**MongoDB Queries**

1: Find the average salary for each combination of formatted\_experience\_level and work\_type

db.LinkedinJobAnalysisData.aggregate([

{

$group: {

\_id: {

experience\_level: "$formatted\_experience\_level",

work\_type: "$work\_type"

},

avg\_salary: {$avg: {$add: ["$max\_salary", "$min\_salary"] } }

}

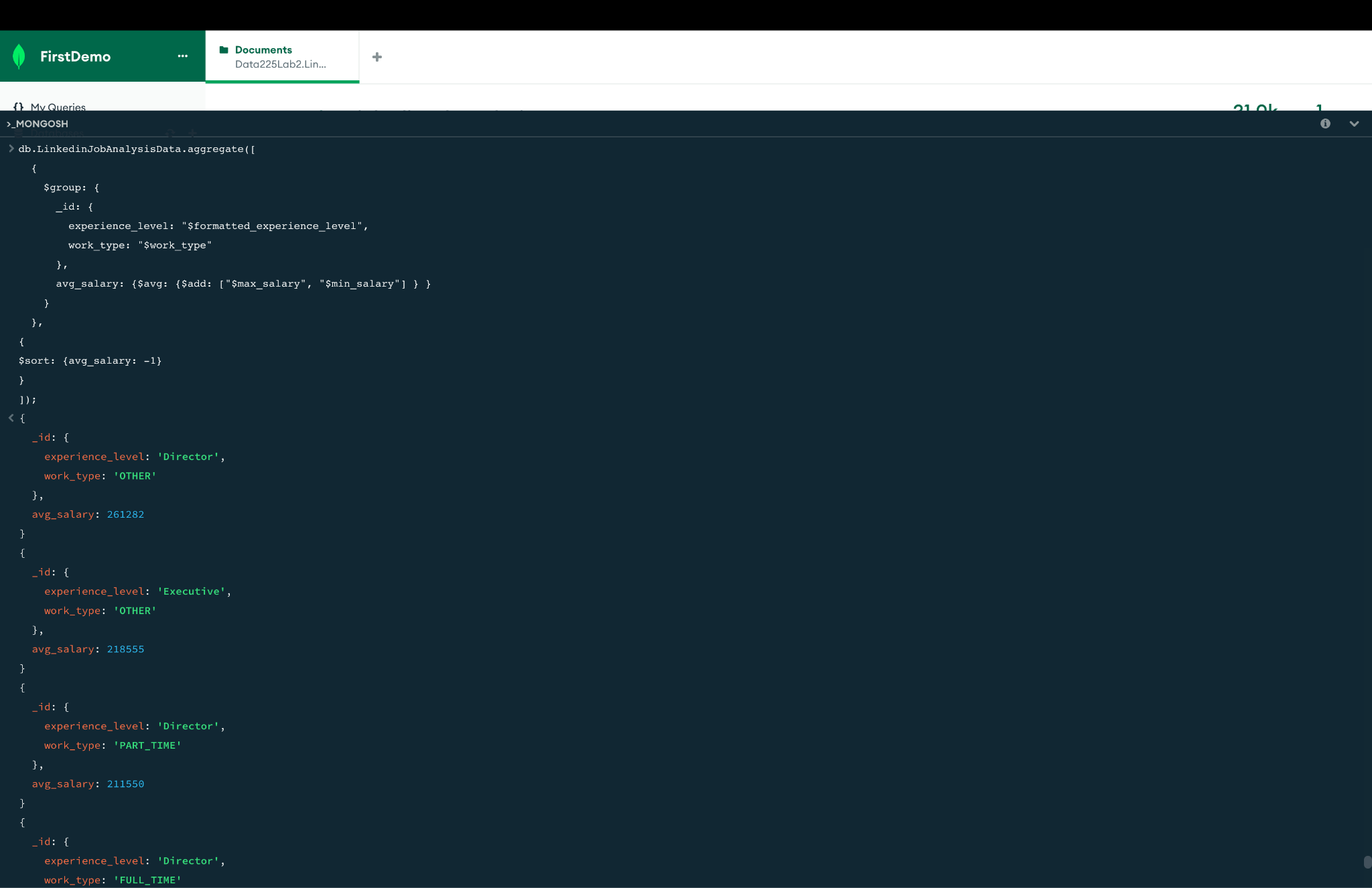
},

{

$sort: {avg\_salary: -1}

}

]);

This query can be used to find the average salary for each combination of experience level and work type. From this query we can say that experience level Director seems to be having the highest average salary with work type as other, followed by executive experience level.  
  


2: Most common specialty and which companies

db.LinkedinJobAnalysisData.aggregate([

{

$unwind: "$company.speciality"

},

{

$group: {

\_id: "$company.speciality",

companyCount: { $sum: 1 },

companies: { $addToSet: "$company.company\_name" }

}

},

{

$sort: { companyCount: -1 }

},

{

$limit: 1

},

{

$project: {

\_id: 0,

mostCommonSpeciality: "$\_id",

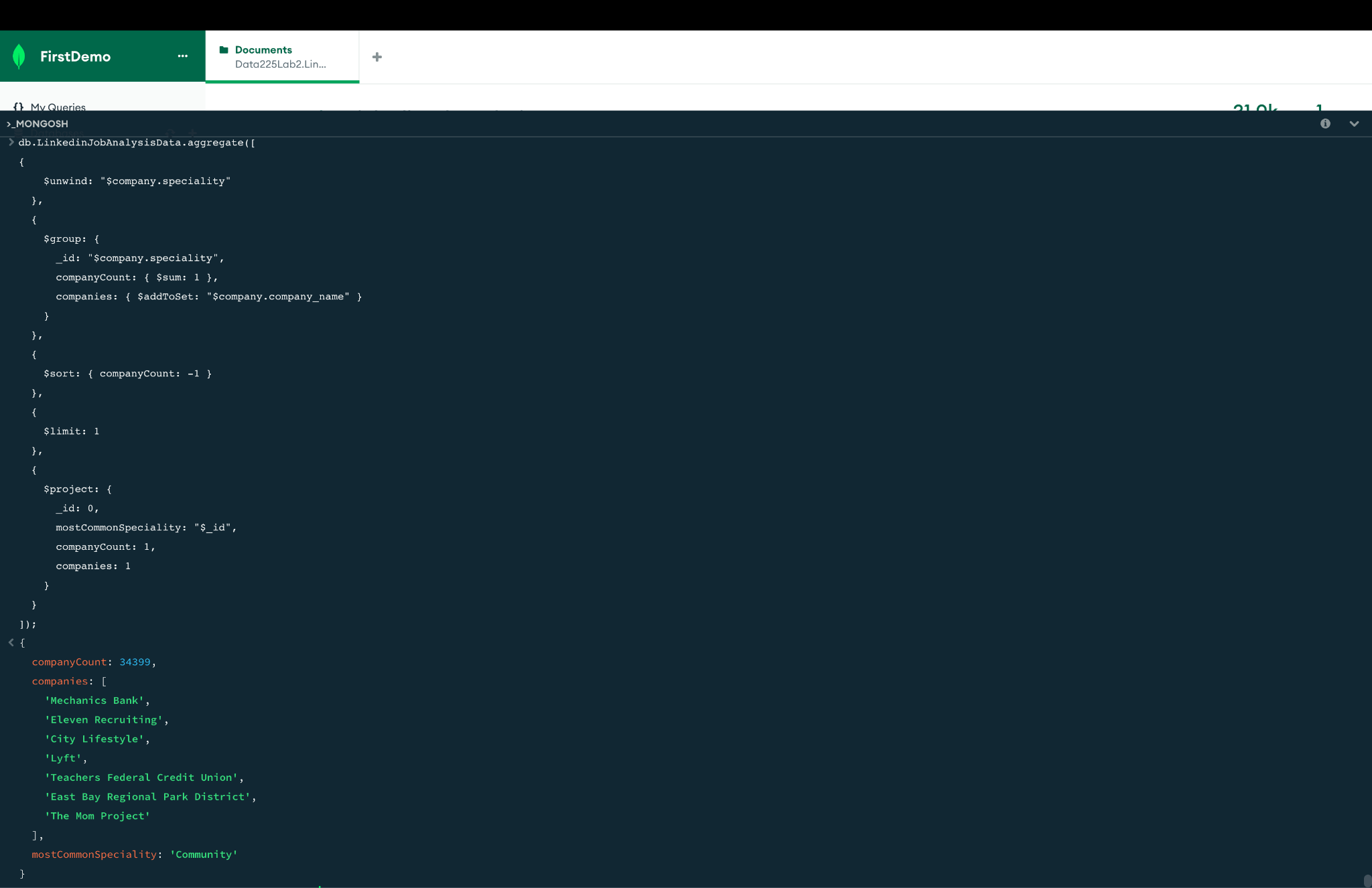
companyCount: 1,

companies: 1

}

}

]);

The above query can be used to find which specialty is most in demand and what all companies have this speciality.From the output , community is the most common speciality with companies like Lyft ,The Moms project, etc. having this specialty  
  


3: 10 Companies with the Most Job Listings:

db.LinkedinJobAnalysisData.aggregate([

{

$group: {

\_id: "$company\_details.company\_name",

jobCount: { $sum: 1 }

}

},

{

$sort: { jobCount: -1 }

},

{

$limit: 10

},

{

$project: {

\_id: 0, // Exclude the default \_id field

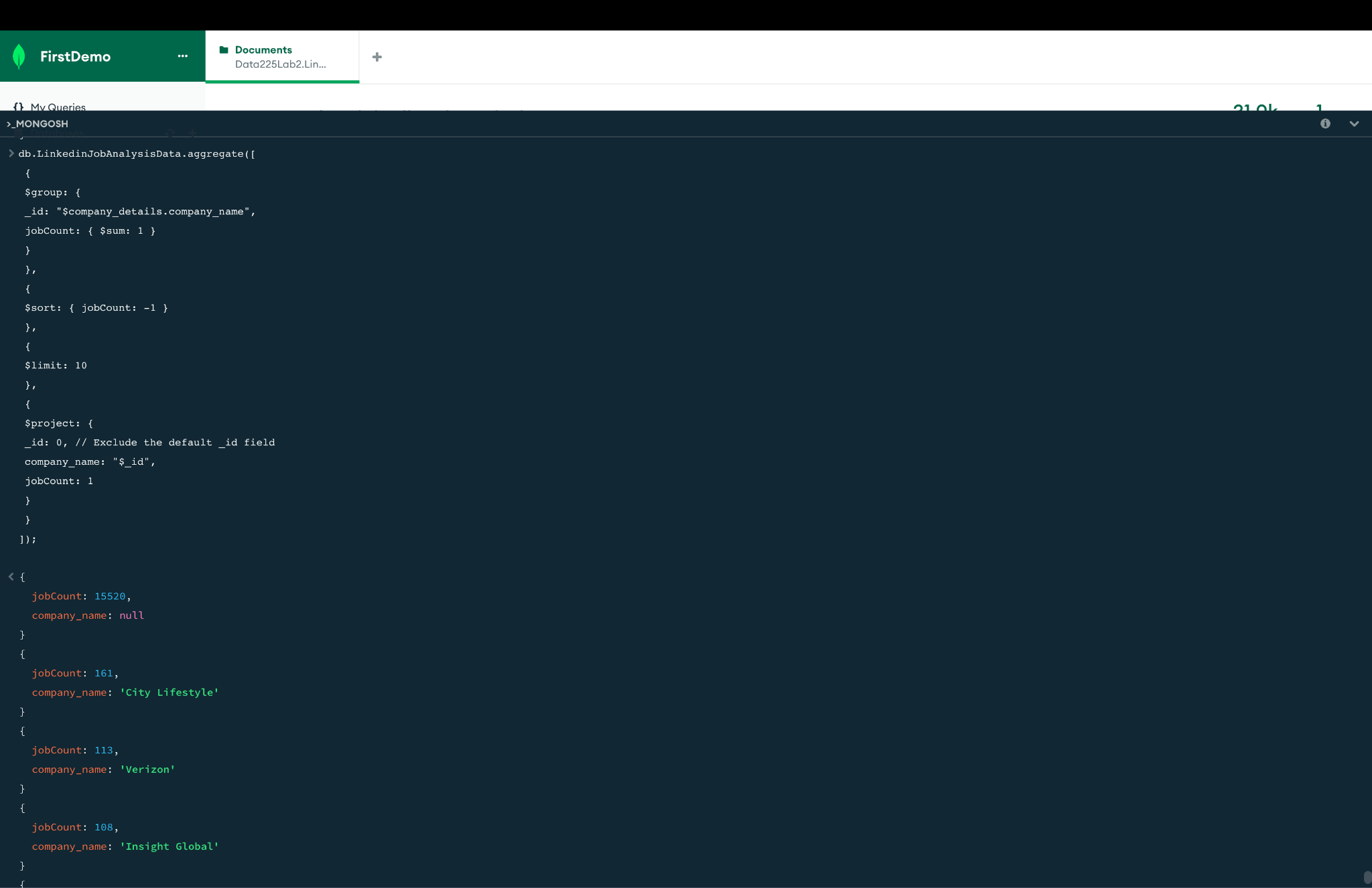
company\_name: "$\_id",

jobCount: 1

}

}

]);

The company that has the greatest number of postings are from the company ‘null’. From this we can tell that there are a lot of job postings where this company name is not mentioned. From the second company in the output is City Lifestyle which has a total postings of 161 offers.  
  


4:Ratio of the employee count to the applications

db.LinkedinJobAnalysisData.aggregate([

{

$match: {

$and: [

{ "company\_details.employee\_count": { $ne: null } },

{ "applies": { $ne: null } }

]

}

},

{

$group: {

\_id: "$company\_details.company\_name",

total\_employee\_count: { $first: "$company\_details.employee\_count" },

total\_applications: { $sum: "$applies" }

}

},

{

$project: {

\_id: 0,

company\_name: "$\_id",

employee\_to\_application\_ratio: { $divide: ["$total\_employee\_count", "$total\_applications"] }

}

},

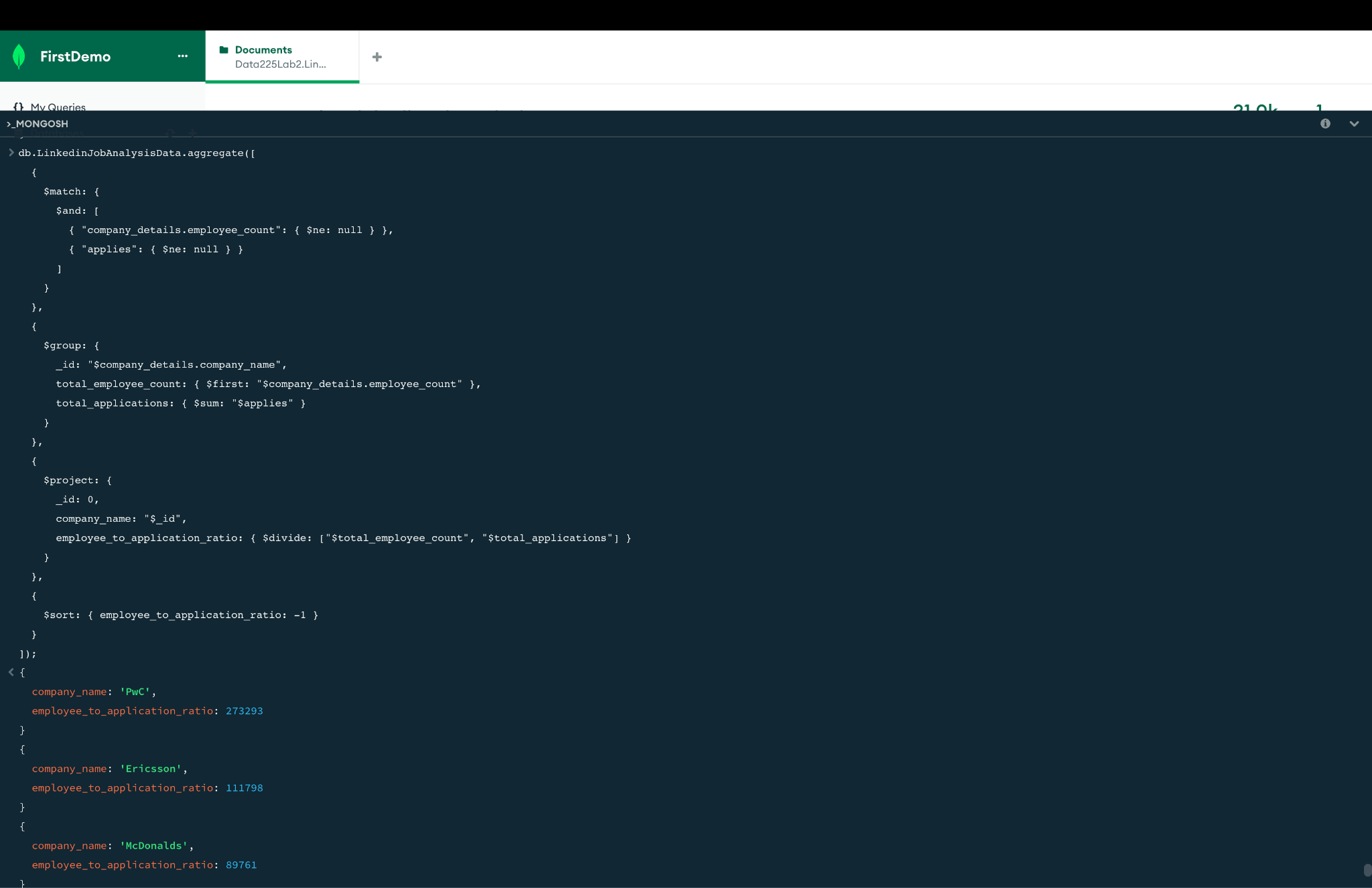
{

$sort: { employee\_to\_application\_ratio: -1 }

}

]);

This calculates the ratio of the employee count with the applications, here the company PWC has the highest ratio with 273293, followed by Ericsson, McDonald and Starbucks. However, it is important to keep in mind that if there are a lot of job postings this subsequently leads to a lot of job applications.





5:Job Postings with High Application-to-View Ratio

db.LinkedinJobAnalysisData.aggregate([

{

$match: {

"views": { $gt: 0 },

"applies": { $gt: 0 },

$expr: { $gt: ["$views", "$applies"] }

}

},

{

$project: {

\_id: 0,

job\_posting\_url: 1,

company\_name: "$company\_details.company\_name",

views: 1,

applies: 1,

application\_to\_view\_ratio: { $divide: ["$applies", "$views"] }

}

},

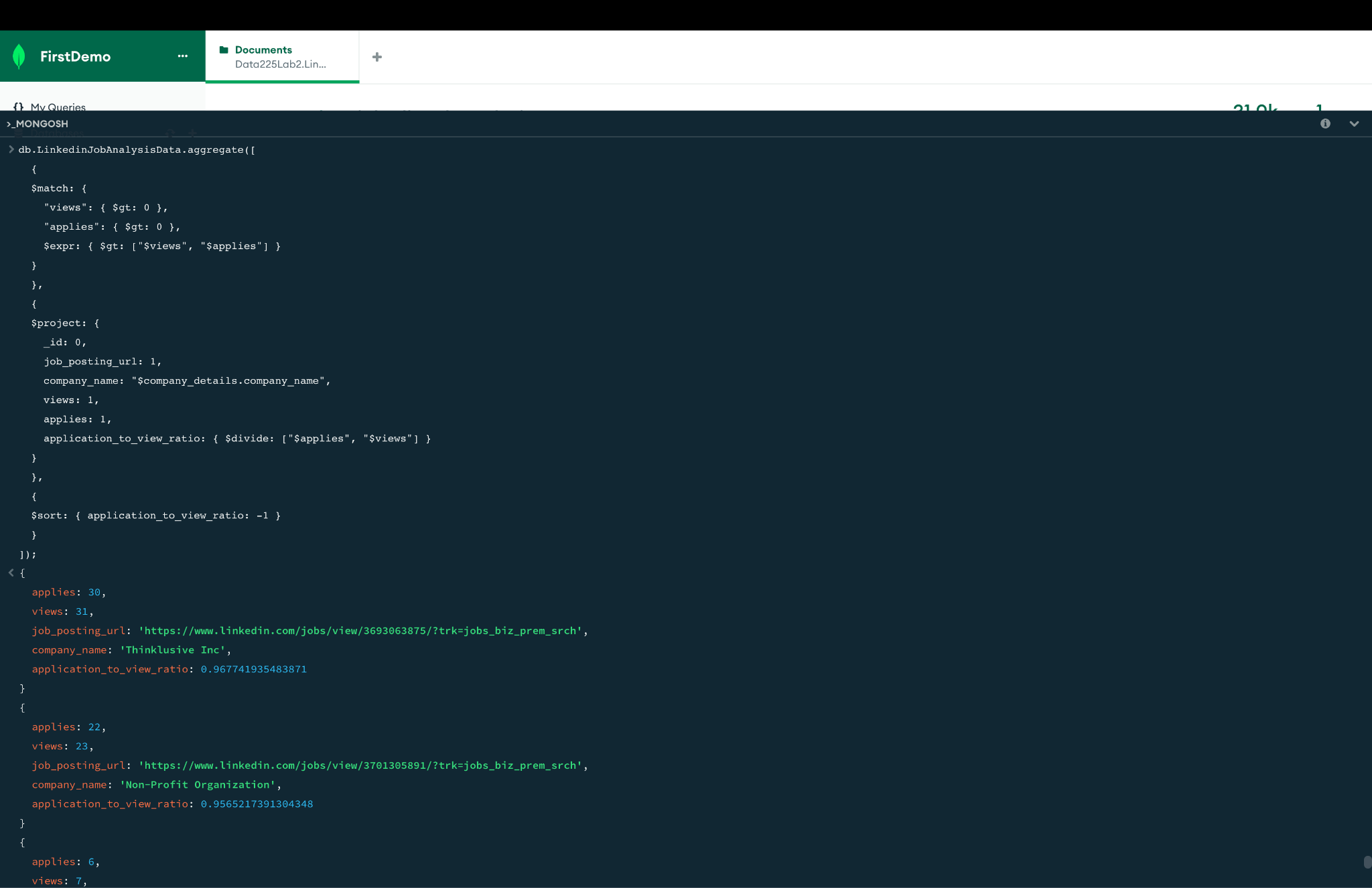
{

$sort: { application\_to\_view\_ratio: -1 }

}

]);

Similarly, like the above query this finds the ratio of Application to view of the job postings. Here we can understand the relationship of the views and the applications for that particular job posting.



6. Which state generates more jobs?

|  |
| --- |
| db.CompanyJobDetail.aggregate([{ |
| $unwind: "$company" |
| }, { |
| $group: { |
| \_id: "$company.state", |
| jobcount: { |
| $count: {} |
| } |
| } |
| }]).sort({ |
| jobcount: -1 |
| }) |

A computer screen with text

Description automatically generated

The Linkedin Data set shows California alone has the highest job requirement in the USA, and this requirement is greater than the combined requirement from Texas and New York, which are from the East coast.

7. Let us check which Job title has highest job requirement in California

|  |
| --- |
| db.CompanyJobDetail.aggregate([{ |
| $match: { |
| "job\_location": { |
| $regex: 'CA' |
| } |
| } |
| }, { |
| "$group": { |
| \_id: "$job\_title", |
| "jobcount": { |
| $count: {} |
| } |
| } |
| }, { |
| "$project": { |
| "\_id": 1, |
| "jobcount": 1 |
| } |
| }]).sort({ |
| jobcount: -1 |
| }) |
|  |

A white background with green and yellow text

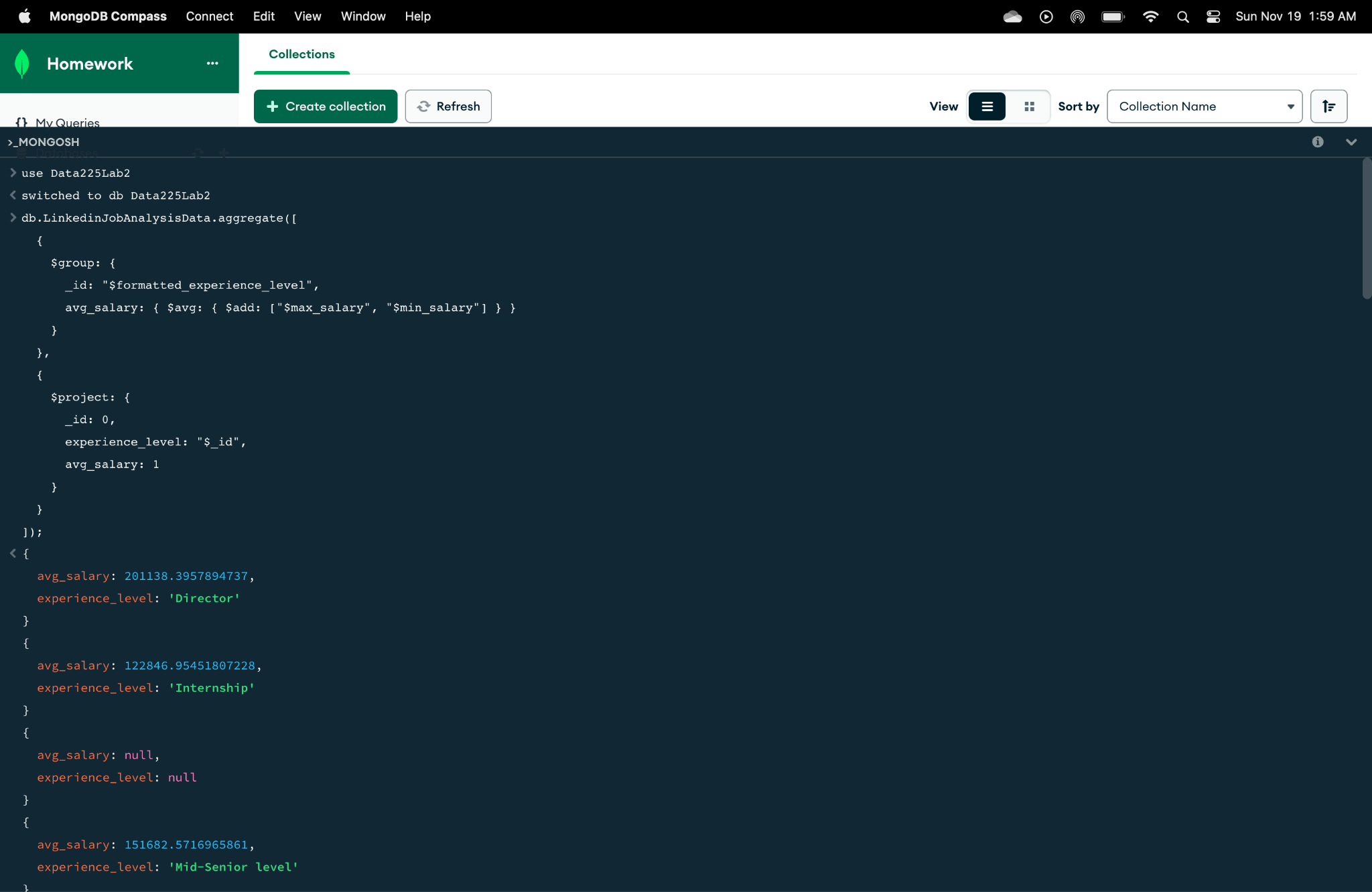
Description automatically generated

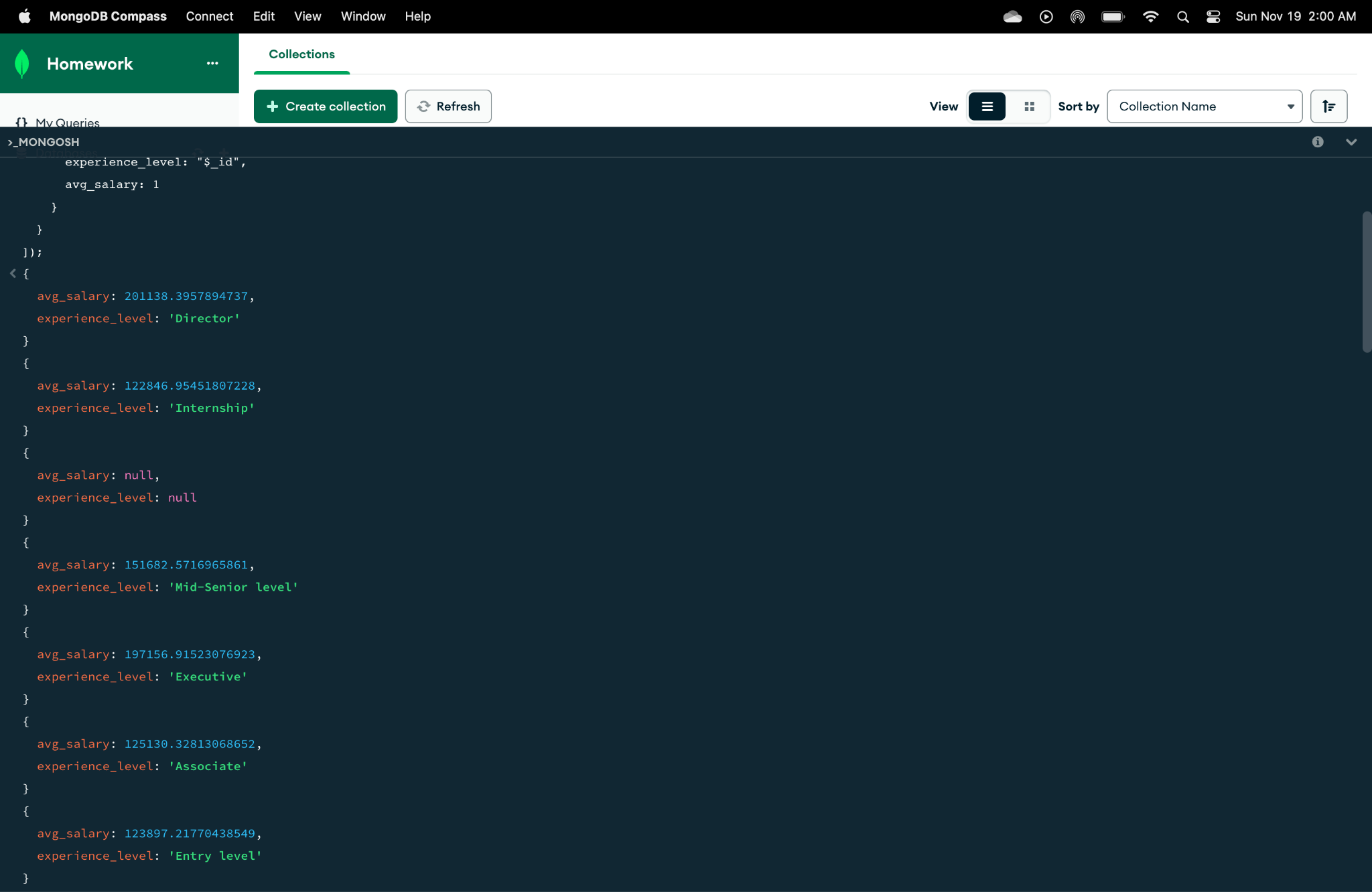
So the query result shows Sales Director(owner/operator) title has the highest job requirement in California state followed by Executive Assistant title.

8. Number of jobs posted from companies having highest followers. Are the companies having a high volume of followers generating the required amount of jobs?



|  |
| --- |
| { |
| db.JobPostFromCompany.aggregate([{ |
| $unwind: "$company\_details" |
| }, { |
| "$group": { |
| \_id: "$company\_details.company\_name", |
| maxfollower: { |
| $max: "$company\_details.follower\_count" |
| } |
| } |
| }, { |
| $sort: { |
| "maxfollower": -1 |
| } |
| }, { |
| $limit: 5 |
| }]) |
| } |

So, Amazon has the highest followers followed by Google and Unilever. We will explore how many jobs were created by these top following companies.



This query uses the LinkedinJobAnalysisData collection to calculate the average salary for each experience level. The average salary is determined by taking the mean of the minimum and maximum incomes for each level, and it groups job posts based on formatted\_experience\_level. Following that, a projection of the results shows the average salary and level of experience. This approach offers insight into the ways in which salary expectations vary among various job market experience levels.

10. Finds the most common job title

db.LinkedinJobAnalysisData.aggregate([

{

$group: {

\_id: "$job\_title",

count: { $sum: 1 }

}

},

{

$sort: { count: -1 }

},

{

$limit: 1

},

{

$project: {

\_id: 0,

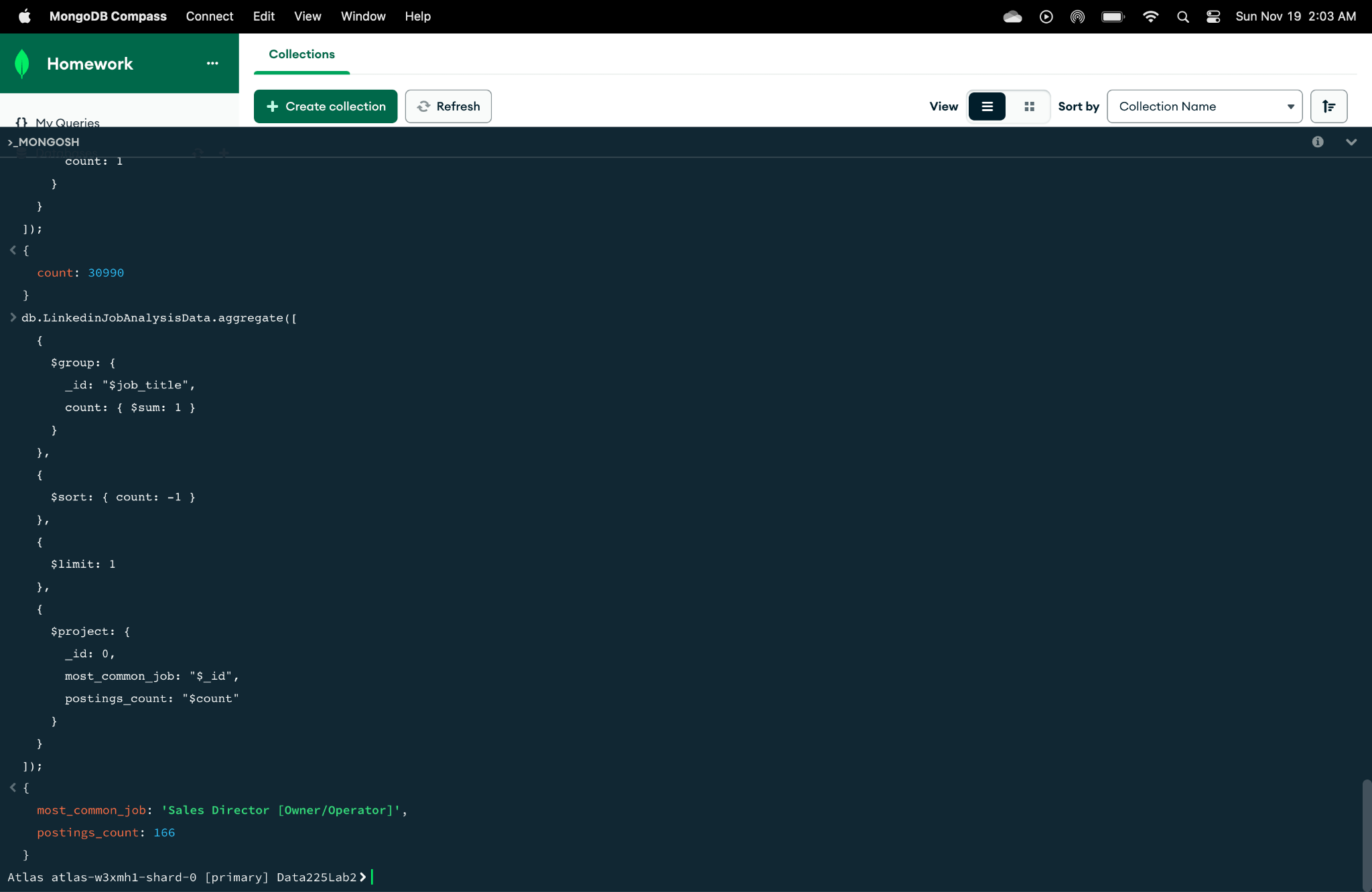
most\_common\_job: "$\_id",

postings\_count: "$count"

}

}

]);



The above query identifies the most common job title in the `LinkedinJobAnalysisData` collection. It counts the occurrences of each title and groups the data by 'job\_title'. This count is then used to arrange the results in decreasing order. In order to highlight the job title that appears most frequently in the dataset, the query restricts the output to the single most frequent job title and provides it along with the total number of postings for that title.

11.Identify Job Titles with the Most Significant Salary Differences

db.LinkedinJobAnalysisData.aggregate([

{

$group: {

\_id: "$job\_title",

avg\_salary\_difference: { $avg: { $subtract: ["$max\_salary", "$min\_salary"] } }

}

},

{

$sort: { avg\_salary\_difference: -1 }

},

{

$limit: 10

},

{

$project: {

\_id: 0,

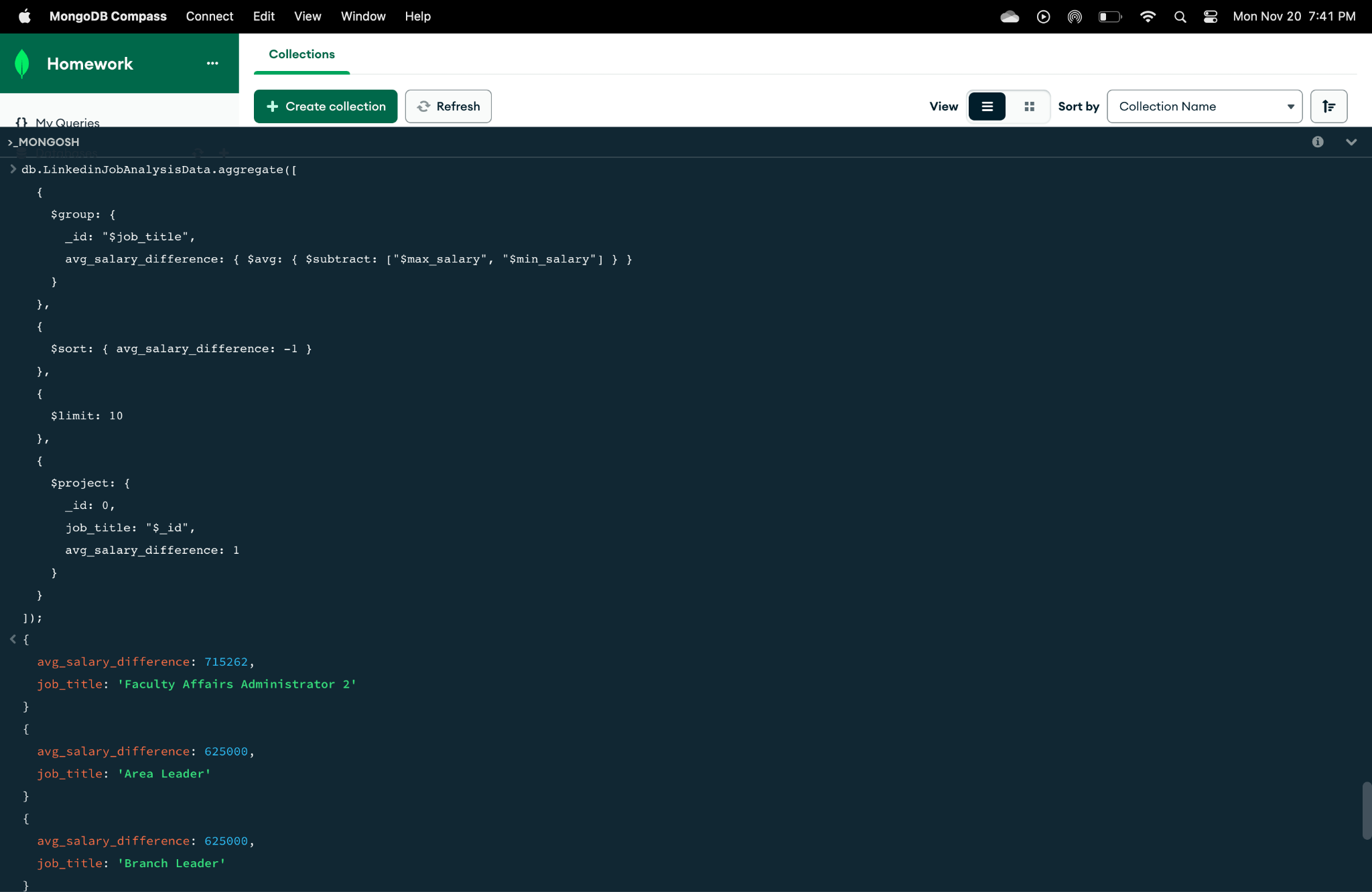
job\_title: "$\_id",

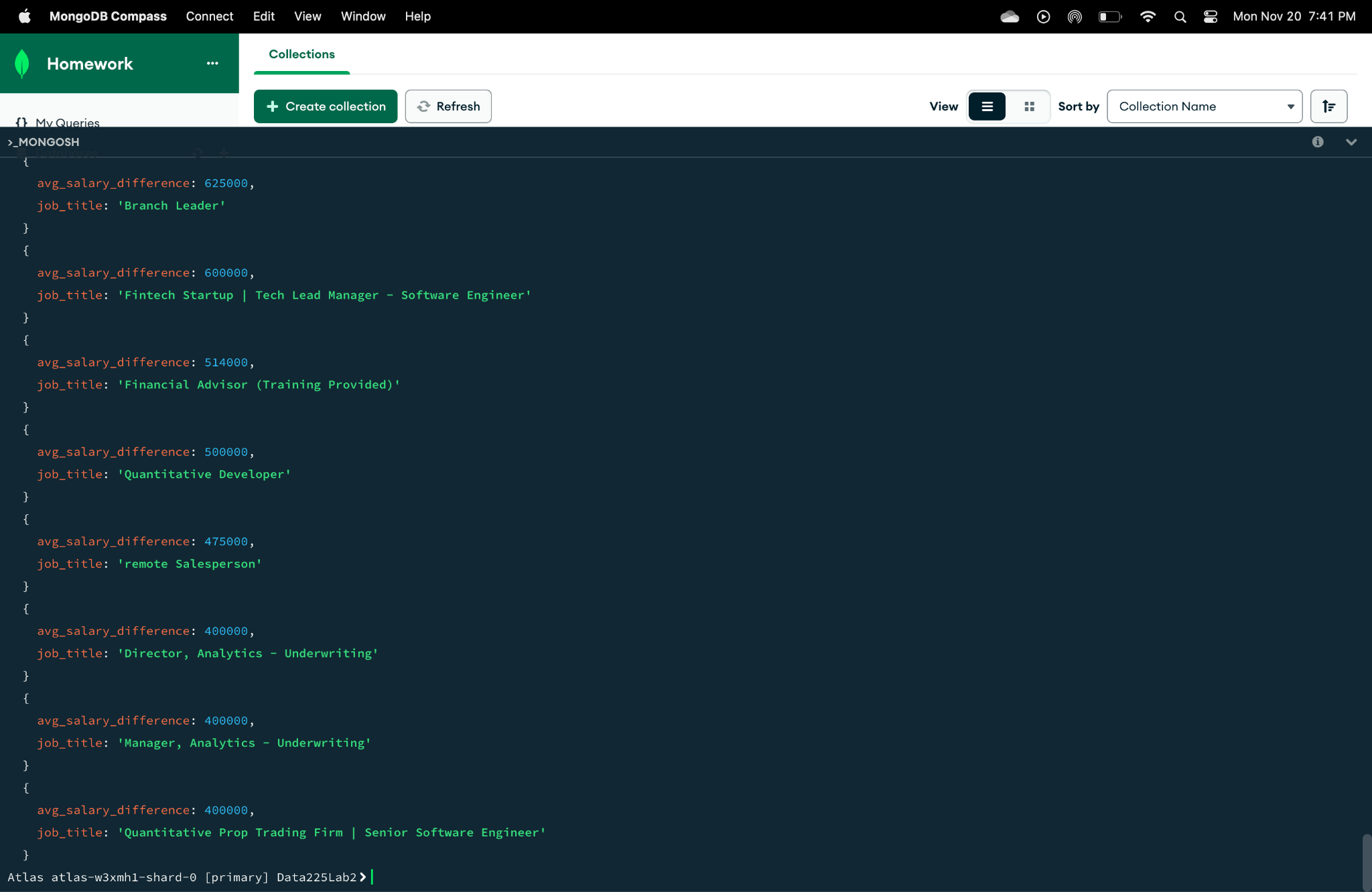
avg\_salary\_difference: 1

}

}

]);





In order to determine which job titles have the biggest disparities between their maximum and minimum salary offers, this query examines the 'LinkedinJobAnalysisData' collection. For every job title, the wage differential is computed and sorted in descending order. By limiting the output to the top 10, the query highlights the job titles with the greatest salary variability. This approach helps identify the roles with the greatest salary ranges available in the job market.

12: Calculate the average salary for jobs in each country and work type combination.

db.LinkedinJobAnalysis.aggregate([

{

$group: {

\_id: { country: "$company\_details.country", work\_type: "$formatted\_work\_type" },

avg\_salary: { $avg: "$max\_salary" }

}

},

{

$project: {

country: "$\_id.country",

work\_type: "$\_id.work\_type",

avg\_salary: 1,

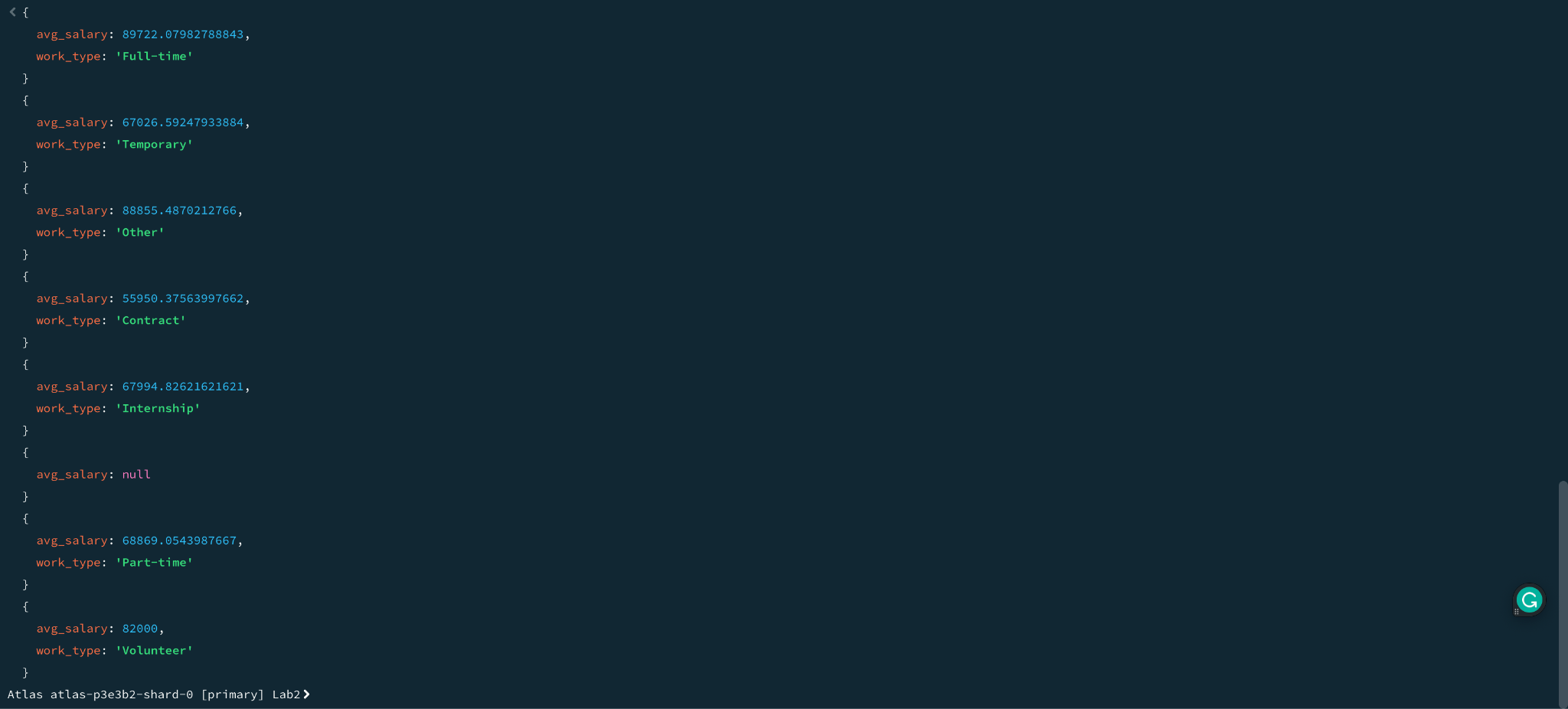
\_id: 0

}

}

]);





Explanation : This query performs an aggregation operation on the `LinkedinJobAnalysis` collection. It groups the data by `country` and `work\_type`, and calculates the average `max\_salary` for each group. The result is a list of countries, work types, and their corresponding average salaries.

13:Identify companies with the highest average number of job views for their postings.

db.LinkedinJobAnalysis.aggregate([

{

$group: {

\_id: "$company\_details.company\_name",

avg\_views: { $avg: "$views" }

}

},

{

$sort: { avg\_views: -1 }

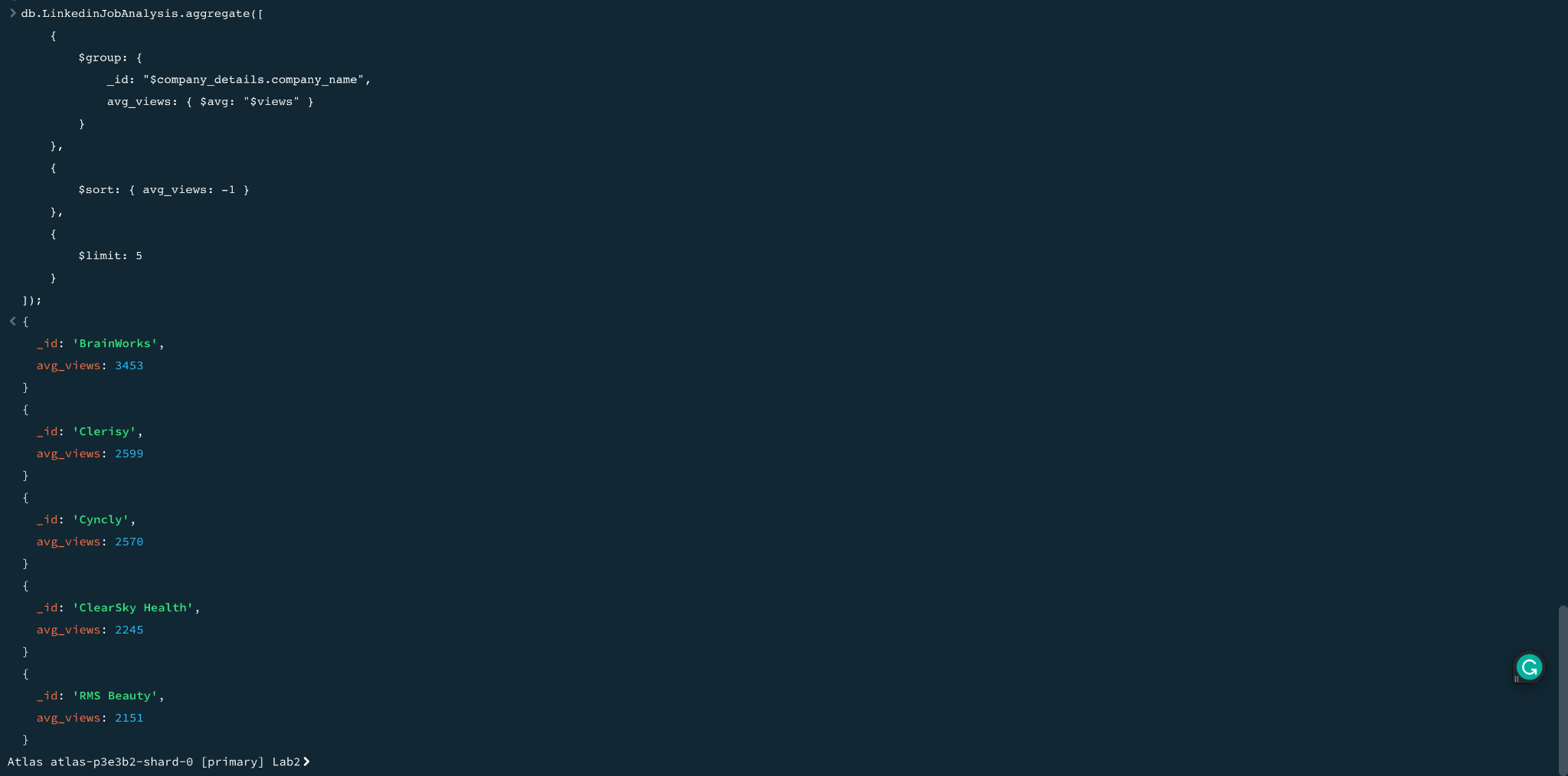
},

{

$limit: 5

}

]);



Explanation : The query groups the data by company\_name and calculates the average number of views for each company. The results are then sorted in descending order of avg\_views. Finally, it limits the output to the top 5 companies with the highest average views.

14:Find job titles and locations with the highest median salary, ordered by median salary in descending order.

db.LinkedinJobAnalysis.aggregate([

{

$sort: { "med\_salary": -1 }

},

{

$project: {

job\_title: 1,

location: 1,

med\_salary: 1

}

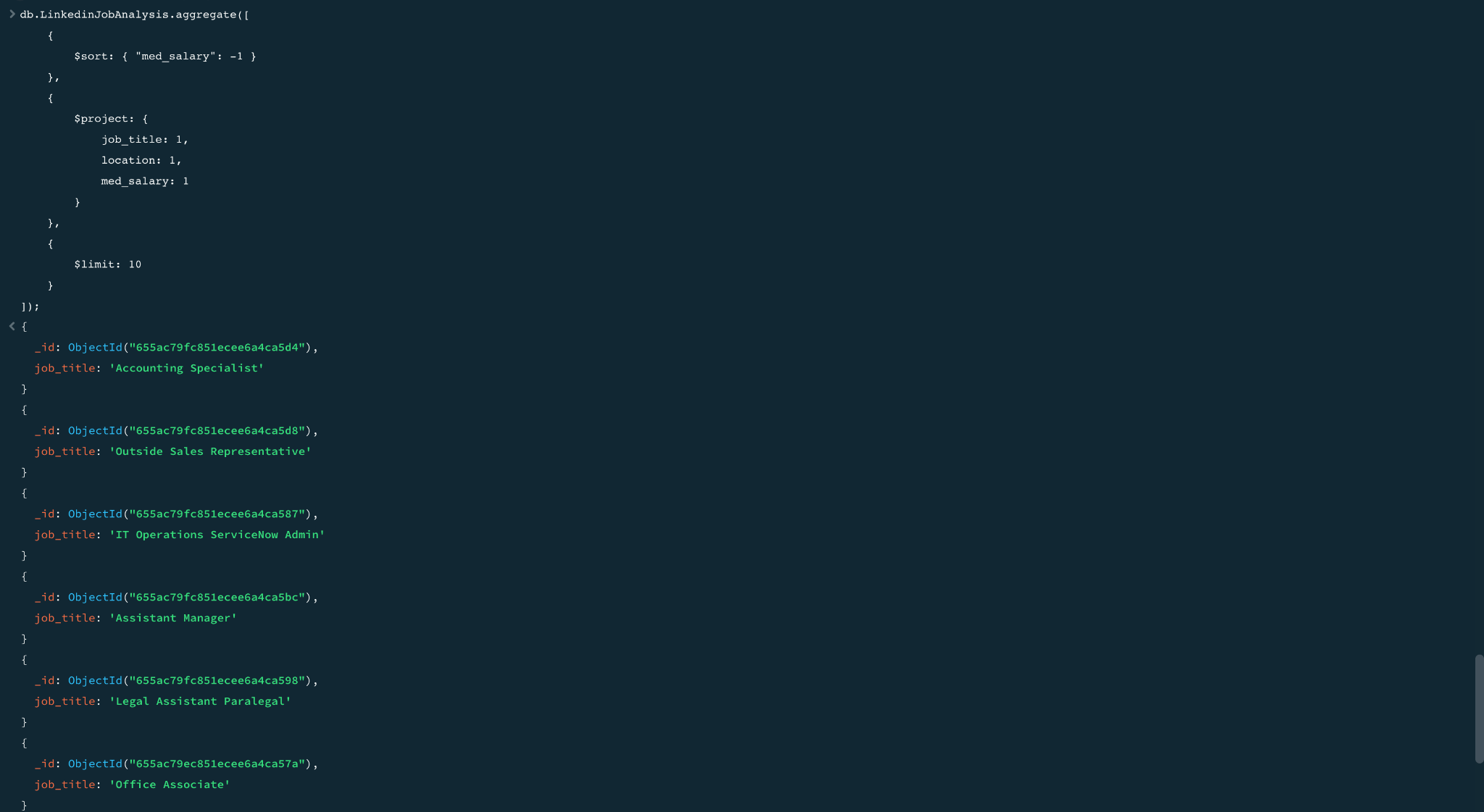
},

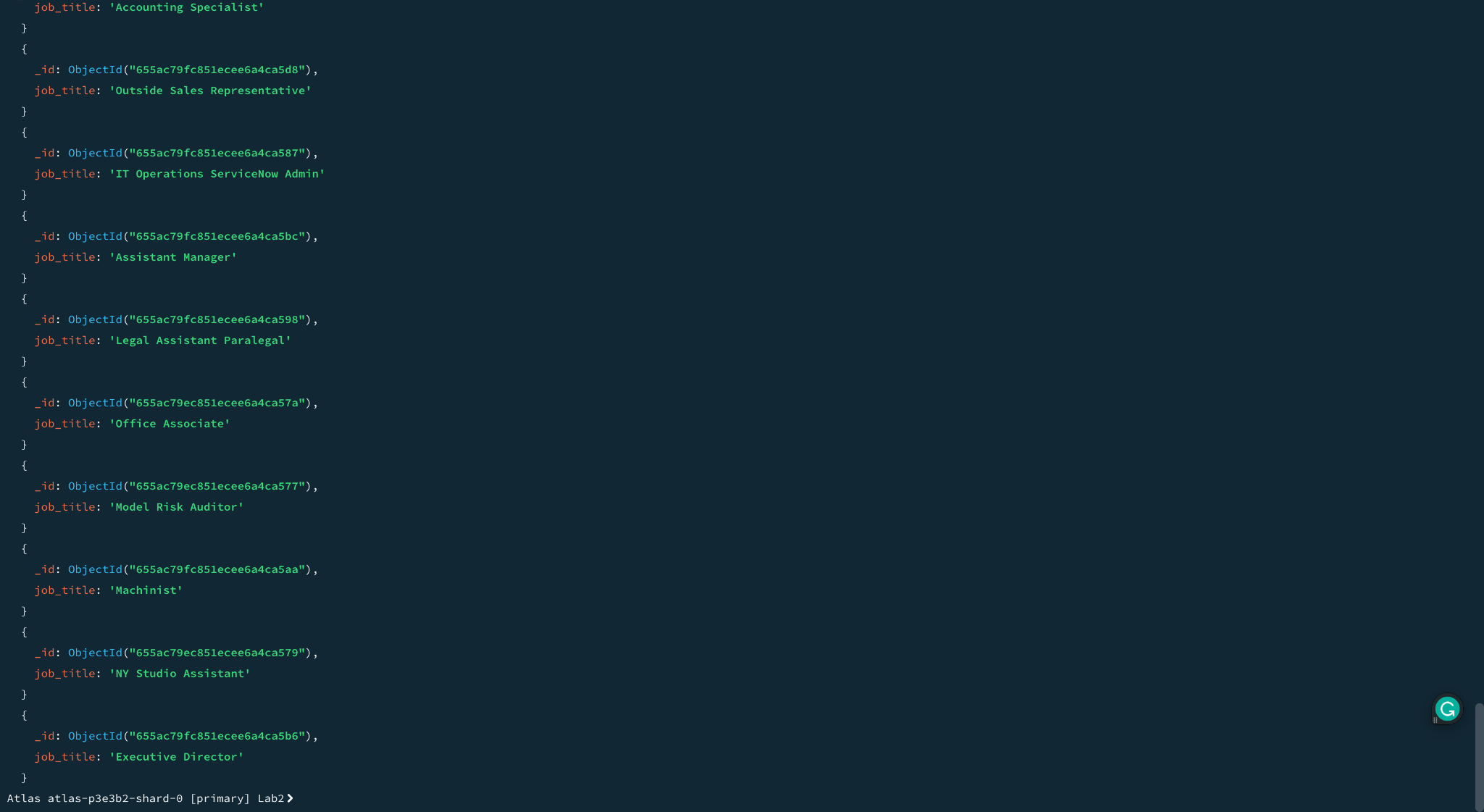
{

$limit: 10

}

]);





Explanation : The query sorts the data in descending order based on the `med\_salary` field. The query then projects or selects the `job\_title`, `location`, and `med\_salary` fields from the sorted data. Finally, it limits the output to the top 10 records with the highest median salaries.

15: List Job Titles with High Views-to-Applications Ratio (Retrieve job titles with a high ratio of views to applications, indicating high interest relative to the number of applications.)

db.LinkedinJobAnalysis.aggregate([

{

$match: {

applies: { $gt: 0 }

}

},

{

$project: {

job\_title: 1,

views\_to\_applications\_ratio: { $divide: ["$views", "$applies"] }

}

},

{

$sort: { views\_to\_applications\_ratio: -1 }

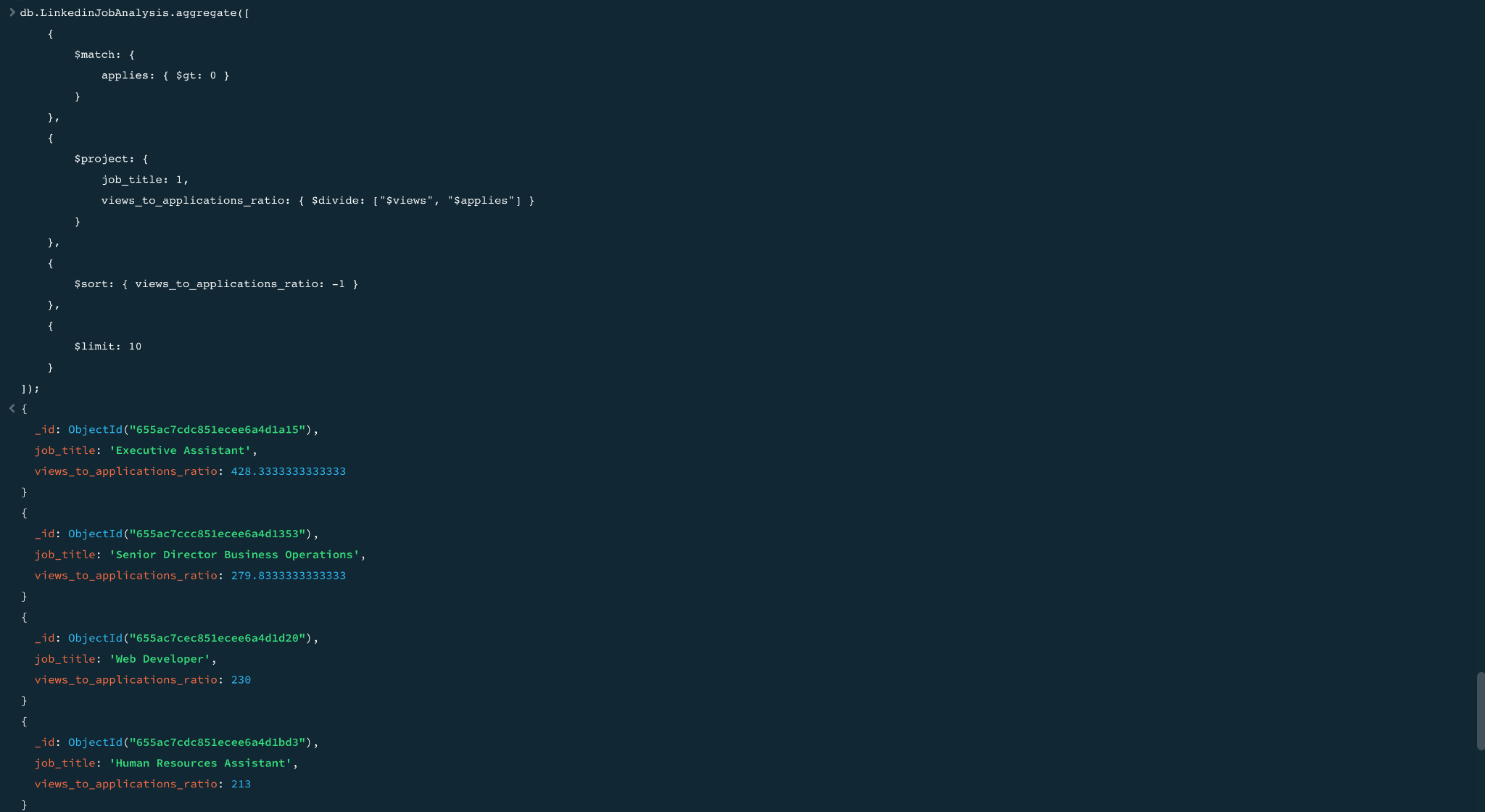
},

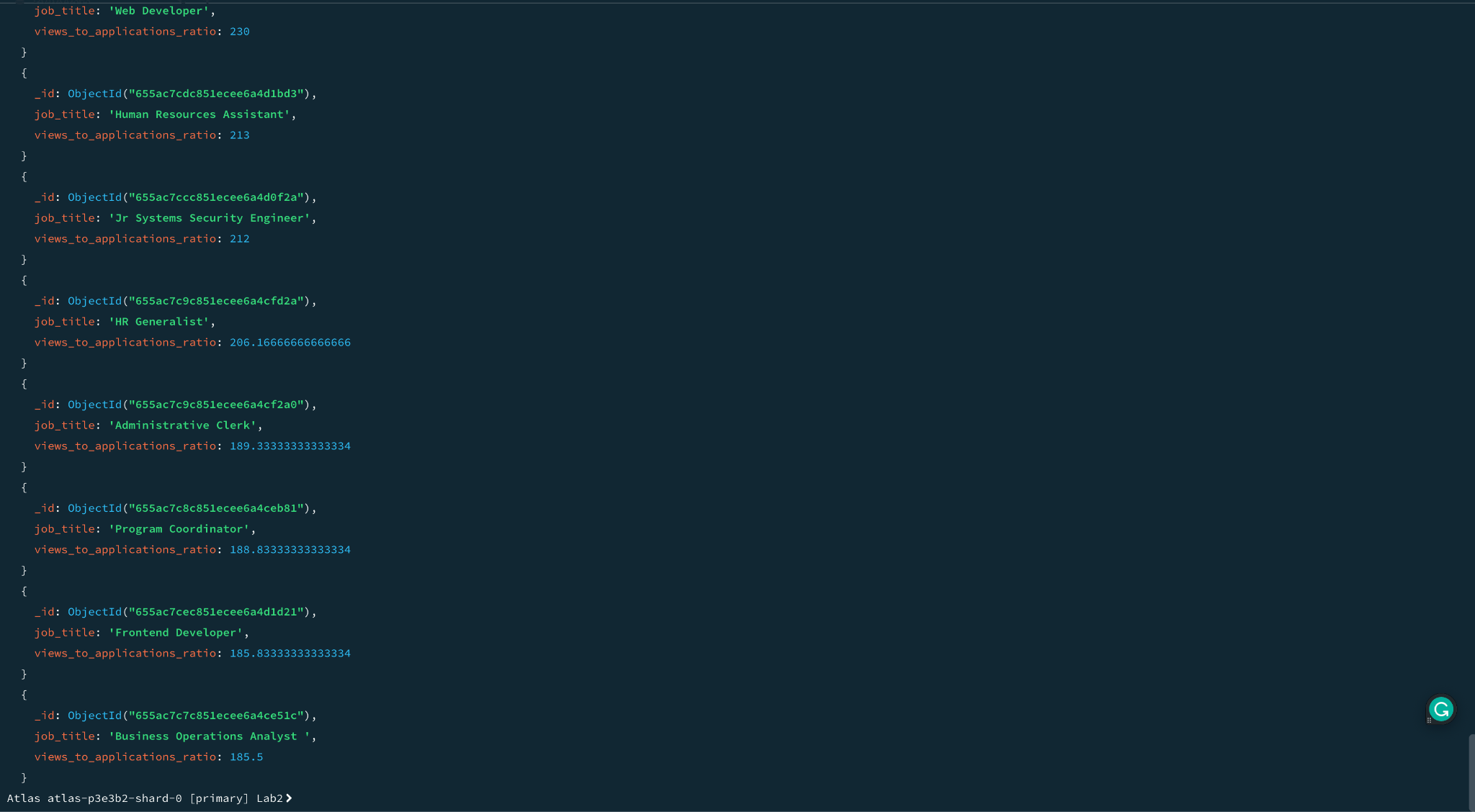
{

$limit: 10

}

]);





Explanation : This query first filters the data to include only those records where `applies` is greater than 0. Then, it calculates the ratio of `views` to `applies` for each job and includes the `job\_title` in the output. The results are sorted in descending order based on the `views\_to\_applications\_ratio`. Finally, it limits the output to the top 10 records with the highest views-to-applications ratio.

16:Identify Job Titles with the Highest Increase in Applications (Find job titles with the highest percentage increase in the number of applications compared to the previous data collection.)

db.LinkedinJobAnalysis.aggregate([

{

$sort: { "applies": -1 }

},

{

$group: {

\_id: "$job\_title",

previous\_applies: { $first: "$applies" },

current\_applies: { $first: "$applies" }

}

},

{

$project: {

job\_title: "$\_id",

percentage\_increase: { $multiply: [{ $divide: [{ $subtract: ["$current\_applies", "$previous\_applies"] }, "$previous\_applies"] }, 100] }

}

},

{

$sort: { percentage\_increase: -1 }

},

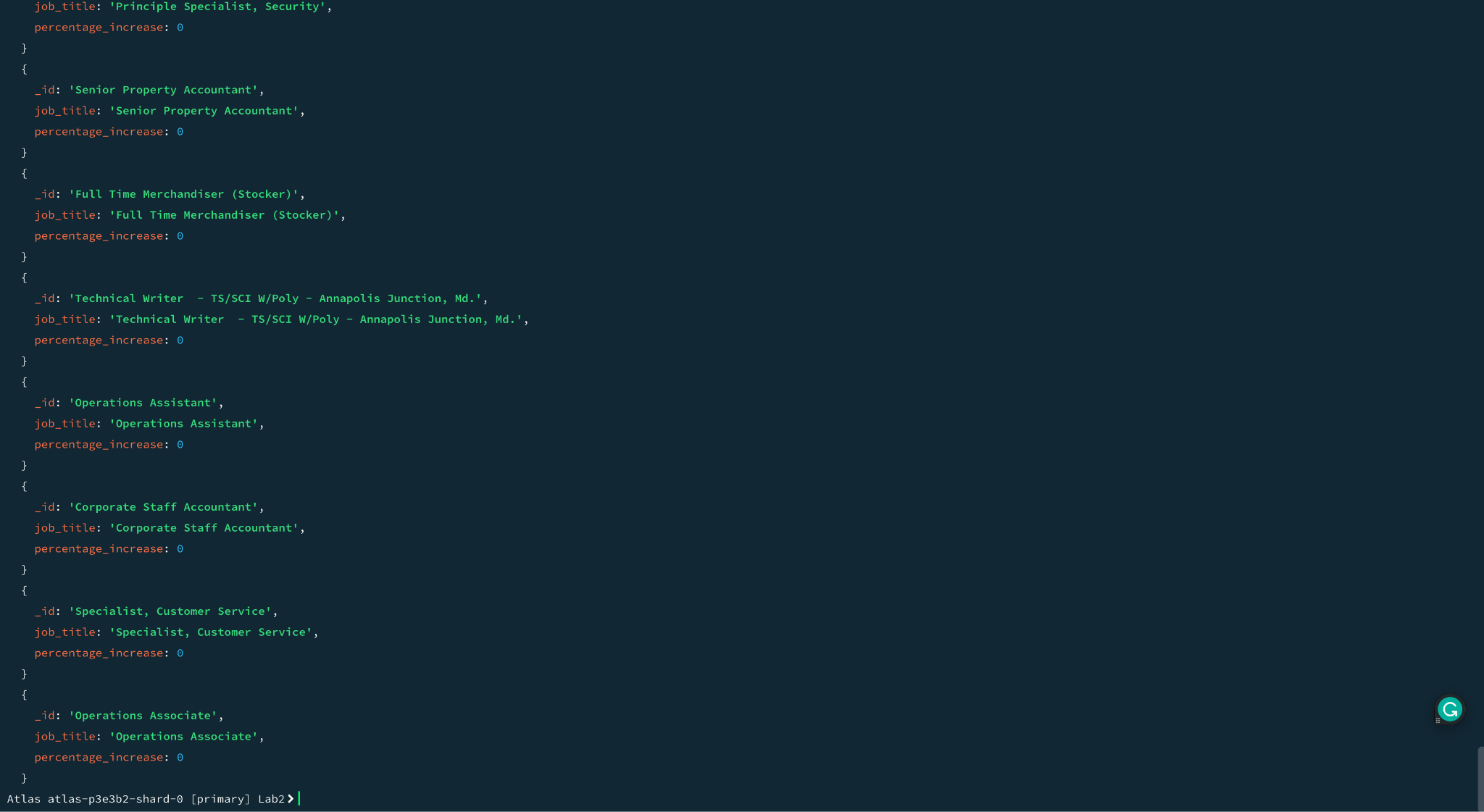
{

$limit: 10

}

]);





Explanation : The query sorts the documents based on the number of job applications in descending order. Then, it groups the documents by job title and retains the first document's number of applies as both the previous and current applies for each group. After that, it calculates the percentage increase in job applications for each job title. The documents are then sorted in descending order based on this calculated percentage increase. Finally, the output is limited to the top 10 documents with the highest percentage increase in job applications.

17. Display Companies names job titles that have least views and applies ( for HR to know how a company should not be and for candidates to choose jobs wid least competition ) ( table : jobs\_T + company\_T ) :

db.Jobs\_T.aggregate([

{

$lookup:

{

from: "Company\_T",

localField: "company.company\_name",

foreignField: "company\_details.company\_name",

as: "company\_info"

}

},

{

$unwind: "$company\_info"

},

{

$project:

{

\_id: 0,

company\_name: "$company.company\_name",

job\_title: 1,

views: "$company\_info.views",

applies: "$company\_info.applies"

}

},

{

$sort:

{

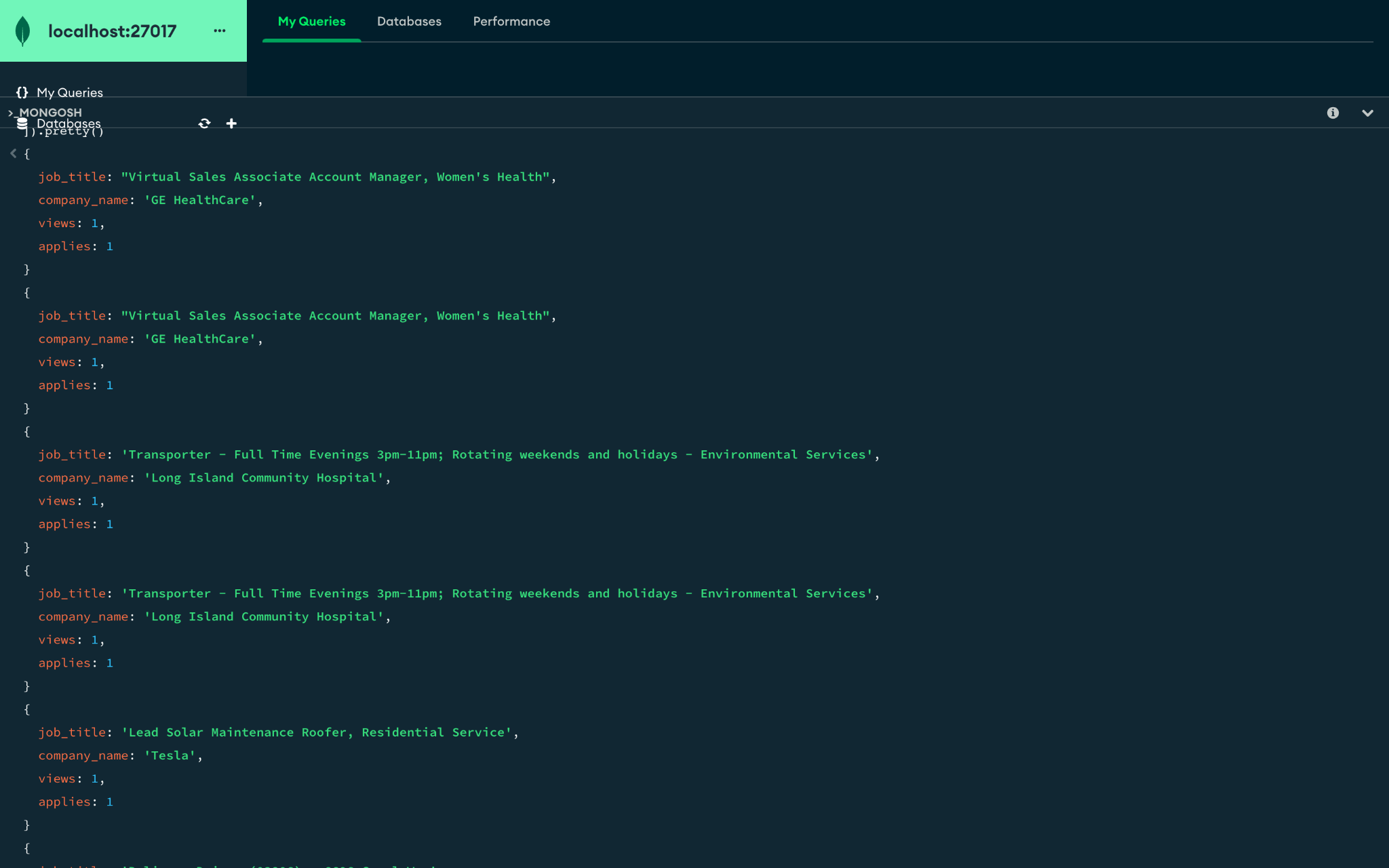
views: 1,

applies: 1

}

}

]).pretty()



18. To know which unique titles are costing the most from sum of their max salarys in all the companies (who is eating most of the money ( write it in diplomatic way ) ) (table: jobs\_T ) :

db.Jobs\_T.aggregate([

{

$group:

{

\_id: "$job\_title",

total\_max\_salary: { $sum: "$max\_salary" }

}

},

{

$sort:

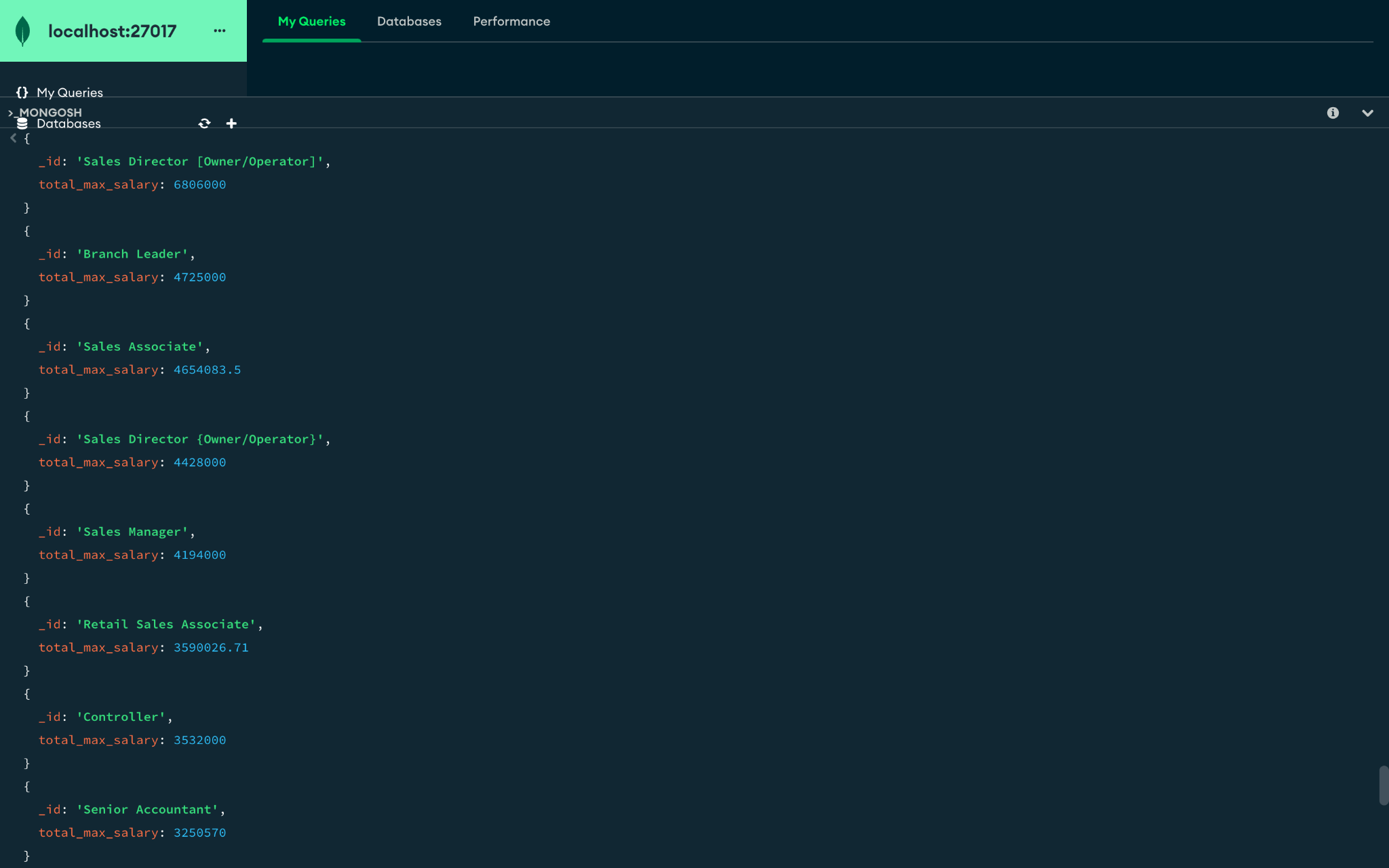
{

total\_max\_salary: -1

}

}

]).pretty()



19. Which job titles are least posted in all the companies (these workers can be outsourced )( table : jobs\_T ) :

db.Jobs\_T.aggregate([

{

$group:

{

\_id: "$job\_title",

count: { $sum: 1 }

}

},

{

$sort:

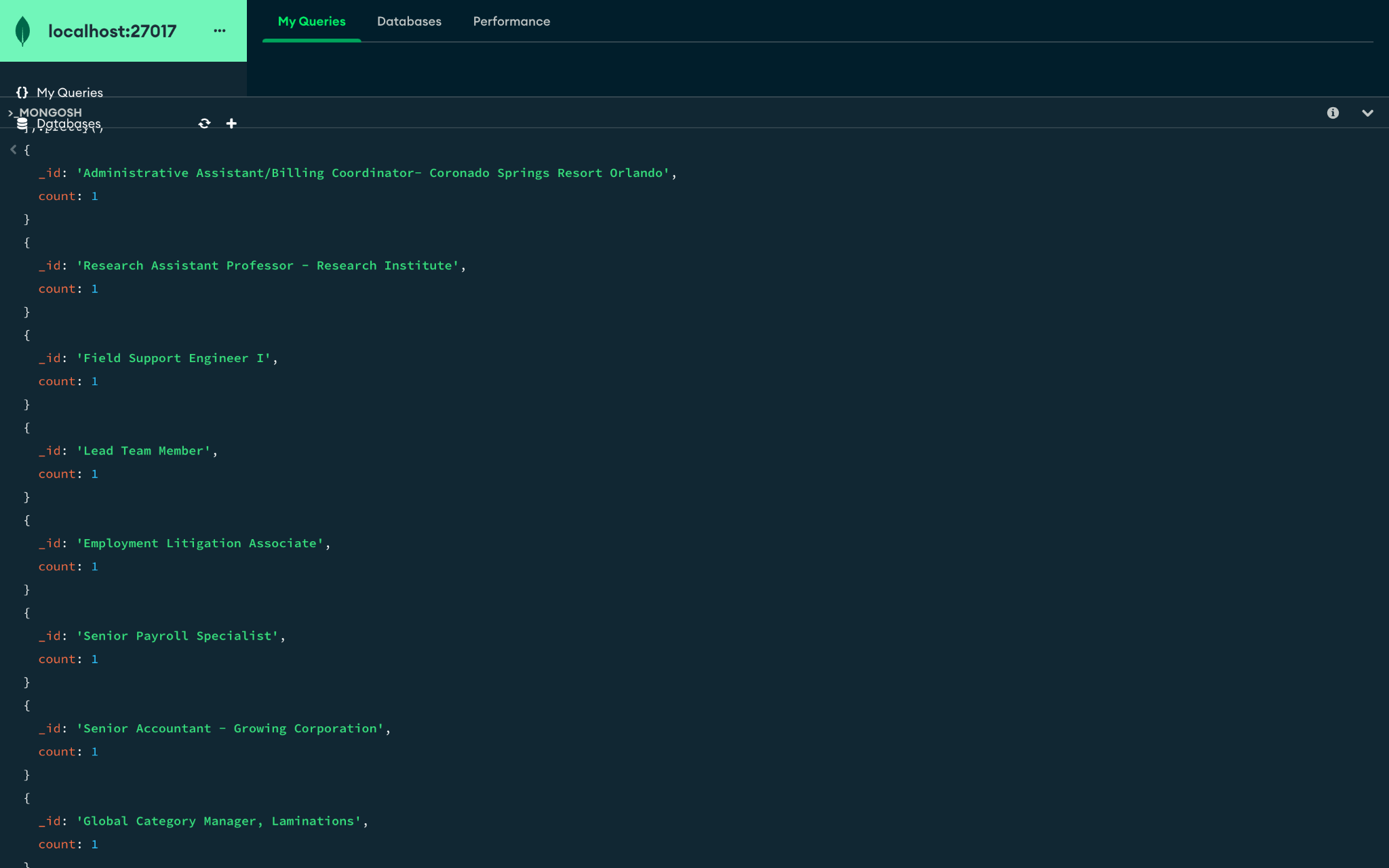
{

count: 1

}

}

]).pretty()



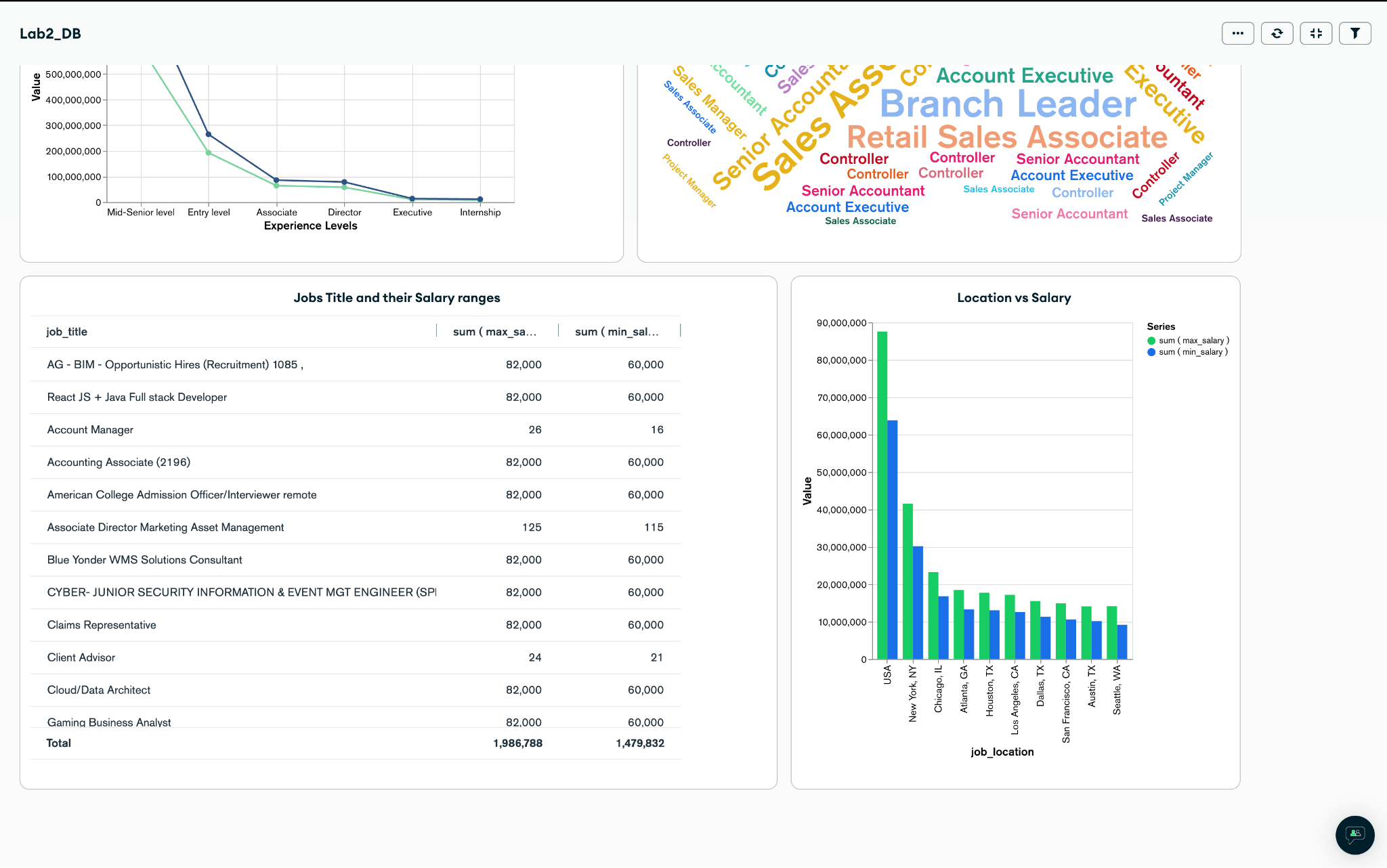
Data Visualization:

**DASHBOARD**

This dashboard is a comprehensive tool designed for project managers to track and analyze their project's progress and performance. It provides a visual representation of various key metrics related to job applications, job vacancies, and salary ranges.

****

****

****

**Link :**

<https://charts.mongodb.com/charts-project-0-hocab/public/dashboards/6ac354c5-8fb2-4b65-afa3-78a64c99edad>

**Features:**

**- Application Type based on Work Category**: This bar graph provides insights into the number of applications received for different work categories.

**- Map view of countries with max vacancies**: This map gives a geographical representation of job vacancies, highlighting the countries with the most job openings.

**- Salary Range based on Experience Level:** This line graph shows the salary range for different positions, providing a clear view of compensation based on experience levels.

**- Project Manager Position based on Salary Range:** This bar graph presents the salary range for different positions within the project management domain.

**Usage**

To use this dashboard, simply navigate through the various charts and graphs. Hover over specific data points for more detailed information. This tool is designed to aid project managers in making informed decisions based on real-time data.