Relational Databases with MySQL Week 8 Coding Assignment

Points possible: 70

Tyler Livermore

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: Using a text editor of your choice, write the queries that accomplishes the objectives listed below. Take screenshots of the queries and results and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document to the repository. Additionally, push an .sql file with all your queries to the same repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

Write queries to address the following business needs.

1. I want to know how many employees with each title were born after 1965-01-01.

```
SELECT count(*) AS number_hired, t.title

FROM employees e

INNER JOIN titles t USING (emp_no)

WHERE e.hire_date > ‘1965-01-01’

GROUP BY t.title;
```

2. I want to know the average salary per title.

```
SELECT round(avg(s.salary), 2) AS ‘Average Salary’, t.title

FROM salaries s
```

JOIN titles t USING (emp_no)

GROUP BY t.title;

3. How much money was spent on salary for the marketing department between the years 1990 and 1992?

SELECT sum(s.salary) AS 'Total Salary', d.dept_name

FROM salaries s

JOIN dept_emp de USING (emp_no)

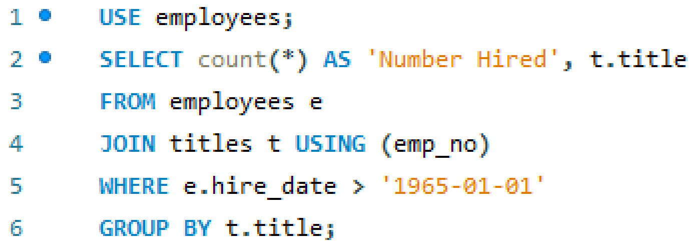
JOIN departments d USING (dept_no)

WHERE d.dept_name = 'Marketing'

AND s.from_date BETWEEN '1990-01-01' and '1992-12-31';

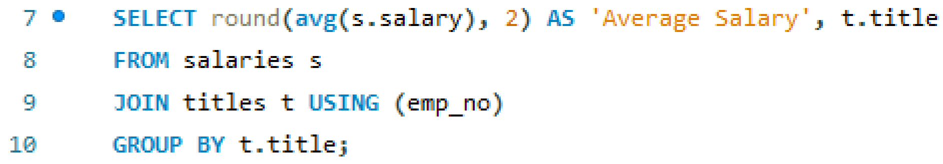
Screenshots of Queries:

1.



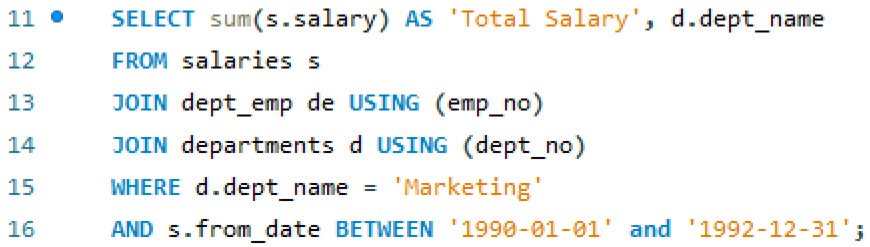
```
USE employees;
SELECT count(*) AS 'Number Hired', t.title
FROM employees e
JOIN titles t USING (emp_no)
WHERE e.hire_date > '1965-01-01'
GROUP BY t.title;
```

2.



```
SELECT round(avg(s.salary), 2) AS 'Average Salary', t.title
FROM salaries s
JOIN titles t USING (emp_no)
GROUP BY t.title;
```

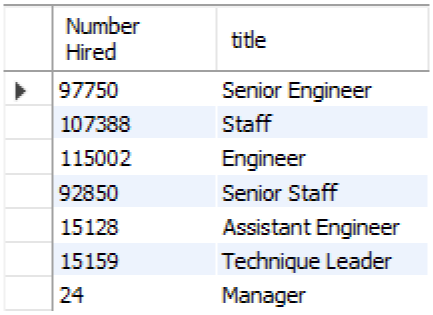
3.



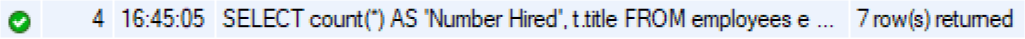
```
SELECT sum(s.salary) AS 'Total Salary', d.dept_name
FROM salaries s
JOIN dept_emp de USING (emp_no)
JOIN departments d USING (dept_no)
WHERE d.dept_name = 'Marketing'
AND s.from_date BETWEEN '1990-01-01' and '1992-12-31';
```

Screenshots of Query Results (only include the last 20 rows):

1.

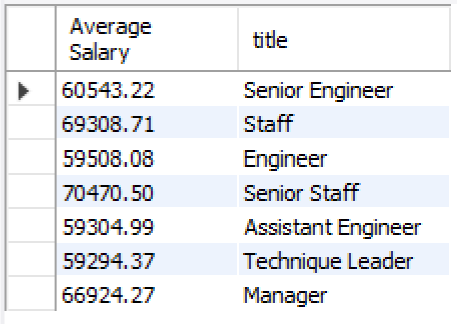


Number Hired	title
97750	Senior Engineer
107388	Staff
115002	Engineer
92850	Senior Staff
15128	Assistant Engineer
15159	Technique Leader
24	Manager

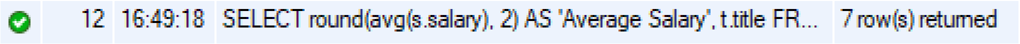


```
✓  4  16:45:05  SELECT count(*) AS 'Number Hired', t.title FROM employees e ...  7 row(s) returned
```

2.

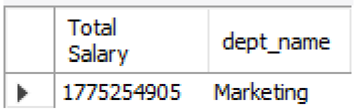


Average Salary	title
60543.22	Senior Engineer
69308.71	Staff
59508.08	Engineer
70470.50	Senior Staff
59304.99	Assistant Engineer
59294.37	Technique Leader
66924.27	Manager

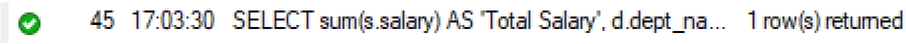


```
✓  12  16:49:18  SELECT round(avg(s.salary), 2) AS 'Average Salary', t.title FR...  7 row(s) returned
```

3.

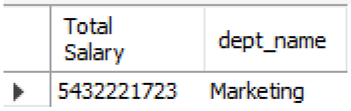


Total Salary	dept_name
1775254905	Marketing



```
✓  45  17:03:30  SELECT sum(s.salary) AS 'Total Salary', d.dept_na...  1 row(s) returned
```

3 - Testing that different date range yields different sum to confirm that the query is applying the date range and is not just summing all rows (changed 1992 to 1996):



Total Salary	dept_name
5432221723	Marketing

Sum is larger as expected.

URL to GitHub Repository:

https://github.com/tylerjlivermore/Week8_MySQL_RelationshipsAndFunctions