

LAB 2: Linked In Job Postings MQL Queries

1. List of jobs that have 'analyst' in the title, are full-time and based in San Jose, CA

Query:

```
db.job_postings.find({ title: /analyst/i, formatted_work_type: /fulltime/i, location: /San Jose/i})
```

The screenshot displays the MongoDB Atlas interface with two queries. The top query is on the 'employee_counts' collection, and the bottom query is on the 'job_postings' collection. Both queries use the same MQL filter: { title: /analyst/i, formatted_work_type: /fulltime/i, location: /San Jose/i}.

Top Query: LinkedIn_job_postings_DB.employee_counts

Documents: 12.6k, Indexes: 1

Pipeline: \$match

Query: db.job_postings.find({ title: /analyst/i, formatted_work_type: /fulltime/i, location: /San Jose/i})

Preview of documents:

```
{ "_id": ObjectId("655837ad1b54a1c70b09bafd"), "company_id": 81149246, "employee_count": 6, "follower_count": 91, "time_recorded": 1692644644.2779734, "time_recorded_ts": "2023-08-21T19:04:04.277+00:00", "date_recorded": "2023-08-21" }
```

Bottom Query: LinkedIn_job_postings_DB.job_postings

Documents: 15.5k, Indexes: 1

Pipeline: \$match

Query: db.job_postings.find({ title: /analyst/i, formatted_work_type: /fulltime/i, location: /San Jose/i})

PIPELINE OUTPUT:

```
{ "_id": ObjectId("655838311b54a1c70b091242"), "job_id": 3693068226, "company_id": 5110, "title": "Credit Control Analyst Temporary ", "description": "Credit Control Analyst TemporaryAssignment duration is 91120232162023 ...", "max_salary": 35, "med_salary": NaN, "min_salary": 35, "pay_period": "HOURLY", "formatted_work_type": "Fulltime", "location": "San Jose CA", "applies": 21, "original_listed_time": 1690000000000, "remote_allowed": NaN }
```

Query Performance:

Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visual Tree Raw Output

COLSCAN

Returned 1 Execution Time 198 ms

Documents Examined: 15520

Query Performance Summary

1 documents returned

15520 documents examined

201 ms execution time

Is not sorted in memory

0 index keys examined

No index available for this query.

+ -

Close

2. Companies with the highest increase in follower count (top 5)

Query:

db.employee_counts.aggregate([{ \$sort: { time_recorded: -1 } }, { \$group: { _id: "\$company_id", follower_count: { \$first: "\$follower_count" } } }, { \$sort: { follower_count: -1 } }, { \$limit: 5 }])

linkedinjobposti... Aggregations LinkedIn_job_pos...

My Queries Databases Search LinkedIn_job_postings_DB companies employee_counts job_postings admin local test

LinkedIn_job_postings_DB.employee_counts 12.6k 1 DOCUMENTS INDEXES

Documents Aggregations Schema Indexes Validation

Pipeline \$sort \$group \$sort \$limit Explain Export Run More Options

db.employee_counts.aggregate([{ \$sort: { time_recorded: -1 } }, { \$group: { _id: "\$company_id", follower_count: { \$first: "\$follower_count" } } }, { \$sort: { follower_count: -1 } }, { \$limit: 5 }])

Generate

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE PREVIEW STAGES TEXT

12551 Documents in the collection

Preview of documents

_id: ObjectId('655837ad1b54a1c70b09baf')

company_id: 81149246

employee_count: 6

follower_count: 91

time_recorded: 1692644644.2779734

time_recorded_ts: 2023-08-21T19:04:04.277+00:00

date_recorded: "2023-08-21"

_id: ObjectId('655837ad1b54a1c70b09baf')

company_id: 10033339

employee_count: 3

follower_count: 187

time_recorded: 1692644644.2779734

time_recorded_ts: 2023-08-21T19:04:04.277+00:00

date_recorded: "2023-08-21"

_id: ObjectId('655837ad1b54a1c70b09baff')

company_id: 6049228

employee_count: 20

follower_count: 82

time_recorded: 1692644645.1013184

time_recorded_ts: 2023-08-21T19:04:05.101+00:00

date_recorded: "2023-08-21"

Stage 1 \$sort

Output after \$sort stage (Sample of 10 documents)

Query Performance:

✕

Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visual Tree

Raw Output

> COLLSCAN

Returned 12551 Execution Time 21 ms

Documents Examined: 12551

Query Performance Summary

4 documents returned

12551 documents examined

53449 ms execution time

Is not sorted in memory

0 index keys examined

No index available for this query.

+
-

Close

4. Highest salary ranges for different roles in (one particular industry.)

Query:
db.job_postings.aggregate([{\$group: {_id: "\$title",salaryVariability: {\$stdDevSamp: { \$subtract: ["\$max_salary", "\$min_salary"]}}}},{\$sort: { salaryVariability: -1 } },{\$limit: 5 }]);

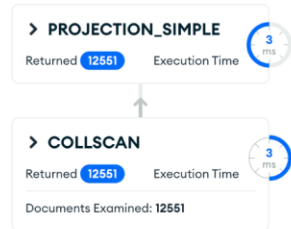
Query Performance:

Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visual Tree

Raw Output



Query Performance Summary

- 1 documents returned
- 12551 documents examined
- 22 ms execution time
- Is not sorted in memory
- 0 index keys examined
- No index available for this query.



Close

5. Count of sponsored "Data Analyst" job postings for each location.

Query:

```
db.job_postings.aggregate([ { $match: { sponsored: 1, title: "Data Analyst" } }, { $group: { _id: "$location", count: { $sum: 1 } } } ] );
```

My Queries

Databases

Search

LinkedIn_job_postings_DB

companies

employee_counts

job_postings

admin

local

test

LinkedIn_job_postings_DB.job_postings

15.5k 1
DOCUMENTS INDEXES

Documents Aggregations Schema Indexes Validation

Pipeline \$match \$group Explain Export Run More Options

```
db.job_postings.aggregate([ { $match: { sponsored: 1, title: "Data Analyst" } }, { $group: { _id: "$location", count: { $sum: 1 } } } ]
```

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE

PREVIEW STAGES TEXT

15520 Documents in the collection

Preview of documents

```
{
  "_id": ObjectId("65582ef31b54a1c70b0904a0"),
  "job_id": 133114754,
  "company_id": 77766802,
  "title": "Sales Manager",
  "description": "Are you a dynamic and creative marketing professional looking to make ...",
  "max_salary": NaN
}
```

```
{
  "_id": ObjectId("65582ef41b54a1c70b0904a1"),
  "job_id": 133196985,
  "company_id": 1089558,
  "title": "Model Risk Auditor",
  "description": "Join Us as a Model Risk Auditor Showcase Your Financial Services Audi...",
  "max_salary": NaN
}
```

```
{
  "_id": ObjectId("65582ef41b54a1c70b0904a2"),
  "job_id": 381055942,
  "company_id": 96654689,
  "title": "Business Manager",
  "description": "Business ManagerFirst Baptis Church ForneyForney Texas Rep to Exe...",
  "max_salary": NaN
}
```

Stage 1 \$match PREVIEW

LinkedIn_job_postings_DB.job_postings

15.5k
DOCUMENTS IND

Documents Aggregations Schema Indexes Validation

Pipeline \$match \$group Explain Export Run More Options

```
db.job_postings.aggregate([ { $match: { sponsored: 1, title: "Data Analyst" } }, { $group: { _id: "$location", count: { $sum: 1 } } } ]
```

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE

PREVIEW STAGES TEXT

```
1 [
2   {
3     $match: {
4       sponsored: 1,
5       title: "Data Analyst",
6     },
7   },
8   {
9     $group: {
10      _id: "$location",
11      count: {
12        $sum: 1,
13      },
14    },
15  },
16 ]
```

PIPELINE OUTPUT

Sample of 1 document

```
{
  "_id": "Oklahoma City OK",
  "count": 1
}
```

OUTPUT OPTIONS

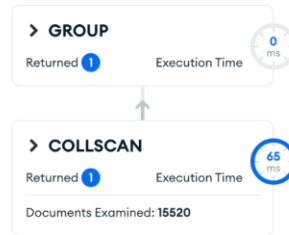
Query Performance:

Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visual Tree

Raw Output



Query Performance Summary

- 1 documents returned
- 15520 documents examined
- 65 ms execution time
- Is not sorted in memory
- 0 index keys examined
- No index available for this query.

6. Find the companies that don't offer 401(k) as a benefit.

Query:

```
db.job_postings.aggregate([{$match: {benefits: {$ne: "401k"}}}, {$project: {company_id: 1, title: 1, benefits: 1}}])
```

My Queries

Databases

Search

Linkedin_job_postings_DB

companies

employee_counts

job_postings

admin

local

test

15.5k

1

DOCUMENTS

INDEXES

Documents

Aggregations

Schema

Indexes

Validation

Pipeline

\$match

\$project

+

db.job_postings.aggregate([{\$match: {benefits: {\$ne: "401k"}}}, {\$project: {company_id: 1, title: 1, benefits: 1}}])

×

Generate

👍

🔗

Untitled - modified

SAVE

CREATE NEW

EXPORT TO LANGUAGE

PREVIEW

{ } STAGES

TEXT

⚙️

1

[

2

{ "\$match": { "benefits": { "\$ne": "401k" } },

3

{ "\$project": { "company_id": 1, "title": 1, "benefits": 1 } }

4

]

PIPELINE OUTPUT

Sample of 10 documents

_id: ObjectId('65582ef41b54a1c70b0904a1')

company_id: 1089558

title: "Model Risk Auditor"

benefits: NaN

_id: ObjectId('65582ef41b54a1c70b0904a2')

company_id: 96654609

title: "Business Manager"

benefits: NaN

_id: ObjectId('65582ef41b54a1c70b0904a3')

company_id: 1244539

title: "NY Studio Assistant"

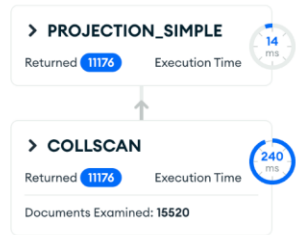
Query Performance:

Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visual Tree

Raw Output



Query Performance Summary

11176 documents returned

15520 documents examined

257 ms execution time

Is not sorted in memory

0 index keys examined

No index available for this query.



Close

7. Find jobs in California where the max salary is greater than 200K and min salary is greater than 125K

Query:

```
db.job_postings.aggregate([{$match: {location: {$regex: "CA"}, max_salary: {$lt: 200000}, min_salary: {$gt: 125000}}}, {$project: {_id: 0, job_id: 1, company_id: 1, max_salary: 1, min_salary: 1}}])
```

linkedinjobposti... Aggregations LinkedIn_job_pos...

My Queries

Databases

Search

LinkedIn_job_postings_DB

companies

employee_counts

job_postings

admin

local

test

LinkedIn_job_postings_DB.job_postings

15.5k 1

DOCUMENTS INDEXES

Documents Aggregations Schema Indexes Validation

Pipeline \$match \$project

db.job_postings.aggregate([{\$match: {location: {\$regex: "CA"}, max_salary: {\$lt: 200000}, min_salary: {\$gt: 125000}}}, {\$project: [_id: 0, job_id: 1, company_id: 1, max_salary: 1, min_salary: 1]}])

Generate

PREVIEW STAGES TEXT

Stage 1 \$match

```
1 {
2   location: {
3     $regex: "CA",
4   },
5   max_salary: {
6     $lt: 200000,
7   },
8   min_salary: {
9     $gt: 125000,
10  },
11 }
```

Output after \$match stage (Sample of 10 documents)

_id: ObjectId('65582f291b54a1c70b0906e4')

job_id: 3693044340

company_id: 95675928

title: "Salesforce Technical Architect
GCCitizens and Locals Only"

description: "Employment TypeFulltime
Job Description Technical
Architect will be th..."

max_salary: 180000

_id: ObjectId('65582f391b54a1c70b090794')

job_id: 3693045257

company_id: 9889773

title: "Construction Project Manager"

description: "Construction Project
ManagerLaFata Contract Servi
LCS an Anser Advl..."

max_salary: 160000

med_salary: N/A

Stage 2 \$project

LinkedIn_job_postings_DB.job_postings

15.5K 1

DOCUMENTS INDEXES

Documents Aggregations Schema Indexes Validation

Pipeline \$match \$project

db.job_postings.aggregate([{\$match: {location: {\$regex: "CA"}, max_salary: {\$lt: 200000}, min_salary: {\$gt: 125000}}}, {\$project: [_id: 0, job_id: 1, company_id: 1, max_salary: 1, min_salary: 1]}])

Generate

PREVIEW STAGES TEXT

Stage 1 \$match

```
1 {
2   location: {
3     $regex: "CA",
4   },
5   max_salary: {
6     $lt: 200000,
7   },
8   min_salary: {
9     $gt: 125000,
10  },
11 }
```

PIPELINE OUTPUT

Sample of 10 documents

job_id: 3693044340

company_id: 95675928

max_salary: 180000

min_salary: 160000

job_id: 3693045257

company_id: 9889773

max_salary: 160000

min_salary: 140000

job_id: 3693045703

company_id: 104243

max_salary: 155000

Query Performance:
Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visual Tree Raw Output



Query Performance Summary

- 80 documents returned
- 15520 documents examined
- 191 ms execution time
- Is not sorted in memory
- 0 index keys examined
- No index available for this query.

+
-

Close

8. Companies with most common job skills.

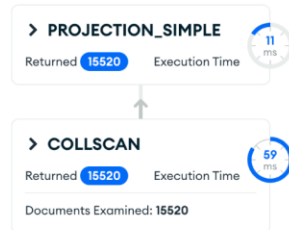
Query:
db.job_postings.aggregate([{ \$unwind: "\$skill_abr" }, { \$group: { _id: "\$skill_abr", count: { \$sum: 1 } } }, { \$sort: { count: -1 } }, { \$limit: 10 }]);

Query Performance:

Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visual Tree Raw Output



Query Performance Summary

- 10 documents returned
- 15520 documents examined
- 107 ms execution time
- Is not sorted in memory
- 0 index keys examined
- No index available for this query.

9. List all companies in a specific city (ex: New York) and their average employee counts.

Query:

```
db.getCollection('companies').aggregate([{$lookup: {from: 'employee_counts', localField: 'company_id', foreignField: 'company_id', as: 'employee_counts'}}, {$unwind: {path: '$employee_counts', includeArrayIndex: 'string', preserveNullAndEmptyArrays: true}}, {$group: {_id: {company_name: '$company_name', city: '$city'}, avg_employee_count: {$avg: '$employee_counts.employee_count'}}}, {$match: {'_id.city': 'New York'}}, {$project: {_id: 0, company_name: '$_id.company_name', city: '$_id.city', avg_employee_count: 1}}]);
```

The screenshot shows the MongoDB Compass interface for the `LinkedIn_job_postings_DB.companies` collection. The aggregation pipeline is defined in Stage 5:

```
1 /**
2  * specifications: The fields to
3  * include or exclude.
4  */
5 {
6   _id: 0,
7   company_name: "$_id.company_name",
8   city: "$_id.city",
9   avg_employee_count: 1
10 }
```

The output after the `$project` stage (Sample of 10 documents) is shown:

```
{
  avg_employee_count: 26898,
  company_name: "Compass",
  city: "New York"
}
```

```
{
  avg_employee_count: 1322,
  company_name: "Office of the New York State Attorney General",
  city: "New York"
}
```

Buttons for `Generate aggregation`, `Explain`, `Export`, `Run`, and `More Options` are visible. A `+ Add Stage` button is at the bottom.

My Queries
Databases
Search
LinkedIn_job_postings_DB
companies
employee_counts
job_postings
admin
local

LinkedIn_job_postings_DB.companies

Documents: 0 Aggregations: 1 Schemas: 1 Indexes: 1 Validation: 1

Pipeline: **Lookup** Sunwind Sgroup Smatch Sproject

Generate aggregation Explain Export Run More Options

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE

```

1 [
2   {
3     $lookup:
4       /**
5        * from: The target collection.
6        * localField: The local join field.
7        * foreignField: The target join field.
8        * as: The name for the results.
9        * pipeline: Optional pipeline to run on the foreign collection.
10       * let: Optional variables to use in the pipeline field stages.
11       */
12     {
13       from: "employee_counts",
14       localField: "company_id",
15       foreignField: "company_id",
16       as: "employee_counts",
17     },
18   },
19   {
20     $sunwind:
21       /**
22        * path: Path to the array field.
23        * includeArrayIndex: Optional name for index.
24        * preserveNullAndEmptyArrays: Optional
25        */

```

PIPELINE OUTPUT
Sample of 10 documents

avg_employee_count: 159
company_name: "Zentalis Pharmaceuticals"
city: "New York"

avg_employee_count: 0
company_name: "Giglid"
city: "New York"

avg_employee_count: 279
company_name: "CARTO"
city: "New York"

avg_employee_count: 124
company_name: "Coda SearchStaffing"
city: "New York"

Query Performance:

Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visual Tree

Raw Output

PROJECTION_DEFAULT

Returned: 275 Execution Time: 0 ms

COLLSCAN

Returned: 275 Execution Time: 36 ms

Documents Examined: 6063

Query Performance Summary

275 documents returned

6063 documents examined

2266 ms execution time

Is not sorted in memory

0 index keys examined

No index available for this query.

+
-

Close

10. Identifying companies with the highest average maximum salaries for job postings.

Query:

```

db.job_postings.aggregate([{$lookup: {from: "companies",localField: "company_id",
foreignField: "company_id",as: "company_info"}},{$group: {_id:
"$company_id",avgMaxSalary: {$avg: {$toDouble: {$ifNull: ["$max_salary",
0]}},company_info: {$push: "$company_info"}},{$sort: {avgMaxSalary: -1}}]);

```


My Queries

Databases

Search

LinkedIn_job_postings_DB

companies

employee_counts

job_postings

admin

local

test

LinkedIn_job_postings_DB.job_postings

15.5k
DOCUMENTS

1
INDEXES

Documents Aggregations Schema Indexes Validation

Pipeline \$lookup \$group \$sort

Explain Export Run More Options

db.job_postings.aggregate([{\$lookup: {from: "companies", localField: "company_id", foreignField: "company_id", as: "company_info"}}, {\$group: {

Generate

Untitled - modified SAVE CREATE NEW EXPORT TO LANGUAGE

PREVIEW STAGES TEXT

15520 Documents in the collection

Preview of documents

```
{
  "_id": ObjectId("65582ef31b54a1c70b0904a0"),
  "job_id": 133114754,
  "company_id": 77766802,
  "title": "Sales Manager",
  "description": "Are you a dynamic and creative marketing professional looking to make ...",
  "max_salary": NaN
}
```

```
{
  "_id": ObjectId("65582ef41b54a1c70b0904a1"),
  "job_id": 133196985,
  "company_id": 1089558,
  "title": "Model Risk Auditor",
  "description": "Join Us as a Model Risk Auditor Showcase Your Financial Services Audi...",
  "max_salary": NaN
}
```

```
{
  "_id": ObjectId("65582ef41b54a1c70b0904a2"),
  "job_id": 381055942,
  "company_id": 96654609,
  "title": "Business Manager",
  "description": "Business ManagerFirst Baptis Church ForneyForney Texas Rej to Exe...",
  "max_salary": NaN
}
```

Stage1 \$lookup

Output after \$lookup stage (Sample of 10 documents)

Documents Aggregations Schema Indexes Validation

Pipeline \$lookup \$group \$sort Edit

Explain Export Run More Options

ALL RESULTS OUTPUT OPTIONS

Showing 1 - 20 count results VIEW

```
{
  "_id": 92699799,
  "avgMaxSalary": 766666.6666666666,
  "company_info": Array (3)
}
```

```
{
  "_id": 78124,
  "avgMaxSalary": 675000,
  "company_info": Array (9)
}
```

```
{
  "_id": 5058,
  "avgMaxSalary": 500000,
  "company_info": Array (1)
}
```

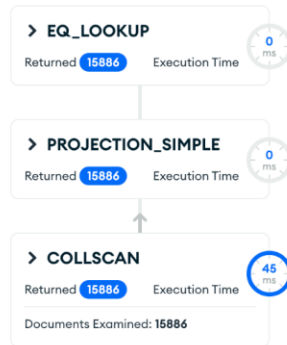
```
{
  "_id": 18627874,
  "avgMaxSalary": 500000,
  "company_info": Array (1)
}
```

```
{
  "_id": 28439179
}
```

Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visual Tree Raw Output



+
-

Query Performance Summary

- 6031 documents returned
- 15886 documents examined
- 62 ms execution time
- Is not sorted in memory
- 0 index keys examined
- No index available for this query.

Close

11. Which companies have the highest number of specialties by industry?

Query:

```
db.companies.aggregate([{$unwind:"$speciality"},{$group:{_id:{company_id:"$company_id",industry:"$industry",company_name:"$company_name"},specialtiesCount:{$sum:1}}},{$sort:{"_id.industry":1,specialtiesCount:-1}},{$group:{_id:"$_id.industry",topCompany:{$first:{company_id:"$_id.company_id",company_name:"$_id.company_name",specialtiesCount:"$specialtiesCount"}}}},{$project:{_id:1,topCompany:1}}]);
```

LinkedIn_job_postings_DB.companies

6.1k
DOCUMENTS

1
INDEXES

Documents **Aggregations** Schema Indexes Validation

Pipeline ▾ **\$unwind** \$group \$sort \$group \$project

Generate aggregation + ⓘ

Explain

Export

Run

More Options ▶

mypipe - modified

SAVE ▾

+ CREATE NEW

EXPORT TO LANGUAGE

PREVIEW

STAGES

TEXT

⚙️

Stage 5 \$project

```
1 ▾ /**
2   * specifications: The fields to
3   * include or exclude.
4   */
5 ▾ {
6     _id: 1,
7     topCompany: 1,
8     totalTopCompanies: 1
9 }
```

Output after \$project stage (Sample of 10 documents)

```
▸ _id: Array (1)
▸ topCompany: Object
  company_id: 2654590
  company_name: "AIXIAL GROUP"
  specialtiesCount: 20
```

```
▸ _id: Array (1)
▸ topCompany: Object
  company_id: 14255
  company_name: "CJ Logistics America"
  specialtiesCount: 46
```

+ Add Stage

[Learn more about aggregation pipeline stages](#)

LinkedIn_job_postings_DB.companies

6.1k
DOCUMENTS

1
INDEXES

Documents **Aggregations** Schema Indexes Validation

Pipeline ▾ **\$unwind** \$group \$sort \$group \$project

Generate aggregation + ⓘ

Explain

Export

Run

More Options ▶

mypipe - modified

SAVE ▾

+ CREATE NEW

EXPORT TO LANGUAGE

PREVIEW

STAGES

TEXT

⚙️

```
1 ▾ [
2   {
3     $unwind:
4     /**
5      * path: Path to the array field.
6      * includeArrayIndex: Optional name for index.
7      * preserveNullAndEmptyArrays: Optional
8      * toggle to unwind null and empty values.
9     */
10    {
11      path: "$speciality",
12    },
13  },
14  {
15    $group:
16    /**
17     * _id: The id of the group.
18     * fieldN: The first field name.
19    */
20    {
21      _id: {
22        company_id: "$company_id",
23        industry: "$industry",
24        company_name: "$company_name",
25      },
26      specialtiesCount: {
27        $sum: 1,
28      },
29    },
30  },
31  {
32    $sort:
33    /**
34     * Provide any number of field/order pairs
```

PIPELINE OUTPUT

Sample of 10 documents

OUTPUT OPTIONS ▾

```
▸ _id: Array (1)
▸ topCompany: Object
```

```
▸ _id: Array (1)
▸ topCompany: Object
```

```
▸ _id: Array (1)
▸ topCompany: Object
```

```
▸ _id: Array (1)
▸ topCompany: Object
```

```
▸ _id: Array (1)
▸ topCompany: Object
```

```
▸ _id: Array (1)
▸ topCompany: Object
```

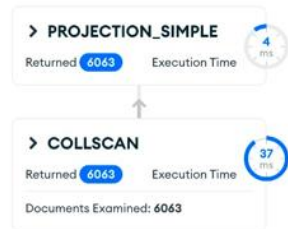
```
▸ _id: Array (1)
▸ topCompany: Object
```

Query Performance:

Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. [Learn more](#)

Visual Tree Raw Output



Query Performance Summary

- 142 documents returned
- 6063 documents examined
- 145 ms execution time
- Is not sorted in memory
- 0 index keys examined
- No index available for this query.

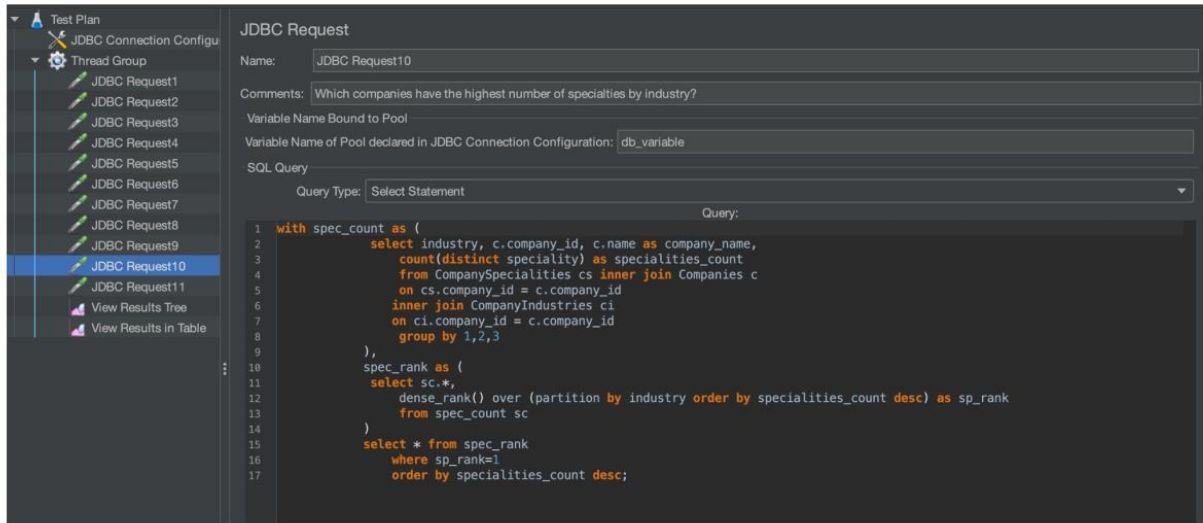


Close

MySQL Queries for comparing the performance measurements

The screenshot shows a 'Test Plan' interface with a 'JDBC Request' configuration. The 'Name' field is 'JDBC Request2'. The 'Comments' field contains 'List all companies in a specific city and their average employee counts.' The 'Variable Name Bound to Pool' field is empty. The 'Variable Name of Pool declared in JDBC Connection Configuration' field is 'db_variable'. The 'SQL Query' field contains the following SQL code:

```
1 WITH CompanyEmployeeCounts AS (  
2     SELECT c.name, c.city, ec.employee_count,  
3           AVG(ec.employee_count) OVER (PARTITION BY c.city) AS avg_employee_count  
4     FROM Companies c  
5     LEFT JOIN EmployeesCount ec ON c.company_id = ec.company_id  
6 )  
7 SELECT name, city, avg_employee_count  
8 FROM CompanyEmployeeCounts  
9 WHERE city = 'New York';
```



View Results in Table										
Name: View Results in Table										
Comments:										
Write results to file / Read from file										
Filename: <input type="text"/> <input type="button" value="Browse..."/> Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>										
Start Time	Thread Name	Label	Sample Time(ms)	Connect Time(ms)	Bytes	Sent Bytes	Latency	Sample #	Status	
00:53:00.335	Thread Group 1-1	JDBCTest Request1	1907	1800	10319	0	1906	1		
00:53:02.243	Thread Group 1-1	JDBCTest Request2	180	0	23602	0	178	2		
00:53:02.423	Thread Group 1-1	JDBCTest Request3	215	0	591	0	215	3		
00:53:02.639	Thread Group 1-1	JDBCTest Request4	287	0	119518	0	283	4		
00:53:02.927	Thread Group 1-1	JDBCTest Request5	138	0	115	0	137	5		
00:53:03.065	Thread Group 1-1	JDBCTest Request6	38676	1	5746	0	38673	6		
00:53:41.737	Thread Group 1-1	JDBCTest Request7	498	0	200045	0	477	7		
00:53:42.235	Thread Group 1-1	JDBCTest Request8	103	0	65	0	102	8		
00:53:42.338	Thread Group 1-1	JDBCTest Request9	111	0	15	0	111	9		
00:53:42.449	Thread Group 1-1	JDBCTest Request10	1047	0	7156	0	1045	10		
00:53:43.496	Thread Group 1-1	JDBCTest Request11	117	0	26954	0	116	11		

We have compared the performance of the JDBCTest Request 2 and JDBCTest Request 10 from MySQL with MongoDB. We found that the performance is better in MongoDB. Below is the table with the execution of all the queries

	Functional Requirements	Execution Time (milliseconds)
1	List of jobs that have 'analyst' in the title are full-time and based in San Jose, CA	198
2	Companies with the highest increase in follower count (top 5)	40

3	List of jobs by companies that have 'big data' or 'data analytics' as a specialty in NY state ordered by descending median salary	21
4	Highest salary ranges for different roles in (one particular industry)	6
5	Count of sponsored "Data Analyst" job postings for each location	65
6	Find the companies that don't offer 401(k) as a benefit.	254
7	Find jobs in California where the max salary is greater than 200K and min salary is greater than 125K	187
8	Companies with most common job skills	70
9	List all companies in a specific city (ex: New York) and their average employee counts	36 (MySQL Query Performance: 180)
10	Identifying companies with the highest average maximum salaries for job postings.	45
11	Which companies have the highest number of specialties by industry?	41 (MySQL Query Performance: 1047)

Group Members

Anshul Yadav

Sakshi Jain

Shreya Sree Matta

Veena Ramesh Beknal