

Relational Databases with MySQL Week 8 Coding Assignment

Points possible: 70

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.
2. I want to know the average salary per title.
3. How much money was spent on salary for the marketing department between the years 1990 and 1992?

Screenshots of Queries:

1.

```
1 • desc titles;
2 • desc employees;
3
4 • SELECT t.title as "Job Title", count(e.emp_no) as "Number of Employees"
5   FROM titles t
6  INNER JOIN employees e ON t.emp_no = e.emp_no
7  WHERE e.birth_date > '1965-01-01'
8  GROUP BY t.title;
```

2.

```
      desc salaries;
12 • desc titles;
13
14 • SELECT titles.title as "Job Title", avg(salary) as "Average Salary"
15   FROM titles
16  INNER JOIN salaries on salaries.emp_no = titles.emp_no
17  GROUP BY titles.title;
18
```

3.

```
21 • SELECT *
22   FROM departments
23  GROUP BY dept_name
24  LIMIT 20;
25
26 • SELECT d.dept_name as "Department", count(s.salary) as "Total Salary"
27   FROM salaries s
28  INNER JOIN dept_emp de on de.emp_no = s.emp_no
29  INNER JOIN departments d on de.dept_no = d.dept_no
30  WHERE d.dept_no = 'd001' AND s.from_date >= '1990-01-01' AND s.to_date <= '1992-12-31'
31  GROUP BY d.dept_name;
```

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

1.

```
1 • desc titles;
2 • desc employees;
3
4 • SELECT t.title as "Job Title", count(e.emp_no) as "Number of Employees"
5 FROM titles t
6 INNER JOIN employees e ON t.emp_no = e.emp_no
7 WHERE e.birth_date > '1965-01-01'
8 GROUP BY t.title;
```

100% 18:8

Result Grid Filter Rows: Search Export:




	Job Title	Number of Employee...
►	Senior Staff	612
▢	Staff	703
▢	Technique Leader	95
▢	Senior Engineer	589
▢	Engineer	657
▢	Assistant Engineer	97

2.

Result Grid Filter Rows: Search Export:

	Job Title	Average Salary
►	Senior Engineer	60543.2191
▢	Staff	69309.1023
▢	Engineer	59508.0397
▢	Senior Staff	70470.8353
▢	Assistant Engineer	59304.9863
▢	Technique Leader	59294.3742
▢	Manager	66924.2706

3.

Result Grid   Filter Rows: <input type="text" value="Search"/> Export: 			
	Department	Total Salary	
▶	Marketing	16484	

URL to GitHub Repository:

<https://github.com/AaronL1981/Java-Week-08.git>