Chapter 4: Core Aggregation - Multidimensional Grouping

Lab - $facets

Back to the Question

The solution we used follows, following the requirement that we use only one database access

COPY

```
db.movies.aggregate([
  {
    $match: {
      metacritic: { $gte: 0 },
      "imdb.rating": { $gte: 0 }
    }
  },
  {
    $project: {
      _id: 0,
      metacritic: 1,
      imdb: 1,
      title: 1
    }
  },
  {
    $facet: {
      top_metacritic: [
        {
          $sort: {
            metacritic: -1,
            title: 1
          }
        },
        {
          $limit: 10
        },
        {
          $project: {
            title: 1
          }
        }
      ],
      top_imdb: [
        {
          $sort: {
            "imdb.rating": -1,
            title: 1
          }
        },
        {
          $limit: 10
        },
        {
          $project: {
            title: 1
          }
        }
      ]
    }
  },
  {
    $project: {
      movies_in_both: {
        $setIntersection: ["$top_metacritic", "$top_imdb"]
      }
    }
  }
])
```

We begin with a `$match` and `$project` stage to only look at documents with the relevant fields, and project away needless information

COPY

```
{
  $match: {
    metacritic: { $gte: 0 },
    "imdb.rating": { $gte: 0 }
  }
},
{
  $project: {
    _id: 0,
    metacritic: 1,
    imdb: 1,
    title: 1
  }
},
```

Next follows our `$facet` stage. Within each facet, we need sort in descending order for `metacritic` and `imdb.ratting` and ascending for `title`, limit to 10 documents, then only retain the title

COPY

```
{
  $facet: {
    top_metacritic: [
      {
        $sort: {
          metacritic: -1,
          title: 1
        }
      },
      {
        $limit: 10
      },
      {
        $project: {
          title: 1
        }
      }
    ],
    top_imdb: [
      {
        $sort: {
          "imdb.rating": -1,
          title: 1
        }
      },
      {
        $limit: 10
      },
      {
        $project: { title: 1 }
      }
    ]
  }
},
```

Lastly, we use a `$project` stage to find the intersection of `top_metacritic` and `top_imdb`, producing the titles of movies in both categories

COPY

```
{
  $project: {
    movies_in_both: {
      $setIntersection: ["$top_metacritic", "$top_imdb"]
    }
  }
}
```

This results in the following output

```
{ "movies_in_both" : [ { "title" : "The Godfather" } ]
```

COPY