# 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:**

**delimiter //**

**-- replace employee first name**

**create procedure replaceEmployee(in employeeNumber int, in firstName varchar(30), in LastName varchar(30), in newFirstName varchar(30))**

**begin**

**update employees**

**set first\_name = newFirstName**

**where first\_name = firstName**

**and last\_name = lastName;**

**-- display updated rows**

**end //**

**delimiter ;**

**delimiter //**

**-- Get the total salary and place it in a variable**

**create procedure getTotalSalary()**

**begin**

**declare totalSalary bigint default 0;**

**select sum(salary)**

**into totalSalary**

**from salaries;**

**select totalSalary;**

**end//**

**delimiter ;**

**delimiter //**

**delimiter //**

**-- Determine if an employee gets high medium or low pay**

**create procedure getPayLevel()**

**begin**

**select s.salary,**

**case**

**when s.salary > 80000 then "High Salary"**

**when s.salary > 50000 and s.salary < 80000 then "Medium Salary"**

**else "Low Salary"**

**end as 'Pay level'**

**from salaries s**

**inner join employees e**

**on s.emp\_no = e.emp\_no;**

**end//**

**delimiter ;**

**delimiter //**

**-- Repeat employee name**

**create procedure repeatName(in empNum int, in repeatNum int)**

**begin**

**select repeat(first\_name, repeatNum)**

**from employees**

**where emp\_no = empNum;**

**end//**

**delimiter ;**

**delimiter //**

**-- return an employees full name**

**create procedure fullName(in empNum int, out fullName varchar(50))**

**begin**

**select concat(first\_name, ' ' , last\_name)**

**into fullName**

**from employees**

**where emp\_no = empNum;**

**end//**

**delimiter ;**

**call replaceEmployee(10001, 'Georgi', 'Facello', 'Tim');**

**call getTotalSalary();**

**call getPayLevel();**

**call repeatName(10001, 4);**

**call fullName(10001, @out\_value);**

**select @out\_value;**

**URL to GitHub Repository:**

[**https://github.com/jvgreen/Promineo-Boot-Camp/blob/master/mysql/week4/week4.sql**](https://github.com/jvgreen/Promineo-Boot-Camp/blob/master/mysql/week4/week4.sql)