



MySQL e Java JDBC - Tutorial

Lars Vogel (c) 2009 - 2020 vogella GmbH

Versão 1.4,

29.09.2016

TABELA DE CONTEÚDOS

1. Conexão ao banco de dados com Java
2. Introdução ao MySQL
3. MySQL JDBC driver
4. Exercício: crie banco de dados de exemplo
5. Java JDBC
6. Links e Literatura

MySQL e Java JDBC. Este tutorial descreve como usar o Java JDBC para se conectar ao MySQL e realizar consultas SQL, inserções de banco de dados e exclusões.

1. Conexão ao banco de dados com Java

A interface para acessar bancos de dados relacionais do Java é a Java Database Connectivity (JDBC). Via JDBC você cria uma conexão com o banco de dados, emite consultas de banco de dados e atualiza, bem como recebe os resultados.

O JDBC fornece uma interface que permite realizar operações SQL independentemente da instância do banco de dados usado. Para usar o JDBC, você requer a implementação específica do banco de dados do driver JDBC.

2. Introdução ao MySQL

Para aprender a instalar e usar o MySQL, consulte MySQL - Tutorial.

A descrição a seguir assumirá que você instalou com sucesso o MySQL e sabe como acessar o MySQL através da linha de comando.

3. MySQL JDBC driver

Para se conectar ao MySQL de Java, você tem que usar o driver JDBC do MySQL. O driver MySQL JDBC é chamado de MySQL Connector/J. Você encontra o driver JDBC mais recente sob a seguinte URL: <http://dev.mysql.com/downloads/connector/j>.

O download contém um arquivo que exigimos mais tarde.JAR

4. Exercício: crie banco de dados de exemplo

Neste exercício você cria um novo banco de dados, um novo usuário e uma tabela de exemplo. Para isso, conecte-se ao servidor MySQL através do cliente da linha de comando.mysql

Crie um novo banco de dados chamado feedback e comece a usá-lo com o seguinte comando.

```
create database feedback;
```

```
use feedback;
```

Crie um usuário com o seguinte comando.

```
CREATE USER sqluser IDENTIFIED BY 'sqluserpw';
```

```
grant usage on *.* to sqluser@localhost identified by 'sqluserpw';
```

```
grant all privileges on feedback.* to sqluser@localhost;
```

Agora, crie uma tabela de banco de dados de exemplo com conteúdo de exemplo através da seguinte instrução SQL.

```
CREATE TABLE comments (  
    id INT NOT NULL AUTO_INCREMENT,  
    MYUSER VARCHAR(30) NOT NULL,  
    EMAIL VARCHAR(30),  
    WEBPAGE VARCHAR(100) NOT NULL,  
    DATUM DATE NOT NULL,  
    SUMMARY VARCHAR(40) NOT NULL,  
    COMMENTS VARCHAR(400) NOT NULL,  
    PRIMARY KEY (ID)  
);
```

```
INSERT INTO comments values (default, 'lars', 'myemail@gmail.com', 'https://www.vogella.com/',  
'2009-09-14 10:33:11', 'Summary', 'My first comment' );
```

5. Java JDBC

Create a Java project and a package called de.vogella.mysql.first.

Create a folder and copy the JDBC driver into this folder. Add the JDBC driver to your classpath. See Adding jars to the classpath for details.lib

Create the following class to connect to the MySQL database and perform queries, inserts and deletes. It also prints the metadata (table name, column names) of a query result.

```
package de.vogella.mysql.first;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Date;

public class MySQLAccess {

    private Connection connect = null;
    private Statement statement = null;
    private PreparedStatement preparedStatement = null;
    private ResultSet resultSet = null;

    public void readDataBase() throws Exception {
        try {
            // This will load the MySQL driver, each DB has its own driver
            Class.forName("com.mysql.jdbc.Driver");

            // Setup the connection with the DB
            connect = DriverManager
                .getConnection("jdbc:mysql://localhost/feedback?"
                    + "user=sqluser&password=sqluserpw");

            // Statements allow to issue SQL queries to the database
            statement = connect.createStatement();

            // Result set get the result of the SQL query
```

Dia 27 – Conexão com o Banco de dados

```
resultSet = statement
    .executeQuery("select * from feedback.comments");
writeResultSet(resultSet);

// PreparedStatements can use variables and are more efficient
preparedStatement = connect
    .prepareStatement("insert into feedback.comments values (default, ?,
?, ?, ?, ?, ?)");
// "myuser, webpage, datum, summary, COMMENTS from feedback.comments");
// Parameters start with 1
preparedStatement.setString(1, "Test");
preparedStatement.setString(2, "TestEmail");
preparedStatement.setString(3, "TestWebpage");
preparedStatement.setDate(4, new java.sql.Date(2009, 12, 11));
preparedStatement.setString(5, "TestSummary");
preparedStatement.setString(6, "TestComment");
preparedStatement.executeUpdate();

preparedStatement = connect
    .prepareStatement("SELECT myuser, webpage, datum, summary,
COMMENTS from feedback.comments");
resultSet = preparedStatement.executeQuery();
writeResultSet(resultSet);

// Remove again the insert comment
preparedStatement = connect
    .prepareStatement("delete from feedback.comments where myuser= ? ;");
preparedStatement.setString(1, "Test");
preparedStatement.executeUpdate();
```

Dia 27 – Conexão com o Banco de dados

```
resultSet = statement
.executeQuery("select * from feedback.comments");
writeMetaData(resultSet);

} catch (Exception e) {
    throw e;
} finally {
    close();
}

}

private void writeMetaData(ResultSet resultSet) throws SQLException {
    // Now get some metadata from the database
    // Result set get the result of the SQL query

    System.out.println("The columns in the table are: ");

    System.out.println("Table: " + resultSet.getMetaData().getTableName(1));
    for (int i = 1; i <= resultSet.getMetaData().getColumnCount(); i++){
        System.out.println("Column " + i + " " +
resultSet.getMetaData().getColumnName(i));
    }
}

private void writeResultSet(ResultSet resultSet) throws SQLException {
    // ResultSet is initially before the first data set
    while (resultSet.next()) {
        // It is possible to get the columns via name
```

Dia 27 – Conexão com o Banco de dados

```
// also possible to get the columns via the column number
// which starts at 1
// e.g. resultSet.getString(2);
String user = resultSet.getString("myuser");
String website = resultSet.getString("webpage");
String summary = resultSet.getString("summary");
Date date = resultSet.getDate("datum");
String comment = resultSet.getString("comments");
System.out.println("User: " + user);
System.out.println("Website: " + website);
System.out.println("summary: " + summary);
System.out.println("Date: " + date);
System.out.println("Comment: " + comment);
    }
}
```

// You need to close the resultSet

```
private void close() {
    try {
        if (resultSet != null) {
            resultSet.close();
        }

        if (statement != null) {
            statement.close();
        }

        if (connect != null) {
            connect.close();
        }
    }
}
```

```
        }  
    } catch (Exception e) {  
  
    }  
}  
  
}
```

Crie o seguinte programa principal para testar sua classe.

```
package de.vogella.mysql.first.test;  
  
import de.vogella.mysql.first.MySQLAccess;  
  
public class Main {  
    public static void main(String[] args) throws Exception {  
        MySQLAccess dao = new MySQLAccess();  
        dao.readDataBase();  
    }  
  
}
```

Video

<https://www.youtube.com/watch?v=ZonDIJB4b-Q>

Exercicio

1) Quais os principais 3 importes para trabalhar com Banco de Dados:

```
a) import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
b) import java.sql.Connection;  
import java.sql.util;
```



```
import java.sql.SQLException;  
c) import java.sql.Strings;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
d) import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLEx;
```

2)Qual desses termos não faz parte de uma conexão de banco de dados:

- A)HostName = null;
- B>Password = null;
- C)dataBasePay = null;**
- D)jdbcDriver = null;

3) Qual desses termos faz parte de uma conexão de banco de dados

- A)url = null;**
- B)String = null;
- C)UserDatabase = null;
- D) Nem uma delas

Prova

1)O que é JDBC?

- a) É o compilador do SQL server.
- b)É uma API que reúne conjuntos de classes e interfaces escritas na linguagem Java na qual possibilita se conectar através de um driver específico do banco de dados desejado.**
- c)JDBC é o servidor do MYSQL que recebe os dados de uma determinada linguagem de programação
- d)Nem uma das alternativas.

2)O que o Pacote java.sql oferece no java

- a) Esse pacote oferece a biblioteca Java o acesso e processamento de dados em uma fonte de dado.**
- b) Import todas as ferramentas do SQL e MYSQL

Dia 27 – Conexão com o Banco de dados

c) Este pacote oferece biblioteca SQL o acesso e processamento de dados em uma fonte de dado

d)Nem uma das

3) Qual desses termos faz parte de uma conexão de banco de dados

A)url = null;

B)String = null;

C)UserDatabase = null;

D) Nem uma delas