1. **Find Total Revenue by Genre:**

db.movies\_tmdb.aggregate([

{ $unwind: "$genres" },

{ $group: {

\_id: "$genres.name",

TotalRevenue: { $sum: "$revenue" }

}},

{ $sort: { TotalRevenue: -1 }}

]);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**2. Get a list of Top 5 Profitable Films:**

db.movies\_tmdb.aggregate([

{ $project: {

title: 1,

profit: { $subtract: ["$revenue", "$budget"] }

}},

{ $sort: { profit: -1 }},

{ $limit: 5 }

]);

A screenshot of a computer

Description automatically generated

**3. Find Cast Influence on Revenue**

db.movie\_db.aggregate([

{ $unwind: "$cast" },

{ $group: {

\_id: "$cast.name",

TotalRevenue: { $sum: "$revenue" },

MoviesCount: { $sum: 1 }

}},

{ $sort: { TotalRevenue: -1 }}

]);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**4. Find Revenue Comparison by Production Country:**

db.movie\_db.aggregate([

{ $unwind: "$production\_countries" },

{ $group: {

\_id: "$production\_countries.name",

TotalRevenue: { $sum: "$revenue" }

}},

{ $sort: { TotalRevenue: -1 }}

]);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**5.Find Average Rating by Genre:**

db.movie\_db.aggregate([

{ $unwind: "$genres" },

{ $group: {

\_id: "$genres.name",

AverageRating: { $avg: "$vote\_average" }

}},

{ $sort: { AverageRating: -1 }}

]);

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**6. To find count of movies by genre:**

db.movie\_db.aggregate([

{ $unwind: "$genres" },

{ $group: {

\_id: "$genres.name",

MovieCount: { $sum: 1 }

}},

{ $sort: { MovieCount: -1 }}

]);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

SQL queries

1. **List Top 10 Grossing Films.**

db.movies.find({}, {title: 1, revenue: 1, budget: 1})

.sort({revenue: -1})

.limit(10);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

2. **Find Correlation Between Budget and Revenue.**

db.movies.find({$and: [{budget: {$ne: null}}, {revenue: {$ne: null}}]},

{budget: 1, revenue: 1});

A screenshot of a computer

Description automatically generated

3**. Find films with High Ratings and Their Commercial Success.**

db.movies.find({$and: [{vote\_average: {$gt: 8}}, {revenue: {$ne: null}}]},

{title: 1, vote\_average: 1, revenue: 1})

.sort({revenue: -1});

A screenshot of a computer

Description automatically generated

4. **Find Movies with a Specific Genre.**

db.movie\_db.find({"genres.name": "Action"}, {title: 1, revenue: 1, budget: 1})

.limit(5);

A screenshot of a computer

Description automatically generated

5. **List Movies with Budgets Exceeding a Certain Value.**

db.movie\_db.find({budget: {$gt: 200000000}}, {title: 1, budget: 1})

.sort({budget: -1})

.limit(5);

A screenshot of a computer

Description automatically generated

6. **Retrieve Movies with High Viewer Ratings but Low Box Office Revenue.**

db.movie\_db.find({$and: [{vote\_average: {$gt: 8}}, {revenue: {$lt: 10000000}}]},

{title: 1, vote\_average: 1, revenue: 1})

.limit(5);

A screenshot of a computer

Description automatically generated

7. **Show Movies with Multiple Languages.**

db.movie\_db.find({"spoken\_languages.1": {$exists: true}}, {title: 1, spoken\_languages: 1})

.limit(2);

A screenshot of a computer

Description automatically generated

8. **List Movies with a Specific Original Language.**

db.movie\_db.find({original\_language: "ja"}, {title: 1, original\_language: 1, revenue: 1})

.limit(5);

A screenshot of a computer

Description automatically generated

9. **Find Movies that Significantly Outperformed their Budget.**

db.movie\_db.find({$expr: {$gt: ["$revenue", {$multiply: ["$budget", 5]}]}},

{title: 1, budget: 1, revenue: 1})

.limit(5);

A screenshot of a computer

Description automatically generated

10. **List Movies with a Large Number of Crew Members**.

db.movie\_db.find({"crew.10": {$exists: true}}, {title: 1, crew: 1})

.limit(2);

A screenshot of a computer

Description automatically generated