

EXPERIMENT 20

```
> db.movies.insertOne({
  "_id": ObjectId("573a1390f29313caabcd42e8"),
  "plot": "A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.",
  "genres": ["Short", "Western"],
  "runtime": 11,
  "cast": [
    "A.C. Abadie",
    "Gilbert M. 'Broncho Billy' Anderson",
    "George Barnes",
    "Justus D. Barnes"
  ]
})
< {
  acknowledged: true,
  insertedId: ObjectId('573a1390f29313caabcd42e8')
}
DBMS> |
```

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

```
> db.movies.find({ "year": 1893 })
<
DBMS> |
```

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

```
> db.movies.find({ "runtime": { $gt: 120 } })
<
DBMS> |
```

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

```
> db.movies.find({ "genres": "Short" })
< {
  _id: ObjectId('573a1390f29313caabcd42e8'),
  plot: 'A group of bandits stage a brazen train hold-up, only to find a determined posse hot on their heels.',
  genres: [
    'Short',
    'Western'
  ],
  runtime: 11,
  cast: [
    'A.C. Abadie',
    'Gilbert M. 'Broncho Billy' Anderson',
    'George Barnes',
    'Justus D. Barnes'
  ]
}
DBMS> |
```

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

```
> db.movies.find({ "director": "William K.L. Dickson" })  
<  
DBMS > |
```

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

```
> db.movies.find({ "country": "USA" })  
<  
DBMS >
```

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

```
> db.movies.find({ "rating": "UNRATED" })  
<  
DBMS >
```

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

```
> db.movies.find({ "imdb_votes": { $gt: 1000 } })  
<  
DBMS >
```

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

```
> db.movies.find({ "imdb_rating": { $gt: 7 } })  
<  
DBMS > |
```

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

```
> db.movies.find({ "tomato_viewer_rating": { $gt: 4 } })  
<  
DBMS > |
```

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

```
> db.movies.find({ "awards": { $exists: true, $ne: null } })  
<  
DBMS > |
```

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.

```
> db.movies.find(
  { "awards.nominations": { $gt: 0 } }, // Find movies with at least one nomination
  {
    "title": 1,
    "languages": 1,
    "released": 1,
    "directors": 1,
    "writers": 1,
    "awards": 1,
    "year": 1,
    "genres": 1,
    "runtime": 1,
    "cast": 1,
    "countries": 1,
    "_id": 0 // Exclude the default _id field
  }
)
<
DBMS >
```

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".

```
> db.movies.find(
  { "cast": "Charles Kayser" }, // Filter movies with Charles Kayser in the cast
  {
    "title": 1,
    "languages": 1,
    "released": 1,
    "directors": 1,
    "writers": 1,
    "awards": 1,
    "year": 1,
    "genres": 1,
    "runtime": 1,
    "cast": 1,
    "countries": 1,
    "_id": 0 // Exclude the default _id field
  }
)
<
DBMS >
```

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

```
> db.movies.find(
  { "released": ISODate("1893-05-09") }, // Filter movies released on May 9, 1893
  {
    "title": 1,
    "languages": 1,
    "released": 1,
    "directors": 1,
    "writers": 1,
    "countries": 1,
    "_id": 0 // Exclude the default _id field
  }
)
<
DBMS >
```

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.

```
> db.movies.find(
  { "title": { $regex: /scene/i } }, // Filter movies with "scene" in the title
  {
    "title": 1,
    "languages": 1,
    "released": 1,
    "directors": 1,
    "writers": 1,
    "countries": 1,
    "_id": 0 // Exclude the default _id field
  }
)
<
DBMS >
```