**数据库实验报告**

——亦云博客

2016202115 吕芳锐

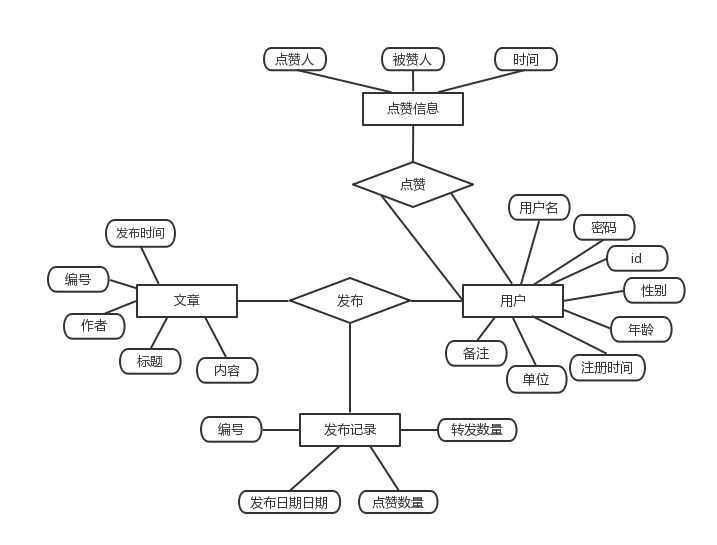
1. 组内分工

前端页面：搜索展示页面、页头、页尾、文章编辑页面、登录注册选择页面

后端数据库：数据库建立、和文章相关操作，包括文章的关键字搜素、排序、发布文章以及关注页面展示。

1. 核心代码
2. 数据库建立

我们建立了五张基础表，分别是users、article、follow、repost、discuss，五张表内分别存放用户信息、文章信息、点赞信息、转发信息、评论信息。E-R图如下：



代码如下：

create database yiyun;

use yiyun;

create table users(id int primary key,

passwd varchar(100),

username char(100));

create table article( id int primary key,

title varchar(100),

content longtext,

userid int,

publishtime datetime,

username varchar(45),

click int,

flag int

);

create table follow(

a int,

b int,

publishtime datetime

);

create table repost(

userid int,

articleid int,

publishtime datetime);

create table discuss(

userid int,

articleid int,

content longtext,

publishtime datetime);

1. 数据库连接封装

我们使用的是MySQL数据库，在代码中将数据库连接封装，方便以后使用：

//引用包

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.naming.NamingException;

public class DBConnector { //建立连接类

private static Connection con;

private static String driver = "com.mysql.jdbc.Driver";

private static String url = "jdbc:mysql://183.174.60.33/yiyun?autoReconnect=true&amp;autoReconnectForPools=true&characterEncoding=UTF-8"; //连接数据库地址

private static String user = "root";

private static String password ="web@ipv6";

private Statement st;

static {

try {

Class.forName(driver);

con = DriverManager.getConnection(url,user,password);

} catch (SQLException e) {

System.out.println("DBConnector 29:"+e.getMessage());

} catch (ClassNotFoundException e) {

System.out.println("DBConnector 31:"+e.getMessage());

}

}

public DBConnector(){ //连接类

try {

if(con!=null&&!con.isClosed()) {

st=con.createStatement();

}else {

Class.forName(driver);

con = DriverManager.getConnection(url,user,password);

st=con.createStatement();

}

} catch (SQLException e) {

System.out.println("DBConnector 29:"+e.getMessage());

e.printStackTrace();

} catch (ClassNotFoundException e) {

System.out.println("DBConnector 31:"+e.getMessage());

e.printStackTrace();

}

}

public ResultSet excuteQuery(String sql) { //执行sql

try {

return st.executeQuery(sql);

} catch (SQLException e) {

System.out.println("DBConnector 24:"+e.getMessage());

return null;

}

}

public boolean execute(String sql) { //执行sql

boolean b=false;

try {

st.execute(sql);

b=true;

} catch (SQLException e) {

System.out.println("DBConnector 40:"+e.getMessage());

b=false;

}finally {

return b;

}

}

public int getRows(ResultSet rs) { //得到结果数量

int rowcount=0;

if(rs!=null) {

try {

rs.last(); //直接执行跳到结果集的最后一行

rowcount = rs.getRow(); //这一句就能得到结果集的行数

rs.beforeFirst(); //重新执行到第一行的前一行，以便查询结果集的集体内容

} catch (SQLException e) {

return 0;

}

}

return rowcount;

}

@Override

protected void finalize() throws Throwable {

if(st!=null)st.close();

super.finalize();

}

public static void main(String[] args) {

DBConnector db=new DBConnector();

System.out.println("success");

}

}

1. jsp页面使用样例

以页头为例，其他模块与之用法类似就不赘述了，主要是session的使用和getParameter方法的使用

<%@page import="Bean.User"%>

<%@page import="Action.UserAction"%>

<%@page import="Impl.UserImpl" %>

//引用需要的包

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

//设定编码格式

/\*\*\*\*\*\*\*\*\*\*\*\*进入jsp代码\*\*\*\*\*\*\*\*\*\*\*\*\*/

<%

//获取项目根地址

String basePath = request.getContextPath();

//将根地址假如页面级变量方便之后通过EL表达式调用

pageContext.setAttribute("basePath", basePath);

//通过session中的user获取登录状态

User user = (User)session.getAttribute("user");

//也把变量放入pageContext 方便使用EL表达式

pageContext.setAttribute("user", user);

%>

<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- 上述3个meta标签\*必须\*放在最前面，任何其他内容都\*必须\*跟随其后！ -->

<title>header</title>

//使用bootstrap、jquery

<!-- css -->

<link href="../css/bootstrap.css" rel="stylesheet" />

<link href="../css/header\_and\_footer.css" rel="stylesheet" />

<!--javascript-->

<script src="../js/jquery-3.3.1.js"></script>

<script src="../js/require.js"></script>

<script src="../js/bootstrap.js"></script>

<script src="../js/header-and-footer.js"></script>

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->

<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->

<!--[if lt IE 9]>

<script src="https://cdn.bootcss.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script src="https://cdn.bootcss.com/respond.js/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body id="header-body">

<nav class="header-nav navbar navbar-inverse navbar-fixed-top">

<div class="container-fluid">

<!-- Brand and toggle get grouped for better mobile display -->

<div class="navbar-header">

<button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">

<span class="sr-only">Toggle navigation</span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

<a class="navbar-brand" id="header\_brand" href="../homepage.jsp" >

<img alt="Brand" id="brand\_img" style="height :140%;display:inline;float:left;margin-top:-6px;" src="../image/header\_brand\_image.png">

<span style="float:left;padding:10px 10px;margin-left:10px;">亦云</span>

</a>

</div>

<!-- Collect the nav links, forms, and other content for toggling -->

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

<ul class="nav navbar-nav">

<li><a href="../homepage.jsp">首页</a></li>

<li><a href="#">热门</a></li>

<li><a href="../PersonalCenter/guanzhu.jsp">发布</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<li>

<!-- Button trigger modal -->

//判断是否登录，登录显示个人信息，否则显示登录按键

<%

if(session.getAttribute("user")==null ){

%>

<a href="#" onclick="window.open('../login\_and\_signup/sign\_up.html')">

登录|注册

</a>

<%

}else{

%>

<a href="../PersonalCenter/PersonalCenter.jsp" >

<%=user.getUsername() %>

</a>

<%

}

%>

</li>

<li><a href="#">帮助</a></li>

</ul>

//搜索栏，action定位到搜索代码

<form id="footer-searcher" class="navbar-form navbar-left navbar-right" action="../searcher/searcher.jsp" method="post">

<span class="form-group">

<input id="footer-searcher-input" type="text" name="key" class="form-control" placeholder="搜索">

</span>

<button type="submit" id="footer-searcher-button" class="btn btn-default" >

<span class="glyphicon glyphicon-search" aria-hidden="true"></span>

</button>

</form>

</div>

<!-- /.navbar-collapse -->

</div>

<!-- /.container-fluid -->

</nav>

</body>

</html>

后台java代码样例：（以搜索代码为例）

* + - 1. Article类数据结构定义：

package Bean;

public class Article {

private String title;//标题

private String content;内容

private String username;//用户名

private String publishtime;//发布时间

private String click;//点击量

private int userid;//用户id

private int id;//文章id

private int flag;//转发量

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

//此后省略大量get、set函数

}

* + - 1. 搜索执行方法（移除了无关代码）：

//引用包

package Impl;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.text.SimpleDateFormat;

import com.mysql.fabric.xmlrpc.base.Data;

import Bean.Article;

import DBConfig.DBConnector;

public class ArticleImpl {//创建操作类

//根据个数num获取文章的数目，进行排序后返回，这里按照默认顺序排序

public static Article[] getArticles(int num) {

Article[] as = new Article[num];//创建接收结果的数组

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

as[i]= new Article();

as[i].setTitle("title"+i);

as[i].setContent("content"+i);

}

//建立数据库连接

DBConnector d=new DBConnector();

//执行的sql语句

String sql="select \* from article order by id";

//得到结果

ResultSet re=d.excuteQuery(sql);

//将结果录入

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

try {

re.next();

as[i].setTitle(re.getString("title"));

as[i].setContent(re.getString("content"));

as[i].setClick(re.getString("click"));

as[i].setId(re.getInt("id"));

as[i].setPublishtime(re.getString("publishtime"));

as[i].setUserid(re.getInt("userid"));

as[i].setUsername(re.getString("username"));

as[i].setFlag(re.getInt("flag"));

System.out.println(as[i].getUsername());

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return as;

}

//改函数重构了getArticle

//根据个数num获取文章的数目,key是关键字，sort是排序方法，默认0；进行排序后返回

public static Article[] getArticles(String key,String sort) throws SQLException {

//建立数据库连接

DBConnector d=new DBConnector();

//创建sql语句

String sql="select \* from article where (title Like \"%"+ key + "%\" )or(content Like \"%"+ key + "%\" )";

//按照类别选择排序方式

switch(sort) {

case "0":sql+="order by id desc";break;

case "1":sql+="order by click desc";break;

case "2":sql+="order by flag desc";break;

}

ResultSet re=d.excuteQuery(sql);

// 将游标移动到最后一行上

re.last();

// 得到当前的 row number，在 JDBC 中，row number 从1开始，所以这里就相当于行数

int rowCount = re.getRow();

//此时游标执行了最后一行，如果我们后面还想从头开始调用 next()遍历整个结果集，我们可以将游标移动到第一行前面

re.beforeFirst();

// 通过上述这步操作，我们算是回复了结果集到初始状态（即刚查询出来的状态）

//录入结果

Article[] as = new Article[rowCount];

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

as[i]= new Article();

}

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

try {

re.next();

as[i].setTitle(re.getString("title"));

as[i].setContent(re.getString("content"));

as[i].setClick(re.getString("click"));

as[i].setId(re.getInt("id"));

as[i].setPublishtime(re.getString("publishtime"));

as[i].setUserid(re.getInt("userid"));

as[i].setUsername(re.getString("username"));

as[i].setFlag(re.getInt("flag"));

System.out.println(as[i].getUsername());

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return as;

}

//测试用主函数

public static void main(String[] args) {

//getArticles(10);

}

}

1. 遇到的问题及细化
2. 前端的制作异常艰难，想做好好看、好用的前端非常不容易，我们在借鉴别人的基础上，使用bootstrap+jsp+jquery搭建自己的前端页面。
3. Jsp页面如果不指定编码格式就会乱码，在查询了问题后在每个jsp页面上都加入了制定编码的语句
4. 我们使用mysql数据库是为了能在自己电脑上编写程序，在校外时更为方便，但始料未及的是因为不熟悉所以安装数据库非常缓慢拖后了进读。
5. 没有理清数据库和前端的关系，在一开始代码很混乱，在查阅资料后了解到前后端分离的思想，开始了封装各种函数，在使用起来确实更为便利。
6. 对java语言完全不了解，在编写代码时比较吃力。在“菜鸟教程”网站上学习过后才开始写代码
7. Java调试功能使用不熟练，经常找不到错误所在，只能靠在控制台打印来调试，在编写完大作业后，对Java语言更为了解，积累了大量的经验。
8. 总结与收获

这次大作业对我而言非常有收获。真真正正了解了如何自己搭建一个网站、如何使用java语言、jsp。对网站的前端、后端都有了一个清晰的认识，深刻了解到数据库在其中发挥的作用。自己做出一个功能完善的网站非常有成就感。同时老师要求使用github，也让我对github和git的使用有所了解，这是一项非常实用的技能。同时我也了解到了很多最新的工程理念：比如前后端分离、数据库角色、视图定义等，这对我未来的发展很有好处。我也了解到了诸如session、post、get等方法的使用。总之，这次大作业让我收益匪浅。