1. Find all movies with full information from the 'movies' collection that released in the year 1893.

A) db.movies.find({year:1893})

2. Find all movies with full information from the 'movies' collection that have a runtime greater than 120 minutes.

A) db.movies.find({runtime:{$gt:120}})

3. Find all movies with full information from the 'movies' collection that have "Short" genre.

A) db.movies.find({genres:"Short"})

4. Retrieve all movies from the 'movies' collection that were directed by "William K.L. Dickson" and include complete information for each movie.

A) db.movies.find({directors:"William K.L. Dickson"})

5. Retrieve all movies from the 'movies' collection that were released in the USA and include complete information for each movie.

A) db.movies.find({countries:"USA"})

6. Retrieve all movies from the 'movies' collection that have complete information and are rated as "UNRATED".

A) db.movies.find({rated:"UNRATED"})

7. Retrieve all movies from the 'movies' collection that have complete information and have received more than 1000 votes on IMDb.

A) db.movies.find({"imdb.votes":{$gt:1000}})

8. Retrieve all movies from the 'movies' collection that have complete information and have an IMDb rating higher than 7.

A) db.movies.find({"imdb.rating":{$gt:7}})

9. Retrieve all movies from the 'movies' collection that have complete information and have a viewer rating higher than 4 on Tomatoes.

A) db.movies.find({"tomatoes.viewer.rating":{$gt:4}})

10. Retrieve all movies from the 'movies' collection that have received an award.

A) db.movies.find({"awards.wins":{$gt:0}})

11. Find all movies with title, languages, released, directors, writers, awards, year, genres, runtime, cast, countries from the 'movies' collection in MongoDB that have at least one nomination.

A) db.movies.find({"awards.nominations":{$gt:0}},{title:1,languages:1,released:1,directors:1,writers:1,awards:1,year:1,genres:1,runtime:1,countries:1})

12. Find all movies with title, languages, released, directors, writers, awards, year, genres, runtime, cast, countries from the 'movies' collection in MongoDB with cast including "Charles Kayser".

A) db.movies.find( { cast: "Charles Kayser" }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, awards: 1, year: 1, genres: 1, runtime: 1, cast: 1, countries: 1, \_id: 0 } )

13. Retrieve all movies with title, languages, released, directors, writers, countries from the 'movies' collection in MongoDB that released on May 9, 1893.

A) db.movies.find({ released: ISODate("1893-05-09T00:00:00.000Z") }, { title: 1, languages: 1, released: 1, directors: 1, writers: 1, countries: 1 })

14. Retrieve all movies with title, languages, released, directors, writers, countries from the 'movies' collection in MongoDB that have a word "scene" in the title.

A) db.movies.find({title:/Scene/},{title:1,languages:1,released:1,directors:1,writers:1,countries:1})

15. Find all movies with title, languages, released, directors, viewer, writers, countries from the 'movies' collection in MongoDB that have a viewer rating of at least 3 and less than 4 on Tomatoes.

A) db.movies.find( { 'tomatoes.viewer.rating': { $gte: 3, $lt: 4 } }, { title: 1, languages: 1, released: 1, directors: 1, 'tomatoes.viewer': 1, writers: 1, countries: 1 } )

16. Retrieve all movies with title, languages, released, year, directors, writers, countries from the 'movies' collection in MongoDB that released before the year 1900.

A) db.movies.find( { year: { $lt: 1900 }, type: "movie" }, { title: 1, languages: 1, released: 1, year: 1, directors: 1, writers: 1, countries: 1 } )

17. Find all movies with title, languages, fullplot, released, directors, writers, countries from the 'movies' collection in MongoDB that have a fullplot containing the word "fire".

A) db.movies.find({fullplot: { $regex: /fire/i } }, { title: 1, languages: 1, fullplot: 1, released: 1, directors: 1, writers: 1, countries: 1 })

18. Return all movies with title, languages, plot, released, directors, writers, and countries from the 'movies' collection in MongoDB where the word "beer" mentioned in the plot.

A) db.movies.find({ plot: { $regex: /beer/i } }, { title: 1, languages: 1, plot: 1, released: 1, directors: 1, writers: 1, countries: 1 })

19. Return all movies with title, languages, fullplot, released, directors, writers, and countries from the 'movies' collection in MongoDB where the word "metal" mentioned in the fullplot.

A) db.movies.find({ fullplot: /metal/i },{ title: 1, languages: 1, fullplot: 1, released: 1, directors: 1, writers: 1, countries: 1 });

20. Find all movies with title, languages, released, runtime, directors, writers, countries from the 'movies' collection in MongoDB that have a runtime between 60 and 90 minutes.

A) db.movies.find({ runtime: { $gte: 60, $lte: 90 } },{ title: 1, languages: 1, released: 1, runtime: 1, directors: 1, writers: 1, countries: 1 })

21. Find all movies with title, languages, released, runtime, directors, writers, countries, imdb from the 'movies' collection in MongoDB for the top 5 movies with the highest IMDb ratings.

A) db.movies.find({}, {title: 1,languages: 1,released: 1,runtime: 1,directors: 1,writers: 1,countries: 1,imdb: 1}).sort({ imdb: -1 }).limit(5)

22. Find all movies from the 'movies' collection in MongoDB with the average runtime of movies released in each country.

A) db.movies.aggregate([{$match: { type: 'movie' }},{$group: {\_id: '$countries',averageRuntime: { $avg: '$runtime' }}}])

23. Find from the 'movies' collection in MongoDB with the most common genre among the movies.

A) db.movies.aggregate([{$unwind: '$genres'},{$group: {\_id: '$genres',count: { $sum: 1 }}},{ $sort: { count: -1 }},{ $limit: 1}])

24. Find the movies released in the year with the highest average IMDb rating from the 'movies' collection in MongoDB.

A) db.movies.aggregate([{$group: {\_id: "$year",averageRating: { $avg: "$imdb.rating" },movies: { $push: "$$ROOT" }}},{ $sort: { averageRating: -1 }},{ $limit: 1},{$unwind: "$movies"},{$replaceRoot: { newRoot: "$movies" }}])

25. Find the top 10 directors with the most movies from the 'movies' collection in MongoDB.

A) db.movies.aggregate([{$group: {\_id: "$directors",movieCount: { $sum: 1 }}},{$sort: { movieCount: -1 }},{$limit: 10}])

26. Write a query in MongoDB to find the average IMDb rating for movies with different ratings (e.g., 'PG', 'R', 'G') from the 'movies' collection.

A) db.movies.aggregate([{ $group: {\_id: "$rated",averageRating: { $avg: "$imdb.rating" }}}])

27. Write a query in MongoDB to find the oldest movie with an award win from the 'movies' collection.

A) db.movies.find({ "awards.wins": { $gt: 0 } }).sort({ year: 1 }).limit(1)

28. Write a query in MongoDB to find the movie with the highest IMDb rating and viewer rating on Tomatoes from the 'movies' collection.

A) db.movies.aggregate([{$project: {\_id: 1,title: 1,imdbRating: "$imdb.rating",tomatoViewerRating: "$tomatoes.viewer.rating"}},{$sort: {imdbRating: -1,tomatoViewerRating: -1}},{$limit: 1}])