



JDBC数据库连接技术

武永亮

讲授思路

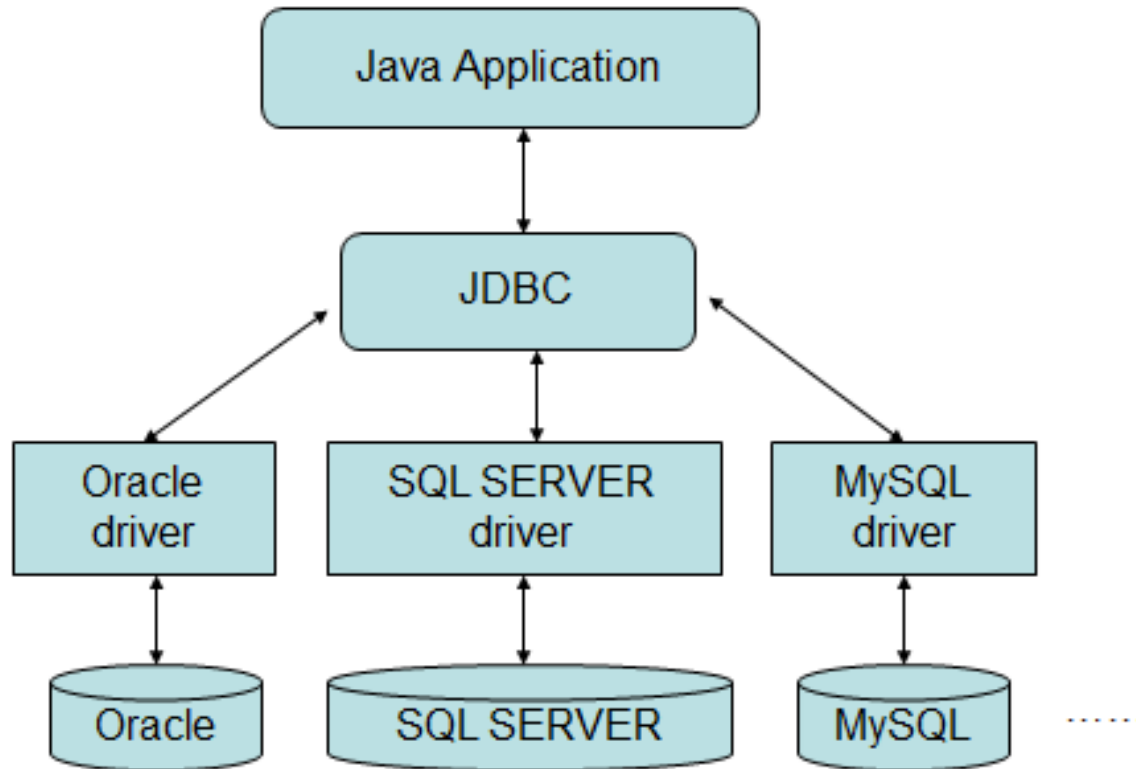
- JDBC的概念和类型
- JDBC中主要的类和接口
- 用JDBC进行数据库连接

JDBC的概念和类型

- JDBC是Java Database Connectivity的简称
- 是由一组Java语言编写的类和接口组成，是一种用于执行SQL语句的规范
- 官方网址：
<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136101.html>
- JDBC API 提供两类主要接口：
 - 一是面向开发人员的java.sql程序包，使得Java程序员能够进行数据库连接，执行SQL查询，并得到结果集合。
 - 一是面向底层数据库厂商的JDBC Drivers。

JDBC的概念和类型

- JDBC是Java Database Connectivity的简称
- 是由一组Java语言编写的类和接口组成，是一种用于执行SQL语句的规范



JDBC中主要的类和接口

- Driver接口
- DriverManager类
- Connection类
- Statement类
- PreparedStatement类
- ResultSet类

Driver

- Driver接口：每个JDBC数据库驱动程序都会提供Driver接口的具体实现
- 如果想连接数据库，必须先加载数据库厂商提供的数据库驱动程序
- 不同类型的JDBC数据库驱动程序在编程时的加载方法也不同。
- JDBC驱动程序加载方法：
 - `Class.forName("jdbcdriver_classname").newInstance();`
 - 对于MySQL数据
 - `jdbcdriver_classname=com.mysql.jdbc.Driver。`

DriverManager

- DriverManager是 JDBC 的管理层，管理一组JDBC驱动程序的基本服务。
- DriverManager类的主要作用是跟踪可用的驱动程序，并在数据库和相应驱动程序之间建立连接。
- 调用Class.forName()方法将显式地加载驱动程序类。
- 调用DriverManager.getConnection()方法将建立与数据库的连接，得到与数据库连接的Connection对象。

Connection

- Connection类是JDBC规范中的最核心的类。
 - Statement对象和ResultSet对象等都直接或者间接的来源于它
- Connection对象表示与特定数据库的连接(会话)。
 - 得到Statement对象
 - createStatement()
 - prepareStatement(String sql)
 - prepareCall(String sql)
 - 为了保证数据库事务的原子性，Connection可以设置手动提交事务

```
conn.setAutoCommit(false);
try {
    Statement stmt=conn.createStatement();
    stmt.execute (sql);
    conn.commit();
} catch (SQLException e1) {
    conn.rollback();//事务回滚
}
```


Statement

- Statement是向数据库提交SQL语句并返回相应结果的工具。语句可以是SQL查询、修改、插入或者删除
- PreparedStatement接口
 - 防止SQL注入攻击(使用占位符 “?”)
 - 提高SQL的执行性能(在执行之前有预处理)
 - 避免使用SQL方言
 - 提高JDBC中有关SQL代码的可读性
- CallableStatement接口用于执行 SQL 存储过程的接口

PreparedStatement

- PreparedStatement类：是Statement的子接口
- 用PreparedStatement类效率会更高。使用PreparedStatement有很多优势，总结如下：
 - 防止SQL注入攻击(使用占位符 “?”)。
 - 提高SQL的执行性能(在执行之前有预处理)。
 - 避免使用SQL方言提高JDBC中有关SQL代码的可读性。
- PreparedStatement实例要通过Connection对象调用prepareStatement(String sql)方法获得。

PreparedStatement

- PreparedStatement中常用的一些方法：
 - `public ResultSetMetaData getMetaData()`
 - `public void setInt(int parameterIndex, int x)`
 - `public void setFloat(int parameterIndex, float x)`
 - `public void setString(int parameterIndex, String x)`
 - `public void setDate(int parameterIndex, Date x)`

ResultSet

- ResultSet表示数据库结果集的数据表，通常通过执行查询数据库的语句生成
- ResultSet 对象具有指向其当前数据行的光标
- ResultSet类型和特性

用JDBC进行数据库连接

- 从数据库中查询出员工的信息，并打印到控制台上。

```
Class.forName("com.mysql.jdbc.Driver");
String url =
"jdbc:mysql://127.0.0.1:3306/em?useUnicode=true&characterE
ncoding=UTF-8";
Connection conn = DriverManager.getConnection(url, "root", "");
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from employees");
while (rs.next()) {
    System.out.println(rs.getString("name"));
}
conn.close();
```

创建JDBC应用的步骤

- 1.加载数据库的驱动程序（前提首先引入数据库驱动的jar包）
 - `Class.forName("driverName");`
- 2.建立数据库连接
 - `String connStr = "jdbc:<JDBC驱动程序名:端口号>:数据源"`
 - `Connection`
`conn=DriverManager.getConnection (connStr);`
- 3.执行数据库操作SQL
 - `Statement stmt = conn.createStatement();`
 - `ResultSet rs = stmt.executeQuery(sql);`
- 4.关闭数据库连接
 - `conn.close();`

总结

- JDBC的概念和类型
- JDBC中主要的类和接口
- 用JDBC进行数据库连接



Thank You