03 PreparedStatement 执行sql的对象

JDBC

一：SQL 注入问题

> 1、当我们输入以下密码，我们发现我们账号和密码都不对竟然登录成功了

```
请输入用户名：
amdin
请输入密码：
'amdin' or '1'='1
拼接的sql语句如下：
select * from user where name='amdin' and password='amdin' or '1'='1'
```

> 2、由于有SQL注入的问题，那么PreparedStatement的优势就来了

二：PreparedStatement 接口

> 1、继承结构与作用

概述 软件包 类 使用 树 已过时 索引 帮助

Java™ Platfo
Standard Ed.

上一个类 下一个类 框架 无框架 所有类
摘要： 嵌套 | 字段 | 构造方法 | 方法 详细信息： 字段 | 构造方法 | 方法

java.sql
接口 PreparedStatement

所有超级接口：
Statement, Wrapper

所有已知子接口：
CallableStatement

public interface PreparedStatement
extends Statement

表示预编译的 SQL 语句的对象。

SQL 语句被预编译并存储在 PreparedStatement 对象中。然后可以使用此对象多次高效地执行该语句。

> 2、因为有预先编译的功能，提高 SQL 的执行效率
>
> 3、预编译的SQL：参数使用?作为占位符
>
> 4、可以有效的防止 SQL 注入的问题，安全性更高

2、接口中的方法

接口中的方法	**描述**
PreparedStatement prepareStatement(String sql)	指定预编译的 SQL 语句，SQL 语句中使用占位符?创建一个语句对象
int executeUpdate()	执行 DML，增删改的操作，返回影响的行数。
ResultSet executeQuery()	执行 DQL，查询的操作，返回结果集
void setDouble(int parameterIndex, double x)	将指定参数设置为给定 Java double 值。
void setFloat(int parameterIndex, float x)	将指定参数设置为给定 Java REAL 值。
void setInt(int parameterIndex, int x)	将指定参数设置为给定 Java int 值。
void setLong(int parameterIndex, long x)	将指定参数设置为给定 Java long 值。
void setObject(int parameterIndex, Object x)	使用给定对象设置指定参数的值。
void setString(int parameterIndex, String x)	将指定参数设置为给定 Java String 值。

3、案例

> 1、使用预编译进行查询SQL语句

```
1.      @Test
2.      public void test10() throws Exception {
```

```
            //连接数据库判断是否登录成功
            Connection conn = null;
            PreparedStatement pstmt =  null;
            ResultSet rs = null;
            //1.获取连接
            try {
                conn =  JDBCUtils.getConnection();
                String sql = "select * from t_cost where costName = ? and costMark = ?";
                pstmt = conn.prepareStatement(sql);
                //给?赋值
                pstmt.setString(1,"轮船票");
                pstmt.setString(2,"0");
                //4.执行查询,不需要传递sql
                rs = pstmt.executeQuery();
                ArrayList arr=new ArrayList<Cost>();
                // 让游标向下移动一行
                while(rs.next()){
                // 获取数据
                Cost cost=  new Cost();
                cost.setCostId(rs.getInt(1));
                cost.setCostName(rs.getString(2));
                cost.setCostMark(rs.getString("costMark"));
                cost.setCostDesc(rs.getString("costName"));
                arr.add(cost);
                }
                System.out.println(arr);

            } catch (SQLException e) {
                e.printStackTrace();
            }finally {
                JDBCUtils.close(rs,pstmt,conn);
            }
    }
```

4、PreparedSatement 的好处

> 1、prepareStatement()会先将 SQL 语句发送给数据库预编译
>
> 2、PreparedStatement 会引用着预编译后的结果
>
> 3、可以多次传入不同的参数给 PreparedStatement 对象并执行

> 4、减少 SQL 编译次数，提高效率
>
> 5、安全性更高，没有 SQL 注入的隐患
>
> 6、提高了程序的可读性