# 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.

**#1 -- THIS CREATES A PROCEDURE THAT TRACKS ALL OF THE MALE EMPLOYEES**

**DELIMITER //**

**CREATE PROCEDURE gender\_male()**

**BEGIN**

**SELECT \* FROM employees**

**WHERE gender = 'M';**

**END//**

**CALL gender\_male();**

**#2 -- THIS CREATES A PROCEDURE THAT GIVES ME ALL OF THE MALE**

**drop procedure if exists gender\_male;**

**EMPLOYEES WHOS BIRTHDAY IS IN APRIL 1ST 1987**

**DELIMITER //**

**CREATE PROCEDURE gender\_male\_birthday ()**

**BEGIN**

**SET @birth := "1987-04-01";**

**SELECT \* FROM employees**

**WHERE gender ='M' AND birth\_date = @birthDate;**

**END //**

**CALL gender\_male\_birthday();**

**#3 -- THIS PROCEDURE CREATES A MAXIMUM SALARY FOR A EMPLOYEE BY THE TITLE**

**DROP PROCEDURE IF EXISTS maxSalaryOfEmployeeByTitle;**

**CREATE PROCEDURE maxSalaryOfEmployeeByTitle (**

**IN employeeeTitles VARCHAR(321),**

**OUT maximumSalary INT**

**)**

**BEGIN**

**SELECT max(s.salary)**

**INTO maximumSalary**

**FROM employees e**

**INNER JOIN salaries s USING (emp\_no)**

**INNER JOIN titles t USING (emp\_no)**

**GROUP BY t.titles**

**HAVING t.titles = employeeTitles;**

**END//**

**CALL maxSalaryOfEmployeeByTitle('Manager', @maximumSalary);**

**#4 --THIS PROCEDURE LETS YOU GET THE EMPLOYEES FIRST NAME LAST NAME AND TITLE**

**DROP PROCEDURE IF EXISTS getEmployeeInfoByJobTitle;**

**DELIMITER //**

**CREATE PROCEDURE getEmployeeInfoByJobTitle(**

**IN empInfoByTitle VARCHAR(321)**

**)**

**BEGIN**

**SELECT e.first\_name, e.last\_name, t.title**

**FROM employees e**

**INNER JOIN titles t USING (emp\_no)**

**WHERE t.title = empInfoByTitle;**

**END//**

**CALL getEmployeeInfoByJobTitle("Engineer");**

**#5 -- THIS WILL PRODUCE ALL OF THE INFORMATION FOR THE EMPLOYEE INCLUDING A HIRE DATE AND EMPLOYEE NUMBER**

**DROP PROCEDURE IF EXISTS basicEmpInfo;**

**DELIMITER //**

**CREATE PROCEDURE basicEmpInfo ()**

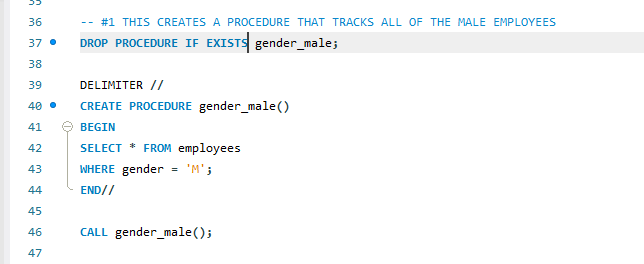
**BEGIN**

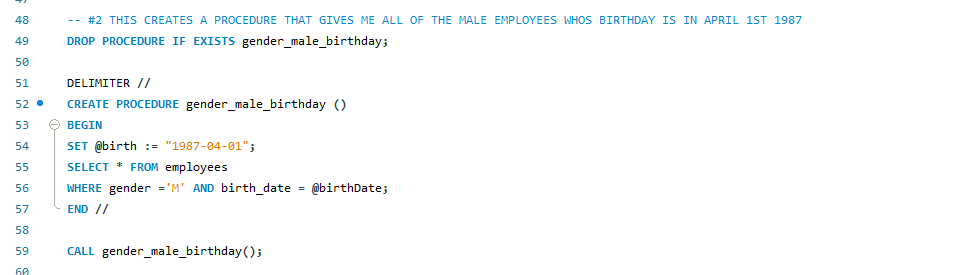
**SELECT emp\_no,first\_name,last\_name,gender,hire\_date,birth\_date FROM employees;**

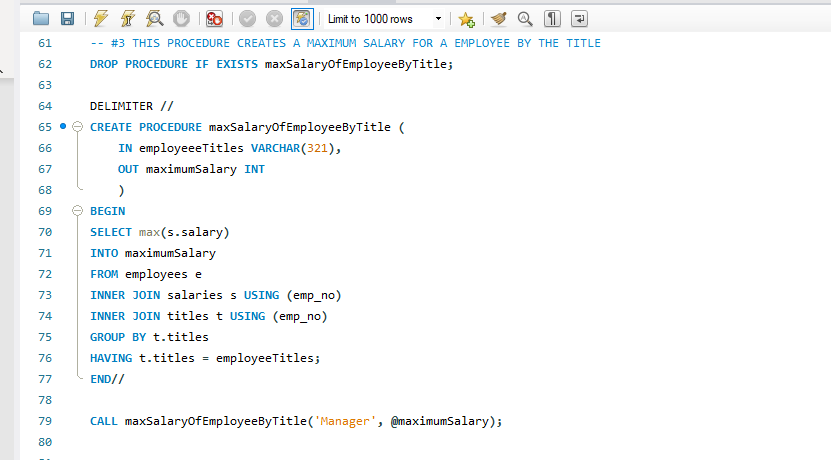
**END //**

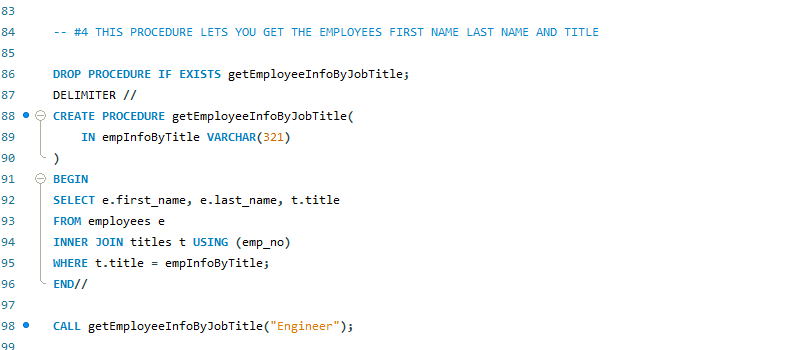
**CALL basicEmpInfo();**

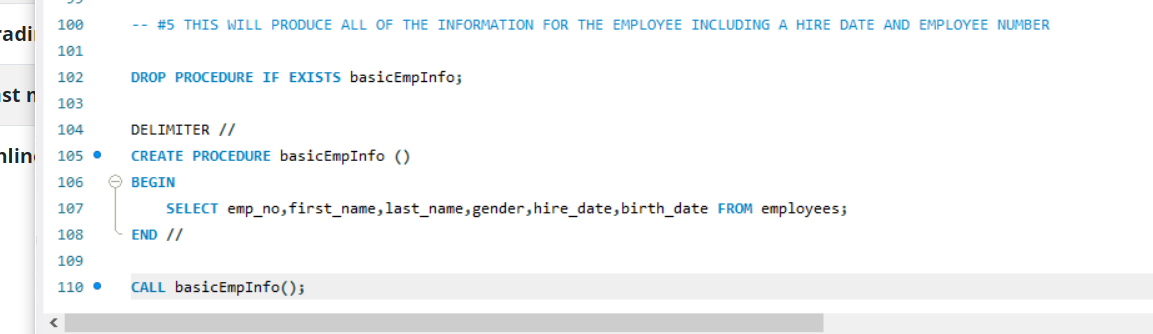
**Screenshots:**











**URL to GitHub Repository:** [**https://github.com/jchernandez2123/week4MysqlCodingAssignment**](https://github.com/jchernandez2123/week4MysqlCodingAssignment)