Relational Databases with MySQL Week 4 Coding Assignment

Sean Flaherty

**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:**

1)

Mysql> CREATE PROCEDURE MaxSalary ()

BEGIN

SELECT MAX(salary)

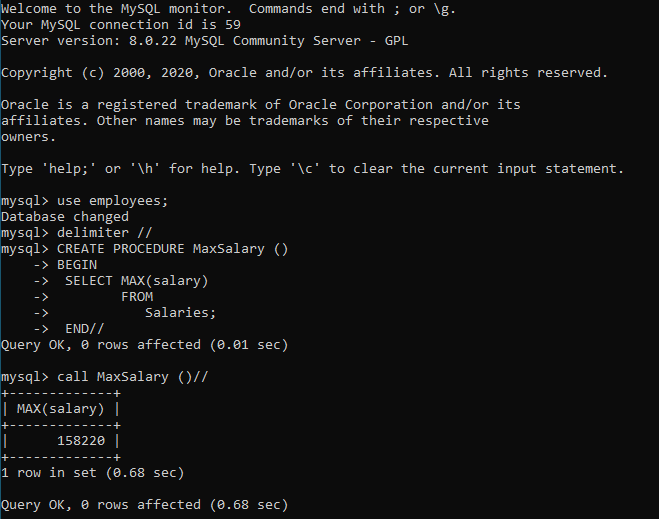
FROM

Salaries;

END//

call MaxSalary ()//

**Description:** This procedure will call on the max function to determine what the highest salary from the salaries table is.

****

2)

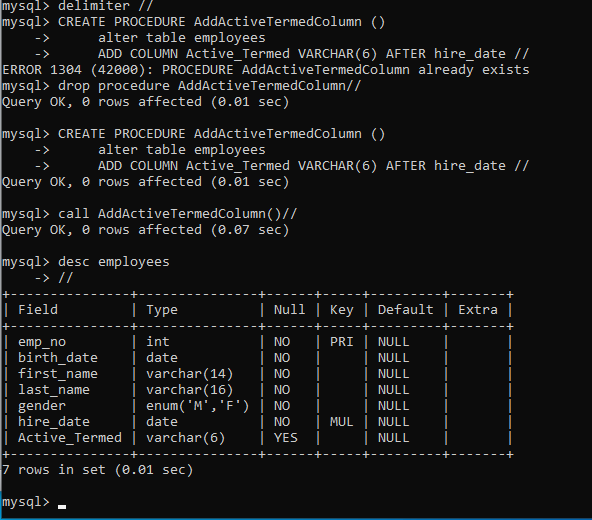
Mysql>CREATE PROCEDURE AddActiveTermedColumn ()

alter table employees

ADD COLUMN Active\_Termed VARCHAR(6) AFTER hire\_date //

call AddActiveTermedColumn()

**Description**: This procedure will make an Active Termed column in the employees table.

****

3)

mysql> CREATE PROCEDURE updatingstatus (IN status\_date varchar(10))

update EMPLOYEES

SET Active\_Termed = CASE

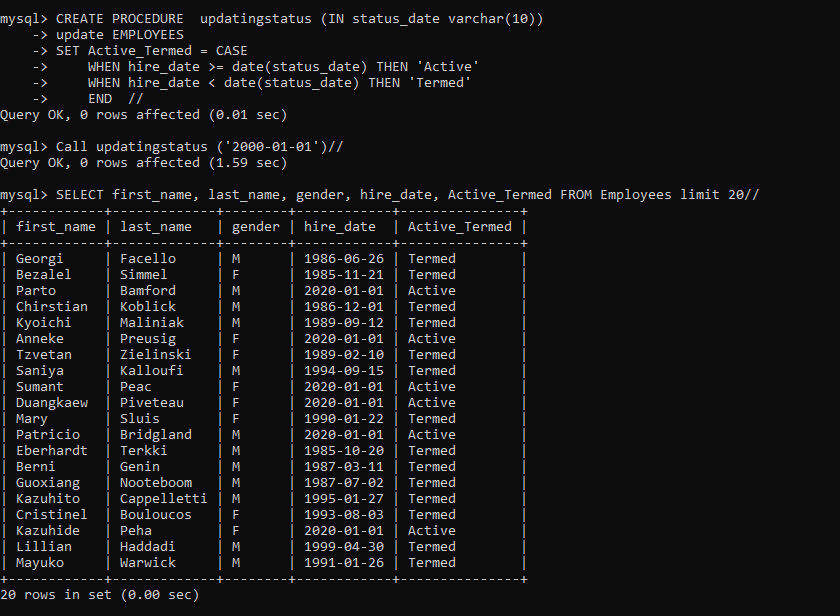
WHEN hire\_date >= date(status\_date) THEN 'Active'

WHEN hire\_date < date(status\_date) THEN 'Termed'

END //

Call updatingstatus ('2000-01-01')

**Description:** This procedure will make populate Active Termed column in the employees table that was just created based off a date that the user puts in. The call I have the date set off of January 1, 2000. So anyone hired before then was termed, and everyone hired after is still active. This needed to be ran as a Varchar in order for it to work.

****

4)

mysql> CREATE PROCEDURE oldtimers (IN passed\_date varchar(10))

Begin

select

a.emp\_no,

a.first\_name,

a.last\_name

FROM employees a

inner join titles b using (emp\_no)

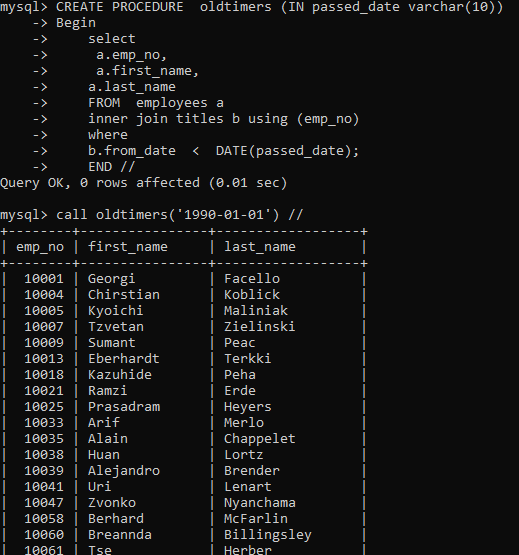
where

b.from\_date < DATE(passed\_date);

END //

call oldtimers('1990-01-01') //

**Description:** Similar concept to the Active Termed but it uses joins and the from\_date to generate what employees were at the companies since the January 1st, 1990 (which is a variable that can be changed), and are now officially “old timers.” Also, needed to be ran as a varchar to work.

****

5)

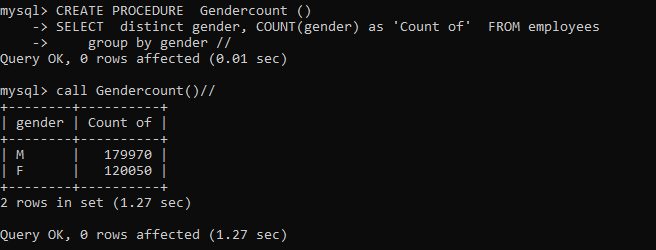
CREATE PROCEDURE Gendercount ()

SELECT distinct gender, COUNT(gender) as 'Count of' FROM employees

group by gender //

call Gendercount()//

**Description:** The procedure is using the count function and separating the gender into males and females. So we are able to see how many of each gender there is.

****

**URL to GitHub Repository:**

<https://github.com/sflaherty50/Week-4-SQL>