JDBC Tutorial

Stored Procedure:

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.

So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

In this tutorial, we'll learn about the stored procedures in SQL and how to use them with examples.

In SQL, stored procedure is a set of statement(s) that perform some defined actions. We make stored procedures so that we can reuse statements that are used frequently.

Stored procedures are similar to functions in programming. They can accept parameters, and perform operations when we call them.

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `us\_customers`()

BEGIN

SELECT customer\_id, first\_name

FROM Customers

WHERE Country = 'USA';

END$$

DELIMITER ;

https://www.softwaretestinghelp.com/mysql-stored-procedure/

**CREATE** **TABLE** studentMarks (stud\_id **SMALLINT**(5) NOT NULL AUTO\_INCREMENT **PRIMARY** **KEY**, total\_marks **INT**, grade **VARCHAR**(5));

**INSERT** **INTO** studentMarks(total\_marks, grade) **VALUES**(450, 'A'), (480, 'A+'), (490, 'A++'), (440, 'B+'),(400, 'C+'),(380,'C')

,(250, 'D'),(200,'E'),(100,'F'),(150,'F'),(220, 'E');

DELIMITER $$

**CREATE** **PROCEDURE** GetStudentData()

**BEGIN**

**SELECT** \* **FROM** studentMarks;

**END**$$

DELIMITER ;

CALL GetStudentData();

With Parameters:

DELIMITER //

**CREATE** **PROCEDURE simplilearn.**spGetDetailsByStudentName(IN studentId **INT**)

**BEGIN**

**SELECT** \* **FROM** studentMarks **where** stud\_id = studentId;

**END** //

DELIMITER ;

CALL **simplilearn**.spGetDetailsByStudentName(1);

Creating Procedure with Output Parameters

In the previous section, we learned about using Input (IN) parameters which were mentioned when it was being called.

Let’s now see how we can use Output or OUT parameters.

**For example:** Suppose we want to calculate the average marks of all the students from the studentMarks table and return the average as an OUT field.

|  |
| --- |
| DELIMITER //  **CREATE** **PROCEDURE** simplilearn.spGetAverageMarks(**OUT** average **DECIMAL**(5,2))  **BEGIN**  **SELECT** AVG(total\_marks) **INTO** average **FROM** studentMarks;  **END** //  DELIMITER ; |

CALL simplilearn.spGetAverageMarks(@average\_marks);

Once it is executed, you can run SELECT for the OUT variable, to fetch the result.

**SELECT** @average\_marks;

DELIMITER //

**CREATE** **PROCEDURE** simplilearn.spCountOfBelowAverage(**OUT** countBelowAverage **INT**)

**BEGIN**

**DECLARE** avgMarks **DECIMAL**(5,2) **DEFAULT** 0;

**SELECT** AVG(total\_marks) **INTO** avgMarks **FROM** studentMarks;

**SELECT** COUNT(\*) **INTO** countBelowAverage **FROM** studentMarks **WHERE** total\_marks < avgMarks;

**END** //

DELIMITER ;

CALL simplilearn.spCountOfBelowAverage(@countBelowAverage);

**SELECT** @countBelowAverage;

<https://cs.uwaterloo.ca/~tozsu/courses/CS348/F12/assignments/StudentCourseSchema.html> - project schema

<https://github.com/MujtabaMohsin/Administrative-portal-for-Learner-Academy/tree/main/build/classes/com/simplilearn/models>

<https://dba.stackexchange.com/questions/278459/school-database-schema>

DELIMITER //

**CREATE** **PROCEDURE simplilearn.**GetEmployeeDetails()

**BEGIN**

**SELECT** \* **FROM** employee;

**END** //

DELIMITER ;

DELIMITER //

**CREATE** **PROCEDURE simplilearn.**GetEmployeeDetailsById(IN empId INT)

**BEGIN**

**SELECT** \* **FROM** employee where id=empId;

**END** //

DELIMITER ;

DELIMITER //

**CREATE** **PROCEDURE simplilearn.**spGetDetailsByStudentName(IN studentId **INT**)

**BEGIN**

**SELECT** \* **FROM** studentMarks **where** stud\_id = studentId;

**END** //

DELIMITER ;