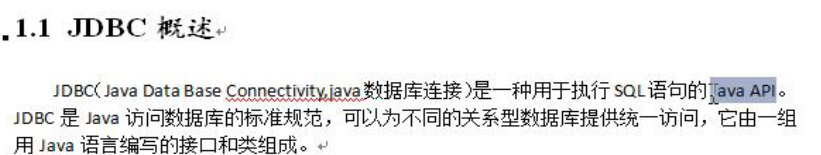
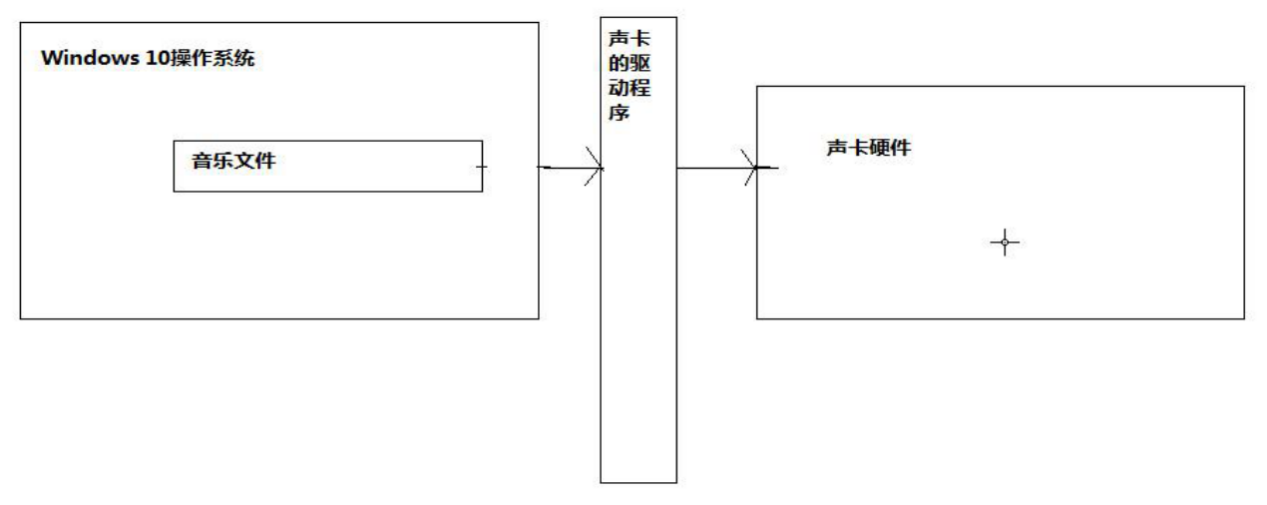
1 外连接没有null时（即都有对应时）和内连接有什么区别？？？？？？？？？

2 主表：查询的主要数据在哪个表哪个就是主表。**不太固定**。

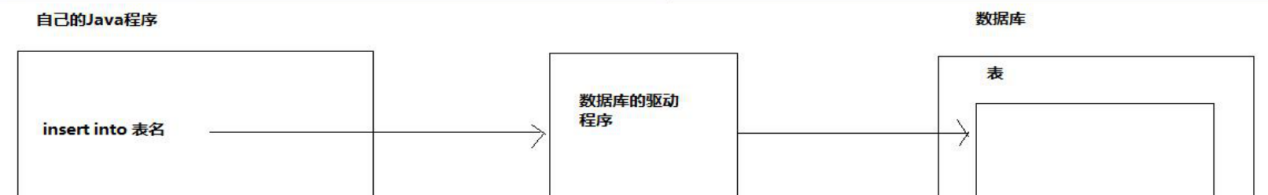
3

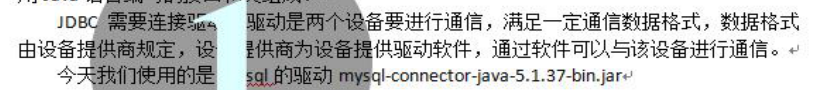


4 驱动程序是为了调用硬件：



5 数据库驱动程序：



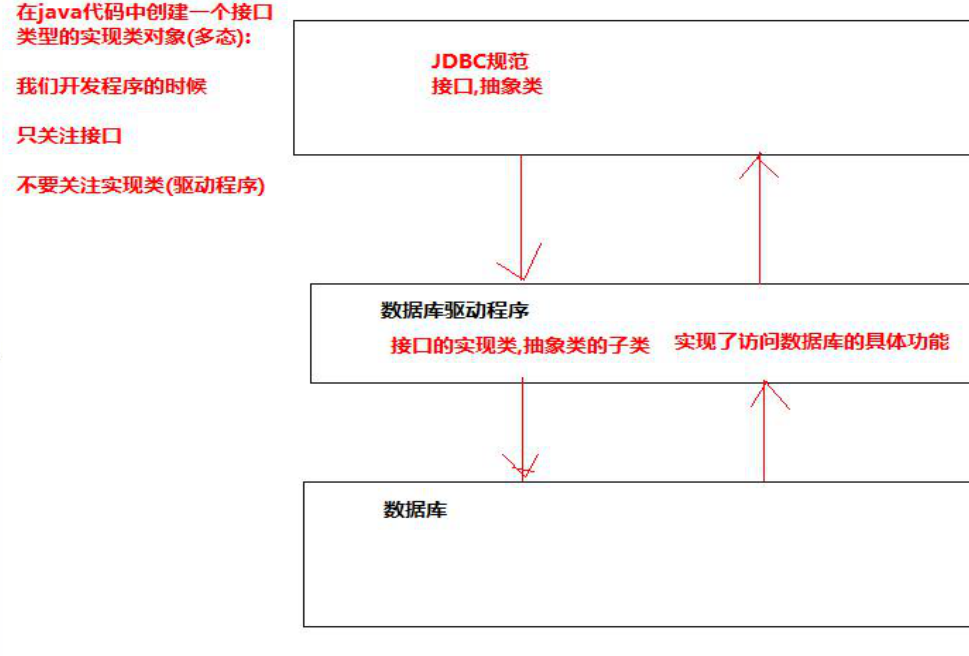


6 **数据库的驱动程序由数据库提供商提供**：（**区分规范和规范的实现类**）



**JDBC规范提供接口和抽象类，不同的数据库驱动都要实现这些接口和抽象类。**接口和抽象类的意义是对外暴露方法，隐藏实现细节。（**毕竟接口和抽象类都是在封装方法**）

7



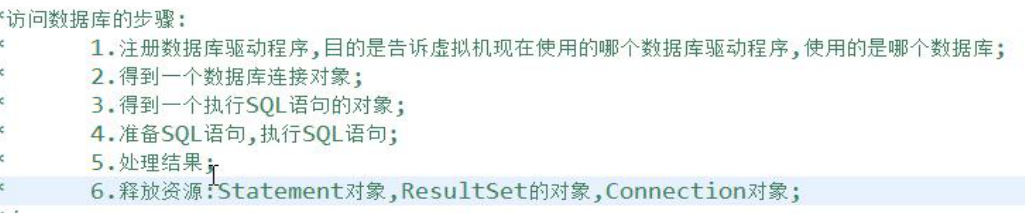
8 一个**jar包**包含很多**类文件**。

9 javax.sql中有jar包中的类。

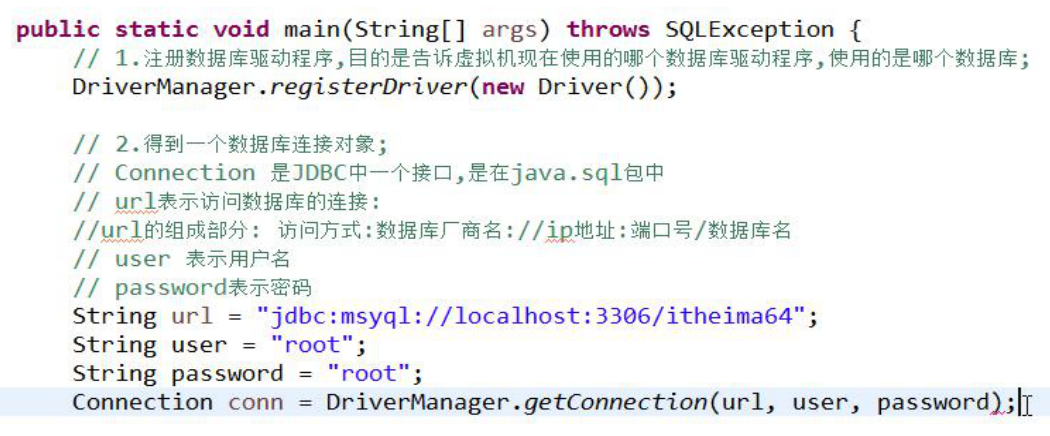
10



11



12 编程实例：

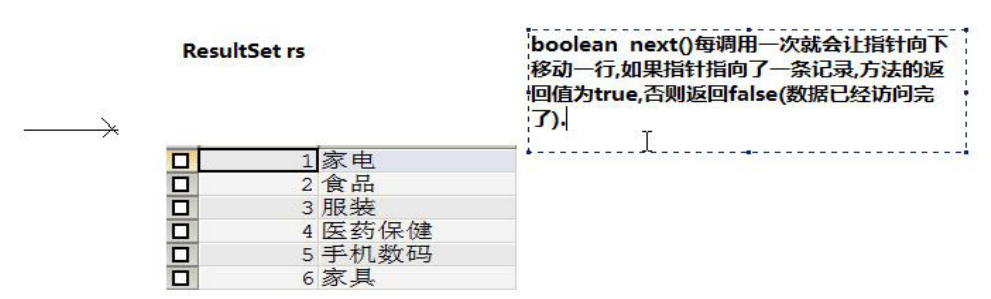






打印出来的是结果集在堆内存的地址。

13 结果集指针：（第一次循环指向结果集的第一项的上方，因此首先next()）



14 解析结果集：（rs.next()表示指针向下移动一位）（resultset是一个普通的对象，而不是集合）对ResultSet对象的介绍：（**他是一个含有光标的数据表而不是一个光标**）

public interface **ResultSet**

extends [Wrapper](mk:@MSITStore:C:\\Users\\书豪\\Desktop\\JDK_API_1.6.CHM::/java/sql/../../java/sql/Wrapper.html" \o "java.sql 中的接口)

表示数据库结果集的数据表，通常通过执行查询数据库的语句生成。

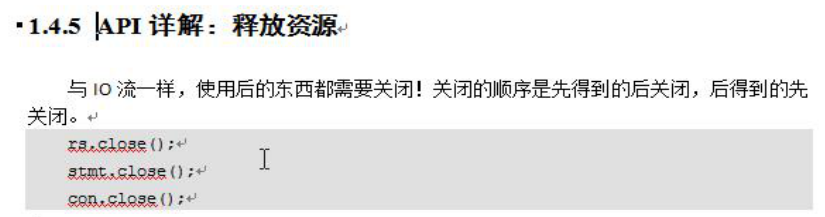
ResultSet 对象具有指向其当前数据行的光标。最初，光标被置于第一行之前。next 方法将光标移动到下一行；因为该方法在 ResultSet 对象没有下一行时返回 false，所以可以在 while 循环中使用它来迭代结果集。

**Stat.executeQuery()在查询和插入时返回的值不同，查询时返回的是数据表，插入时返回的是受影响的行数，也就是插入的行数。**





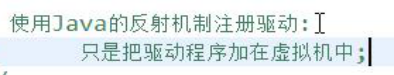
15

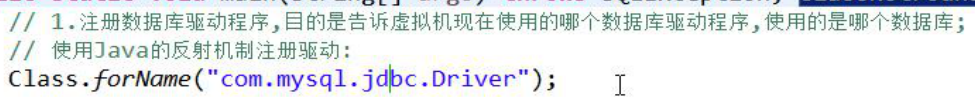


16



17 下面是注册驱动的第二种方法：

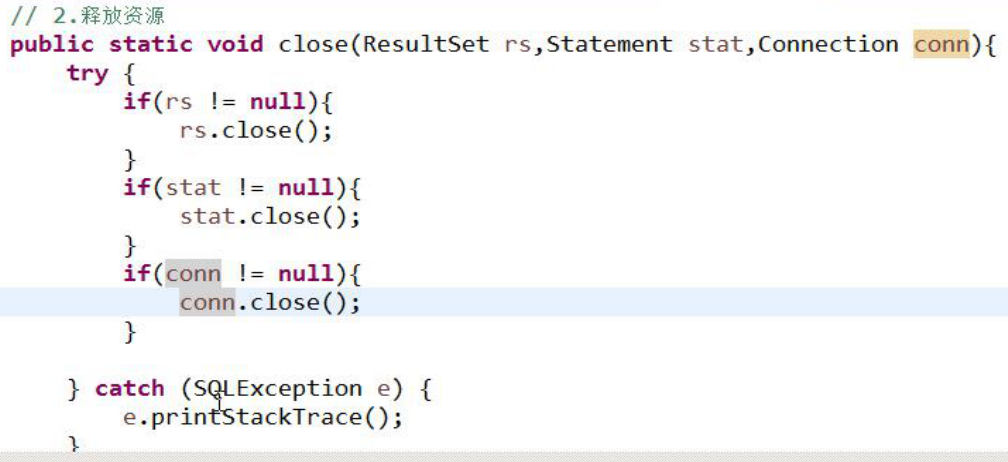




18 可以自己创建一个包然后创建类的静态方法，



依次关闭资源：



19 JDBC向表中插入记录：（**注意双引号里面用单引号**）

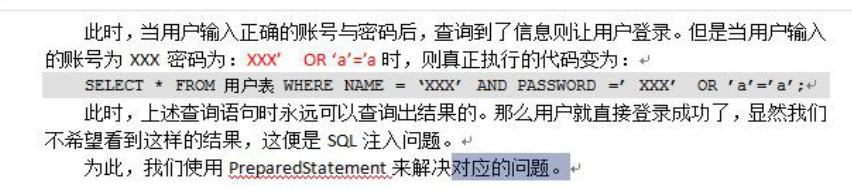


**rowNumber变量代表修改的记录的个数**

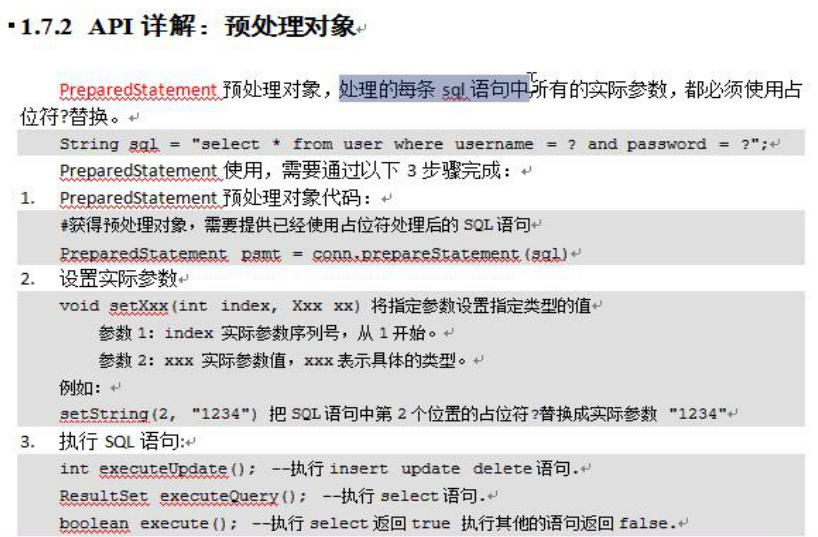
增删改查的在讲义里，自己看，这里不完整？？？？？？？？？？？？？

20 下面的可以获得整个表：（**sql注入，即不正常的查询引发不正常的结果**）（sql注入的目的是获得数据）



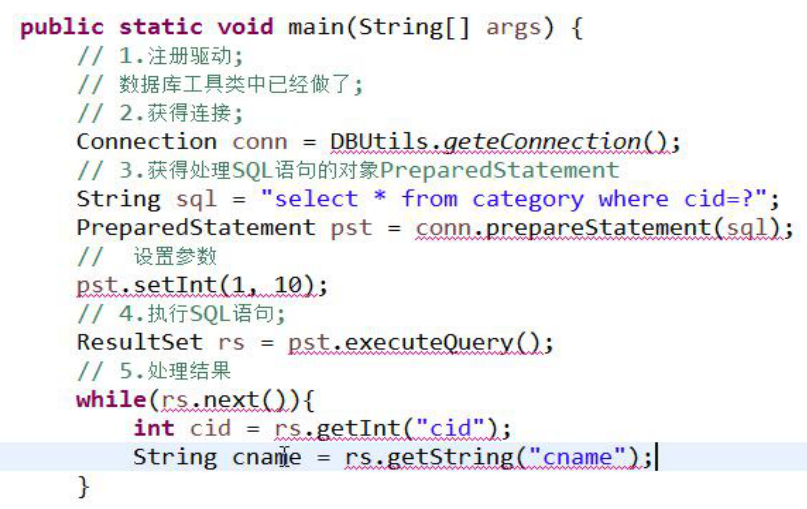


21 预处理防止注入：（**就是将createStatement变成了PreparedStatement，同时将sql语句从executeQuery里前移到了PreparedStatement里**）



22 防注入实例（预处理对象）：





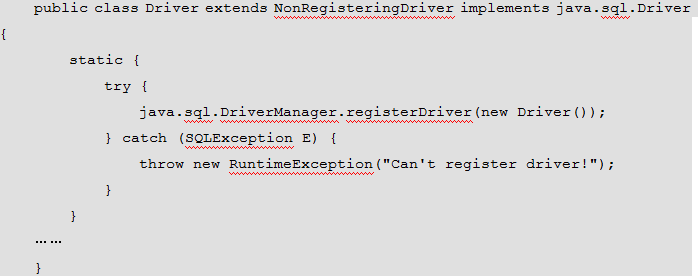


集体抛所有异常可以选定所有的代码然后抛出异常。**PreparedStatement导包要导sql包，不能导JDBC包。上面的setInt()的第二个参数是待检测的用户输入的查询。第一个参数表示的索引是从1开始的而不是0.**

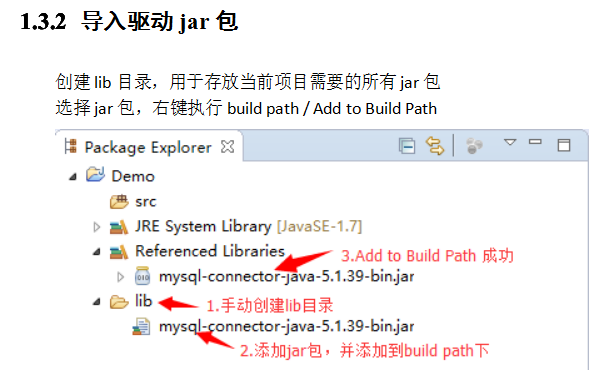
23 JDBC优化在文档里面。即第24项。

24 注册驱动：

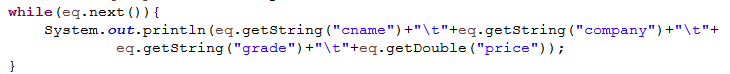




25 使用的是mysql，因此导入mysql的jar包并添加路径：



26 getString()方法里面是字符串而不是变量：



27 导包要导**import** java.sql.Statement;而不是import com.mysql.jdbc。总之就是包名中不带jdbc而是带sql。

28 查询的话关闭三次，增删改的话关闭两次，因为增删改的时候返回的是整型表示受影响的记录数。

29