为什么要始终使用PreparedStatement代替Statement? PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/144/2021_2022__E4_B8_BA_E4_BB_80_E4_B9_88_E8_c104_144599.htm 在JDBC应用中,如果你已经是稍有水平开发者,你就应该始终以PreparedStatement代替Statement.也就是说,在任何时候都不要使用Statement.基于以下的原因:一.代码的可读性和可维护性.虽然用PreparedStatement来代替Statement会使代码多出几行,但这样的代码无论从可读性还是可维护性上来说.都比直接用Statement的代码高很多档次:stmt.executeUpdate("insert into tb_name (col1,col2,col2,col4) values (" var1 "," var2 "," var3 "," var4 ")").perstmt = con.prepareStatement("insert into tb_name (col1,col2,col2,col4) values (?,?,?,?)").perstmt.setString(1,var1).perstmt.setString(2,var2).perstmt.setString(3,var3).perstmt.setString(4,var4).perstmt.executeUpdate().不用我多说,对于第一种方法.别说其他人去读你的代码,就是你自己过一段时间再去读,都会觉得伤心.二.PreparedStatement尽最大可能提高性能.每一种数据库都会尽最大努力对预编译语句提供最大的性能优化.因为预编译语句有可能被重复调用.所以语句在被DB的编译器编译后的执行代码被缓存下来,那么下次调用时只要是相同的预编译语句就不需要编译,只要将参数直接传入编译过的语句执行代码中(相当于一个涵数)就会得到执行.这并不是说只有一个Connection中多次执行的预编译语句被缓存,而是对于整个DB中,只要预编译的语句语法和缓存中匹配.那么在任何时候就可以不需要再次编译而可以直接执行.而statement的语句中,即使是相同一操作,而由于每次操

作的数据不同所以使整个语句相匹配的机会极小,几乎不太可能匹配.比如:insert into tb_name (col1,col2) values (11,22).insert into tb_name (col1,col2) values (11,23).即使是相同操作但因为数据内容不一样,所以整个个语句本身不能匹配,没有缓存语句的意义.事实是没有数据库会对普通语句编译后的执行代码缓存.当然并不是所以预编译语句都一定会被缓存,数据库本身会用一种策略,比如使用频度等因素来决定什么时候不再缓存已有的预编译结果.以保存有更多的空间存储新的预编译语句.三.最重要的一点是极大地提高了安全性.即使到目前为止,仍有一些人连基本的恶义SQL语法都不知道.String sql = "0select * from tb_name where name= " varname " and passwd=" varpasswd "".如果我们把[or 1 = 1]作为varpasswd传入进来.用户名随意,看看会成为什么?0select * from tb_name = 随意 and passwd = or 1 = 1.因为1=1肯定成立,所以可以任何通过验证.更有甚者:把[.0drop table tb_name.]作为varpasswd传入进来,则:0select * from tb_name = 随意 and passwd = .0drop table tb_name.有些数据库是不会让你成功的,但也有很多数据库就可以使这些语句得到执行.而如果你使用预编译语句.你传入的任何内容就不会和原来的语句发生任何匹配的关系.只要全使用预编译语句,你就用不着对传入的数据做任何过虑.而如果使用普通的statement,有可能要对0drop,.等做费尽心机的判断和过虑.上面的几个原因,还不足让你在任何时候都使用PreparedStatement吗?