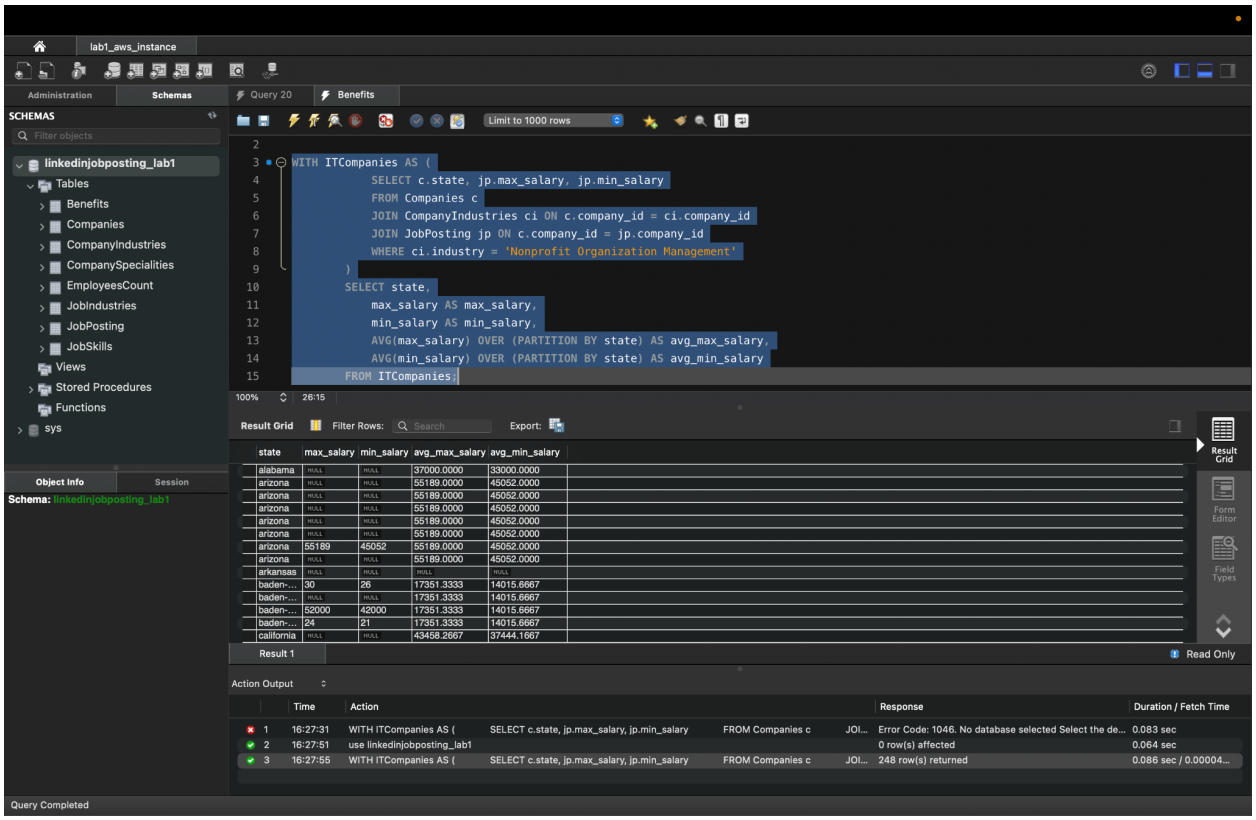
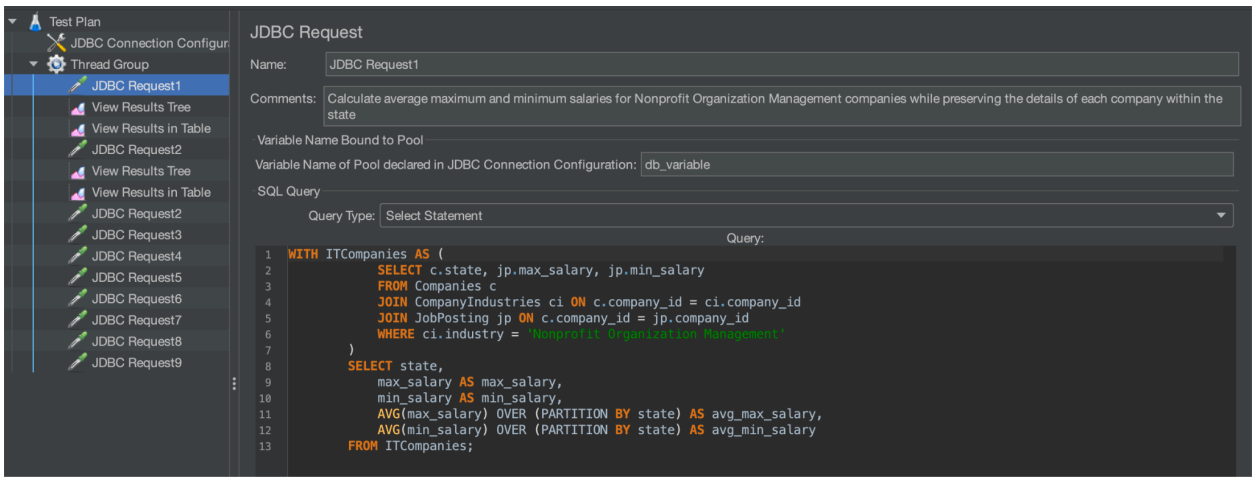
Purpose- SQL Query Execution in AWS

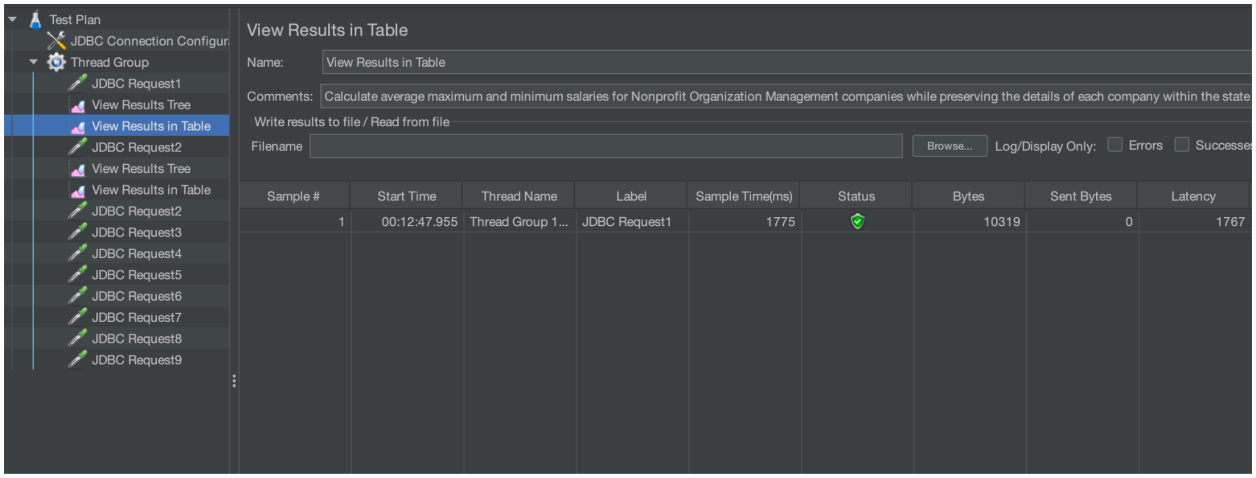
Query 1:

MYSQL:



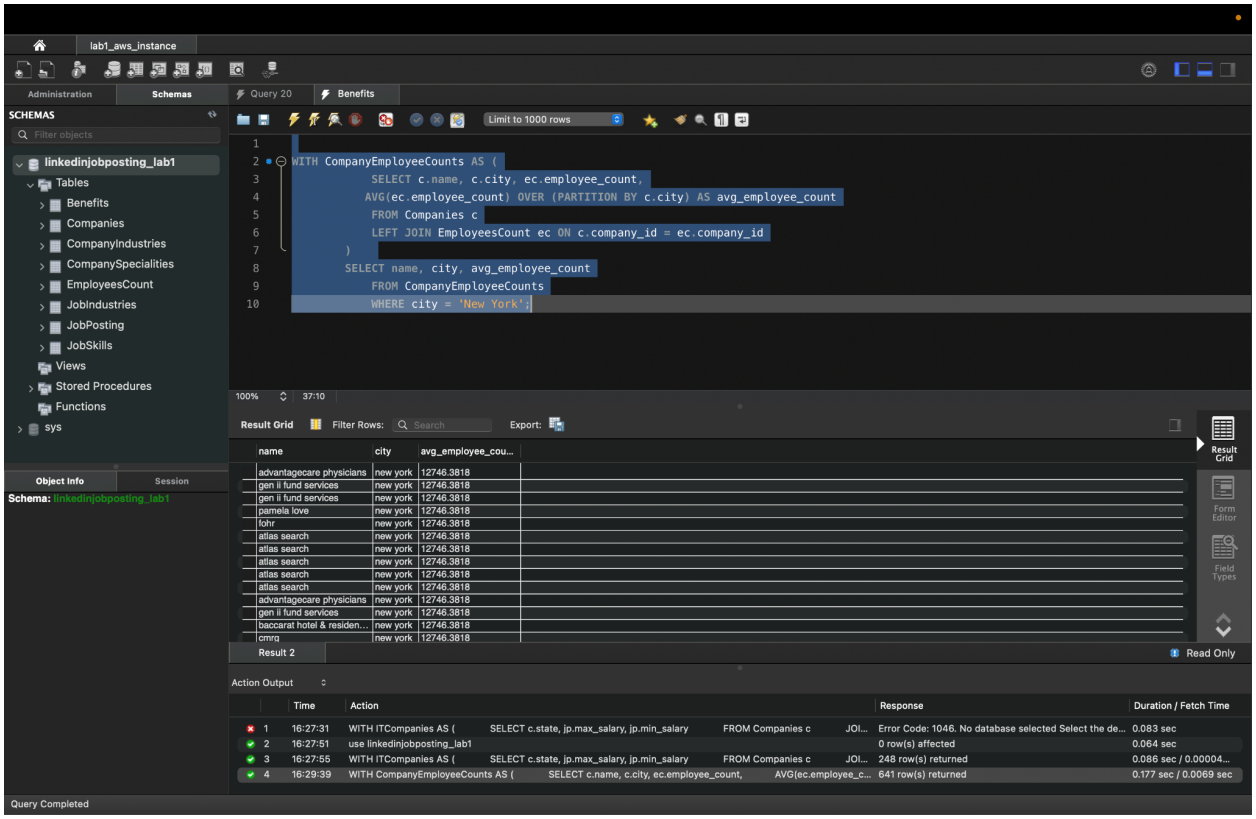
JMETER:



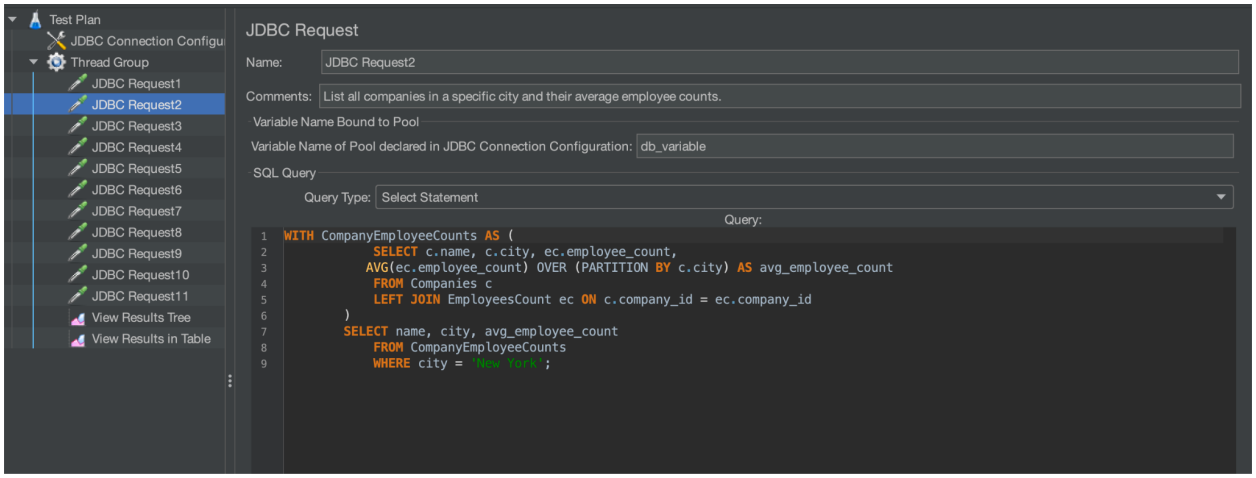


Query 2:

MYSQL:

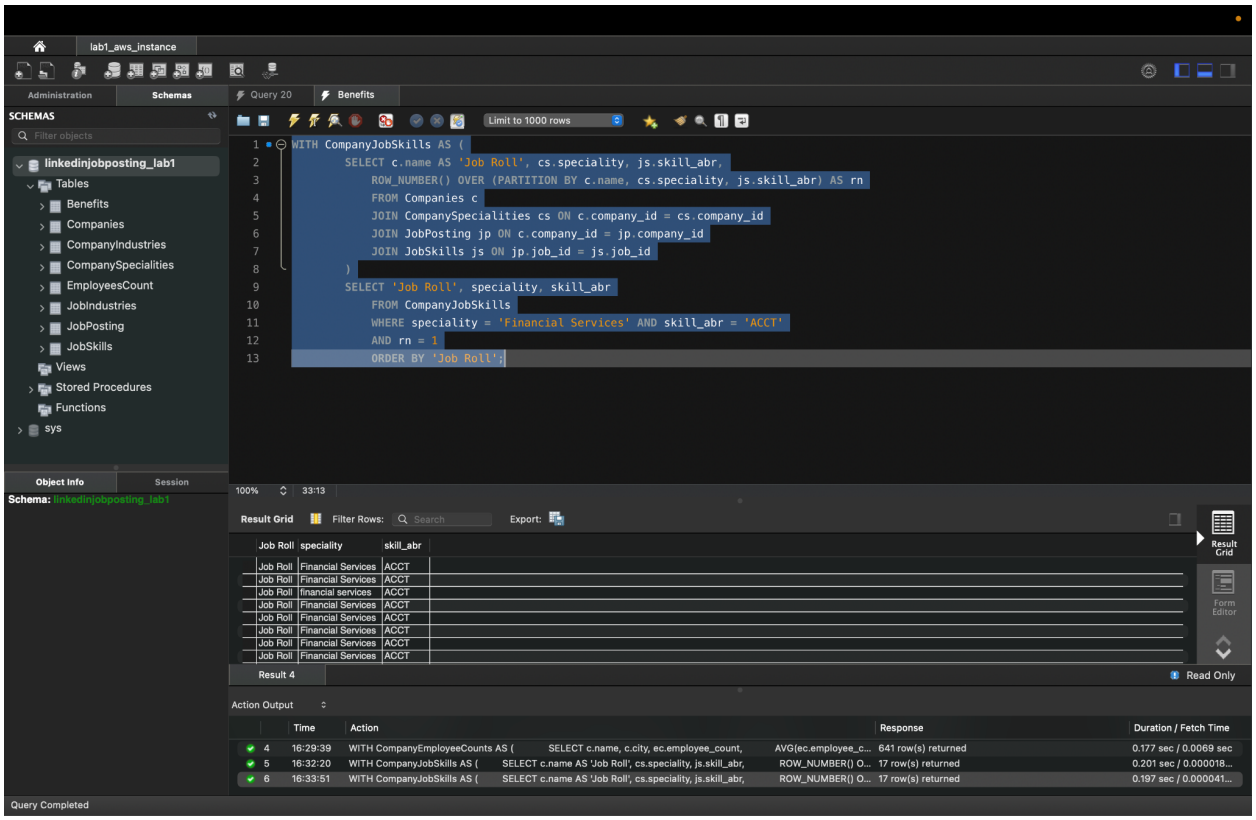


JMETER:

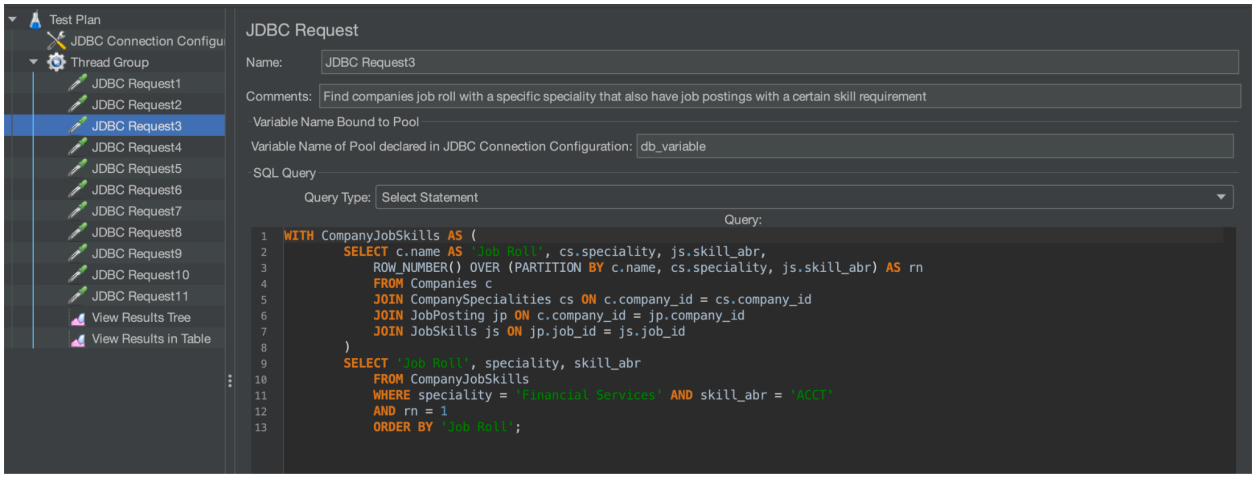


Query 3:

MYSQL:

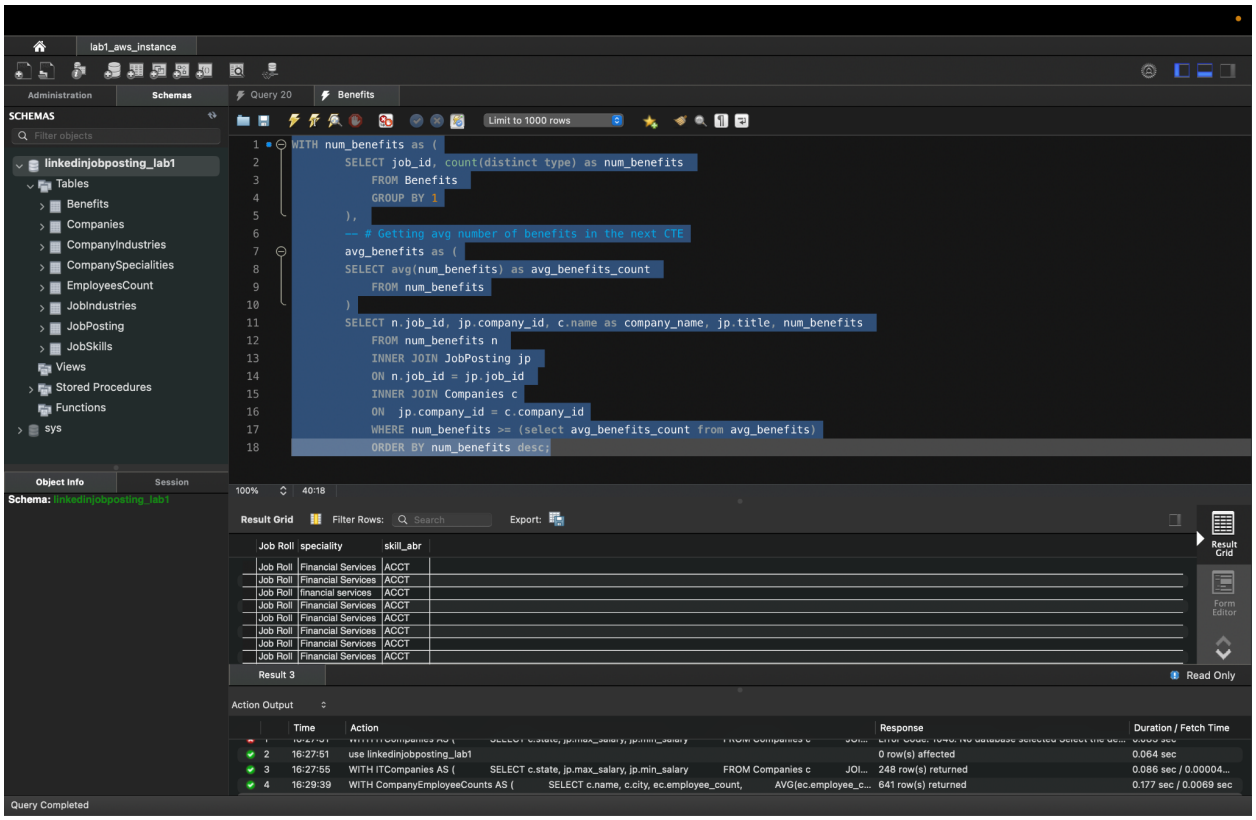


JMETER:

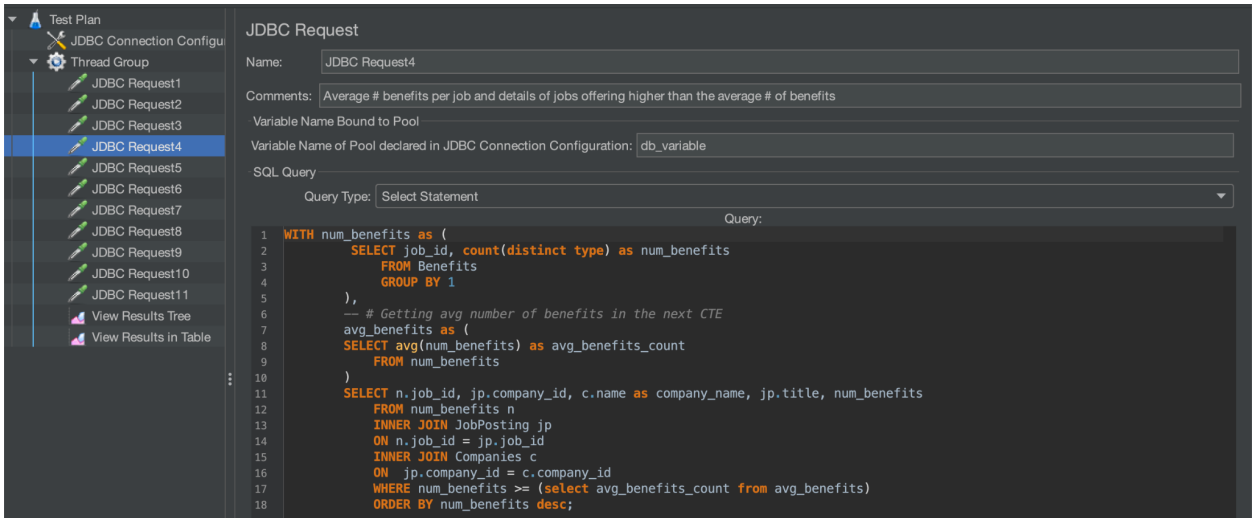


Query4:

MYSQL:

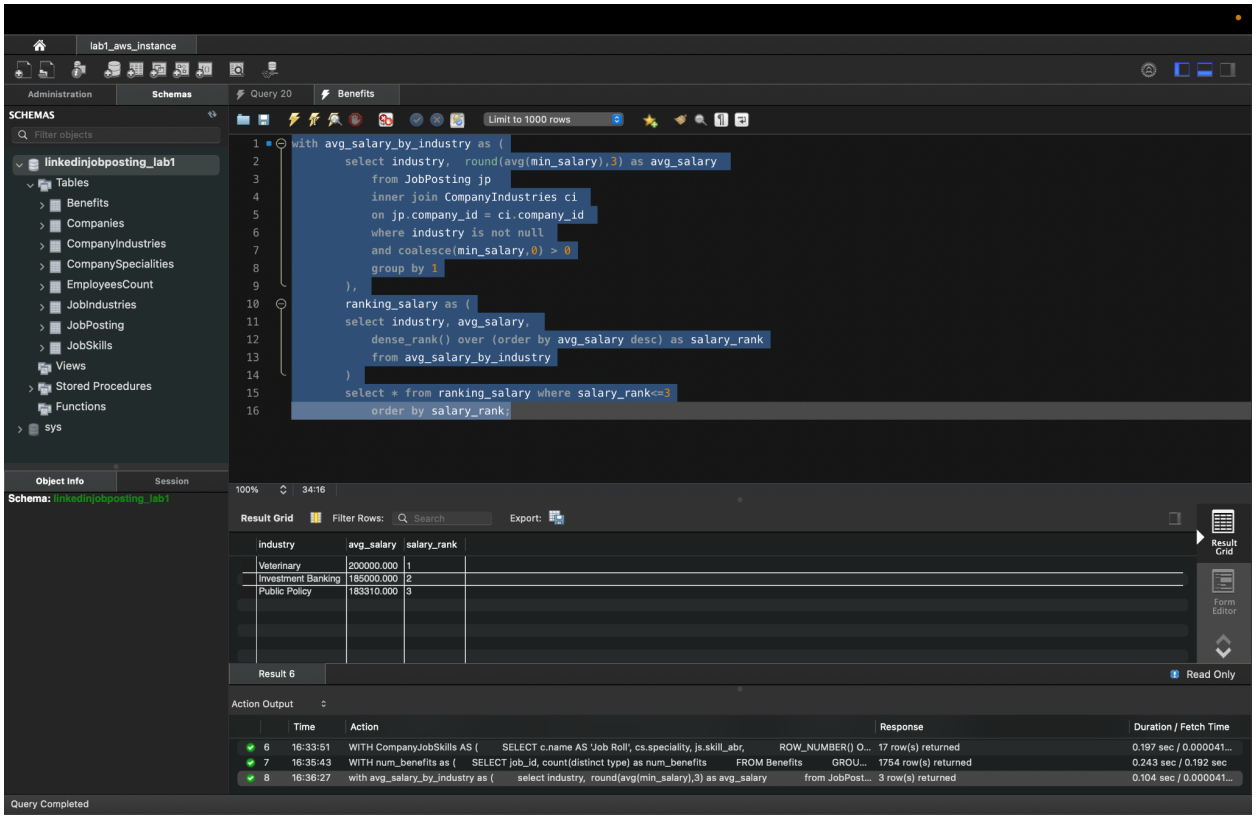


JMETER:

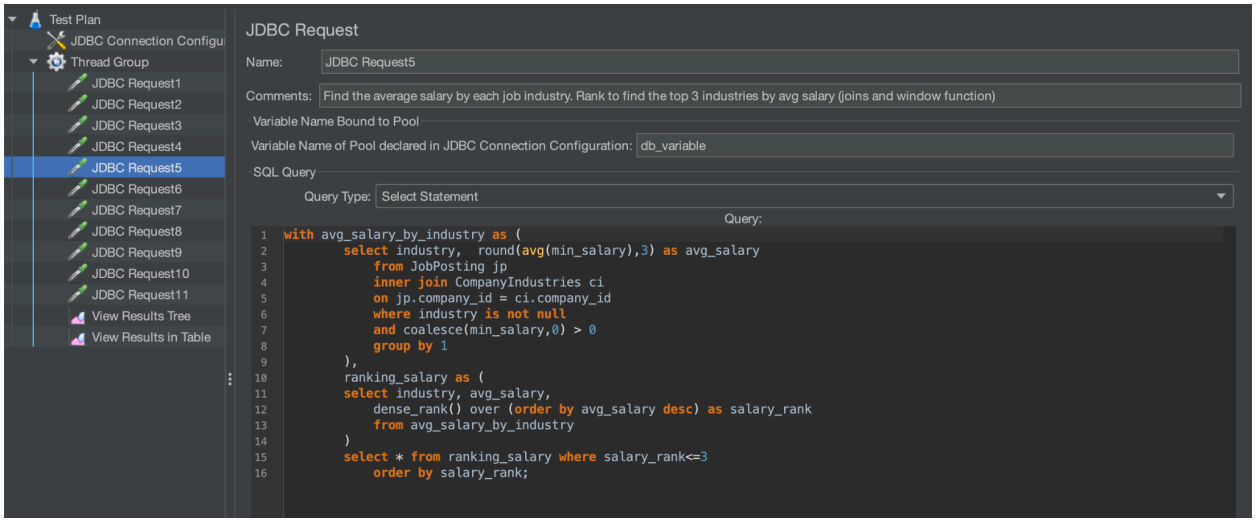


Query 5:

MYSQL:

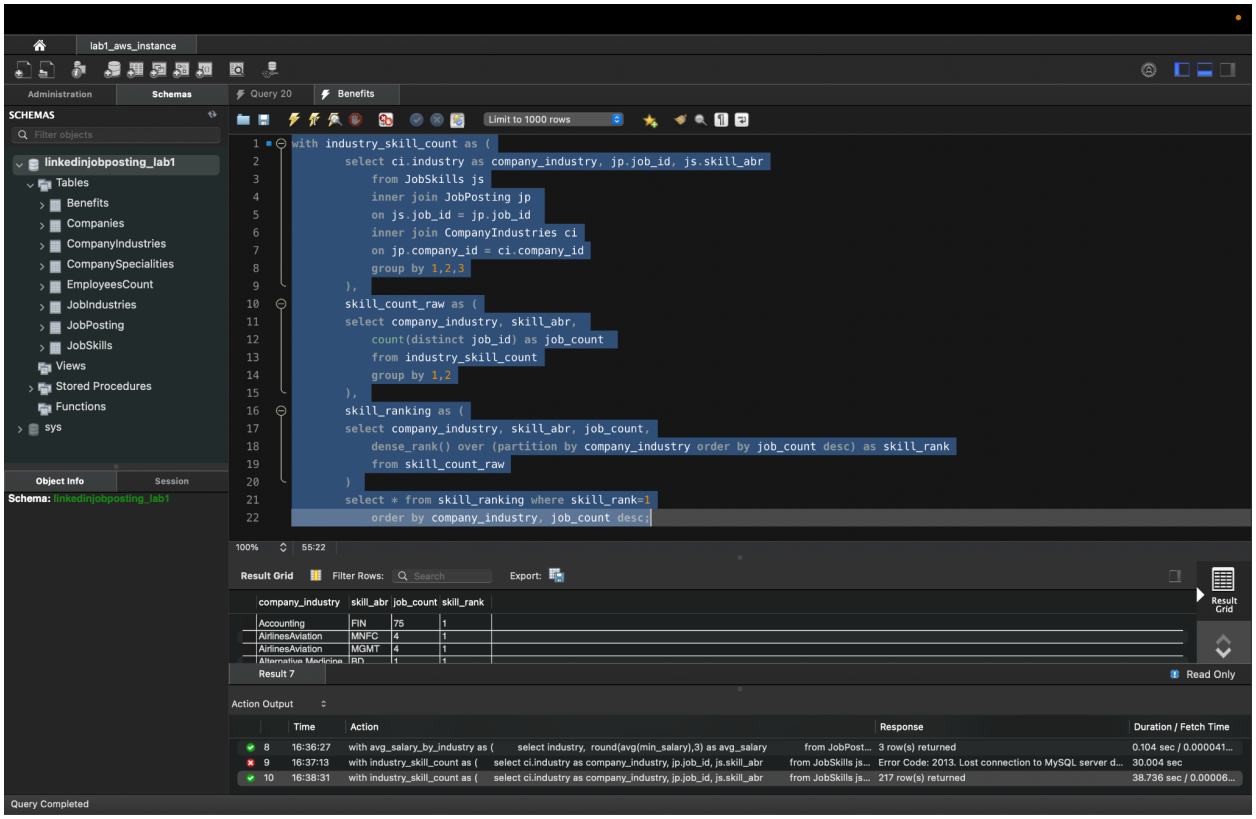


JMETER:

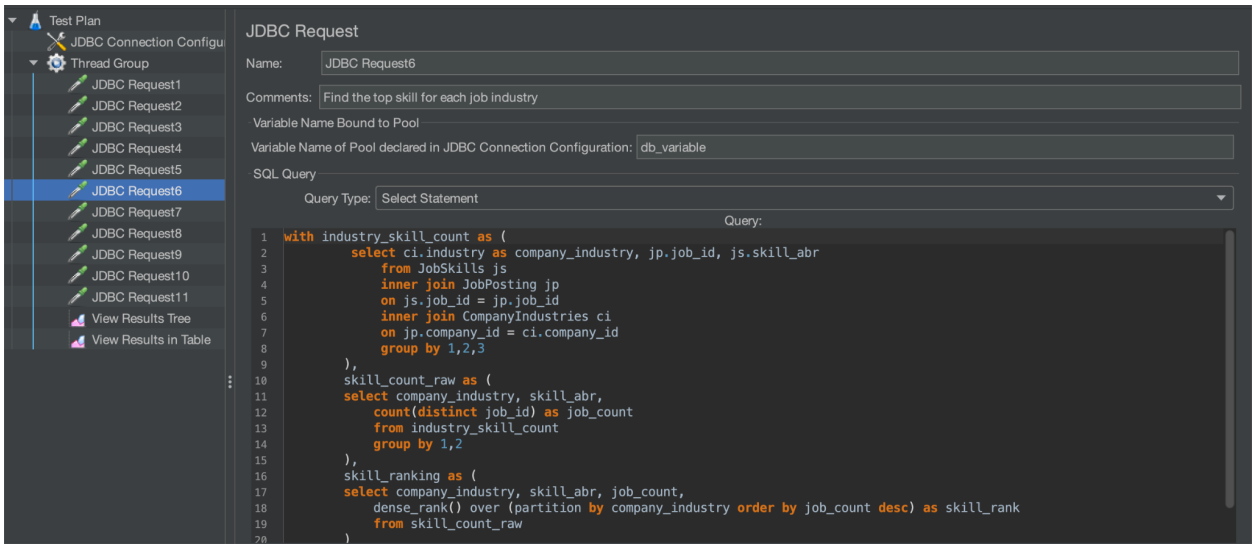


Query 6:

MYSQL:

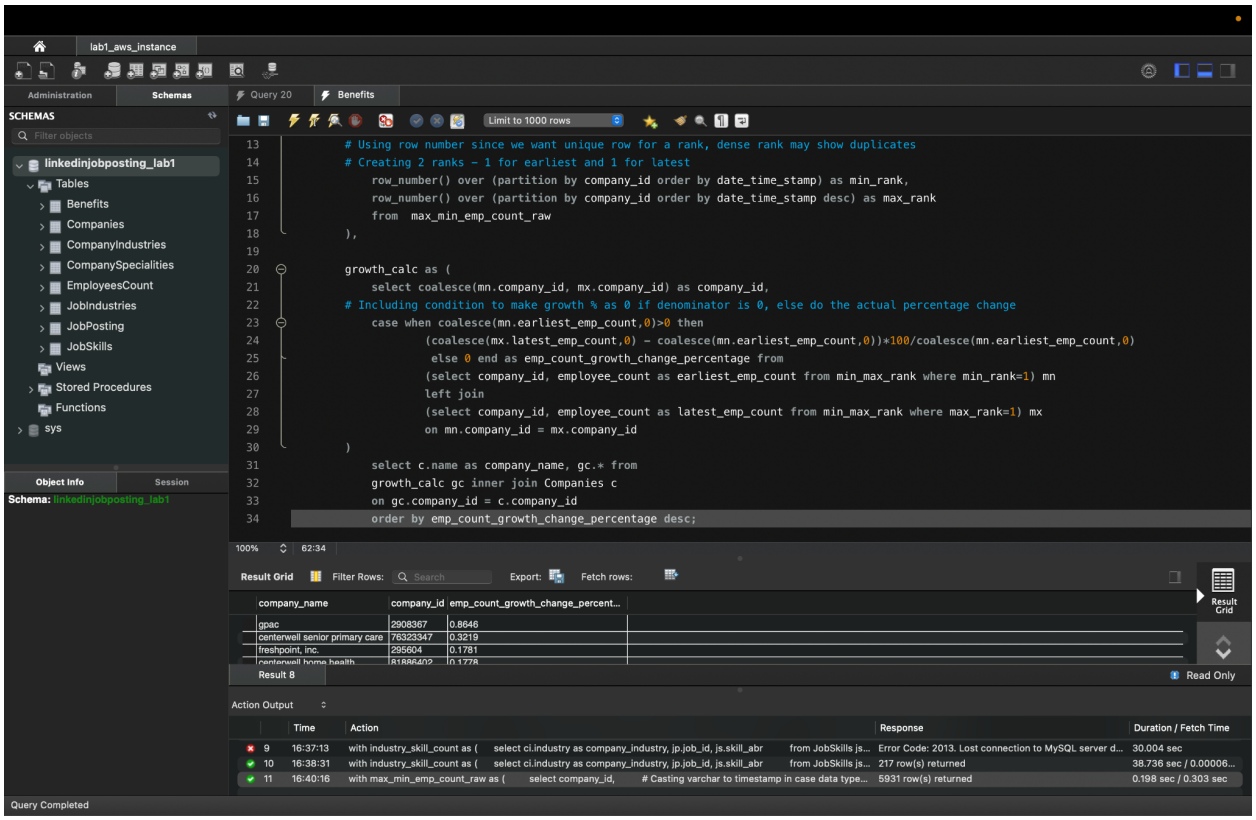


JMETER:

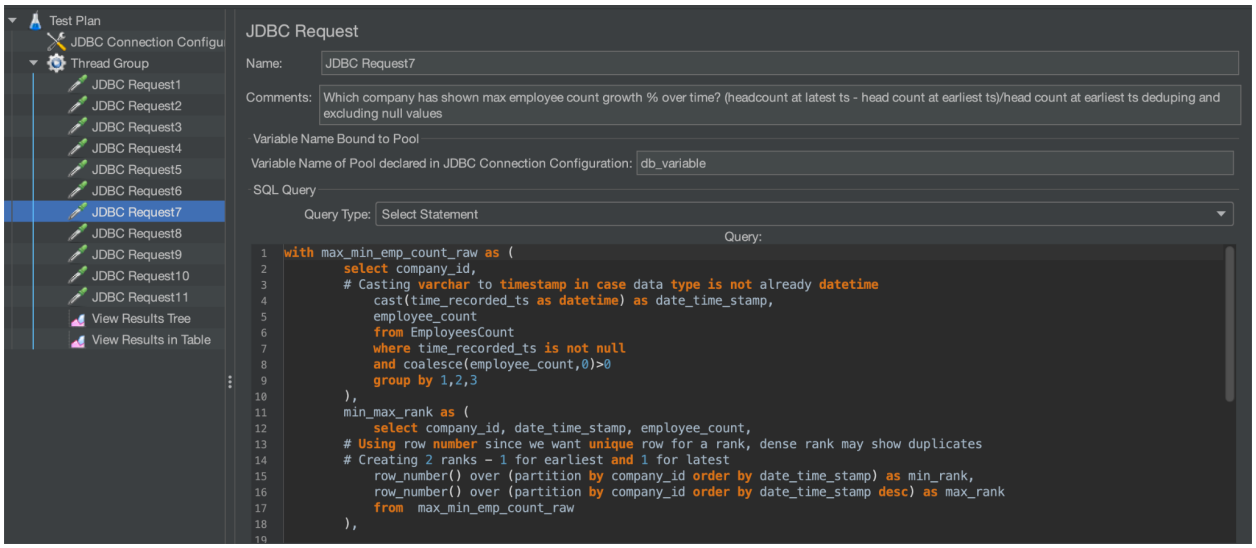


Query 7:

MYSQL:

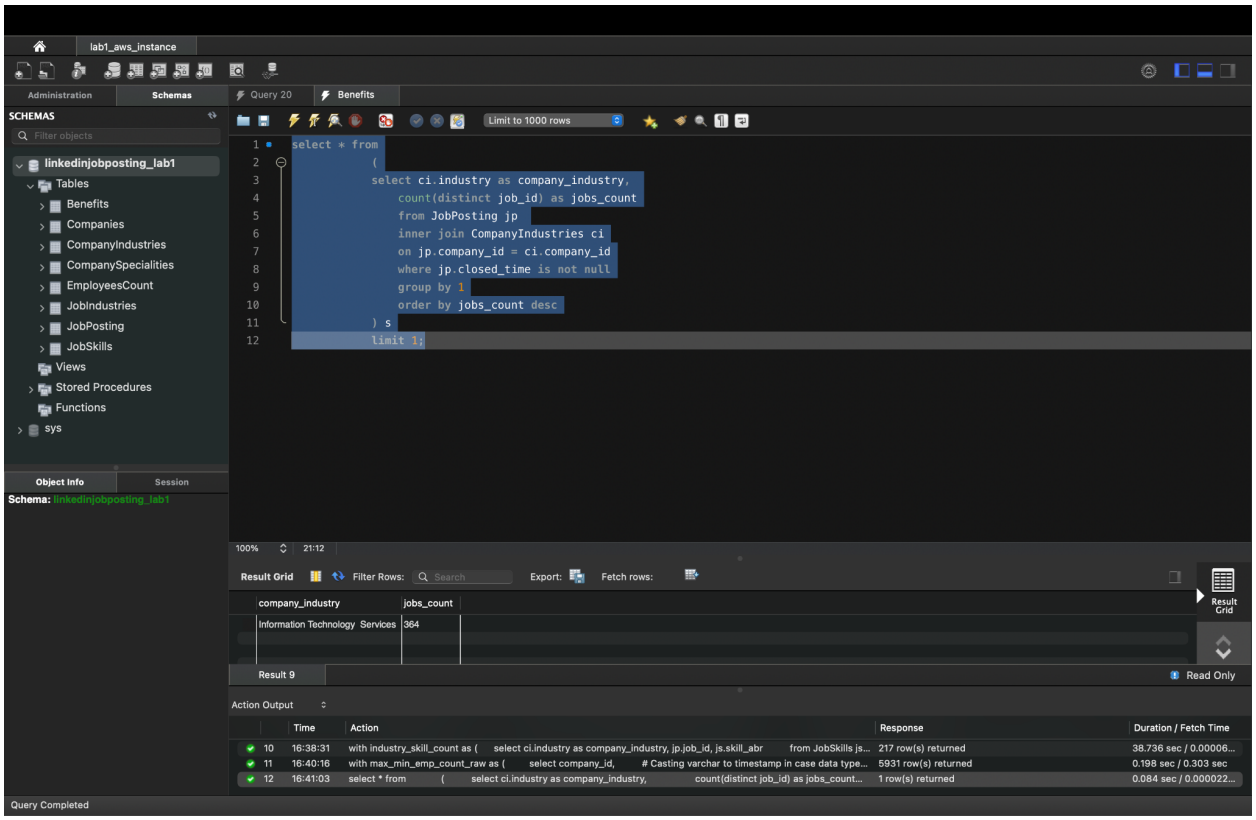


JMETER:

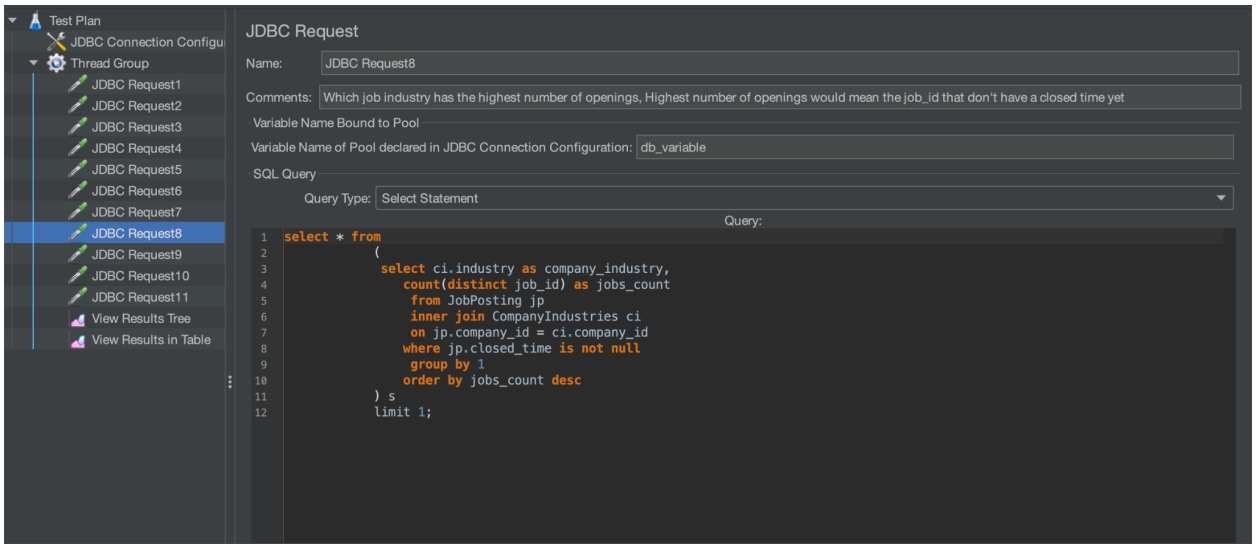


Query 8:

MYSQL:

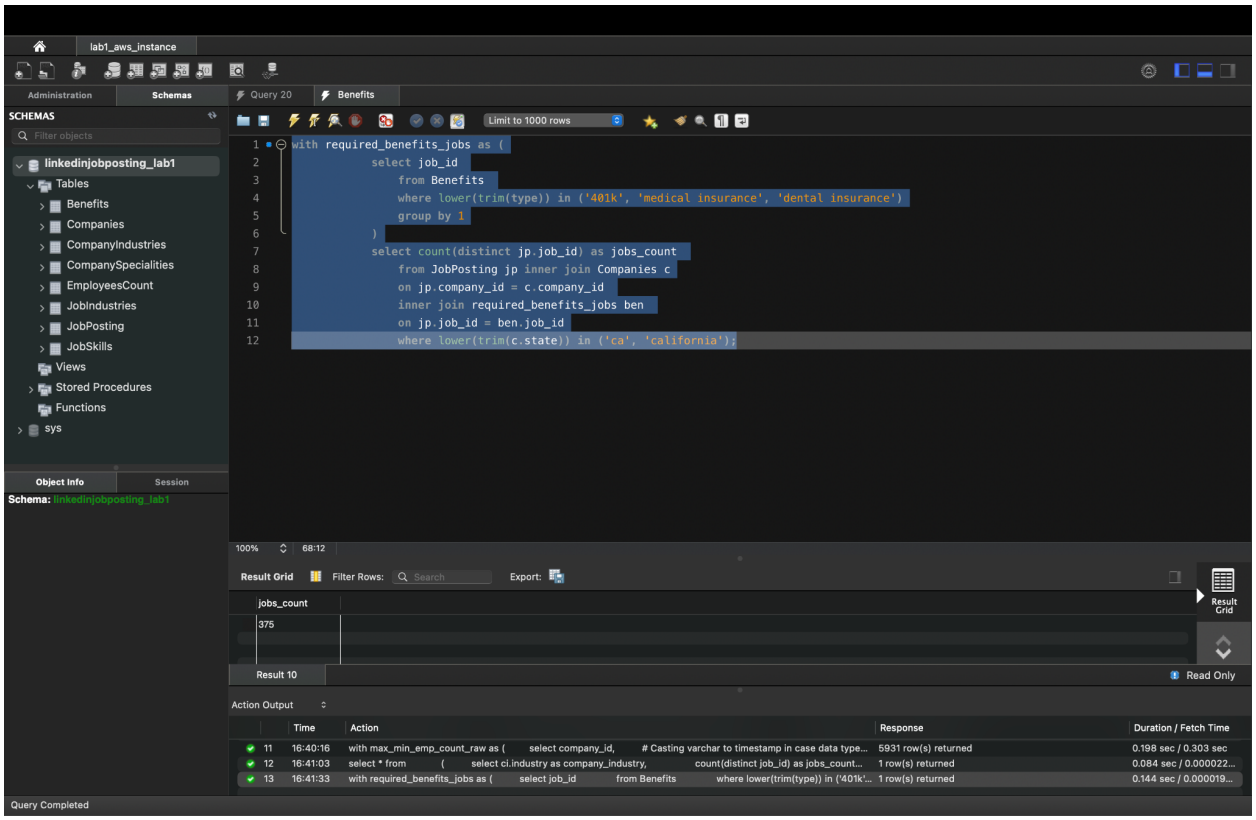


JMETER:

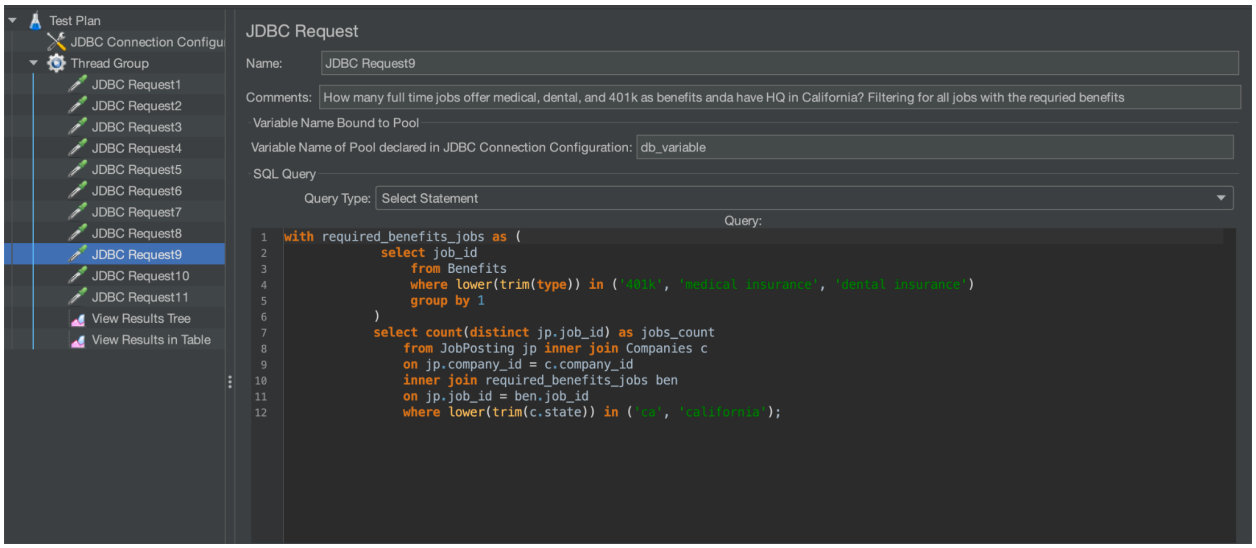


Query 9:

MYSQL:

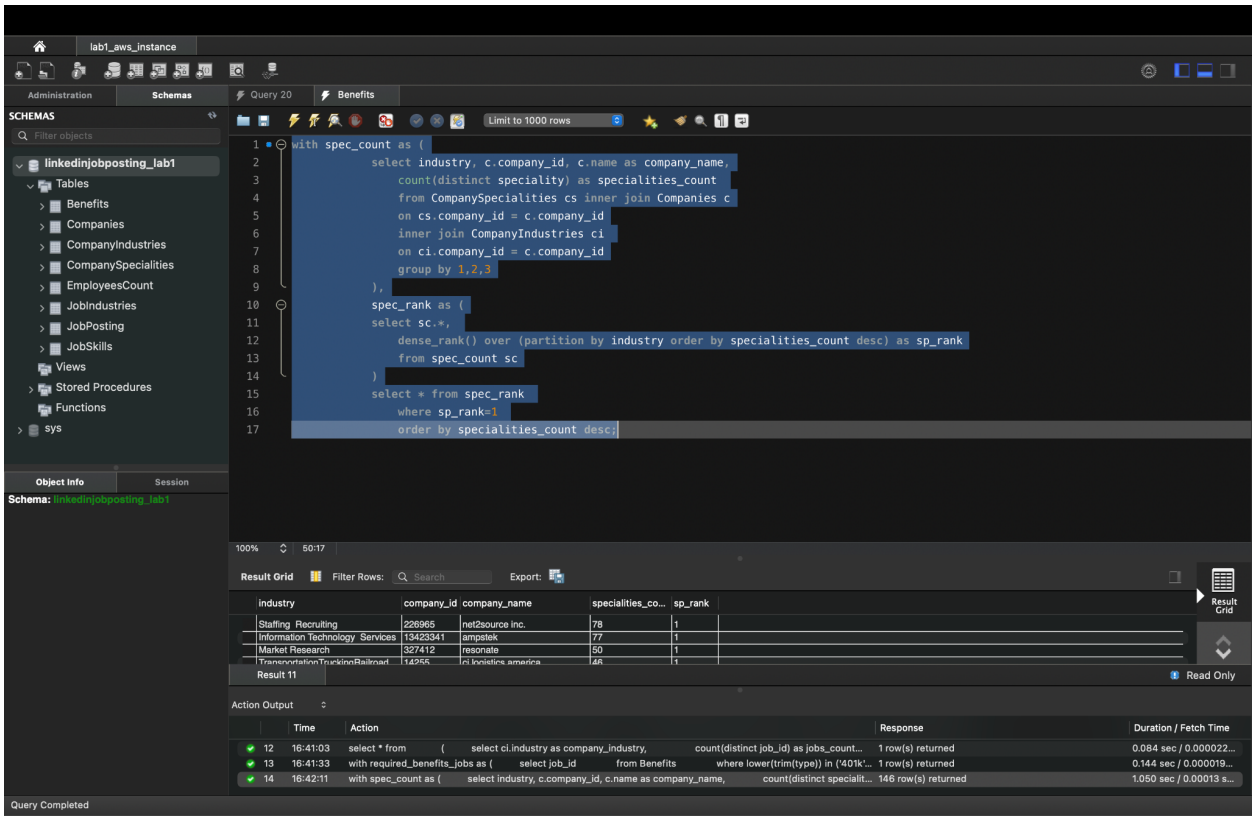


JMETER:

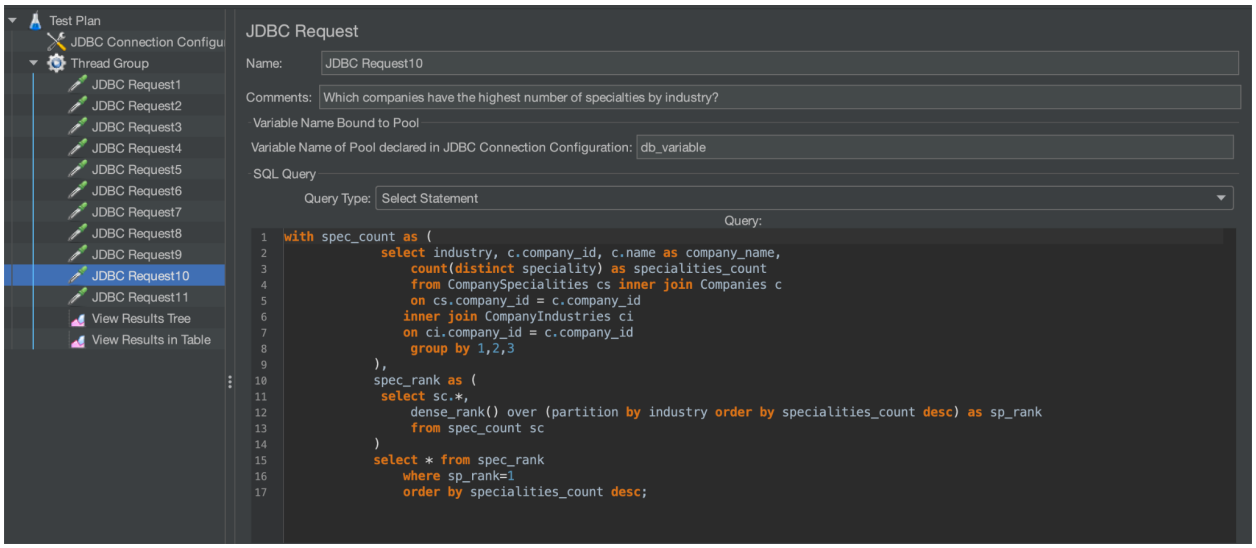


Query 10:

MYSQL:

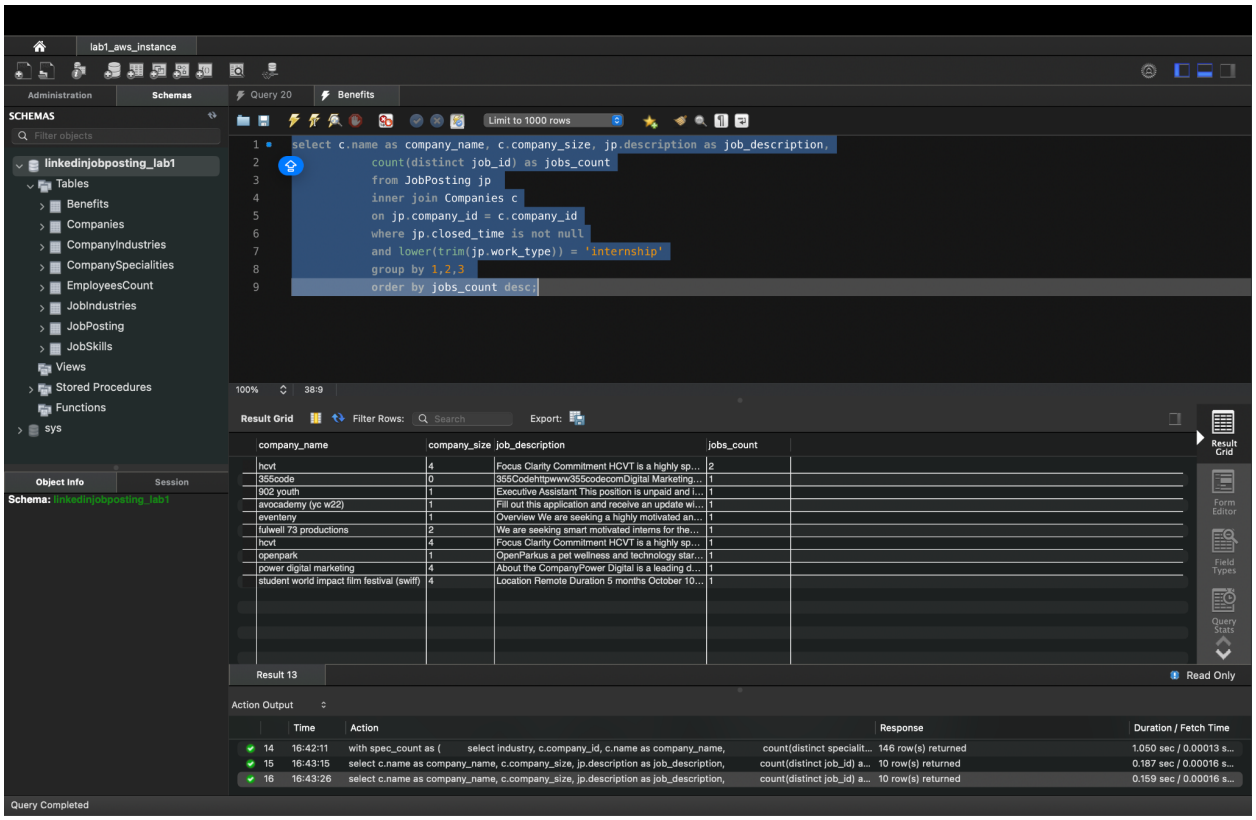


JMETER:

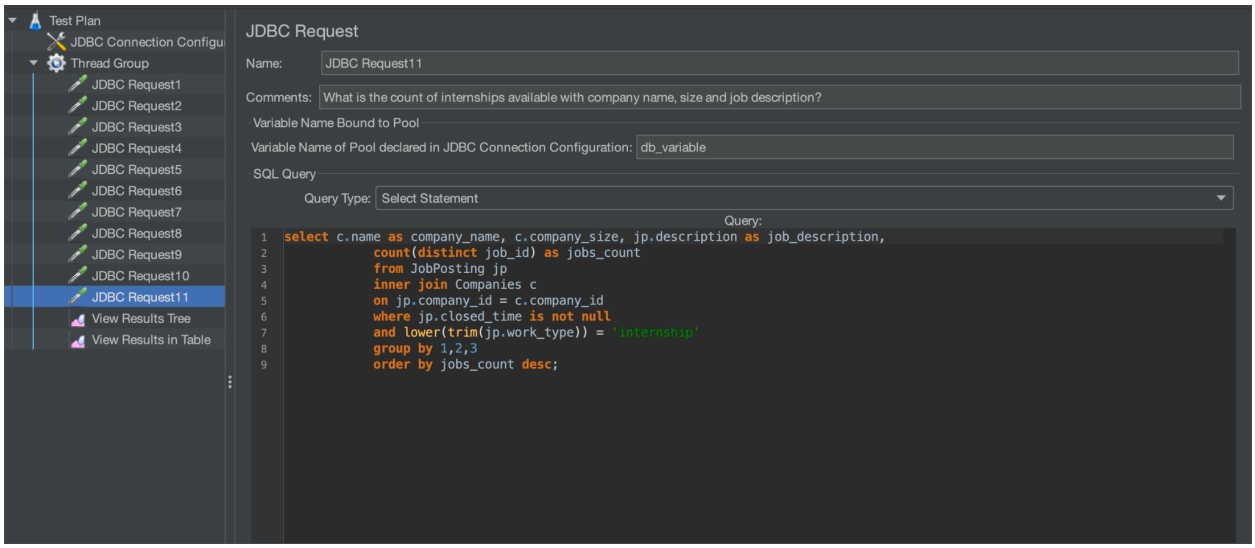


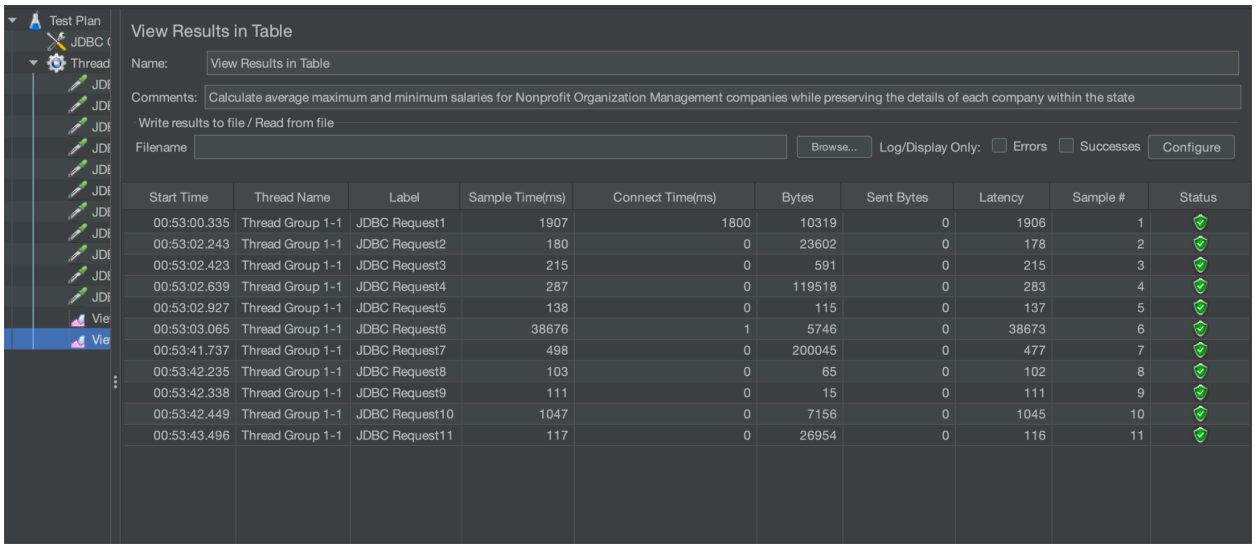
Query 11:

MYSQL:

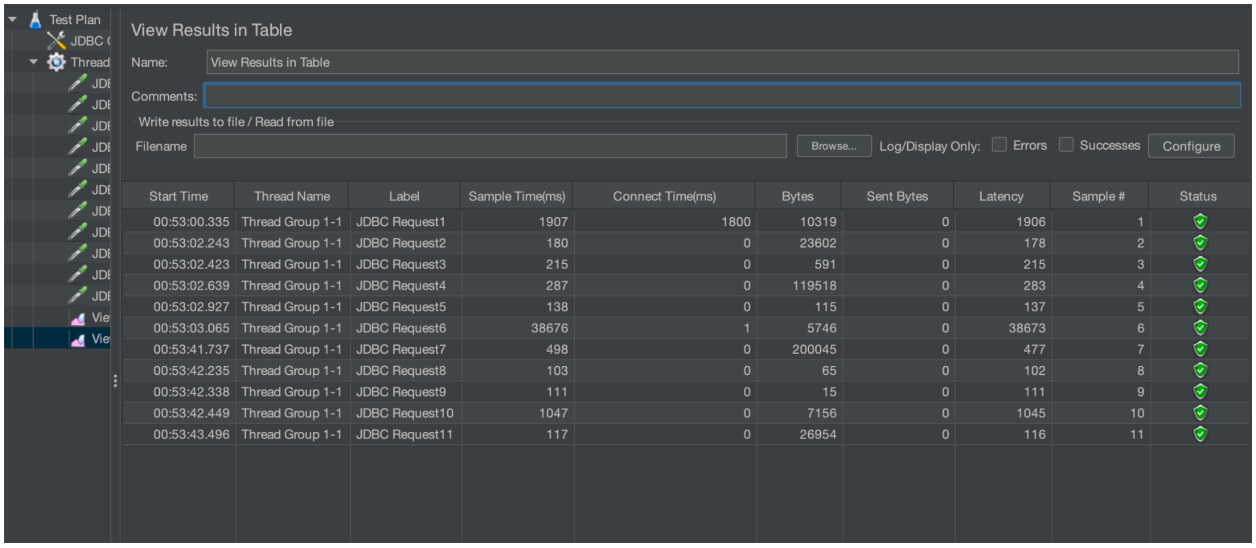


JMETER:


Test Plan
JDBC Connection Configu
Thread Group
JDBC Request1
JDBC Request2
JDBC Request3
JDBC Request4
JDBC Request5
JDBC Request6
JDBC Request7
JDBC Request8
JDBC Request9
JDBC Request10
JDBC Request11
View Results Tree
View Results in Table
JDBC Request
Name:
JDBC Request11
Comments:
What is the count of internships available with company name, size and job description?
Variable Name Bound to Pool
Variable Name of Pool declared in JDBC Connection Configuration:
db_variable
SQL Query
Query Type:
Select Statement
Query:
select c.name as company_name, c.company_size, jp.description as job_description,
count(distinct job_id) as jobs_count
from JobPosting jp
inner join Companies c
on jp.company_id = c.company_id
where jp.closed_time is not null
and lower(trim(jp.work_type)) = 'internship'
group by 1,2,3
order by jobs_count desc;



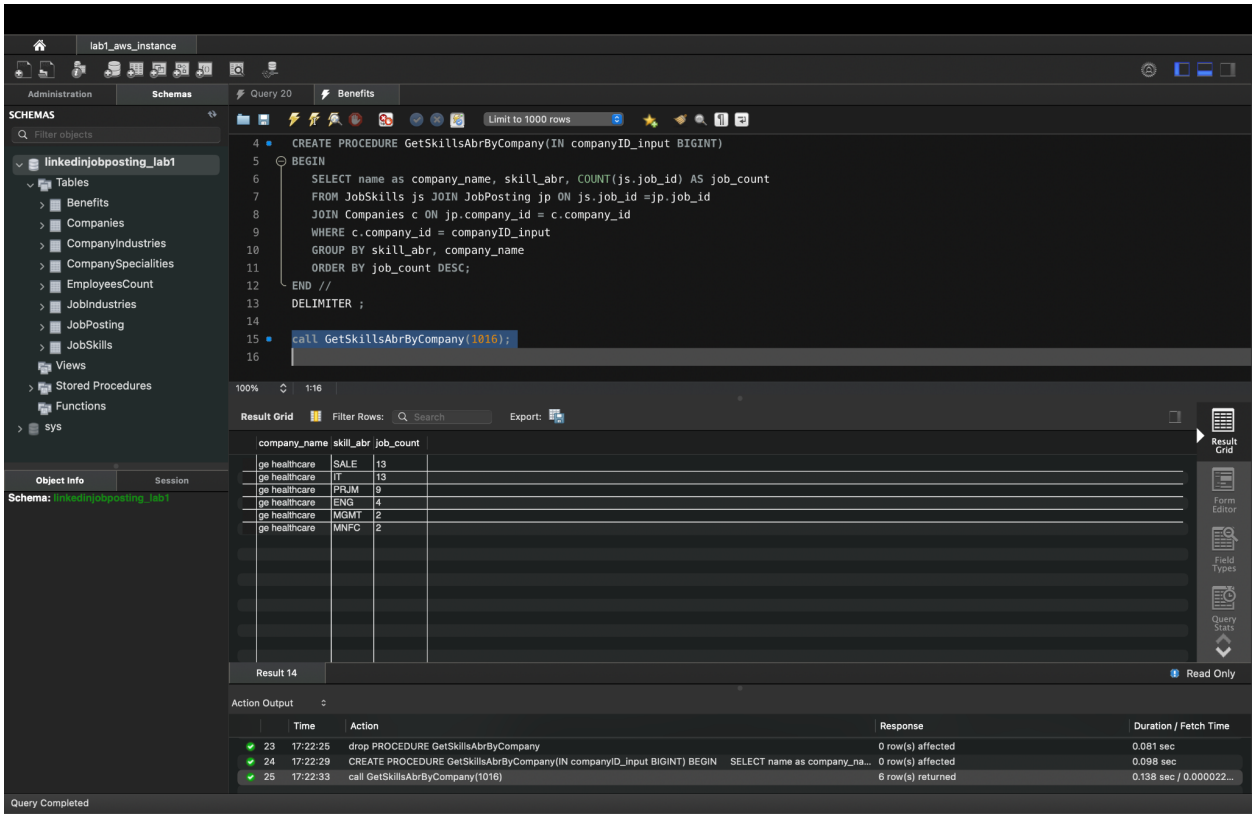
Start Time	Thread Name	Label	Sample Time(ms)	Connect Time(ms)	Bytes	Sent Bytes	Latency	Sample #	Status
00:53:00.335	Thread Group 1-1	JDBC Request1	1907	1800	10319	0	1906	1	
00:53:02.243	Thread Group 1-1	JDBC Request2	180	0	23602	0	178	2	
00:53:02.423	Thread Group 1-1	JDBC Request3	215	0	591	0	215	3	
00:53:02.639	Thread Group 1-1	JDBC Request4	287	0	119518	0	283	4	
00:53:02.927	Thread Group 1-1	JDBC Request5	138	0	115	0	137	5	
00:53:03.065	Thread Group 1-1	JDBC Request6	38676	1	5746	0	38673	6	
00:53:41.737	Thread Group 1-1	JDBC Request7	498	0	200045	0	477	7	
00:53:42.235	Thread Group 1-1	JDBC Request8	103	0	65	0	102	8	
00:53:42.338	Thread Group 1-1	JDBC Request9	111	0	15	0	111	9	
00:53:42.449	Thread Group 1-1	JDBC Request10	1047	0	7156	0	1045	10	
00:53:43.496	Thread Group 1-1	JDBC Request11	117	0	26954	0	116	11	



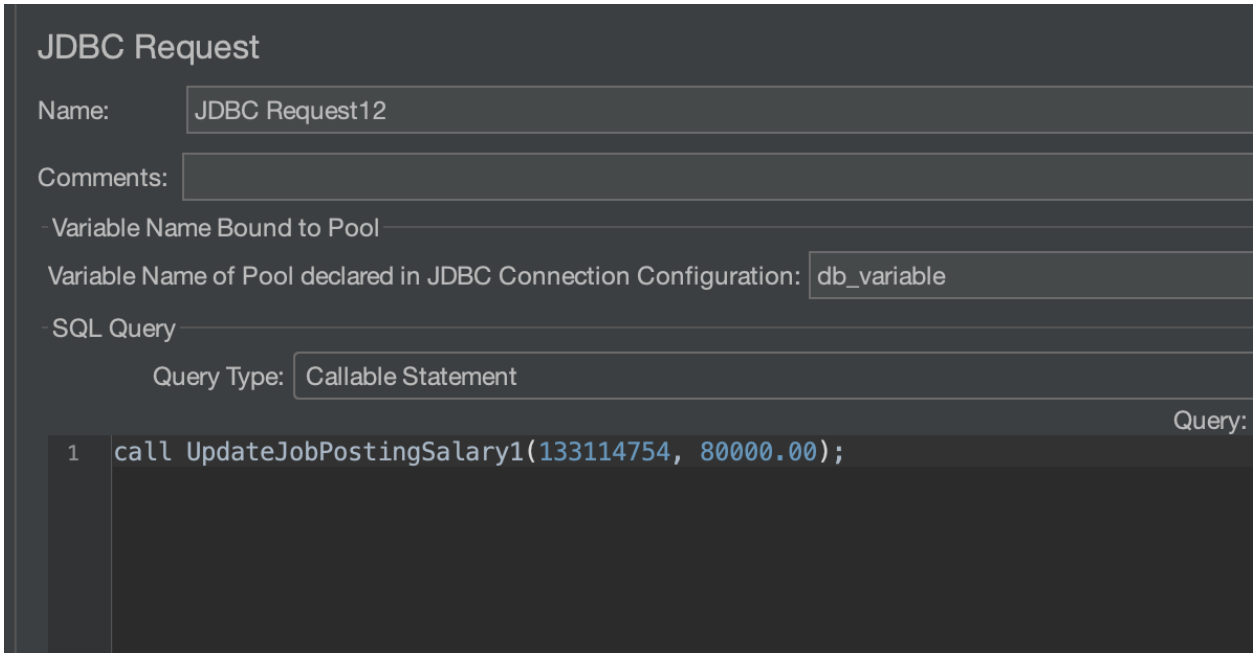
Start Time	Thread Name	Label	Sample Time(ms)	Connect Time(ms)	Bytes	Sent Bytes	Latency	Sample #	Status
00:53:00.335	Thread Group 1-1	JDBC Request1	1907	1800	10319	0	1906	1	
00:53:02.243	Thread Group 1-1	JDBC Request2	180	0	23602	0	178	2	
00:53:02.423	Thread Group 1-1	JDBC Request3	215	0	591	0	215	3	
00:53:02.639	Thread Group 1-1	JDBC Request4	287	0	119518	0	283	4	
00:53:02.927	Thread Group 1-1	JDBC Request5	138	0	115	0	137	5	
00:53:03.065	Thread Group 1-1	JDBC Request6	38676	1	5746	0	38673	6	
00:53:41.737	Thread Group 1-1	JDBC Request7	498	0	200045	0	477	7	
00:53:42.235	Thread Group 1-1	JDBC Request8	103	0	65	0	102	8	
00:53:42.338	Thread Group 1-1	JDBC Request9	111	0	15	0	111	9	
00:53:42.449	Thread Group 1-1	JDBC Request10	1047	0	7156	0	1045	10	
00:53:43.496	Thread Group 1-1	JDBC Request11	117	0	26954	0	116	11	

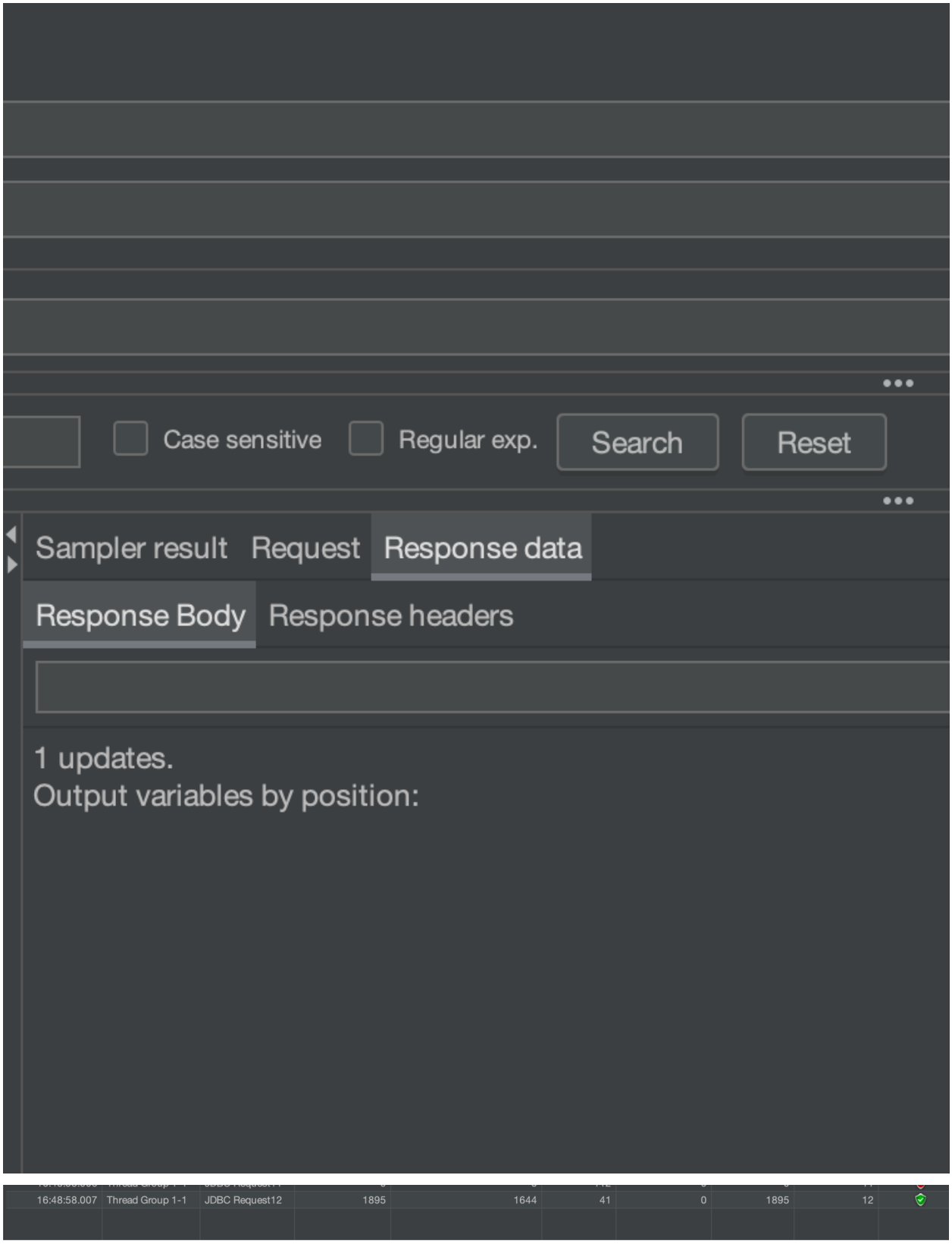
Procedure 1:

MYSQL:


lab1_aws_instance
Administration
Schemas
Query 20
Benefits
SCHEMAS
Filter objects
linkedinjobposting_lab1
Tables
Benefits
Companies
CompanyIndustries
CompanySpecialities
EmployeesCount
JobIndustries
JobPosting
JobSkills
Views
Stored Procedures
Functions
sys
Object Info
Session
Schema: linkedinjobposting_lab1
Limit to 1000 rows
4 CREATE PROCEDURE GetSkillsAbrByCompany(IN companyID_input BIGINT)
5 BEGIN
6 SELECT name as company_name, skill_abr, COUNT(js.job_id) AS job_count
7 FROM JobSkills js JOIN JobPosting jp ON js.job_id =jp.job_id
8 JOIN Companies c ON jp.company_id = c.company_id
9 WHERE c.company_id = companyID_input
10 GROUP BY skill_abr, company_name
11 ORDER BY job_count DESC;
12 END //
13 DELIMITER ;
14
15 call GetSkillsAbrByCompany(1016);
16
100%
1:16
Result Grid
Filter Rows:
Search
Export:
company_name skill_abr job_count
ge healthcare SALE 13
ge healthcare IT 13
ge healthcare PRJM 9
ge healthcare ENG 4
ge healthcare MGMT 2
ge healthcare MNFC 2
Result 14
Read Only
Action Output
Time Action Response Duration / Fetch Time
23 17:22:25 drop PROCEDURE GetSkillsAbrByCompany 0 row(s) affected 0.081 sec
24 17:22:29 CREATE PROCEDURE GetSkillsAbrByCompany(IN companyID_input BIGINT) BEGIN SELECT name as company_na... 0 row(s) affected 0.098 sec
25 17:22:33 call GetSkillsAbrByCompany(1016) 6 row(s) returned 0.138 sec / 0.000022...
Query Completed

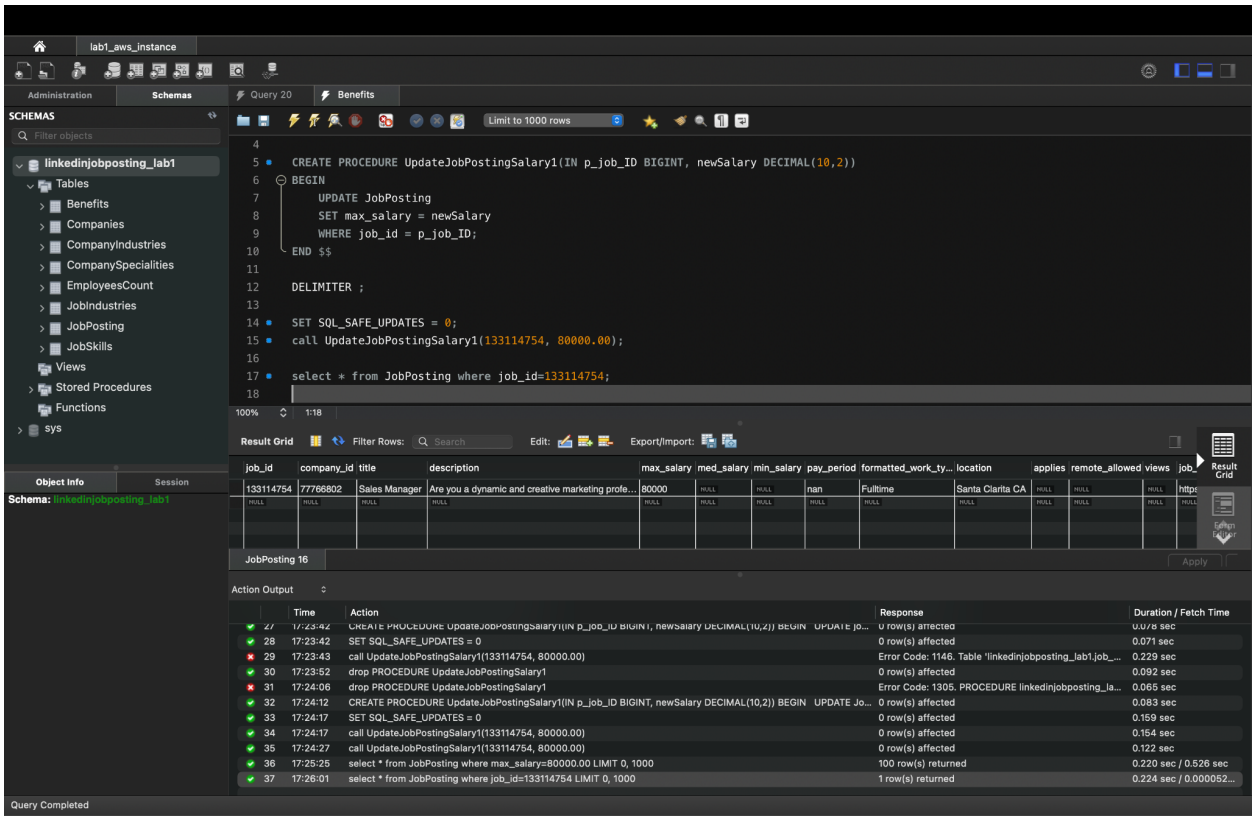
JMETER:


JDBC Request
Name:
JDBC Request12
Comments:
Variable Name Bound to Pool
Variable Name of Pool declared in JDBC Connection Configuration:
db_variable
SQL Query
Query Type:
Callable Statement
Query:
1 call UpdateJobPostingSalary1(133114754, 80000.00);


Case sensitive
Regular exp.
Search
Reset
Sampler result
Request
Response data
Response Body
Response headers
1 updates.
Output variables by position:
16:48:58.007
Thread Group 1-1
JDBC Request12
1895
1644
41
0
1895
12

Procedure 2:

MYSQL:



JMETER:

JDBC Request

Name: JDBC Request12

Comments: What is the count of internships available with company name, size and job description?

Variable Name Bound to Pool

Variable Name of Pool declared in JDBC Connection Configuration: db_variable

SQL Query

Query Type: Callable Statement

Query:

```
1  call GetSkillsAbrByCompany(1016);
```

Parameter values:

Parameter types:

Variable names:

Result variable name: result

Query timeout (s):

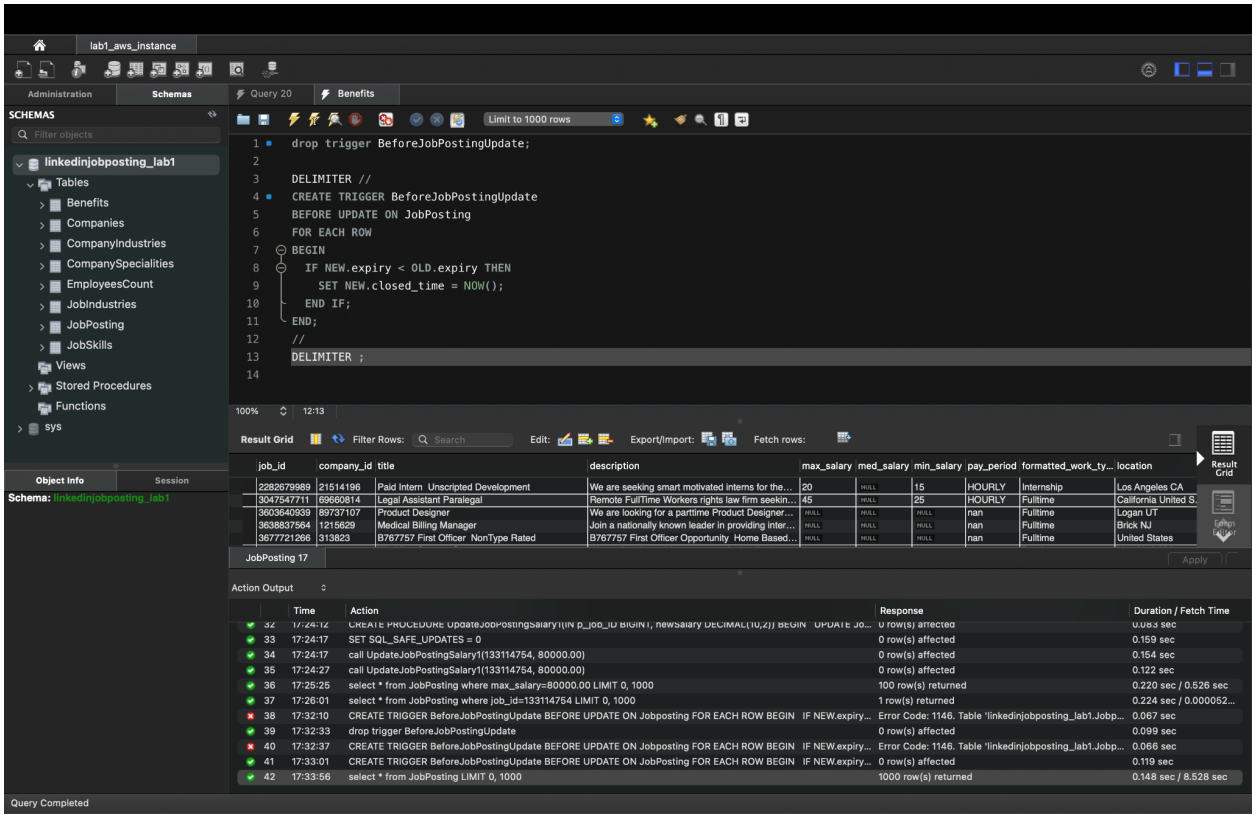
Limit ResultSet:

Handle ResultSet: Store as Object

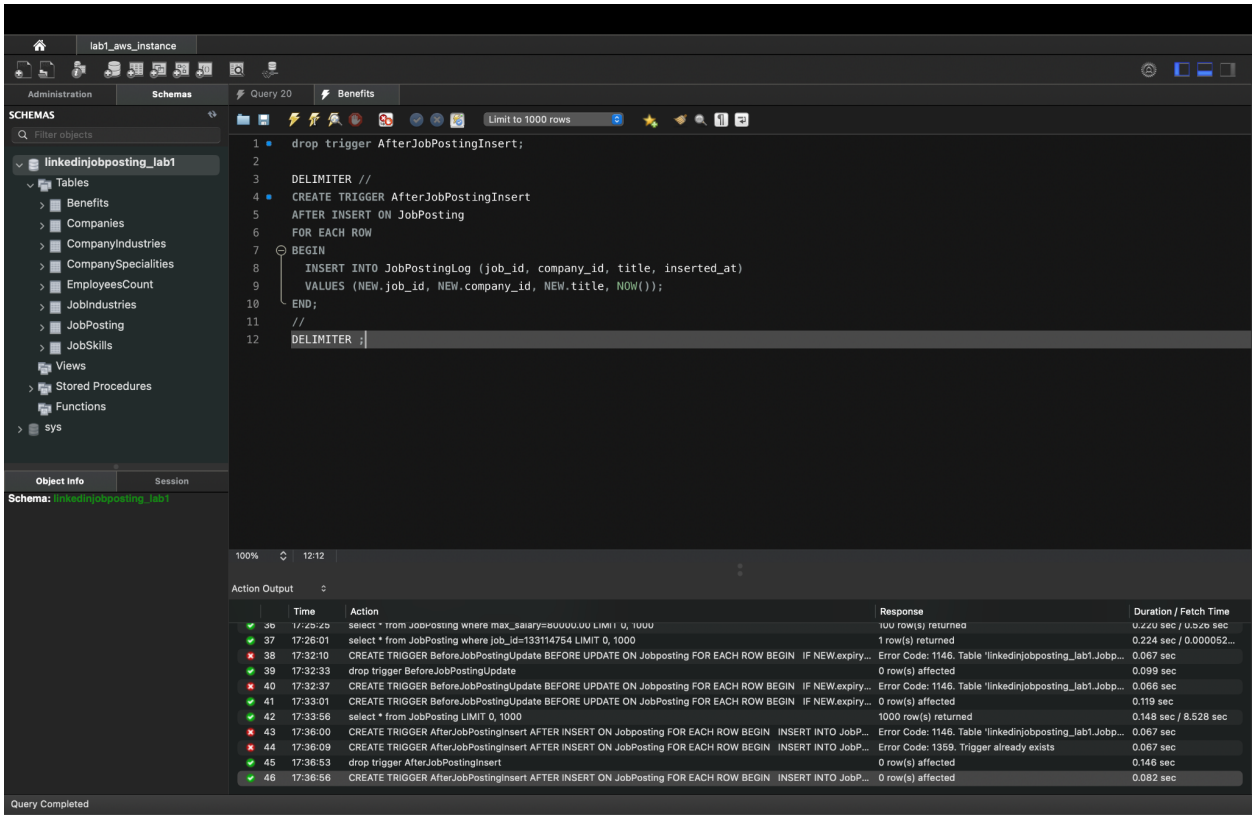

Sampler result
Request
Response data
Response Body
Response headers
company_name skill_abr job_count
ge healthcare SALE 13
ge healthcare IT 13
ge healthcare PRJM 9
ge healthcare ENG 4
ge healthcare MGMT 2
ge healthcare MNFC 2
0 updates.
Output variables by position:


16:46:48.526 Thread Group 1-1 JDBC Request12 2002 1650 200 0 2001 12

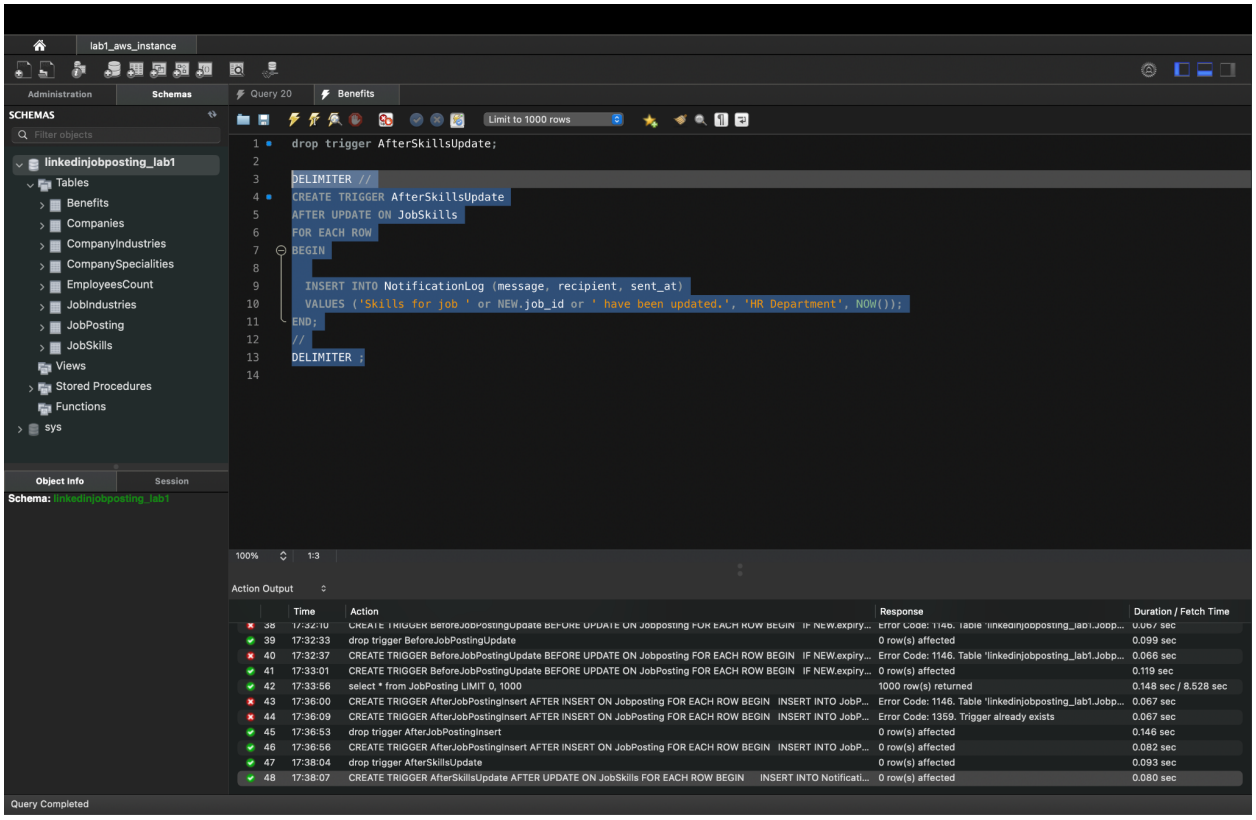
Trigger 1:


lab1_aws_instance
Administration
Schemas
Query 20
Benefits
SCHEMAS
Filter objects
linkedinjobposting_lab1
Tables
Benefits
Companies
CompanyIndustries
CompanySpecialities
EmployeesCount
JobIndustries
JobPosting
JobSkills
Views
Stored Procedures
Functions
sys
Object Info
Session
Schema: linkedinjobposting_lab1
Limit to 1000 rows
1 drop trigger BeforeJobPostingUpdate;
2
3 DELIMITER //
4 CREATE TRIGGER BeforeJobPostingUpdate
5 BEFORE UPDATE ON JobPosting
6 FOR EACH ROW
7 BEGIN
8 IF NEW.expiry < OLD.expiry THEN
9 SET NEW.closed_time = NOW();
10 END IF;
11 END;
12 //
13 DELIMITER ;
14
100% 12:13
Result Grid
Filter Rows:
Search
Edit:
Export/Import:
Fetch rows:
job_id company_id title description max_salary med_salary min_salary pay_period formatted_work_ty... location
2282679989 21514196 Paid Intern Unscripted Development We are seeking smart motivated interns for the... 20 NULL 15 HOURLY Internship Los Angeles CA
3047547711 69660814 Legal Assistant Paralegal Remote FullTime Workers rights law firm seekin... 45 NULL 25 HOURLY Fulltime California United S...
3603640939 89737107 Product Designer We are looking for a parttime Product Designer... NULL NULL NULL nan Fulltime Logan UT
3638837564 1215629 Medical Billing Manager Join a nationally known leader in providing inter... NULL NULL NULL nan Fulltime Brick NJ
3677721266 313823 B767757 First Officer NonType Rated B767757 First Officer Opportunity Home Based... NULL NULL NULL nan Fulltime United States
Result Grid
JobPosting 17
Apply
Action Output
Time Action Response Duration / Fetch Time
33 17:24:17 SET SQL_SAFE_UPDATES = 0 0 row(s) affected 0.159 sec
34 17:24:17 call UpdateJobPostingSalary1(133114754, 80000.00) 0 row(s) affected 0.154 sec
35 17:24:27 call UpdateJobPostingSalary1(133114754, 80000.00) 0 row(s) affected 0.122 sec
36 17:25:25 select * from JobPosting where max_salary=80000.00 LIMIT 0, 1000 100 row(s) returned 0.220 sec / 0.526 sec
37 17:26:01 select * from JobPosting where job_id=133114754 LIMIT 0, 1000 1 row(s) returned 0.224 sec / 0.000052...
38 17:32:10 CREATE TRIGGER BeforeJobPostingUpdate BEFORE UPDATE ON Jobposting FOR EACH ROW BEGIN IF NEW.expiry... Error Code: 1146. Table 'linkedinjobposting_lab1.Jobp... 0.067 sec
39 17:32:33 drop trigger BeforeJobPostingUpdate 0 row(s) affected 0.099 sec
40 17:32:37 CREATE TRIGGER BeforeJobPostingUpdate BEFORE UPDATE ON Jobposting FOR EACH ROW BEGIN IF NEW.expiry... Error Code: 1146. Table 'linkedinjobposting_lab1.Jobp... 0.066 sec
41 17:33:01 CREATE TRIGGER BeforeJobPostingUpdate BEFORE UPDATE ON JobPosting FOR EACH ROW BEGIN IF NEW.expiry... 0 row(s) affected 0.119 sec
42 17:33:56 select * from JobPosting LIMIT 0, 1000 1000 row(s) returned 0.148 sec / 8.528 sec
Query Completed

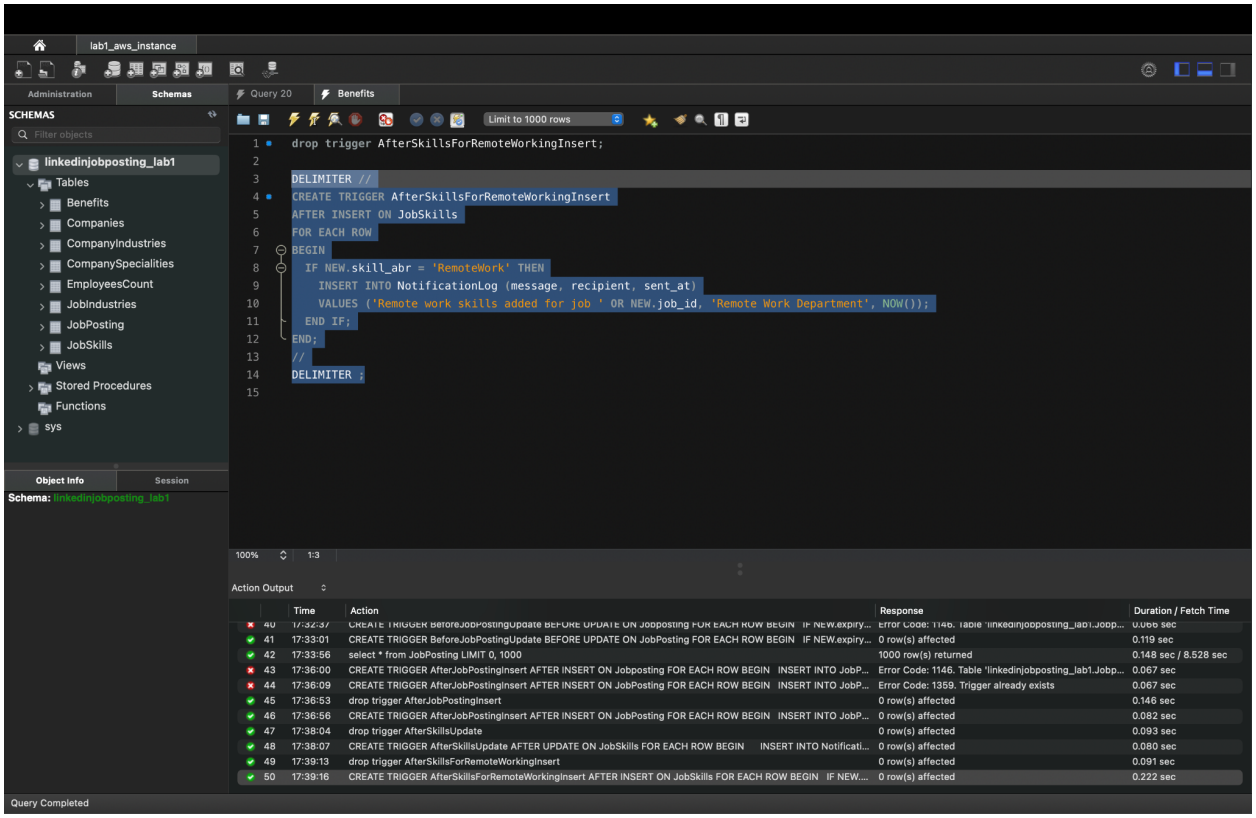
Trigger 2:


lab1_aws_instance
Administration
Schemas
Query 20
Benefits
SCHEMAS
Filter objects
linkedinjobposting_lab1
Tables
Benefits
Companies
CompanyIndustries
CompanySpecialities
EmployeesCount
JobIndustries
JobPosting
JobSkills
Views
Stored Procedures
Functions
sys
Object Info
Session
Schema: linkedinjobposting_lab1
Limit to 1000 rows
1 drop trigger AfterJobPostingInsert;
2
3 DELIMITER //
4 CREATE TRIGGER AfterJobPostingInsert
5 AFTER INSERT ON JobPosting
6 FOR EACH ROW
7 BEGIN
8 INSERT INTO JobPostingLog (job_id, company_id, title, inserted_at)
9 VALUES (NEW.job_id, NEW.company_id, NEW.title, NOW());
10 END;
11 //
12 DELIMITER ;
100% 12:12
Action Output
Time Action Response Duration / Fetch Time
37 17:26:01 select * from JobPosting where job_id=133114754 LIMIT 0, 1000 1 row(s) returned 0.224 sec / 0.000052...
38 17:32:10 CREATE TRIGGER BeforeJobPostingUpdate BEFORE UPDATE ON Jobposting FOR EACH ROW BEGIN IF NEW.expiry... Error Code: 1146. Table 'linkedinjobposting_lab1.Jobp... 0.067 sec
39 17:32:33 drop trigger BeforeJobPostingUpdate 0 row(s) affected 0.099 sec
40 17:32:37 CREATE TRIGGER BeforeJobPostingUpdate BEFORE UPDATE ON Jobposting FOR EACH ROW BEGIN IF NEW.expiry... Error Code: 1146. Table 'linkedinjobposting_lab1.Jobp... 0.066 sec
41 17:33:01 CREATE TRIGGER BeforeJobPostingUpdate BEFORE UPDATE ON JobPosting FOR EACH ROW BEGIN IF NEW.expiry... 0 row(s) affected 0.119 sec
42 17:33:56 select * from JobPosting LIMIT 0, 1000 1000 row(s) returned 0.148 sec / 8.528 sec
43 17:36:00 CREATE TRIGGER AfterJobPostingInsert AFTER INSERT ON Jobposting FOR EACH ROW BEGIN INSERT INTO JobP... Error Code: 1146. Table 'linkedinjobposting_lab1.Jobp... 0.067 sec
44 17:36:09 CREATE TRIGGER AfterJobPostingInsert AFTER INSERT ON JobPosting FOR EACH ROW BEGIN INSERT INTO JobP... Error Code: 1359. Trigger already exists 0.067 sec
45 17:36:53 drop trigger AfterJobPostingInsert 0 row(s) affected 0.146 sec
46 17:36:56 CREATE TRIGGER AfterJobPostingInsert AFTER INSERT ON JobPosting FOR EACH ROW BEGIN INSERT INTO JobP... 0 row(s) affected 0.082 sec
Query Completed

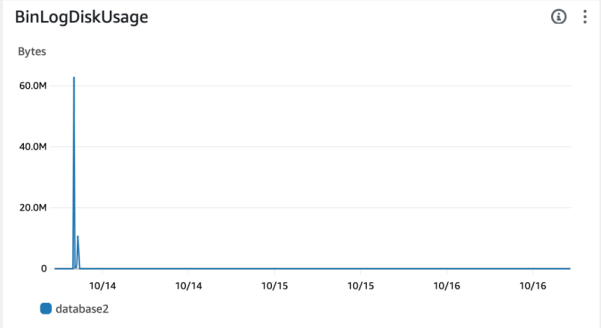
Trigger 3:

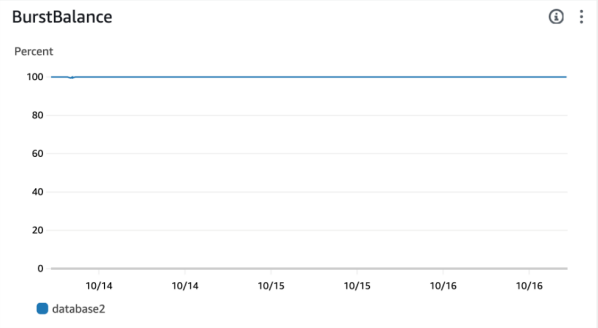


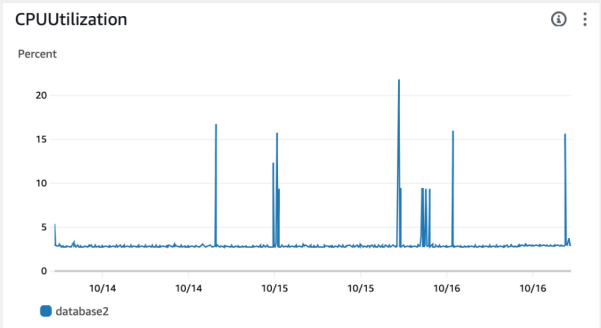
Trigger 4:

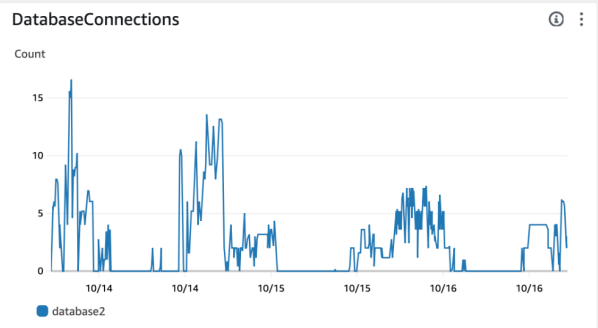


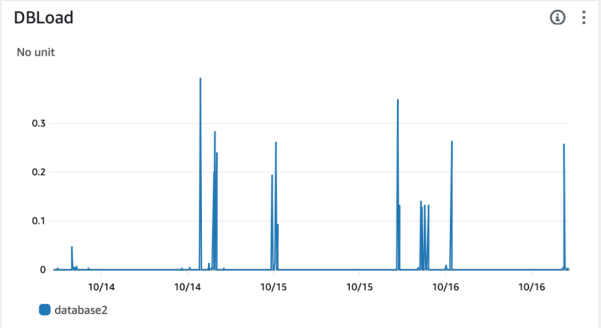
AWS Performance Measurements:

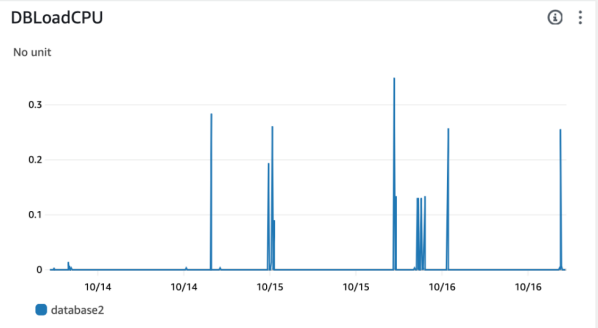

BinLogDiskUsage
Bytes
60.0M
40.0M
20.0M
0
10/14
10/14
10/15
10/15
10/16
10/16
database2

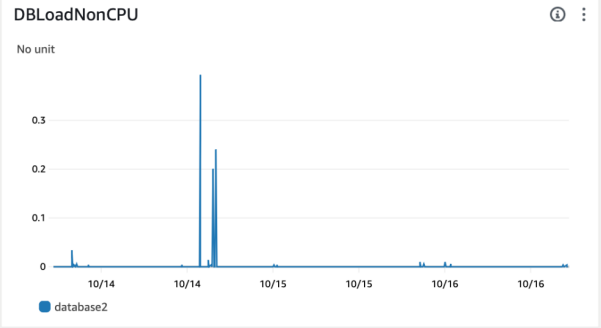

BurstBalance
Percent
100
80
60
40
20
0
10/14
10/14
10/15
10/15
10/16
10/16
database2

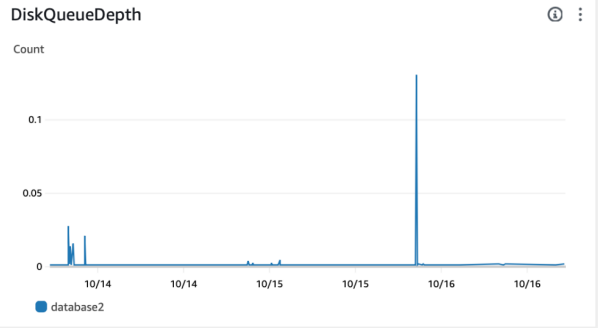

CPUUtilization
Percent
20
15
10
5
0
10/14
10/14
10/15
10/15
10/16
10/16
database2

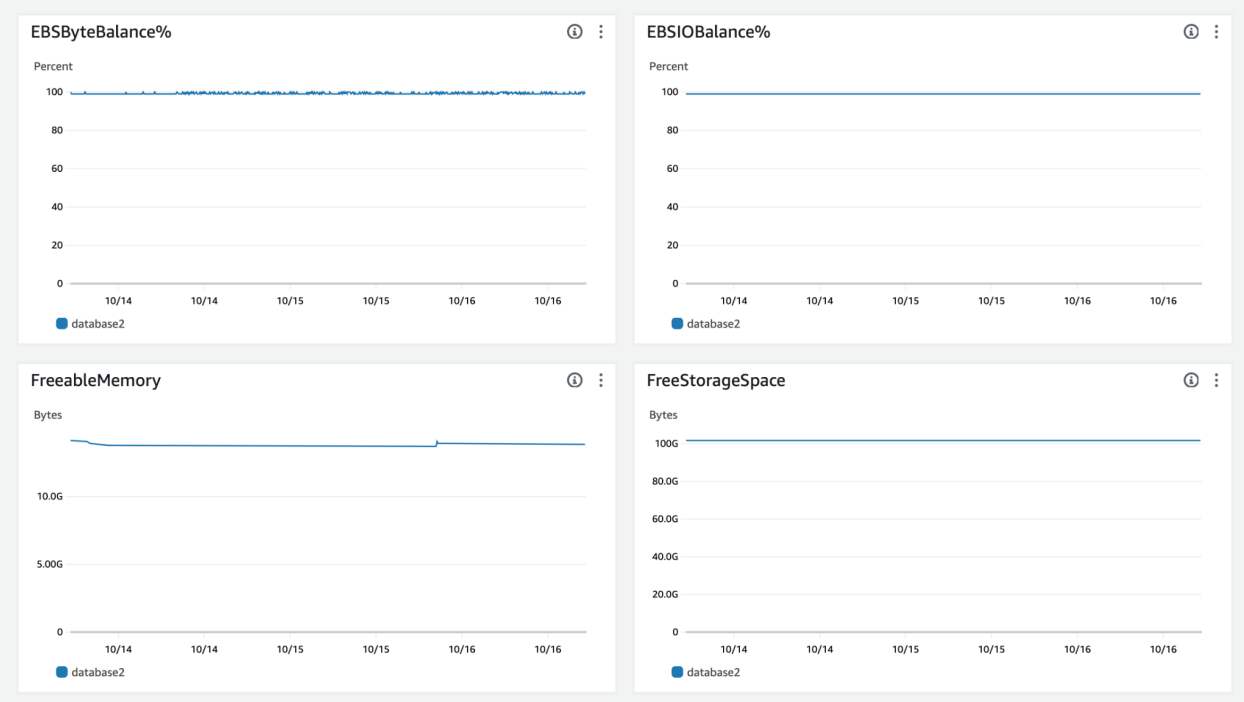

DatabaseConnections
Count
15
10
5
0
10/14
10/14
10/15
10/15
10/16
10/16
database2


DBLoad
No unit
0.3
0.2
0.1
0
10/14
10/14
10/15
10/15
10/16
10/16
database2

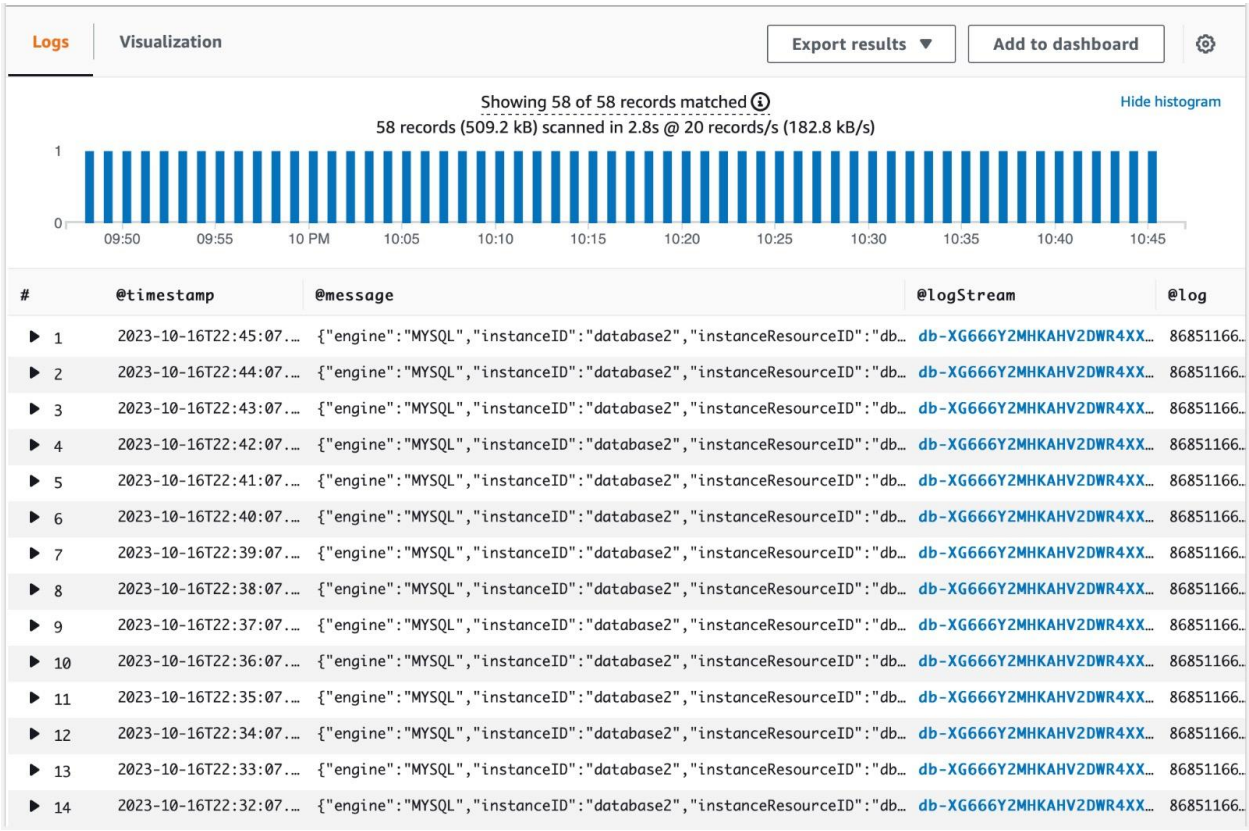

DBLoadCPU
No unit
0.3
0.2
0.1
0
10/14
10/14
10/15
10/15
10/16
10/16
database2


DBLoadNonCPU
No unit
0.3
0.2
0.1
0
10/14
10/14
10/15
10/15
10/16
10/16
database2


DiskQueueDepth
Count
0.1
0.05
0
10/14
10/14
10/15
10/15
10/16
10/16
database2



Logs from AWS cloud watch



#	@timestamp	@message	@logStream	@log
▶ 1	2023-10-16T22:45:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 2	2023-10-16T22:44:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 3	2023-10-16T22:43:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 4	2023-10-16T22:42:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 5	2023-10-16T22:41:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 6	2023-10-16T22:40:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 7	2023-10-16T22:39:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 8	2023-10-16T22:38:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 9	2023-10-16T22:37:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 10	2023-10-16T22:36:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 11	2023-10-16T22:35:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 12	2023-10-16T22:34:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 13	2023-10-16T22:33:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…
▶ 14	2023-10-16T22:32:07.…	{"engine":"MYSQL","instanceID":"database2","instanceResourceID":"db…	db-XG666Y2MHKAHV2DWR4XX…	86851166…