

Purpose- SQL Query Execution in AWS

Query 1:

MYSQL:

The screenshot shows the DBeaver interface with the following details:

- Top Bar:** Administration, Schemas, Query 20, Benefits.
- Schemas Panel:** Shows the schema `linkedinjobposting_lab1` selected, containing Tables, Views, Stored Procedures, Functions, and sys.
- Query Editor:** Displays a complex SQL query using a WITH clause to calculate average maximum and minimum salaries by state for companies in the 'Nonprofit Organization Management' industry. The query includes joins between Companies, CompanyIndustries, and JobPosting tables.
- Result Grid:** Shows the execution results for the query, including columns: state, max_salary, min_salary, avg_max_salary, and avg_min_salary. The results are grouped by state, with multiple rows for each state due to the partitioned aggregate functions.
- Action Output:** Shows the history of actions taken, including the execution of the query and its results.

Time	Action	Response	Duration / Fetch Time			
16:27:31	WITH ITCompanies AS (SELECT c.state, jp.max_salary, jp.min_salary	FROM Companies c	JOIN JobPosting jp ON c.company_id = jp.company_id	Error Code: 1046. No database selected Select the database	0.083 sec
16:27:51	use linkedinjobposting_lab1				0 row(s) affected	0.064 sec
16:27:55	WITH ITCompanies AS (SELECT c.state, jp.max_salary, jp.min_salary	FROM Companies c	JOIN JobPosting jp ON c.company_id = jp.company_id	248 row(s) returned	0.086 sec / 0.0004...

JMETER:

Test Plan

JDBC Connection Configur.

Thread Group

- JDBC Request1
 - View Results Tree
 - View Results in Table
- JDBC Request2
 - View Results Tree
 - View Results in Table
- JDBC Request3
 - View Results Tree
 - View Results in Table
- JDBC Request4
 - View Results Tree
 - View Results in Table
- JDBC Request5
 - View Results Tree
 - View Results in Table
- JDBC Request6
 - View Results Tree
 - View Results in Table
- JDBC Request7
 - View Results Tree
 - View Results in Table
- JDBC Request8
 - View Results Tree
 - View Results in Table
- JDBC Request9
 - View Results Tree
 - View Results in Table

JDBC Request

Name: JDBC Request1

Comments: Calculate average maximum and minimum salaries for Nonprofit Organization Management companies while preserving the details of each company within the state

Variable Name Bound to Pool

Variable Name of Pool declared in JDBC Connection Configuration: db_variable

SQL Query

Query Type: Select Statement

Query:

```
1 WITH ITCompanies AS (
2     SELECT c.state, jp.max_salary, jp.min_salary
3     FROM Companies c
4     JOIN CompanyIndustries ci ON c.company_id = ci.company_id
5     JOIN JobPosting jp ON c.company_id = jp.company_id
6     WHERE ci.industry = 'Nonprofit Organization Management'
7 )
8     SELECT state,
9         max_salary AS max_salary,
10        min_salary AS min_salary,
11        AVG(max_salary) OVER (PARTITION BY state) AS avg_max_salary,
12        AVG(min_salary) OVER (PARTITION BY state) AS avg_min_salary
13    FROM ITCompanies;
```

Test Plan

- JDBC Connection Configur.
- Thread Group
 - JDBC Request1
 - View Results Tree
 - View Results in Table**
 - JDBC Request2
 - View Results Tree
 - View Results in Table
 - JDBC Request3
 - JDBC Request4
 - JDBC Request5
 - JDBC Request6
 - JDBC Request7
 - JDBC Request8
 - JDBC Request9

View Results in Table

Name: View Results in Table

Comments: Calculate average maximum and minimum salaries for Nonprofit Organization Management companies while preserving the details of each company within the state.

Write results to file / Read from file

Filename: Log/Display Only: Errors Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency
1	00:12:47.955	Thread Group 1...	JDBC Request1	1775	✓	10319	0	1767

Query 2:

MYSQL:

lab1_aws_instance

Administration Schemas

SCHEMAS

Filter objects

linkedinijobposting_lab1

- Tables
 - Benefits
 - Companies
 - CompanyIndustries
 - CompanySpecialties
 - EmployeesCount
 - JobIndustries
 - JobPosting
 - JobSkills
 - Views
- Stored Procedures
- Functions
- sys

Object Info Session

Schema: linkedinijobposting_lab1

```

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';
10

```

Result Grid

name	city	avg_employee_cou...
advantagecare physicians	new york	12746.3818
gen ii fund services	new york	12746.3818
gen ii fund services	new york	12746.3818
panama love	new york	12746.3818
fohr	new york	12746.3818
atlas search	new york	12746.3818
atlas search	new york	12746.3818
atlas search	new york	12746.3818
atlas search	new york	12746.3818
atlas search	new york	12746.3818
advantagecare physicians	new york	12746.3818
gen ii fund services	new york	12746.3818
baccarat hotel & residen...	new york	12746.3818
cmq	new york	12746.3818

Result 2

Action Output

Time	Action	Response	Duration / Fetch Time		
1 16:27:31	WITH ITCompanies AS (SELECT c.state, jp.max_salary, jp.min_salary	FROM Companies c	JOIN...	Error Code:1046. No database selected Select the de... 0.083 sec
2 16:27:51	use linkedinijobposting_lab1				0 row(s) affected 0.064 sec
3 16:27:55	WITH ITCompanies AS (SELECT c.state, jp.max_salary, jp.min_salary	FROM Companies c	JOIN...	248 row(s) returned 0.086 sec / 0.00004...
4 16:29:39	WITH CompanyEmployeeCounts AS (SELECT c.name, c.city, ec.employee_count, AVG(ec.employee_c...			641 row(s) returned 0.177 sec / 0.0069 sec

Query Completed

JMETER:

The screenshot shows the JMeter Test Plan interface. On the left, there's a tree view with nodes like 'Test Plan', 'JDBC Connection Configuration', 'Thread Group', and several 'JDBC Request' nodes (Request1 through Request11). The 'JDBC Request2' node is selected. The main panel displays the configuration for 'JDBC Request2'. It includes fields for 'Name' (set to 'JDBC Request2'), 'Comments' (describing the query), 'Variable Name Bound to Pool' (set to 'db_variable'), and 'SQL Query'. The 'Query Type' is set to 'Select Statement'. The SQL code is as follows:

```

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';

```

Query 3:

MYSQL:

The screenshot shows MySQL Workbench. The left sidebar shows the database schema with a schema named 'linkedinjobposting_lab1'. The central pane displays a query editor with the following SQL code:

```

1 WITH CompanyJobSkills AS (
2     SELECT c.name AS 'Job Roll', cs.specialty, js.skill_abr,
3         ROW_NUMBER() OVER (PARTITION BY c.name, cs.specialty, js.skill_abr) AS rn
4     FROM Companies c
5     JOIN CompanySpecialties cs ON c.company_id = cs.company_id
6     JOIN JobPosting jp ON c.company_id = jp.company_id
7     JOIN JobSkills js ON jp.job_id = js.job_id
8 )
9 SELECT 'Job Roll', specialty, skill_abr
10    FROM CompanyJobSkills
11   WHERE specialty = 'Financial Services' AND skill_abr = 'ACCT'
12   AND rn = 1
13 ORDER BY 'Job Roll';

```

The results grid shows the following data:

Job Roll	specialty	skill_abr
Job Roll	Financial Services	ACCT
Job Roll	Financial Services	ACCT
Job Roll	Financial services	ACCT
Job Roll	Financial Services	ACCT
Job Roll	Financial Services	ACCT
Job Roll	Financial Services	ACCT
Job Roll	Financial Services	ACCT
Job Roll	Financial Services	ACCT

The bottom pane shows the 'Action Output' table with three rows of log entries.

JMETER:

JDBC Request

Name: JDBC Request3

Comments: Find companies job roll with a specific speciality that also have job postings with a certain skill requirement

Variable Name Bound to Pool

Variable Name of Pool declared in JDBC Connection Configuration: db_variable

SQL Query

Query Type: Select Statement

```

1 WITH CompanyJobSkills AS (
2     SELECT c.name AS `Job Roll`, cs.speciality, js.skill_abr,
3        ROW_NUMBER() OVER (PARTITION BY c.name, cs.speciality, js.skill_abr) AS rn
4     FROM Companies c
5     JOIN CompanySpecialties cs ON c.company_id = cs.company_id
6     JOIN JobPosting jp ON c.company_id = jp.company_id
7     JOIN JobSkills js ON jp.job_id = js.job_id
8 )
9     SELECT `Job Roll`, speciality, skill_abr
10    FROM CompanyJobSkills
11   WHERE speciality = 'Financial Services' AND skill_abr = 'ACCT'
12   AND rn = 1
13   ORDER BY `Job Roll`;

```

Query4:

MYSQL:

Query 20 Benefits

```

1 WITH num_benefits as (
2     SELECT job_id, count(distinct type) as num_benefits
3     FROM Benefits
4     GROUP BY 1
5 ),
6 -- # Getting avg number of benefits in the next CTE
7 avg_benefits as (
8     SELECT avg(num_benefits) as avg_benefits_count
9     FROM num_benefits
10 )
11     SELECT n.job_id, jp.company_id, c.name as company_name, jp.title, num_benefits
12     FROM num_benefits n
13     INNER JOIN JobPosting jp
14     ON n.job_id = jp.job_id
15     INNER JOIN Companies c
16     ON jp.company_id = c.company_id
17     WHERE num_benefits >= (select avg_benefits_count from avg_benefits)
18     ORDER BY num_benefits desc;

```

Result Grid

Job Roll	speciality	skill_abr
Job Roll	Financial Services	ACCT
Job Roll	Financial Services	ACCT
Job Roll	Financial services	ACCT
Job Roll	Financial Services	ACCT
Job Roll	Financial Services	ACCT
Job Roll	Financial Services	ACCT
Job Roll	Financial Services	ACCT
Job Roll	Financial Services	ACCT

Action Output

Time	Action	Response	Duration / Fetch Time
10:27:51	use linkedinjobposting_lab1	Error code: 1040. No database selected. Selecting default database.	0.000 sec
16:27:51	use linkedinjobposting_lab1	0 row(s) affected	0.064 sec
16:27:55	WITH Companies AS (248 row(s) returned	0.086 sec / 0.00004...
16:29:39	WITH CompanyEmployeeCounts AS (641 row(s) returned	0.177 sec / 0.0069 sec

JMETER:

JDBC Request

Name: JDBC Request4

Comments: Average # benefits per job and details of jobs offering higher than the average # of benefits

Variable Name Bound to Pool

Variable Name of Pool declared in JDBC Connection Configuration: db_variable

SQL Query

Query Type: Select Statement

```

1 WITH num_benefits AS (
2     SELECT job_id, count(distinct type) AS num_benefits
3         FROM Benefits
4             GROUP BY 1
5 ),
6 -- # Getting avg number of benefits in the next CTE
7 avg_benefits AS (
8     SELECT avg(num_benefits) AS avg_benefits_count
9         FROM num_benefits
10 )
11 SELECT n.job_id, jp.company_id, c.name AS company_name, jp.title, num_benefits
12     FROM num_benefits n
13 INNER JOIN JobPosting jp
14 ON n.job_id = jp.job_id
15 INNER JOIN Companies c
16 ON jp.company_id = c.company_id
17 WHERE num_benefits >= (SELECT avg_benefits_count FROM avg_benefits)
18 ORDER BY num_benefits DESC;

```

Query 5:

MYSQL:

Query 20 Benefits

```

1 with avg_salary_by_industry AS (
2     select industry, round(avg(min_salary),3) as avg_salary
3         from JobPosting jp
4             inner join CompanyIndustries ci
5                 on jp.company_id = ci.company_id
6                 where industry is not null
7                 and coalesce(min_salary,0) > 0
8                     group by 1
9 ),
10 ranking_salary AS (
11     select industry, avg_salary,
12         dense_rank() over (order by avg_salary desc) as salary_rank
13             from avg_salary_by_industry
14 )
15 select * from ranking_salary where salary_rank<=3
16         order by salary_rank;

```

Result Grid

industry	avg_salary	salary_rank
Veterinary	2000000.00	1
Investment Banking	1850000.00	2
Public Policy	183310.00	3

Action Output

Time	Action	Response	Duration / Fetch Time
6 16:33:51	WITH CompanyJobSkills AS (SELECT c.name AS 'Job Roll', cs.specialty, js.skill_abr, ROW_NUMBER() O...	0.197 sec / 0.000041...
7 16:35:43	WITH num_benefits AS (SELECT job_id, count(distinct type) as num_benefits FROM Benefits GROU...	0.243 sec / 0.192 sec
8 16:36:27	with avg_salary_by_industry AS (select industry, round(avg(min_salary),3) as avg_salary from JobPost...	0.104 sec / 0.000041...

JMETER:

JDBC Request

Name: JDBC Request5

Comments: Find the average salary by each job industry. Rank to find the top 3 industries by avg salary (joins and window function)

Variable Name Bound to Pool

Variable Name of Pool declared in JDBC Connection Configuration: db_variable

SQL Query

Query Type: Select Statement

```

1 with avg_salary_by_industry as (
2     select industry, round(avg(min_salary),3) as avg_salary
3         from JobPosting jp
4             inner join CompanyIndustries ci
5                 on jp.company_id = ci.company_id
6                     where industry is not null
7                         and coalesce(min_salary,0) > 0
8                         group by 1
9 ),
10 ranking_salary as (
11     select industry, avg_salary,
12         dense_rank() over (order by avg_salary desc) as salary_rank
13             from avg_salary_by_industry
14 )
15 select * from ranking_salary where salary_rank<=3
16     order by salary_rank;

```

Query 6:

MYSQL:

Query 20 Benefits

Limit to 1000 rows

```

1 with industry_skill_count as (
2     select ci.industry as company_industry, jp.job_id, js.skill_abr
3         from JobSkills js
4             inner join JobPosting jp
5                 on js.job_id = jp.job_id
6                     inner join CompanyIndustries ci
7                         on jp.company_id = ci.company_id
8                             group by 1,2,3
9 ),
10 skill_count_raw as (
11     select company_industry, skill_abr,
12         count(distinct job_id) as job_count
13             from industry_skill_count
14                 group by 1,2
15 ),
16 skill_ranking as (
17     select company_industry, skill_abr, job_count,
18         dense_rank() over (partition by company_industry order by job_count desc) as skill_rank
19             from skill_count_raw
20 )
21 select * from skill_ranking where skill_rank=1
22     order by company_industry, job_count desc;

```

Result Grid

company_industry	skill_abr	job_count	skill_rank
Accounting	FIN	75	1
AirlinesAviation	MNFC	4	1
AirlinesAviation	MGMT	4	1
Alternative Medicine	RFD	1	1

Action Output

Time	Action	Response	Duration / Fetch Time
8	with avg_salary_by_industry as (select industry, round(avg(min_salary),3) as avg_salary from JobPosting... 3 row(s) returned	from JobPosting... 3 row(s) returned	0.104 sec / 0.000041...
9	with industry_skill_count as (select ci.industry as company_industry, jp.job_id, js.skill_abr from JobSkills... Error Code: 2013. Lost connection to MySQL server d...	from JobSkills... Error Code: 2013. Lost connection to MySQL server d...	30.004 sec
10	with industry_skill_count as (select ci.industry as company_industry, jp.job_id, js.skill_abr from JobSkills... 217 row(s) returned	from JobSkills... 217 row(s) returned	38.736 sec / 0.00006...

JMETER:

JDBC Request

Name: JDBC Request6

Comments: Find the top skill for each job industry

Variable Name Bound to Pool

Variable Name of Pool declared in JDBC Connection Configuration: db_variable

SQL Query

Query Type: Select Statement

```

1 with industry_skill_count as (
2     select ci.industry as company_industry, jp.job_id, js.skill_abr
3         from JobSkills js
4             inner join JobPosting jp
5                 on js.job_id = jp.job_id
6                     inner join CompanyIndustries ci
7                         on jp.company_id = ci.company_id
8                             group by 1,2,3
9 ),
10 skill_count_raw as (
11     select company_industry, skill_abr,
12         count(distinct job_id) as job_count
13             from industry_skill_count
14                 group by 1,2
15 ),
16 skill_ranking as (
17     select company_industry, skill_abr, job_count,
18         dense_rank() over(partition by company_industry order by job_count desc) as skill_rank
19     from skill_count_raw
20 )

```

Query 7:

MYSQL:

lab1_aws_instance

Administration Schemas

SCHEMAS Filter objects

linkedInjobposting_lab1

- Tables
 - Benefits
 - Companies
 - CompanyIndustries
 - CompanySpecialties
 - EmployeesCount
 - JobIndustries
 - JobPosting
 - JobSkills
 - Views
- Stored Procedures
- Functions
- sys

Object Info Session Schema: linkedInjobposting_lab1

```

13 # Using row number since we want unique row for a rank, dense rank may show duplicates
14 # Creating 2 ranks - 1 for earliest and 1 for latest
15 row_number() over (partition by company_id order by date_time_stamp) as min_rank,
16 row_number() over (partition by company_id order by date_time_stamp desc) as max_rank
17 from max_min_emp_count_raw
18 ),
19 growth_calc as (
20     select coalesce(mn.company_id, mx.company_id) as company_id,
21 # Including condition to make growth % as 0 if denominator is 0, else do the actual percentage change
22     case when coalesce(mn.earliest_emp_count,0)>0 then
23         (coalesce(mx.latest_emp_count,0) - coalesce(mn.earliest_emp_count,0))/100/coalesce(mn.earliest_emp_count,0)
24     else 0 end as emp_count_growth_change_percentage
25     from (select company_id, employee_count as earliest_emp_count from min_max_rank where min_rank=1) mn
26         left join
27             (select company_id, employee_count as latest_emp_count from min_max_rank where max_rank=1) mx
28         on mn.company_id = mx.company_id
29     )
30     select c.name as company_name, gc.* from
31     growth_calc gc inner join Companies c
32     on gc.company_id = c.company_id
33     order by emp_count_growth_change_percentage desc;
34

```

Result Grid

company_name	company_id	emp_count_growth_change_percent...
goco	2908367	0.8645
centerwell senior primary care	76326347	0.3219
freshpoint, inc.	295504	0.1781
centerwell prime health	81986402	0.1778

Result 8

Action Output

Time	Action	Response	Duration / Fetch Time
9 16:37:13	with industry_skill_count as (select ci.industry as company_industry, jp.job_id, js.skill_abr from JobSkills js...	Error Code: 2013. Lost connection to MySQL server d...	30.004 sec
10 16:38:31	with industry_skill_count as (select ci.industry as company_industry, jp.job_id, js.skill_abr from JobSkills js...	217 row(s) returned	38.736 sec / 0.00006...
11 16:40:16	with max_min_emp_count_raw as (select company_id, # Casting varchar to timestamp in case data type...	5931 row(s) returned	0.198 sec / 0.303 sec

Query Completed

JMETER:

The screenshot shows the JMeter Test Plan interface. A 'JDBC Request' element is selected in the tree view. The configuration details are as follows:

- Name:** JDBC Request7
- Comments:** Which company has shown max employee count growth % over time? (headcount at latest ts - head count at earliest ts)/head count at earliest ts deduping and excluding null values
- Variable Name Bound to Pool:** db_variable
- SQL Query:** Set to "Select Statement". The query code is displayed below.

```

1  with max_min_emp_count_raw as (
2    select company_id,
3      # Casting varchar to timestamp in case data type is not already datetime
4      cast(time_recorded_ts as datetime) as date_time_stamp,
5      employee_count
6      from EmployeesCount
7      where time_recorded_ts is not null
8      and coalesce(employee_count,0)>0
9      group by 1,2,3
),
10 min_max_rank as (
11   select company_id, date_time_stamp, employee_count,
12   # Using row number since we want unique row for a rank, dense rank may show duplicates
13   # Creating 2 ranks - 1 for earliest and 1 for latest
14   row_number() over (partition by company_id order by date_time_stamp) as min_rank,
15   row_number() over (partition by company_id order by date_time_stamp desc) as max_rank
16   from max_min_emp_count_raw
17 ),
18
19

```

Query 8:

MYSQL:

The screenshot shows MySQL Workbench. A query is running in the background, and the results are displayed in a grid:

company_industry	jobs_count
Information Technology Services	364

The query history at the bottom shows the following actions:

Action	Time	Response	Duration / Fetch Time
with industry_skill_count as (select ci.industry as company_industry, jp.job_id, js.skill_abr from JobSkills js inner join CompanyIndustries ci on jp.company_id = ci.company_id where jp.closed_time is not null group by 1 order by jobs_count desc) s limit 1;	16:38:31	217 row(s) returned	38.736 sec / 0.00006...
with max_min_emp_count_raw as (select company_id, # Casting varchar to timestamp in case data type... 5931 row(s) returned	16:40:16		0.198 sec / 0.303 sec
select * from (select ci.industry as company_industry, count(distinct job_id) as jobs_count... 1 row(s) returned	16:41:03		0.084 sec / 0.000022...

JMETER:

JDBC Request

Name: JDBC Request8

Comments: Which job industry has the highest number of openings. Highest number of openings would mean the job_id that don't have a closed time yet

Variable Name Bound to Pool

Variable Name of Pool declared in JDBC Connection Configuration: db_variable

SQL Query

Query Type: Select Statement

```

1 select * from (
2     select ci.industry as company_industry,
3            count(distinct job_id) as jobs_count
4      from JobPosting jp
5      inner join CompanyIndustries ci
6        on jp.company_id = ci.company_id
7      where jp.closed_time is not null
8      group by 1
9      order by jobs_count desc
10 ) s
11 limit 1;
12

```

Query 9:

MYSQL:

Query 20 | Benefits

```

1 with required_benefits_jobs as (
2     select job_id
3         from Benefits
4     where lower(trim(type)) in ('401k', 'medical insurance', 'dental insurance')
5     group by 1
6 )
7 select count(distinct jp.job_id) as jobs_count
8     from JobPosting jp inner join Companies c
9       on jp.company_id = c.company_id
10      inner join required_benefits_jobs ben
11        on jp.job_id = ben.job_id
12      where lower(trim(c.state)) in ('ca', 'california');

```

Result Grid

jobs_count
375

Action Output

Time	Action	Response	Duration / Fetch Time
11 16:40:16	with max_min_emp_count_raw as (select company_id, # Casting varchar to timestamp in case data type... 5931 row(s) returned		0.198 sec / 0.303 sec
12 16:41:03	select * from (select ci.industry as company_industry, count(distinct job_id) as jobs_count... 1 row(s) returned		0.084 sec / 0.000022...
13 16:41:33	with required_benefits_jobs as (select job_id from Benefits where lower(trim(type)) in ('401k'... 1 row(s) returned		0.144 sec / 0.000019...

JMETER:

The screenshot shows the JMeter Test Plan interface. A 'JDBC Request' element is selected in the tree view. The configuration details are as follows:

- Name:** JDBC Request9
- Comments:** How many full time jobs offer medical, dental, and 401k as benefits and have HQ in California? Filtering for all jobs with the required benefits
- Variable Name Bound to Pool:** db_variable
- SQL Query:**
 - Query Type:** Select Statement
 - Query:**

```

1 with required_benefits_jobs as (
2     select job_id
3         from Benefits
4     where lower(trim(type)) in ('401k', 'medical insurance', 'dental insurance')
5         group by 1
6 )
7 select count(distinct jp.job_id) as jobs_count
8     from JobPosting jp inner join Companies c
9     on jp.company_id = c.company_id
10    inner join required_benefits_jobs ben
11    on jp.job_id = ben.job_id
12    where lower(trim(c.state)) in ('ca', 'california');

```

Query 10:

MYSQL:

The screenshot shows MySQL Workbench. A complex query is being run against the 'linkedinjobposting_lab1' schema. The query is:

```

with spec_count as (
    select industry, c.company_id, c.name as company_name,
           count(distinct speciality) as specialities_count
      from CompanySpecialties cs inner join Companies c
        on cs.company_id = c.company_id
       inner join CompanyIndustries ci
        on ci.company_id = c.company_id
       group by 1,2,3
),
spec_rank as (
    select sc.*,
           dense_rank() over (partition by industry order by specialities_count desc) as sp_rank
      from spec_count sc
)
select * from spec_rank
   where sp_rank=1
   order by specialities_count desc;

```

The results are displayed in a grid:

industry	company_id	company_name	specialities_co...	sp_rank
Staffing_Recruiting	226965	net2source inc.	78	1
Information Technology Services	15423341	amplitek	77	1
Market Research	327412	resonate	50	1
Transportation_Trucking_Railroad	142916	ci logistics america	48	1

Result 11

Action Output

Time	Action	Response	Duration / Fetch Time
12 16:41:03	select * from (select ci.industry as company_industry, count(distinct job_id) as jobs_count...)	1 row(s) returned	0.084 sec / 0.000022...
13 16:41:33	with required_benefits_jobs as (select job_id from Benefits where lower(trim(type)) in ('401k'...)	1 row(s) returned	0.144 sec / 0.000019...
14 16:42:11	with spec_count as (select industry, c.company_id, c.name as company_name, count(distinct specialit...)	146 row(s) returned	1.050 sec / 0.00013 s...

Query Completed

JMETER:

The screenshot shows the JMeter Test Plan interface. A 'JDBC Request' element is selected in the tree view under a 'Thread Group'. The configuration panel displays the following details:

- Name:** JDBC Request10
- Comments:** Which companies have the highest number of specialties by industry?
- Variable Name Bound to Pool:** db_variable
- SQL Query:**
 - Query Type:** Select Statement
 - Query:**

```

1 with spec_count as (
2     select industry, c.company_id, c.name as company_name,
3            count(distinct speciality) as specialities_count
4     from CompanySpecialties cs inner join Companies c
5     on cs.company_id = c.company_id
6     inner join CompanyIndustries ci
7     on ci.company_id = c.company_id
8     group by 1,2,3
9 ),
10 spec_rank as (
11     select sc.*,
12            dense_rank() over (partition by industry order by specialities_count desc) as sp_rank
13     from spec_count sc
14 )
15 select * from spec_rank
16   where sp_rank=1
17   order by specialities_count desc;

```

Query 11:

MYSQL:

The screenshot shows the MySQL Workbench interface. A query is being run against the 'linkedinjobposting_lab1' schema:

```

1 select c.name as company_name, c.company_size, jp.description as job_description,
2        count(distinct job_id) as jobs_count
3     from JobPosting jp
4     inner join Companies c
5     on jp.company_id = c.company_id
6    where jp.closed_time is not null
7    and lower(trim(jp.work_type)) = 'internship'
8   group by 1,2,3
9   order by jobs_count desc;

```

The results grid displays the following data:

company_name	company_size	job_description	jobs_count
hcvt	4	Focus Clarity Commitment HCVT is a highly sp...	2
355code	0	355Codehttp://www.355code.com/Digital Marketing...	1
902 youth	1	Executive Assistant This position is unpaid and ...	1
avocademy (yc w22)	1	Fill out this application and receive an update wi...	1
eventeny	1	Overview We are seeking a highly motivated an...	1
fulwell 73 productions	2	We are seeking smart motivated interns for the...	1
hcvt	4	Focus Clarity Commitment HCVT is a highly sp...	1
openpark	1	OpenParkus a pet wellness and technology star...	1
power digital marketing	4	About the CompanyPower Digital is a leading d...	1
student world impact film festival (swift)	4	Location Remote Duration 5 months October 10...	1

Action Output:

Action	Time	Response	Duration / Fetch Time
with spec_count as (select industry, c.company_id, c.name as company_name, count(distinct speciality) as specialities_count)	16:42:11	146 row(s) returned	1.050 sec / 0.00013 s...
select c.name as company_name, c.company_size, jp.description as job_description, count(distinct job_id) as jobs_count	16:43:15	10 row(s) returned	0.187 sec / 0.00016 s...
select c.name as company_name, c.company_size, jp.description as job_description, count(distinct job_id) as jobs_count	16:43:26	10 row(s) returned	0.159 sec / 0.00016 s...

JMETER:

Test Plan

JDBC Connection Configuration

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

```
1 select c.name as company_name, c.company_size, jp.description as job_description,
2       count(distinct jp.job_id) as jobs_count
3     from JobPosting jp
4   inner join Companies c
5     on jp.company_id = c.company_id
6   where jp.closed_time is not null
7     and lower(trim(jp.work_type)) = 'internship'
8   group by 1,2,3
9   order by jobs_count desc;
```

Test Plan

JDBC

Thread

View Results in Table

Name: View Results in Table

Comments: Calculate average maximum and minimum salaries for Nonprofit Organization Management companies while preserving the details of each company within the state

Write results to file / Read from file

Filename [] Browse... Log/Display Only: Errors Successes 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	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	✓

Test Plan

JDBC

Thread

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename [] Browse... Log/Display Only: Errors Successes 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	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_ews_instance

Administration Schemas

SCHEMAS

linkedInJobposting_lab1

- Tables
 - Benefits
 - Companies
 - CompanyIndustries
 - CompanySpecialties
 - EmployeesCount
 - JobIndustries
 - JobPosting
 - JobSkills
- Views
- Stored Procedures
 - Functions
- sys

Object Info Session Schema: linkedInJobposting_lab1

Query 20 Benefits

```

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

```

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

Action Output

Action	Time	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		0 row(s) affected	0.098 sec
25 17:22:33 call GetSkillsAbrByCompany(1016)		6 row(s) returned	0.138 sec / 0.000022...

Read Only

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);

```

The screenshot shows a JMeter interface with the following elements:

- Search Bar:** At the top, there is a search input field, a "Case sensitive" checkbox, a "Regular exp." checkbox, a "Search" button, and a "Reset" button.
- Response Tabs:** Below the search bar, there are tabs for "Sampler result", "Request", "Response data" (which is selected), "Response Body", and "Response headers".
- Results Table:** A table below the tabs displays the following data:

16:48:58.007	Thread Group 1-1	JDBC Request12	1895	1644	41	0	1895	12

Procedure 2:

MYSQL:

The screenshot shows the MySQL Workbench interface. The left sidebar displays the schema structure for 'linkedinjobposting_lab1'. The main area contains a query editor with the following SQL code:

```

4   CREATE PROCEDURE UpdateJobPostingSalary1(IN p_job_ID BIGINT, newSalary DECIMAL(10,2))
5   BEGIN
6       UPDATE JobPosting
7           SET max_salary = newSalary
8           WHERE job_id = p_job_ID;
9   END $$

10  DELIMITER ;
11
12
13
14  SET SQL_SAFE_UPDATES = 0;
15  call UpdateJobPostingSalary1(133114754, 80000.00);
16
17  select * from JobPosting where job_id=133114754;
18

```

The result grid shows one row of data for a job posting with ID 133114754. The columns include job_id, company_id, title, description, max_salary, med_salary, min_salary, pay_period, formatted_work_ty..., location, applies, remote_allowed, views, and job... . The description field contains the text: "Are you a dynamic and creative marketing prof...".

The action output table lists the execution steps with their times, actions, responses, and durations:

Time	Action	Response	Duration / Fetch Time
✓ 27 17:23:42	CREATE PROCEDURE UpdateJobPostingSalary1(IN p_job_ID BIGINT, newSalary DECIMAL(10,2)) BEGIN UPDATE JobPosting SET max_salary = newSalary WHERE job_id = p_job_ID; END \$\$	0 row(s) affected	0.00 sec
✓ 28 17:23:42	SET SQL_SAFE_UPDATES = 0;	0 row(s) affected	0.071 sec
✗ 29 17:23:43	call UpdateJobPostingSalary1(133114754, 80000.00)	Error Code: 1146. Table 'linkedinjobposting_lab1.job...' doesn't exist	0.229 sec
✓ 30 17:23:52	drop PROCEDURE UpdateJobPostingSalary1	0 row(s) affected	0.092 sec
✗ 31 17:24:06	drop PROCEDURE UpdateJobPostingSalary1	Error Code: 1305. PROCEDURE linkedinjobposting_lab1.UpdateJobPostingSalary1 doesn't exist	0.065 sec
✓ 32 17:24:12	CREATE PROCEDURE UpdateJobPostingSalary1(IN p_job_ID BIGINT, newSalary DECIMAL(10,2)) BEGIN UPDATE JobPosting SET max_salary = newSalary WHERE job_id = p_job_ID; END \$\$	0 row(s) affected	0.083 sec
✓ 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 / 0.00052...

Query Completed

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:

Administration Schemas

SCHEMAS

linkedInJobposting_lab1

- Tables
 - Benefits
 - Companies
 - CompanyIndustries
 - CompanySpecialties
 - EmployeesCount
 - JobIndustries
 - JobPosting
 - JobSkills
- Views
- Stored Procedures
- Functions

Schemas: linkedInJobposting_lab1

Object Info Session

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows: Result Grid

```

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

Query Completed

Trigger 2:

Administration Schemas

SCHEMAS

linkedInJobposting_lab1

- Tables
 - Benefits
 - Companies
 - CompanyIndustries
 - CompanySpecialties
 - EmployeesCount
 - JobIndustries
 - JobPosting
 - JobSkills
- Views
- Stored Procedures
- Functions

Schemas: linkedInJobposting_lab1

Object Info Session

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows: Result Grid

```

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
35 17:25:25	select * from JobPosting where max_salary=00000.00 LIMIT 0, 1000	100 row(s) returned	0.220 sec / 0.220 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... Error Code: 1146. Table 'linkedinjobposting_lab1.Jobp...' 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:

The screenshot shows the MySQL Workbench interface with the schema `linkedinjobposting_lab1` selected. In the central query editor, the following SQL code is displayed:

```
1 • drop trigger AfterSkillsUpdate;
2
3 • DELIMITER //
4 • CREATE TRIGGER AfterSkillsUpdate
5 AFTER UPDATE ON JobSkills
6 FOR EACH ROW
7 BEGIN
8
9     INSERT INTO NotificationLog (message, recipient, sent_at)
10    VALUES ('Skills for job ' || NEW.job_id || ' have been updated.', 'HR Department', NOW());
11 END;
// 
DELIMITER ;
```

The Action Output pane at the bottom shows the execution history with the following log entries:

Time	Action	Response	Duration / Fetch Time
35 17:32:10	CREATE TRIGGER BeforeJobPostingUpdate BEFORE UPDATE ON Jobposting FOR EACH ROW BEGIN IF NEW.expiry... Error Code: 1146. Table 'linkedinjobposting_lab1.Jobp...' UU6 / sec	0 row(s) affected	0.099 sec
39 17:32:33	drop trigger BeforeJobPostingUpdate	0 row(s) affected	0.066 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...' 0.119 sec	0 row(s) affected	0.119 sec
41 17:33:01	CREATE TRIGGER BeforeJobPostingUpdate BEFORE UPDATE ON Jobposting FOR EACH ROW BEGIN IF NEW.expiry... 0 row(s) affected	1000 row(s) returned	0.148 sec / 8.528 sec
42 17:33:56	select * from JobPosting LIMIT 0, 1000	0 row(s) affected	0.067 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...' 0.146 sec	0 row(s) affected	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 row(s) affected	0 row(s) affected	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 row(s) affected	0.082 sec
47 17:38:04	drop trigger AfterSkillsUpdate	0 row(s) affected	0.093 sec
48 17:38:07	CREATE TRIGGER AfterSkillsUpdate AFTER UPDATE ON JobSkills FOR EACH ROW BEGIN INSERT INTO Notificati... 0 row(s) affected	0 row(s) affected	0.080 sec

Query Completed

Trigger 4:

lab1_ews_instance

Administration Schemas

SCHEMAS

linkedInJobposting_lab1

- Tables
 - Benefits
 - Companies
 - CompanyIndustries
 - CompanySpecialties
 - EmployeesCount
 - JobIndustries
 - JobPosting
 - JobSkills
 - Views
- Stored Procedures
- Functions
- sys

Object Info Session

Schema: linkedInJobposting_lab1

Query 20 Benefits

DELIMITER //

```

1 drop trigger AfterSkillsForRemoteWorkingInsert;
2
3 DELIMITER //
4 CREATE TRIGGER AfterSkillsForRemoteWorkingInsert
5 AFTER INSERT ON JobSkills
6 FOR EACH ROW
7 BEGIN
8   IF NEW.skill_abr = 'RemoteWork' THEN
9     INSERT INTO NotificationLog (message, recipient, sent_at)
10    VALUES ('Remote work skills added for job' OR NEW.job_id, 'Remote Work Department', NOW());
11   END IF;
12 END;
13 //
14 DELIMITER ;
15

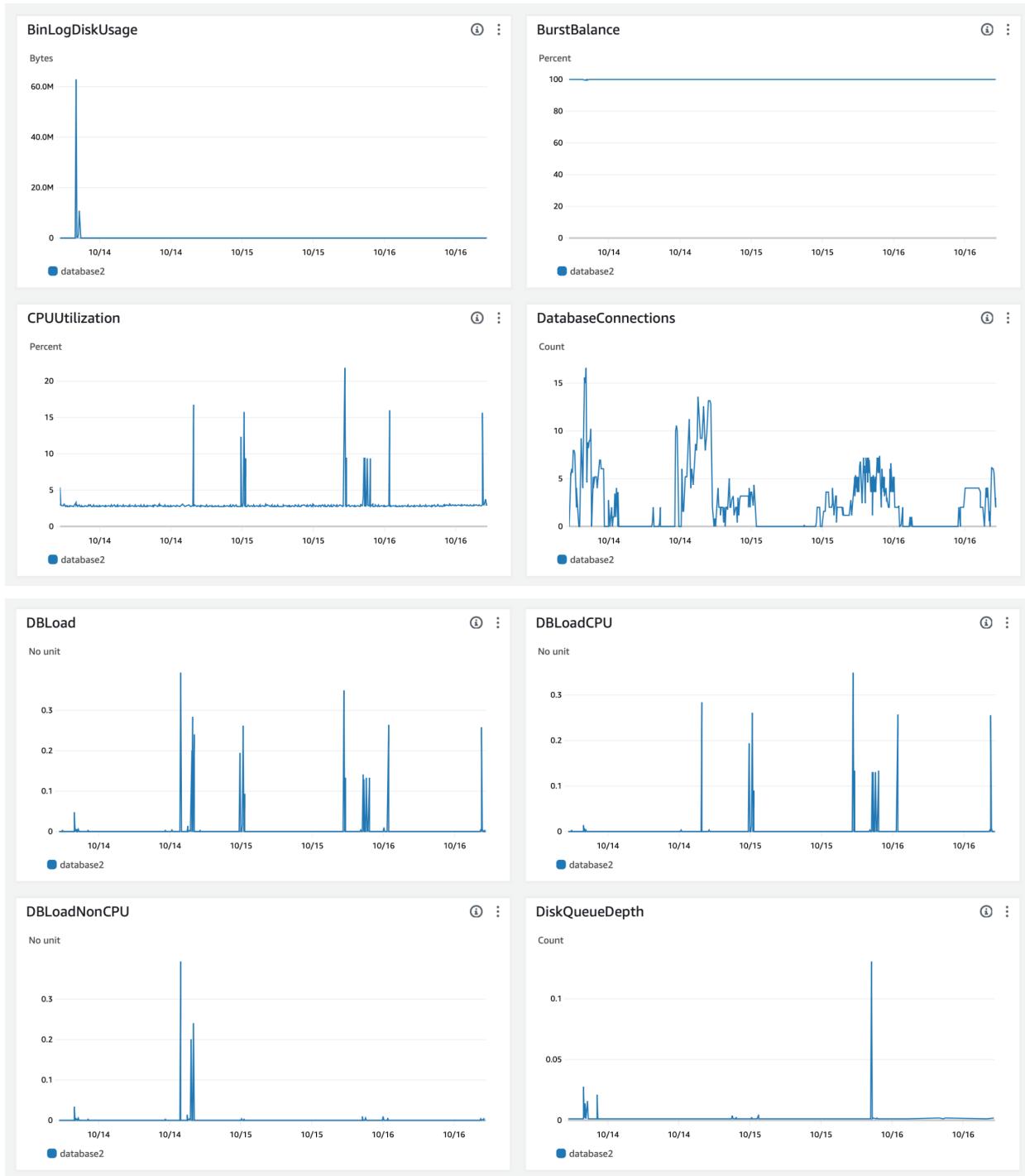
```

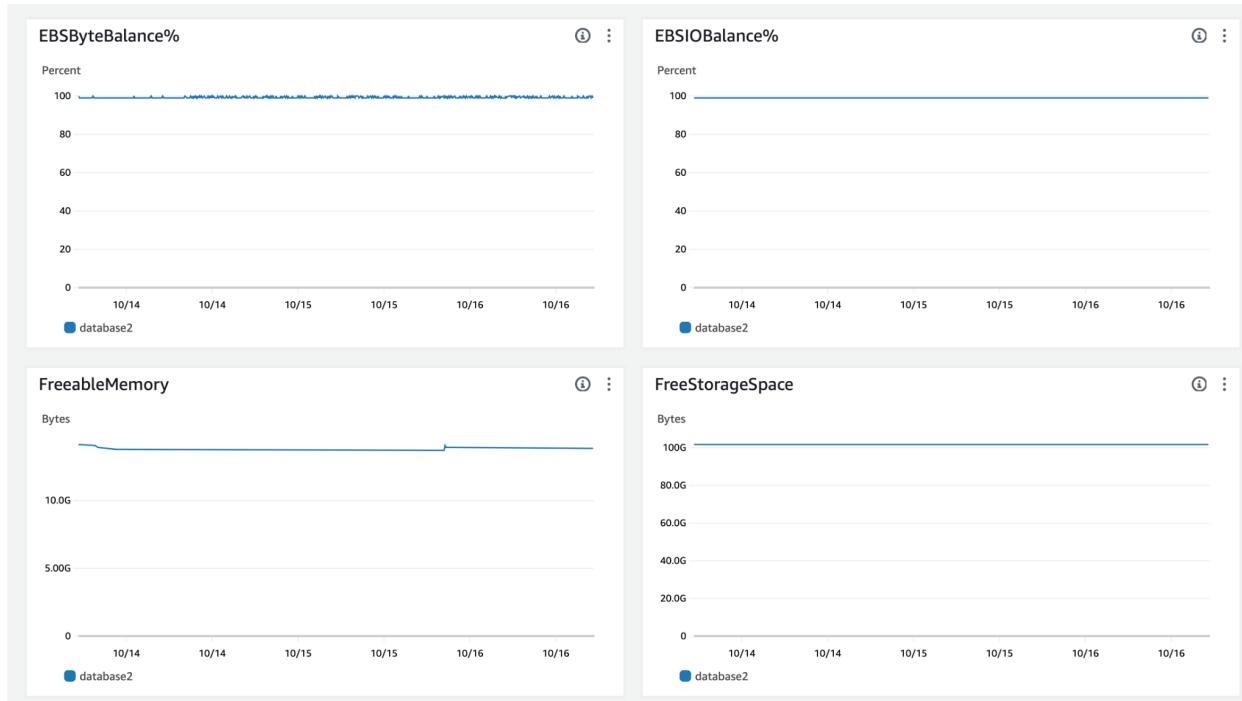
Action Output

Time	Action	Response	Duration / Fetch Time
40	CREATE TRIGGER BeforeJobPostingUpdate BEFORE UPDATE ON Jobposting FOR EACH ROW BEGIN IF NEW.expiry... error Code: 1146. Table 'linkedInJobposting_lab1.Jobposting' doesn't have a primary key or auto-increment column.	0 row(s) affected	0.000 sec
41	CREATE TRIGGER BeforeJobPostingUpdate BEFORE UPDATE ON Jobposting FOR EACH ROW BEGIN IF NEW.expiry... 0 row(s) affected	0 row(s) affected	0.119 sec
42	select * from JobPosting LIMIT 0, 1000	1000 row(s) returned	0.148 sec / 8.528 sec
43	CREATE TRIGGER AfterJobPostingInsert AFTER INSERT ON Jobposting FOR EACH ROW BEGIN INSERT INTO JobP... Error Code: 1146. Table 'linkedInJobposting_lab1.Jobposting' doesn't have a primary key or auto-increment column.	0 row(s) affected	0.067 sec
44	CREATE TRIGGER AfterJobPostingInsert AFTER INSERT ON Jobposting FOR EACH ROW BEGIN INSERT INTO JobP... Error Code: 1359. Trigger already exists	0 row(s) affected	0.067 sec
45	drop trigger AfterJobPostingInsert	0 row(s) affected	0.146 sec
46	CREATE TRIGGER AfterJobPostingInsert AFTER INSERT ON Jobposting FOR EACH ROW BEGIN INSERT INTO JobP... 0 row(s) affected	0 row(s) affected	0.082 sec
47	drop trigger AfterSkillsUpdate	0 row(s) affected	0.093 sec
48	CREATE TRIGGER AfterSkillsUpdate AFTER UPDATE ON JobSkills FOR EACH ROW BEGIN INSERT INTO Notificati... 0 row(s) affected	0 row(s) affected	0.080 sec
49	drop trigger AfterSkillsForRemoteWorkingInsert	0 row(s) affected	0.091 sec
50	CREATE TRIGGER AfterSkillsForRemoteWorkingInsert AFTER INSERT ON JobSkills FOR EACH ROW BEGIN IF NEW... 0 row(s) affected	0 row(s) affected	0.222 sec

Query Completed

AWS Performance Measurements:





Logs from AWS cloud watch

