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.

**Be sure to execute the Create\_Stored\_Procedure.sql first and then execute the Run\_Stored\_Procedure.sql to run each stored procedure.**

**Description for each stored procedure**:

**Stored Procedure 1**:

This stored procedure takes a row limit number, it takes in a total\_salary variable in modifies it with a new row limited value, how many names start with the letter j variable (which gets changed in the body of the stored proc and returned with the new value.

How to use:

1st parameter: pass in a number for row\_limiter parameter

2nd parameter: pass in a variable @emp\_first\_name\_j

3rd parameter: pass in a variable @total\_salary\_row\_limited

For example, call the stored procedure as so:

Call employees.stored\_proc1(100, @emp\_first\_name\_j, @total\_salary\_row\_limited);

You can then show variables’ output to the screen as so:

select @emp\_first\_name\_j,

@total\_salary\_row\_limited;

**Stored Procedure 2**:

The stored procedure creates a table, then inserts to the table which employees make more than 90000.

It gets the total number of rows of that table. Then does a loop through the number and inserts into the who\_made\_90000\_plus a detailed description.

of if an employee makes more than or equal to 90000 or more than or equal to 100000 dollars in salary.

How to use: simply call the stored procedure without parameter,

For example, call the stored procedure as so:

call employees.stored\_proc2();

**Stored Procedure 3**:

This stored procedure is taken in an employee number, birth date, first name, last name, gender letter, and hire date.

It uses that info and inserts it into the employees table. Then shows the inserted data to the user.

How to use:

1st parameter: pass in any number less than 10000 to ensure it is not taken.

2nd parameter: pass in a date in this format YYYY-MM-DD

3rd parameter: pass in a string first name

4th parameter: pass in a string last name

5th parameter: pass in a enum char (M, F)

6th parameter: pass in a date in this format YYYY-MM-DD

For example, call the stored procedure as so:

call employees.stored\_proc3(1, '2002-01-01', 'Billy', 'Bob', 'M', date(now()));

**Stored Procedure 4**:

The stored procedure takes a manager\_no and then compares manager's salary to all of their employees and sees who makes the most. It then tells the user who makes the most with a select statement.

How to use: pass in an emp\_no of any manager from the dept\_manager table.

For example, call the stored procedure as so:

call employees.stored\_proc4(111692);

**Stored Procedure 5**:

This stored procedure takes in a number of terms that the series will have and then will create the series and will finally list it to the user.

How to use: pass in a number that represents how many terms you want in the series. Keep in mind that the store procedure can only output up to 254 chars, so please use a reasonable number.

For example, call the stored procedure as so:

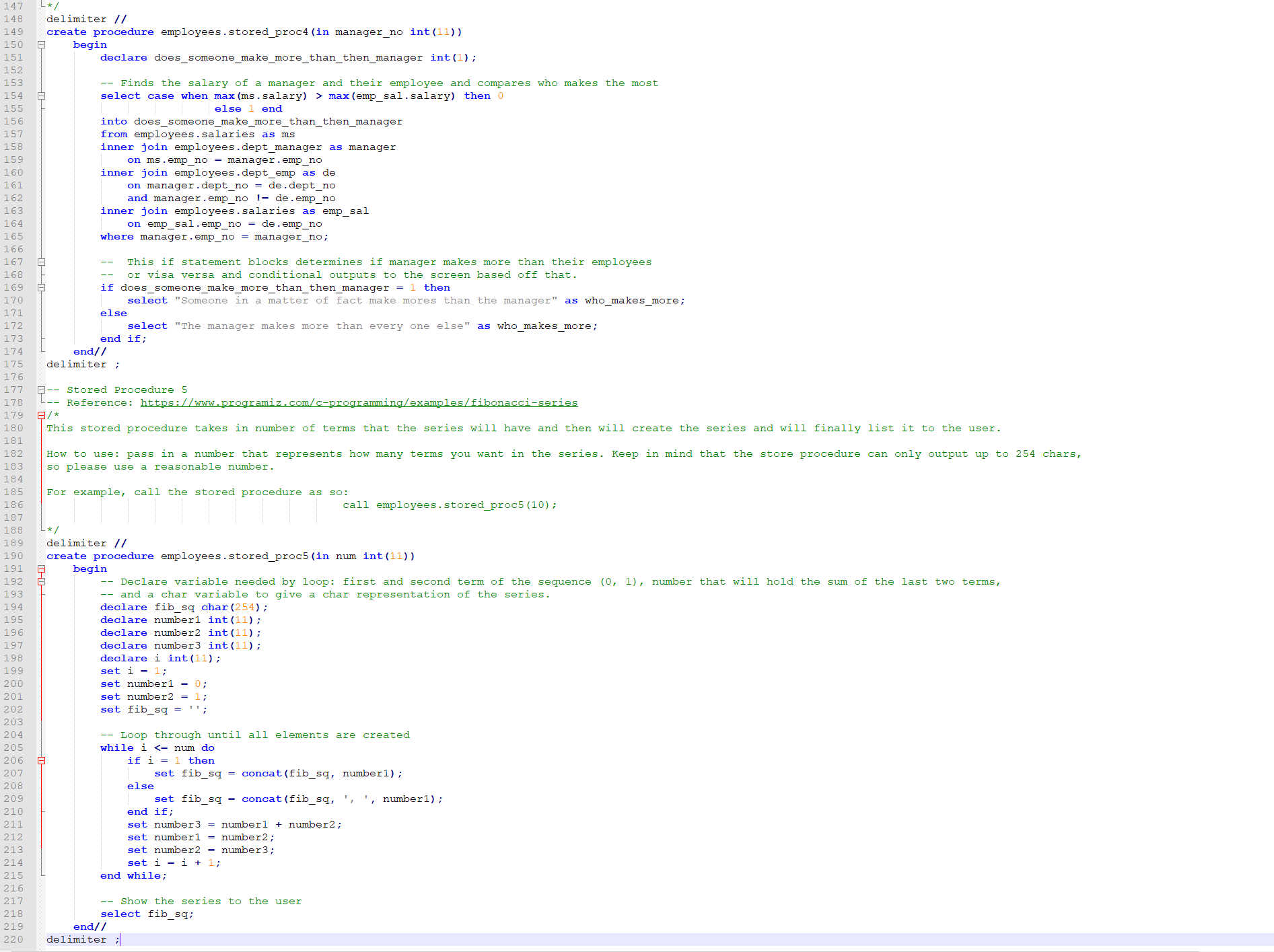
call employees.stored\_proc5(10);

**Screenshots:**

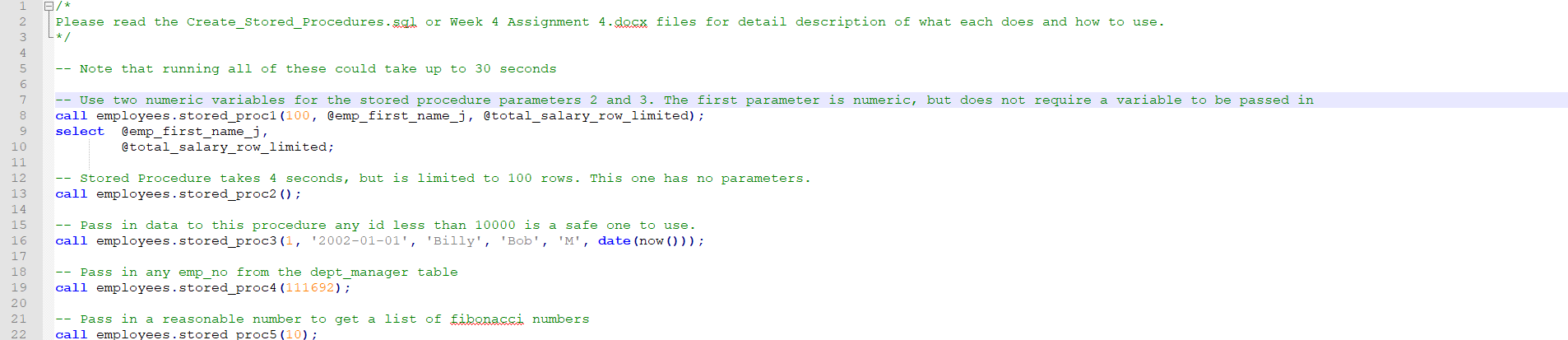
**I recommend zooming in at 150%.**

Create\_Stored\_Procedure.sql





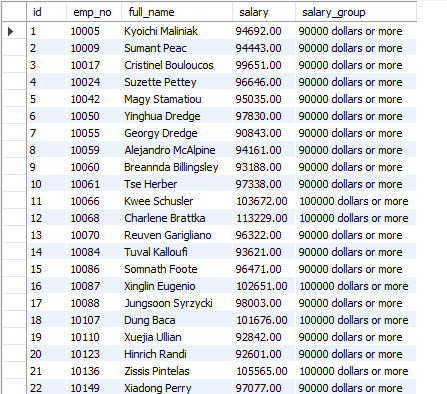
Run\_Stored\_Procedure.sql



Stored Procedure 1 Results:



Stored Procedure 2 Results:



Stored Procedure 3 Results:



Stored Procedure 4 Results:



Stored Procedure 5 Results:



**URL to GitHub Repository:** <https://github.com/jamesaiello42/Intro-to-MySQL-Week-4-Coding-Assignment>