



**دانشگاه صنعتی امیرکبیر**  
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

استاد درس: دکتر محمد اکبری

اسفند ۱۴۰۳

**تمرین اول**

درس پایگاه داده



## توضیح تمرین

پروژه سیستم پیشنهاد فیلم با استفاده از MongoDB ایجاد کنید که از مدل سند برای مدیریت سلسله‌مراتبی داده‌های فیلم استفاده می‌کند. MongoDB برای این سیستم ایده‌آل است زیرا:

۱. بدون نیاز به جداول سخت‌گیرانه با ویژگی‌های متنوع فیلم کار می‌کند
۲. آرایه‌های ژانر/کارگردان/بازیگر را به صورت بومی پردازش می‌کند
۳. منطق پیچیده پیشنهاد را در کوئری‌های واحد اجرا می‌کند
۴. خواندن سریع‌تر داده‌های تو در تو در مقایسه با SQL. ارائه می‌دهد

## دستورالعمل‌های راه‌اندازی (ویندوز)

### ۱. نصب سرور: MongoDB

- دانلود: [https://fastdl.mongodb.org/windows/mongodb-windows-x86\\_64-8.0.10-signed.msi](https://fastdl.mongodb.org/windows/mongodb-windows-x86_64-8.0.10-signed.msi)

- اضافه کردن C:\Program Files\MongoDB\Server\8.0\bin به System PATH

### ۲. نصب shell: MongoDB

- دانلود: <https://downloads.mongodb.com/compass/mongosh-2.5.2-win32-x64.zip>
- استخراج به C:\mongosh
- اضافه کردن C:\mongosh\bin به System PATH

### ۳. شروع سرویس:

```
net start MongoDB
```

### ۴. بارگیری مجموعه داده:

(آ) اسکریپت برای خواندن داده‌ها بنویسید

(ب) فایل movies.json را با تابع readFileSync از ماژول fs بخوانید



(ج) داده‌های JSON را با تابع `JSON.parse` تجزیه کنید

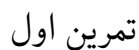
(د) از متد `insertMany()` برای افزودن اسناد به مجموعه `movies` استفاده کنید

اجرا:

```
mongosh load-movies.js
```

نمونه داده بعد ریختن داده ها

```
movieDB> db.movies.findOne()
{
  _id: ObjectId('684f1025116872a7ed50eb67'),
  title: 'Order traditional',
  genres: [ 'Fantasy' ],
  release_year: 2013,
  directors: [ 'Amy Sims', 'Melissa Foster' ],
  cast: [
    'Ruben Yang',
    'Debbie Phillips',
    'Bradley Chavez',
    'Sheila Diaz',
    'Michael Lyons',
    'John Mooney MD'
  ],
  rating: 7.4,
  votes: 991252,
  description: 'Task lay century great move perform. Perform one hit step. World incre
}
```



۱. توزیع ژانر کوئری برای شمارش فیلم‌ها بر اساس ژانر، به ترتیب از بیشترین به کمترین

۲. کارگردان‌های برتر کوئری برای یافتن کارگردان‌های دارای بیش از ۳ فیلم و نمایش میانگین امتیاز آن‌ها

۳. ارتباطات بازیگر کوئری برای یافتن فیلم‌هایی که حداقل ۲ بازیگر مشترک با فیلم "Speech" دارند

## مدیریت کاربر

- ایجاد کاربر: درج سند جدید در مجموعه users
- ورود کاربر: یافتن کاربر بر اساس نام کاربری
- افزودن فیلم به دیده‌شده‌ها: به‌روزرسانی سند کاربر با افزودن شناسه فیلم به آرایه watched\_movies
- حذف فیلم از دیده‌شده‌ها: به‌روزرسانی سند کاربر با حذف شناسه فیلم از آرایه

## نمونه کاربرد

[illegible]



```
]
}
```

### جستجوی فیلم

- جستجوی فیلم‌ها: جستجوی بدون حساسیت به حروف در فیلد title
- نمایش فیلم‌های دیده‌شده: یافتن فیلم‌ها بر اساس شناسه‌های موجود در آرایه watched\_movies کاربر

### موتور پیشنهاد

۱. دریافت فیلم‌های دیده‌شده کاربر: بازیابی آرایه watched\_movies کاربر

۲. محاسبه وزن‌های ترجیحی: برای هر فیلم دیده‌شده:

- شمارش ظاهر شدن ژانرها
- شمارش ظاهر شدن کارگردان‌ها
- شمارش ظاهر شدن بازیگران

نرمال‌سازی شمارش‌ها به وزن‌ها (مجموع برابر ۰.۱)

۳. امتیازدهی فیلم‌های دیده‌نشده:

```
totalScore =
    (SUM(genre_weights * match) * 0.4 +
    (SUM(director_weights * match) * 0.25 +
    (SUM(actor_weights * match) * 0.15 +
    (rating / 20) +
    (LOG10(votes) / 100
```

۴. دریافت پیشنهادات:

- یافتن فیلم‌های موجود در لیست دیده‌شده‌ها
- محاسبه امتیازات با استفاده از خط لوله تجمیع



- مرتب‌سازی بر اساس امتیاز نزولی
- محدود کردن به ۲۰ نتیجه برتر

### پیاده‌سازی رابط گرافیکی (امتیازی)

- ورود/ثبت نام
- نوار جستجو با نتایج لحظه ای
- پنل فیلم‌های دیده‌شده
- نمایش پیشنهادات
- داشبورد کوئری‌های مدیریتی

### کوئری‌های مدیریتی (پیاده‌سازی و گزارش)

۱. تخصص کارگردان- ژانر کوئری برای یافتن کارگردان‌هایی که عمدتاً روی یک ژانر تمرکز دارند (درصد بالایی از فیلم‌های آن‌ها در یک ژانر)
۲. شبکه همکاری بازیگر کوئری برای یافتن بازیگرانی که مکرراً با Henry Robert ظاهر می‌شوند
۳. جدول زمانی محبوبیت ژانر کوئری برای ردیابی تغییرات محبوبیت ژانر بر اساس دهه
۴. همکاری بازیگر- کارگردان کوئری برای یافتن زوج‌های بازیگر- کارگردان با بالاترین میانگین امتیازات

### راهنمایی

- برای مدیریت کاربران:

```
// create user
```

```
db.users.insertOne({username: "test", password: "<hash>", watched_movies: []})
```



```
// login
db.users.findOne({username: "test"})
```

- برای جستجوی فیلم:

```
db.movies.find(
  {title: {$regex: "action", $options: "i"}},
  {title: 1, release_year: 1}
)
```

- راهنمای کوئری‌های مدیریتی

#### – تخصص کارگردان-ژانر:

۱. جدا کردن ژانرها و کارگردانها
۲. گروه‌بندی بر اساس کارگردان و ژانر
۳. محاسبه درصد تخصص
۴. فیلتر کارگردان‌های متخصص

#### – شبکه همکاری بازیگر:

۱. یافتن فیلم‌های حاوی بازیگر هدف
۲. جدا کردن لیست بازیگران
۳. حذف بازیگر هدف
۴. گروه‌بندی بر اساس بازیگر همکار
۵. محاسبه تعداد همکاری‌ها و میانگین امتیاز

#### – جدول زمانی محبوبیت ژانر:

۱. محاسبه دهه انتشار
۲. جدا کردن ژانرها
۳. گروه‌بندی بر اساس دهه و ژانر
۴. محاسبه تعداد فیلم‌ها و میانگین امتیاز



- همکاری بازیگر-کارگردان:

۱. جدا کردن کارگردان ها و بازیگران

۲. گروه بندی بر اساس زوج کارگردان-بازیگر

۳. محاسبه میانگین امتیاز

۴. مرتب سازی نزولی

• راهنمایی موتور پیشنهاد

```
// MongoDB Aggregation Pipeline
db.movies.aggregate([
  // Stage 1: Exclude watched movies
  {$match: {_id: {$nin: user_watched_ids}}},

  // Stage 2: Calculate match scores
  {$addFields: {
    genre_score: {
      $sum: {
        $map: {
          input: "$genres",
          as: "g",
          in: user_genre_weights[$$g] || 0
        }
      }
    },
    director_score: { /* similar logic */ },
    actor_score: { /* similar logic */ }
  }},

  // Stage 3: Compute total score
```





```
{ $addFields: {  
  total_score: {  
    $add: [  
      { $multiply: [ "$genre_score", 0.4 ] },  
      { $multiply: [ "$director_score", 0.25 ] },  
      { $multiply: [ "$actor_score", 0.15 ] },  
      { $divide: [ "$rating", 20 ] },  
      { $divide: [ { $log10: "$votes" }, 100 ] }  
    ]  
  }  
}},  
  
// Stage 4: Sort and limit  
{ $sort: { total_score: -1 } },  
{ $limit: 20 }  
])
```

## خروجی مورد انتظار

### مستندات راه اندازی

- تصویر اتصال موفق mongosh

- خروجی db.movies.countDocuments()

### کوئری های اولیه

- برای هر کوئری:



- متن کوئری
- نتایج (تصویر یا جدول)
- توضیح مختصر

### سیستم پیشنهاد

- توضیح الگوریتم امتیازدهی
- نمونه پیشنهادات برای کاربر تست
- کد سیستم پیشنهادات

### کوئری‌های مدیریتی

- برای هر کوئری:
- متن کوئری
- نتایج
- تحلیل یک جمله‌ای

### امتیازی (رابط گرافیکی)

- کد کامل پیاده‌سازی رابط
- ویدئوی دموی ۳۰ ثانیه‌ای (اختیاری)