

MySQL Procedures (LAB Lecture)



Department of Information Technology
Database Systems II (IT216)
Spring 2024-2025
Week 12 – April 27, 2025
Lecturer: Soma Soleimanzadeh



1

Contents

- Stored Routines in MySQL
 - Functions
 - Procedures
- Procedures
- Syntax of Creating Procedure in MySQL
- Procedure vs. Function

2

Syntax of Creating Procedure in MySQL

DELIMITER %%

```
CREATE PROCEDURE procedure_name (IN / OUT parameter1 datatype1,  
                                IN / OUT parameter2 datatype2,  
                                ... )
```

BEGIN

<Declaration Part of Procedure>

<Execution Part of Procedure>

END%%

DELIMITER ;

3

Procedure

DELIMITER //

```
CREATE PROCEDURE procedure_name  
(IN / OUT parameter1 datatype1,  
IN / OUT parameter2 datatype2,  
... )
```

BEGIN

<Declaration Part of Procedure>

<Execution Part of Procedure>

END//

DELIMITER ;

Function

DELIMITER //

```
CREATE FUNCTION function_name  
(input_parameter1 datatype1,  
input_parameter2 datatype2,  
... )
```

RETURNS return_datatype

BEGIN

<Declaration Part of Function>

<Execution Part of Function>

RETURN return_value;

END//

DELIMITER ;

4

Let's Create the Database and Tables

- Create a database named **school** and activate it.
- Create the **Student** table and enter the data inside the table.
 - **SID** is **primary key** and **auto_increment**.

```
Create Database school;
Use school;

Create Table Student
(  SID int auto_increment,
   SName varchar(30),
   Stage int,
   Mark int,
   primary key (SID)
);
```

Student

<u>SID</u>	SName	Stage	Mark
1	Sara	3	55
2	Lana	3	80
3	Farhad	4	70

5

Example 1 – A Procedure without IN and OUT Parameters

- Create a procedure named **thirdStage** to display all information of students in 3rd stage.
- Invoke (call) the procedure.

```
DELIMITER %%
CREATE Procedure thirdStage ( )
BEGIN
    SELECT * from Student where Stage=3;
END %%
DELIMITER ;
```

```
CALL thirdStage( );
```

Student

<u>SID</u>	SName	Stage	Mark
1	Sara	3	55
2	Lana	3	80
3	Farhad	4	70

6

Example 2 – A Procedure with Only IN Parameter

- Create a procedure named **addBonus** that takes ID of a student, and updates his/her mark by adding 5 bonus mark.
- Call the procedure by passing SID=2 to the procedure.

```
DELIMITER $$  
CREATE Procedure addBonus (IN ID int )  
BEGIN  
    UPDATE Student SET Mark = Mark + 5 WHERE SID = ID;  
END $$  
DELIMITER ;
```

```
CALL addBonus( 2 );
```

```
SELECT * FROM Student;
```

Student

SID	SName	Stage	Mark
1	Sara	3	55
2	Lana	3	80
3	Farhad	4	70

7

Example 3 – A Procedure with Only OUT Parameter

- Create a procedure named **MyProcedure** that calculates and displays the average mark of all students.
- Call the procedure.

```
DELIMITER $$  
CREATE Procedure MyProcedure (OUT averageAge decimal (3,1) )  
BEGIN  
    SELECT avg(Mark) INTO averageAge FROM Student;  
END $$  
DELIMITER ;
```

```
CALL MyProcedure(@result );
```

```
SELECT @result;
```

Student

SID	SName	Stage	Mark
1	Sara	3	55
2	Lana	3	80
3	Farhad	4	70

8

Lab Work!

- Create a procedure named **MarkLetter** that takes a student ID, and displays his/her mark in letter according to the following conditions:

- mark $\geq$ 80 $\rightarrow$ 'A'
- mark $\geq$ 60 and mark $<$ 80 $\rightarrow$ 'B'
- mark $<$ 60 $\rightarrow$ 'C'

Student

SID	SName	Stage	Mark
1	Sara	3	55
2	Lana	3	80
3	Farhad	4	70

```
DELIMITER $$
```

```
CREATE Procedure MarkLetter (IN ID int , OUT letter varchar (1) )
```

```
BEGIN
```

```
    DECLARE stuMark int;
```

```
    SELECT Mark INTO stuMark FROM Student WHERE SID = ID;
```

```
    SELECT
```

```
        CASE
```

```
            WHEN stuMark  $\geq$  80 THEN 'A'
```

```
            WHEN stuMark  $\geq$  60 AND stuMark  $<$  80 THEN 'B'
```

```
            ELSE 'C'
```

```
        END INTO letter;
```

```
END$$
```

```
DELIMITER ;
```

```
CALL MarkLetter(3, @result );
```

```
SELECT @result;
```

9