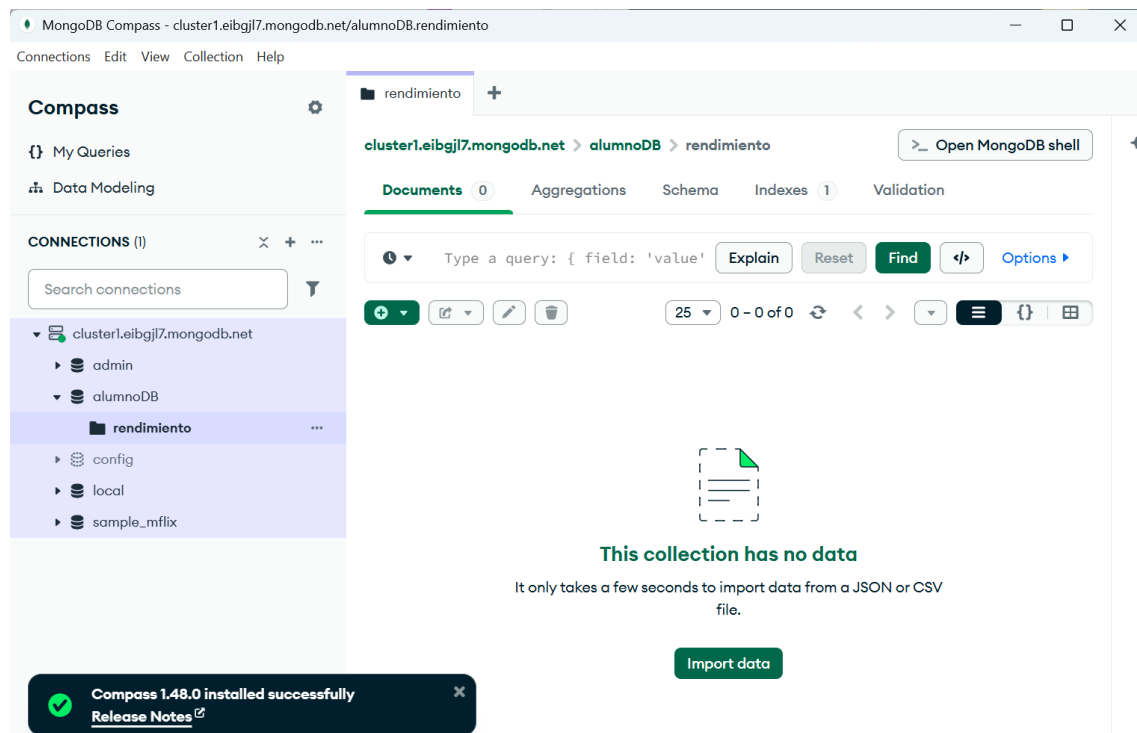
Para la tarea me descargo de Kaggle un archivo csv: Rendimiento de estudiantes, que contiene los siguientes campos:

- gender
- race/ethnicity
- parental level of education
- lunch
- test preparation course
- math score
- reading score
- writing score

Se creó un clúster en MongoDB Atlas y se importó el CSV a la base de datos **alumnoDB** y la colección **rendimiento**.



Importamos los datos



Explicación del proceso:

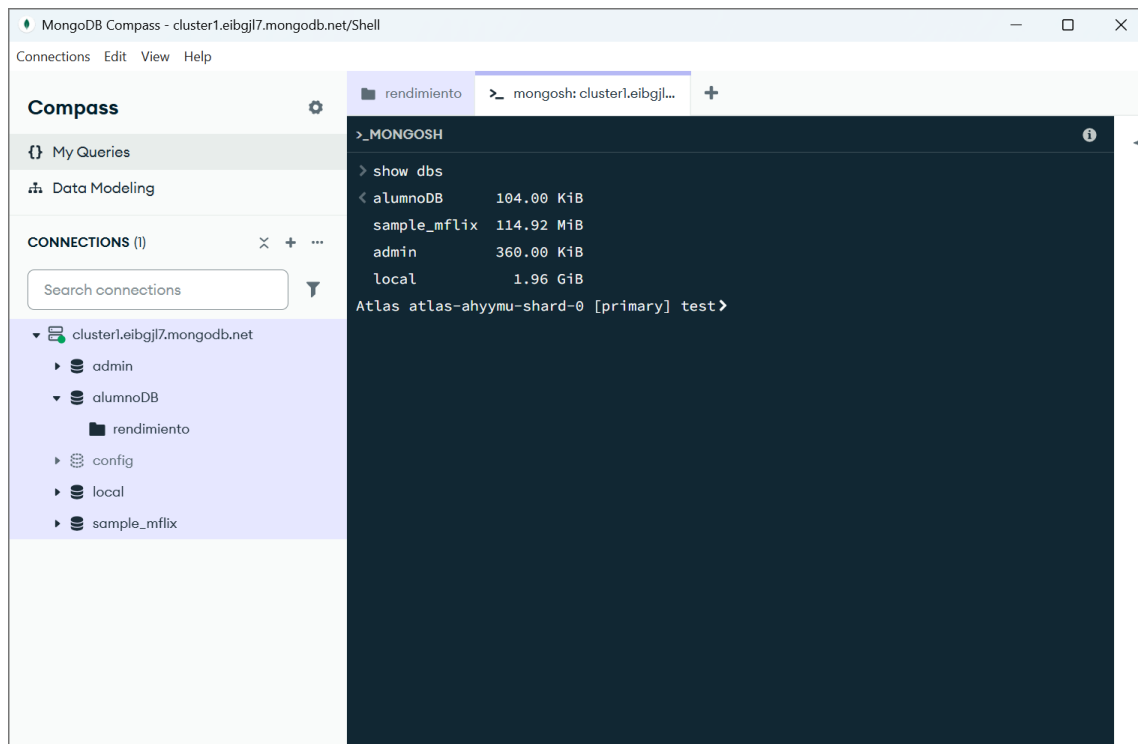
1. Abrir MongoDB Compass.
2. Crear la base de datos alumnoDB.
3. Crear la colección rendimiento.
4. Usar la opción **Import Data** y seleccionar el CSV.

La importación también se podría hacer de forma manual desde el terminal con el comando mongoimport:

```
mongoimport --uri "mongodb+srv://<user>:<pwd>@<cluster-url>/alumnoDB" \  
--collection rendimiento \  
--type csv --headerline \  
--file /ruta/a/rendimiento.csv
```

LISTAR BASES DE DATOS DEL SERVIDOR

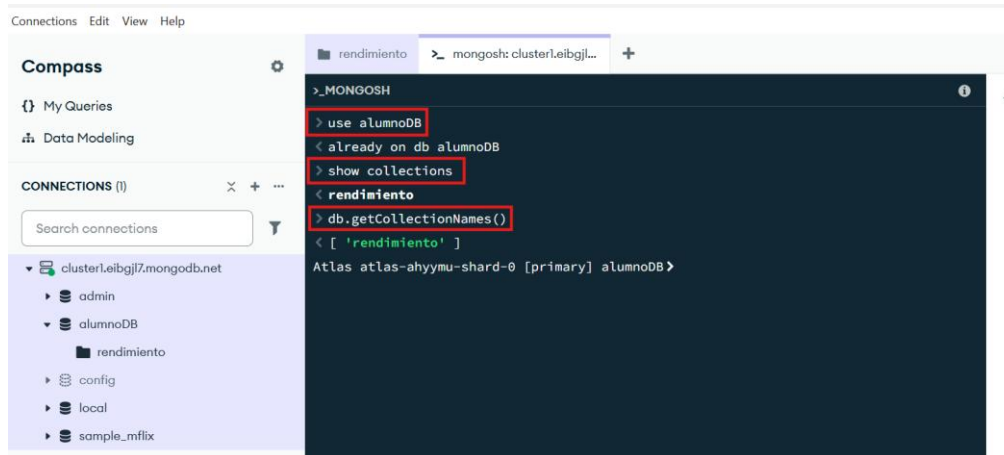
Show dbs



Para realizar consultas de una Base de Datos, lo primero que tendremos que hacer, será ubicarnos en ella → ***use <nombre base de datos>***

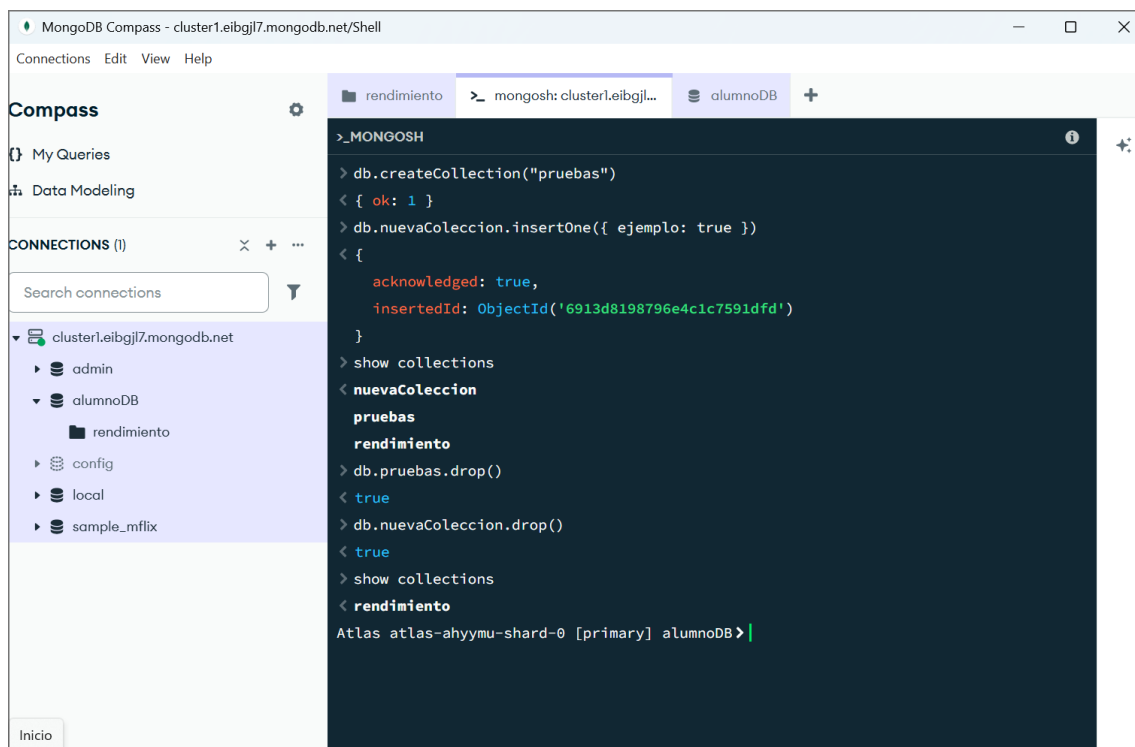
LISTAR COLECCIONES DE UNA BASE DE DATOS

Podremos utilizar ***show collections*** o con JS ***db.getCollectionNames()***



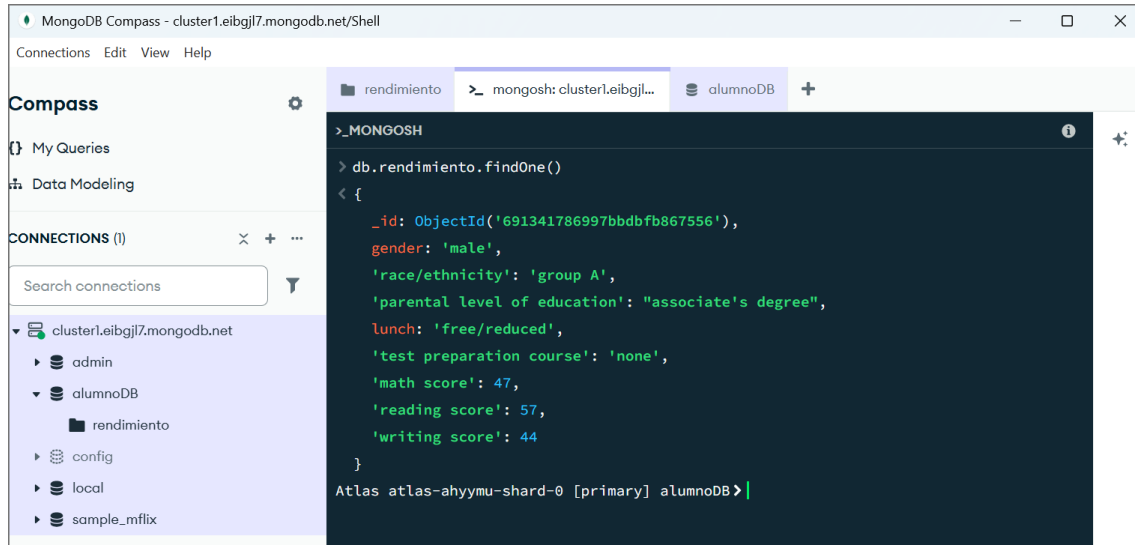
CREAR Y ELIMINAR COLECCIONES

Para crear usamos el comando ***db.createCollection*** o con ***insertOne***. Para eliminar, lo haremos con ***drop***

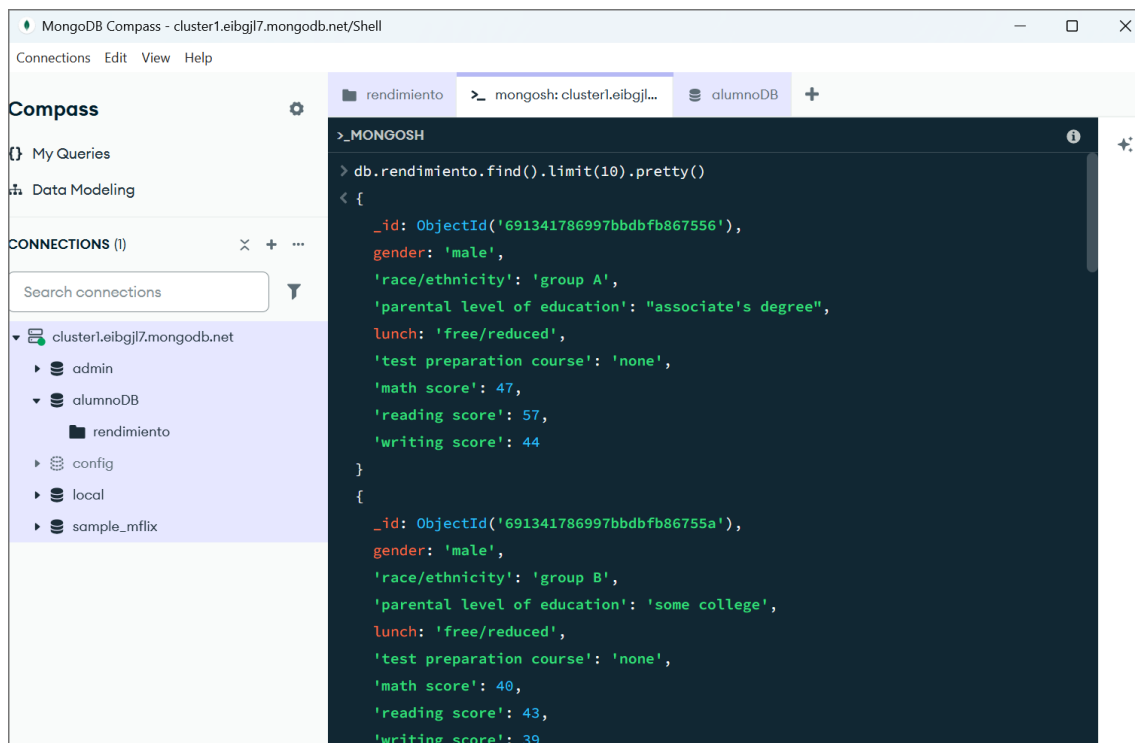


CONSULTA SIMPLE SOBRE UNA COLECCIÓN

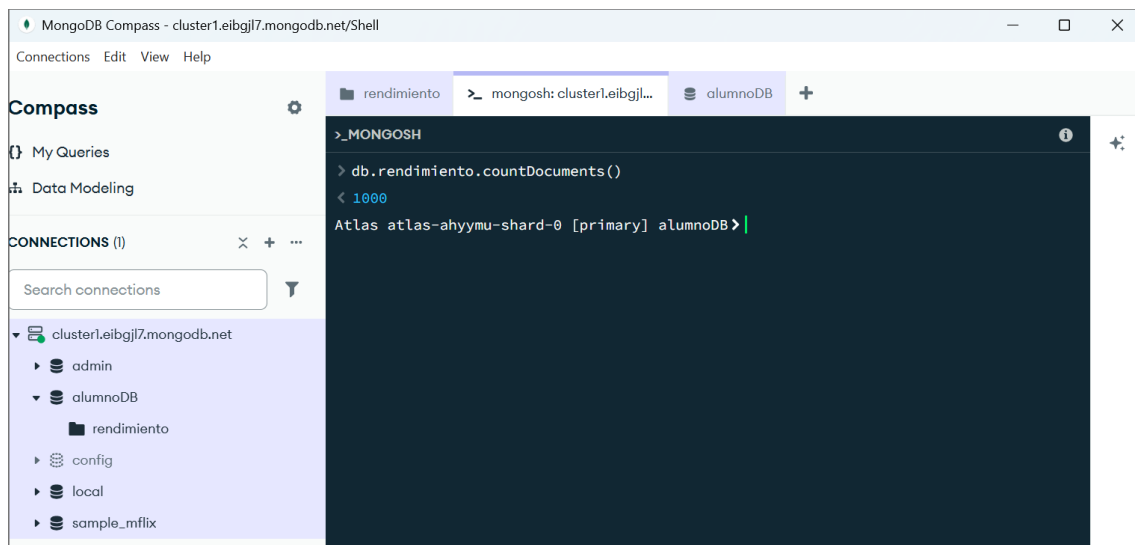
- Mostrar un documento



- Ver los primeros 10 documentos

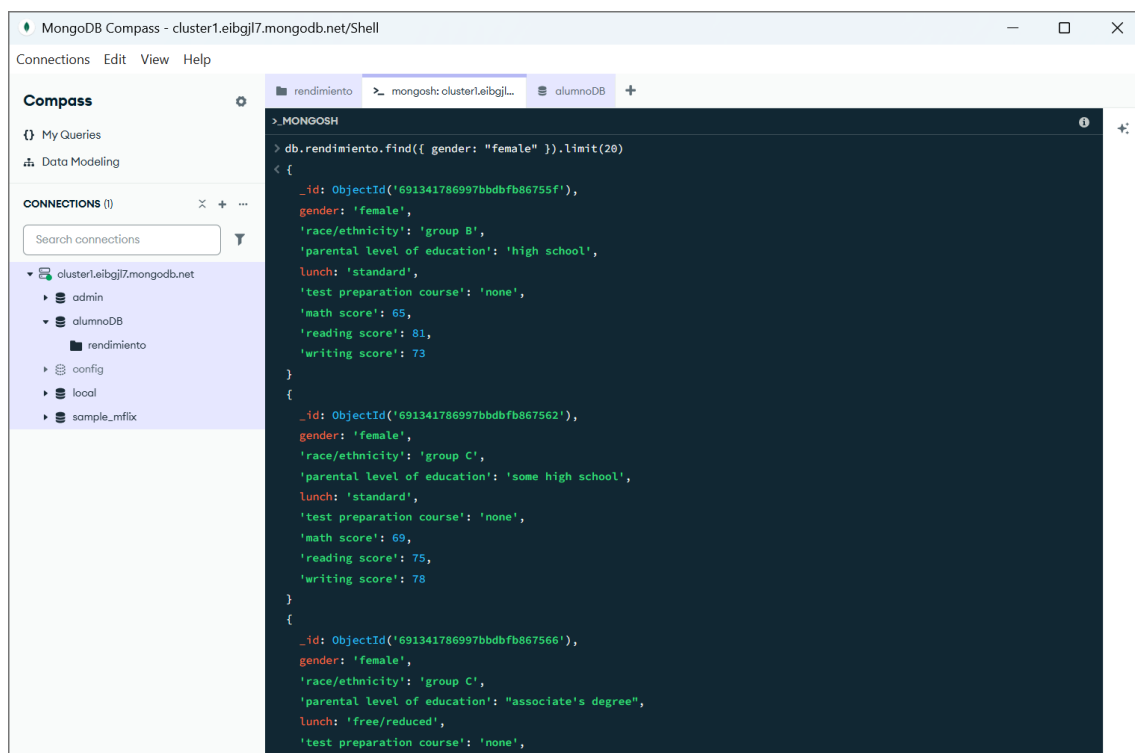


- Contar documentos

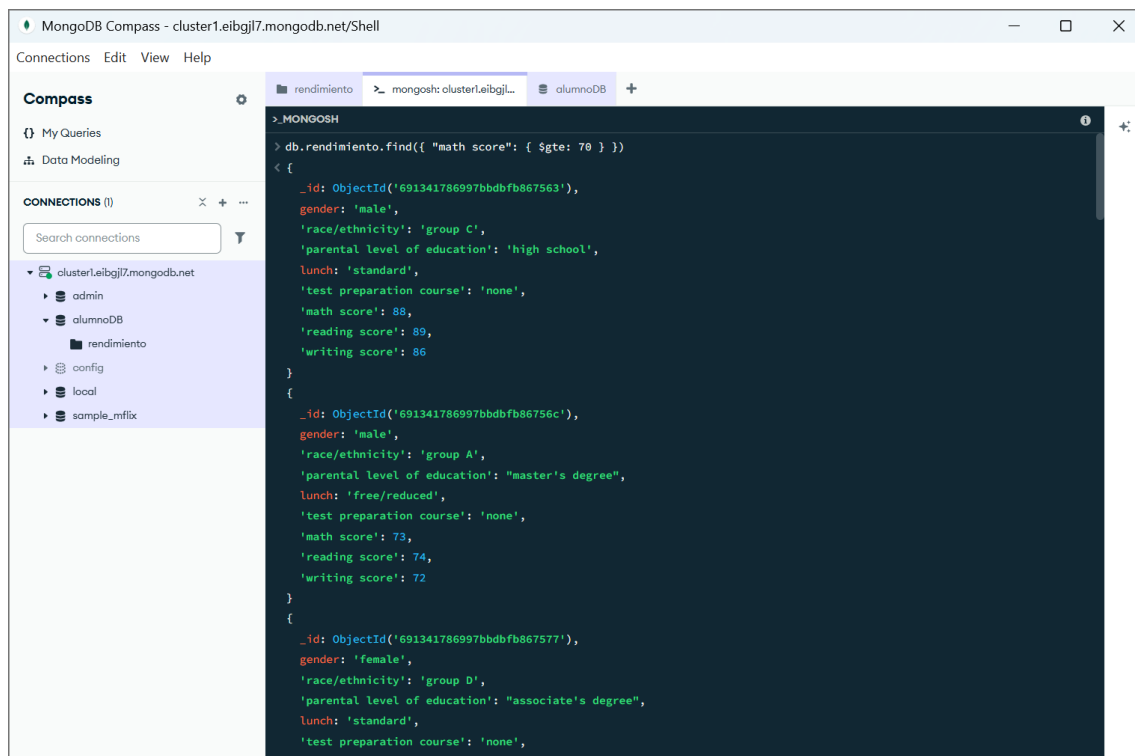


CONSULTA CON FILTROS

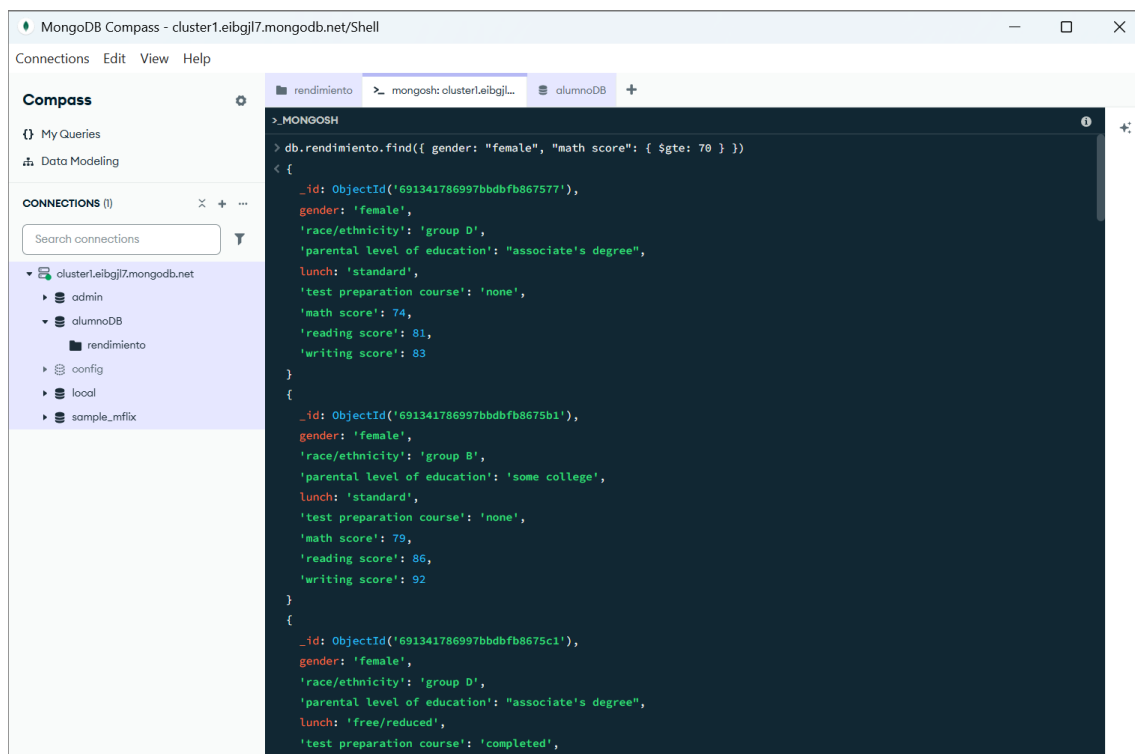
- Todos los estudiantes femeninos



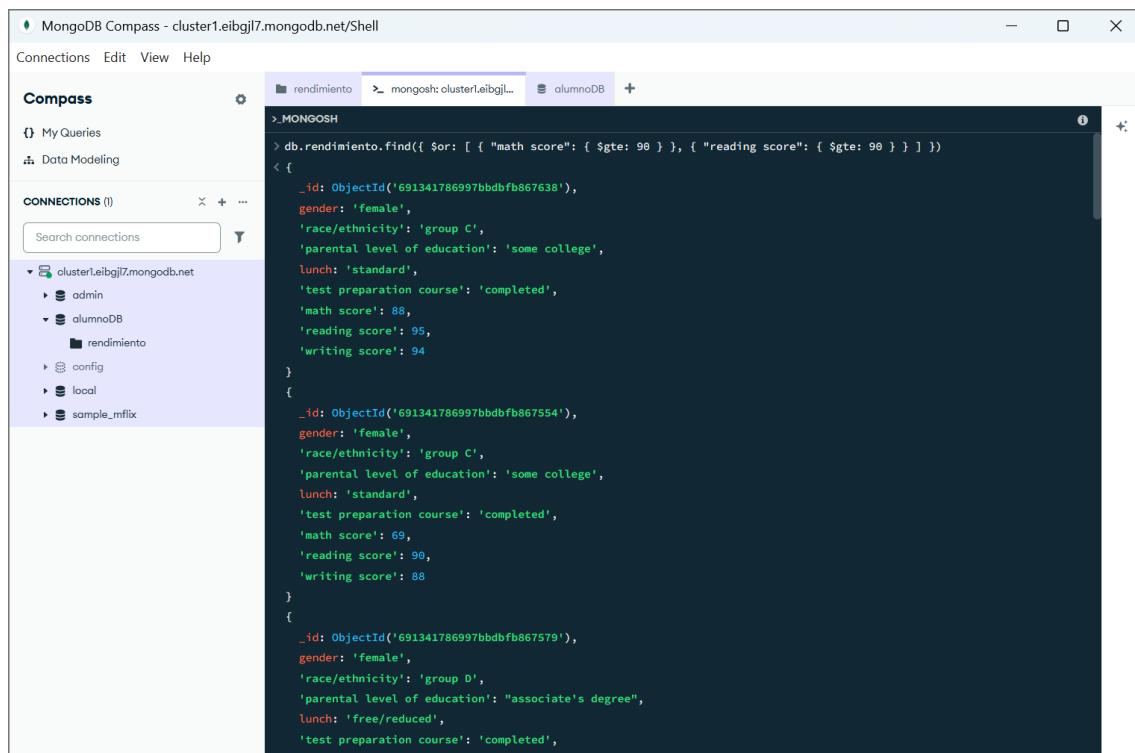
- Estudiantes con math score ≥ 70



- Varios filtros: género y math

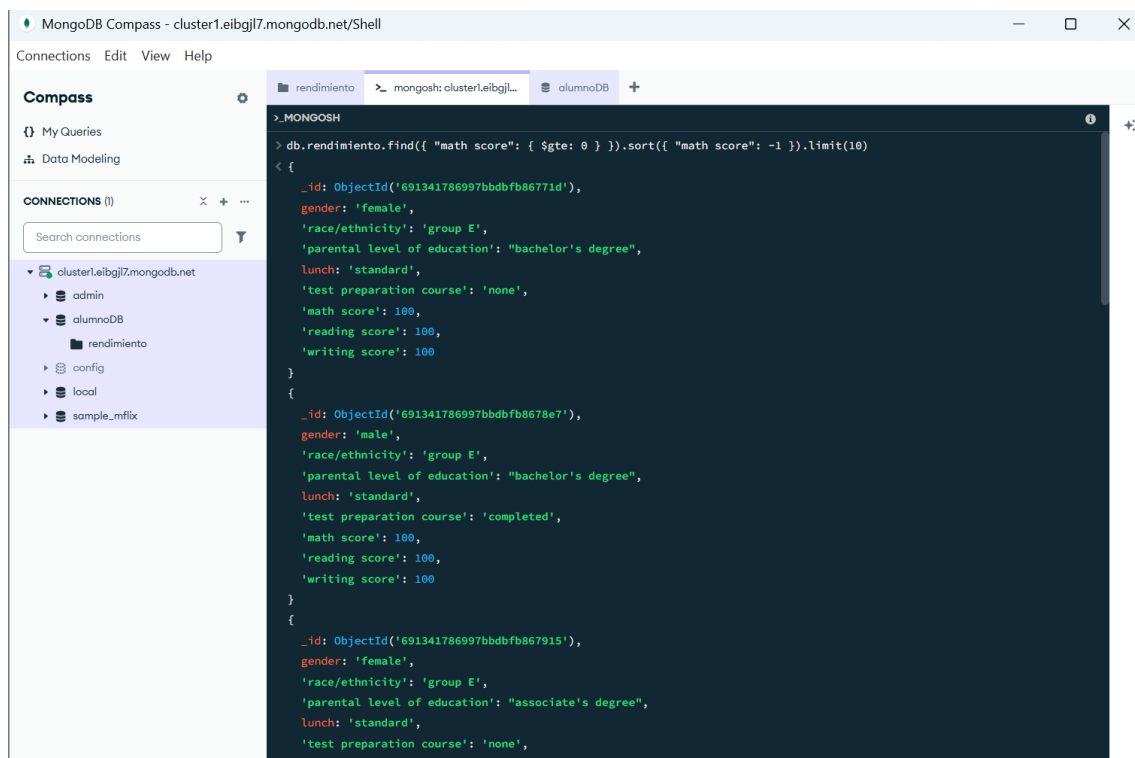


- Filtro con or



CONSULTA CON FILTROS Y ORDENACIÓN

- Orden descendente por math score



- Orden ascendente por reading score

The screenshot shows the MongoDB Compass interface. The left sidebar displays the 'Connections' panel with a search bar and a list of connections. The main panel shows the 'MongoShell' tab with a query executed in the 'rendimiento' database. The query is: `db.rendimiento.find({ "reading score": { $gte: 0 } }).sort({ "reading score": 1 }).limit(10)`. The results are displayed as a JSON array of three documents, sorted by 'reading score' in ascending order.

```
> MONGOSH
> db.rendimiento.find({ "reading score": { $gte: 0 } }).sort({ "reading score": 1 }).limit(10)
< [
  {
    "_id": ObjectId('691341786997bbdbfb86758e'),
    "gender": 'female',
    "race/ethnicity": 'group C',
    "parental level of education": 'some high school',
    "lunch": 'free/reduced',
    "test preparation course": 'none',
    "math score": 0,
    "reading score": 17,
    "writing score": 10
  },
  {
    "_id": ObjectId('691341786997bbdbfb86769a'),
    "gender": 'male',
    "race/ethnicity": 'group A',
    "parental level of education": 'some college',
    "lunch": 'free/reduced',
    "test preparation course": 'none',
    "math score": 28,
    "reading score": 23,
    "writing score": 19
  },
  {
    "_id": ObjectId('691341786997bbdbfb8677a7'),
    "gender": 'male',
    "race/ethnicity": 'group B',
    "parental level of education": 'high school',
    "lunch": 'free/reduced',
    "test preparation course": 'none',
    "math score": 28,
    "reading score": 23,
    "writing score": 19
  }
]
```

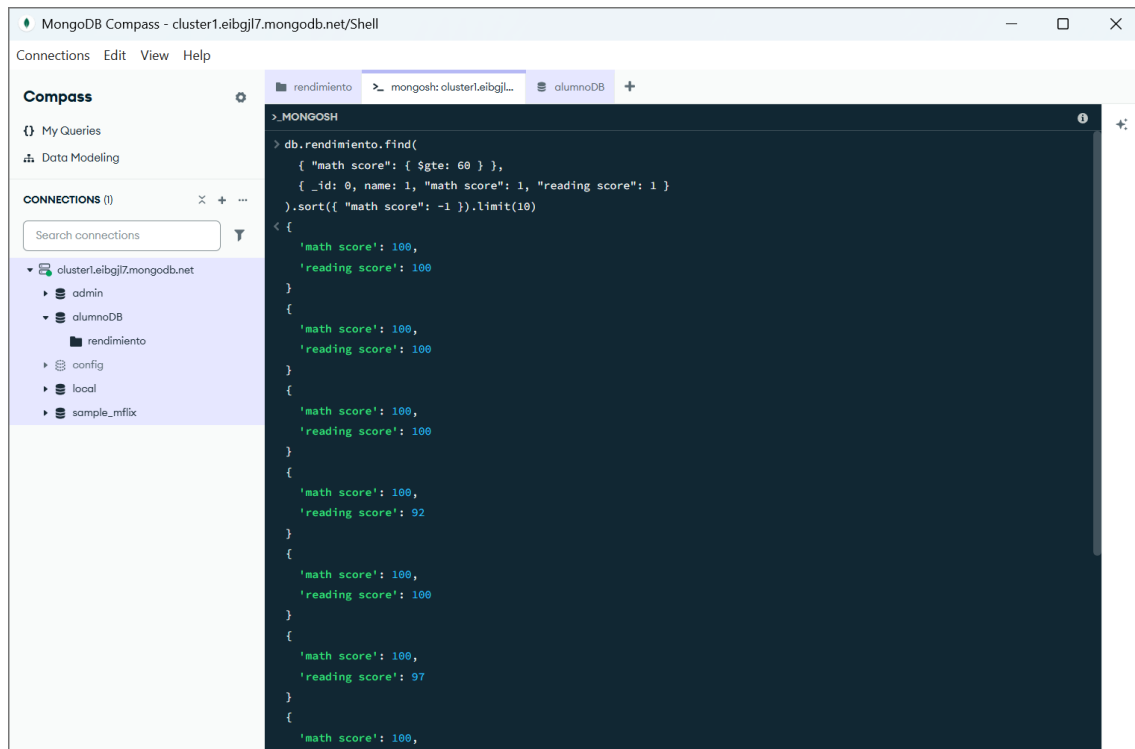
- Orden por varias claves

The screenshot shows the MongoDB Compass interface. The left sidebar displays the 'Connections' panel with a search bar and a list of connections. The main panel shows the 'MongoShell' tab with a query executed in the 'rendimiento' database. The query is: `db.rendimiento.find().sort({ gender: 1, "math score": -1 }).limit(20)`. The results are displayed as a JSON array of three documents, sorted by 'gender' in ascending order and 'math score' in descending order.

```
> db.rendimiento.find().sort({ gender: 1, "math score": -1 }).limit(20)
< [
  {
    "_id": ObjectId('691341786997bbdbfb86771d'),
    "gender": 'female',
    "race/ethnicity": 'group E',
    "parental level of education": "bachelor's degree",
    "lunch": 'standard',
    "test preparation course": 'none',
    "math score": 100,
    "reading score": 100,
    "writing score": 100
  },
  {
    "_id": ObjectId('691341786997bbdbfb867716'),
    "gender": 'female',
    "race/ethnicity": 'group E',
    "parental level of education": 'some college',
    "lunch": 'standard',
    "test preparation course": 'none',
    "math score": 100,
    "reading score": 92,
    "writing score": 97
  },
  {
    "_id": ObjectId('691341786997bbdbfb867915'),
    "gender": 'female',
    "race/ethnicity": 'group E',
    "parental level of education": "associate's degree",
    "lunch": 'standard',
    "test preparation course": 'none',
    "math score": 100,
    "reading score": 92,
    "writing score": 97
  }
]
```

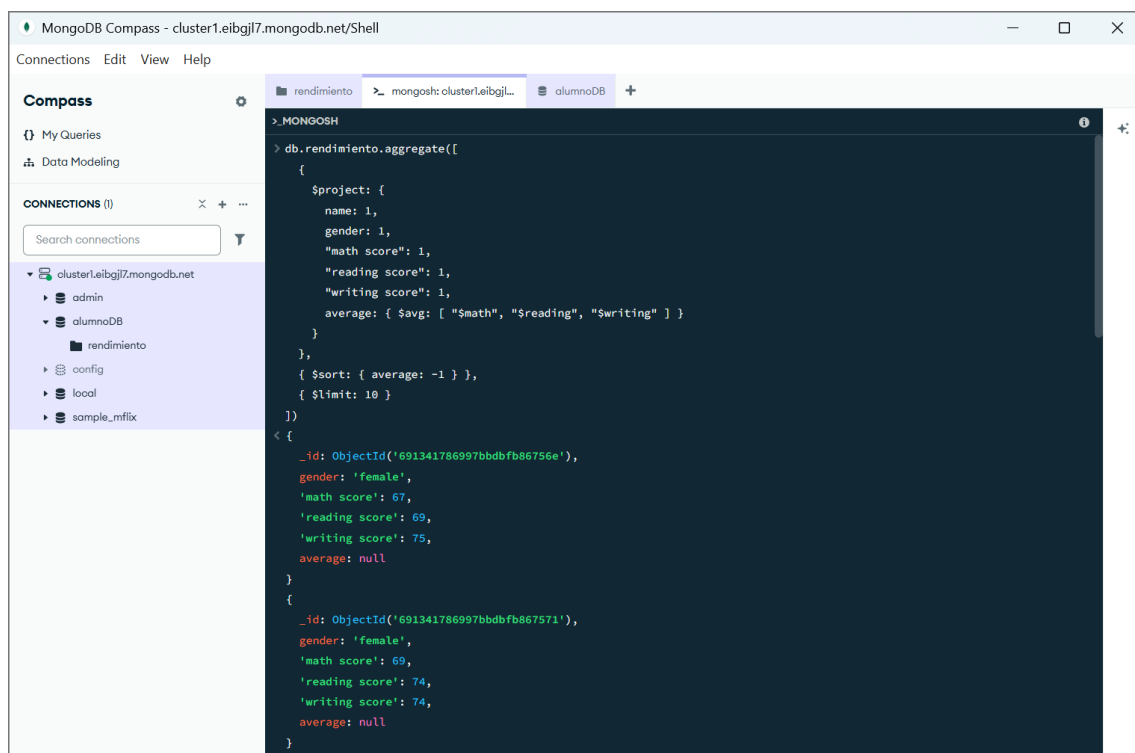
CONSULTA CON FILTROS, ORDENACIÓN Y PROYECCIÓN

- Mostrar sólo campos relevantes sin id

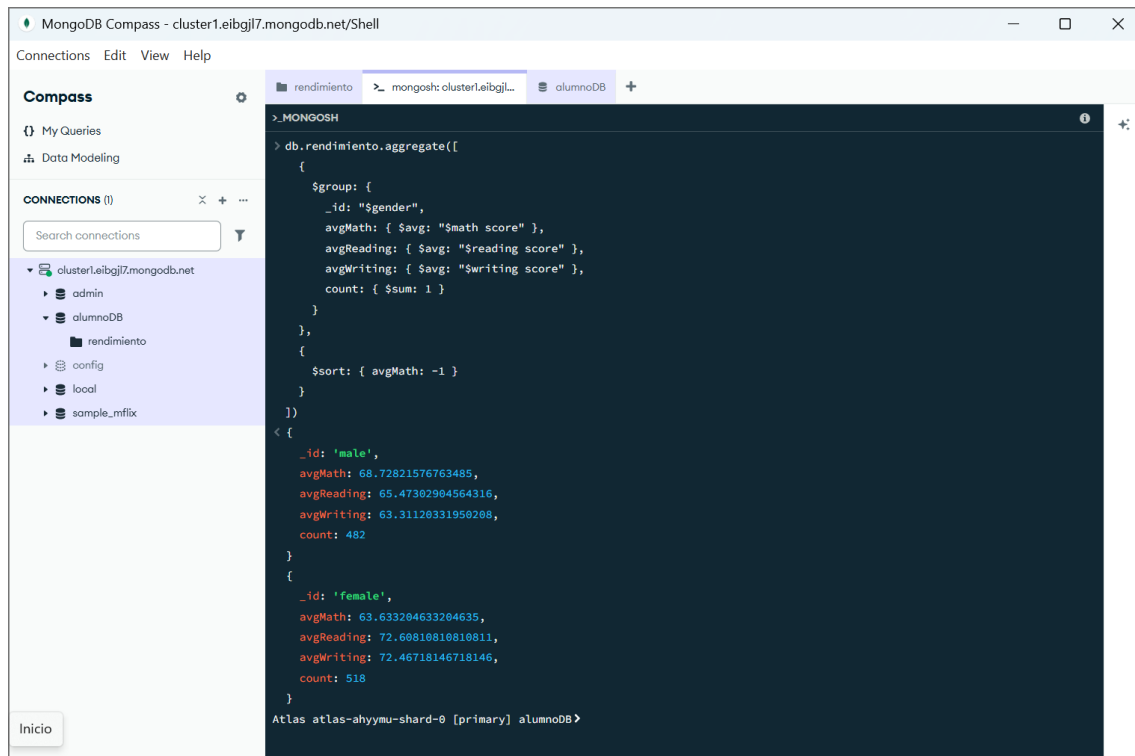


INVESTIGACIONES Y CONSULTAS DE AGREGACIÓN

- Cálculo del promedio por estudiante



- Media de puntuaciones por género



The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: cluster1.eibgij7.mongodb.net, admin, alumnoDB, and a collection named 'rendimiento'. The main window shows a MongoDB shell with the following aggregation query:

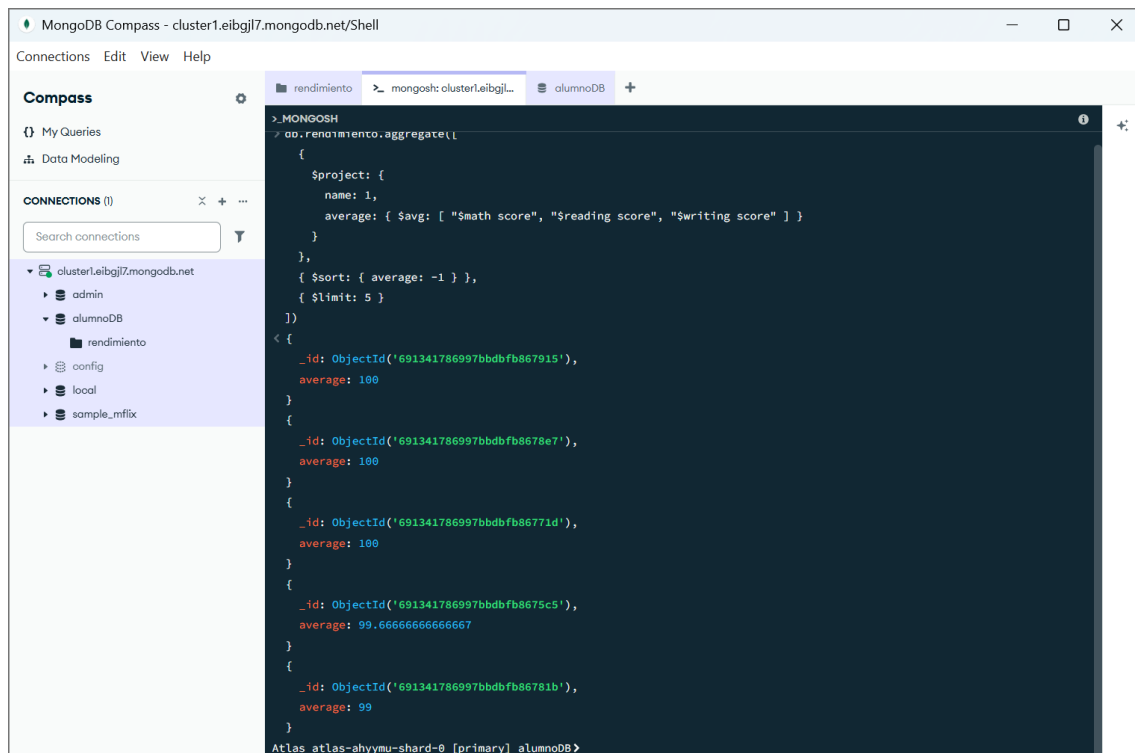
```
> db.rendimiento.aggregate([
  {
    $group: {
      _id: "$gender",
      avgMath: { $avg: "$math score" },
      avgReading: { $avg: "$reading score" },
      avgWriting: { $avg: "$writing score" },
      count: { $sum: 1 }
    }
  },
  {
    $sort: { avgMath: -1 }
  }
])
```

The query results are displayed below the shell:

```
< {
  _id: 'male',
  avgMath: 68.72821576763485,
  avgReading: 65.47302904564316,
  avgWriting: 63.31120331950208,
  count: 482
}
{
  _id: 'female',
  avgMath: 63.633204633204635,
  avgReading: 72.60810810810811,
  avgWriting: 72.46718146718146,
  count: 518
}
```

The status bar at the bottom indicates the connection is to 'Atlas atlas-ahyumu-shard-0 [primary] alumnoDB'.

- Top 5 alumnos por promedio



The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: cluster1.eibgij7.mongodb.net, admin, alumnoDB, and a collection named 'rendimiento'. The main window shows a MongoDB shell with the following aggregation query:

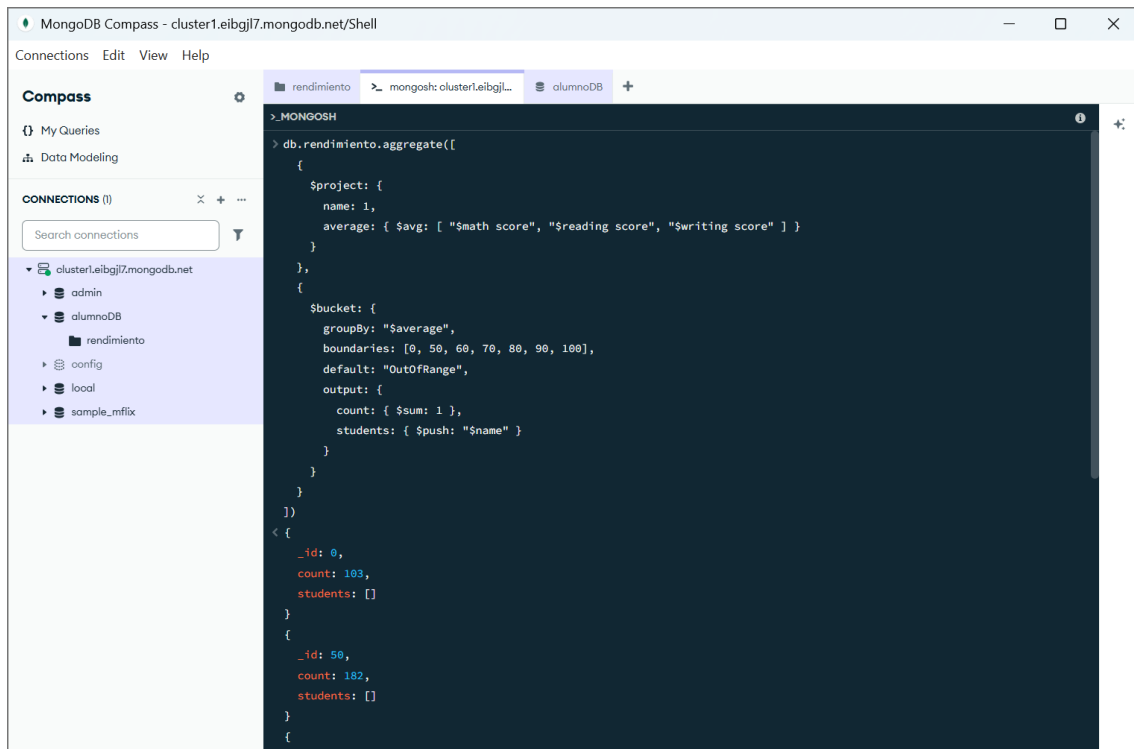
```
> db.rendimiento.aggregate([
  {
    $project: {
      name: 1,
      average: { $avg: [ "$math score", "$reading score", "$writing score" ] }
    }
  },
  {
    $sort: { average: -1 }
  },
  {
    $limit: 5
  }
])
```

The query results are displayed below the shell:

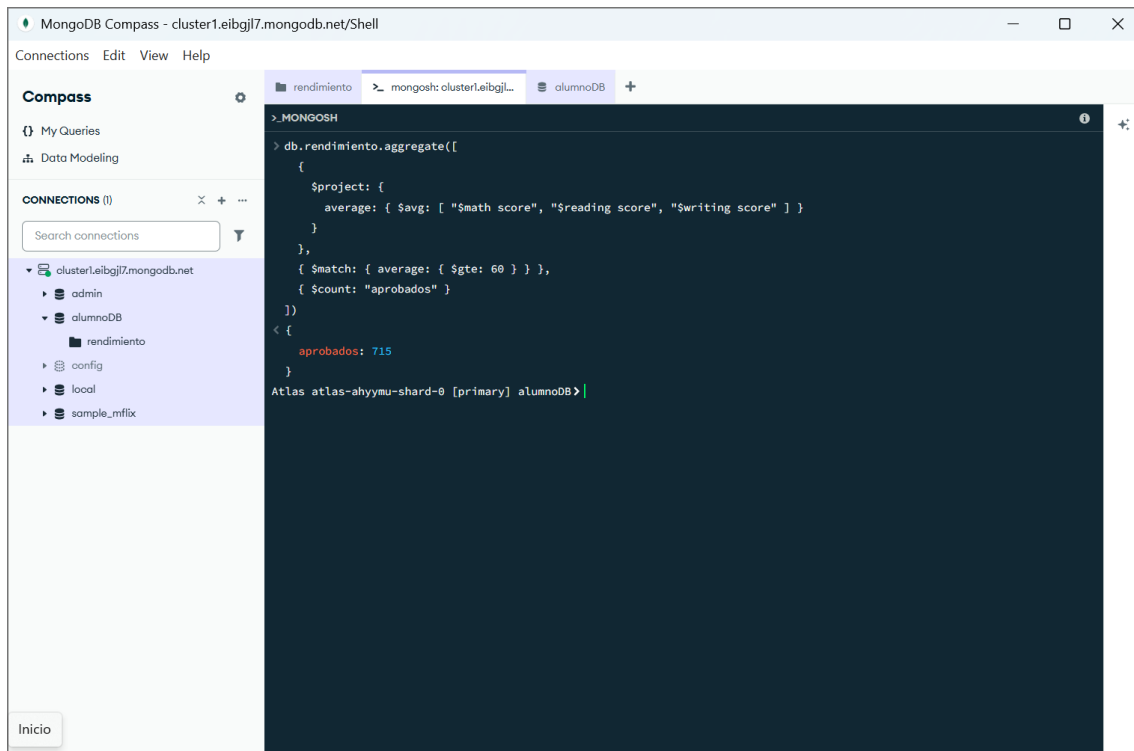
```
< {
  _id: ObjectId('691341786997bbdbfb867915'),
  average: 100
}
{
  _id: ObjectId('691341786997bbdbfb8678e7'),
  average: 100
}
{
  _id: ObjectId('691341786997bbdbfb86771d'),
  average: 100
}
{
  _id: ObjectId('691341786997bbdbfb8675c5'),
  average: 99.66666666666667
}
{
  _id: ObjectId('691341786997bbdbfb86781b'),
  average: 99
}
```

The status bar at the bottom indicates the connection is to 'Atlas atlas-ahyumu-shard-0 [primary] alumnoDB'.

- Distribución por rangos de nota media

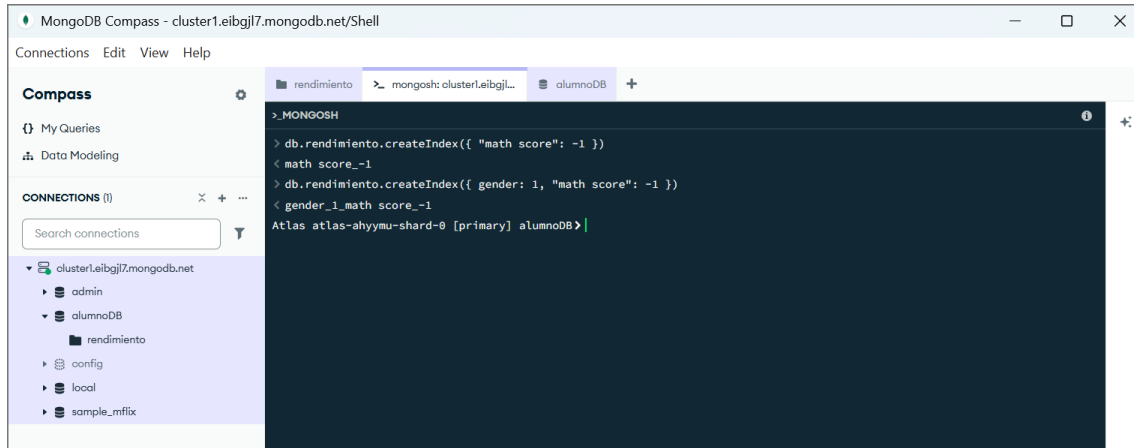


- Contar cuántos aprueban

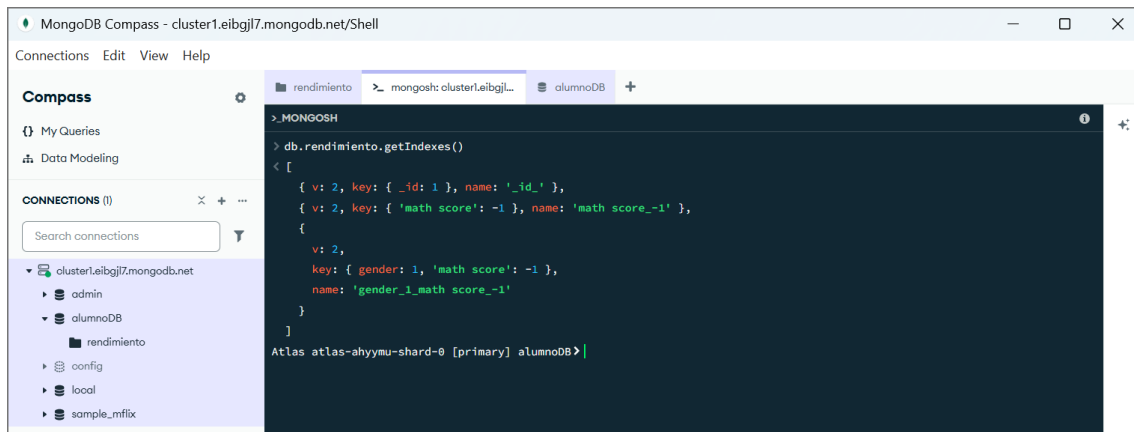


ÍNDICES PARA OPTIMIZACIÓN

- Crear índices para mejorar el rendimiento de las consultas

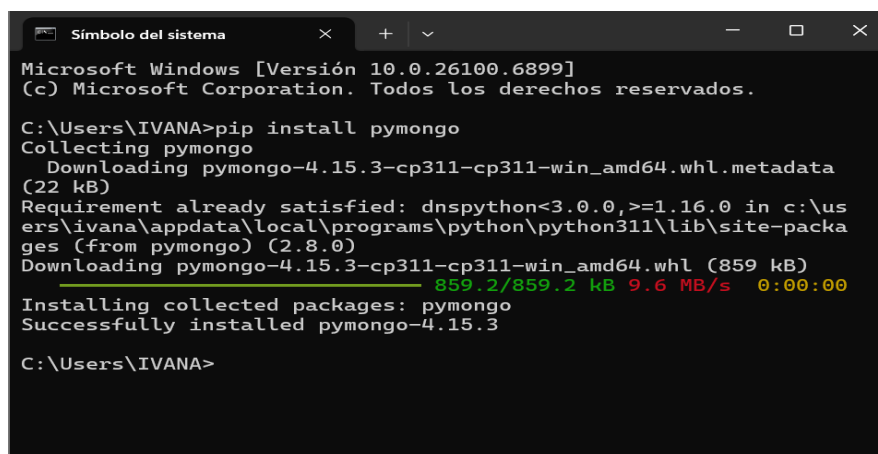


- Comprobar índices



INTEGRACIÓN CON PYTHON (PYMONGO + PANDAS)

Para scripts y análisis en Python instalamos la librería pymongo



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.26100.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\IVANA>pip install pymongo pandas
Requirement already satisfied: pymongo in c:\users\ivana\appdata\local\programs\python\python311\lib\site-packages (4.15.3)
Requirement already satisfied: pandas in c:\users\ivana\appdata\local\programs\python\python311\lib\site-packages (2.3.3)
Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in c:\users\ivana\appdata\local\programs\python\python311\lib\site-packages (from pymongo) (2.8.0)
Requirement already satisfied: numpy>=1.23.2 in c:\users\ivana\appdata\local\programs\python\python311\lib\site-packages (from pandas) (2.2.6)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\ivana\appdata\roaming\python\python311\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\ivana\appdata\local\programs\python\python311\lib\site-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in c:\users\ivana\appdata\local\programs\python\python311\lib\site-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in c:\users\ivana\appdata\roaming\python\python311\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)

C:\Users\IVANA>
```

CÓDIGO Y RESULTADOS DEL SCRIPT DE PYTHON

```
# Título: Ejemplos de MongoDB
# Autor: Ivana Sánchez Pérez

# colores
azul = '\033[94m'
rosa = '\033[38;5;200m'
turquesa = '\033[38;5;44m'
reset = '\033[0m'

print(f"\n{rosa}===== Ejemplos de MongoDB ====={reset}")
print()

from pymongo import MongoClient
import pandas as pd

# Conexión
uri = "mongodb+srv://isanper504a:Peluchingordi111%40@cluster1.eibgjl7.mongodb.net/alumnoDB"
client = MongoClient(uri)
db = client["alumnoDB"]
col = db["rendimiento"]

# 1) Ver algunos documentos (sin mostrar el _id)
docs = col.find({}, {"_id": 0}).limit(5)
df = pd.DataFrame(list(docs))
print(f"{azul}Primeros registros:{reset}")
print(f"{turquesa}{df}{reset}")

# 2) Agregación: promedio por género
pipeline = [
    {
        "$group": {
            "_id": "$gender",
            "avgMath": {"$avg": "$math score"},
            "avgReading": {"$avg": "$reading score"},
        }
    }
]
```

```

        "avgWriting": {"$avg": "$writing score"},
        "count": {"$sum": 1}
    }
},
{"$sort": {"avgMath": -1}}
]
result = list(col.aggregate(pipeline))
print(f"\n{azul}\nPromedio de puntuaciones por género:{reset}")
for r in result:
    print(f"{turquesa}{r}{reset}")

# 3) Consulta: top 10 por math
top_math = list(col.find({}, {"_id":0, "name":1, "math score":1}).sort("mathscore", -1).limit(10))
print(f"\n{azul}Top math:{reset}", top_math)

# 4)Consulta con filtro y ordenación
top_math = list(col.find({}, {"_id": 0, "gender": 1, "math score": 1})
                .sort("math score", -1)
                .limit(5))
print(f"\n{azul}Top 5 alumnos con mejor nota en matemáticas: {reset}")
print(pd.DataFrame(top_math))

# 5) Estudiantes con math score >= 80
filtro = {"math score": {"$gte": 80}}
resultados = list(col.find(filtro, {"_id": 0, "gender": 1, "math score": 1}))
print(f"\n{azul}\nEstudiantes con math score >= 80:{reset}")
print(pd.DataFrame(resultados))

pipeline = [
    {
        "$group": {
            "_id": "$test preparation course",
            "avgMath": {"$avg": "$math score"},
            "avgReading": {"$avg": "$reading score"},
            "avgWriting": {"$avg": "$writing score"},
            "total": {"$sum": 1}
        }
    },
    {"$sort": {"avgMath": -1}}
]

result = list(col.aggregate(pipeline))
print(f"\n{azul}Promedio según haber hecho curso de preparación o no:{reset}")
for r in result:
    print(f"{turquesa}{r}{reset}")

```


RESULTADO DE LA EJECUCIÓN

```
Símbolo del sistema
C:\Users\IVANA\Desktop>python ejemplo_python.py

===== Ejemplos de MongoDB =====

Primeros registros:
gender race/ethnicity parental level of education lunch test preparation course math score reading score writing score
0 male group A associate's degree free/reduced none 47 57 48
1 male group B some college free/reduced none 45 43 39
2 female group B high school standard none 65 81 73
3 female group C some high school standard none 69 75 78
4 male group C high school standard none 88 89 86

Promedio de puntuaciones por género:
['_id': 'male', 'avgMath': 68.72821576763485, 'avgReading': 65.47392904564316, 'avgWriting': 63.31128331960288, 'count': 482]
['_id': 'female', 'avgMath': 63.633204633204635, 'avgReading': 72.60810810810811, 'avgWriting': 72.46718146718146, 'count': 518]

Top math: [{'math score': 67}, {'math score': 69}, {'math score': 88}, {'math score': 69}, {'math score': 50}, {'math score': 69}, {'math score': 48}, {'math score': 47}, {'math score': 65}, {'math score': 73}]

Top 5 alumnos con mejor nota en matemáticas:
gender math score
0 female 100
1 male 100
2 female 100
3 male 100
4 male 100

Estudiantes con math score >= 80:
gender math score
0 female 100
1 male 100
2 female 100
3 male 100
4 male 100
...
188 female 80
189 male 80
190 male 80
191 male 80
192 male 80

[193 rows x 2 columns]

Promedio según haber hecho curso de preparación o no:
['_id': 'completed', 'avgMath': 69.65853872525699, 'avgReading': 73.89385474866335, 'avgWriting': 74.41899441348782, 'total': 358]
['_id': 'none', 'avgMath': 64.8778816199377, 'avgReading': 66.53426791277259, 'avgWriting': 64.50467289719626, 'total': 642]

C:\Users\IVANA\Desktop>
```