import java.sql.\*;

/\*\*

\* Test calling stored procedure with IN parameters

\*

\*

\*/

public class IncreaseSalariesForDepartment {

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

Connection myConn = null;

CallableStatement myStmt = null;

try {

// Get a connection to database

myConn = DriverManager.*getConnection*(

"jdbc:mysql://localhost:3306/demo", "student", "student");

String theDepartment = "Engineering";

int theIncreaseAmount = 100;

// Show salaries BEFORE

System.*out*.println("Salaries BEFORE\n");

*showSalaries*(myConn, theDepartment);

// Prepare the stored procedure call

myStmt = myConn

.prepareCall("{call increase\_salaries\_for\_department(?, ?)}");

// Set the parameters

myStmt.setString(1, theDepartment);

myStmt.setDouble(2, theIncreaseAmount);

// Call stored procedure

System.*out*.println("\n\nCalling stored procedure. increase\_salaries\_for\_department('" + theDepartment + "', " + theIncreaseAmount + ")");

myStmt.execute();

System.*out*.println("Finished calling stored procedure");

// Show salaries AFTER

System.*out*.println("\n\nSalaries AFTER\n");

*showSalaries*(myConn, theDepartment);

} catch (Exception exc) {

exc.printStackTrace();

} finally {

*close*(myConn, myStmt, null);

}

}

private static void showSalaries(Connection myConn, String theDepartment) throws SQLException {

PreparedStatement myStmt = null;

ResultSet myRs = null;

try {

// Prepare statement

myStmt = myConn

.prepareStatement("select \* from employees where department=?");

myStmt.setString(1, theDepartment);

// Execute SQL query

myRs = myStmt.executeQuery();

// Process result set

while (myRs.next()) {

String lastName = myRs.getString("last\_name");

String firstName = myRs.getString("first\_name");

double salary = myRs.getDouble("salary");

String department = myRs.getString("department");

System.*out*.printf("%s, %s, %s, %.2f\n", lastName, firstName, department, salary);

}

} catch (Exception exc) {

exc.printStackTrace();

} finally {

*close*(myStmt, myRs);

}

}

private static void close(Connection myConn, Statement myStmt,

ResultSet myRs) throws SQLException {

if (myRs != null) {

myRs.close();

}

if (myStmt != null) {

myStmt.close();

}

if (myConn != null) {

myConn.close();

}

}

private static void close(Statement myStmt, ResultSet myRs)

throws SQLException {

*close*(null, myStmt, myRs);

}

}

SQL PROCEDURES:

--

-- DEFINE STORED PROCEDURES

--

use demo;

DELIMITER $$

DROP PROCEDURE IF EXISTS `get\_count\_for\_department`$$

CREATE DEFINER=`student`@`localhost` PROCEDURE `get\_count\_for\_department`(IN the\_department VARCHAR(64), OUT the\_count INT)

BEGIN

SELECT COUNT(\*) INTO the\_count FROM employees where department=the\_department;

END$$

DELIMITER ;

DELIMITER $$

DROP PROCEDURE IF EXISTS `get\_employees\_for\_department`$$

CREATE DEFINER=`student`@`localhost` PROCEDURE `get\_employees\_for\_department`(IN the\_department VARCHAR(64))

BEGIN

SELECT \* from employees where department=the\_department;

END$$

DELIMITER ;

DELIMITER $$

DROP PROCEDURE IF EXISTS `greet\_the\_department`$$

CREATE DEFINER=`student`@`localhost` PROCEDURE `greet\_the\_department`(INOUT department VARCHAR(64))

BEGIN

SET department = concat('Hello to the awesome ', department, ' team!');

END$$

DELIMITER ;

DELIMITER $$

DROP PROCEDURE IF EXISTS `increase\_salaries\_for\_department`$$

CREATE DEFINER=`student`@`localhost` PROCEDURE `increase\_salaries\_for\_department`(IN the\_department VARCHAR(64), IN increase\_amount DECIMAL(10,2))

BEGIN

UPDATE EMPLOYEES SET salary= salary + increase\_amount where department=the\_department;

END$$

DELIMITER ;