

数据库分页

张海宁

贵州大学

hnzhang1@gzu.edu.cn

May 5, 2018

- 1 数据库分页
- 2 分页实例
- 3 修改数据库记录
- 4 Appendix
 - 相关资源

数据库分页是 web 开发中的常用技术。

在数据量大的情况下，将所有的数据显示在同一个页面中，给查看带来不便的同时，也给服务器带来了压力，这种情况下就需要对数据进行分页查询。

分页查询的方法

介绍两种数据库分页查询的机制：

- ① 一次查出所有数据，然后再根据用户的选择进行显示（通过用户点击第几页）
- ② 通过数据库自身的分页机制进行分次查询，用户点击第几页就去数据库中查询第几页

在数据量很大的时候，采用第一种机制是不明智的。

数据库自身的分页机制

数据库自身的分页机制是与数据库紧密关联的，每种数据库实现的方式都不一样。此处以 MySQL 为例。

MySQL 数据库通过 *limit* 关键字来控制 *select* 语句返回的记录数。

limit 接收一或两个整数参数

- ① With one argument, the value specifies the number of rows to return from the beginning of the result set.

```
SELECT * FROM tbl LIMIT 5;
```

- ② With two arguments, the first argument specifies the offset of the first row to return, and the second specifies the maximum number of rows to return. The offset of the initial row is 0 (not 1).

```
SELECT * FROM tbl LIMIT 5,10;
```

Question: 思考一下这种分页机制的使用方式。

以教师信息查询为例。

定义一个 bean 用于封装要查询的信息

Teacher.java 主要属性

```
private int staffid;  
private String nm;
```

创建一个类用于封装数据库的相关操作

此处以 Db.java 为例。一般情况下，一个封装了数据库操作的类要包含的主要功能有：

- ① 获取数据库连接 Connection 对象
- ② 关闭数据库连接

由于要实现分页查询，所以考虑将查询数据相关的操作封装到 Db 类中。

分页查询数据的方法

Db.java 中的 find 方法

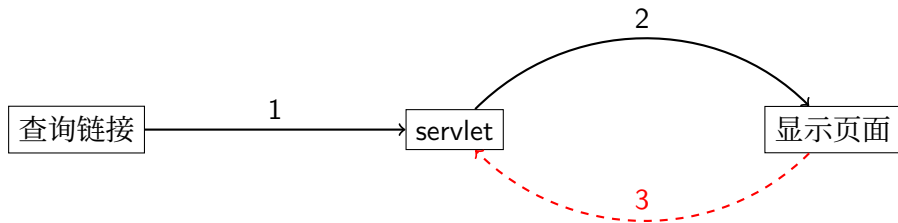
```
public ArrayList<Teacher> find(int page,int countPerPage){
    ArrayList<Teacher> teacherList = new ArrayList<Teacher>();
    String sql="select id,name,logDate from teachers limit ?,?";
    PreparedStatement ps = conn.prepareStatement(sql);
    ps.setInt(1, page);  ps.setInt(2, countPerPage);
    ResultSet rs=ps.executeQuery();
    while(rs.next()) {
        Teacher t = new Teacher();
        t.setStaffid(rs.getInt("id"));
        t.setNm(rs.getString("name"));
        t.setLogDate(rs.getString("logDate"));
        teacherList.add(t);
    }
    return teacherList;
}
```

获取总的页数

总页数 = 总记录数 ÷ 每页记录数

```
public int getTotalPages(int countPerPage) {  
    int page=0;  
    int totalCount=0;  
    String sql="select count(*) from teachers";  
    try {  
        Statement st = conn.createStatement();  
        ResultSet rs = st.executeQuery(sql);  
        rs.next();  
        totalCount =rs.getInt(1);  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    page = (totalCount/countPerPage)+1;  
    return page;  
}
```

查询



查询流程

- ① 1 -> 2
- ② 1 -> 2 -> 3 -> 2

查询链接

在 teacher.jsp 页面的合适位置，放置一个查询链接，通过本链接去调用负责查询的 servlet。

查询链接

```
<form action="QueryTeacher" method="post">  
<input type="submit" value="query teacher info"/>  
</form>
```

负责查询数据的 servlet 要完成以下工作：

- ① 查询指定的记录
offset and how many
- ② 生成页数相关的信息
current page and total page
- ③ 将结果传递给显示页面

QueryTeacher.java-doPost 查询指定的记录

```
ArrayList<Teacher> tLst=null;
int curPage = 1;
int TotalPage=0;
int countPerPage = 5;
Db d = new Db();
TotalPage=d.getTotalPages(countPerPage);
d=new Db();
if(request.getParameter("page")==null) {
    tLst=d.find(1, countPerPage);
}else {
    curPage = Integer.parseInt(
        request.getParameter("page"));
    tLst=d.find(curPage, countPerPage);
}
request.setAttribute("teacherList", tLst);
```

QueryTeacher.java-doPost 生成页数相关信息

```
StringBuilder strb = new StringBuilder();
strb.append("<div><form name='qt' action='QueryTeacher'
    method='post'>");
strb.append("第<select name='page'
    onchange='document.qt.submit()'>");
for(int i=1;i<=TotalPage;i++) {
    if(i==curPage) {
        strb.append("<option value='"+i+"' selected='selected'>"
            +i+"</option>");
    }else {
        strb.append("<option value='"+i+"'>" +i+"</option>");
    } }
strb.append("</select>页，共"+TotalPage+"页");
strb.append("</form></div>");
request.setAttribute("pageBar", strb);
```

QueryTeacher.java-doPost 将 request 传递给显示页面

```
request.getRequestDispatcher("teacher_list.jsp")  
    .forward(request, response);
```

这条语句的意思是：将本 servlet 处理之后的 request 封装起来，传递给 *teacher_list.jsp* 页面。

teacher_list.jsp

```
<div class="content main">
```

当前系统中教师信息如下：


```
<table>
```

```
<c:forEach items="${teacherList }" var="t" >
```

```
<tr><td>${t.staffid}</td><td>${t.nm }</td></tr>
```

```
</c:forEach>
```

```
</table>
```

```
${pageBar}
```

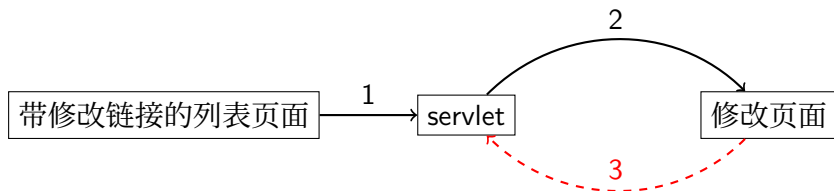
```
</div>
```

我们学习了如何往数据库里写数据，如何从数据库里查数据。
现在来学习如何修改/删除数据库里的记录。

update, delete from

- **UPDATE** Person **SET** FirstName = 'Fred' **WHERE** LastName = 'Wilson'
- **DELETE FROM** Person **WHERE** LastName = 'Wilson'

可以看出，要修改或删除某条记录，关键在于要准确定位某条记录。可以通过主键来定位。



带修改链接的列表页面

teacher_list.jsp

```
<c:forEach items="${teacherList }" var="t" >
<tr><td>${t.staffid}</td><td>${t.nm }</td>
<td>
<a href='ModifyTeacher?id=${t.staffid }&name=${t.nm}''>
修改</a></td></tr>
</c:forEach>
```

ModifyTeacher.java - doGet

```
request.setCharacterEncoding("utf-8");  
request.setAttribute("id", request.getParameter("id"));  
request.setAttribute("name", request.getParameter("name"));  
request.getRequestDispatcher("teacher_modify.jsp")  
    .forward(request, response);
```

ModifyTeacher.java - doPost

```
request.setCharacterEncoding("utf-8");
System.out.println(request.getParameter("id")+" "+
    request.getParameter("name"));
Db d = new Db();
int rz = d.modifyTeacher(request.getParameter("id"),
    request.getParameter("name"));
if(rz==1) {
    response.getWriter().println("modify successfully.");
}else {
    response.getWriter().println("something wrong.");
}
```

Db.java - modifyTeacher

```
public int modifyTeacher(String id,String name) {  
    int rz=0;  
    String sql = "update teachers set name=? where id=?";  
    try {  
        PreparedStatement ps = conn.prepareStatement(sql);  
        ps.setInt(2, Integer.parseInt(id));  
        ps.setString(1, name);  
        rz=ps.executeUpdate();  
        getClose();  
    } catch (SQLException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
    return rz;  
}
```

teacher_modify.jsp

```
<form action="ModifyTeacher" method="post">
<table>
<tr><td>${id}
<input type="hidden" name="id" value="${id }"/></td>
<td><input type="text" name="name" value="${name }"/></td>
<td><input type="submit" value="提交修改"/></td></tr>
```


The End

Appendix

ppt、项目源代码及实验指导书的地址

- ① ppt
<https://github.com/gmsft/ppt/tree/master/javaweb>
- ② lab-java 项目
<https://github.com/gmsft/javaweb>
- ③ 实验指导书
放在 ppt 的仓库里