# 操作数据库步骤

## 注册驱动

注冊驱动有三种方式：

　　1.Class.forName(“com.mysql.jdbc.Driver”);

　　 推荐这样的方式，不会对详细的驱动类产生依赖

　　2. DriverManager.registerDriver(com.mysql.jdbc.Driver);

　　 会对详细的驱动类产生依赖

　　3. System.setProperty(“jdbc.drivers”, “driver1:driver2”);

　　 尽管不会对详细的驱动类产生依赖；但注冊不太方便。所以非常少使用

## 建立连接

通过Connection建立连接，Connection是一个接口类。其功能是与数据库进行连接（会话）。

　　建立Connection接口类对象：

　　Connection conn =DriverManager.getConnection(url, user, password);

　　当中URL的格式要求为：

　　JDBC:子协议:子名称//主机名:port/数据库名？属性名=属性值&…

　　如："jdbc:mysql://localhost:3306/test“

　　user即为登录数据库的username，如root

password即为登录数据库的密码，为空就填””

## 创建运行SQL的语句

运行对象Statement负责运行SQL语句。由Connection对象产生。

　　Statement st = connection.createStatement();

　　Statement接口类还派生出两个接口类PreparedStatement和CallableStatement，这两个接口类对象为我们提供了更加强大的数据訪问功能。

　　PreparedStatement能够对SQL语句进行预编译，这样防止了 SQL注入 提高了安全性。

　　PreparedStatement ps=connection.prepareStatement( "update user set id=? where username=?”); ————sql语句中庸 ？ 作为通配符，变量值通过参数设入：ps.setObject(1, object);

　　而且预编译结果能够存储在PreparedStatement对象中。当多次运行SQL语句时能够提高效率。

作为Statement的子类，PreparedStatement继承了Statement的全部函数。

## 运行语句

运行对象Statement 或 PreparedStatement 提供两个经常使用的方法来运行SQL语句。

　　executeQuery(Stringsql),该方法用于运行实现查询功能的sql语句。返回类型为ResultSet（结果集）。

　　如：ResultSet rs =st.executeQuery(sql);

　　executeUpdate(Stringsql),该方法用于运行实现增、删、改功能的sql语句，返回类型为int，即受影响的行数。

如：int flag = st.executeUpdate(sql);

## 处理运行结果

　ResultSet对象

　　ResultSet对象负责保存Statement运行后所产生的查询结果。

　　结果集ResultSet是通过游标来操作的。

　　游标就是一个可控制的、能够指向随意一条记录的指针。

　　有了这个指针我们就能轻易地指出我们要对结果集中的哪一条记录进行改动、删除，或者要在哪一条记录之前插入数据。一个结果集对象中仅仅包括一个游标。

　　另外，借助ResultSetMetaData ，可以将数据表的结构信息都查出来。

　　ResultSetMetaData rsmd= resultSet.getMetaData();

## 释放资源

Connection conn = null;

Statement state = null;

ResultSet set = null;

Finally中释放资源

set.close();

state.close();

conn.close();

这三个资源是释放

倒着顺序释放

注意的问题：

如果程序刚创建上面这几个对象之后就抛异常了，释放的时候就会抛出空指针异常。所以释放时应该判断。

If(set != null){

Set.close();

}

If(state != null){

State.close();

}

...................