	INGENIERÍA DE SOFTWARE
	GUÍA 2
	MONGODB – Operaciones y CRUD

Operaciones

1.- Para ejecutar comandos mongosh, primero debe conectarse a una implementación de MongoDB.

Abrir un terminal y ejecutar:

mongosh

2.- Cambiar bases de datos

Para mostrar la base de datos que está utilizando, escriba:

db

La operación debería devolver test, que es la base de datos predeterminada.

3.- Para cambiar de base de datos, emita el use <db> asistente, como en el siguiente ejemplo:

use dbi12

```
test> db
test
test> use dbi12
switched to db dbi12
dbi12> db
dbi12
dbi12> _
```

```
dbi12> use test
switched to db test
test> use dbi12
switched to db dbi12
dbi12>
```


4.- Crear una nueva base de datos y colección

Para crear una nueva base de datos, emita el use <db> comando con la base de datos que le gustaría crear. Por ejemplo, los siguientes comandos crean tanto la base de datos dbubicaciones la colección coordenadas utilice la operación insertOne()

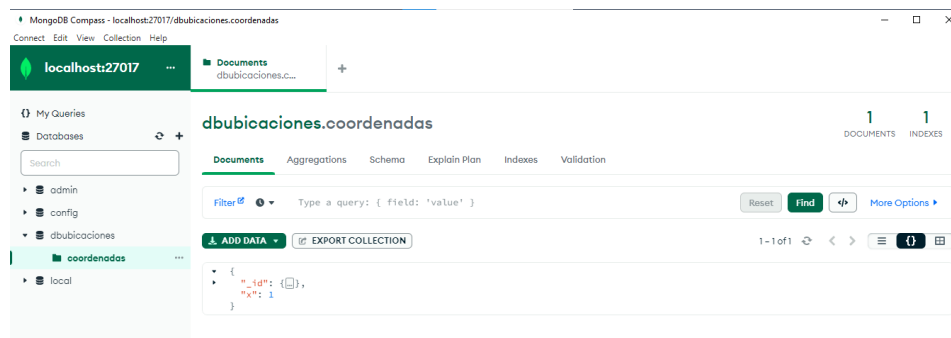
```
test> use dbubicaciones
switched to db dbubicaciones
dbubicaciones>
```

```
test> use dbubicaciones
switched to db dbubicaciones
dbubicaciones> db.coordenadas.insertOne({x:1});
{
  acknowledged: true,
  insertedId: ObjectId("641b637c88bc068804ed246d")
}
dbubicaciones>
```

Si una colección no existe, MongoDB crea la colección cuando almacena datos por primera vez para esa colección.

	INGENIERÍA DE SOFTWARE
	GUÍA 2
	MONGODB – Operaciones y CRUD

5.- Observar lo realizado desde Compass.



Terminar un comando en ejecución

Para finalizar un comando en ejecución o una consulta mongosh, presione Ctrl + C.

Borrar la consola mongosh

El comando cls borra la consola. También puede borrar la consola con Ctrl + L y console.clear().

Realizar operaciones CRUD

Las operaciones CRUD crean, leen, actualizan y eliminan documentos.

Crear

Las operaciones de creación o inserción agregan nuevos documentos a una colección. Si la colección no existe, las operaciones de creación también crean la colección.

Puede insertar un solo documento o varios documentos en una sola operación.

El shell de MongoDB proporciona los siguientes métodos para insertar documentos en una colección:

Para insertar un solo documento, utilice:


`db.collection.insertOne()`

Para insertar varios documentos, utilice:

`db.collection.insertMany()`

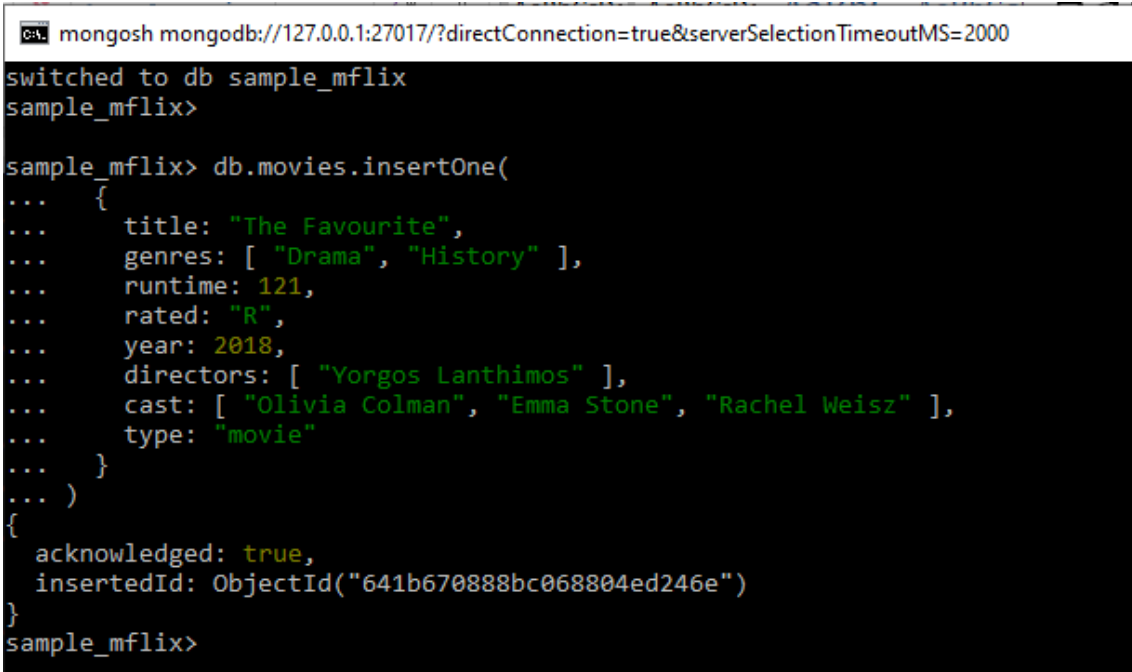
- Insertar un solo documento

`db.collection.insertOne()` inserta un solo documento en una colección. Si el documento no especifica un campo `_id`, MongoDB agrega el campo `_id` con un valor `ObjectId` al nuevo documento.

	INGENIERÍA DE SOFTWARE
	GUÍA 2
	MONGODB – Operaciones y CRUD

```
use sample_mflix

db.movies.insertOne(
  {
    title: "The Favourite",
    genres: [ "Drama", "History" ],
    runtime: 121,
    rated: "R",
    year: 2018,
    directors: [ "Yorgos Lanthimos" ],
    cast: [ "Olivia Colman", "Emma Stone", "Rachel Weisz" ],
    type: "movie"
  }
)
```




```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
switched to db sample_mflix
sample_mflix>
sample_mflix> db.movies.insertOne(
...   {
...     title: "The Favourite",
...     genres: [ "Drama", "History" ],
...     runtime: 121,
...     rated: "R",
...     year: 2018,
...     directors: [ "Yorgos Lanthimos" ],
...     cast: [ "Olivia Colman", "Emma Stone", "Rachel Weisz" ],
...     type: "movie"
...   }
... )
{
  acknowledged: true,
  insertedId: ObjectId("641b670888bc068804ed246e")
}
sample_mflix>
```

insertOne() devuelve un documento que incluye el valor del campo `_id` del documento recién insertado.

Para recuperar el documento insertado, lea la colección:

```
db.movies.find( { title: "The Favourite" } )
```

	INGENIERÍA DE SOFTWARE
	GUÍA 2
	MONGODB – Operaciones y CRUD

CA. mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
... )
{
  acknowledged: true,
  insertedId: ObjectId("641b670888bc068804ed246e")
}
sample_mflix> db.movies.find( { title: "The Favourite" } )
[
  {
    _id: ObjectId("641b670888bc068804ed246e"),
    title: 'The Favourite',
    genres: [ 'Drama', 'History' ],
    runtime: 121,
    rated: 'R',
    year: 2018,
    directors: [ 'Yorgos Lanthimos' ],
    cast: [ 'Olivia Colman', 'Emma Stone', 'Rachel Weisz' ],
    type: 'movie'
  }
]
sample_mflix>
```

Para asegurarte de devolver el documento que insertaste, puedes consultar por `_id`.

- Insertar varios documentos

`db.collection.insertMany()` puede insertar varios documentos en una colección. Pase una matriz de documentos al método. Si los documentos no especifican un campo `_id`, MongoDB agrega el campo `_id` con un valor `ObjectId` a cada documento.

Para insertar dos nuevos documentos en la `sample_mflix.movies` colección:

```
db.movies.insertMany([
  {
    title: "Jurassic World: Fallen Kingdom",
    genres: [ "Action", "Sci-Fi" ],
    runtime: 130,
    rated: "PG-13",
    year: 2018,
    directors: [ "J. A. Bayona" ],
    cast: [ "Chris Pratt", "Bryce Dallas Howard", "Rafe Spall" ],

    type: "movie"
  },
  {
    title: "Tag",
    genres: [ "Comedy", "Action" ],
    runtime: 105,
    rated: "R",
    year: 2018,
    directors: [ "Jeff Tomsic" ],
```



INGENIERÍA DE SOFTWARE

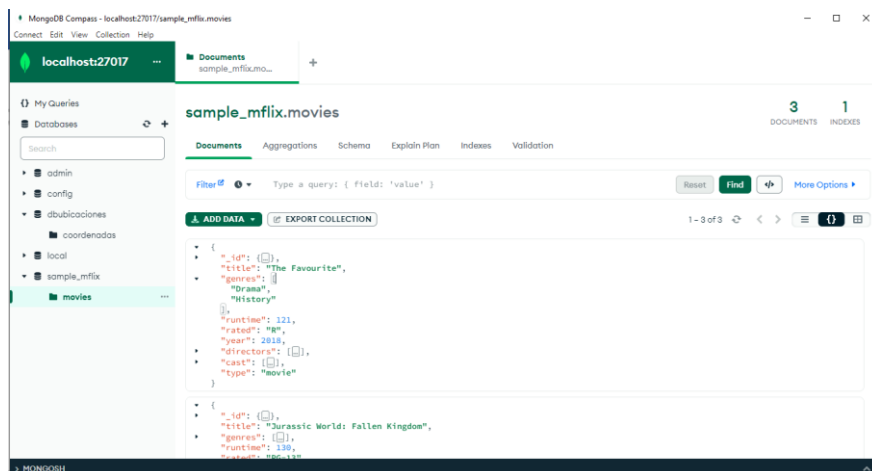
GUÍA 2


MONGODB – Operaciones y CRUD

```
cast: [ "Annabelle Wallis", "Jeremy Renner", "Jon Hamm" ],
type: "movie"
}
})
```

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
sample_mflix> db.movies.insertMany([
...   {
...     title: "Jurassic World: Fallen Kingdom",
...     genres: [ "Action", "Sci-Fi" ],
...     runtime: 130,
...     rated: "PG-13",
...     year: 2018,
...     directors: [ "J. A. Bayona" ],
...     cast: [ "Chris Pratt", "Bryce Dallas Howard", "Rafe Spall" ],
...     type: "movie"
...   },
...   {
...     title: "Tag",
...     genres: [ "Comedy", "Action" ],
...     runtime: 105,
...     rated: "R",
...     year: 2018,
...     directors: [ "Jeff Tomsic" ],
...     cast: [ "Annabelle Wallis", "Jeremy Renner", "Jon Hamm" ],
...     type: "movie"
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("641b689c88bc068804ed246f"),
    '1': ObjectId("641b689c88bc068804ed2470")
  }
}
sample_mflix>
```

Observamos lo realizado desde Compass:



	INGENIERÍA DE SOFTWARE
	GUÍA 2
	MONGODB – Operaciones y CRUD

insertMany() devuelve un documento que incluye los valores del campo _id de los documentos recién insertados.

Realizo búsquedas:

```
sample_mflix> db.movies.find({runtime:105})
[
  {
    _id: ObjectId("641b689c88bc068804ed2470"),
    title: 'Tag',
    genres: [ 'Comedy', 'Action' ],
    runtime: 105,
    rated: 'R',
    year: 2018,
    directors: [ 'Jeff Tomsic' ],
    cast: [ 'Annabelle Wallis', 'Jeremy Renner', 'Jon Hamm' ],
    type: 'movie'
  }
]
sample_mflix> db.movies.find({runtime:130})
[
  {
    _id: ObjectId("641b689c88bc068804ed246f"),
    title: 'Jurassic World: Fallen Kingdom',
    genres: [ 'Action', 'Sci-Fi' ],
    runtime: 130,
    rated: 'PG-13',
    year: 2018,
    directors: [ 'J. A. Bayona' ],
    cast: [ 'Chris Pratt', 'Bryce Dallas Howard', 'Rafe Spall' ],
    type: 'movie'
  }
]
sample_mflix> _
```

Consulta de Documentos

Utilice el db.collection.find() en MongoDB Shell para consultar documentos en una colección.

[*db.collection.find\(\)*](#)

Leer todos los documentos de una colección:

Para leer todos los documentos de la colección, pase un documento vacío como parámetro de filtro de consulta al método de búsqueda. El parámetro de filtro de consulta determina los criterios de selección.


```
db.movies.find()
```

Esta consulta es equivalente a:

```
SELECT * FROM movies
```



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
sample_mflix> db.movies.find()
[
  {
    _id: ObjectId("641b670888bc068804ed246e"),
    title: 'The Favourite',
    genres: [ 'Drama', 'History' ],
    runtime: 121,
    rated: 'R',
    year: 2018,
    directors: [ 'Yorgos Lanthimos' ],
    cast: [ 'Olivia Colman', 'Emma Stone', 'Rachel Weisz' ],
    type: 'movie'
  },
  {
    _id: ObjectId("641b689c88bc068804ed246f"),
    title: 'Jurassic World: Fallen Kingdom',
    genres: [ 'Action', 'Sci-Fi' ],
    runtime: 130,
    rated: 'PG-13',
    year: 2018,
    directors: [ 'J. A. Bayona' ],
    cast: [ 'Chris Pratt', 'Bryce Dallas Howard', 'Rafe Spall' ],
    type: 'movie'
  },
  {
    _id: ObjectId("641b689c88bc068804ed2470"),
    title: 'Tag',
    genres: [ 'Comedy', 'Action' ],
    runtime: 105,
    rated: 'R',
    year: 2018,
    directors: [ 'Jeff Tomsic' ],
    cast: [ 'Annabelle Wallis', 'Jeremy Renner', 'Jon Hamm' ],
    type: 'movie'
  }
]
sample_mflix>
```

	INGENIERÍA DE SOFTWARE
	GUÍA 2
	MONGODB – Operaciones y CRUD

Especificar condición de igualdad:

Para seleccionar documentos que coincidan con una condición de igualdad, especifique la condición como un <field>:<value> par en el documento de filtro de consulta.

```
db.movies.find( { "title": "Titanic" } )
```

Esto es equivalente a:

```
SELECT * FROM movies WHERE title = "Titanic"
```

```
sample_mflix> db.movies.find( { "title": "Titanic" } )
sample_mflix> db.movies.find( { "title": "Tag" } )
[
  {
    _id: ObjectId("641b689c88bc068804ed2470"),
    title: 'Tag',
    genres: [ 'Comedy', 'Action' ],
    runtime: 105,
    rated: 'R',
    year: 2018,
    directors: [ 'Jeff Tomsic' ],
    cast: [ 'Annabelle Wallis', 'Jeremy Renner', 'Jon Hamm' ],
    type: 'movie'
  }
]
sample_mflix> _
```


Especificar condiciones mediante operadores de consulta:

Usar operadores de consulta en un documento de filtro de consulta para realizar comparaciones y evaluaciones más complejas. Los operadores de consulta en un documento de filtro de consulta tienen el siguiente formato:

```
{ <field1>: { <operator1>: <value1> }, ... }
```

Para devolver todas las películas de la sample_mflix.movies colección que están clasificadas PG o PG-13:

```
db.movies.find( { rated: { $in: [ "PG", "PG-13" ] } } )
```


	INGENIERÍA DE SOFTWARE
	GUÍA 2
	MONGODB – Operaciones y CRUD

```
sample_mflix> db.movies.find( { rated: { $in: [ "PG", "PG-13" ] } } )
[
  {
    _id: ObjectId("641b689c88bc068804ed246f"),
    title: 'Jurassic World: Fallen Kingdom',
    genres: [ 'Action', 'Sci-Fi' ],
    runtime: 130,
    rated: 'PG-13',
    year: 2018,
    directors: [ 'J. A. Bayona' ],
    cast: [ 'Chris Pratt', 'Bryce Dallas Howard', 'Rafe Spall' ],
    type: 'movie'
  }
]
sample_mflix> _
```

Esta operación corresponde a la siguiente instrucción SQL:

```
SELECT * FROM movies WHERE rated in ("PG", "PG-13")
```

Especificar operadores lógicos (AND/ OR)


Una consulta compuesta puede especificar condiciones para más de un campo en los documentos de la colección. Implícitamente, una AND conjunción lógica conecta las cláusulas de una consulta compuesta para que la consulta seleccione los documentos de la colección que cumplen todas las condiciones.

Para devolver películas que se clasificadas R y tienen una duración de 121:

```
db.movies.find( { rated : "R" , runtime : 121 } )
```

```
sample_mflix> db.movies.find( { rated : "R" , runtime : 121 } )
[
  {
    _id: ObjectId("641b670888bc068804ed246e"),
    title: 'The Favourite',
    genres: [ 'Drama', 'History' ],
    runtime: 121,
    rated: 'R',
    year: 2018,
    directors: [ 'Yorgos Lanthimos' ],
    cast: [ 'Olivia Colman', 'Emma Stone', 'Rachel Weisz' ],
    type: 'movie'
  }
]
sample_mflix>
```

Utilizar el `$or` operador para especificar una consulta compuesta que une cada cláusula con una conjunción lógica `OR` para que la consulta seleccione los documentos de la colección que cumplen al menos una condición.

	INGENIERÍA DE SOFTWARE
	GUÍA 2
	MONGODB – Operaciones y CRUD

Para devolver películas de la `sample_mflix.movies` colección que se lanzaron en 2017 o su título es “Tag”:

```
db.movies.find({$or: [{title : "Tag"}, {year : 2017}]})
```

```
sample_mflix> db.movies.find( { $or: [{title : "Tag"}, {year : 2017}]} )
[
  {
    _id: ObjectId("641b689c88bc068804ed246f"),
    title: 'Jurassic World: Fallen Kingdom',
    genres: [ 'Action', 'Sci-Fi' ],
    runtime: 130,
    rated: 'PG-13',
    year: 2017,
    directors: [ 'J. A. Bayona' ],
    cast: [ 'Chris Pratt', 'Bryce Dallas Howard', 'Rafe Spall' ],
    type: 'movie'
  },
  {
    _id: ObjectId("641b689c88bc068804ed2470"),
    title: 'Tag',
    genres: [ 'Comedy', 'Action' ],
    runtime: 105,
    rated: 'R',
    year: 2018,
    directors: [ 'Jeff Tomsic' ],
    cast: [ 'Annabelle Wallis', 'Jeremy Renner', 'Jon Hamm' ],
    type: 'movie'
  }
]
sample_mflix> 
```

Actualizar Documentos

El shell de MongoDB proporciona los siguientes métodos para actualizar documentos en una colección:

Para actualizar un solo documento, use

[`db.collection.updateOne\(\)`](#)

Para actualizar varios documentos, utilice


[`db.collection.updateMany\(\)`](#)

Para reemplazar un documento, utilice

[`db.collection.replaceOne\(\)`](#)

Para actualizar un documento, MongoDB proporciona operadores de actualización, como

[`\$set`](#), para modificar los valores de los campos.

	INGENIERÍA DE SOFTWARE
	GUÍA 2
	MONGODB – Operaciones y CRUD

Para usar los operadores de actualización, pase a los métodos de actualización un documento de actualización de la forma:


```
{
  <update operator>: { <field1>: <value1>, ... },
  <update operator>: { <field2>: <value2>, ... },
  ...
}
```

Actualizar un solo documento


Utilizar el `db.collection.updateOne()` para actualizar el primer documento que coincida con un filtro especificado.

Para actualizar el primer documento de la `sample_mflix.movies` colección donde `titles` igual a "Tag":

```
sample_mflix> db.movies.updateOne ( { title: "Tag" }, { $set: { year: 2023 } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

 mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=

```
sample_mflix> db.movies.find({title : "Tag"})
[
  {
    _id: ObjectId("641b689c88bc068804ed2470"),
    title: 'Tag',
    genres: [ 'Comedy', 'Action' ],
    runtime: 105,
    rated: 'R',
    year: 2023,
    directors: [ 'Jeff Tomsic' ],
    cast: [ 'Annabelle Wallis', 'Jeremy Renner', 'Jon Hamm' ],
    type: 'movie'
  }
]
sample_mflix>
```

	INGENIERÍA DE SOFTWARE
	GUÍA 2
	MONGODB – Operaciones y CRUD

Actualizar varios documentos

Utilizar el `db.collection.updateMany()` para actualizar todos los documentos que coincidan con un filtro específico.

```
sample_mflix> db.movies.updateMany( {rated:"R"}, {$set: {year: 2015}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
sample_mflix> db.movies.find({rated : "R"})
[
  {
    _id: ObjectId("641b670888bc068804ed246e"),
    title: 'The Favourite',
    genres: [ 'Drama', 'History' ],
    runtime: 121,
    rated: 'R',
    year: 2015,
    directors: [ 'Yorgos Lanthimos' ],
    cast: [ 'Olivia Colman', 'Emma Stone', 'Rachel Weisz' ],
    type: 'movie'
  },
  {
    _id: ObjectId("641b689c88bc068804ed2470"),
    title: 'Tag',
    genres: [ 'Comedy', 'Action' ],
    runtime: 105,
    rated: 'R',
    year: 2015,
    directors: [ 'Jeff Tomsic' ],
    cast: [ 'Annabelle Wallis', 'Jeremy Renner', 'Jon Hamm' ],
    type: 'movie'
  }
]
sample_mflix>
```

Eliminar documentos

El shell de MongoDB proporciona los siguientes métodos para eliminar documentos de una colección:

Para eliminar varios documentos, utilice

`db.collection.deleteMany()`


Para eliminar un solo documento, utilice

`db.collection.deleteOne()`

Eliminar todos los documentos que coincidan con una condición

Puede especificar criterios o filtros que identifiquen los documentos que desea eliminar. El filtro use la misma sintaxis que las operaciones de lectura.

Para especificar condiciones de igualdad, utilice `<field>:<value>` expresiones en el documento de filtro de consulta.

	INGENIERÍA DE SOFTWARE
	GUÍA 2
	MONGODB – Operaciones y CRUD

Para eliminar todos los documentos que coincidan con un criterio de eliminación, pase un parámetro de filtro al deleteMany() método.

```
sample_mflix> db.movies.deleteMany({title: "Tag"})
{ acknowledged: true, deletedCount: 1 }
sample_mflix> db.movies.find()
[
  {
    _id: ObjectId("641b670888bc068804ed246e"),
    title: 'The Favourite',
    genres: [ 'Drama', 'History' ],
    runtime: 121,
    rated: 'R',
    year: 2015,
    directors: [ 'Yorgos Lanthimos' ],
    cast: [ 'Olivia Colman', 'Emma Stone', 'Rachel Weisz' ],
    type: 'movie'
  },
  {
    _id: ObjectId("641b689c88bc068804ed246f"),
    title: 'Jurassic World: Fallen Kingdom',
    genres: [ 'Action', 'Sci-Fi' ],
    runtime: 130,
    rated: 'PG-13',
    year: 2017,
    directors: [ 'J. A. Bayona' ],
    cast: [ 'Chris Pratt', 'Bryce Dallas Howard', 'Rafe Spall' ],
    type: 'movie'
  }
]
sample_mflix>
```

Eliminar solo un documento que coincida con una condición

Para eliminar como máximo un solo documento que coincida con un filtro específico (aunque varios documentos pueden coincidir con el filtro especificado), utilice el db.collection.deleteOne() método.



```
sample_mflix> db.movies.deleteOne({rated: "PG-13"})
{ acknowledged: true, deletedCount: 1 }
sample_mflix> db.movies.find()
[
  {
    _id: ObjectId("641b670888bc068804ed246e"),
    title: 'The Favourite',
    genres: [ 'Drama', 'History' ],
    runtime: 121,
    rated: 'R',
    year: 2015,
    directors: [ 'Yorgos Lanthimos' ],
    cast: [ 'Olivia Colman', 'Emma Stone', 'Rachel Weisz' ],
    type: 'movie'
  }
]
sample_mflix>
```

Eliminar todos los documentos

Para eliminar todos los documentos de una colección, pase un vacío filtrar documento {} a la `db.collection.deleteMany()` método.

```
sample_mflix> db.movies.deleteMany({})
{ acknowledged: true, deletedCount: 1 }
sample_mflix> db.movies.find()
[]
sample_mflix>
```