

# <第一章>对原生态jdbc程序中问题的总结



作者：杨旭东

时间：8/14/2017 9:50:26 PM

## 1. 需求

使用jdbc查询mysql数据库中用户表的记录

## 2. 程序代码

```
package com.yxd.jdbc;
import java.sql.*;
/**
 * Description 通过传统的jdbc程序，总结其中的问题
 * Author 杨旭东
 * Date 6/14/2017 8:52:54 PM
 * Version 1.0
 */
public class JdbcTest {
    public static void main(String[] args) {

        // 数据库连接
        Connection connection = null;
        // 预编译的Statement，使用预编译的Statement可以提高数据库性能
        PreparedStatement preparedStatement = null;
        // 数据结果集
        ResultSet resultSet = null;

        try {
            // 加载数据库驱动
            Class.forName("com.mysql.jdbc.Driver");

            // 通过驱动管理类获取数据库连接
            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo?charset=utf-8", "root", "root");
            // 定义sql语句，? 表示占位符
            String sql = "select * from user where id=?";
            // 获得预处理Statement
            preparedStatement = connection.prepareStatement(sql);
            // 设置参数，第一个参数为sql语句中参数的序号（从1开始），第二个参数为设置的参数值
            preparedStatement.setObject(1, 1);
            // 向数据库发出sql执行查询，查询出结果集
            resultSet = preparedStatement.executeQuery();
            // 遍历查询结果集
            while (resultSet.next()) {
                System.out.println(resultSet.getString("id") + " " + resultSet.getString("username"));
            }

        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                resultSet.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
            try {
                preparedStatement.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
    }  
    try {  
        connection.close();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}  
}  
}
```

### 3. 问题总结

---

1. 数据库连接，使用时创建，不使用立即释放，对数据库进行频繁开启和关闭，造成数据库资源浪费，影响了数据库的性能。  
设想：使用数据库连接池管理数据库连接。
2. 将sql语句硬编码到java代码中，如果sql语句要修改，要重新编译java代码，不利于系统的维护。  
设想：将sql语句配置到xml配置文件中，即使sql变化，不需要对java代码进行重新编译。
3. 向PreparedStatement中设置参数，对占位符位置和设置参数值，硬编码到java代码中，不利于系统维护。  
设想：将sql语句及占位符和参数全部配置到xml中。
4. 从ResultSet中遍历结果集数据时，存在硬编码，将获取表的字段进行硬编码，不利于系统维护。  
设想：将查询的结果集自动映射成java对象