Yaoxi Luo
Prof. Scott Leutenegger
COMP – 3421
Mar.07, 2016

Assignment 7

(1)jdbcFunction1.java:

(Description)
In this function, it allows the user to change the Employee ID in table Orders. In a company, there is always a situation about firing employees. So the company can use this function easy to use a new employee id to instead of the fired employee id in table Orders. The programming will ask the user the new Employee ID and the old Employee ID. Then it will update the new Employee ID to table Orders automatically.

(Java code)

```
import java.sql.* ;

import java.io.* ;
import java.util.Random ;
import java.util.Scanner ;

public class jdbcFunction1 {
  public static void main(String[] args) {

        int old_emp_id, new_emp_id;

        Connection conn = null ;
        Scanner user_input = new Scanner(System.in);

        try {

                conn = DriverManager.getConnection("jdbc:mysql://
localhost/ToySale?" +
                                   "user=lcocox&password=ll" ) ;

        }
        catch (SQLException ex ) {
          System.out.println("SQLException: " + ex.getMessage()) ;
          System.out.println("SQLState: " + ex.getSQLState()) ;
          System.out.println("VendorError: " + ex.getErrorCode()) ;
        }

        Statement stmt = null ;
```

```
        try {

          System.out.println("What's the new employee's id?");
          new_emp_id = user_input.nextInt();
          System.out.println("Which the old employee's id you want to
change in an order?");
          old_emp_id = user_input.nextInt();
          System.out.println("You entered the New Employee Id is " +
new_emp_id + ", and the Old Employee ID is " + old_emp_id + ".");

          String valueString;

          stmt = conn.createStatement() ;

          valueString = "update Orders set Emp_id = ";
          valueString += Integer.toString(new_emp_id);
          valueString += " where Emp_id = ";
          valueString += Integer.toString(old_emp_id);
          valueString += ";";

          System.out.println("About to execute: " + valueString);

          stmt.executeUpdate( valueString );

          stmt.close() ;
          conn.close() ;

        }

        catch (SQLException ex ) {
          System.out.println("SQLException: " + ex.getMessage()) ;
          System.out.println("SQLState: " + ex.getSQLState()) ;
          System.out.println("VendorError: " + ex.getErrorCode()) ;
        }
  }
}
```

(Script File)

Run jdbcFunction1.java in Terminal:

```
lcocoxs-MacBook-Air:Documents lcocox$ javac jdbcFunction1.java
lcocoxs-MacBook-Air:Documents lcocox$ java jdbcFunction1
Mon Mar 07 20:57:35 MST 2016 WARN: Establishing SSL connection without
server's identity verification is not recommended. According to MySQL
5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be
established by default if explicit option isn't set. For compliance
```

```
with existing applications not using SSL the verifyServerCertificate
property is set to 'false'. You need either to explicitly disable SSL
by setting useSSL=false, or set useSSL=true and provide truststore for
server certificate verification.
What's the new employee's id?
99
Which the old employee's id you want to change in an order?
88
You entered the New Employee Id is 99, and the Old Employee ID is 88.
About to execute: update Orders set Emp_id = 99 where Emp_id = 88;

-------------------------------------------

Results in MySQL:

(Before I ran jdbcFunction1.java:)

mysql> select * from Orders where Emp_id = 88;
+----------+--------+--------+------------+
| Order_id | Cus_id | Emp_id | OrderDate  |
+----------+--------+--------+------------+
|      428 |   2292 |     88 | 2013-11-08 |
|      861 |    169 |     88 | 2014-03-03 |
|     1892 |    603 |     88 | 2014-04-07 |
|     3189 |   1167 |     88 | 2012-03-01 |
|     4779 |    702 |     88 | 2012-11-15 |
+----------+--------+--------+------------+
5 rows in set (0.00 sec)

mysql> select * from Orders where Emp_id = 99;
+----------+--------+--------+------------+
| Order_id | Cus_id | Emp_id | OrderDate  |
+----------+--------+--------+------------+
|      610 |    955 |     99 | 2012-04-04 |
|     1906 |    576 |     99 | 2013-09-27 |
|     2891 |   2315 |     99 | 2014-05-06 |
|     3450 |      3 |     99 | 2013-12-14 |
|     4238 |   1927 |     99 | 2014-08-06 |
+----------+--------+--------+------------+
5 rows in set (0.01 sec)

(After I ran jdbcFunction1.java:)

mysql> select * from Orders where Emp_id = 99;
+----------+--------+--------+------------+
| Order_id | Cus_id | Emp_id | OrderDate  |
+----------+--------+--------+------------+
|      428 |   2292 |     99 | 2013-11-08 |
|      610 |    955 |     99 | 2012-04-04 |
|      861 |    169 |     99 | 2014-03-03 |
```

```
|     1892 |    603 |     99 | 2014-04-07 |
|     1906 |    576 |     99 | 2013-09-27 |
|     2891 |   2315 |     99 | 2014-05-06 |
|     3189 |   1167 |     99 | 2012-03-01 |
|     3450 |      3 |     99 | 2013-12-14 |
|     4238 |   1927 |     99 | 2014-08-06 |
|     4779 |    702 |     99 | 2012-11-15 |
+----------+--------+--------+------------+
10 rows in set (0.00 sec)

mysql> select * from Orders where Emp_id = 88;
Empty set (0.00 sec)
```

****

(2)jdbcFunction2.java:

(Description)
In this function, it allows the user to insert new employee to table Employee. In a company, there is a situation about hiring the new employee. So the company can use this function easy to insert a new employee data to table Employee. The programming will ask the user to input the information by request and insert them automatically.

(code)

```
import java.sql.* ;

import java.io.* ;
import java.util.Random ;
import java.util.Scanner ;

public class jdbcFunction2 {
  public static void main(String[] args) {

    int emp_id, age;
    String name, gender, hireDate;

    Connection conn = null ;
    Scanner user_input = new Scanner(System.in) ;

    try {

         conn = DriverManager.getConnection("jdbc:mysql://localhost/
ToySale?" +
                          "user=lcocox&password=ll" ) ;

    }
```

```
    catch (SQLException ex ) {
        System.out.println("SQLException: " + ex.getMessage()) ;
        System.out.println("SQLState: " + ex.getSQLState()) ;
        System.out.println("VendorError: " + ex.getErrorCode()) ;
    }


    Statement stmt = null ;

    try {

      System.out.println("Enter (Emp_id, Name, Gender, Age, HireDate)
of an Employee you wish to enter: ") ;
      emp_id = user_input.nextInt() ;
      name = user_input.next() ;
      gender = user_input.next();
      age = user_input.nextInt() ;
      hireDate = user_input.next() ;
      System.out.println( "You entered: " + emp_id + " " + name + " "
+ gender + " " + age + " " + hireDate) ;

      String valueString ;

      stmt = conn.createStatement() ;

          valueString = "insert into Employee values(" ;
          valueString += "'" + Integer.toString(emp_id) + "'" ;
          valueString += ", '" + name + "'" ;
      valueString += ", '" + gender + "'" ;
          valueString += ", '" + Integer.toString(age) + "'" ;
          valueString += ", '" + hireDate + "'" ;
          valueString += ");" ;

          System.out.println( "About to execute: " + valueString ) ;
          stmt.executeUpdate( valueString ) ;

        stmt.close() ;
        conn.close() ;

    }

    catch (SQLException ex ) {
        System.out.println("SQLException: " + ex.getMessage()) ;
        System.out.println("SQLState: " + ex.getSQLState()) ;
        System.out.println("VendorError: " + ex.getErrorCode()) ;
    }
  }
}

-------------------------------------------
```

(Script File)

Run jdbcFunction2.java in Terminal:

```
lcocoxs-MacBook-Air:Documents lcocox$ javac jdbcFunction2.java
lcocoxs-MacBook-Air:Documents lcocox$ java jdbcFunction2
Mon Mar 07 22:34:55 MST 2016 WARN: Establishing SSL connection without
server's identity verification is not recommended. According to MySQL
5.5.45+, 5.6.26+ and 5.7.6+ requirements SSL connection must be
established by default if explicit option isn't set. For compliance
with existing applications not using SSL the verifyServerCertificate
property is set to 'false'. You need either to explicitly disable SSL
by setting useSSL=false, or set useSSL=true and provide truststore for
server certificate verification.
Enter (Emp_id, Name, Gender, Age, HireDate) of an Employee you wish to
enter:
1002 Vince1 M 25 2016-03-07
You entered: 1002 Vince1 M 25 2016-03-07
About to execute: insert into Employee values('1002', 'Vince1', 'M',
'25', '2016-03-07');
```

Results in MySQL:

(Before I ran jdbcFunction2.java:)

```
mysql> select * from Employee where Emp_id > 990;
+--------+------------+--------+-----+------------+
| Emp_id | Name       | Gender | Age | HireDate   |
+--------+------------+--------+-----+------------+
|    991 | Christ991  | F      |  48 | 2003-01-18 |
|    992 | Christ992  | F      |  46 | 2002-11-28 |
|    993 | Christ993  | M      |  41 | 1990-06-15 |
|    994 | Christ994  | M      |  23 | 1998-01-04 |
|    995 | Christ995  | F      |  47 | 1999-11-18 |
|    996 | Christ996  | M      |  39 | 2006-05-03 |
|    997 | Christ997  | F      |  29 | 2010-09-11 |
|    998 | Christ998  | F      |  31 | 2004-05-23 |
|    999 | Christ999  | M      |  32 | 1994-07-17 |
|   1000 | Christ1000 | F      |  34 | 2002-01-02 |
|   1001 | Gloria1    | F      |  24 | 2016-03-11 |
+--------+------------+--------+-----+------------+
11 rows in set (0.00 sec)
```

(After I ran jdbcFunction2.java:)

```
mysql> select * from Employee where Emp_id > 990;
+--------+------------+--------+-----+------------+
```

Emp_id	Name	Gender	Age	HireDate
991	Christ991	F	48	2003-01-18
992	Christ992	F	46	2002-11-28
993	Christ993	M	41	1990-06-15
994	Christ994	M	23	1998-01-04
995	Christ995	F	47	1999-11-18
996	Christ996	M	39	2006-05-03
997	Christ997	F	29	2010-09-11
998	Christ998	F	31	2004-05-23
999	Christ999	M	32	1994-07-17
1000	Christ1000	F	34	2002-01-02
1001	Gloria1	F	24	2016-03-11
1002	Vince1	M	25	2016-03-07

12 rows in set (0.00 sec)

****