Mysql 根据 多个id删除多条记录

1.问题描述： Mysql 根据 多个id删除多条记录？

常见于checkbox 多选按钮组，批量删除数据。

2.解决方案：

3.java代码实现过程

4.主要知识前端ajax发送一个数组到后台用到 arr.join(“,”)分割数据方法

5. // 以逗号分割数据才能将数据发送到后台，并用request.getParameter（）获取.

6. $(".select input[type='checkbox']:checked").each(**function** (){}判断当前被选中的checkbox

7.前端代码：

$("#deleteBtn").click(**function**(){

**var** abid = [];

$(".select input[type='checkbox']:checked").each(**function** () {

**var** arr = $(**this**).parent().parent().next().text();

console.log("");

abid.push(arr);

});

**if**( abid.length==0){

alert("没有选项哟");

}

**else**{

confirm("确定删除?", "", **function** (isConfirm) {

**if** (isConfirm) {

console.log("确认");

// 一部传输bid 数组

$.ajax({

url:'servlet/DeleteServlet',

type:'post',

dataType:'json',

// 以逗号分割数据才能将数据发送到后台，并用request.getParameter（）获取.

data:{"abid" :abid.join(',') },

success:**function**( data ){

console.log(data);

**if**( data == 1 ){

alert("删除成功！");

}**else**{

alert("删除失败！");

}

load(parseInt($("#panel-footer ul>li a.active").text()));

}

})

}

})

}

});

8.后台servlet 代码 doGet方法

9.其中 **PageContext**.*setDeleteList*(deleteList);为 modal在中，是静态的TreathLocal类型的类，

存在这里面的数据以线程为单位存在（既从前端都dao层都能直接访问和定义）。

**PrintWriter** **out** = response.getWriter();

// String Bid=request.getParameter("Bid");

**String** **aBid**=request.getParameter("abid");

// System.out.println("批量删除操作");

// System.out.println(aBid);

**try** {

**if**(aBid!=**null**&&!aBid.isEmpty()) {

// 将前台的数据解析成字符串数组；并存在数组中，之后添加在setDeleteList（）

**ArrayList**<String> **deleteList** = **new** ArrayList<String>();

// 以逗号为分割符号读取内容

**String**[] **temp** = aBid.split(",");

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

deleteList.add(temp[i]);

// System.out.println("这里是前台数据");

// System.out.println(temp[i]);

}

// 初始化 Treathlocal中的 PageContext.setDeleteList（）

**PageContext**.*setDeleteList*(deleteList);

IBookDao **book** = **daoFactory**.*getBookDao*();

**boolean** **flag**= book.deleteBooks();

**if**( flag) {

// 删除成功为1

out.print(1);

}**else** {

// 删除失败为 0

out.print(0);

}

// System.out.println(flag);

}

} **catch** (**Exception** **e**) {

e.printStackTrace();

}

10.数据库访问层dao层代码：

**public** **boolean** **deleteBooks**() {

**boolean** **flag** = **false**;

Connection **con** = **null**;

**PreparedStatement** **ps**=**null**;

// java 变量一定要赋初始值

**String** **temp** = **null**;

**ArrayList**<String> **deleteList** = **new** ArrayList<String>();

// 调用PageContext.getDeleteList(); 方法，返回ArrayList 类型的数组

**try** {

deleteList = **PageContext**.*getDeleteList*();

con = **DBCon**.*getConnection*();

**String** **sql** = "delete from book where bid in(";

Iterator<String> **iterator** = deleteList.iterator();

**while**(iterator.hasNext()) {

// temp = "'" +temp+ " '";

temp = iterator.next();

sql += "'"+temp+"',";

}

// 取个小巧

sql +="'')";

// System.out.println(sql);

// System.out.println(temp);

**if**(temp!=**null**&&!temp.isEmpty()) {

flag =**true**;

}

ps = (**PreparedStatement**) con.prepareStatement(sql);

ps.execute();

} **catch** (**Exception** **e**) {

flag =**false**;

e.printStackTrace();

}**finally** {

**DBCon**.*close*(ps);

**DBCon**.*close*(con);

**PageContext**.*removeDeleteList*();

}

**return** flag;

}

Delete from  table  where   uname  in('aaa',bbb','ccc','ddd','eee',ffff'');

select \* from book where bid in('b0038','b0039','b0040','b0041','')