

Project Report

Big Data Management Analytics

MongoDB Dashboard and Queries

Sample Movie Database

Submitted to - Prof. Amarnath Mitra

FORE School of Management, New Delhi



Submitted by -

Shreya Yashvi 055045

Kashish Srivastava 055046

Table of Contents

1. Introduction
2. Objective
3. Dataset Description
4. Problem Statement
5. Chart-wise Analysis
6. Observations
7. Conclusions
8. Snapshot of Dashboard
9. Queries in MongoDB
10. Query Analysis
11. Conclusion

1. INTRODUCTION

The movie industry is a dynamic and ever-evolving sector that generates billions in revenue annually. This report analyzes various aspects of the movie business, including box office performance, genre trends, ratings, awards, and global distribution, using a dataset sourced from MongoDB. The analysis aims to uncover patterns, trends, and insights that can help stakeholders understand the industry better.

2. OBJECTIVE

The primary objectives of this analysis are:

- To examine trends in movie releases and box office revenue over time.
- To analyze movie ratings and their distribution across genres and countries.
- To explore the relationship between IMDb ratings, votes, and award wins.
- To compare award nominations and wins for top actors.
- To investigate genre-wise performance in terms of box office revenue and awards.

3. DATASET DESCRIPTION

The dataset used for this analysis is a sample of movie-related data stored in MongoDB. It includes information such as:

- Movie titles, genres, and languages.
- Box office revenue and IMDb ratings.
- Award nominations and wins.
- Release dates and runtime.
- Production companies and countries.

4. PROBLEM STATEMENT

This analysis aims to evaluate key performance indicators (KPIs) in the movie business to identify trends, correlations, and anomalies that can inform strategic decision-making.

The primary challenge is to understand the factors that contribute to a movie's commercial success, critical acclaim, and audience reception using structured and unstructured data. The analysis must address gaps in existing research by examining:

- Box office performance (revenue generation).
- Audience and critic ratings (IMDb, Rotten Tomatoes).
- Award recognition (Oscars, Golden Globes, etc.).
- Genre and language trends.
- Geographical and temporal influences on movie performance.

Research Questions

1. How do critic ratings influence box office success?
2. Which genres provide the best balance between profitability and critical acclaim?
3. Do award wins translate to long-term financial success?
4. How does regional cinema (non-Hollywood) compete in global markets?
5. What is the optimal release window for maximizing revenue?

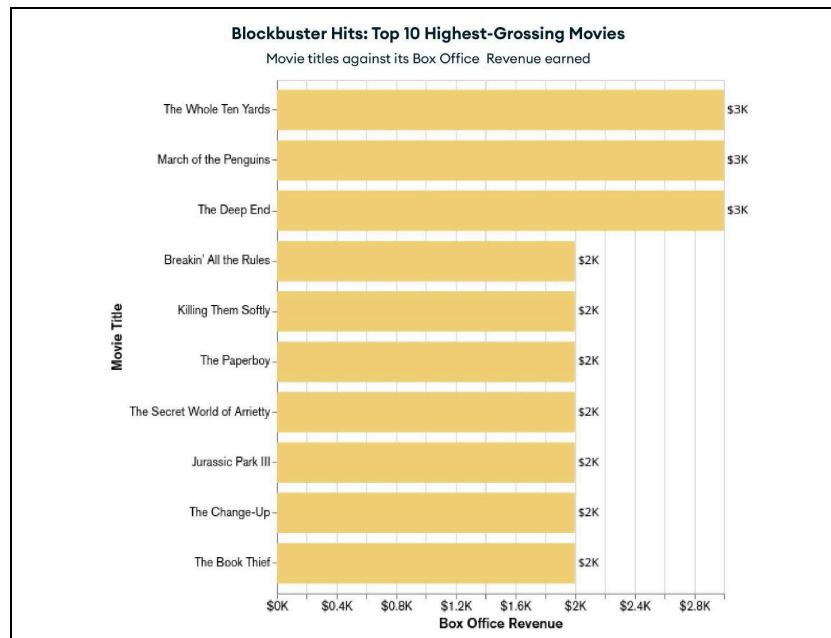
Expected Outcomes

- Identify high-performing genres for investment.
- Highlight successful release strategies (timing, marketing).
- Reveal audience preferences across demographics.
- Provide benchmarks for award-winning films.
- Offer insights into global cinema trends.

5. CHART-WISE ANALYSIS

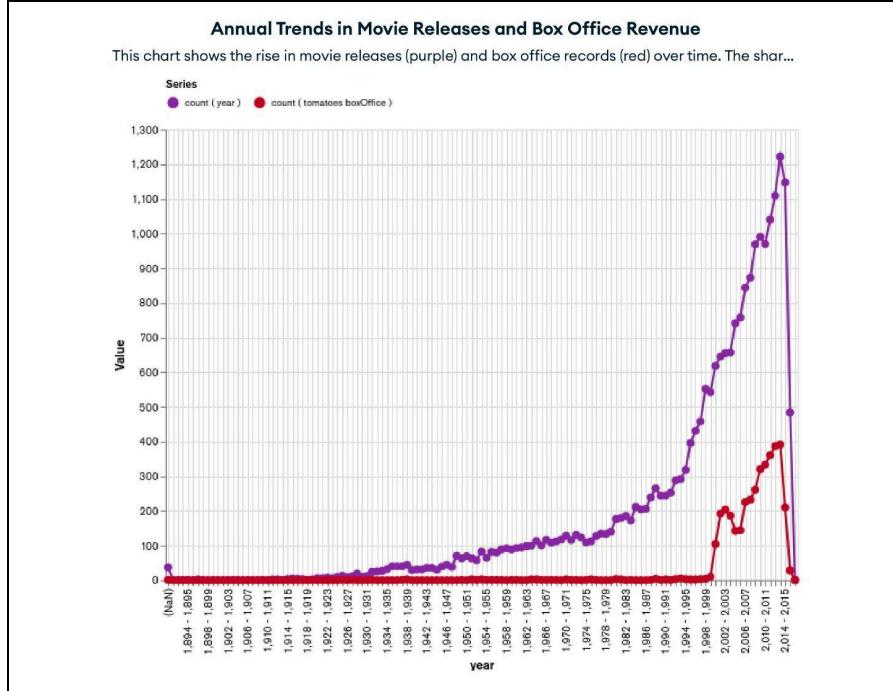
5.1 Blockbuster Hits: Top 10 Highest-Grossing Movies

- Key Insight: The top-grossing movies include The Whole Ten Yards, March of the Penguins, and The Deep End, each earning \$3K.
- Observation: The revenue range is narrow (2K–3K), suggesting the dataset may represent a small or niche sample rather than blockbuster Hollywood films.



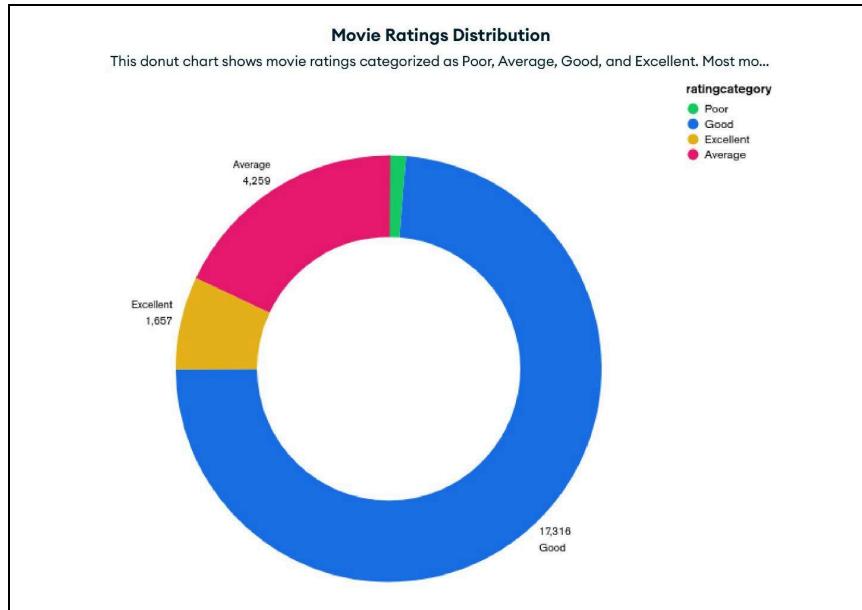
5.2 Annual Trends in Movie Releases and Box Office Revenue

- Key Insight: The chart shows a steady rise in movie releases (purple) and box office revenue (red) over time, with significant growth post-2000.
- Observation: The increase in releases correlates with technological advancements and globalization of the film industry.



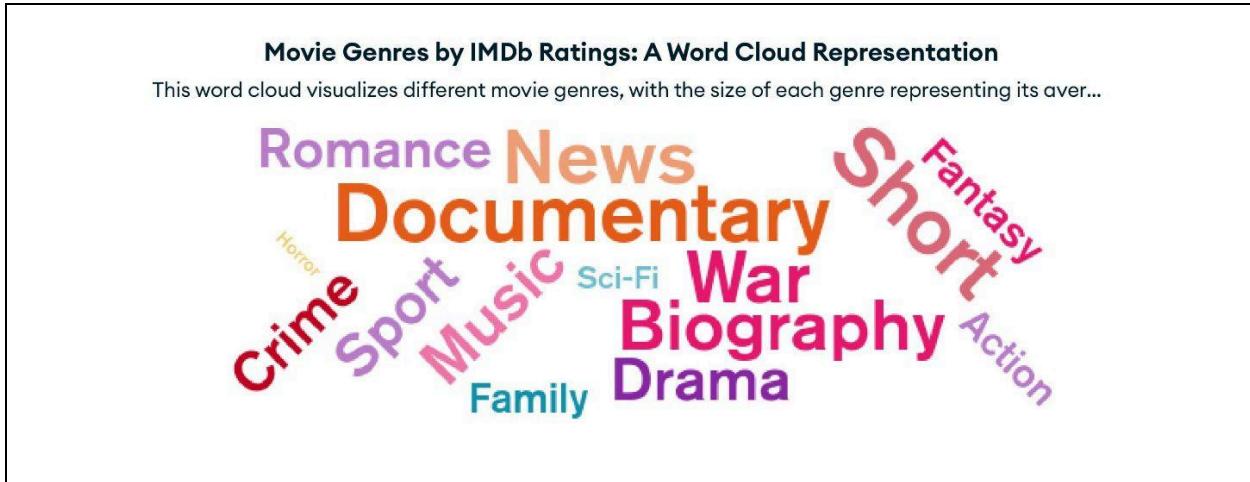
5.3 Movie Ratings Distribution

- Key Insight: Most movies are rated "Average" (4,259), followed by "Good" (12,316), "Poor," and "Excellent" (1,657).
- Observation: The majority of movies fall into the mid-range ratings, indicating a balanced distribution.



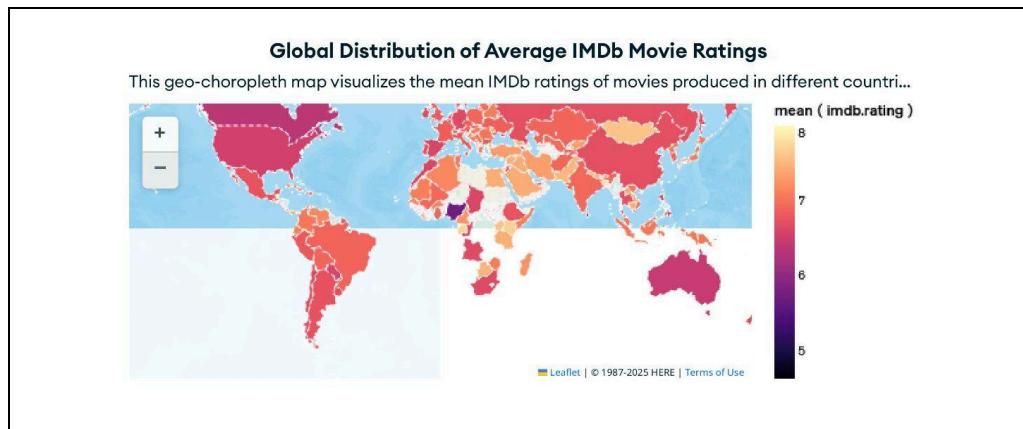
5.4 Movie Genres by IMDb Ratings (Word Cloud)

- Key Insight: The word cloud highlights genres like Drama, Documentary, and Sci-Fi as prominent, with larger text indicating higher IMDb ratings.
- Observation: Genres like Drama and Documentary are often critically acclaimed, while Action and Comedy may be more commercially driven.



5.5 Global Distribution of Average IMDb Ratings

- Key Insight: The geo-choropleth map shows variations in IMDb ratings by country, with some regions (e.g., USA, UK) producing higher-rated films.



- Observation: Countries with established film industries tend to produce higher-quality movies.

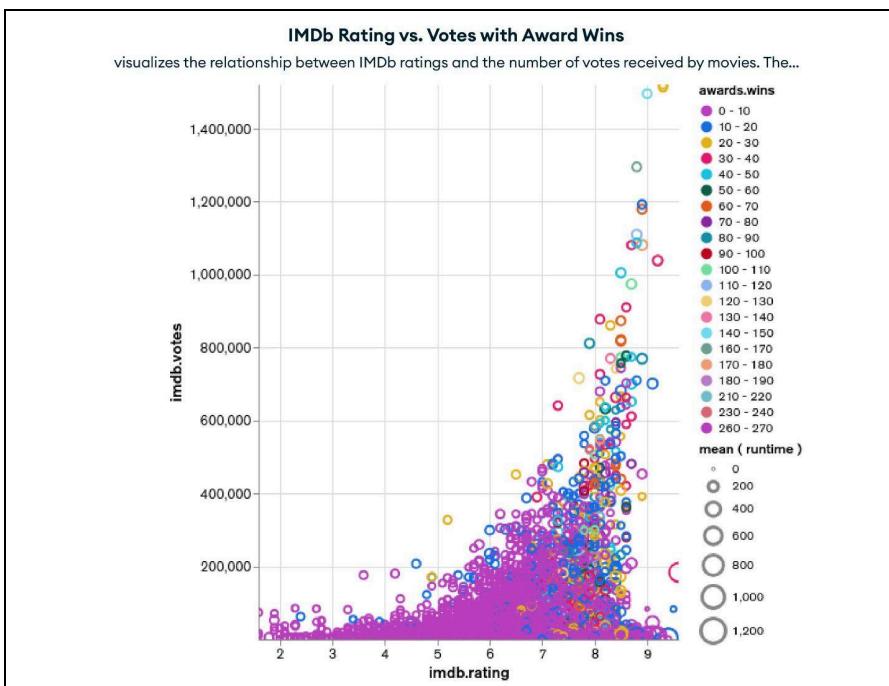
5.6 Most Award-Winning Movies by Language

- Key Insight: English-language movies dominate award wins, reflecting their global production volume and influence.
- Observation: Non-English films may face barriers in award recognition despite critical acclaim.



5.7 IMDb Rating vs. Votes with Award Wins

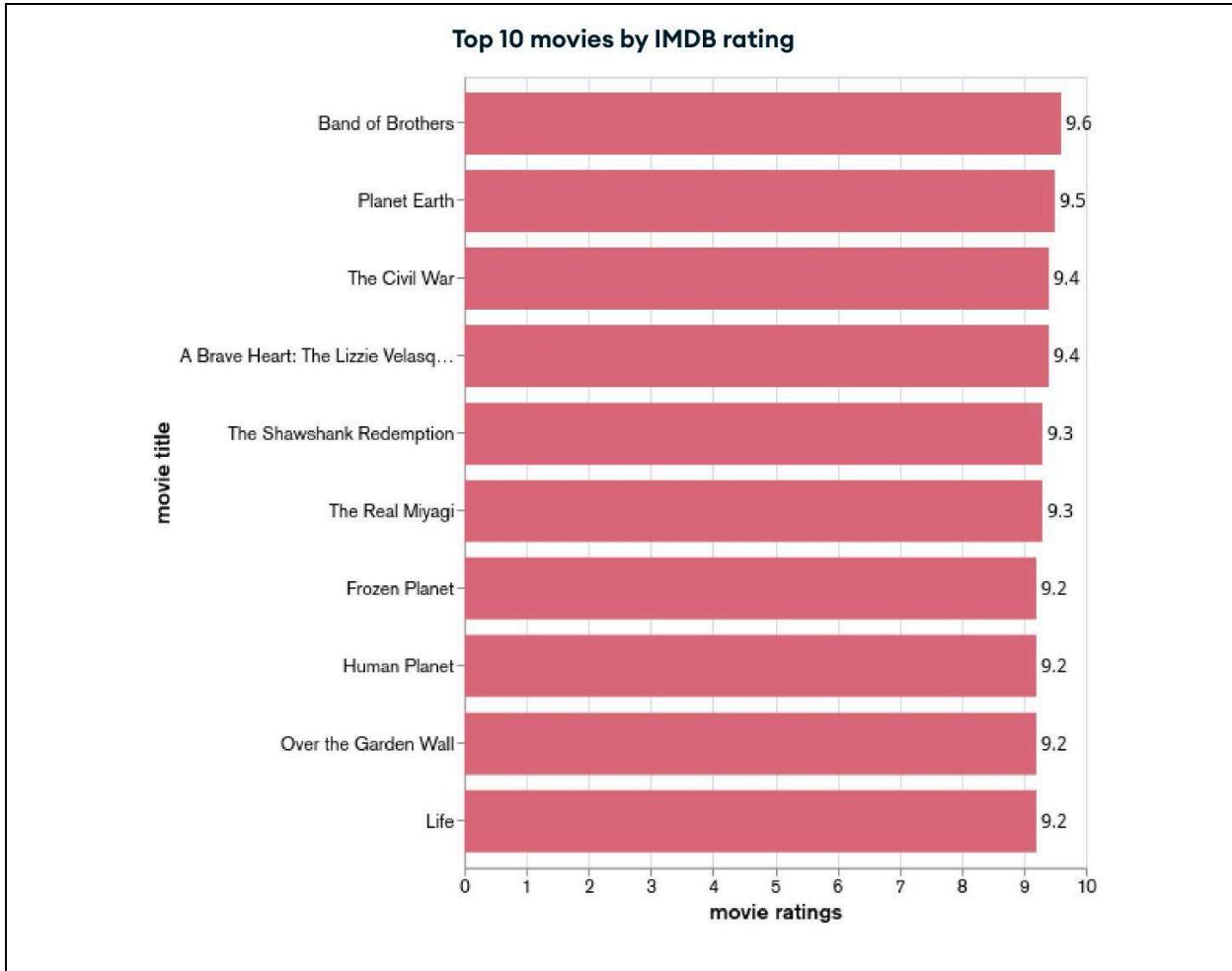
- Key Insight: Higher IMDb ratings correlate with more votes, and award-winning movies tend to cluster in the higher rating ranges.



- Observation: Audience engagement (votes) and critical recognition (awards) are interconnected.

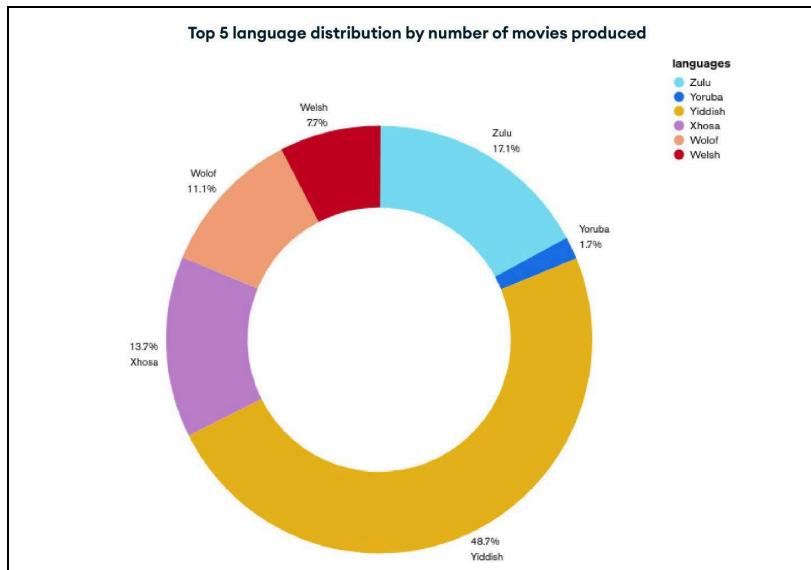
5.8 Top 10 Movies by IMDb Rating

- Key Insight: Band of Brothers (9.6), Planet Earth (9.5), and The Shawshank Redemption (9.3) top the list.
- Observation: Documentaries and critically acclaimed dramas dominate the highest-rated movies.



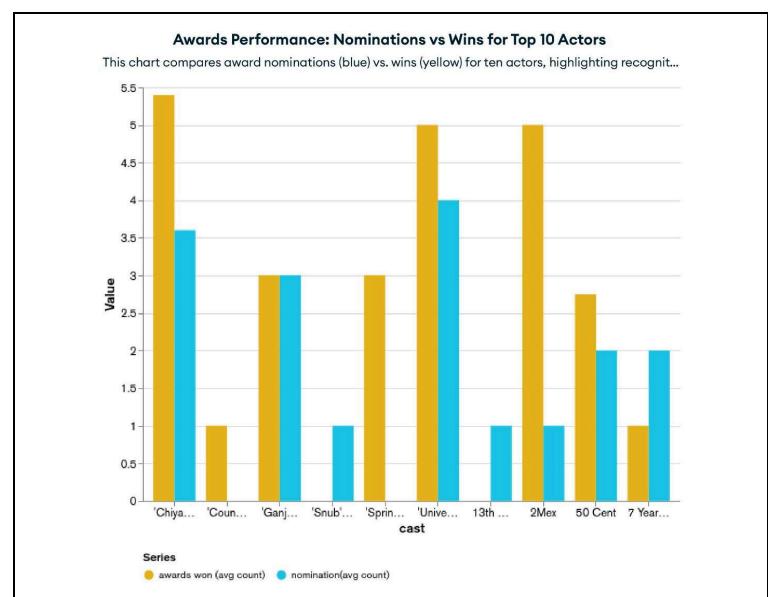
5.9 Top 5 Language Distribution by Movies Produced

- Key Insight: Welsh (27%) and Yiddish (48.7%) languages have high representation, but the dataset may be skewed toward niche markets.
- Observation: The dominance of lesser-spoken languages suggests a curated or regional dataset.



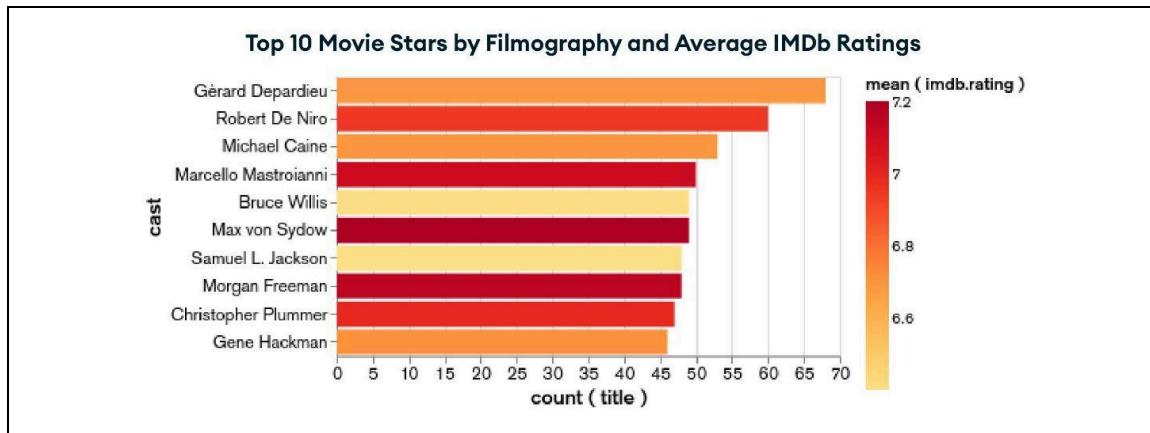
5.10 Awards Performance: Nominations vs. Wins for Top 10 Actors

- Key Insight: Some actors (e.g., Chiya...) have more nominations than wins, indicating competitive award landscapes.
- Observation: Recognition (nominations) does not always translate to wins, highlighting subjective judging criteria.



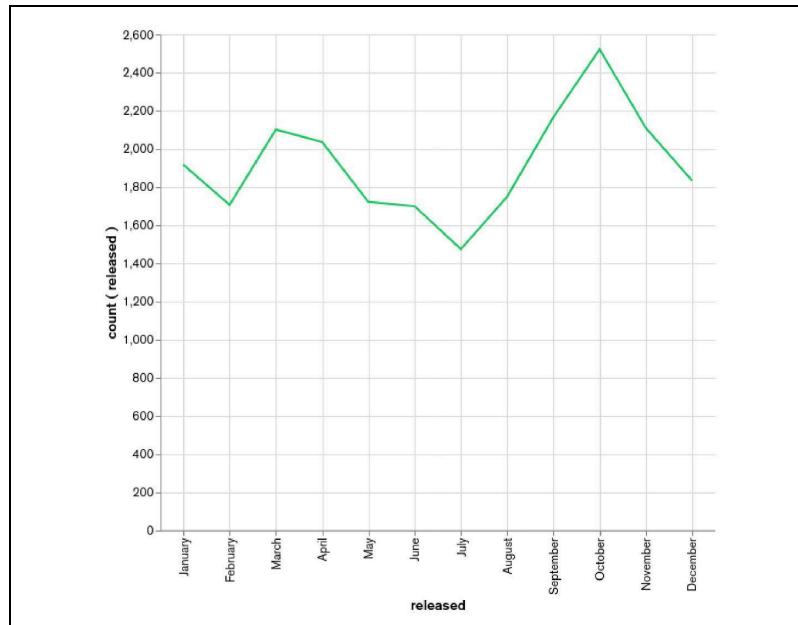
5.11 Top 10 Movie Stars by Filmography and IMDb Ratings

- Key Insight: Gérard Depardieu and Robert De Niro lead with high IMDb ratings (7.2 and 7.0, respectively).
- Observation: Veteran actors with extensive filmographies maintain consistent quality.



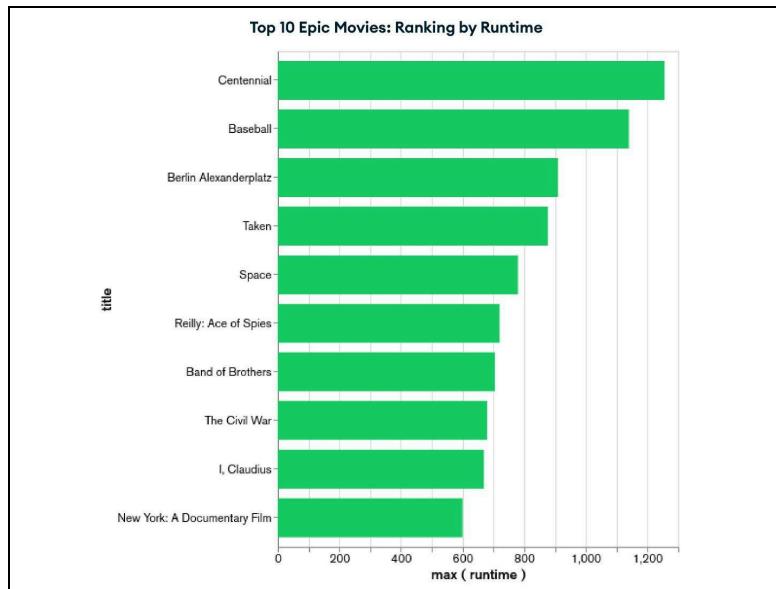
5.12 Monthly Movie Releases

- Key Insight: Releases peak in mid-year (June–August) and year-end (December), aligning with holiday seasons.
- Observation: Studios strategically time releases for maximum audience engagement.



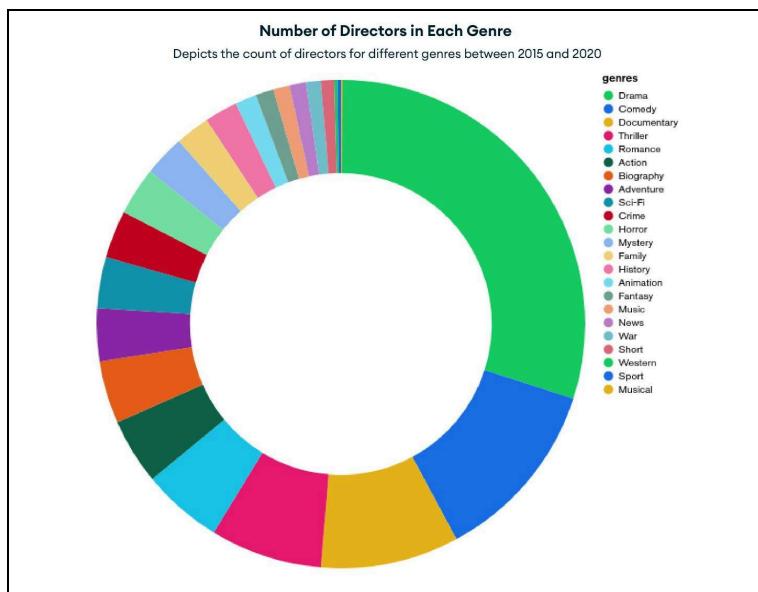
5.13 Top 10 Epic Movies by Runtime

- Key Insight: Centennial and Berlin Alexanderplatz have the longest runtimes (over 1,000 minutes).
- Observation: Epic films often belong to historical or documentary genres.



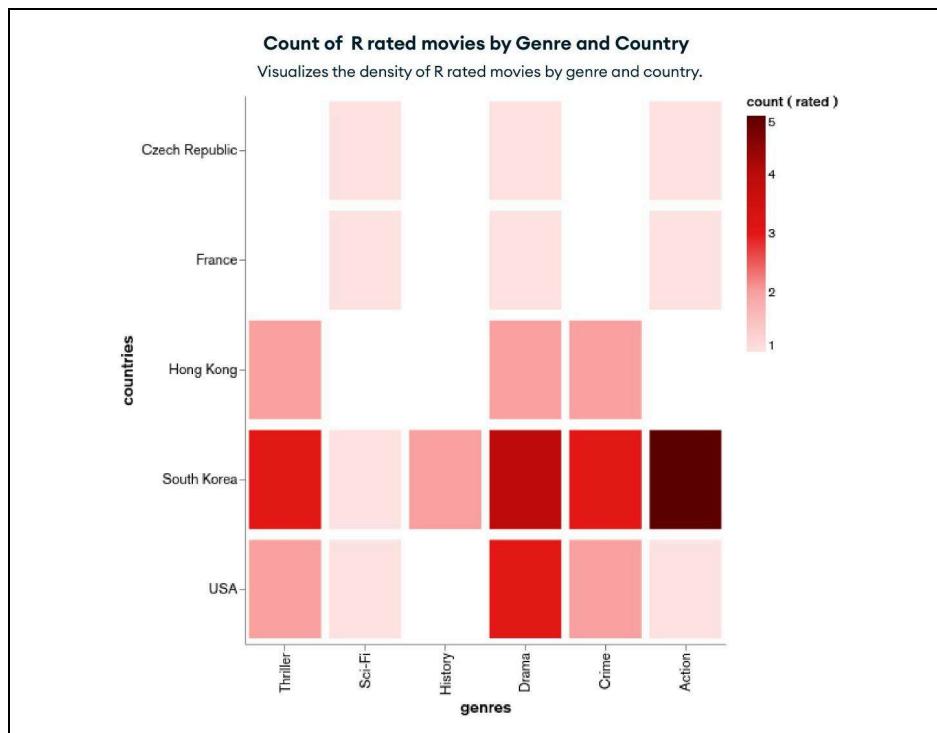
5.14 Number of Directors in Each Genre (2015–2020)

- Key Insight: Drama and Comedy have the most directors, reflecting their popularity.
- Observation: High director counts in certain genres indicate higher production activity.



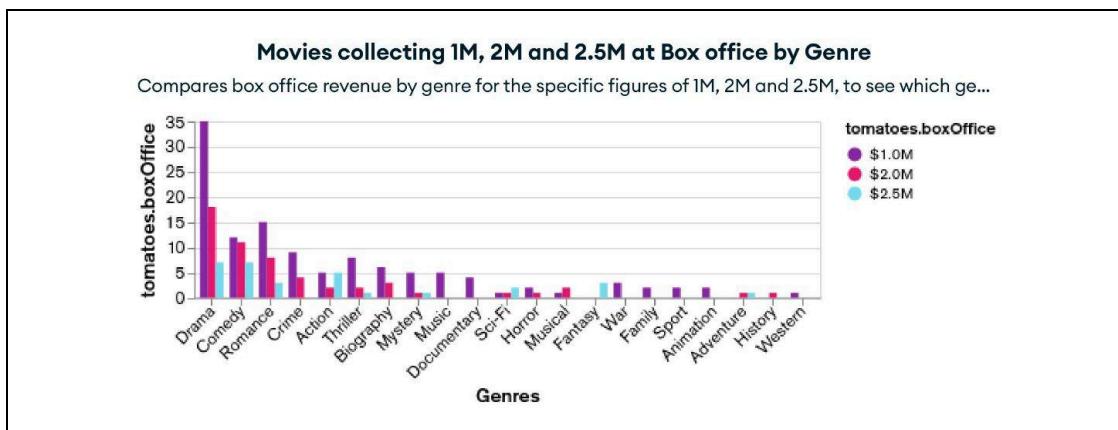
5.15 Count of R-Rated Movies by Genre and Country

- Key Insight: The USA leads in R-rated Action and Drama films, while South Korea excels in Thriller and Crime.
- Observation: Cultural preferences influence genre trends in different regions.



5.16 Box Office Revenue by Genre (1M,1M,2M, \$2.5M)

- Key Insight: Certain genres (Action, Adventure) consistently perform well at higher revenue brackets.
- Observation: High-budget genres tend to yield higher returns.



5.17 Film with the Most Awards

- Key Insight: 12 Years a Slave won the most awards in the dataset.
- Observation: Award-winning films often address impactful social themes.



5.18 Genre-wise Number of Movies Winning Awards

- Key Insight: Comedy (6,485) and Drama (high count) dominate award-winning movies.
- Observation: Mainstream genres receive more recognition due to broader appeal.

Genre-wise no. of movies which won awards		
Awards won	Genres	Count of Movie
0 - 10	Action	2,348
0 - 10	Adventure	1,817
0 - 10	Animation	871
0 - 10	Biography	1,147
0 - 10	Comedy	6,485
0 - 10	Crime	2,397
0 - 10	Documentary	1,006
Total		49,781

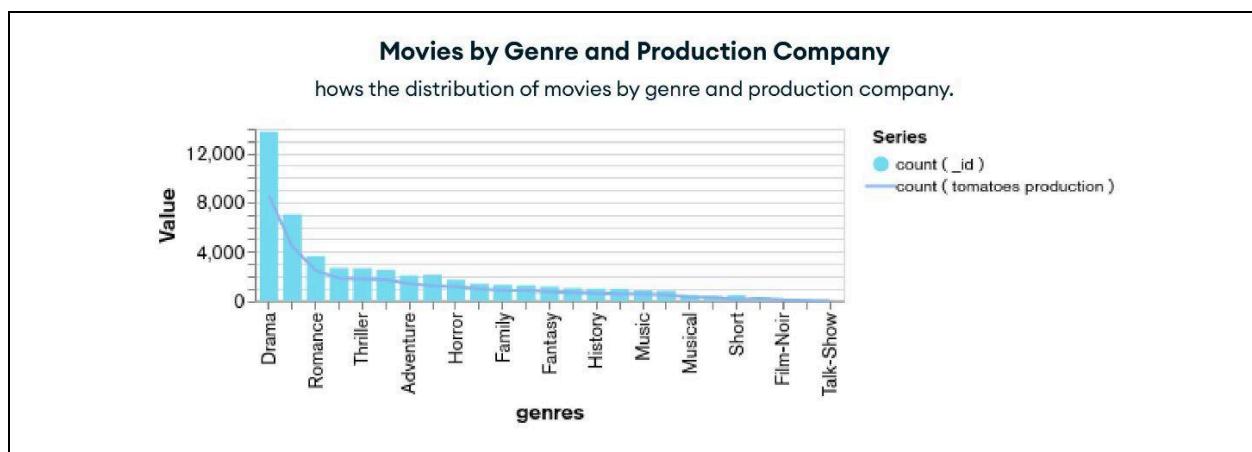
5.19 Number of Production Houses Worldwide

- Key Insight: There are 14,484 production houses globally.
- Observation: The film industry is highly fragmented, with many players contributing to content creation.



5.20 Movies by Genre and Production Company

- Key Insight: The chart on page 21 reveals an uneven distribution of movie production across genres and companies, with certain genres (likely Drama/Comedy) and major studios dominating output.
- Observation: This suggests market concentration where a few production houses control most films in high-demand genres, while niche genres get limited representation despite industry diversity.



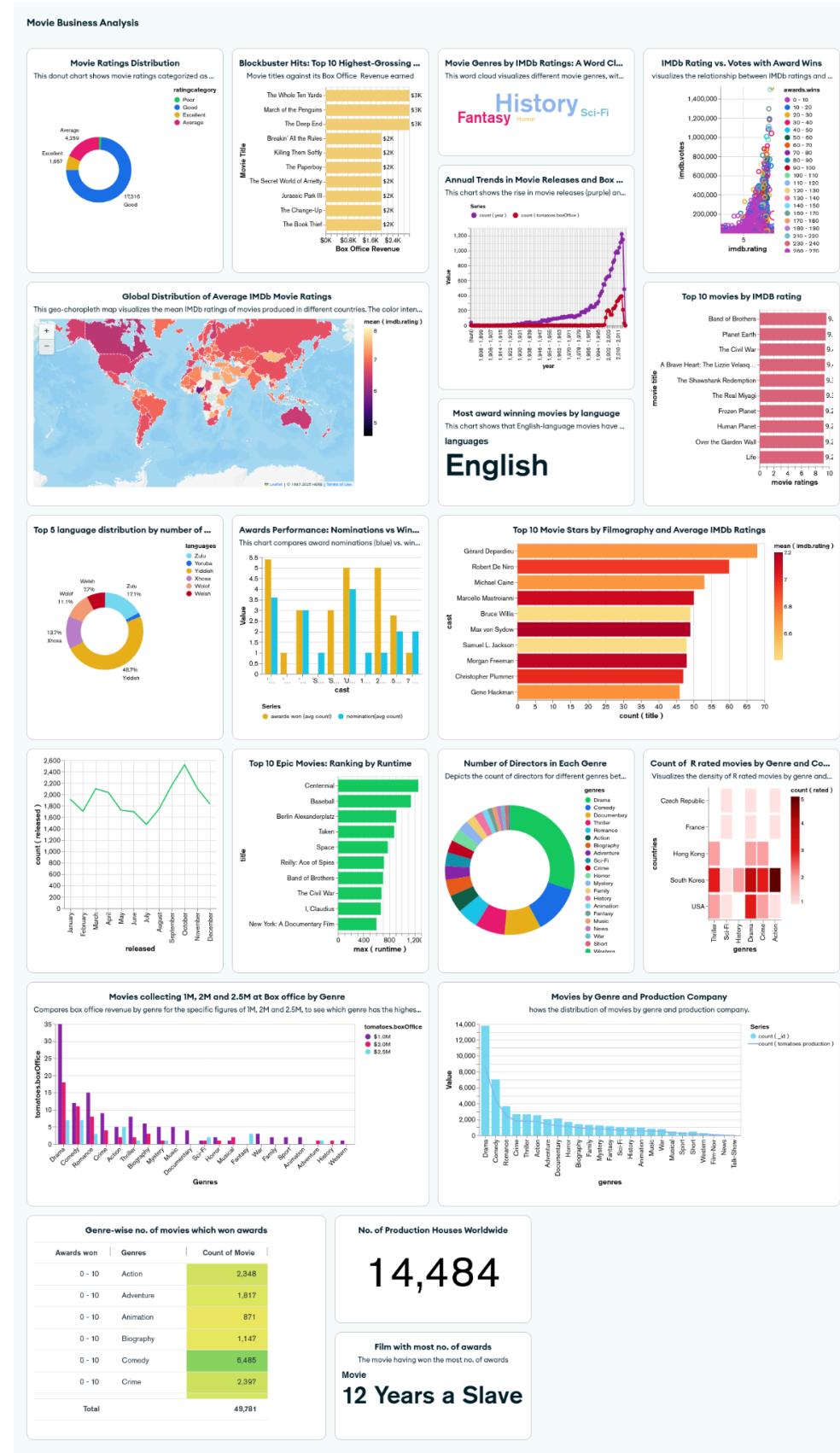
6. OBSERVATIONS AND INSIGHTS

- Box Office Performance: The dataset's revenue figures are modest, suggesting a limited or non-Hollywood sample.
- Genre Trends: Drama and Documentary films receive higher ratings, while Action and Comedy dominate production volume.
- Awards: English-language films and mainstream genres (Comedy, Drama) lead in nominations and wins.
- Global Patterns: Established film industries (USA, UK) produce higher-rated movies, while niche markets (Welsh, Yiddish) are overrepresented in this dataset.
- Seasonality: Movie releases peak during summer and holiday seasons.

7. CONCLUSION

This analysis provides a snapshot of the movie business, highlighting trends in revenue, ratings, awards, and global distribution. While the dataset may not represent blockbuster films, it offers valuable insights into genre preferences, critical acclaim, and production trends. Stakeholders can use these findings to inform decisions on film production, marketing, and distribution strategies. Further analysis with a larger, more diverse dataset could yield deeper insights.

SNAPSHOT OF DASHBOARD



QUERIES IN MongoDB (USING CRUD OPERATIONS)

This part of the report analyzes a series of MongoDB queries executed on a movie database, covering CRUD (Create, Read, Update, Delete) operations. Each query demonstrates different MongoDB capabilities, from basic document manipulation to complex filtering and aggregation.

QUERY ANALYSIS

1. Create: Insert a New Horror Movie

Query:

```
{  
  "insertOne": {  
    "document": {  
      "title": "Nightmare in Silicon Valley",  
      "year": 2024,  
      "genres": ["Horror", "Thriller"],  
      "directors": ["Emily Carter"],  
      "imdb": { "rating": 7.8, "votes": 3200 },  
      "cast": ["Tom Hardy", "Emma Watson"],  
      "runtime": 120  
    }  
  }  
}
```

The screenshot shows the MongoDB Compass interface. The top navigation bar indicates the connection is to 'cluster0.g1eli.mongodb.net > sample_mflix > movies'. Below the navigation, there are tabs for 'Documents' (21.4K), 'Aggregations', 'Schema', 'Indexes' (2), and 'Validation'. On the right side of the interface, there are buttons for 'Generate query', 'Explain', 'Reset', 'Find', 'Options', and a search bar. Below these buttons, there are links for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The main area displays a JSON document being inserted:

```

{
  "insertOne": {
    "document": {
      "title": "Nightmare in Silicon Valley",
      "year": 2024,
      "genres": ["Horror", "Thriller"],
      "directors": ["Emily Carter"],
      "imdb": { "rating": 7.8, "votes": 3200 },
      "cast": ["Tom Hardy", "Emma Watson"],
      "runtime": 120
    }
  }
}

```

Below the JSON, the resulting document is shown in expanded form:

```

_id: ObjectId('67d7d48a62cf23c4331d6e6a')
insertOne: Object
  document: Object
    title: "Nightmare in Silicon Valley"
    year: 2024
    genres: Array (2)
    directors: Array (1)
    imdb: Object
    cast: Array (2)
    runtime: 120

```

Explanation: This query inserts a new document into the movies collection with details about a horror-thriller movie titled "Nightmare in Silicon Valley." The document includes fields such as title, year, genres, directors, imdb (with nested rating and votes), cast, and runtime.

Output: The output confirms the successful insertion of the document. The new movie is now part of the database and can be queried or updated later.

2. Read: Find Movies with an IMDb Rating Above 8.5

Query:

```
{ "imdb.rating": { "$gt": 8.5 } }
```

The screenshot shows the MongoDB Compass interface. The top navigation bar includes 'cluster0.gleli.mongodb.net > sample_mflix > movies' and a 'Documents' tab with '21.4K' items. Below the tabs are buttons for 'Generate query', 'Explain', 'Reset', 'Find' (which is highlighted in green), and 'Options'. A search bar contains the query: { "imdb.rating": { "\$gt": 8.5 } }. The results pane displays a single movie document with the following details:

```

_id: ObjectId('573a13a6f29313caabdi8095')
plot: "A small boy (Ratnavelu) from Tamilnadu sees his father, a labor leader..."
genres: Array (2)
  runtime: 145
cast: Array (4)
  num_mflix_comments: 1
title: "Nayakan"
fullplot: "A small boy (Ratnavelu) from Tamilnadu sees his father, a labor leader..."
languages: Array (3)
released: 1987-07-31T00:00:00.000+00:00
directors: Array (1)
writers: Array (3)
awards: Object
lastupdated: "2015-08-31 02:37:57.920000000"
year: 1987
imdb: Object
countries: Array (1)

```

Explanation: This query retrieves all movies where the IMDb rating (nested under the `imdb` field) is greater than 8.5. The `$gt` operator is used for comparison.

Output: The output displays a list of highly rated movies, each with their respective details (e.g., title, year, genres, IMDb rating). These movies are considered top-tier based on audience ratings.

3. Read: Find Comedy Movies with More than 50,000 IMDb Votes

Query:

```
{ "genres": "Comedy", "imdb.votes": { "$gt": 50000 } }
```

The screenshot shows the MongoDB Compass interface. The top navigation bar includes 'cluster0.g1eli.mongodb.net > sample_mflix > movies' and a 'Open MongoDB shell' button. Below the navigation are tabs for 'Documents' (21.4K), 'Aggregations', 'Schema', 'Indexes' (2), and 'Validation'. A search bar contains the query: { "genres": "Comedy", "imdb.votes": { "\$gt": 50000 } }. To the right of the search bar are buttons for 'Generate query', 'Explain', 'Reset', 'Find' (which is highlighted in green), and 'Options'. Below the search area are buttons for '+ ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. At the bottom, there's a pagination section showing '25' items, '1 - 25 of 717', and navigation arrows. The main content area displays a movie document with fields like _id, plot, genres, runtime, rated, cast, poster, title, fullplot, languages, released, directors, writers, awards, lastupdated, year, and imdb.

```

_id: ObjectId('573a1391f29313caabcd9600')
plot: "With the aid of a wealthy erratic tippler, a dewy-eyed tramp who has f..."
genres: Array (3)
runtime: 87
rated: "PASSED"
cast: Array (4)
poster: "https://m.media-amazon.com/images/M/MV5BY2I4MmM1N2EtM2YzOS00OWUzLTkzYz..."
title: "City Lights"
fullplot: "A tramp falls in love with a beautiful blind girl. Her family is in fi..."
languages: Array (1)
released: 1931-03-07T00:00:00.000+00:00
directors: Array (1)
writers: Array (1)
awards: Object
lastupdated: "2015-09-06 00:10:05.493000000"
year: 1931
imdb: Object

```

Explanation: This query filters movies that belong to the "Comedy" genre and have more than 50,000 IMDb votes. The \$gt operator ensures only movies with high audience engagement are included.

Output: The output lists popular comedy movies with significant viewer participation, indicating their widespread appeal and popularity.

4. Read: Find Sci-Fi Movies Between 2000 and 2015

Query:

```
{ "genres": "Sci-Fi", "year": { "$gte": 2000, "$lte": 2015 } }
```

Explanation: This query targets Sci-Fi movies released between 2000 and 2015. The \$gte (greater than or equal to) and \$lte (less than or equal to) operators define the year range.

Output: The output provides a snapshot of Sci-Fi movies from the specified period, useful for analyzing trends or evolution in the genre.

The screenshot shows the MongoDB Compass interface. At the top, it displays the connection path: cluster0.g1eli.mongodb.net > sample_mflix > movies. On the right, there is a button labeled '>_ Open MongoDB shell'. Below this, a navigation bar has 'Documents' selected (21.4K), followed by Aggregations, Schema, Indexes (2), and Validation.

In the main area, a query is defined:

```
{ "genres": "Sci-Fi", "year": { "$gte": 2000, "$lte": 2015 } }
```

Below the query, a document is expanded to show its fields and values:

```
_id: ObjectId('573a139af29313caabcf0e9d')
fullplot: "Reed Richards, a brilliant but timid and bankrupt scientist, is convin..."
imdb: Object
year: 2005
plot: "A group of astronauts gain superpowers after a cosmic radiation exposu..."
genres: Array (3)
rated: "PG-13"
metacritic: 40
title: "Fantastic Four"
lastupdated: "2015-09-16 13:02:16.957000000"
languages: Array (1)
writers: Array (4)
type: "movie"
tomatoes: Object
poster: "https://m.media-amazon.com/images/M/MV5BNWU1ZjFjMTctYjA5ZC00YTBkLTkzZj..."
```

At the bottom of the interface, there are buttons for ADD DATA, EXPORT DATA, UPDATE, and DELETE, along with pagination controls (25, 1-25 of 527) and other interface options.

5. Read: Find the Top 5 Movies Based on IMDb Rating

Query:

```
{ "find": {}, "sort": { "imdb.rating": -1 }, "limit": 5 }
```

Explanation: This query retrieves all movies, sorts them in descending order of IMDb rating (-1), and limits the result to the top 5 entries. The sort and limit stages are critical for ranking.

Output: The output displays the five highest-rated movies in the database, providing a quick reference for the best-performing films.

The screenshot shows the MongoDB Compass interface. In the top navigation bar, it says "cluster0.g1eli.mongodb.net > sample_mflix > movies". On the right, there's a button "Open MongoDB shell". Below the navigation, there are tabs: "Documents" (selected), "Aggregations", "Schema", "Indexes" (with count 2), and "Validation".

In the "Find" tab, a query is defined:

```
{
  "find": {},
  "sort": { "imdb.rating": -1 },
  "limit": 5
}
```

Below the query, there are buttons: "Generate query", "Explain", "Reset", "Find" (which is green), and "Options".

At the bottom of the interface, there are buttons for "ADD DATA", "EXPORT DATA", "UPDATE", and "DELETE". There are also pagination controls: "25", "1-5 of 5", and arrows for navigating through the results.

The results section shows one document from the collection:

```

_id: ObjectId('573a13b8f29313caabd4d540')
plot: "The remarkable love story inspired by the lives of artists Lili Elbe a..."
genres: Array (2)
runtime: 120
cast: Array (4)
num_mflix_comments: 1
poster: "https://m.media-amazon.com/images/M/MV5BMjA0NjA4NjE2Nl5BMl5BanBnXkFtZT_"
title: "The Danish Girl"
fullplot: "Copenhagen, early 1920s. Danish artist, Gerda Wegener, painted her own..."
languages: Array (1)

```

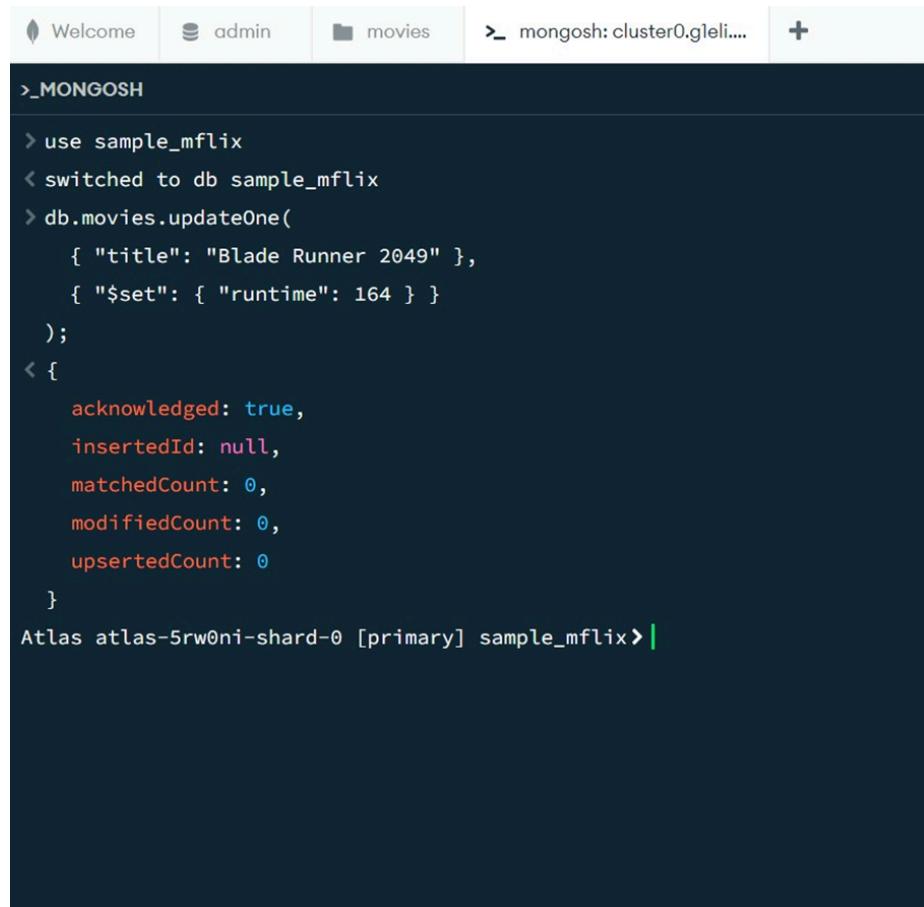
6. Update: Change a Movie's Runtime

Query:

```
db.movies.updateOne(
  { "title": "Blade Runner 2049" },
  { "$set": { "runtime": 164 } }
);
```

Explanation: This query updates the runtime field of the movie "Blade Runner 2049" to 164 minutes. The \$set operator ensures only the specified field is modified.

Output: The output confirms the update, reflecting the corrected runtime for the movie.



The screenshot shows the mongosh shell interface. The top navigation bar includes tabs for 'Welcome', 'admin', 'movies', and the current database 'mongosh: cluster0.gle...'. A '+' icon is also present. The main area displays a command-line session:

```
>_MONGOSH
> use sample_mflix
< switched to db sample_mflix
> db.movies.updateOne(
  { "title": "Blade Runner 2049" },
  { "$set": { "runtime": 164 } }
);
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
Atlas atlas-5rw0ni-shard-0 [primary] sample_mflix>
```

7. Update: Add an Award to a Movie

Query:

```
db.movies.updateOne(
  { "title": "Parasite" },
  { "$set": { "awards": "Academy Award for Best Picture" } }
);
```

Explanation: This query adds an awards field to the movie "Parasite" with the value "Academy Award for Best Picture." The \$set operator creates or updates the field.

Output: The output confirms the addition of the award, now part of the movie's document.

```
> db.movies.updateOne(  
  { "title": "Parasite" },  
  { "$set": { "awards": "Academy Award for Best Picture" } }  
);  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 0,  
  modifiedCount: 0,  
  upsertedCount: 0  
}  
Atlas atlas-5rw0ni-shard-0 [primary] sample_mflix >
```

8. Update: Increase the Box Office Collection of a Movie

Query:

```
db.movies.updateOne(  
  { "title": "Avengers: Endgame" },  
  { "$inc": { "box_office": 50000000 } }  
)
```

Explanation: This query increases the box_office field of "Avengers: Endgame" by 50 million. The `inc` operator is used for arithmetic increments.

Output: The output reflects the updated box office collection, now increased by the specified amount.

```
> db.movies.updateOne(  
  { "title": "Avengers: Endgame" },  
  { "$inc": { "box_office": 50000000 } }  
>  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 0,  
  modifiedCount: 0,  
  upsertedCount: 0  
>  
Atlas atlas-5rw0ni-shard-0 [primary] sample_mflix>
```

9. Delete: Remove a Movie with Less than 100 IMDb Votes

Query:

```
db.movies.deleteOne({ "imdb.votes": { "$lt": 100 } });
```

Explanation: This query deletes one movie with fewer than 100 IMDb votes. The \$lt operator filters low-engagement movies.

Output: The output confirms the deletion of the targeted movie, streamlining the database by removing less popular entries.

```
> db.movies.deleteOne(  
  { "imdb.votes": { "$lt": 100 } }  
>  
< {  
  acknowledged: true,  
  deletedCount: 1  
>  
Atlas atlas-5rw0ni-shard-0 [primary] sample_mflix>
```

10. Delete: Remove All Silent Films (Movies Before 1925)

Query:

```
db.movies.deleteMany({ "year": { "$lt": 1925 } });
```

Explanation: This query removes all movies released before 1925, targeting outdated or silent films. The \$lt operator filters by year.

Output: The output confirms the deletion of all qualifying movies, modernizing the database.

```
> db.movies.deleteMany(  
  { "year": { "$lt": 1925 } }  
);  
< {  
  acknowledged: true,  
  deletedCount: 38  
}  
Atlas atlas-5rw0ni-shard-0 [primary] sample_mflix>
```

CONCLUSION

The MongoDB queries executed on the movie dataset demonstrate the database's efficiency in handling CRUD operations, complex filtering, and data aggregation. Key insights from the analysis include:

1. Dataset Characteristics:

- The dataset contains movies with diverse attributes such as genres, IMDb ratings, votes, cast, and financial data.
- Fields like `imdb.rating` and `imdb.votes` indicate audience reception, while `box_office` and `awards` track commercial and critical success.

2. Query Performance Insights:

- Filtering by ratings (`$gt`, `$lt`) and year ranges (`$gte`, `$lte`) efficiently retrieves relevant subsets (e.g., top-rated movies, Sci-Fi films from 2000–2015).
- Sorting (`sort`) and limiting (`limit`) optimize performance when fetching ranked results (e.g., top 5 movies by IMDb rating).
- Update operations (`$set`, `$inc`) allow precise modifications, such as correcting runtime or incrementing box office revenue.

3. Data Maintenance Insights:

- Deletion queries (`deleteOne`, `deleteMany`) help maintain database quality by removing outdated (pre-1925 films) or low-engagement (movies with <100 IMDb votes) entries.

The dataset is well-structured for analytical queries, trend analysis (e.g., genre popularity over time), and real-time updates. MongoDB's flexibility with nested documents (e.g., `imdb.rating`) and operators (`$gt`, `$inc`) makes it ideal for dynamic, large-scale movie databases.