

JDBC 操作数据库步骤

利用 JDBC 编写代码操作数据库主要分为六个步骤，如图 7-1 所示。

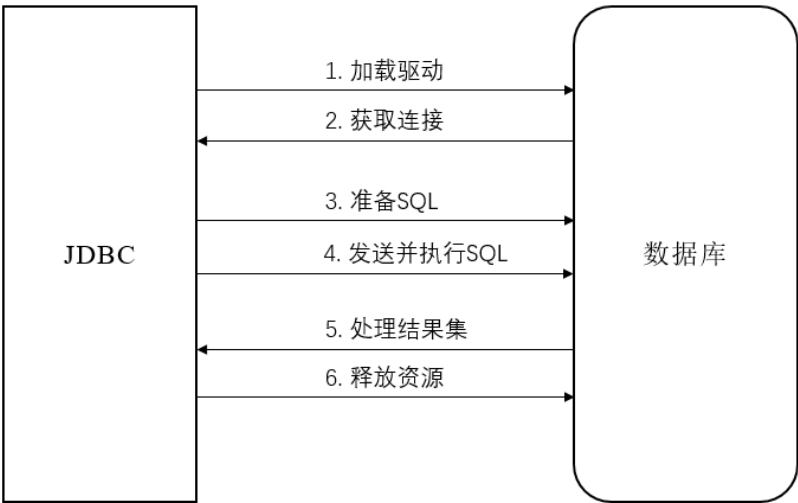


图 7-1 JDBC 操作数据库的步骤

1. 加载并注册数据库驱动

注册数据库驱动的具体方法有两种，示例代码如下。

```
DriverManager.registerDriver(driver);
Class.forName("DriverName");
```

2. 获取数据库连接

获取数据库连接的示例代码如下。

```
Connection conn = DriverManager.getConnection(
    String url,String user, Stringpwd);
```

从上述代码可以看出，`getConnection` 方法中有 3 个参数，它们分别表示连接数据库的 URL 地址、登录数据库的用户名和密码。以 MySQL 数据库为例，其 URL 地址的书写格式如下。

```
jdbc:mysql://hostname:port/databasename
```

上述代码中，`jdbc:mysql:`是固定的写法，`mysql` 指的是 MySQL 数据库，`hostname` 指的是主机的名称。如果数据库在本机中，`hostname` 可以为 `localhost` 或 `127.0.0.1`。如果要连接的数据库在其他主机上，`hostname` 为所要连接主机的 IP 地址。`port` 指的是连接数据库的端口号，MySQL 数据库端口号默认为 3306，而 `databasename` 指的是 MySQL 中相应数据库的名称。

3. 创建 Statement 对象，准备 SQL 语句

SQL 语句不包含参数时，使用基本的 Statement 对象，示例代码如下。

```
string sql="select * from users";  
Statement stmt = conn.createStatement(sql);
```

SQL 语句包含参数时，使用 PreparedStatement 对象，示例代码如下。

```
string sql =" select * from users where id = ?";  
PreparedStatement stmt =conn.prepareStatement(sql);  
stmt.setInt(1,5); //设置 sql 语句中第一个参数的值为 5
```

4. 使用 Statement 对象执行 SQL 语句

执行查询语句，使用 executeQuery 方法，返回代表结果集的 ResultSet 对象，示例代码如下。

```
ResultSet rs = stmt.executeQuery(sql);
```

执行 DML 语句，例如 INSERT、UPDATE 和 DELETE，使用 executeUpdate 方法，返回受 SQL 语句影响的行数，示例代码如下。

```
int affectedRows = stmt.executeUpdate(sql);
```

5. 处理 ResultSet 结果集

如果执行的 SQL 语句是查询语句，执行结果将返回一个 ResultSet 对象，该对象里保存了 SQL 语句查询的结果。程序可以通过操作该 ResultSet 对象来读取查询结果，示例代码如下。

```
string sql="select * from users";  
Statement stmt = conn.createStatement();  
ResultSet rs = stmt.executeQuery(sql);  
while (rs.next()){  
    int id = rs.getInt("id");  
    String name = rs.getString("name");  
    .....//省略具体业务处理代码  
}
```

6. 关闭连接，释放资源

每次操作数据库结束后都要关闭数据库连接，释放资源，包括关闭 ResultSet、Statement 和 Connection 等资源。