



## Chapter 1: Basic Aggregation - \$match and \$project

### Optional Lab - Expressions with \$project

[Back to the Question](#)

One solution is below.

COPY

```
db.movies.aggregate([
  {
    $match: {
      cast: { $elemMatch: { $exists: true } },
      directors: { $elemMatch: { $exists: true } },
      writers: { $elemMatch: { $exists: true } }
    }
  },
  {
    $project: {
      _id: 0,
      cast: 1,
      directors: 1,
      writers: {
        $map: {
          input: "$writers",
          as: "writer",
          in: {
            $arrayElemAt: [
              {
                $split: ["$$writer", " ("]
              },
              0
            ]
          }
        }
      }
    }
  }
])
```

```

    $project: {
      labor_of_love: {
        $gt: [
          { $size: { $setIntersection: ["$cast", "$directors", "$writers"] }
        ],
        0
      ]
    }
  },
  {
    $match: { labor_of_love: true }
  },
  {
    $count: "labors of love"
  }
])

```

With our first **\$match** stage, we filter out documents that are not an array or have an empty array for the fields we are interested in.

```

{
  $match: {
    cast: { $elemMatch: { $exists: true } },
    directors: { $elemMatch: { $exists: true } },
    writers: { $elemMatch: { $exists: true } }
  }
},

```

 COPY

Next is a **\$project** stage, removing the **\_id** field and retaining both the **directors** and **cast** fields. We replace the existing **writers** field with a new computed value, cleaning up the strings within **writers**

```

{
  $project: {
    _id: 0,
    cast: 1,
    directors: 1,
    writers: {
      $map: {
        input: "$writers",
        as: "writer",
        in: {
          $arrayElemAt: [
            {
              $split: ["$$writer", " ("]
            },
            0
          ]
        }
      }
    }
  }
}

```

 COPY

```

    }
  }
}
}
}
}
},

```

We use another `$project` stage to computer a new field called `labor_of_love` that ensures the intersection of `cast`, `writers`, and our newly cleaned `directors` is greater than 0. This definitely means that at least one element in each array is identical! `$gt` will return true or false.

```

{
  $project: {
    labor_of_love: {
      $gt: [
        { $size: { $setIntersection: ["$cast", "$directors", "$writers"] }
      ],
      0
    }
  }
},

```

 COPY

Lastly, we follow with a `$match` stage, only allowing documents through where `labor_of_love` is **true**. In our example we use a `$match` stage, but `itcount()` works too.

```

{
  $match: { labor_of_love: true }
},
{
  $count: "labors of love"
}

// or

{
  $match: { labor_of_love: true }
}
]).itcount()

```

 COPY

This produces 1597, as expected.

Proceed to next section