Ejemplo 1: Operaciones con bases de datos

```sql
-- Ejemplo 1
-- Crear una base de datos -> Nombre propio de preferencia
create database jesusismael;
use jesusismael;
drop table jesusismael;

```

Ejemplo 2: Realizando operaciones con tablas

```sql
-- Ejemplo 2
-- Descargar datos del enlace users, ratings y movies
-- Crear tabla users
create database if not exists jesusismael;
drop table if exists jesusismael;

create table if not exists users (
    id_users int primary key,
    gender varchar(1),
    age int,
    occup int,
    zip_code varchar(20)
);
```

RETO 1

```sql
-- Reto 1
-- 1.1 Definir los campos y tipos de datos para la tabla movies haciendo uso de los archivos mo
    -- id_movies -> int
    -- title  -> varchar(n)
    -- genres -> varchar(n)

-- 1.2 Crear la tabla movies (recuerda usar el mismo nombre del archivo sin la extensión para v
create table if not exists movies (
    id_movies int primary key,
    title varchar(80),
    genres varchar(80)
);
describe movies;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
id_movies	int	NO	PRI	NULL	
title	varchar(80)	YES		NULL	
genres	varchar(80)	YES		NULL	

Result Grid

```sql
-- 1.3 Definir los campos y tipos de datos para la tabla ratings haciendo uso de los archivos r
    -- users_id -> int
    -- movies_id -> int
    -- rating -> int
    -- time_stamp -> bigint
-- 1.4 Crear la tabla ratings (recuerda usar el mismo nombre del archivo sin la extensión para
create table if not exists ratings (
    users_id int,
    movies_id int,
    rating int,
    time_stamp bigint
--  foreign key(users_id) references users(id_users),
--  foreign key(movies_id) references movies(id_movies)
);
describe ratings;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
users_id	int	YES		NULL	
movies_id	int	YES		NULL	
rating	int	YES		NULL	
time_stamp	bigint	YES		NULL	

Result Grid | Form

RETO 2

```sql
-- Reto 2
-- 2.1 Usando como base el archivo movies.dat, limpiarlo e importar los datos en la tabla movies creada en el Reto 1.
select count(*)
from movies;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(*)
3882

```sql
-- 2.2 Usando como base el archivo ratings.dat, limpiarlo e importar los datos en la tabla ratings creada en el Reto 2.
-- Activar la importación de datos por comandos
select @@local_infile;
set global local_infile=1;
SHOW VARIABLES LIKE "local_infile"; -- status local_infile [on]
SHOW VARIABLES LIKE "secure_file_priv"; -- ruta de donde ser cargaran los datos

-- Carga de la tabla ratings por comandos
load data infile 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/ratings.csv'
into table ratings
fields terminated by ','
enclosed by '"'
lines terminated by '\n'
ignore 1 lines;
```

```sql
-- Número de registros en la tabla ratings
select count(*)
from ratings;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(*)
1000211

```sql
-- 2.3 Finalmente, añade un registro en cada tabla usando INSERT INTO.
insert into movies (id_movies,title,genres) values (4000,'The hidden blade',"Assassins's Creed");
insert into ratings (users_id,movies_id,rating, time_stamp) values (8000,200,10,978303276);

 select *
 from users
 order by id_users desc
 limit 1;
```

Result Grid | Filter Rows:

id_users	gender	age	occup	zip_code
8000	F	23	20	45150
NULL	NULL	NULL	NULL	NULL

```sql
 select *
 from movies
 order by id_movies desc
 limit 1;
```

Result Grid | Filter Rows:

id_movies	title	genres
4000	The hidden blade	Assassins's Creed
NULL	NULL	NULL

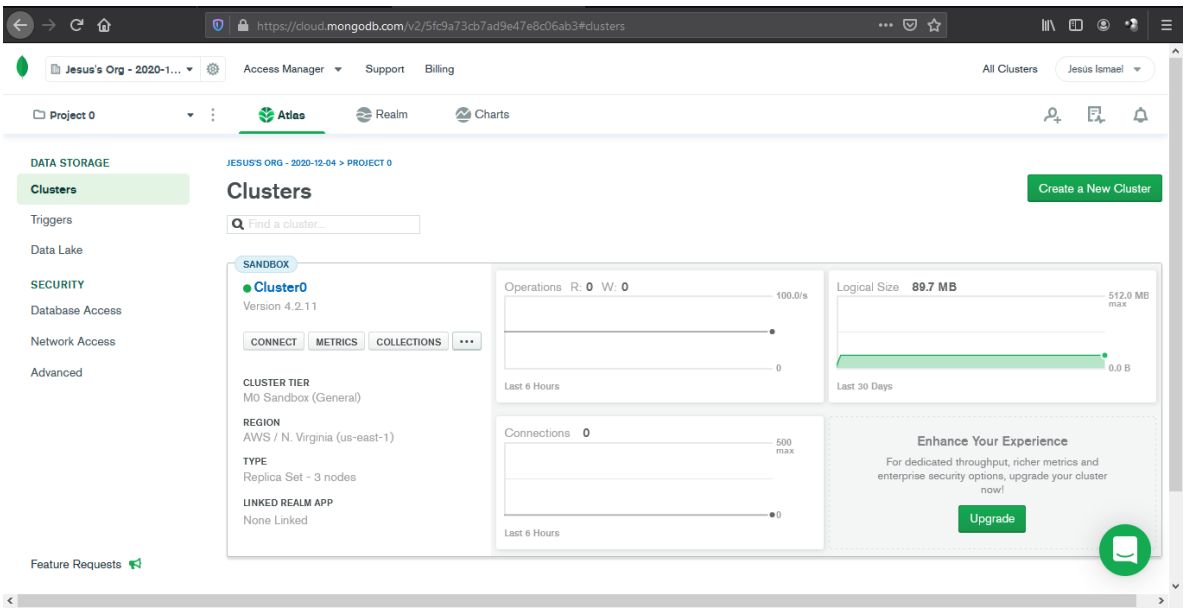
```sql
 select *
 from ratings
 order by users_id desc
 limit 1;
```

Result Grid | Filter Rows:

users_id	movies_id	rating	time_stamp
8000	200	10	978303276

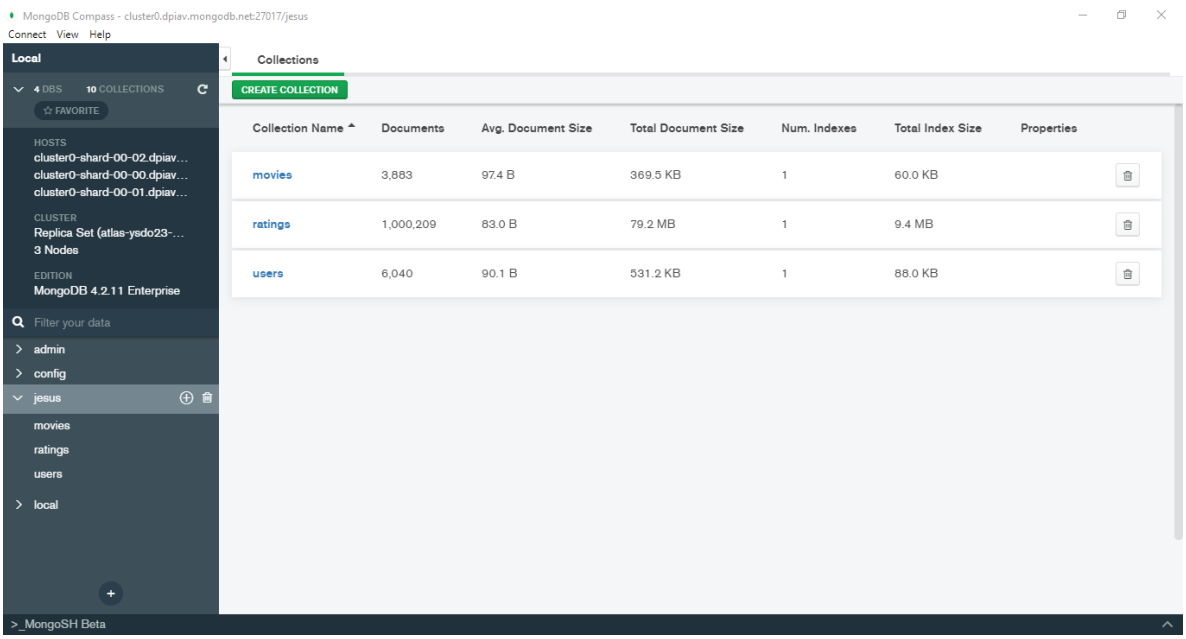
Ejemplo 4 – Configuración de MongoDB en la nube

Creación de cuenta propia en mongo atlas



Ejemplo 5 – Operaciones con bases de datos

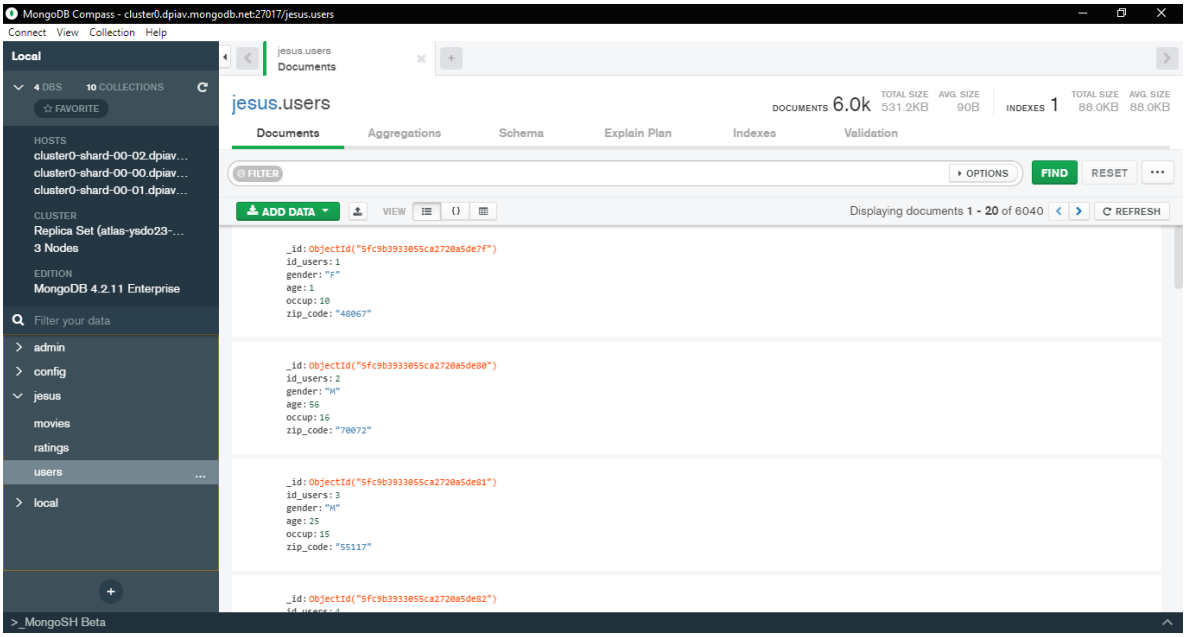
Enlazar MongoDB Compass y crear las colecciones de movies, users y ratings.



Ejemplo 6: Realizando operaciones con Colecciones e importando datos

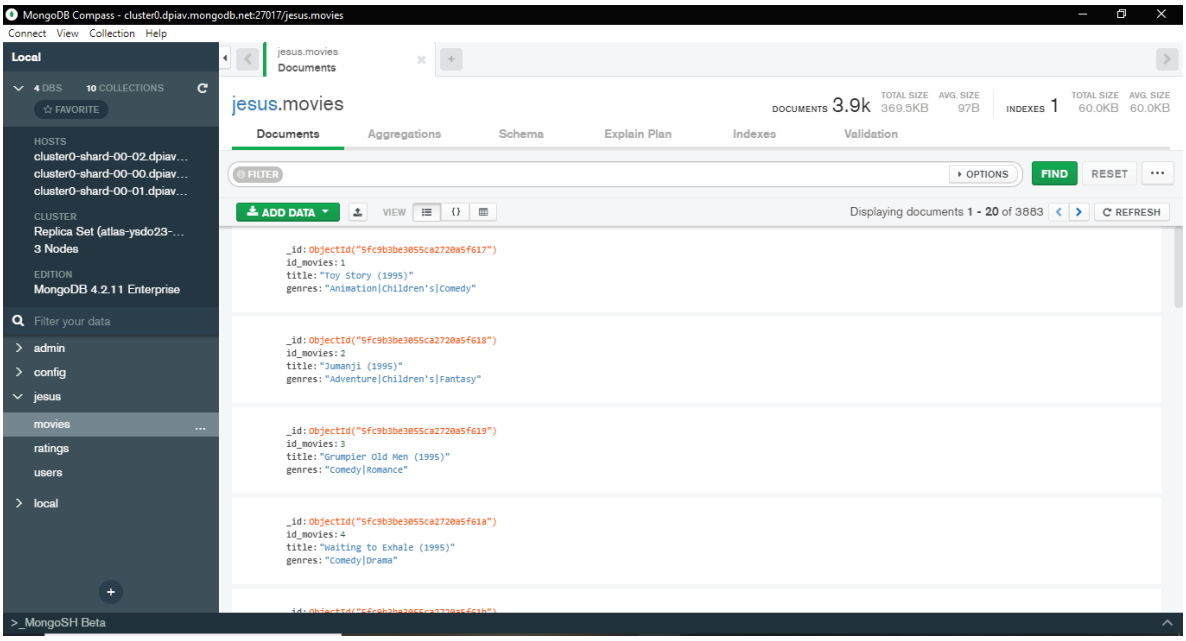
Importar datos a cada colección. Tomar de referencia los archivos .csv antes guardados

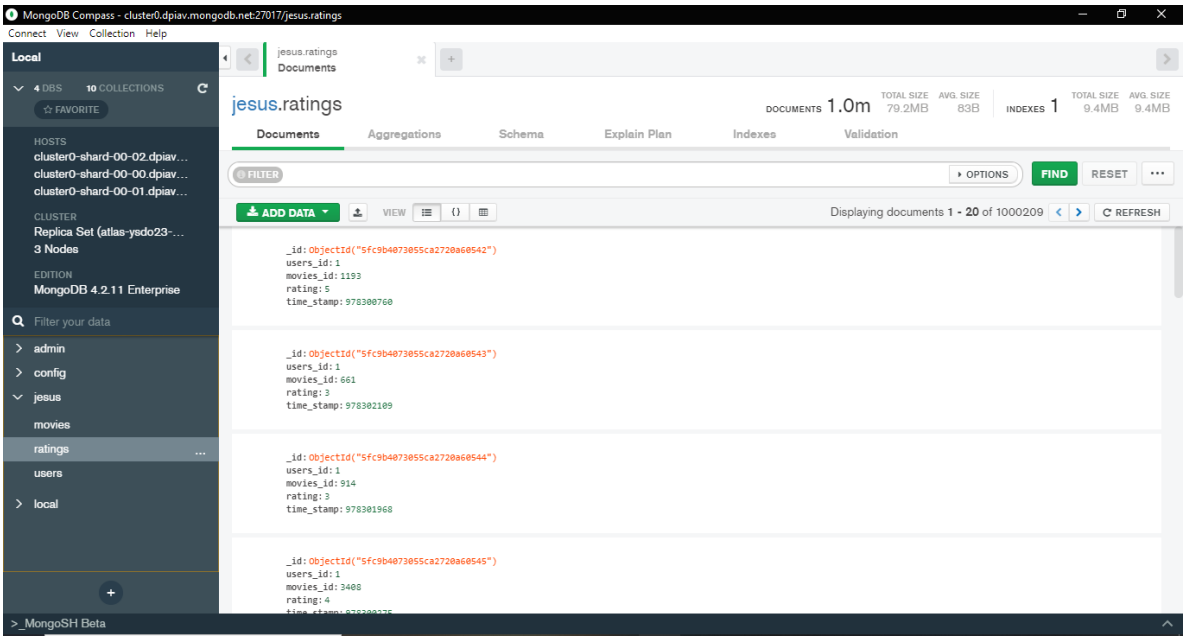
Importación de documentos a la colección de **users**



RETO 3

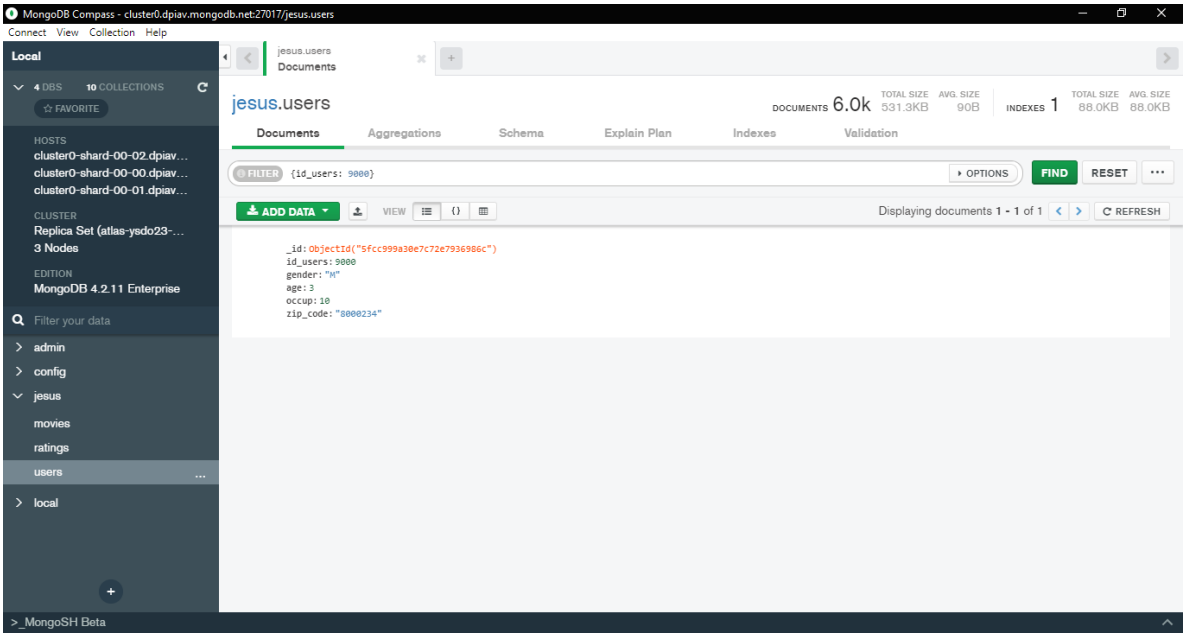
Importar documentos a las colecciones **movies** y **ratings**

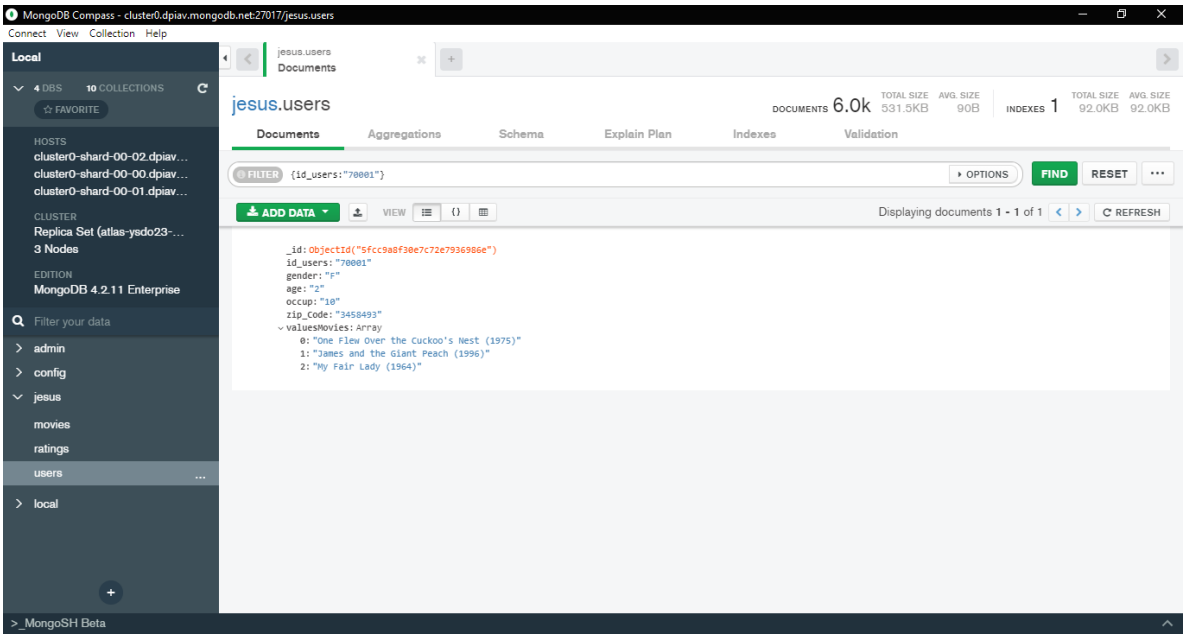


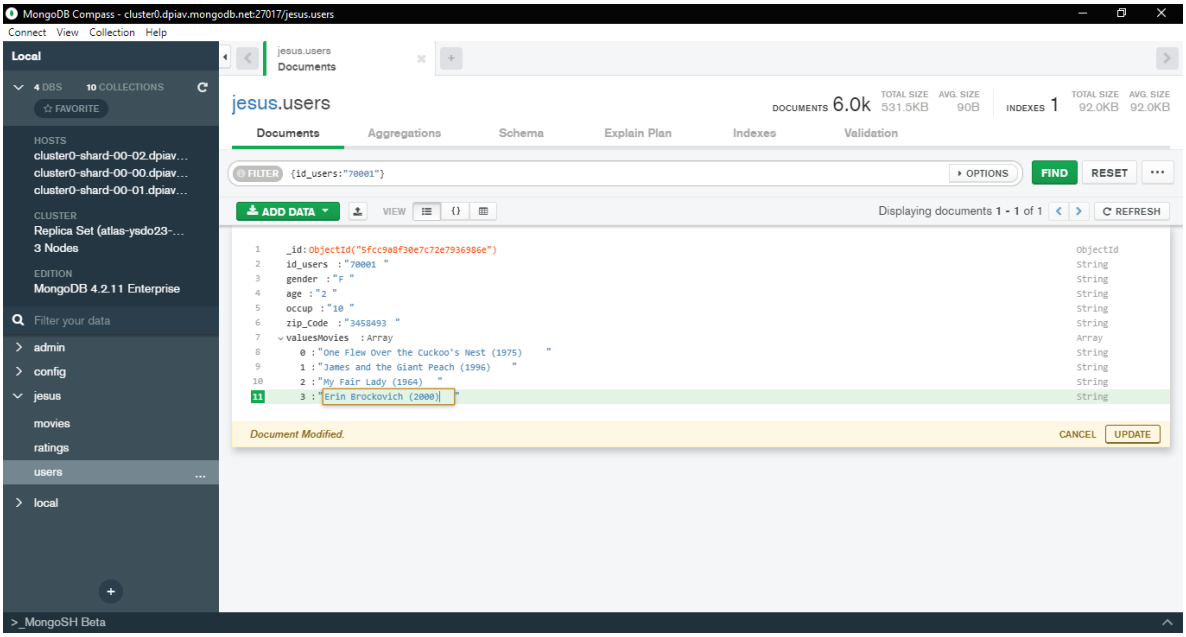


Ejemplo 7: Realizando operaciones con Documentos

Insertar documentos de manera manual en **users** y realizar filtros y actualizaciones a los mismos documentos.







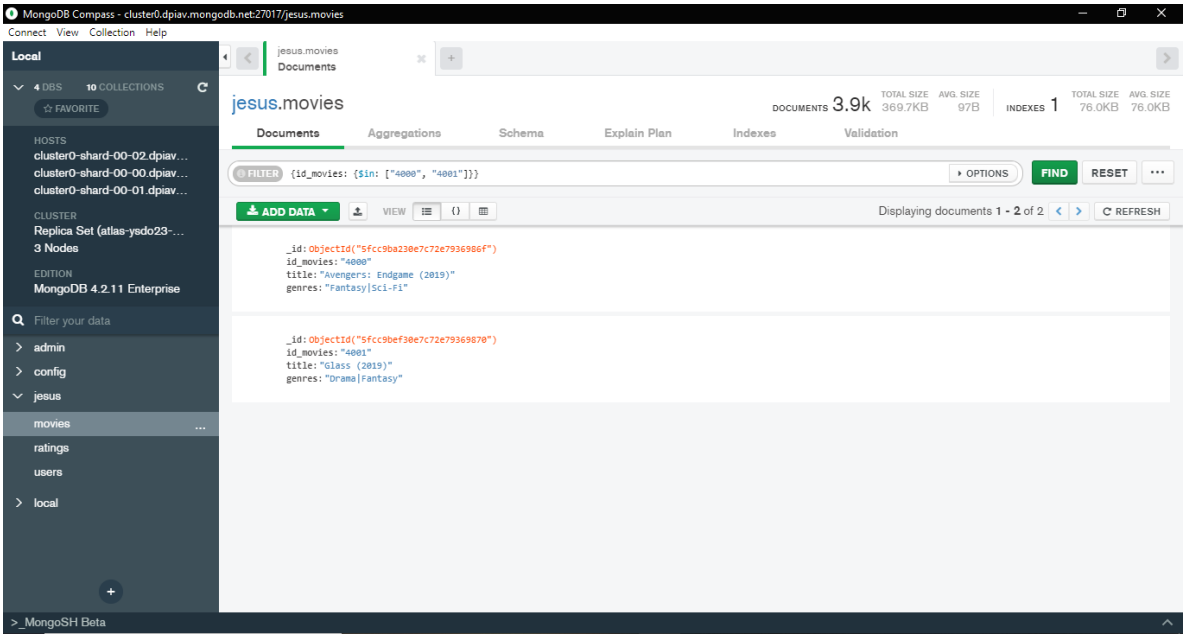
EJERCICIOS SESIÓN 4

Agregar los siguientes registros en formato CSV a la Colección movies

4000,Avengers: Endgame (2019),Fantasy|Sci-Fi
4001,Glass (2019),Drama|Fantasy

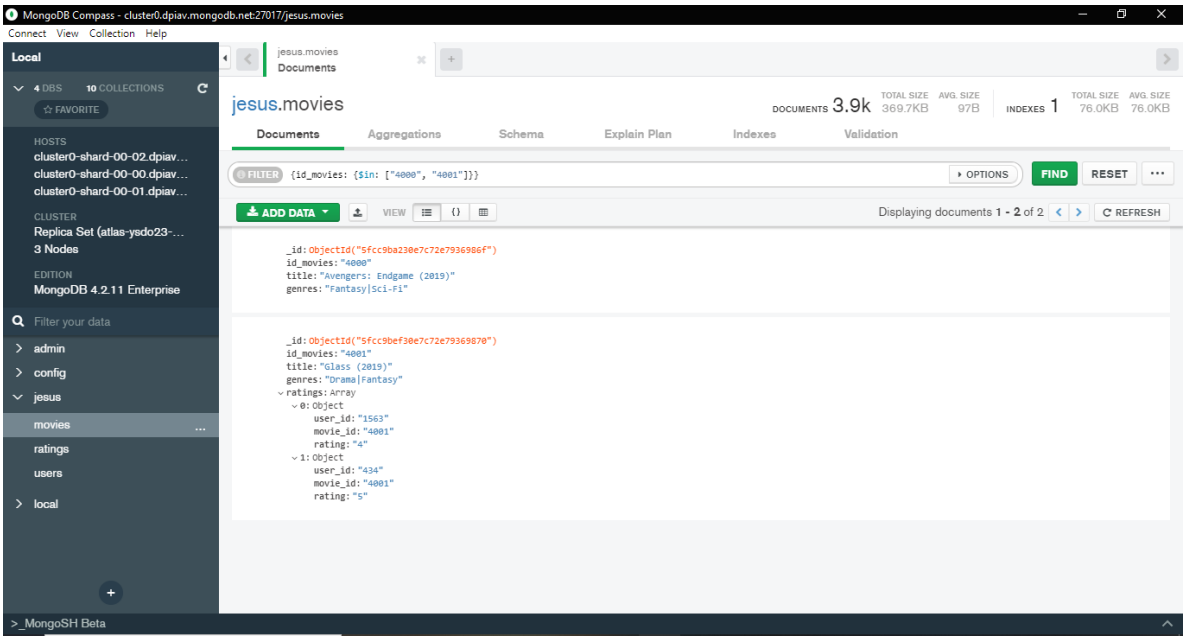
Buscar documento por filtro

{id: {$in: ["4000", "4001"]}}



Modificar el documento con id=4001 en la Colección movies para que contenga la siguiente información:

```
{
 id:"4001",
 titulo:"Glass (2019)",
 genres:"Drama|Fantasy",
 valoraciones: [
  {
   userid: "1563",
   movieid: "4001",
   rating: "4"
  },
  {
   userid: "434",
   movieid: "4001",
   rating: "5"
  }
 ]
}
```


MongoDB Compass - cluster0.dpiav.mongodb.net:27017/jesus.movies
Connect View Collection Help
Local
4 DBS 10 COLLECTIONS
FAVORITE
HOSTS
cluster0-shard-00-02.dpiav...
cluster0-shard-00-00.dpiav...
cluster0-shard-00-01.dpiav...
CLUSTER
Replica Set (atlas-ysdo23-...
3 Nodes
EDITION
MongoDB 4.2.11 Enterprise
Filter your data
admin
config
jesus
movies
ratings
users
local
jesus.movies
Documents
jesus.movies
DOCUMENTS 3.9k
TOTAL SIZE 369.7KB
AVG. SIZE 97B
INDEXES 1
TOTAL SIZE 76.0KB
AVG. SIZE 76.0KB
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
FILTER {id_movies: {$in: ["4000", "4001"]}}
OPTIONS
FIND
RESET
ADD DATA
VIEW
Displaying documents 1 - 2 of 2
REFRESH
_id: ObjectId("5fcc9ba230e7c72e7936986f")
id_movies: "4000"
title: "Avengers: Endgame (2019)"
genres: "Fantasy|Sci-Fi"
_id: ObjectId("5fcc9bef30e7c72e79369870")
id_movies: "4001"
title: "Glass (2019)"
genres: "Drama|Fantasy"
ratings: Array
0: Object
user_id: "1563"
movie_id: "4001"
rating: "4"
1: Object
user_id: "434"
movie_id: "4001"
rating: "5"
>_MongoSH Beta