

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.

```

delimiter //
create procedure MaxSalary(OUT highestsalary int)
begin
    select max(salary) into highestsalary from salaries;
end; //
delimiter ;

call MaxSalary(@M);
select @M;

```

```

delimiter //
create procedure MinSalary(OUT lowestsalary int)
begin
    select min(salary) into lowestsalary from salaries;
end; //
delimiter ;

call MinSalary(@M);
select @M;

```

DELIMITER \$\$

```

CREATE PROCEDURE IsEmployeeAgeGroupSenior(
    IN pEmployeeNumber INT,
    OUT pAgeGroup VARCHAR(20))
BEGIN
    DECLARE birthDate date DEFAULT 0;

    SELECT birth_date
    INTO birthDate
    FROM employees
    WHERE emp_no = pEmployeeNumber;

    IF (NOW() - birthDate) > 55 THEN
        SET pAgeGroup = 'SENIOR';
    ELSE
        SET pAgeGroup = 'NOT SENIOR';
    END IF;
END$$

```

DELIMITER ;

```

call IsEmployeeAgeGroupSenior(14178,@ageGroup);

```

```
select @ageGroup;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE IsEmployeeAgeGroupAdult(  
    IN pEmployeeNumber INT,  
    OUT pAgeGroup VARCHAR(20))  
BEGIN  
    DECLARE birthDate date DEFAULT 0;  
  
    SELECT birth_date  
    INTO birthDate  
    FROM employees  
    WHERE emp_no = pEmployeeNumber;  
  
    IF (NOW() - birthDate) > 35 AND (NOW() - birthDate) <= 55 THEN  
        SET pAgeGroup = 'ADULT';  
    ELSE  
        SET pAgeGroup = 'NOT ADULT';  
    END IF;  
END$$
```

```
DELIMITER ;
```

```
call IsEmployeeAgeGroupAdult(12176,@ageGroup);  
select @ageGroup;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE IsEmployeeAgeGroupYouth(  
    IN pEmployeeNumber INT,  
    OUT pAgeGroup VARCHAR(20))  
BEGIN  
    DECLARE birthDate date DEFAULT 0;  
  
    SELECT birth_date  
    INTO birthDate  
    FROM employees  
    WHERE emp_no = pEmployeeNumber;  
  
    IF (NOW() - birthDate) > 18 AND (NOW() - birthDate) <= 35 THEN
```

```

        SET pAgeGroup = 'YOUTH';
    ELSE
        SET pAgeGroup = 'NOT YOUTH';
    END IF;
END$$

DELIMITER ;

call IsEmployeeAgeGroupYouth(17628,@ageGroup);
select @ageGroup;

```

Description to Procedures:

The first and the second procedures are used to display the maximum or minimum salary an employee gets. It uses an OUT parameter, which is used to pass the max or min salary value when the procedure is called.

The third, fourth, and the fifth parameters are used to categorize a given employee under age group categories. It uses both an IN and OUT parameter, it uses an employee number as an input and outputs the age group as a text. In addition to that it also uses a conditional statement IF to check the age category.

Screenshots:

```

1  DELIMITER $$
2
3  CREATE PROCEDURE GetEmployeeAgeGroup(
4      IN pEmployeeNumber INT,
5      OUT pAgeGroup VARCHAR(20))
6  BEGIN
7      DECLARE birthDate date DEFAULT 0;
8
9      SELECT birth_date
10     INTO birthDate
11     FROM employees
12     WHERE emp_no = pEmployeeNumber;
13
14     IF NOW() - birthDate > 56 THEN
15         SET pAgeGroup = 'SENIOR';
16     ELSE
17         SET pAgeGroup = 'NOT SENIOR';
18     END IF;
19 END$$
20
21 DELIMITER ;
22
23 call GetEmployeeAgeGroup(14178,@ageGroup);
24 select @ageGroup;
25

```

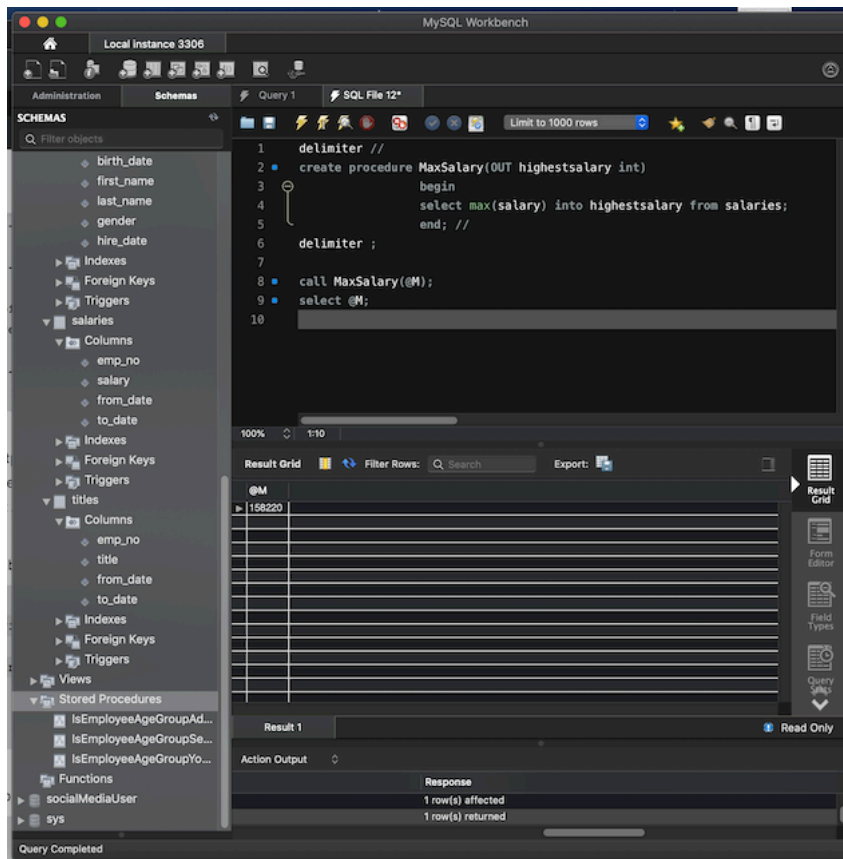
Result Grid

@ageGroup
SENIOR

Result 5 Read Only

Action Output

	Time	Ac...	Response	Duration / Fetch Time
52	21:05:21	EX...	OK	0.000 sec



URL to GitHub Repository:

<https://github.com/dwold/Week10Assignment>