# (一)、JDBC

参考：

<http://www.admin10000.com/document/4071.html>

<http://www.journaldev.com/2489/jdbc-statement-vs-preparedstatement-sql-injection-example>

官方文档：<https://dev.mysql.com/doc/connector-j/5.1/en/>

jdbc源码

## 一、获取连接的过程

1 Class.forName(“”) , 类加载器会加载jdbc中实现的Driver类，执行static中的代码。 代码中会生成一个自己的 Driver对象，并注册到jdk中的DriverManager中。

2 调用DriverManger中的静态成员函数 getConnection(), 在函数中会调用注册driver对象的connect方法。从而获得连接。

## 二、使用PreparedStatement代替Statement

1 Statement 不安全，sql 注入攻击 。

而PreparedStatement能避免，因为会把所有的特殊字符转义。

2 Statement是静态写死的sql语句；而PreparedStatement能传入参数执行动态的查询

3 效率更高

4 返回值：FORWARD\_ONLY

## 三、PreparedStatement 的api

继承自Statement

### 1 单条api

1 boolean execute():

返回值：如果第一个结果是ResutlSet,则返回true； 否则返回false。

2 java.sql.ResultSet executeQuery()

不会返回null

3 int executeUpdate()

增、删、改都调用这个api

### 2 批量API

1 addBatch(String sql) + executeBatch() // 用于 增、删、改

2 addBatch() + executeBatch

### 3 其他

1 获取查询的结果

ResultSet rs = stmt.getResultSet();

2 获取update影响的条数

int getUpdateCount()（ 检查返回值-1， 如果没有更新成功，返回-1，还是0？？？）

## 四、事务管理

<http://wangqinqin.iteye.com/blog/547277>

在Connection配置中，默认为自动提交；但当我们进行批量操作时，最好使用手动提交；但一定要记得，当提交结束后，再设置为自动提交。因为这个链接可能其他地方也在用，防止带来不良影响。

con.setAutoCommit(**false**);//1,首先把Auto commit设置为false,不让它自动提交

con.commit();//2,进行手动提交（commit）

con.setAutoCommit(**true**);//3,提交完成后回复现场将Auto commit,还原为true,

同时在异常处理中，进行数据回滚和恢复默认设置。

1. con.rollback();//4,当异常发生执行catch中SQLException时，记得要rollback(回滚)；
2. System.out.println("插入失败，回滚！");
3. con.setAutoCommit(**true**);

## 五、资源关闭

参考：

<https://dev.mysql.com/doc/connector-j/5.1/en/connector-j-usagenotes-statements.html#connector-j-examples-execute-select>

和

<http://wangqinqin.iteye.com/blog/547277>

1 关闭查询返回的结果： ResultSet

2 关闭Statement

3 关闭连接， Connection

# (二)、Proxool连接池

参考：<http://www.csyor.com/518.html>

<http://blog.sina.com.cn/s/blog_7ffb8dd501019bon.html>

官网： <http://proxool.sourceforge.net/>

## 1 proxool同jdbc的关系

a mysql-jdbc是提供一个连接和对mysql操作的api；

而proxool 提供的是连接池，因此底层肯定调用的mysql-jdbc-connector

同C3P0一样的连接池技术。

1 依赖包：

**proxool.jar**

**proxool-cglib.jar**

b proxool

## 2 三种配置方式

1 把连接池的配置信息放入Properties

a 对url做了改变：增加了proxool.alias:driver:url

b 使用con=DriverManger.getConnetcion(url)获取连接。

2 Programmatically

a 对url做了改变，增加proxool.alias:driver:url

b 使用ProxoolFacade注册

ProxoolFacade.registerConnectionPool(url, info);

C DriverManager传入别名

connection = DriverManager.getConnection("proxool.test");

3 把连接池的配置信息放入xml中

1 首先配置并解析xml配置文件

JAXPConfigurator.configure("src/java-test/org/logicalcobwebs/proxool/configuration/test-no-ns.xml", false);

2 同上，DriverManager传入别名获取连接

## 3 连接池属性配置

# (三)、Spring中使用

# (四)、Servlet中使用

1 在web.xml中把proxool.xml配置文件路径**init-param中配置，**

**然后再程序中读取。**

**2 参考官网：直接在web.xml中配置并加载**

# (五)、ORM