This is Google's cache of https://ibytecode.com/blog/how-to-retrieve-all-rows-from-mysql-table-using-jdbc/. It is a snapshot of the page as it appeared on 8 Dec 2019 01:21:56 GMT. The current page could have changed in the meantime. Learn more.

Full version Text-only version View source

Tip: To quickly find your search term on this page, press Ctrl+F or ⌘-F (Mac) and use the find bar.

How to retrieve all rows from MySQL table using JDBC

By ***Nithya Vasudevan***

October 14, 2016

PROJECT DESCRIPTION

- This JDBC example explains how to retrieve all rows from a MySQL database table.
- We use an 'Employee' table created in MySQL and retrieve all employee 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) (To install JDK refer this link in Windows, Ubuntu)
- Eclipse Indigo IDE for Java EE Developers (3.7.1) (To install Eclipse, refer this link)
- Mysql-5.5 (To install MySQL read this page)
- MySQL Connector/J 5.1 JAR file

CREATE TABLE IN MYSQL

This example uses an Employee table 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 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();
                }
        }

        public 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 datadase 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 java.util.ArrayList;
import java.util.List;
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 List<Employee> getEmployees() throws SQLException {
                String query = "SELECT * FROM employee";
                List<Employee> list = new ArrayList<Employee>();
                Employee employee = null;
                ResultSet rs = null;
                try {
                        connection = ConnectionFactory.getConnection();
                        statement = connection.createStatement();
                        rs = statement.executeQuery(query);
                        while (rs.next()) {
                                employee = new Employee();
                                /*Retrieve one employee details
                                and store it in employee object*/
                                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")));

                                //add each employee to the list
                                list.add(employee);
                        }
                } finally {
                        DbUtil.close(rs);
                        DbUtil.close(statement);
                        DbUtil.close(connection);
                }
                return list;
        }
}
```

This class defines a method getEmployees() which retrieves all rows from Employee table. This method reads data from database ResultSet, stores it in employee object, add this object to the list and returns the list of employees to the caller.

SELECTDEMO.JAVA

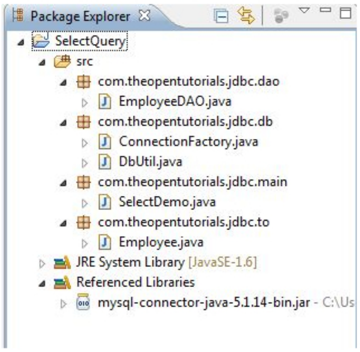
This is the Java Application client class with main() method which calls the method in DAO class and displays the result to the user.

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) {
                //Get all employees
                getEmployees();
        }

        private static void getEmployees() {
                EmployeeDAO empDao = new EmployeeDAO();
                List<Employee> employees;
                try {
                        employees = empDao.getEmployees();
                        for (Employee employee : employees) {
                                displayEmployee(employee);
                displayEmployee(employee);
                //System.out.println(employee);
            }
        } 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

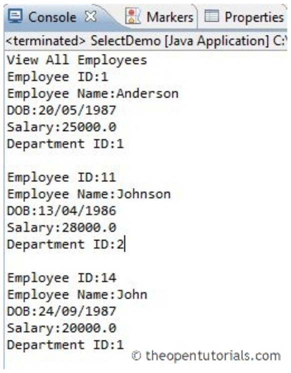
Complete folder structure of this project is shown below.



```
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.

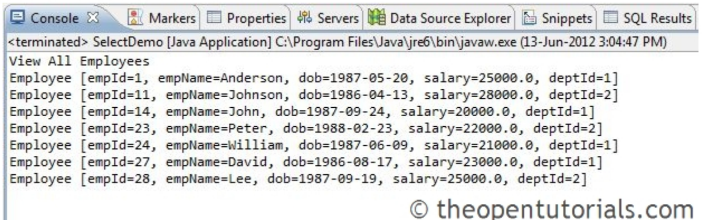


```
<terminated> SelectDemo [Java Application] C:\
View All Employees
Employee ID:1
Employee Name:Anderson
DOB:20/05/1987
Salary:25000.0
Department ID:1

Employee ID:11
Employee Name:Johnson
DOB:13/04/1986
Salary:28000.0
Department ID:2

Employee ID:14
Employee Name:John
DOB:24/09/1987
Salary:20000.0
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 3:04:47 PM)
View All Employees
Employee [empId=1, empName=Anderson, dob=1987-05-20, salary=25000.0, deptId=1]
Employee [empId=11, empName=Johnson, dob=1986-04-13, salary=28000.0, deptId=2]
Employee [empId=14, empName=John, dob=1987-09-24, salary=20000.0, deptId=1]
Employee [empId=23, empName=Peter, dob=1988-02-23, salary=22000.0, deptId=2]
Employee [empId=24, empName=William, dob=1987-06-09, salary=21000.0, deptId=1]
Employee [empId=27, empName=David, dob=1986-08-17, salary=23000.0, deptId=1]
Employee [empId=28, empName=Lee, dob=1987-09-19, salary=25000.0, deptId=2]
```

DAO Pattern Java | Jdbc Examples | Jdbc Mysql Example | Jdbc Read All Rows From Table
Jdbc Select All Records | Jdbc Select Query | Jdbc Select Statement

About

Nithya Vasudevan

✉

COMMENTS

Niemi February 20, 2019 Reply

forgot to add “import java.util.List;” in SelectDemo.

LEAVE A COMMENT

LEAVE YOUR COMMENT

Name (Required) | Email (Required) | Website

□ Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

This site uses Akismet to reduce spam. Learn how your comment data is processed.

PRODUCTS

- Visitor Master
- Garage Master
- Builder Master
- Call Service Management
- Invoice

SERVICES

- Mobile Solutions
- Cloud Deployment Services
- Web Solutions
- Business Technology Consulting
- Graphic Designs
- Search Engine Optimization
- Social Media Marketing
- Software Training

STAY IN TOUCH

iByteCode Technologies
B10, Mogappair Industrial
Estate, Mogappair East,
Chennai, INDIA
Email: contact@ibytecode.com
Phone: +91 44 4385 1836

f in

© 2019 iByteCode. ALL RIGHTS RESERVED.