DB Assignment 3

Abdullah Alshamrani

Date: 11/22/2024

For each query, we'll include:

1. The problem description.
2. A screenshot of the query and result.
3. A brief explanation.

Query 1: Over how many years was the unemployment data collected?

```
> use test
< already on db test
> /*
    Query 1: Over how many years was the unemployment data collected?
    This query uses the distinct() function to extract all unique years
    and calculates the total count.
  */

  db.unemployment.distinct("Year").length;
< 27
```

Explanation: This query retrieves all unique years in the dataset using distinct() and counts them.

Query 2: How many states were reported on in this dataset?

```
> /*
    Query 2: How many states were reported on in this dataset?
    This query extracts all unique states using distinct()
    and calculates the count.
  */

  db.unemployment.distinct("State").length;
< 47
```

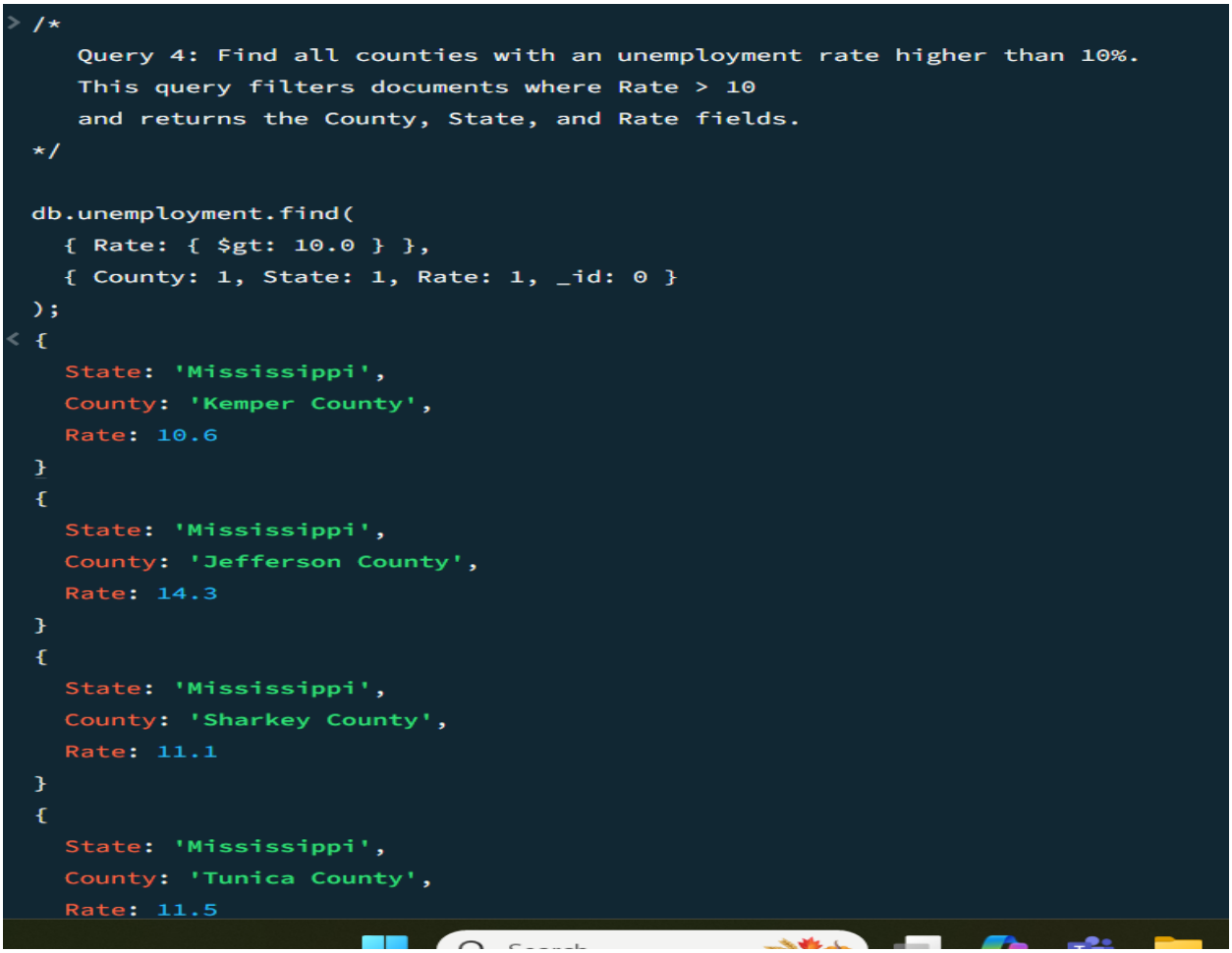
Explanation: This query fetches all unique state names from the dataset using distinct().

Query 3: What does this query compute?

```
> /* This query counts all documents where the unemployment rate (Rate) is less than 1.0% */
  db.unemployment.find({ Rate: { $lt: 1.0 } }).count();
< 657
```

Explanation: This query filters documents where the Rate field is less than 1.0 and counts them. 657 counties have an unemployment rate less than 1%.

Query 4: Find all counties with an unemployment rate higher than 10%



```
> /*
    Query 4: Find all counties with an unemployment rate higher than 10%.
    This query filters documents where Rate > 10
    and returns the County, State, and Rate fields.
  */

  db.unemployment.find(
    { Rate: { $gt: 10.0 } },
    { County: 1, State: 1, Rate: 1, _id: 0 }
  );
< {
    State: 'Mississippi',
    County: 'Kemper County',
    Rate: 10.6
  }
  {
    State: 'Mississippi',
    County: 'Jefferson County',
    Rate: 14.3
  }
  {
    State: 'Mississippi',
    County: 'Sharkey County',
    Rate: 11.1
  }
  {
    State: 'Mississippi',
    County: 'Tunica County',
    Rate: 11.5
```

Explanation: This query filters for Rate > 10 and projects only the relevant fields: County, State, and Rate. And it shows a list of counties with unemployment rates above 10%.

Query 5: Calculate the average unemployment rate across all states.

```
> /*
    Query 5: Calculate the average unemployment rate across all states.
    This query uses the $group stage to compute the average Rate.
  */

  db.unemployment.aggregate([
    {
      $group: {
        _id: null,
        averageRate: { $avg: "$Rate" }
      }
    }
  ]);
< {
    _id: null,
    averageRate: 6.1750097115006755
  }
```

Explanation: This query calculates the average value of the Rate field using the $avg aggregation operator. And it results in : 6.175% average unemployment rate.

Query 6: Find all counties with an unemployment rate between 5% and 8%

```
> /*
    Query 6: Find all counties with an unemployment rate between 5% and 8%.
    This query filters for rates between 5% and 8% and returns relevant fields.
  */

  db.unemployment.find(
    { Rate: { $gte: 5.0, $lte: 8.0 } },
    { County: 1, State: 1, Rate: 1, _id: 0 }
  );
< {
    State: 'Mississippi',
    County: 'Newton County',
    Rate: 6.1
  }
  {
    State: 'Mississippi',
    County: 'Monroe County',
    Rate: 7.9
  }
  {
    State: 'Mississippi',
    County: 'Hinds County',
    Rate: 6.1
  }
  {
    State: 'Mississippi',
    County: 'Calhoun County',
    Rate: 6.9
```

Explanation: This query uses a range filter to identify counties within the specified rate. And it results in a list of counties with unemployment rates between 5% and 8%.

Query 7: Find the state with the highest unemployment rate

```
> /*
    Query 7: Find the state with the highest unemployment rate.
    This query sorts the data by Rate in descending order and limits the result to 1.
  */

  db.unemployment.aggregate([
    { $sort: { Rate: -1 } },
    { $limit: 1 },
    { $project: { State: 1, Rate: 1, _id: 0 } }
  ]);
< {
    State: 'Colorado',
    Rate: 58.4
  }
```

Result: Colorado with a rate of 58.4%.
Explanation: This query sorts the data by Rate in descending order and retrieves the top result using $limit.

Query 8: Count how many counties have an unemployment rate above 5%

```
> /*
    Query 8: Count how many counties have an unemployment rate above 5%.
    This query counts all documents where Rate > 5.
  */

  db.unemployment.find({ Rate: { $gt: 5.0 } }).count();
< 510173
```

Result: 510,173 counties have unemployment rates above 5%.
Explanation: This query filters for Rate > 5 and uses .count() to count the matching documents.

Query 9: Calculate the average unemployment rate per state by year

```
> /*
    Query 9: Calculate the average unemployment rate per state by year.
    This query groups data by both State and Year and computes the average Rate.
  */

  db.unemployment.aggregate([
    {
      $group: {
        _id: { State: "$State", Year: "$Year" },
        averageRate: { $avg: "$Rate" }
      }
    },
    { $sort: { "_id.State": 1, "_id.Year": 1 } }
  ]);
< {
    _id: {
      State: 'Alabama',
      Year: 1990
    },
    averageRate: 8.226990049751244
  }
  {
    _id: {
      State: 'Alabama',
      Year: 1991
    },
    averageRate: 9.0818407960199
```

Explanation: This query uses $group to group by State and Year and computes the average rate for each group.

Extra Credit 10: Total unemployment rate across all counties per state

```
> /*
    Extra Credit 10: Total unemployment rate across all counties per state.
  */

  db.unemployment.aggregate([
    {
      $group: {
        _id: "$State",
        totalRate: { $sum: "$Rate" }
      }
    },
    { $sort: { totalRate: -1 } }
  ]);
< {
    _id: 'Texas',
    totalRate: 339866.2
  }
  {
    _id: 'Kentucky',
    totalRate: 263184.8
```

Result: Texas has the highest total unemployment rate: 339,866.2.
Explanation: This query calculates the sum of Rate across all counties grouped by State.

Extra Credit Query 11: Total unemployment rate for states with data from 2015 onward

```
> /*
    Extra Credit 11: Total unemployment rate for states with data from 2015 onwa
  */

  db.unemployment.aggregate([
    { $match: { Year: { $gte: 2015 } } },
    {
      $group: {
        _id: "$State",
        totalRate: { $sum: "$Rate" }
      }
    },
    { $sort: { totalRate: -1 } }
  ]);
< {
    _id: 'Texas',
    totalRate: 30114.4
  }
  {
    _id: 'Kentucky',
    totalRate: 18535.3
  }
  {
    _id: 'Virginia',
    totalRate: 15594.6
  }
```

Result: Texas leads with a total rate of 30114.4.
Explanation: This query filters data from 2015 onwards using $match and calculating the total unemployment rate for each state.