**Texto

Descripción generada automáticamente**

**INGENIERÍA EN SISTEMAS DE INFORMACIÓN**

**MATERIA:** SISTEMAS DE GESTIÓN DE BASES DE DATOS

**NIVEL:** 3

**CICLO LECTIVO:** 2024 – 2DO. CUATRIMESTRE

**DOCENTES:**

PROFESOR: I.S.I. ANDRÉS PABLO FANTÍN

J.T.P.: I.S.I. JUAN CARLOS FERNÁNDEZ

AUXILIAR ADSCRIPTA: LUCIANA CAMPESTRINI

**Grupo n.º2:**

* Arduña Zago, Agustín Juan Luis
* Berón de Astrada, Santiago Agustín
* Fernández, Facundo Nahuel
* Kinweiler, Víctor Fabricio
* Schefer, Mauricio Nicolás
* Velazco Gez Schegtel, Juan Ignacio

# Instalación de MongoDB

Siguiendo el manual de MongoDB, lo instalamos:

Texto

Descripción generada automáticamente



Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Ahora que tenemos instalado MongoDB podemos hacer las consultas.

# Realizar por terminal de comandos los siguientes ejercicios

1)Insertar los documentos de películas provistos en los archivos customers.json, films.json y stores.json en la base de datos “Mongodb2024”, en las colecciones “clientes”, “peliculas” y “tiendas” respectivamente. Utilizar la herramienta mongoimport.

Consulta:

mongoimport -d=Mongodb2024 -c=clientes customers.json

mongoimport -d=Mongodb2024 -c=tiendas stores.json

mongoimport -d=Mongodb2024 -c=peliculas films.json

Imagen en blanco y negro

Descripción generada automáticamente con confianza media

Podemos visualizar que la base de datos fue creada correctamente.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

2. Listar los títulos (Title) de todas las películas.

Consulta:

db.peliculas.find({},{Title:1, \_id:0})

Imagen que contiene circuito

Descripción generada automáticamente

3. Listar los nombres y apellidos (First name, Last name) de los actores de la película "AGENT TRUMAN".

Consulta:

db.peliculas.find({ Title: "AGENT TRUMAN" },

{ "Actors.First name": 1, "Actors.Last name": 1, \_id: 0 })

Texto

Descripción generada automáticamente

4.Listar todos los campos de la película "AGENT TRUMAN", excepto los actores.

Consulta:

db.peliculas.find({Title:"AGENT TRUMAN"},{Actors:0})

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

5. Listar los títulos (Title) y las categorías (Category) de todas las películas, ordenados por categoría en forma descendente

Consulta:

db.peliculas.find({},{ Title: 1, Category: 1, \_id: 0 }) .sort({ Category: -1 })



6. Listar el título (Title) y los nombres y apellidos (First name, Last name) de los actores de todas las películas de terror (Category : “Horror”).

Consulta:

db.peliculas.find({Category:"Horror"},

{Title:1,"Actors.First ":1,"Actors.Last name":1,\_id:0})

Interfaz de usuario gráfica

Descripción generada automáticamente

7. Listar el título (Title) y los nombres y apellidos (First name, Last name) de los actores de las películas que tengan exactamente cuatro actores.

Consulta:

db.peliculas.find({"Actors":{$size:4}},

{Title:1,"Actors.First name":1,"Actors.Last name":1,\_id:0})

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

8. Mostrar el título (Title) y los nombres y apellidos (First name, Last name) de los actores de una película que tenga exactamente cuatro actores.

Consulta:

db.peliculas.findOne({"Actors":{$size:4}},

{Title:1,"Actors.First name":1,"Actirs.Last name":1,\_id:0})

Texto

Descripción generada automáticamente

9.Listar los títulos (Title) y la descripción (Description) de las películas que en su descripción tengan la palabra “Panorama”.

Consulta:

db.peliculas.find({$where:"this.Description && this.Description.includes('Panorama')"},{Title:1,Description:1,\_id:0})

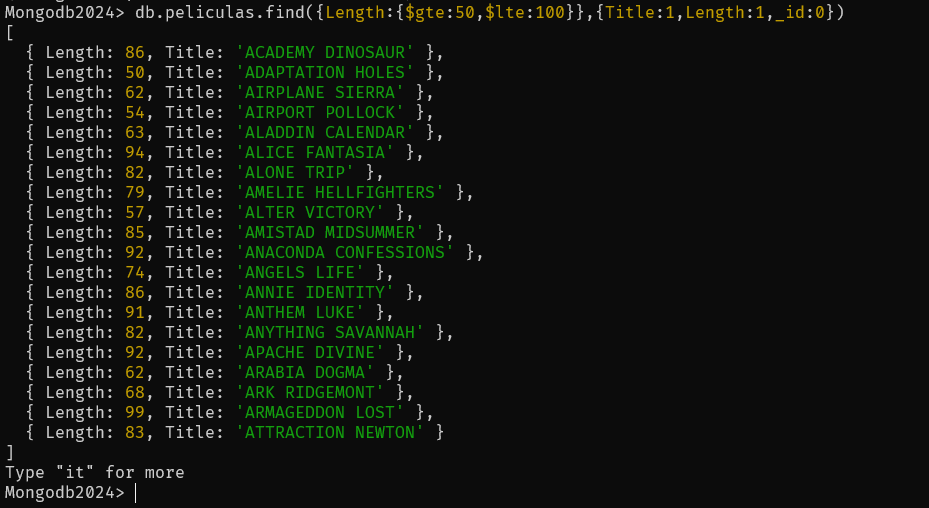
Texto

Descripción generada automáticamente

10. Encontrar las películas que tengan una duración (Length) entre 50 y 100 minutos.

Consulta:

db.peliculas.find({Length:{$gte:50,$lte:100}},{Title:1,Length:1,\_id:0})



11. Mostrar título (Title), descripción (Description), duración (Length) y características especiales (Special Features) de las películas que tengan “detrás de escena” (“Behind the Scenes”) entre sus características especiales.

Consulta:

db.peliculas.find({ "Special Features": "Behind the Scenes" }, { Title: 1, Description: 1, Length: 1, "Special Features": 1, \_id: 0 })

Texto

Descripción generada automáticamente

12. Mostrar título (Title), descripción (Description), duración (Length) y características especiales (Special Features) de las películas que tengan un costo de reposición (Replacement Cost) mayor a 20.

Consulta:

db.peliculas.find({"Replacement Cost": {$gt:20}},{Title:1,Description:1,"Special Features":1,\_id:0,"Replacement Cost":1})

Texto

Descripción generada automáticamente

13. Mostrar los datos de las películas (menos sus actores) que no son comedias (Category : “Comedy”)

Consulta:

db.peliculas.find({Category:{$ne:"Comedy"}},{Actors:0})

Texto

Descripción generada automáticamente

14. Mostrar la cantidad de películas de terror (Category : “Horror”).

Consulta:

db.peliculas.countDocuments({Category:"Horror"})



15. Listar las películas en las que participen actores de apellido (Last name) “KILMER”.

Consulta:

db.peliculas.find({ "Actors.Last name": "KILMER" },

Imagen que contiene circuito

Descripción generada automáticamente { Title: 1, \_id: 0 })

16. Mostrar la cantidad de películas en las que actuó “OPRAH KILMER”.

Consulta:

db.peliculas.countDocuments(

 { "Actors.First name": "OPRAH", "Actors.Last name": "KILMER" });

17. Mostrar las películas en las que actúan juntos “OPRAH KILMER” y “MENA HOPPER”.

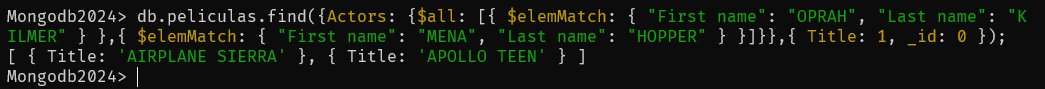
Consulta:

db.peliculas.find({Actors: {$all: [

{ $elemMatch: { "First name": "OPRAH", "Last name": "KILMER" } },

{ $elemMatch: { "First name": "MENA", "Last name": "HOPPER" } }]}

},{ Title: 1, \_id: 0 });



18. Cargue los datos de cada unx de lxs integrantes del grupo como clientes.

Consulta:

db.clientes.insertMany([

{\_id: ObjectId(),

Address: "CALLE 222",

City: "Resistencia",

Country: "Argentina",

District: "Centro",

"First Name": "MAURICIO",

"Last Name": "SCHEFER",

Phone: "3624111111",

Rentals: []},

{\_id: ObjectId(),

Address: "CALLE 223",

City: "Resistencia",

Country: "Argentina",

District: "Norte",

"First Name": "AGUSTIN",

"Last Name": "ARDUÑA",

Phone: "3624111112",

Rentals: []},

{\_id: ObjectId(),

Address: "CALLE 224",

City: "Resistencia",

Country: "Argentina",

District: "Sur",

"First Name": "FACUNDO",

"Last Name": "FERNANDEZ",

Phone: "3624111113",

Rentals: []},

{\_id: ObjectId(),

Address: "CALLE 224",

City: "Resistencia",

Country: "Argentina",

District: "Sur",

"First Name": "JUAN IGNACIO",

"Last Name": "VELAZCO",

Phone: "3624111114",

Rentals: []},

{\_id: ObjectId(),

Address: "CALLE 225",

City: "Resistencia",

Country: "Argentina",

District: "Norte",

"First Name": "SANTIAGO",

"Last Name": "BERON",

Phone: "3624111115",

Rentals: []},

{ \_id: ObjectId(),

Address: "CALLE 226",

City: "Resistencia",

Country: "Argentina",

District: "Este",

"First Name": "VICTOR FABRICIO",

"Last Name": "KINWEILER",

Phone: "3624111116",

Rentals: []}

]);

Texto

Descripción generada automáticamente

19. Agregue el alquiler de dos películas a cada unx de lxs clientes cargadxs en el punto anterior.

Consulta:

const clientesNuevos = [

ObjectId('67243d2099a2eaae07c1c18c'),

ObjectId('67243d2099a2eaae07c1c18d'),

ObjectId('67243d2099a2eaae07c1c18e'),

ObjectId('67243d2099a2eaae07c1c18f'),

ObjectId('67243d2099a2eaae07c1c190'),

ObjectId('67243d2099a2eaae07c1c191')

];

const peliculasRentadas = [

{

"Film Title": " ALTER VICTORY",

"Rental Date": new Date(),

"filmId": 18

},

{

"Film Title": " AMADEUS HOLY",

"Rental Date": new Date(),

"filmId": 19

}

];

db.clientes.updateMany(

{ \_id: { $in: clientesNuevos } },

{ $push: { Rentals: { $each: peliculasRentadas } } }

);

Texto

Descripción generada automáticamente

20. Incrementar el costo de reposición (Replacement Cost) en 5 unidades para todas las películas. Verificar que se realizaron los cambios.

Consulta:

db.peliculas.updateMany({}, { $inc: { "Replacement Cost": 5 } })

Texto

Descripción generada automáticamente

21. Actualizar los documentos de las películas cuyo costo de reposición (Replacement Cost) es mayor a 24, agregando un campo “Pelicula cara” con valor true.

Primero vemos las películas que tienen un Replacement Cost >= 24

Texto

Descripción generada automáticamente

Ahora le agregamos “película cara”:

Texto

Descripción generada automáticamente

Consulta :

db.peliculas.updateMany({"Replacement Cost":{$gt:24}},{$set:{"Película cara":true}})

Verificamos si se actualizo:

Texto

Descripción generada automáticamente

Podemos ver que se agrego el campo correctamente

22. Agregar el actor “Bruce Lee” a la película "ALASKA PHANTOM".

Consulta:

Primero tenemos que buscar el ultimo actor id y sumarle uno

const ultimoId= db.peliculas.aggregate([{$unwind: "$Actors"},{$group:{\_id:null, ultimoId:{$max: "$Actors.actorId"}}}]).toArray()[0].ultimoId +1;

Ahora tenemos que agregar el nuevo actor

db.peliculas.updateOne({Title: "ALASKA PHANTOM" },{ $addToSet: { Actors: { "First name": "Bruce", "Last name": "Lee", actorId: ultimoId} } });

Texto

Descripción generada automáticamente

Volvemos a buscar la película para ver si se cargó el actor.

Texto

Descripción generada automáticamente

23. Borrar la película "ALAMO VIDEOTAPE".

Buscamos la película:

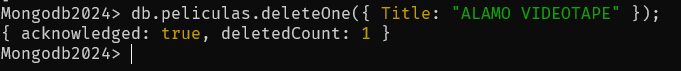
Texto

Descripción generada automáticamente

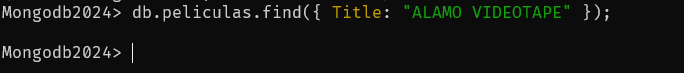
Borramos la película:

Consulta:

db.peliculas.deleteOne({ Title: "ALAMO VIDEOTAPE" });



Como podemos ver queremos buscar nuevamente la película y no devuelve nada:



24. Obtener el promedio de duración de las películas por categoría.

Consulta:

db.peliculas.aggregate([{$group:{\_id: "$Category", duracionProm: { $avg: "$Length" }}},{$project:{\_id:0,Category: "$\_id", duracionProm: 1}}])

Texto

Descripción generada automáticamente