Relational Databases with MySQL Week 4 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, with your Java project code, to the 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 5 stored procedures for the employees database.

Write a description of what each stored procedure does and how to use it.

Procedures should use constructs you learned about from your research assignment and be more than just queries.

**Screenshots:**

This procedure checks for the amount of salaries above a certain income, then assigns that count to the out parameter.



This procedure selects the count of salaries within 10000 of the IN parameter, ala a salary tier. Entering 'call employees\_salary\_tier\_count(60000);' will count the amount of people that make a salary between 60,000 and 70,000 and query it out.



This procedure takes an emp\_no as an IN and sets the length(first\_name) to the OUT. The IF statement deterimines how difficult the name is to say based on the length of the name. After the IF statement does its thing, the procedure then queries the OUT of the difficulty. Did you know theres a guy in here whos first and last name equals 29 total letters? That's Expert difficulty right there.



This procedure takes a department name as an IN and queries what kind of shipping one can expect from said department.



This beautiful procedure taught me some awesome lessons. But, what it does: It takes the emp\_no as an IN, finds the associated first\_name from employees, assigns that value to the declared varchar 'name', then puts it through a case statement. If the name begins with 'a', the procedure assigns 'Archer' to the class variable. At the end I selected the name and class variables to show this connection. If I had more time, I'd create classes for every letter, giving any emp\_no you can think of a class from Dungeons and Dragons.

Some lessons i learned: You NEED a variable to take part in the case statement, wether that be an OUT parameter, INOUT parameter, or something you declare right after the 'begin' part. if you don't, well, IT DON'T. Also, that variable NEEDS a value, so selecting the column into the variable going in the case is a great move.

**URL to GitHub Repository:** [**https://github.com/aspear91/Week10.git**](https://github.com/aspear91/Week10.git)