ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Факультет 1

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №** 5

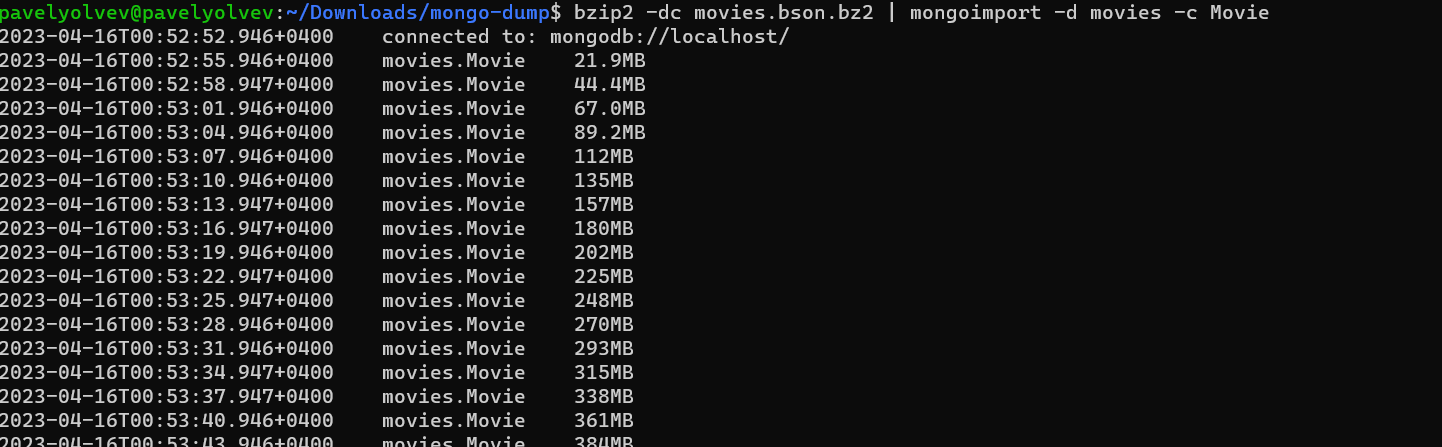
по дисциплине Системы управления базами данных

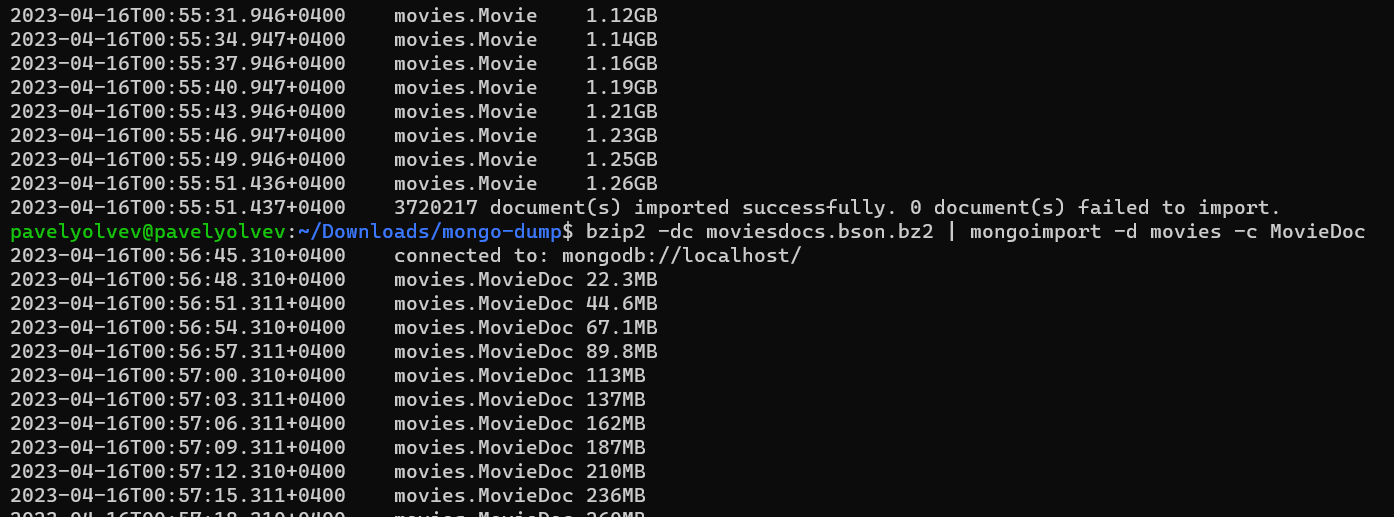
|  |  |  |
| --- | --- | --- |
| **ВЫПОЛНИЛ** | | |
| студент | РПИС-11  (группа) | Яковлев П. Н.  (ФИО) |
|  | **ПРОВЕРИЛ** |  |
| (должность) | | Борисов В.В.  (ФИО) |

Самара

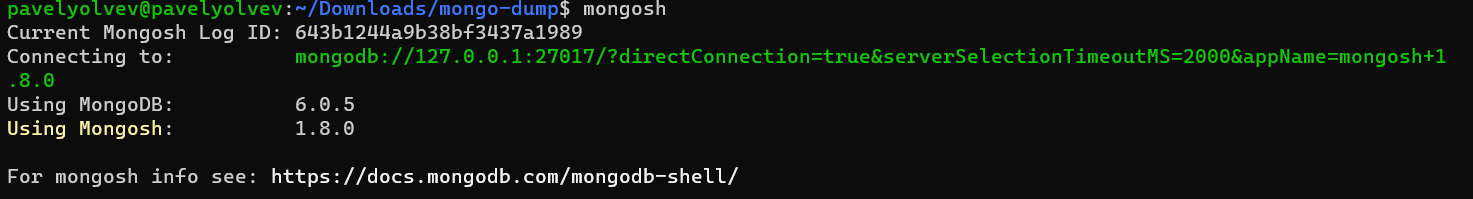
2023

Импорт базы:





С версии 6.0 переход в консоль mongodb осуществляется при помощи команды mongosh:

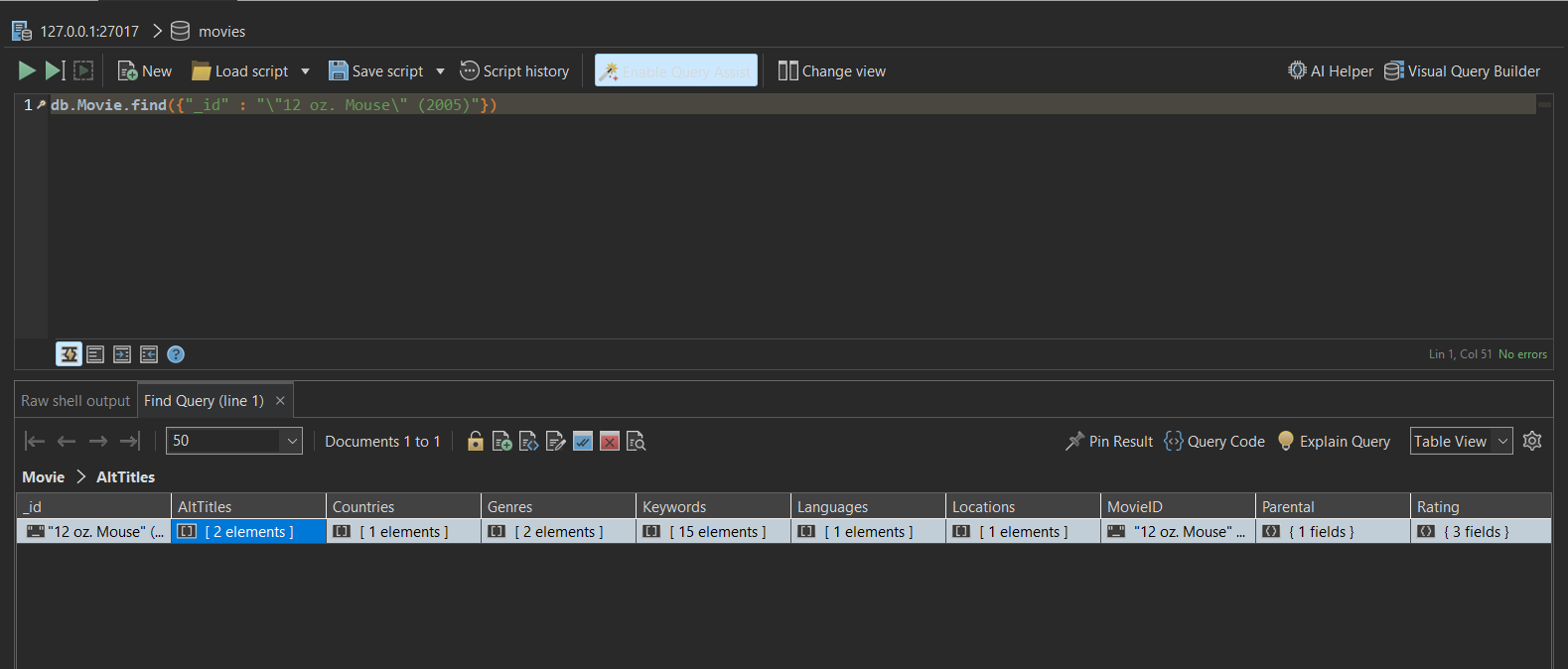


Вывод баз и переключение на базу movies, вывод коллекций:



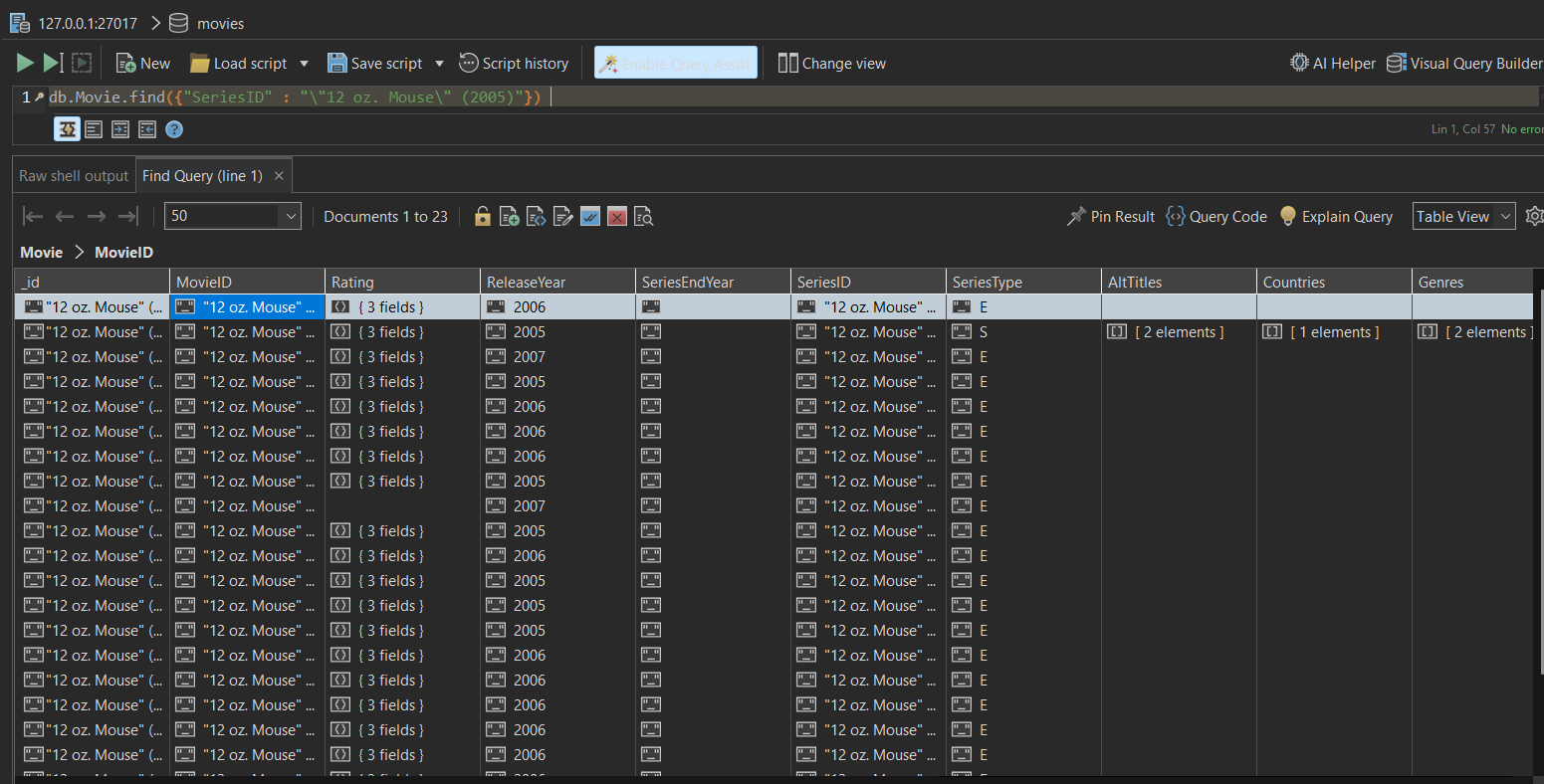
Выполнение запросов:

db.Movie.find({"\_id" : "\"12 oz. Mouse\" (2005)"}) // выбрать сериал или фильм

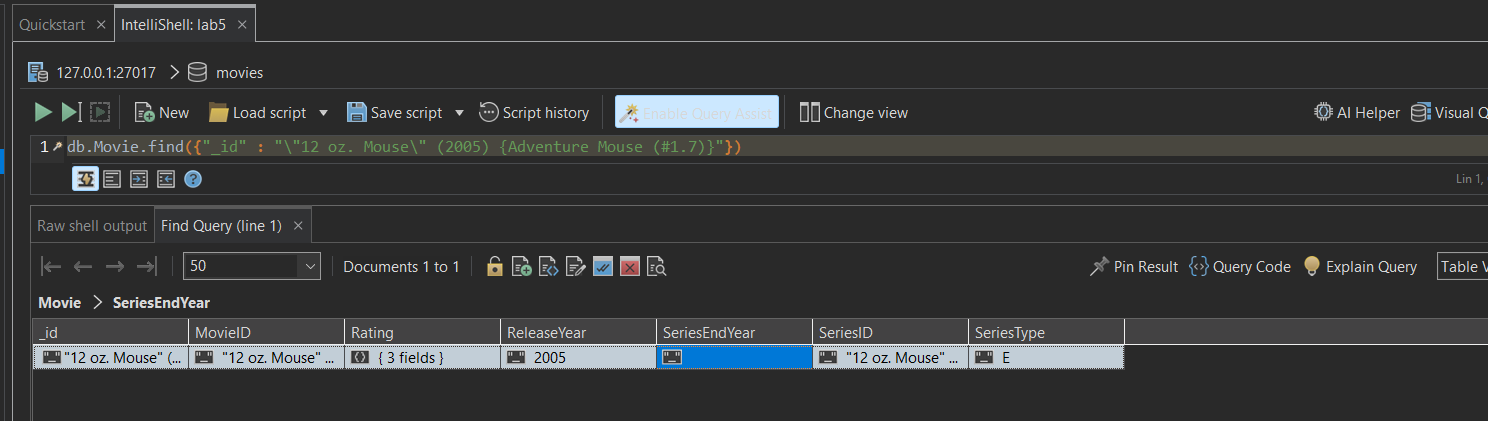




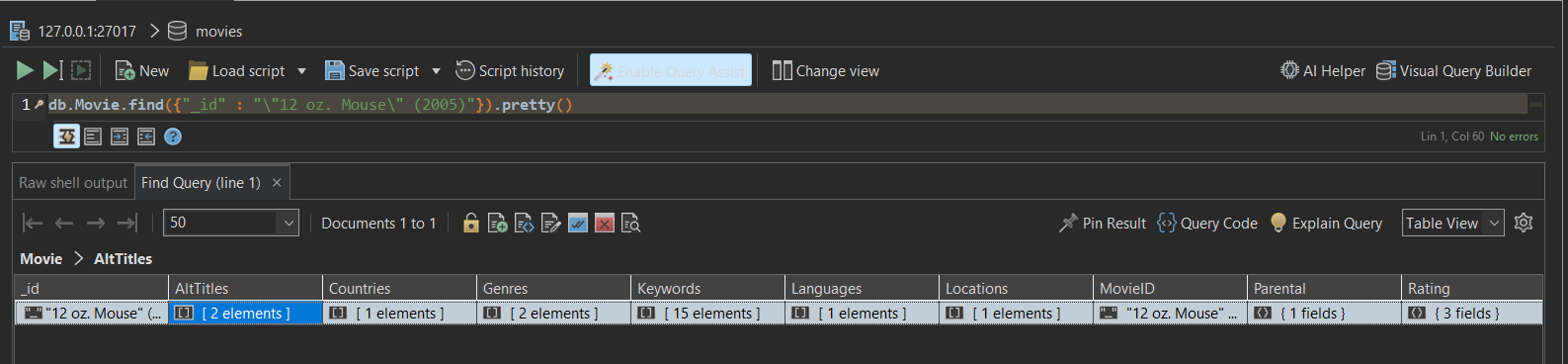
db.Movie.find({"SeriesID" : "\"12 oz. Mouse\" (2005)"}) // выбрать сериал и все эпизоды



db.Movie.find({"\_id" : "\"12 oz. Mouse\" (2005) {Adventure Mouse (#1.7)}"}) // выбрать конкретный эпизод



db.Movie.find({"\_id" : "\"12 oz. Mouse\" (2005)"}).pretty()



Вывод списка индексов:

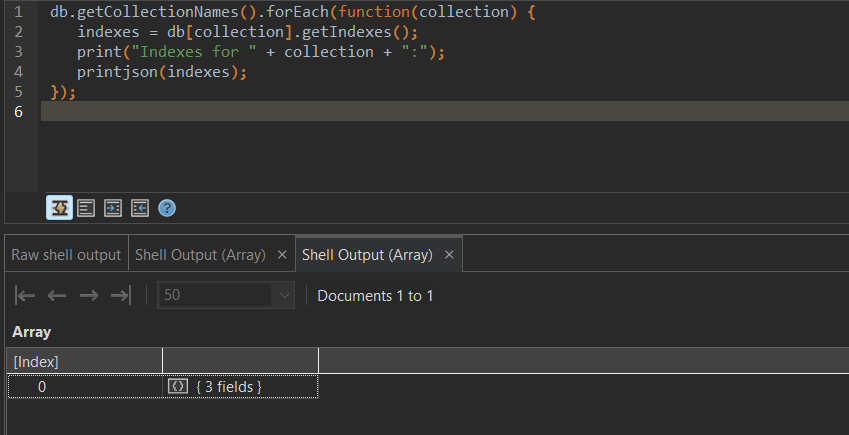
db.getCollectionNames().forEach(function(collection) {

indexes = db[collection].getIndexes();

print("Indexes for " + collection + ":");

printjson(indexes);

});



db.Movie.forEach(function(collection) {

indexes = db[collection].getIndexes();

print("Indexes for " + collection + ":");

printjson(indexes);

});

Преобразование годов из string в int:

db.Movie.updateMany(

{

$or: [

{ ReleaseYear: { $type: "string", $ne: "" } },

{ ReleaseYear: { $exists: false } }

]

},

[

{

$addFields: {

ReleaseYear: {

$cond: {

if: { $and: [ { $eq: [ { $type: "$ReleaseYear" }, "string" ] }, { $ne: [ "$ReleaseYear", "" ] } ] },

then: { $toInt: "$ReleaseYear" },

else: { $toInt: { $substr: [ { $arrayElemAt: [ { $split: [ { $arrayElemAt: [ { $split: [ "$\_id", "(" ] }, 1 ] }, ")" ] }, 0 ] }, 0, 4 ] } }

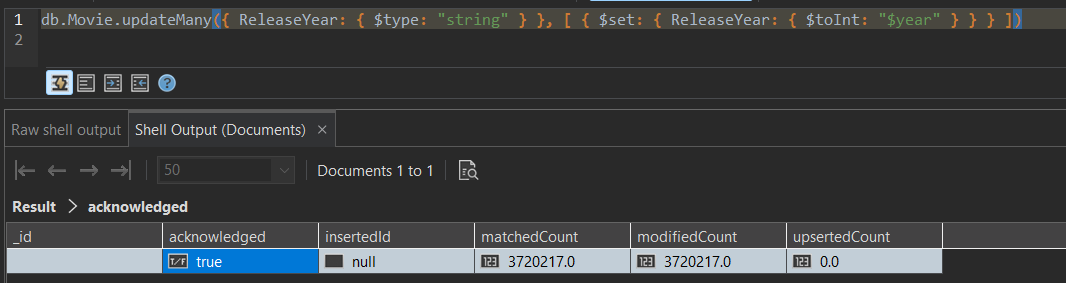
}

}

}

}

]

)

Задания на защиту

**1. Составить запросы:**

**Для каждого года из первой декады XXI века посчитать количество снятых фильмов.**

**db.Movie.aggregate([**

**{**

**$match: {**

**ReleaseYear: { $gte: 2000, $lte: 2009 }**

**}**

**},**

**{**

**$group: {**

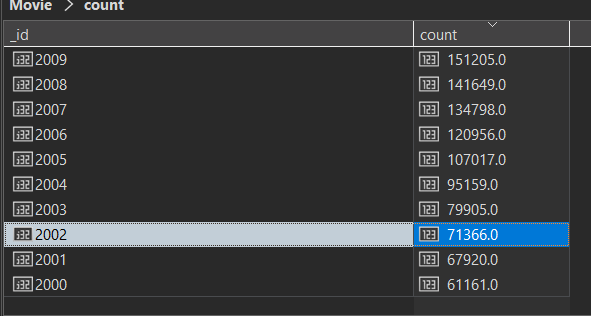
**\_id: "$ReleaseYear",**

**count: { $sum: 1 }**

**}**

**}**

**])**



**Выбрать топ-10 наиболее популярных ключевых слов для фильмов заданной страны.**

db.Movie.aggregate([

{

$match: { Countries: "Russia" }

},

{

$unwind: "$Keywords"

},

{

$group: {

\_id: "$Keywords",

count: { $sum: 1 }

}

},

{

$sort: { count: -1 }

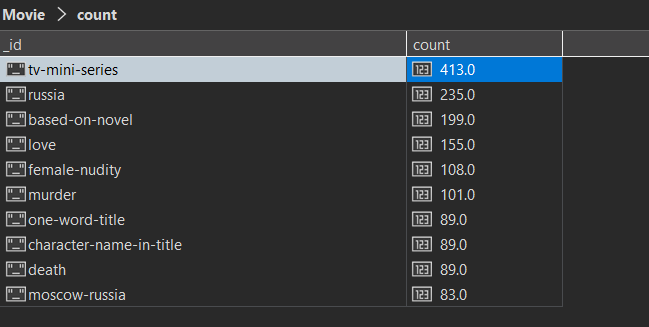
},

{

$limit: 10

}

])



**2. Составить свою формулу рейтинга (общего для всех фильмов или по какому-либо фильтру), используя параметры словарей Movie.Rating (Rating - среднее значение оценкой, RatingVotes - количество оценок) или любые другие.**

ранг фильма = (ax + bv)/(c + v)

где:

X = рейтинг фильма

v = количество голосов

a, b = веса для установки важности оценки или голосов соответственно (для данного списка 0.65 и 0.35)

C = минимальное количество голосов, чтобы попасть в список (для данного списка 25000)

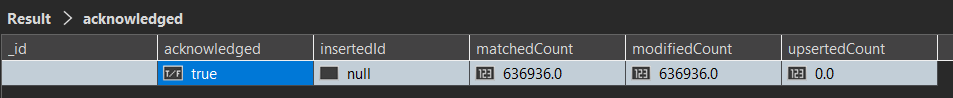
Таким образом формула примет вид:

ранг фильма = (0.65\*x + 0.35\*v)/(25000 + v)

**преобразование рейтинга из строки в число**

**db.Movie.updateMany({ "Rating.RatingVotes": { $type: "string" } }, [ { $set: { "Rating.RatingVotes": { $toInt: "$Rating.RatingVotes" } } } ])**

**db.Movie.updateMany({ "Rating.Rating": { $type: "string" } }, [ { $set: { "Rating.Rating": { $toDouble: "$Rating.Rating" } } } ])**



**Запрос:**

**db.getCollection("Movie").aggregate({**

**$project:{"rank":**

**{$divide: [**

**{ $add: [{ $multiply: ["$Rating.Rating", 4] }, { $multiply: ["$Rating.RatingVotes", 10] }] },**

**{ $add: [25000, "$Rating.RatingVotes"] }**

**]}}},**

**{$sort: { "rank": -1 }},**

**{$limit: 10})**

