create table candidates(id int primary key,first\_name varchar(20),last\_name varchar(20),skill varchar(20));

insert into candidates values(2,'ravi','kumar','java');

show tables;

DELIMITER $$

CREATE PROCEDURE getcandidateskill(IN candidate\_id INT)

BEGIN

SELECT id, first\_name,last\_name, skill

FROM candidates WHERE id = candidate\_id;

END $$

DELIMITER ;

CALL getcandidateskill(2);

Introducing to CallableStatement and stored procedure call syntax

To call stored procedures or stored functions in MySQL from JDBC, you use CallableStatement object, which inherits from [PreparedStatement object](https://www.mysqltutorial.org/mysql-jdbc-update). The general syntax of calling a stored procedure is as follows:

{?= call procedure\_name(param1,param2,...)}

Code language: SQL (Structured Query Language) (sql)

You wrap the stored procedure call within braces ({}). If the stored procedure returns a value, you need to add the question mark and equal (?=) before the call keyword. If a stored procedure does not return any values, you just omit the ?= sign. In case the stored procedure accepts any parameters, you list them within the opening and closing parentheses after the stored procedure’s name.

The following are examples of using the syntax for calling stored procedures in different contexts:

|  |  |
| --- | --- |
| Syntax | Stores Procedures |
| {  call procedure\_name() } | Accept no parameters and return no value |
| { call procedure\_name(?,?) } | Accept two parameters and return no value |
| {?= call procedure\_name() } | Accept no parameter and return value |
| {?= call procedure\_name(?) } | Accept one parameter and return value |

Notice that question mark placeholder (?) can be used for both IN ,OUT, and INOUT parameters. For detailed information on different parameter types in stored procedures, check it out [MySQL stored procedure parameters tutorial](https://www.mysqltutorial.org/stored-procedures-parameters.aspx).

JDBC MySQL stored procedure example

**package** com.revature.mavenprojects.Mavenapp1;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**public** **class** DBUtil {

**public** **static** Connection getConnection() **throws** Exception{

Connection con = DriverManager.*getConnection*("jdbc:mysql://127.0.0.1:3307/revature", "root", "root");

**return** con;

}

}

**package** com.revature.mavenprojects.Mavenapp1;

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.CallableStatement;

**public** **class** Demo {

**public** **static** **void** getSkills(**int** candidateId) **throws** Exception {

//

String query = "{ call getcandidateskill(?) }";

ResultSet rs;

Connection conn = DBUtil.*getConnection*();

CallableStatement stmt = conn.prepareCall(query);

stmt.setInt(1, candidateId);

rs = stmt.executeQuery();

**while** (rs.next()) {

System.***out***.println( rs.getString(1) + " " + rs.getString(2));

}

}

**public** **static** **void** main(String[] args) **throws** Exception{

*getSkills*(2);

}

}