**1.为什么使用PreparedStatement？**

1）使用statement需要进行拼写SQL语句。很繁琐且容易出错。

String sql = **"INSERT INTO examstudent "** + **"VALUES("** + student.getFlowId()  
 + **","** + student.getType()  
 + **",'"** + student.getIdCard()  
 + **"','"** + student.getExamCard()  
 + **"','"** + student.getStudentName()  
 + **"','"** + student.getLocation()  
 + **"',"** + student.getGrade()  
 + **")"**;

使用PreparedStatement

*//创建PreparedStatement对象*

String sql = **"INSERT INTO examstudent VALUES(**?,?,?,?,?,?,?**)"**;

PreparedStatement ps = conn.preparedStatement(sql);  
*//调用PreparedStatement的setXxx(int index,Object val)设置占位符的值，注：索引值从1开始。id自动生成*

*//执行SQL语句：executeQuery()或executeUpdate()*

*//注：此时不需要传入sql语句*

ps.excuteUpdate();

2）可以有效地禁止sql注入

SQL注入是利用某些系统没有对用户输入的数据进行充分的检查，而在用户输入数据中注入非法的SQL语句段或命令，从而利用系统的SQL引擎完成恶意行为的做法。（对于Java而言，要防范SQL注入，只要用PreparedStatement取代Statement就可以了。）

SELECT \* FROM users WHERE username= 'a' OR password = ' AND password = ' OR '1'='1'

@Test  
**public void** testSQLInjection() {  
 String username = **"a' OR password = "**;  
 String password = **"OR '1'='1"**;  
 String sql = **"SELECT** *\** **FROM users WHERE username= '"** + username  
 + **"' AND "** + **"password = '"** + password  
 + **"'"**;  
  
 System.***out***.println(sql);  
  
 Connection conn = **null**;  
 Statement statement = **null**;  
 ResultSet rs = **null**;  
 **try** {  
 conn = JDBCUtil.*getConnection*();  
 statement = conn.createStatement();  
 rs = statement.executeQuery(sql);  
  
 **if** (rs.next()) {  
 System.***out***.println(**"登陆成功！"**);  
 } **else** {  
 System.***out***.println(**"用户名和密码不匹配或用户不存在！"**);  
 }  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 } **finally** {  
 JDBCUtil.*release*(rs, statement, conn);  
 }  
}

**PreparedStatement VS Statement**

* 代码的可读性和可维护性；
* 最大可能提高性能；
* 防止SQL注入。