Jdbc

1、对数据库进行增删改查的操作，简称crud create新建 retrieve查询 update 更新 delete 删除

语言操作数据库的方式：

odbc jdbc – odbc – 数据库 jdbc ---数据库

o:open j:java

jdbc介绍

jdbc(Java Database Connectivity)是一种用于执行Sql语句的java API,可以为多种关系数据库提供统一的访问，它由一组用java语言编写的类和接口组成，jdbc为数据库开发人员提供了一个标准的API，据此可以建立更高级的工具和接口，使数据库开发人员能够用纯java api编写数据库应用程序，同时，jdbc也是个商标名

有了jdbc向各种数据库发送sql语句就是一件很容易的事，换言之有了jdbc API，就不必为访问Sybase数据库专门写一个程序，为访问oracl数据库又专门写一个程序等等，尽管jdbc在java语言层面实现了统一，但不同数据库仍有许多差异，为了更好的实现跨数据库操作，于是诞生了Hibernate项目，Hibernate是对jdbc的再封装，实现了对数据库操作更宽泛的统一和更好的可移植性

案例：通过jdbc驱动程序连接mysql数据库，完成对emp表的crud操作

//1、加载驱动(把需要的驱动程序加入内存)

Class.*forName*("com.mysql.jdbc.Driver");

//2、建立连接对象(得到连接)

String url = "jdbc:mysql://localhost:3306/test?user=root&password=6239333";

Connection ct = DriverManager.*getConnection*(url);

//3、创建Statement或者preparedStatement用于发送sql语句

Statement st = ct.createStatement();

//4、执行crud，创建，备份，删除数据库

A:添加，修改，删除数据

//executeUpdate可以执行添加，修改，删除操作,返回值表示操作了几条记录

**int** i = st.executeUpdate("insert into dept values(50,'保安部','襄阳')");

B:查询操作

//查询操作

//ResultSet结果集,可理解成是一个表的结果集,rs指向结果集的第一行记录的上一行

ResultSet rs = st.executeQuery("select \* from dept");

while(rs.next()){

int deptno = rs.getInt(1);

String dname = rs.getString(2);

String address = rs.getString("address");

System.out.println("部门号："+deptno+"部门名称："+dname+"所在地址："+address);

}

//注：取值可以按编号取值，也可以按列名进行取值

//5、关闭资源，后打开的先关闭，先打开的后关闭

If(ct!=null){ Ct.close()}

Statement和PreateredStatemented的区别

Statement和PreateredStatemented都可以在java程序中把sql语句发送给数据库，并执行，但它们也存在区别：

1. 直接使用Statement，驱动程序一般不会对sql语句作处理而直接交给数据库执行，使用PreparedStatemented,形成预编译的过程(把语句转成二进制文件方便数据库执行)，并且会对语句作字符集的转换，这样做有两个好处，首先对于多次重复执行的语句，使用PreparedStatemented效率会更高一点，并且在这种情况下也比较适合使用batch，其次可以比较好的解决系统的本地化问题
2. PreparedStatemented还能有效的防止危险字符的注入,也就是sql注入的问题

Eg:创建用户表，演示注入漏洞

Eg:新建案例演示PreparedStatemented的使用方法

ps = ct.prepareStatement("select \* from dept where deptno = ? ");

ps.setInt(1,10);

ps.executeQuery();

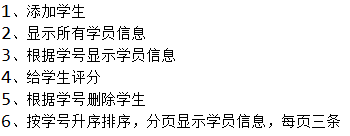
ps可以使用注入的方式来传值，这样可以防止注入漏洞

//如何解决中文乱码问题：

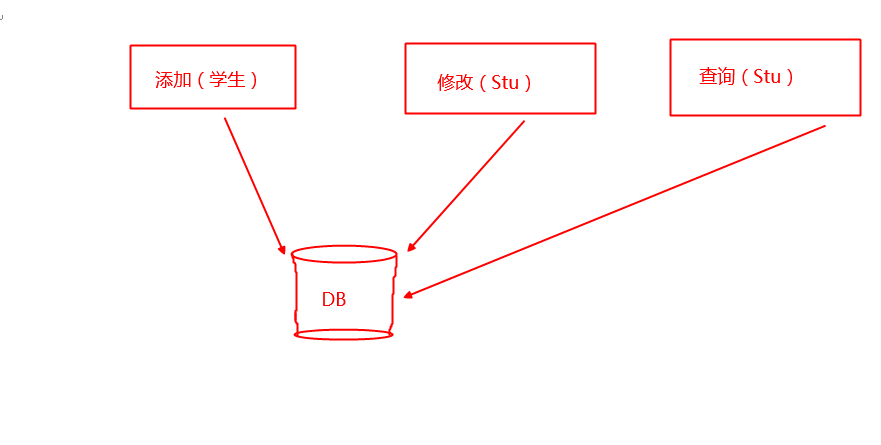
useUnicode=true&characterEncoding=UTF-8

综合案例：

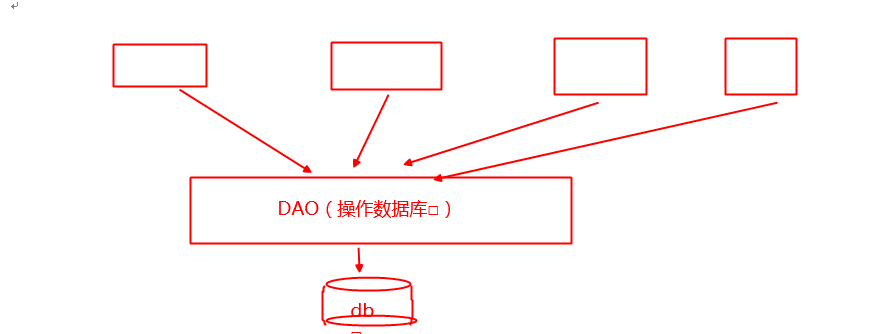
学生管理系统，改进，使用jdbc连接数据库，完成对学生的crud操作



该案例已完成，但也存在不足，我们可以看到，我们在学生管理类中操作数据库，如果还有班级表，教师表的管理，我们对班级，老师的操作，又要做重新连接数据库，获取连接等操作，我们想，能不能对我们的项目做一个改进，使业务与数据操作分离，



我们做如下改进



1. 我们新建一个SqlHelper类来对我们的操作进行封装

我们对增删改进行封装操作

**public** **boolean** update(String sql,String pars[]){

**boolean** b = **false**;

**try** {

ps= ct.prepareStatement(sql);

**for**(**int** i = 0;i<pars.length;i++){

ps.setString(i+1, pars[i]);

}

ps.executeUpdate();

b = **true**;

} **catch** (Exception e) {

e.printStackTrace();

}**finally**{ **this**.close();}

我们对查询进行封装操作，返回一个ResultSet

**public** ResultSet search(String sql,String pars[]){

**try** {

ps = ct.prepareStatement(sql);

**if**(pars!=**null**){

**for**(**int** i = 0;i<pars.length;i++){

ps.setString(i+1, pars[i]);

}

}

rs = ps.executeQuery();

**return** rs;}

问题：

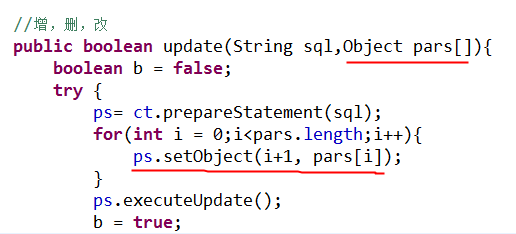
1. 我们看到，当传入的数据不是String类型时，我们需要把它转为String类型，比较繁琐，而且当使用分页查询时，如果传入当前页类型为String时会出现问题，
2. 我们每次调用都new SqlHelepre而每次新建时都需要加载驱动，建立连接，而这种资源会占用较多内存，如果不考虑并发的情况，我们希望内存中同一时该只有一个连接存在，该怎么解决
3. 我们的查询方法返回类型为rs，而rs为一种资源，一直打开并不安全，我们可否不用返回rs而直接把查询的数据返回给调用层

解决方法：

1. 传值时使用Object[] 数组来传递参数 ，注入时使用ps.setObject()来注入值，
2. 当不用考虑多线程并发问题时，我们可以创建一个单例的DbManager类，在该类中创建一个静态的Connection，当使用连接时就用该单例DbManager来获取连接，保证其唯一性，
3. 我们查询的数据为一个表，该表的每行数据类型可能都不一样，我们可以考虑把每行数据封装成一个Object[]数组，再把这些数组统一存放到一个list里面，然后把list返回即可

案例：

1、



2、

3

