# PreparedStatement接口

## SQL注入

由于没有对用户输入进行充分检查，而SQL又是拼接而成，在用户输入参数时，在参数中添加一些SQL 关键字，达到改变SQL运行结果的目的，也可以完成恶意攻击。

String sql = select \* from user where username ='' and password ='' ;

例如：

一、输入 username: 老李' or '1'='1 password 随意

select \* from user where username ='老李' or '1'='1' and password ='';

and 优先级 执行 高于 or

SQL注入原理是什么?

1.在输入时连接一个永远为真的一个值

2.使用mysql 中的 – *注释*

为什么PreparedStatement 可以防止SQL注入 ？

因为它对sql语句进行预编译。

## 解决SQL注入--预编译

使用PreparedStatement 取代 Statement

PreparedStatement 解决SQL注入原理，运行在SQL中参数以?占位符的方式表示

select \* from user where username = ? and password = ? ;

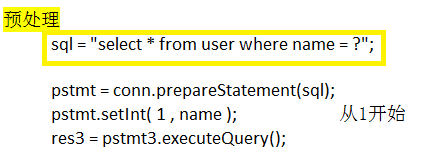
原理：

将带有?的SQL 发送给数据库完成编译 （不能执行的SQL 带有?的SQL 进行编译 叫做），在SQL编译后发现缺少两个参数PreparedStatement 可以将? 代替参数 发送给数据库服务器，因为SQL已经编译过，参数中特殊字符不会当做特殊字符编译，无法达到SQL注入的目的

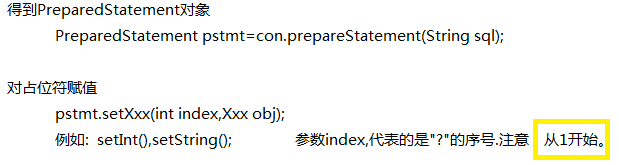
问题：

## PreparedStatement接口

### 定义sql语句



### 创建命令对象



### 命令对象执行sql语句



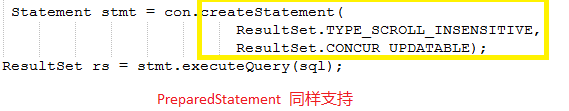
### 批处理操作



# ResultSet 高级应用 ---- 滚动结果集

## 初始化

设置结果集类型，并发策略



## 三种组合

