**Name: Nikhil Jadhav**

**Student ID: 801075504**

**ITCS6160 DATABASE SYSTEMS**

**Dr. Pamela P Thompson**

**Exercise on MongoDB**

**Note**: Please go through the links provided and install MongoDB before you start this assignment. Submit the queries and the screenshots of query execution and results in word or pdf format.

**Links:**

<https://www.mongodb.com/nosql-explained>

<https://www.youtube.com/watch?v=pWbMrx5rVBE>

<https://docs.mongodb.com/manual/>

**Questions:**

1. Create a database called MovieDB. Create a collection named “Movies” under this MovieDB database.

Answers:

**db.createCollection('Movies2');**

****

1. Insert documents to the Movies collection (atleast 10 documents). Let this collection have fields like movieId, title, genre, releaseYear, director, and rating (1-10). Let the genres be comedy, action, horror, thriller and SciFi.

Answers:

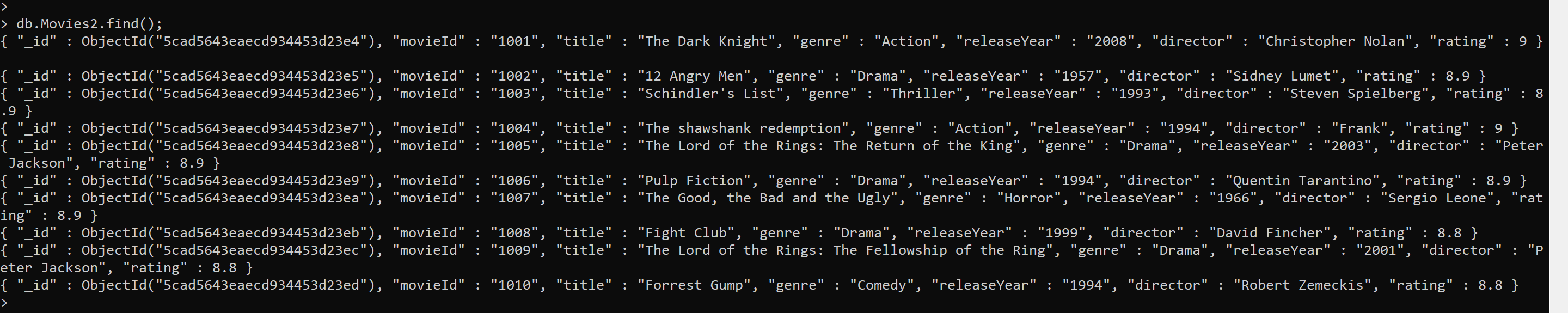
**db.Movies2.insert([{"movieId":"1001", "title":"The Dark Knight", "genre":"Action", "releaseYear":"2008","director":"Christopher Nolan","rating":9}, {"movieId":"1002", "title":"12 Angry Men", "genre":"Drama", "releaseYear":"1957", "director":"Sidney Lumet", "rating":8.9}, {"movieId":"1003", "title":"Schindler's List", "genre":"Thriller", "releaseYear":"1993","director":"Steven Spielberg", "rating":8.9}, {"movieId":"1004", "title":"The shawshank redemption", "genre":"Action", "releaseYear":"1994","director":"Frank","rating":9}, {"movieId":"1005", "title":"The Lord of the Rings: The Return of the King", "genre":"Drama", "releaseYear":"2003","director":"Peter Jackson","rating":8.9}, {"movieId":"1006", "title":"Pulp Fiction", "genre":"Drama", "releaseYear":"1994","director":"Quentin Tarantino","rating":8.9}, {"movieId":"1007", "title":"The Good, the Bad and the Ugly", "genre":"Horror", "releaseYear":"1966","director":"Sergio Leone","rating":8.9}, {"movieId":"1008", "title":"Fight Club", "genre":"Drama", "releaseYear":"1999","director":"David Fincher","rating":8.8}, {"movieId":"1009", "title":"The Lord of the Rings: The Fellowship of the Ring", "genre":"Drama", "releaseYear":"2001","director":"Peter Jackson","rating":8.8}, {"movieId":"1010", "title":"Forrest Gump", "genre":"Comedy", "releaseYear":"1994","director":"Robert Zemeckis","rating":8.8}])**



1. Write a MongoDB query to display all the documents in the collection Movies.

Answers:

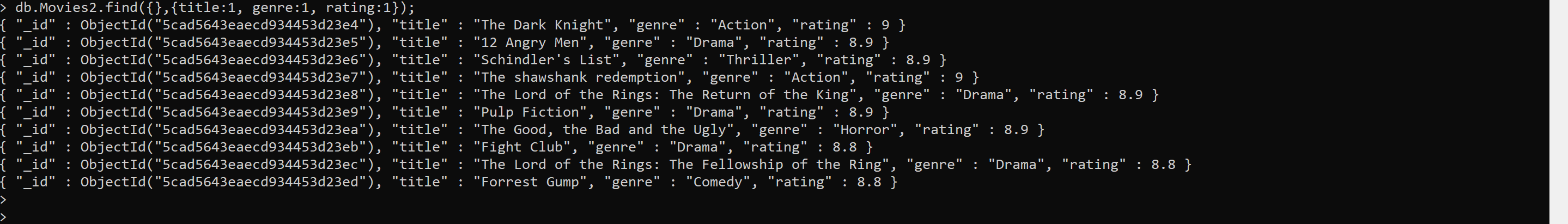
**db.Movies2.find();**

****

1. Write a MongoDB query to display the fields title, genre and rating.

Answers:

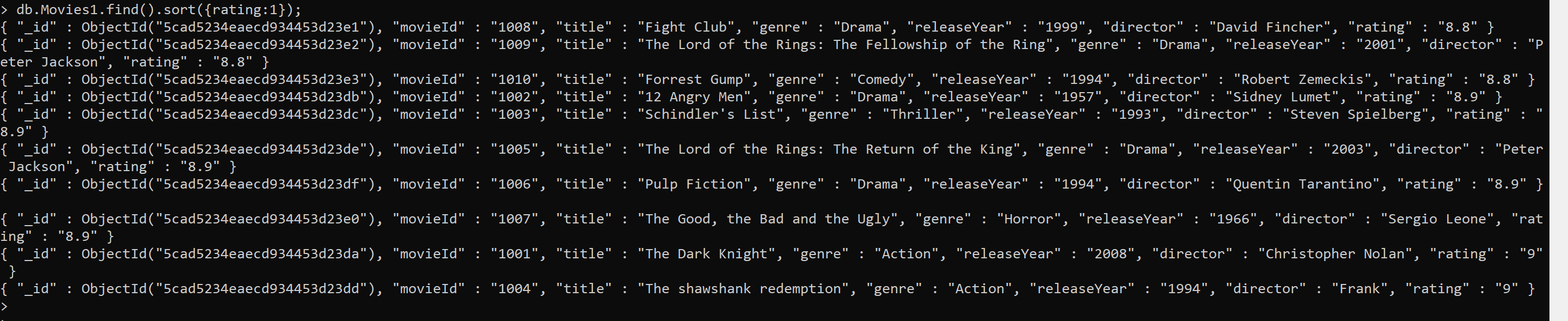
**db.Movies2.find({},{title:1, genre:1, rating:1});**

****

1. Write a MongoDB query to display all the documents sorted by the rating field in ascending order.

Answers:

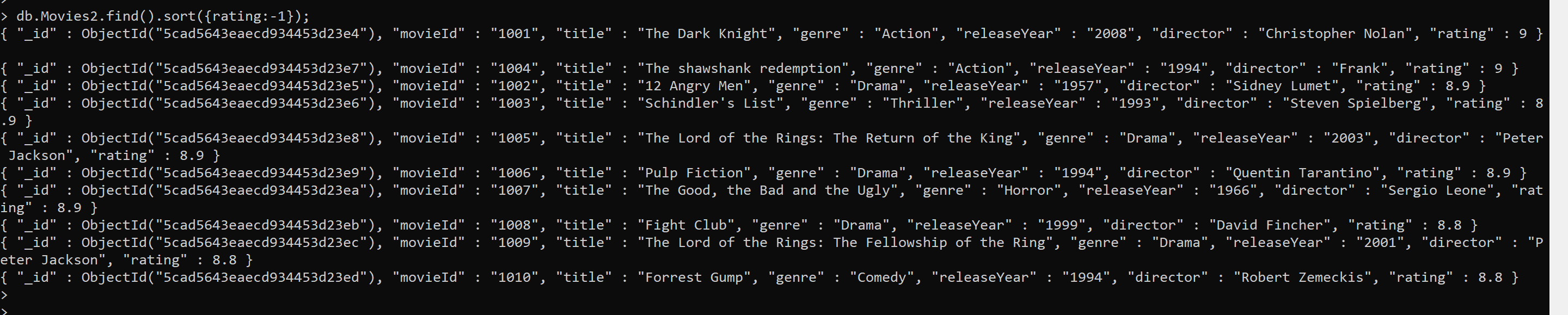
**db.Movies1.find().sort({rating:1});**

****

1. Write a MongoDB query to display all the documents sorted by the rating field in descending order.

Answers:

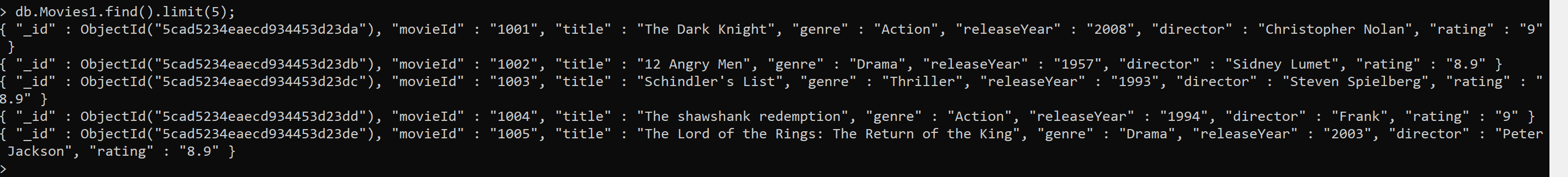
**db.Movies2.find().sort({rating:-1});**

****

1. Write a MongoDB query to display maximum 5 documents from Movies collection.

Answers:

**db.Movies1.find().limit(5);**

****

1. Write a MongoDB query to display the count of the documents where genre is "comedy".

Answers:

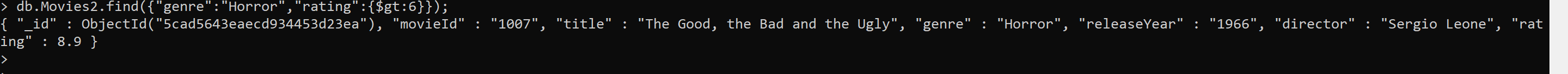
**db.Movies2.find({genre:"Comedy"}).count();**

****

1. Write a MongoDB query to display documents which has genre as "horror" and rating greater than 6.

Answers:

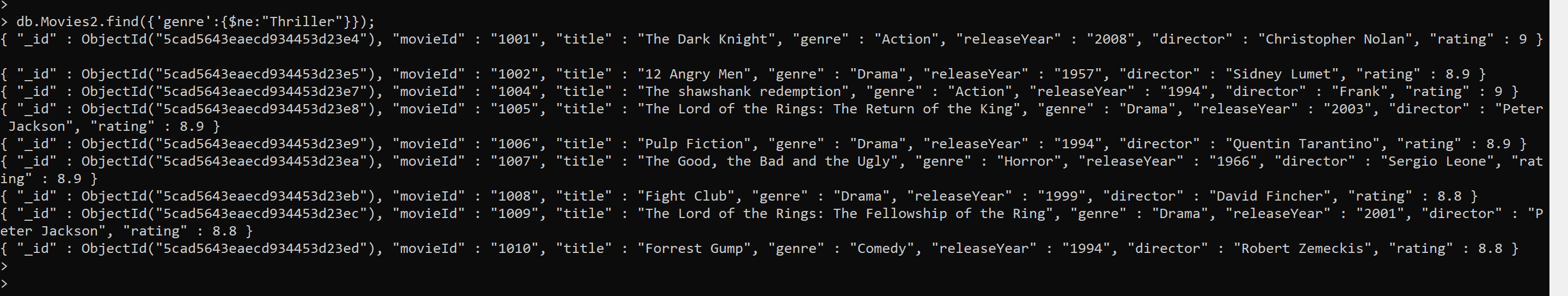
**db.Movies2.find({"genre":"Horror","rating":{$gt:6}});**

****

1. Write a MongoDB query to display documents whose genre is not equal to thriller.

Answers:

**db.Movies2.find({'genre':{$ne:"Thriller"}});**

****