

Lecture 10_Stored Procedures Part 2

DRIVER

The screenshot shows the Eclipse IDE with a project named 'jdbc-lecture-10-stored-procedures-part-2'. The 'src' folder contains a 'Driver.java' file. The code in 'Driver.java' is as follows:

```
1
2
3 import java.sql.*;
4
5 public class Driver {
6
7     public static void main(String[] args) throws SQLException {
8
9         Connection myConn = null;
10        PreparedStatement myStmt = null;
11        ResultSet myRs = null;
12
13        try {
14            // 1. Get a connection to database
15            myConn = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo1", "student", "student");
16
17            // 2. Prepare statement
18            myStmt = myConn.prepareStatement("select * from employees where salary > ? and department=?");
19
20            // 3. Set the parameters
21            myStmt.setDouble(1, 80000);
22            myStmt.setString(2, "Legal");
23
24            // 4. Execute SQL query
25            myRs = myStmt.executeQuery();
26
27            // 5. Display the result set
28            display(myRs);
29        }
30    }
31}
```

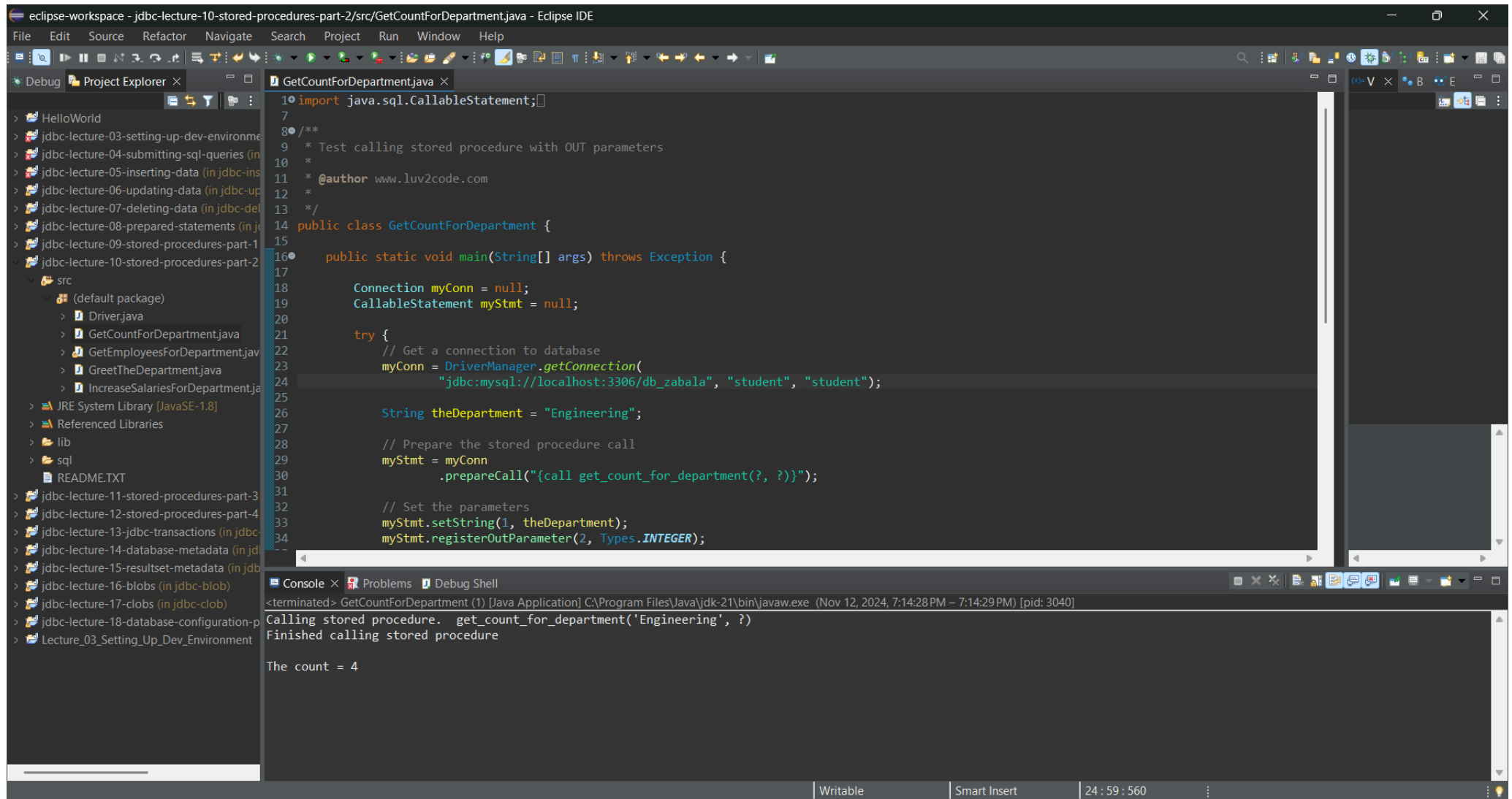
The console output shows the execution of the stored procedure 'increase_salaries_for_department' for the 'Engineering' department with a salary increase of 10000. The output is as follows:

```
<terminated> IncreaseSalariesForDepartment [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 12, 2024, 7:07:37 PM - 7:07:38 PM) [pid: 11304]
Calling stored procedure.  increase_salaries_for_department('Engineering', 10000)
Finished calling stored procedure

Salaries AFTER

Public, Mary, Engineering, 85000.00
Johnson, Lisa, Engineering, 60000.00
Brown, Bill, Engineering, 60000.00
Fowler, Mary, Engineering, 75000.00
```

Get Count for Department



The screenshot displays the Eclipse IDE interface. The main editor window shows the file `GetCountForDepartment.java` with the following code:

```
1 import java.sql.CallableStatement;
2
3 /**
4  * Test calling stored procedure with OUT parameters
5  *
6  * @author www.luv2code.com
7  */
8
9 public class GetCountForDepartment {
10
11     public static void main(String[] args) throws Exception {
12
13         Connection myConn = null;
14         CallableStatement myStmt = null;
15
16         try {
17             // Get a connection to database
18             myConn = DriverManager.getConnection(
19                 "jdbc:mysql://localhost:3306/db_zabala", "student", "student");
20
21             String theDepartment = "Engineering";
22
23             // Prepare the stored procedure call
24             myStmt = myConn
25                 .prepareCall("{call get_count_for_department(?, ?)}");
26
27             // Set the parameters
28             myStmt.setString(1, theDepartment);
29             myStmt.registerOutParameter(2, Types.INTEGER);
30
31         } catch (SQLException e) {
32             e.printStackTrace();
33         } finally {
34             if (myConn != null) myConn.close();
35             if (myStmt != null) myStmt.close();
36         }
37     }
38 }
```

The left sidebar shows the Project Explorer with the following structure:

- src
 - (default package)
 - Driver.java
 - GetCountForDepartment.java
 - GetEmployeesForDepartment.java
 - GreetTheDepartment.java
 - IncreaseSalariesForDepartment.java
- JRE System Library [JavaSE-1.8]
- Referenced Libraries
- lib
- sql
- README.TXT

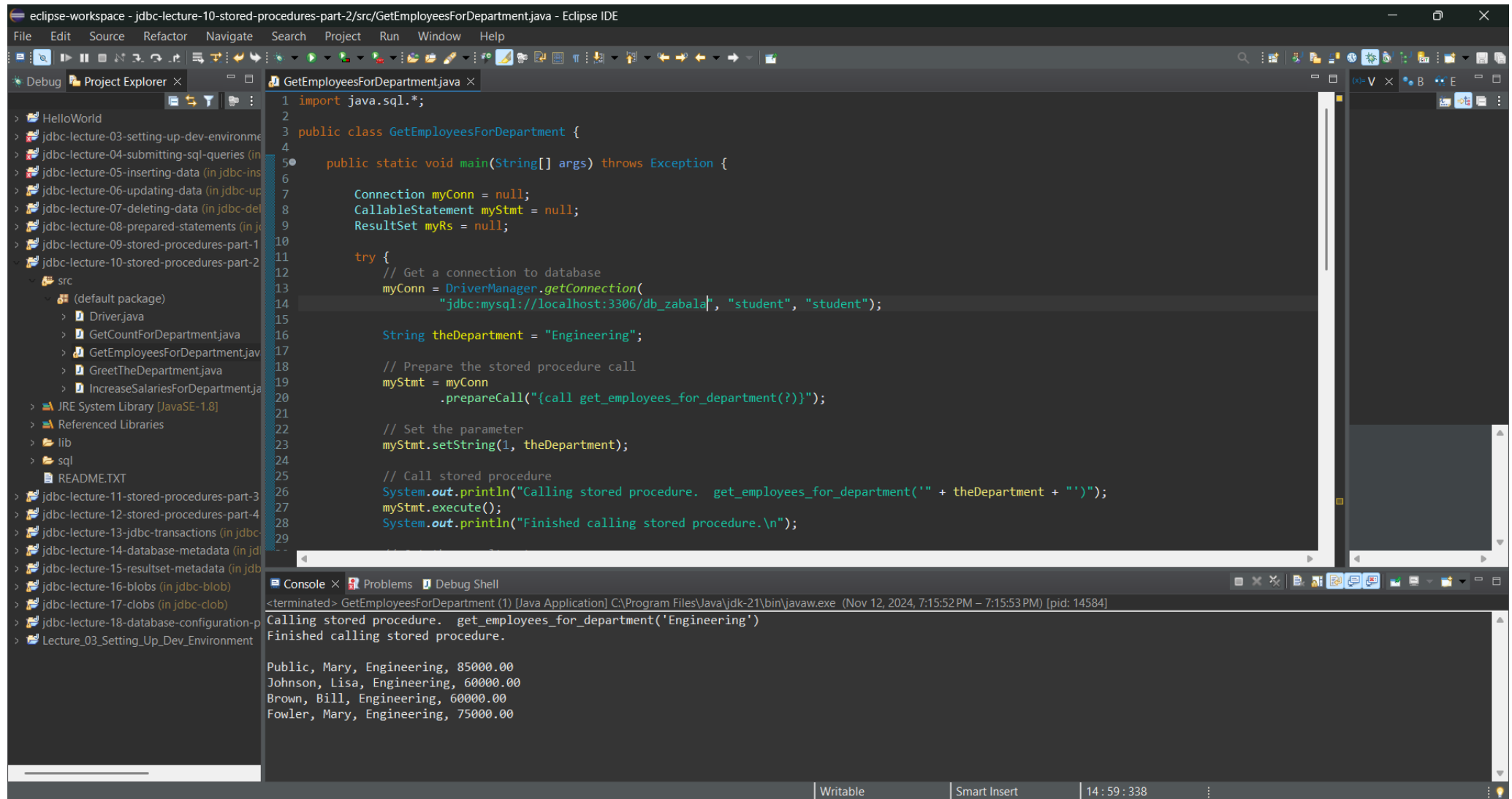
The bottom console window shows the output of the program:

```
<terminated> GetCountForDepartment (1) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 12, 2024, 7:14:28 PM – 7:14:29 PM) [pid: 3040]
Calling stored procedure.  get_count_for_department('Engineering', ?)
Finished calling stored procedure

The count = 4
```

The status bar at the bottom indicates the file is Writable, Smart Insert is enabled, and the time is 24:59:56.

Get Employees for Department



The screenshot shows the Eclipse IDE with a Java project named 'jdbc-lecture-10-stored-procedures-part-2'. The main editor displays the file 'GetEmployeesForDepartment.java'. The code is as follows:

```
1 import java.sql.*;
2
3 public class GetEmployeesForDepartment {
4
5     public static void main(String[] args) throws Exception {
6
7         Connection myConn = null;
8         CallableStatement myStmt = null;
9         ResultSet myRs = null;
10
11         try {
12             // Get a connection to database
13             myConn = DriverManager.getConnection(
14                 "jdbc:mysql://localhost:3306/db_zabala", "student", "student");
15
16             String theDepartment = "Engineering";
17
18             // Prepare the stored procedure call
19             myStmt = myConn
20                 .prepareCall("{call get_employees_for_department(?)}");
21
22             // Set the parameter
23             myStmt.setString(1, theDepartment);
24
25             // Call stored procedure
26             System.out.println("Calling stored procedure.  get_employees_for_department('" + theDepartment + "')");
27             myStmt.execute();
28             System.out.println("Finished calling stored procedure.\n");
29         }
30     }
31 }
```

The console output shows the execution of the program:

```
<terminated> GetEmployeesForDepartment (1) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 12, 2024, 7:15:52 PM - 7:15:53 PM) [pid: 14584]
Calling stored procedure.  get_employees_for_department('Engineering')
Finished calling stored procedure.

Public, Mary, Engineering, 85000.00
Johnson, Lisa, Engineering, 60000.00
Brown, Bill, Engineering, 60000.00
Fowler, Mary, Engineering, 75000.00
```

The status bar at the bottom indicates 'Writable', 'Smart Insert', and the time '14 : 59 : 338'.

Greet the Department

The screenshot shows the Eclipse IDE interface. The main editor displays the file `GreetTheDepartment.java` with the following code:

```
1 import java.sql.*;
2
3 /**
4  * Test calling stored procedure with INOUT parameters
5  *
6  * @author www.luv2code.com
7  *
8  */
9 public class GreetTheDepartment {
10
11     public static void main(String[] args) throws Exception {
12
13         Connection myConn = null;
14         CallableStatement myStmt = null;
15
16         try {
17             // Get a connection to database
18             myConn = DriverManager.getConnection(
19                 "jdbc:mysql://localhost:3306/db_zabala", "student", "student");
20
21             String theDepartment = "Engineering";
22
23             // Prepare the stored procedure call
24             myStmt = myConn
25                 .prepareCall("{call greet_the_department(?)}");
26
27             // Set the parameters
28             myStmt.registerOutParameter(1, Types.VARCHAR);
29             myStmt.setString(1, theDepartment);
30
31             myStmt.execute();
32
33             String result = myStmt.getString(1);
34
35             System.out.println("The result = " + result);
36
37         } catch (SQLException e) {
38             e.printStackTrace();
39         } finally {
40             if (myConn != null) {
41                 myConn.close();
42             }
43             if (myStmt != null) {
44                 myStmt.close();
45             }
46         }
47     }
48 }
```

The left sidebar shows the Project Explorer with the following structure:

- src
 - (default package)
 - Driver.java
 - GetCountForDepartment.java
 - GetEmployeesForDepartment.java
 - GreetTheDepartment.java
 - IncreaseSalariesForDepartment.java
- JRE System Library [JavaSE-1.8]
- Referenced Libraries
 - lib
 - sql
- README.TXT

The bottom console shows the output of the program:

```
<terminated> GreetTheDepartment (1) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 12, 2024, 7:17:06 PM - 7:17:07 PM) [pid: 9080]
Calling stored procedure. greet_the_department('Engineering')
Finished calling stored procedure

The result = Hello to the awesome Engineering team!
```

The status bar at the bottom indicates the file is writable, smart insert is enabled, and the time is 19:59:40.

Increase Salaries for Department

The screenshot displays the Eclipse IDE interface. The main editor window shows the file `IncreaseSalariesForDepartment.java` with the following code:

```
1 import java.sql.*;
2
3 /**
4  * Test calling stored procedure with IN parameters
5  *
6  * @author www.luv2code.com
7  *
8  */
9 public class IncreaseSalariesForDepartment {
10
11     public static void main(String[] args) throws Exception {
12
13         Connection myConn = null;
14         CallableStatement myStmt = null;
15
16         try {
17             // Get a connection to database
18             myConn = DriverManager.getConnection(
19                 "jdbc:mysql://localhost:3306/db_zabala", "student", "student");
20
21             String theDepartment = "Engineering";
22             int theIncreaseAmount = 10000;
23
24             // Show salaries BEFORE
25             System.out.println("Salaries BEFORE\n");
26             showSalaries(myConn, theDepartment);
27
28             // Prepare the stored procedure call
29             myStmt = myConn
30         }
31     }
32 }
```

The left-hand Project Explorer shows a project structure with various lecture-related files. The bottom Console window displays the output of the program:

```
<terminated> IncreaseSalariesForDepartment (1) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Nov 12, 2024, 7:17:56 PM – 7:17:57 PM) [pid: 15260]
Salaries BEFORE
Public, Mary, Engineering, 85000.00
Johnson, Lisa, Engineering, 60000.00
Brown, Bill, Engineering, 60000.00
Fowler, Mary, Engineering, 75000.00

Calling stored procedure.  increase_salaries_for_department('Engineering', 10000)
Finished calling stored procedure
```

The status bar at the bottom indicates the editor is in 'Writable' mode, 'Smart Insert' is active, and the time is 19:59:413.