


Chapter 3: Core Aggregation - Combining Information


Lab - \$group and Accumulators

 [Back to the Question](#)

 COPY


```
db.movies.aggregate([
  {
    $match: {
      awards: /Won \d{1,2} Oscars?/
    }
  },
  {
    $group: {
      _id: null,
      highest_rating: { $max: "$imdb.rating" },
      lowest_rating: { $min: "$imdb.rating" },
      average_rating: { $avg: "$imdb.rating" },
      deviation: { $stdDevSamp: "$imdb.rating" }
    }
  }
])
```

We start by applying the now familiar `$match` filtering, searching documents for the appropriate text stating they won an Oscar

 COPY

```
{
  $match: {
    awards: /Won \d{1,2} Oscars?/
  }
},
```

Next, we have our `$group` stage. By convention, we group all documents together by specifying `null` to ```_id`. We use the [group accumulators](#) `$min`, `$max`, `$avg`, and `$stdDevSamp` to get our results

 COPY

```
{
  $group: {
    _id: null,
    highest_rating: { $max: "$imdb.rating" },
    lowest_rating: { $min: "$imdb.rating" },
    average_rating: { $avg: "$imdb.rating" },
    deviation: { $stdDevSamp: "$imdb.rating" }
  }
}
```

Proceed to next section