Working with MongoDB

Exercise Workbook

Contents

Lab 1: Index 설정하기

1. "title" 필드에 대한 인덱스 설정:

```
db.movieDetails.createIndex ({ title: 1 })
```

이 명령은 "title" 필드에 오름차순으로 인덱스를 설정합니다. 이렇게 함으로써 "title" 값을 이용한 검색이 빠르게 수행될 수 있습니다.

2. "year" 필드에 대한 인덱스 설정:

```
db.movieDetails.createIndex({ year: 1 })
```

이 명령은 "year" 필드에 내림차순으로 인덱스를 설정합니다. 이렇게 함으로써 "year" 값을 이용한 정렬이 빠르게 수행될 수 있습니다.

3. "genres" 필드에 대한 다중 키 인덱스 설정:

```
db.movieDetails.createIndex({ genres: 1 })
```

이 명령은 "genres" 필드에 대한 다중 키 인덱스를 설정합니다. 이렇게 함으로써 "genres" 값을 이용한 멀티키 쿼리가 효율적으로 수행될 수 있습니다.

4. "actors" 필드에 대한 배열 인덱스 설정:

```
db.movieDetails.createIndex({ actors: 1 })
```

이 명령은 "actors" 필드에 대한 배열 인덱스를 설정합니다. 이렇게 함으로써 "actors" 배열 내의 요소를 이용한 쿼리가 효율적으로 수행될 수 있습니다.

5. "imdb.rating" 필드에 대한 인덱스 설정:

```
db.movieDetails.createIndex({ "imdb.rating": -1 })
```

이 명령은 "imdb.rating" 필드에 내림차순으로 인덱스를 설정합니다. 이렇게 함으로써 "imdb.rating" 값을 이용한 정렬이 빠르게 수행될 수 있습니다.

6. "tomato.meter" 필드에 대한 인덱스 설정:

```
db.movieDetails.createIndex({ "tomato.meter": -1 })
```

이 명령은 "tomato.meter" 필드에 내림차순으로 인덱스를 설정합니다. 이렇게 함으로써 "tomato.meter" 값을 이용한 정렬이 빠르게 수행될 수 있습니다.

7. "director" 필드에 대한 텍스트 인덱스 설정:

```
db.movieDetails.createIndex({ director: "text" })
```

이 명령은 "director" 필드에 대한 텍스트 인덱스를 설정합니다. 이렇게 함으로써 "director" 값을 이용한 텍스트 검색이 가능해집니다.

8. "awards.wins" 필드에 대한 인덱스 설정:

```
db.movieDetails.createIndex({ "awards.wins": -1 })
```

이 명령은 "awards.wins" 필드에 내림차순으로 인덱스를 설정합니다. 이렇게 함으로써 "awards.wins" 값을 이용한 정렬이 빠르게 수행될 수 있습니다.

9. "director", “year” 필드에 대한 복합 인덱스 설정:

```
db.movieDetails.createIndex({ director: 1, year: -1 })
```

이 명령은 "director" 필드에 오름차순으로, “year” 필드에 내림차순 인덱스를 설정합니다. 이렇게 함으로써 "genres", “year” 값을 이용한 정렬이 빠르게 수행될 수 있습니다.

10. "runtime" 필드에 대한 인덱스 설정:

이 명령은 "runtime" 필드에 오름차순으로 인덱스를 설정합니다. 이렇게 함으로써 "runtime" 값을 이용한 정렬이 빠르게 수행될 수 있습니다.

11. "countries" 필드에 대한 인덱스 설정:

이 명령은 "countries" 필드에 오름차순으로 인덱스를 설정합니다. 이렇게 함으로써 "countries" 값을 이용한 정렬이 빠르게 수행될 수 있습니다.

12. "director" 필드에 대한 인덱스 설정:

이 명령은 "director" 필드에 오름차순으로 인덱스를 설정합니다. 이렇게 함으로써 "director" 값을 이용한 정렬이 빠르게 수행될 수 있습니다.

13. "imdb.rating" 필드에 대한 인덱스 설정:

이 명령은 "imdb.rating" 필드에 오름차순으로 인덱스를 설정합니다. 이렇게 함으로써 "imdb.rating" 값을 이용한 정렬이 빠르게 수행될 수 있습니다.

14. "tomato.meter" 필드에 대한 인덱스 설정:

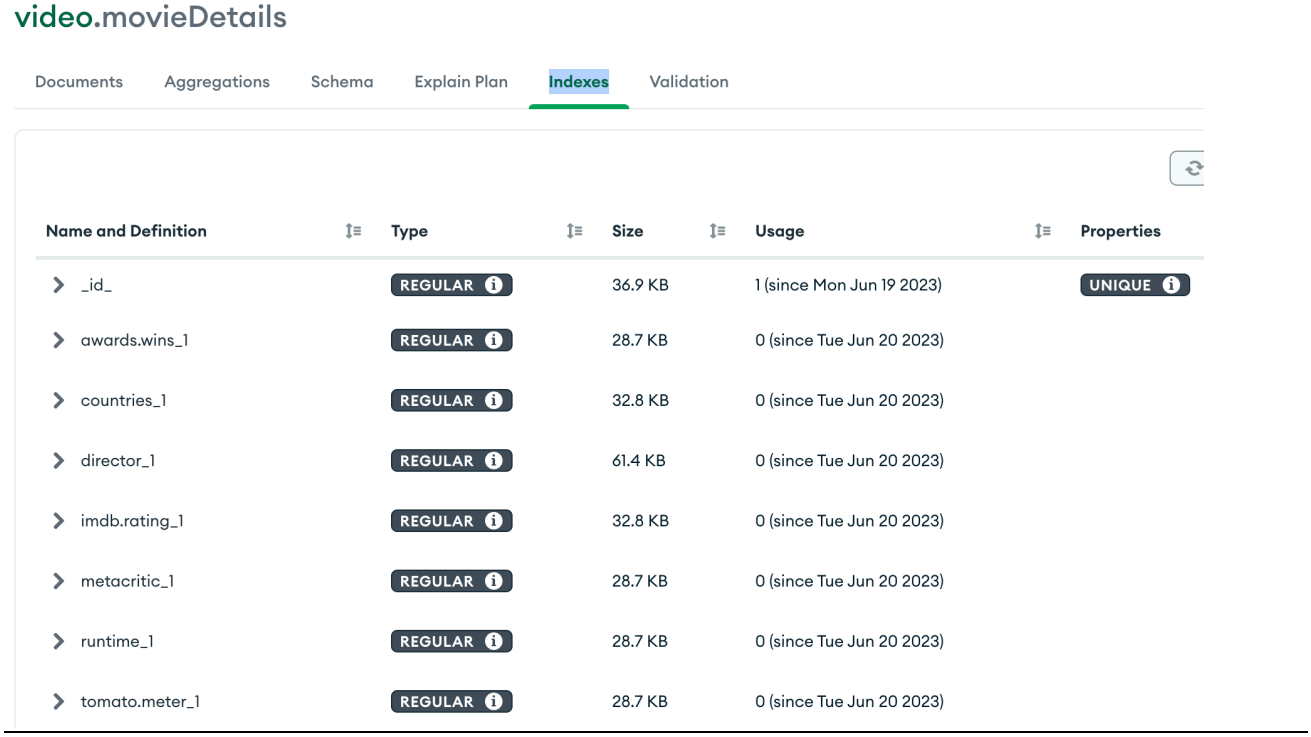
이 명령은 "tomato.meter" 필드에 오름차순으로 인덱스를 설정합니다. 이렇게 함으로써 "tomato.meter" 값을 이용한 정렬이 빠르게 수행될 수 있습니다.

15. "metacritic" 필드에 대한 인덱스 설정:

이 명령은 "metacritic" 필드에 오름차순으로 인덱스를 설정합니다. 이렇게 함으로써 "metacritic" 값을 이용한 정렬이 빠르게 수행될 수 있습니다.

16. "awards.wins" 필드에 대한 인덱스 설정:

이 명령은 "awards.wins" 필드에 오름차순으로 인덱스를 설정합니다. 이렇게 함으로써 "awards.wins" 값을 이용한 정렬이 빠르게 수행될 수 있습니다.



Lab 2: Aggregation pipeline 생성 하기

1. 다음 코드는 MongoDB 의 집계(aggregation) 기능을 사용하여 movieDetails 컬렉션에서 "Western" 장르를 가진 영화들을 찾고, 평가(rated)별로 평균 재생 시간과 영화 개수를 계산하는 작업을 수행합니다. 그리고 평균 재생 시간을 기준으로 내림차순으로 정렬합니다

```
video> db.movieDetails.aggregate([
...   {
...     $match: { genres: "Western" }
...   },
...   {
...     $group: {
...       _id: "$director", // Group by director
...       averageRating: { $avg: "$imdb.rating" },
...       totalVotes: { $sum: "$imdb.votes" }
...     }
...   },
...   {
...     $sort: { averageRating: -1 }
...   },
...   {
...     $limit: 5
...   }
... ])
```

2. 다음 코드는 `$project` 단계를 추가하여 필요한 필드들만 선택합니다. 그리고 `$unwind` 단계를 사용하여 "actors" 배열 필드를 각각의 문서로 분리합니다. 그 후 `$match` 단계를 사용하여 "actors" 필드가 "Henry Fonda"인 문서들을 필터링합니다. 그리고 `$group` 단계를 사용하여 "genres" 필드를 기준으로 그룹화하고, 각 그룹별로 영화 개수를 계산하며 해당 그룹의 영화 제목들을 배열로 저장합니다. 그런 다음 `$sort` 단계를 사용하여 영화 개수에 따라 내림차순으로 정렬하고, `$limit` 단계를 사용하여 상위 5 개 결과만 선택합니다.

```
video> db.movieDetails.aggregate([
... {
...   $project: {
...     title: 1,
...     year: 1,
...     rated: 1,
...     genres: 1,
...     director: 1,
...     actors: 1
...   }
... },
... {
...   $unwind: "$actors"
... },
... {
...   $match: {
...     actors: "Henry Fonda"
...   }
... },
... {
...   $group: {
...     _id: "$genres",
...     count: { $sum: 1 },
...     movies: { $push: "$title" }
...   }
... },
... {
...   $sort: {
...     count: -1
...   }
... },
... {
...   $limit: 5
... }
... ])
```

3. 다음 코드는 `movieDetails` 컬렉션에서 Western 장르의 영화를 찾기 위해 `$match` 단계를 사용합니다. 그 다음 `$group` 단계를 사용하여 `rated` 필드를 그룹화하고, 각 그룹별로 평균 `runtime` 값을 계산하고, 그룹의 영화 개수를 세어줍니다. 마지막으로 `$sort` 단계를 사용하여 평균 `runtime`을 내림차순으로 정렬합니다.

```
video> db.movieDetails.aggregate([
...   {
...     $match: { genres: "Western" }
...   },
...   {
...     $group: {
...       _id: "$rated",
...       averageRuntime: { $avg: "$runtime" },
...       count: { $sum: 1 }
...     }
...   },
...   {
...     $sort: { averageRuntime: -1 }
...   }
... ])
```

LAB 4. 다음 결과 화면과 조건에 맞는 Aggregation pipeline 를 작성 하세요.

이 과제는 $match 단계는 장르 필드를 기준으로 문서를 필터링합니다. 이 예시에서는 "Western" 장르를 가진 문서와 일치합니다.

$project 단계는 출력에 포함할 특정 필드를 선택합니다. 제목, 연도, 감독, 배우, 줄거리, imdb, 토마토 필드를 포함합니다.

$addFields 단계는 문서에 새로운 필드를 추가합니다. 이 예시에서는 $toString 연산자를 사용하여 imdb.votes 필드와 tomato.reviews 필드를 문자열로 변환합니다.

이 파이프라인은 선택한 필드와 투표 및 리뷰를 문자열로 변환하여 일치하는 문서를 반환합니다.

(결과화면)


```
{
  title: 'Once Upon a Time in the West',
  year: 1968,
  director: 'Sergio Leone',
  actors: [
    'Claudia Cardinale',
    'Henry Fonda',
    'Jason Robards',
    'Charles Bronson'
  ],
  plot: 'Epic story of a mysterious stranger with a harmonica who joins forces with a notorious desperado to protect
a beautiful widow from a ruthless assassin working for the railroad.',
  imdb: {
    id: 'tt0064116',
    rating: 8.6,
    votes: 201283,
    votesString: '201283'
  },
  tomato: {
    meter: 98,
    image: 'certified',
    rating: 9,
    reviews: 54,
    fresh: 53,
    consensus: 'A landmark Sergio Leone spaghetti western masterpiece featuring a classic Morricone score.',
    userMeter: 95,
    userRating: 4.3,
    userReviews: 64006,
    reviewsString: '54'
  }
},
```

LAB 5. 다음 결과 화면과 조건에 맞는 Aggregation pipeline 를 작성 하세요.

이 집계 파이프라인은 다음과 같은 작업을 수행합니다:

1. `$match` 단계는 "genres" 필드를 기준으로 문서를 필터링합니다. 이 예시에서는 "Western" 장르의 문서를 선택합니다.

2. `$project` 단계는 출력에 포함할 특정 필드를 선택합니다. "title", "director", "actors", "plot" 필드를 포함합니다.

3. `$addFields` 단계는 "ratingAvg"라는 새로운 필드를 추가합니다. "imdb.rating"과 "tomato.rating" 필드를 사용하여 평균 평점을 계산합니다.

이 파이프라인은 "Western" 장르의 문서에서 선택된 필드와 평균 평점인 "ratingAvg"를 반환합니다.

(결과화면)

```
{
  title: 'Once Upon a Time in the West',
  director: 'Sergio Leone',
  actors: [
    'Claudia Cardinale',
    'Henry Fonda',
    'Jason Robards',
    'Charles Bronson'
  ],
  plot: 'Epic story of a mysterious stranger with a harmonica who joins forces with a notorious desperado to protect a beautiful widow from a ruthless assassin working for the railroad.',
  ratingAvg: null
},
```