How to retrieve a row from MySQL using JDBC

Contents

Project Description

- This JDBC example demonstrates how to retrieve a row from a database table.
- We use an ‘Employee’ table created in MySQL and retrieve a particular employee record details.
- We use **Singleton** (for making database connection), **Data Access Object (DAO)**, **Transfer Object (TO) patterns**.

Prerequisites

Before proceeding with this example, refer this page which gives an overview of the following concepts;

- How to configure JDBC driver in Eclipse
- Various design patterns used in this example like DAO, TO and Singleton pattern,
- The program control flow logic,
- Tables used in JDBC examples,
- Various ways of accessing database tables such as Eclipse Data Source Explorer, MySQL command prompt, phpMyAdmin, etc.

Environment Used

- JDK 6 (Java SE 6)
- Eclipse Indigo IDE for Java EE Developers (3.7.1)
- Mysql-5.5
- MySQL Connector/J 5.1 JAR file

Create table in MySQL

This example uses one table Employee and the description of the table is shown below.

‘Employee’ table

Field	Type	Key	Extra
emp_id	int	Primary Key	auto_increment
emp_name	varchar(50)		
dob	date		
salary	double		
dept_id	int	Foreign key references department(dept_id)	

Create Java Project

Open Eclipse IDE and create a new Java project and name it as **SelectQuery**.

Employee.java (Transfer Object pattern)

In src folder, create a new **package** and name it as **com.theopentutorials.jdbc.to**. Create new class in this package and name it as **Employee**.

```
package com.theopentutorials.jdbc.to;
import java.util.Date;

public class Employee {
    private int empId;
    private String empName;
    private Date dob;
    private double salary;
    private int deptId;

    public int getEmpId() {
        return empId;
    }
    public void setEmpId(int empId) {
        this.empId = empId;
    }
    public String getEmpName() {
        return empName;
    }
    public void setEmpName(String empName) {
        this.empName = empName;
    }
    public Date getDob() {
        return dob;
    }
    public void setDob(Date dob) {
        this.dob = dob;
    }
    public double getSalary() {
        return salary;
    }
```

```
        public void setSalary(double salary) {
            this.salary = salary;
        }
        public void setDeptId(int deptId) {
            this.deptId = deptId;
        }
        public int getDeptId() {
            return deptId;
        }

        //toString()
    }
```

ConnectionFactory.java (singleton pattern)

Create a new class in src folder with the package name as **com.theopentutorials.jdbc.db** and class name as **ConnectionFactory** and click Finish.
Copy the following code:

```
package com.theopentutorials.jdbc.db;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class ConnectionFactory {
    //static reference to itself
    private static ConnectionFactory instance = new ConnectionFactory();
    public static final String URL = "jdbc:mysql://localhost/jdbcdb";
    public static final String USER = "YOUR_DATABASE_USERNAME";
    public static final String PASSWORD = "YOUR_DATABASE_PASSWORD";
    public static final String DRIVER_CLASS = "com.mysql.jdbc.Driver";

    //private constructor
    private ConnectionFactory() {
        try {
            Class.forName(DRIVER_CLASS);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    private Connection createConnection() {
        Connection connection = null;
        try {
            connection = DriverManager.getConnection(URL, USER,
PASSWORD);
        } catch (SQLException e) {
            System.out.println("ERROR: Unable to Connect to Database.");
        }
        return connection;
    }

    public static Connection getConnection() {
        return instance.createConnection();
    }
}
```

Fill in the username and password for your database and enter your database name in the URL string.

DbUtil.java

Create a new class in src folder with the package name as **com.theopentutorials.jdbc.db** and class name as "**DbUtil**" and click Finish.
Copy the following code:

```
package com.theopentutorials.jdbc.db;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class DbUtil {

    public static void close(Connection connection) {
        if (connection != null) {
            try {
                connection.close();
            } catch (SQLException e) {
                /*Ignore*/
            }
        }
    }

    public static void close(Statement statement) {
        if (statement != null) {
            try {
                statement.close();
            } catch (SQLException e) {
                /*Ignore*/
            }
        }
    }

    public static void close(ResultSet resultSet) {
        if (resultSet != null) {
            try {
                resultSet.close();
            } catch (SQLException e) {
                /*Ignore*/
            }
        }
    }
}
```

EmployeeDAO.java (Data Access Object (DAO) pattern)

Create a new class in src folder with the package name as **com.theopentutorials.jdbc.dao** and class name as **EmployeeDAO** and click Finish.
Copy the following code:

```
package com.theopentutorials.jdbc.dao;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import com.theopentutorials.jdbc.db.ConnectionFactory;
import com.theopentutorials.jdbc.db.DbUtil;
import com.theopentutorials.jdbc.to.Employee;

public class EmployeeDAO {
    private Connection connection;
    private Statement statement;

    public EmployeeDAO() { }

    public Employee getEmployee(int employeeId) throws SQLException {
        String query = "SELECT * FROM employee WHERE emp_id=" +
employeeId;
        ResultSet rs = null;
        Employee employee = null;
        try {
            connection = ConnectionFactory.getConnection();
            statement = connection.createStatement();
            rs = statement.executeQuery(query);
            if (rs.next()) {
                employee = new Employee();
                employee.setEmpId(rs.getInt("emp_id"));
                employee.setEmpName(rs.getString("emp_name"));
                employee.setDob(rs.getDate("dob"));
                employee.setSalary(rs.getDouble("salary"));
                employee.setDeptId((rs.getInt("dept_id")));
            }
        } finally {
            DbUtil.close(rs);
            DbUtil.close(statement);
            DbUtil.close(connection);
        }
        return employee;
    }
}
```

This class defines a method getEmployee(int employeeId) which retrieves a particular row with the given emp_id. This method reads data from database ResultSet and stores it in employee object and returns this object to the caller.

SelectDemo.java

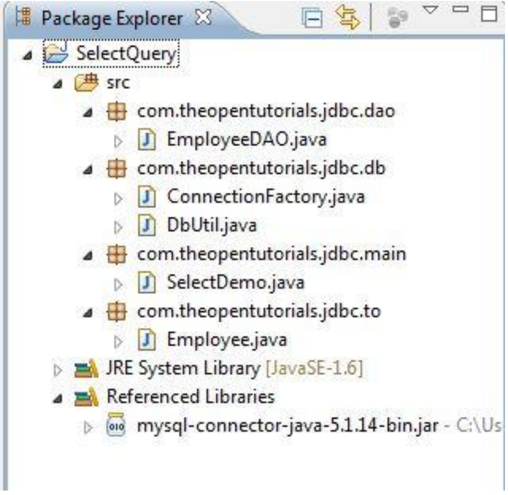
This is the Java Application client class with main() method which calls the method in DAO class passing employee id and displays the result to the user.

Create a new class in src folder with the package name as **com.theopentutorials.jdbc.main** and class name as “**SelectDemo**” and click Finish.
Copy the following code:

```
package com.theopentutorials.jdbc.main;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import com.theopentutorials.jdbc.dao.EmployeeDAO;
import com.theopentutorials.jdbc.to.Employee;

public class SelectDemo {
    public static void main(String[] args) {
        //1. Get Employee
        getEmployee();
    }

    private static void getEmployee() {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        System.out.println("Enter the EmployeeID:");

        try {
            int employeeId = Integer.parseInt(br.readLine());
            EmployeeDAO empDao = new EmployeeDAO();
            Employee employee = empDao.getEmployee(employeeId);
            if(employee != null)
                displayEmployee(employee);
            else
                System.out.println("No Employee with Id: " +
employeeId);
        } catch (NumberFormatException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private static void displayEmployee(Employee employee) {
        System.out.println("Employee ID:" + employee.getEmpId());
        System.out.println("Employee Name:" + employee.getEmpName());

        SimpleDateFormat format = new SimpleDateFormat("dd/MM/yyyy");
        String dob = format.format(employee.getDob());

        System.out.println("DOB:" + dob);
        System.out.println("Salary:" + employee.getSalary());
        System.out.println("Department ID:" + employee.getDeptId());
        System.out.println();
    }
}
```

Folder Structure:

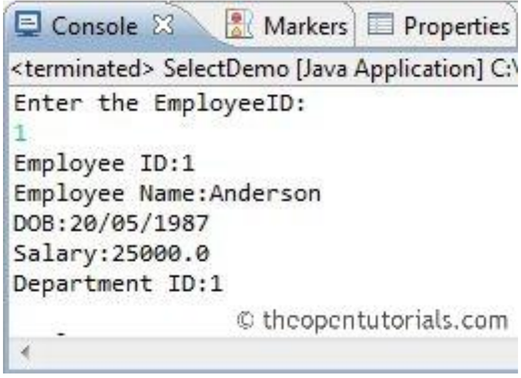
Complete folder structure of this project is shown below.



```
SelectQuery
  src
    com.theopentutorials.jdbc.dao
      EmployeeDAO.java
    com.theopentutorials.jdbc.db
      ConnectionFactory.java
      DbUtil.java
    com.theopentutorials.jdbc.main
      SelectDemo.java
    com.theopentutorials.jdbc.to
      Employee.java
  JRE System Library [JavaSE-1.6]
  Referenced Libraries
    mysql-connector-java-5.1.14-bin.jar - C:\Us
```

Output

Run this SelectDemo class to get the output as shown below.

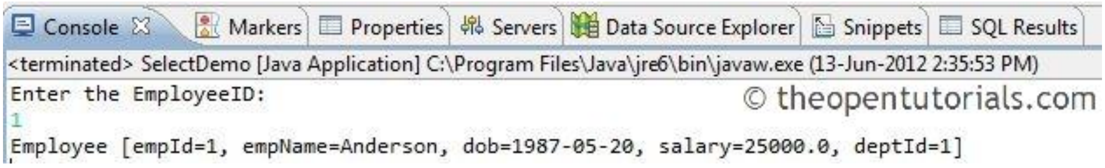


```
<terminated> SelectDemo [Java Application] C:\
Enter the EmployeeID:
1
Employee ID:1
Employee Name:Anderson
DOB:20/05/1987
Salary:25000.0
Department ID:1
```

Alternate method for printing the employee details is to generate the toString() method **in Employee.java class**

public String toString() {
return “Employee [empId=" + empId + ", empName=" + empName + ", dob="
+ dob + ", salary=" + salary + ", deptId=" + deptId + "]“;
}

and replace this line **displayEmployee(employee);** with this **System.out.println(employee);**



```
<terminated> SelectDemo [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe (13-Jun-2012 2:35:53 PM)
Enter the EmployeeID:
1
Employee [empId=1, empName=Anderson, dob=1987-05-20, salary=25000.0, deptId=1]
```