

基于 JSP 技术的数据库浏览工具



学 院 计算机科学与技术

专 业 计算机科学与技术

年 级 2003 级

姓 名 张 辉

指导教师 张坤龙

2007 年 6 月 15 日

摘 要

论文使用 JSP、Tomcat 和 MySQL 开发套件设计和实现了一个基于 JSP 技术的数据库浏览工具 JBrowser, 目标是使对计算机和数据库不熟悉的用户摆脱学习结构化查询语言的负担。通过 JBrowser 这个工具, 数据库管理员可以将自己管理的数据库发布到 Web 上, 随后数据库用户就可以随时随地通过浏览器来浏览数据库中的内容。

论文首先阐述了数据库浏览工具的意义和发展现状。然后介绍了 JSP 技术并对 JSP 和其他 Web 开发语言进行了比较。接下来论文给出了 JBrowser 的设计, 包括体系结构、数据流图和模块划分。最后, 论文对 JBrowser 实现中使用的一些关键技术, 例如元数据浏览、分页和排序、页面间的参数传递等等, 进行了讨论。

关键词: JSP; Tomcat; MySQL; 数据库; 浏览;

目 录

第一章	绪论.....	1
1.1	数据库的浏览.....	1
1.2	国内外现状.....	2
1.3	课题的意义.....	3
1.4	论文的组织.....	3
第二章	JSP 技术.....	4
2.1	JSP 工作原理.....	4
2.2	JSP 页面的基本组成.....	4
2.3	JSP 出错处理.....	5
2.4	JSP, ASP, PHP 的比较.....	6
2.5	JSP 应用的配置.....	7
第三章	JBrowser 的设计.....	8
3.1	系统需求.....	8
3.2	体系结构.....	9
3.3	数据流图.....	10
3.4	模块划分.....	11
第四章	JBrowser 的实现.....	13
4.1	Web 应用发布.....	13
4.2	分页浏览.....	14
4.3	排 序.....	15

4.4	JSP 连接数据库	16
4.5	参数传递.....	18
第五章	总结.....	22
	参考文献.....	23
	致 谢	

第一章 绪论

1.1 数据库的浏览

数据库的浏览,是把 DBMS 中存储的数据信息(表,视图,元组等)从 DBMS 中取出,并按照一定的组织方式由终端表达给用户。数据库浏览的对象为数据,包括用户数据和元数据。用户数据比较容易理解,下面着重解释元数据的概念。

元数据最本质、最抽象的定义为: data about data (关于数据的数据),即关于数据库中的内容、质量、状况和其他特性的信息,这也使得最终用户实际上并不关心元数据。在数据库领域中,元数据有两方面的用途。首先,元数据能提供基于用户的信息帮助用户使用数据,以便就数据是否满足其需求作出正确判断。其次,元数据能支持系统对数据的管理和维护,支持系统以最有效的方式访问数据便于用户查询检索^[1]。

具体举例来说,在 MySQL 中,数据库 test 中的 test 表内的数据就是用户数据,它记录了一条 id 为 1, name 为 a 的数据。这是最终用户关心的数据,因为用户可以在这个表中加入记录,变成一个公司员工信息表或学生信息表。(虽然因为表结构过于简单,只能记录公司中是否有某个员工和这个员工的编号)。相反地,数据库 information_schema 中的所有数据都是元数据,表 COLLATION_CHARACTER_SET_APPLICABILITY 记录的是系统所支持的字符集的信息,表 SCHEMATA 中记录的是数据库的基本信息,比如各个数据库的默认存储字符集。最终用户在使用数据库的时候并不关心这些数据。

数据库的浏览方式主要有三种:基于 Web 的 B/S 结构浏览,非基于 Web 的 C/S 结构浏览,以及直接使用 DBMS 提供的界面或接口输入 SQL 语句直接查询。在这些方式中,基于 Web 的浏览方式应用最为广泛和方便。原因如下:

第一,B/S 结构比 C/S 的维护工作量大大减少。C/S 结构的每一个客户都必须安装和配置客户端软件,频繁的服务器升级和系统维护使得 C/S 结构客户端软件需要不断的更新,极易出现版本不合而造成的浏览错误。而 B/S 结构是基本上完全的瘦客户端,只需要浏览器支持,无需进行客户端的升级。

第二,B/S 结构不用专门编写通讯处理模块,直接用 Web Server 接收和发出信息,节省软件开发周期和开发成本。

第三,B/S 结构采用面向对象技术,代码可重用性好。原有代码只需稍做修改就可以满足不同的需求。B/S 结构也有不足之处,主要为基于 html,代码简单,表现形式有限,很多高级功能无法实现;客户端浏览器很容易暴露源代码,不够安全;浏览器本身结构的设计缺陷导致很容易受攻击等。但基于其强大的优点,基于浏览器/服务器模式的浏览工具在各个行业里的应用越来越广泛。

1.2 国内外现状

1.2.1 SQL 的发展状况

SQL 是用于访问数据库的最常用标准化语言,最早由 IBM 公司使用在其开发的数据库系统中,1986 年 10 月,美国 ANSI 对 SQL 进行规范后作为关系数据库管理系统的标准语言(ANSI X3.135-1986),1987 年得到国际标准化组织的支持成为国际标准^[2]。SQL 标准自 1986 年以来不断演化发展,有 1992 年发布的“SQL:92”,1999 年发布的“SQL:1999”,标准的当前版本是 2003 年发布的“SQL:2003”。

SQL 是高级的非过程化编程语言,允许用户在高层数据结构上工作,其优点主要为:它不要求用户了解对数据的存放方法和存放方式,所以具有完全不同底层结构的不同数据库系统可以使用相同的 SQL 语言作为数据输入与管理的接口。SQL 语言可以嵌套使用,这使它具有极大的灵活性和强大的功能,也意味着用 SQL 语言可以写出非常复杂的语句。SQL 是所有关系数据库的公共语言。由于所有主要的关系数据库管理系统都支持 SQL 语言,用户可使用 SQL 将数据从一个 RDBMS 转到另一个。所有用 SQL 编写的程序都是可以移植的。

1.2.2 DBMS 自带的浏览工具

Oracle: Oracle SQL Developer 是甲骨文公司开发的图形化工具,它具有拖拉功能,可以开发和调试 SQL 和 PL/SQL 代码,并可以帮助开发者完成浏览和建立、运行 SQL 脚本、编辑和调试 PL/SQL 代码、浏览和升级数据等任务。Oracle SQL Developer 具有已推出的 Oracle Database 10g 和 Oracle9i Release 2 的所有功能,甲骨文计划把转移工作台技术融入 Oracle SQL Developer,这样用户就可以从 SQL Server、Sybase、MySQL 产品直接转移到甲骨文的平台。

DB2: IBM Content Manager 为 IBM 的内容管理器软件,支持多层、分布式、基于 Web 的结构可以管理多媒体内容以及传统的文本数据,通过对元数据和文本文件的全文检索功能,内容管理器使用户能够方便地检索相关文档。

另外还有 SqlServer 的查询分析器,Informix 公司服务于动态内容的多层次 Web 应用的开发及运行工具 InformixObjectTranslator 等。

1.2.3 第三方的数据库浏览工具

因为 MySQL 的开源,在第三方工具中,支持 MySQL 的数据库浏览工具是最为常见的,其中最具代表性的就是 PhpMyAdmin,一个用 PHP 实现的第三方工具。PhpMyAdmin 在数据库设计方面的功能非常完备,但在不能浏览及直接编辑表的内容,另外编写自由 SQL 语句的界面使用也不方便。此外,图形界面的 EMS MySQL Manager Lite,界面设置和所完成的功能也非常优秀,能够让人以方便的方式管理 MySQL 数据库。最后,类似工具还有网际数据库浏览器和

MySQL-Front(使用 PHP tunnel)。这两者的工作方式很相似,都需要复制一个 php 的脚本文件到网站上,然后启动一个 Windows 界面的程序,工作时设置连接 php 脚本的 URL。网际数据库浏览器的界面类似 MS-SQL 的查询分析器,在某些方面使用更为方便,比如在带有过滤条件游览表的同时可以编辑表的内容,可以从查询结果网格中复制出字段名及 SQL 表达式等。

在大型数据库中最为成功的是第三方厂商 Quest 为甲骨文数据库产品提供的相关工具,因为本文的数据库浏览工具是基于 MySQL 的,而且作者对于 ORACLE 等大型数据库并不熟悉,所以简略带过。

1.3 课题的意义

社会正在以前所未有的速度向信息化迈进,数据在人们的生产和生活中扮演着越来越重要的角色。在各个行业中对数据的使用最多的是数据查询,但问题是对于计算机行业之外的用户来说,现有的 DBMS 提供的 SQL 查询语言虽然功能强大但并不方便,另外异地查询的需求也得不到满足,所以设计实现一个基于 Web 的数据浏览工具就有了其实际的应用意义。

另外在开发这个数据库浏览工具的时候,可以学习到数据库和动态网络开发技术相关的知识和方法,有助于锻炼专业能力,提高个人水平。

1.4 论文的组织

本文将设计和实现一个名为 JBrowser 的基于 JSP 技术的数据库浏览工具。论文剩余部分的组织结构为:

第二章介绍了 JSP 技术和毕设设计所选用的开发套件;

第三章总体阐述了 JBrowser 的设计,包括体系结构,数据流图等;

第四章说明了 JBrowser 的实现并对使用的一些关键技术,例如应用发布、数据库连接、页面间的参数传递等等进行讨论。

最后在第五章总结本次毕业设计的成果和不足,并提出改进建议。

第二章 JSP 技术

2.1 JSP 工作原理

JSP 是以 Java 技术为基础的,下面首先介绍 JSP 的工作原理,其具体执行分为以下几个步骤:

- (1)浏览器客户端向 JSP 容器(如 Tomcat)发出 JSP 页面请求;
- (2)JSP 容器将对应的页面转换为相应的 Java Servlet 源代码;
- (3)JSP 容器编译生成的源代码生成.class 文件;
- (4)JSP 容器加载运行对应的.class 文件;
- (5)JSP 容器把响应的输出结果发送到浏览器端。

页面的执行过程如图 2-1 所示^[3]。

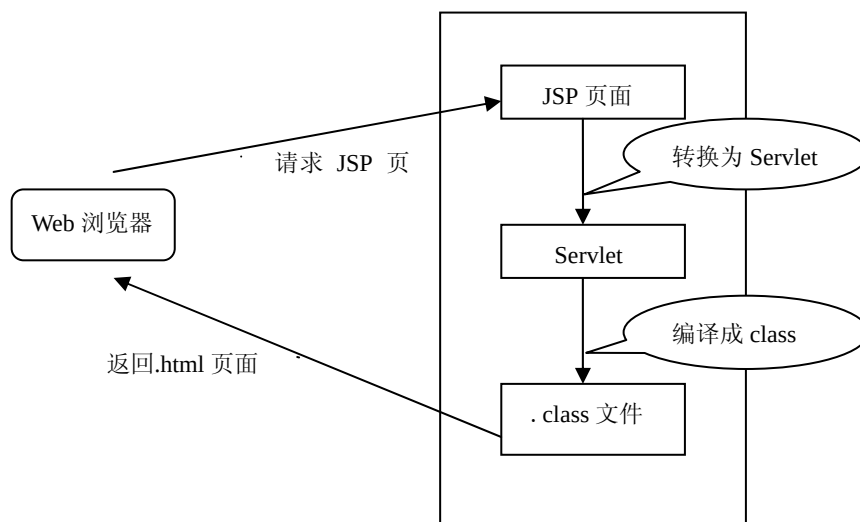


图 2-1 JSP 的执行步骤

由此可见 JSP 容器管理 JSP 页面的生命周期分为两个阶段:转换阶段(translation phase)和执行阶段(execution phase),但需要说明的一点是,Servlet 类一旦加载,在原来的 JSP 页面没有发生改变的情况下,JSP 容器对于浏览器发送过来的请求,跳过第 2、3、4 个步骤,直接执行步骤 5。

2.2 JSP 页面的基本组成

JSP 页面包括 JSP 元素、注释和 HTML 静态页面表达三部分,页面组成如图 2-2。

JSP 指令元素:用于将消息传送到 JSP 驱动,提供全局信息,独立于 JSP 页面接受的任何请求,为翻译阶段提供信息。JSP 指令元素并不产生任何输出页面

信息。

JSP 脚本元素：插入 Java 语言程序代码，其中声明用来声明仅在页面有效的变量和方法，表达式用来返回 String 类型的值，并按顺序 out 对象输出到浏览器中。脚本程序片段可以包含任何 JSP2.0 支持的代码片段脚本，脚本片段必须符合 Java 语言规范。一个 JSP 页面可以有多个脚本片段，在处理时，服务器按顺序执行，由代码决定是否产生输出^[4]。

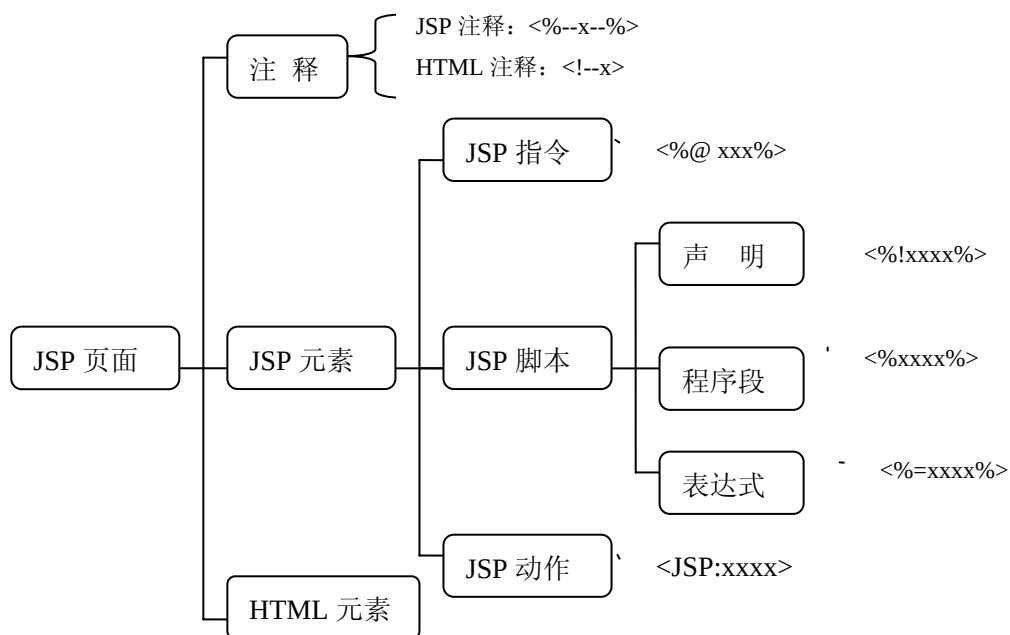


图 2-2 JSP 页面组成

JSP 行为元素：使用标记表示，遵循 XML 语法格式，JSP 规范中指定的行为称为标准行为，其作用是：

- (1)页面包含和跳转，参数设定；
- (2)存取 JavaBean；
- (3)动态产生 XML 元素；
- (4)用于标记文件。

2.3 JSP 出错处理

JSP 执行时错误可能发生在转换阶段或请求处理阶段。转换阶段发生的错误主要集中在编译过程中，所以称为编译错。请求处理阶段发生的错误成为请求处理错，也称为运行错。

编译错的主要生成原因是 JSP 页面存在语法错误，容器无法将 JSP 页面转换成 Servlet 实现类，不能生成对应的响应类。

运行错误是使用 Java 语言的异常处理机制实现的。当异常发生时,通知调用者(如 JSP 页面的实现类)捕捉和处理相应的异常。如果异常不被处理,就会导致将客户端请求及出现的异常转向由 JSP 页面定义的出错处理页面 URL,由该页面处理错误。如果页面没有指定出错处理页面,会使用默认的处理。比如当一个页面要求用户输入数值而用户输入的是字符串的时候,异常产生,从而调用错误处理页面得到输出的错误处理结果。

2.4 JSP, ASP, PHP 的比较

ASP 的特点:精确地说 ASP 是一个中间件,它将 web 上的请求转入到一个解释器中,在这个解释器中将所有 ASP 的 script 进行分析、执行,它可以调用 COM 对象(Active X 控件),对这个对象中的属性和方法进行操作,通过 COM 对象完成更多的工作。ASP 调用的 COM 组件大大扩展了 ASP 的功能,可以用任何支持这种标准的工具开发。

ASP 比较明显的几个缺点:第一,最多为人所诟病的是,它依赖 WINDOWS 操作系统和 COM 组件,无法跨平台使用。虽然 Unix 下也有 ChiliSoft 的插件来支持 ASP,但是 ASP 本身的功能有限,必须通过 ASP+COM 的组合来扩充,而 Unix 下的 COM 实现起来非常困难。第二,安全性问题,IIS 的漏洞曾经使许多网站的源程序大曝光,ASP 代码被人下载。第三,ASP 通过 COM 对象来扩充复杂的功能,虽然 ASP 简单易学,但 COM 的开发远比 Beans 复杂和繁琐,这增加了开发高质量 ASP 程序的难度。

PHP(Hypertext Preprocessor)是 HTML 内嵌式的语言,它混合了 C, JAVA 和 PERL。其主要优点为:第一,运行速度快,便于掌握,可以方便地进行维护和管理。第二,开放源码,所有的源码和文档都可以买卖、复制、编译和传播。第三,同 JSP 一样有平台无关性,可跨平台使用。第四,它是完全免费的,对于需要考虑成本的运营方来说,有很大的诱惑力。

其缺点主要为:首先,PHP 缺乏规模支持,缺乏企业级的和正规的商业支持,难于实现商品化应用的开发。其次,缺乏多层结构支持。对于需要分布计算的大负荷站点,数据库、应用逻辑层、表示逻辑层彼此分开,而且同层也可能根据具体需求划分,组成阵列。PHP 缺乏这种支持,难于提供大型电子商务解决方案。第三点是 PHP 提供的数据库接口支持不统一,这极大限制了它的广泛应用。

JSP(Java Server Page)是完全的面向对象和平台无关的。它的主要优点如下:第一,将内容生成与显示分离,这也是 JSP 规范的终极目标。使用 JSP,WEB 开发人员可以使用 HTML 或者 XML 标识来设计和格式化最终页面^[5]。使用 JSP 标识或者小脚本来生成页面上的动态内容。核心逻辑被封装在标识和 JAVABEAN 中,那么其他人,如 web 管理人员和页面设计者,能够编辑和使用 JSP 页面,而不影响内容的生成,他们不需要知道或不用关心内部的具体的实现。

第二，生成可重用组件，开发人员能够共享和交换执行普遍操作的组件，或者使得这些组件为更多的使用者或者客户团体所使用，这是模块化程序设计的优点。第三，JSP 采用标识简化页面开发，它允许开发人员定制自己的 JSP 标签，创建自己的自定义标签库。这样开发者可以让网页制作者避开网页，将重要的功能以标签的形式提供给更多的网页制作者使用。第四，由于 JSP 页面的内置脚本语言是基于 Java 编程语言的，而且所有的 JSP 页面都被编译成为 Java Servlet，JSP 页面就具有 Java 技术的好处，包括健壮的存储管理和安全性。第五，作为 Java 平台的一部分，JSP 拥有 Java 编程语言“一次编写，各处运行”的特点。JSP 组件(如 JavaBeans 组件和自定义的 JSP 标签等)可跨平台的重复使用。

它的主要缺点为：第一，调试困难。编译器和虚拟机只能报告与 JSP 相对应的 Servlet 中的错误，不利于方便的调试和排错。第二，对于庞大的系统，控制代码会变得非常麻烦。纯 java 的工程，编译一次就能把主要的问题暴露出来，但是 JSP 需要把每个 JSP 页面都访问过才能让它全部编译。有一些页面需要满足一定条件才能访问，在调试过程如果改动了已有的 JSP 代码就会隐藏错误，不易发现冲突，这降低了人的工作效率。第三，运行速度慢，必须将 Class 常驻内存，资源耗费严重。

2.5 JSP 应用的配置

本节主要叙述浏览工具开发所需要的软件工具和运行环境等。论文选用 JSP 作为应用开发语言。目前有许多应用服务器支持 JSP2.0 规范，同时也有许多开发工具支持 JSP2.0 开发。作为学习和掌握 JSP2.0 技术为目的，同时也出于个人计算机配置负载能力考虑，我选用 Windows 操作系统平台下的 J2SE SDK(Java 2 Standard Edition Software Development Kit, Java 2 标准版软件开发包)和 Apache Jakarta Tomcat 来搭建 JSP 开发平台(用 Tomcat 做 JSP 容器,负责接收 Web 请求,并将请求传送到 JSP Web 应用),用 MySQL 作为后台数据库,文本编辑工具选用 UltraEdit,对客户端没有要求,只要是遵循 HTTP 协议的浏览器都可以,在这个工具设计过程中使用的是 IE 和 firefox。

第三章 JBrowser 的设计

在这一章中,主要介绍设计并实现的基于 JSP 技术的数据库浏览工具的整体设计部分,包括系统需求确定,体系结构划分,数据流图设计和模块功能划分等。并且将这一基于 JSP 技术的数据库浏览工具顾名思义命名为 JBrowser。

3.1 系统需求

所设计的数据库浏览工具需要具备以下功能:

第一,支持数据库管理员可以将自己管理的数据库发布到 Web 上。这意味着所访问的数据库内容是程序设计人员不可知的,无论是数据库的数量还是数据库中表和视图的结构。并且要求只要数据库管理员在后台发布了新的数据创建了新的数据库或表,或是更改了原数据库的属性或表的结构,访问程序可以不经任何修改就支持有访问权限的用户对数据进行查询。因此需要访问程序有一定的通用性,不是针对特定结构的数据进行查询。

第二,用户通过浏览器可浏览数据库管理员发布的元数据、表和视图。对于元数据的查询,非计算机行业的最终用户没有必要也不会有兴趣去进行查询,所以对这种用户来说,可以屏蔽其对元数据的查询。对于有对元数据查询需求的用户,需要提供对元数据的查询。

第三,用户界面友好,支持表格式查询和表单式查询两种不同的数据表示方式。表格式查询中,显示时支持分页和排序。页面一次显示表中的一行数据,用于元组中字段的字符较多,表格显示时浏览不便,这时以表单查询方式作为辅助。用户可以根据需求在两种表示方式间切换。

第四,用户可以直接输入 SQL 查询语句。对于比较复杂的查询,比如多表连接、复合查询条件、单纯依靠系统提供的功能组合无法完成,所以提供一个用户输入界面,由用户直接输入 SQL 语句查询。直接输入 SQL 语句查询需要注意的事项主要有:

- 1) 权限检查,用户不允许查看没有 QUERY 权限的数据。
- 2) 操作控制,不允许用户进行 UPDATE、DELECT、INSERT、CREATE 等操作,对于以上几种操作,由于浏览工具的用户没有执行权限,系统会拒绝执行。^[6]
- 3) 错误报告:当用户输入语句错误或进行没有权限的操作时,无法执行,提示用户检查语句重新输入。

软件的运行环境是:客户端要求支持使用 Microsoft Internet Explorer 6.0, 和 firefox, 进行设计时注意 firefox 不支持 shtml 和一些其他未标准化的协议或脚本语言。Web 服务器和 JSP/Servlet 容器使用 Tomcat 5.5, 数据库服务器使用 MySQL 5。

3.2 体系结构

在实现与数据存取密切相关,系统应用扩展和修改频繁,但对于客户操作界面等其它功能要求不强烈的情况下,B/S 体系结构是目前非常常用的。在这种结构下,用户工作界面通过 WWW 浏览器来实现,除极少部分事务逻辑在前端 (Browser)实现,主要事务逻辑在 Server 端实现,形成了表示层、业务逻辑层、数据访问层的三层结构,而且三层结构很大程度上是可以相互独立的,任意一层的改变不影响其它层功能的发挥。^[7]这样的好处是:第一,大大减轻了系统维护与升级的成本和工作量,降低用户总体成本支出。第二,接入方式多样,可以通过 LAN(局域网)、WAN(广域网)、Internet/Intranet(企业内部互连网)等不同的方式接入访问和操作共同的数据库。第三,有效的保护数据平台和管理访问权限,提高系统安全性。JSP 三层体系结构示意图如图 3-1 所示。

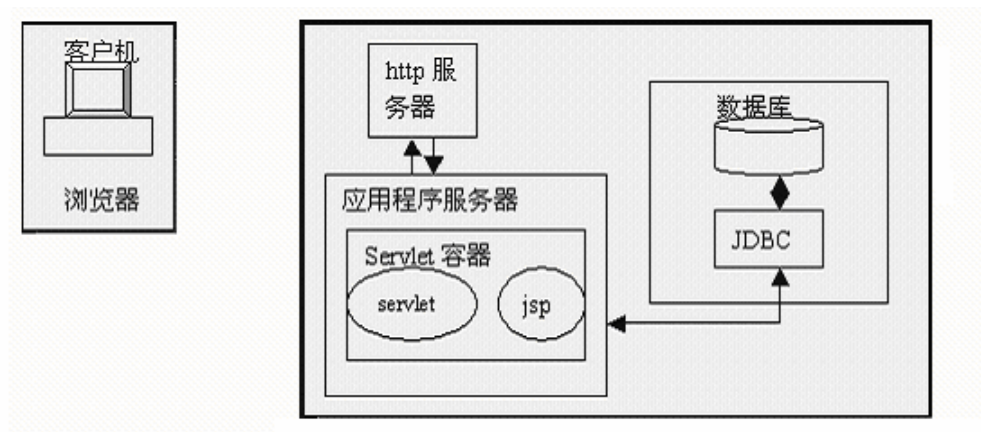


图 3-1 JSP 三层体系结构

表示层：Web 浏览器在表示层中包含系统的显示逻辑，位于客户端。它的任务是由 Web 浏览器向网络上的某一 Web 服务器提出服务请求，Web 服务器对用户身份进行验证后用 HTTP 协议把所需的主页传送给客户端，客户机接受传来的主页文件，并把它显示在 Web 浏览器上。

在业务逻辑层中包含系统的事务处理逻辑，位于 Web 服务器端。它的任务是接受用户的请求，首先需要执行相应的扩展应用程序与数据库进行连接，通过 SQL 等方式向数据库服务器提出数据处理申请，而后数据库服务器将数据处理的结果提交给 Web 服务器，再由 Web 服务器传送回客户端。

在数据访问层中包含系统的数据处理逻辑，位于数据库服务器端。它的任务是接受 Web 服务器对数据库操纵的请求，实现对数据库查询、修改、更新等功

能，把运行结果提交给 Web 服务器。

3.3 数据流图

下面的数据流图描绘数据在系统中流动和处理的过程，反映系统必须完成的逻辑功能，初步规范数据走向，变量应该设置和调用。通过对系统中数据、数据加工的全面分析，勾画出系统的框架。

数据流图分两个部分：一，用户登录认证部分：确定用户是否合法，指定合法用户的权限。对用户权限分进行分级，管理员级别的权限用户拥有对所有数据的查询权力，包括用户数据和元数据。普通用户只能查询由管理员所赋予查询权限的数据，其他数据对普通用户是不可见的。

二，数据查询部分：用户对所查数据的选择，选定所查内容，获得返回结果集的第一页，同时给定下一页的链接，用户可以通过页面间链接查询到所有结果集中的数据。用户还可以输入SQL语句查询，直接执行用户输入语句查询数据库，查询结果的返回方式和通过点击页面链接得到返回集的显示方式一样。另外，用户还可以指定排序方式，依用户指定排序关键字按升序或降序重新查询数据库，并取得结果集，返回给用户具体的数据流图如图3-2所示。

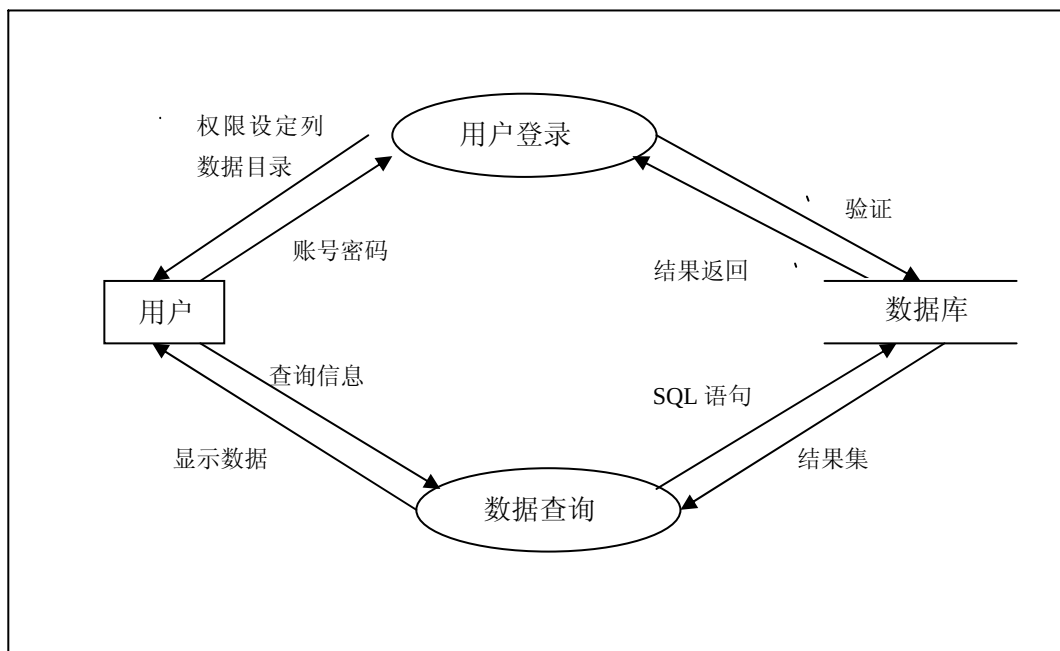


图 3-2 顶级数据流图

用户登录时，如果用户名和密码正确，则返回用户有查询权限的数据库目录，系统跳转到数据查询部分。如果用户名密码验证失败，提示重新输入。由于数据流向较为简单，所以数据流图略去。

数据查询部分的数据流图如图 3-3 所示。

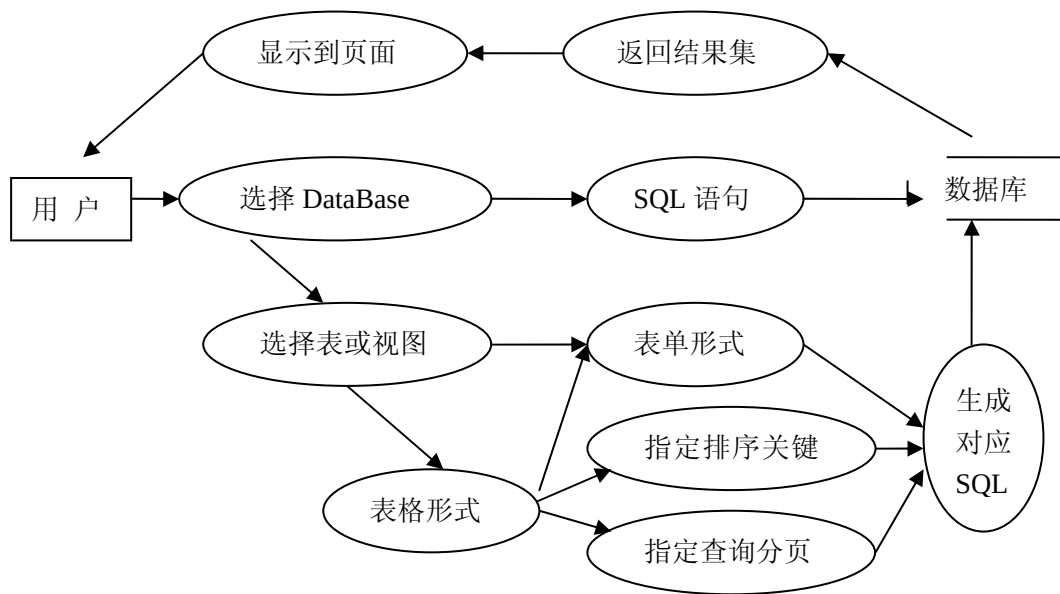


图 3-3 数据查询部分数据流图

3.4 模块划分

这一节描述模块的划分和每个模块的功能，模块划分图如图 3-4，为了使页面逻辑清晰简洁，方便用户使用，将所有的功能性模块放入一个 Frame 框架中展示给用户，所有的 JSP 页面都在这个框架中被调用，当用户通过验证登录到系统之后，体系自动跳转到所示框架。各个模块的功能如下：

数据库信息模块：包含页面为 `databaseshow.jsp`，在用户登录之后，给出用户有查询权限的所有数据库名，并在数据库名上设置超链接，链接内容为选定数据库的表或视图的列表。

表或视图信息模块：开始页面为 `tip.html`，提示用户对数据库进行选择，否则无法查询相应表。当用户点击 `databaseshow.jsp` 上的数据库名时，当前页面跳转到选定数据库的包含的表单 `tableshow.jsp`。对数据库中每个表和视图名建立链接，链接内容为选定表的内容，这里默认的数据查询方式为表格形式查询。

数据显示模块：初始显示用户登录后的信息，当用户选定 `tableshow.jsp` 中的表或视图链接时，跳转到 `resault_of_select.jsp` 显示表的内容，并在表头显示出每一列的列名，当用户第一次点击列名时，当前页跳转到 `resault_of_sort.jsp`，按照升序排列所有数据。用户第二次点击相同列名的时候，当前页跳转到 `resault_of_sort_desc.jsp` 对数据进行降序排列。另外无论是触发排序还是选择表名查看表中的数据，所显示的都是分页之后的第一页页面。用户点击上一页或下一页时跳转到相应页面的指定分页。同时如果用户认为表格形式浏览不方便的话可以选择表单形式浏览，默认返回内容为表格形式浏览的当前页的第一行数据。

SQL 输入模块: 用户可以直接在对话框中输入 SQL 语句, 语句确认无误后, 系统执行用户输入的语句, 由 `resault_of_select.jsp` 显示查询结果。这里要注意的一个问题是对用户输入内容的锁定, 用户输入的只能是符合 SQL 规范的查询语句, 不符合规范的输入和插入、删除、创建等操作将被系统拒绝执行。

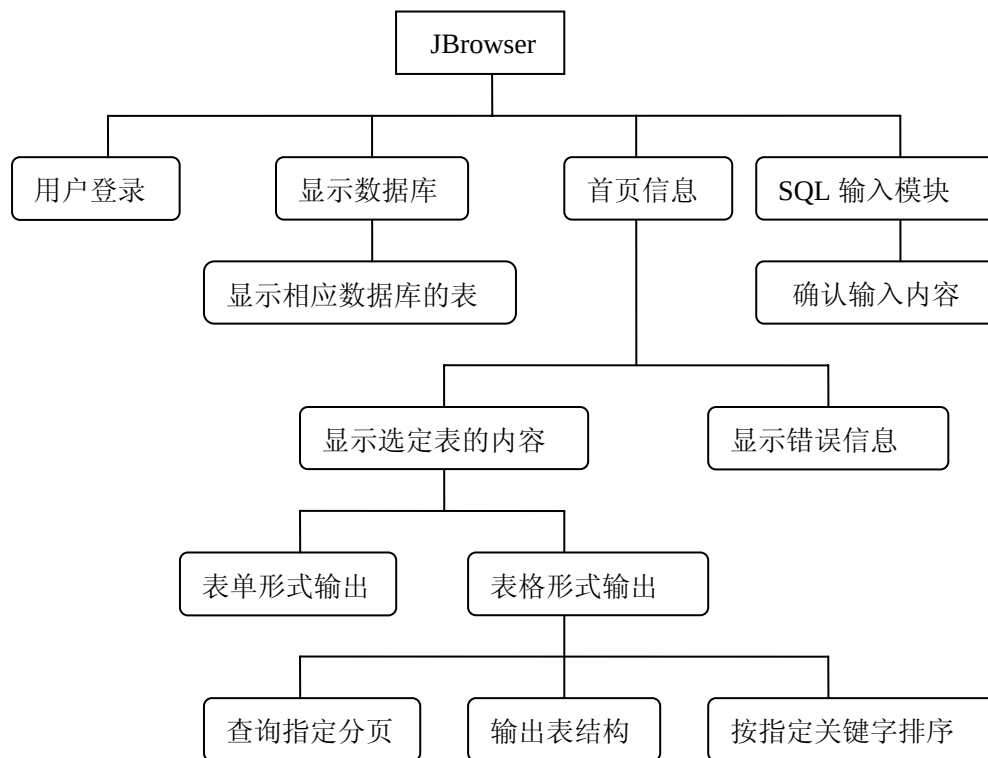


图 3-4 JBrowser 的模块功能划分

第四章 JBrowser 的实现

4.1 Web 应用发布

在系统设计并编码完成, 发布 Web 应用时, 将应用中的所有文件(包括静态 HTML 页面、JSP 页面、Servlet 类、配置描述文件和其它相关文件)打包到一个 War 文件中, 这样既方便了发布和配置, 也可以在配置文件中定义所有的配置参数。

首先在发布前说明, 论文默认在用户安装 JBrowser 之前已经安装了 JDK、Tomcat 和 MySQL, 并且将 JDBC 拷贝到相应的目录中。这三种工具的安装和 JDBC 的添加这里不再赘述。

应用安装的时候, 不需要重新启动 Tomcat, 只要把对应的 War 文件复制到 Tomcat 目录下的 webapps 即可, 它会自动生成相同目录名的应用目录, 这时应用的 URL 为 `http://localhost:8080/warname/filename.jsp`。

在 WEB-INF 文件夹下的 web.xml 文件中有关于 Web 应用配置的选项的详细描述, 与 JBrowser 相关的配置选项包括通用选项和 JSP 选项。

4.1.1 通用选项配置

通用选项与初始化启动文件和使用错误处理页面相关, 具体的配置主要为:

- (1) 将 session 会话的有效时间长度设置为 60 分钟^[8]。

```
<session-config>
  <session-timeout>60</session-timeout>
</session-config>
```

这样将会话的有效时间长度设置为一个小时, 管理员可以按照自己的实际情况修改会话有效时长。

- (2) 指定应用入口文件的文件名为 `index.jsp`。在设定站点访问时最好不让用户直接输入具体的文件名, 用元素 `welcome-file-list` 来指定作为入口文件的文件名:

```
<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

- (3) 指定处理错误的页面。当因为用户输入不符合系统要求出现问题的时候, 指定错误提示页面, 对用户进行提示:

```
<error-page>
  <error-code>500</error-code>
  <location>inputError.jsp</location>
</error-page>
```

完成发布后的系统在用户正常使用时的操作主界面如图 4-1 所示。

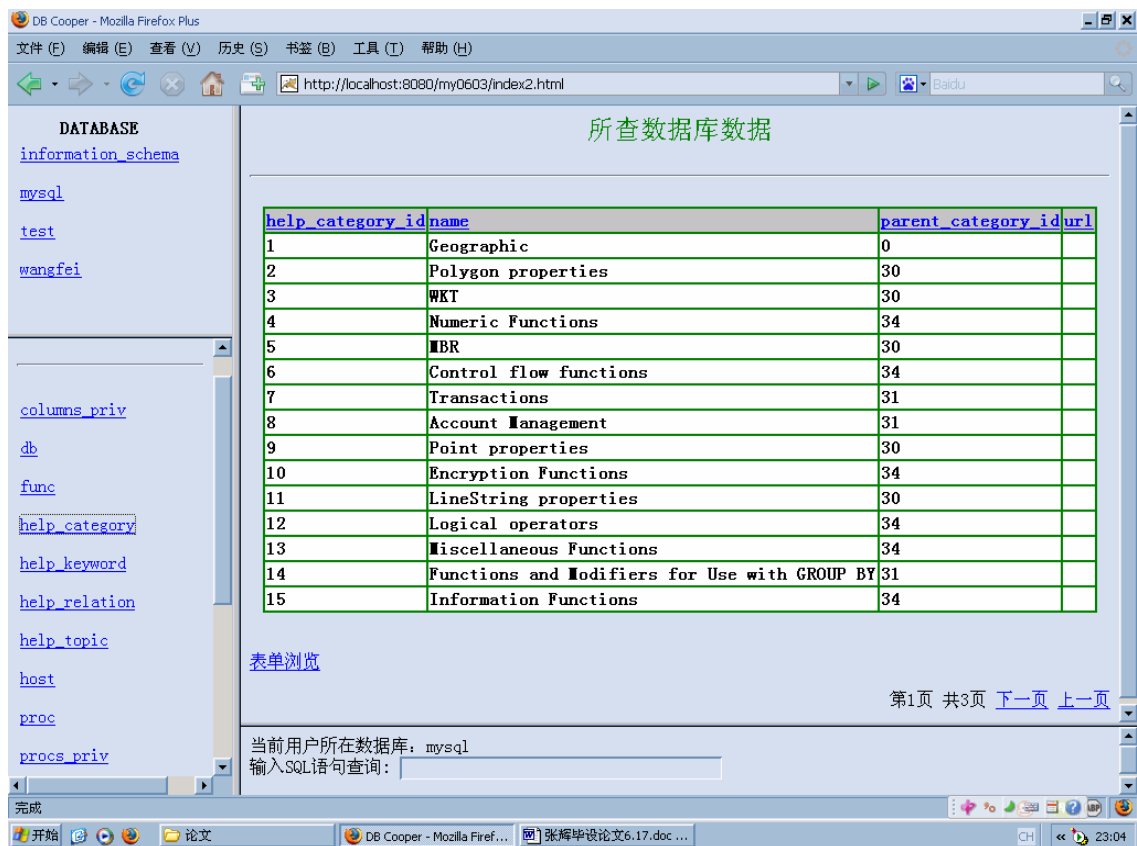


图 4-1 完成登录后用户操作界面示例

4.2 分页浏览

分页显示是 Web 数据库应用中经常需要遇到的问题，当用户的数据库查询结果远远超过了计算机屏幕的显示能力的时候，合理的将数据呈现给用户就成为一个必须考虑的问题，解决这个问题最好的方式就是数据库分页显示。附录有一个数据库分页操作的通用实例，对任何数据库平台上的分页功能都有很好的借鉴意义，本节只要讨论分页时应当注意的几个问题。

第一个问题是分页时，应当如何确定 SQL 语句执行返回的结果集分为多少页。假设 $PageSize$ 为一页显示的记录数， $RowCount$ 为记录总数， $PageCount$ 为总页数，这三个变量都是 int 型的，因为结果集返回的记录最小计数单位是行，同样也不存在半个页，单个的分页只有有或没有。最直观的总页数的计算方法为 $PageCount = RowCount / PageSize$ ，直接用总记录数除以每页显示的记录数来得到所划分的页数。但这带来的问题如下，假设总记录为 3 行，每页显示 2 行(或更广泛一点，总记录数不是每页显示数的整倍)，那么由上面公式计算所得的结果为只需要一页，但这样丢掉了最后一页中的记录。改进的方法是把最后丢掉一页加上来，这样可以找回因为整数除法被舍弃的最后一页，页数的计算方法变成 $PageCount = RowCount / PageSize + 1$ ，但是这又会带来新的问题，假如 $RowCount$

为 `PageSize` 的整倍数, 那么最后加上的这一页实际上是空的, 没有内容。虽然这种方法在编译的时候不会遇到任何的问题, 但用户翻到最后一页发现空空如也, 没效率也影响心情。解决的办法是优化总页数的计算方法: $\text{PageCount} = (\text{RowCount}-1) / \text{PageSize} + 1$ 。这样一来无论 `RowCount` 和 `PageSize` 是不是整倍关系, 都不会出现丢页或空页的情况。

第二个问题是页面边值问题, 如何调整待显示的页码。问题的由来如下: 编码时所设想的情况是在第 N 页时, 点击标示上一页或下一页的链接时, 就能够得到第 $N-1$ 页或第 $N+1$ 页的数据, 但是暴露出来的问题是如果当前页是首页, 上一页的链接如何处理, 或是如果当前页是尾页, 下一页的链接如何处理。这个问题有两种解决方法, 一种是首先判断当前页是不是首页(或尾页), 如果是那么它的上一页(或下一页)的链接置空。另一种方法是, 如果当前页是首页(或尾页)那么人为指定它的上一页(或下一页)链接到何处。比如如果是首页或尾页的话, 可以将链接指定到当页, 又或者是首页将其上一页的链接指向尾页, 对尾页进行相同的处理, 这样形成一个循环。

4.3 排序

当页面载入时, 第一次是按照默认的数据顺序排序的, 一般为依主关键字的升序排序, 或当主关键字排序对用户不需求不重要时, 指定索引字段名 `Table1.IndexFieldName:='xxxx'`; 然后点击表头出发事件, 将排序后的结果集返回页面。还有一个问题是, 当排序和分页同时出现在一个页面的时候, 所排序的对象, 是结果集中所有的记录, 还是只排序当前页面的记录? 如果是排序当前页面的记录, 可以使用 `sort` 方法。但仔细考量过后, 认为当用户需要排序时, 大部分情况是需要对结果集中所有的对象排序。这个时候的做法是: 实现点击表头排序——这个可以再提交一次表单, 然后根据用户点击的表头项信息来生成一个 `order by` 的 `sql` 语句。在排列的时候升序或降序的选择有两种实现方式: 第一种是简单的设定两个页面, 一个升序排列数据, 另一个降序排列, 两个页面在点击表头排序的时候相互调用。第二种实现方式为, 设置变量 `sort`, 用 `sort` 来表示排序显示是升序还是降序(具体实例如下)。当 `sort` 的值为空, 结果集升序排列, 然后设置 `sort` 值为 `DESC`; 当 `sort` 的值为 `DESC` 时, 结果集降序排列, 然后设置 `sort` 值为空。

```
String sql="select * from "+tname+" order by "+key+" "+sort+";";
```

这样就能够解决升序和降序排列的问题。

排序的第二个问题是, 如果需要多个排序条件, 问题如何处理, 比如在一个名为 `Student` 的表中需要查询学号, 姓名, 成绩, 如果想让成绩排列的第一条件, 学号为第二条件怎么办。或者更复杂一点, 让成绩为第一条件降序排列, 学号为第二条件升序排列, 这样表头项信息生成的 `order by` 的 `sql` 语句就要复杂一点:

```
String sql= "select * from student order by id ,name nesc;";
```

因为对数据进行排序的时候需要调用数据库表结构的相关信息,这涉及到了元数据的浏览问题。在 MySQL 中,关于表的表结构、所属数据库、字符集、字段及字段属性等信息都存储在数据库 information_schema 的相应表中,例如所有表级的属性信息都存储在表 TABLE 中,所有字段级的属性信息存储在表 COLUMNS 中。对于表结构的读取,本论文采取了两种方式。

对于管理员权限的用户,这组有对数据库 information_schema 的访问权限,所以这组可以直接访问相关数据库,来查询表的结构信息。对于低权限用户,在查询用户数据需要返回用户数据表的结构时,使用 JDBC 的相关函数完成。关于 JDBC 的具体阐述请参照下一节,这里给出获取表的结构信息所用的函数和相关代码段如下^[15]:

```
Statement stmt=conn.createStatement();
ResultSet rs=stmt.executeQuery(sql);
ResultSetMetaData rsmd=rs.getMetaData();//获取表结构信息
for(int i=1;i<=rsmd.getColumnCount();i++)//获取表的字段总数
{
    out.print(rsmd.getColumnName(i));//输出字段名
    out.print(rsmd.getColumnType(i));//输出字段类型
}
```

4.4 JSP 连接数据库

4.4.1 使用 JSTL 访问数据库

对于 Web 应用来说,获取动态数据往往需要访问关系数据库,尽管对于 Web 应用的设计要求数据库操作的处理应该尽量在业务逻辑层内,但是某些情况下,需要在 JSP 页面直接访问数据库,例如原型设计、测试、小规模或简单应用、开发人员过少等。

JSTL 提供的 SQL 标记库具有与关系型数据库交互的能力,使用 SQL 标记库中简单的标记就能够执行数据库的 DML(数据操纵语言)功能,再结合核心标记库,就可以方便的获取结果集、迭代输出结果集中的数据结果等。

下面以 URL 操作为例简述 JSTL 的标记库访问 SQL,查询数据库标记的一般语法格式为^[9]:

```
<sql:query sql="sqlQuery" //指定查询语句
    var="varName" //指定存储查询结果集的对象变量名
    [scope="|page|request|session|application"] //指定有效范围
    [dataSource="dataSource"] //指定数据源,可预先设定
```

```
[maxRows="maxRows"]
```

//指定查询结果的最大行数, 缺省为-1, 表示没有大小限制

```
[startRow="startRow"]
```

//指定查询结果开始行索引, 缺省为 0, 表示从第一行开始

```
/>
```

实例:

```
<sql:setDataSource
```

```
url="jdbc:mysql://localhost:3306/user"driver="com.mysql.jdbc.Driver"
```

```
user="username"password="012345"var="ds">
```

```
<sql:query sql="select * from userData" var="results"dataSource="${ds}">
```

设定数据源路径后, 获得授权的用户在数据库中查询用户信息中的信息。

4.4.2 使用 JDBC 访问数据库

JDBC(Java Database Connectivity, Java 数据库连接)是 Java 应用中访问表类型的数据源的应用程序设计接口(API)。这些数据源包括基于 SQL 的数据库、存储在电子数据表中的数据和存储在文件中的数据等。大多数数据库都提供 JDBC 驱动程序来支持 JDBC 访问数据库中的数据^[3]。

JDBC 提供的 API 允许开发人员连接数据库、执行 SQL 语句、操纵数据库数据、返回结果集等。对应于特定的数据库的 JDBC 驱动程序还提供一些高级功能, 比如支持数据库连接池等数据库资源的使用。

JSP 页面常用的两个访问 MySQL 数据库的 JDBC 驱动为 MySQL Connector/J 和 CauCho Resin 的 MySQL 驱动, 选择前者进行系统的安装和配置。MySQL Connector/J 驱动程序是一个将 JDBC 调用转换为 MySQL 数据库使用的网络协议的本地 Java 驱动程序, 允许开发人员在 Java 应用中方便的访问数据库中的数据。

下面介绍 JDBC 使用的一般步骤, 首先连接数据库创建一个新的 JDBC 连接, 例如数据库为 myDB, 连接数据库使用的语句如下:

```
Class.forName("com.mysql.jdbc.Driver").newInstance();
```

```
Con=DriverManager.getConnection("jdbc:mysql://localhost:3306/myDB?
```

```
user=username&password=13579
```

```
&useUnicode=true&characterEncoding=GB2312")
```

之后创建一个用于包含需要在数据库中执行的 SQL 语句对象, 用字符串来指定 SQL 语句, 并将结果集赋值给一个指定的变量, 范例如下:

```
stmt = con.createStatement();
```

```
String slct = "select id,name,email from userData where age>60; ";
```

```
rs = stmt.executeQuery(slct);
```

最后在使用完 SQL 返回结果之后, 使用数据库连接的 close()方法释放当前

连接。

```
rs.close();
```

需要说明的是,出于代码重复使用和系统安全的原因,一般不推荐在 JSP 页面的代码中直接访问数据库。当需要对数据库访问时,最好使用 JavaBean 或 JSTL 标记来实现。

4.5 参数传递

JSP 页面之间传递参数的方式有很多,这里选择几种主要的传参方式简要介绍,包括在 URL 当中加入变量传参,使用 JSP 对象传参,FORM 表单传递变量和使用 JSP 动作元素。

4.5.1 URL 传递参数

利用 URL 传参的基本方式,在需要传出参数的页面,其 URL 格式为:

```
<A href="page.jsp?paramer1=value1&paramer2=value2&.....">
```

接受参数的页面接受参数的相应操作为:

```
<%
```

```
String param1=request.getParameter("paramer1");
```

```
String param2=request.getParameter("paramer2");
```

```
.....
```

```
%>
```

依次接受由上一个页面传递来的参数。这里需要注意几个在传递参数中常见的几个问题。

第一个是 URL 的格式问题,也是我在编码中所犯的错误中最耽误时间和无聊的一个。当 URL 需要传递的参数非常多,且参数是由 JSP 的脚本元素表达式(<%= %>)或 out.print();输出到 URL 的时候,整个 URL 就会变得非常长和繁琐,为了便于维护和管理,很多人的第一想法是给 URL 分行,变成如下所示的形式,

```
<A href="page.jsp?paramer1=value1
      &paramer2=value2
      &.....">
```

这样一来就会变得一目了然,非常方便。而且因为 URL 符合 HTML 语法规则,HTML 中,对于回车换行是完全忽略的。但是实际是当参数传递到接受页面之后,在很多参数后面多了一段恼人的空格。检查出来的原因是所用的 UltraEdit 是换行的时候自动缩进的,变量和分割参数的&之间并没有空格,但是在调试中所用的文本编辑器没用相同的功能,造成了不应该出现的错误。

另一个问题是关于 URL 中传递参数包含中文的问题。很多情况下传递中文参数的时候经常出现乱码,引起问题的原因和解决方式主要有以下几个:

(1)确定 JSP 页面头部是否有如下代码段:

```
<%@ page contentType="text/html; charset=GBK" %>
```

并查看各个页面的 charset 是否统一。

(2)如果出于需要或其它原因不能统一 charset 的话,可以在接受页面对收到的变量转码,方法如下:

```
String param=
```

```
new String(request.getParameter("param").getBytes("ISO-8859-1"),"GBK");
```

(3)在某些情况下,需要通过"a.jsp?param=中文"传递参数,这时候需要以下操作:

a.在传参数之前先用 `java.net.URLEncoder.encode(param)`把参数进行转码取值用 `java.net.URLDecoder.decode(param)`再转回中文。

b.在你的 Tomcat 目录-->conf 目录-->server.xml 里找出这段:

```
<Connector
```

```
port="8080" maxThreads="150" minSpareThreads="25"
```

```
.....
```

```
/>
```

在里边加参数:`URLEncoding="gb2312"`这样可以直接用 GB2312 编码进行汉字参数传递^[12]。

4.5.2 JSP 对象传递参数

JSP 页面在处理请求的时候可以创建和访问 Java 对象,JSP 按照作用范围持续时间不同,可以分为页内有效、请求有效。会话有效和应用有效 4 种不同的有效范围。JSP 页面中使用上下文对象方法 `setAttribute()`和 `getAttribute()`来设置和获取对象属性。

页内有效是指对象创建后只能在当前的 JSP 页面内访问,页内有效的对象在响应发送回客户端后失效,有效范围只是当前页。例如:

```
<%
```

```
pageContext.setAttribute("name","zhanghui");
```

```
String theName;
```

```
theName = pageContext.getAttribute("name");
```

```
%>
```

这就是一个简单的页内有效事例代码段。通过 `pageContext.setAttribute()`设置的属性可以被 `pageContext.getAttribute()`获取并赋值给指定的变量,如果在其它页面使用对象方法获取属性,那么参数值并不会传递过去(都为 null)。

请求有效的对象是指,在处理同一个请求时,不同 JSP 页面创建的对象在这些页面中都是可用的,其有效区间为从被创建开始到请求处理结束为止。假设在属性设置页面里设定

```

<%
    request.setAttribute("name","zhanghui");
%>
然后跳转到属性获取页面
<jsp:forward page="requestScopeGet.jsp"/>
然后在属性获取页面设定为
<%
    String theName = request.getAttribute("name");
%>

```

这样参数就可以从方便地一个页面被传递到另一个页面使用。

会话是指客户端和服务端之间持续连接的一段时间,在这段时间内,客户端需要多次和服务端交换信息,方法是把相应的信息存入 session 对象中。session 对象是针对某个用户的会话过程的信息存储,对于每一个 session 都在服务器端对应一个唯一的 ID,当 session 在会话控制中不需要的时候,可以用 invalidate() 方法使 session 失效或当 session 的有效时间过后自动失效,这样原来会话中存储的所有对象都不能被访问。会话有效的相应代码段和请求有效相似,把 request 换成 session 即可。

对于一个应用中不同请求的 JSP 页面创建的对象,如果在这些页面当中都是可用的,这些对象就是应用有效的。应用有效范围最大、影响最长,它的作用范围是从服务器开始执行服务直到服务器结束服务为止。但应用有效应该严格的控制使用范围,一面造成服务器的负载过高。类似的和前面三个相似,同一个应用不同页面间可以通过对象方法 application.getAttribute() 和 application.setAttribute()来设置和获取属性^[11]。

4.5.3 其它传递参数的方法

常用的传递多个参数的方法是使用 form 表单传参,表单定义的一般格式为:

```

<form action="actionPageUrl" method="post">
定义表单控件.....
<input type="submit">
</form>

```

action 属性指定当前用户完成表单输入并提交时执行的动作,一般是服务器端的处理程序(如一个 JSP 页面)的 URL,处理用户提交的数据并生成反馈给浏览器客户端。method 属性指定数据发送方式: GET 还是 POST。使用 GET 方式,查询串的信息会传送到浏览器的地址栏,信息中汉字转换为 ASCII 格式传输。POST 以隐含方式在 HTTP 请求中包含“名-值”对数据,查询串信息不出现在浏览器的地址栏上。

另一个较为常见的传参方式是使用动作元素中的<jsp: forward>, 这个动作可以将浏览器重定位于同一个网站的其它网页或文件。当 <jsp: forward>有 body 内容以指定参数时, 其语法规则如下:

```
<jsp:forward    page="filename.jsp">
<jsp:param    name="参数名 1"    value="参数值 1"/>
.....
<jsp:param    name="参数名 n"    value="参数值 n"/>
</jsp:forward>
```

这样就可以从当前页将所需参数传递到目的页面。

第五章 总结

基于 JSP 技术的数据库浏览工具 JBrowser 可以支持管理员通过后台管理数据库, 不需要改变 JSP 代码就能修改后的数据库进行查询, 支持用户直接输入 SQL 语句查询, 可以对数据按任意指定关键字进行升序或降序排列, 同时支持分页。但是还存在着不足, 在应用功能及技术改进这两方面还需进一步完善。

在程序的应用功能方面需要改进的主要有: 使用更加人性化的树形结构代替左侧的分块导航; 增强通用性, 将更多种类的数据库的查询操作纳入其中; 增强查询定义的种类, 实现对多表的查询操作、丰富条件定义的样式等; 提供对查询结果的更改操作等。

在技术改进方面, 主要有: 进一步加强代码的复用率, 提高代码效率; 可使用前端排序, 以节省时间和系统资源; 使用现有的成熟框架体系, 减少开发难度等。从使用者的角度来进一步增强用户界面的友善和扩大应用功能的覆盖。

参考文献

- [1]Barry D, Cristina L. 2005 Survey of Technologies for Web Application Development[J]. University of California, Irvine, 2005,6: 4—5.
- [2]林 琪,朱涛江. MYSQL 权威指南(第二版)[M]. 北京: 中国电力出版社, 2003. 57—72.
- [3]张文静, 林 琪. JSP 程序设计[M]. 北京: 人民邮电出版社, 2006. 129—136.
- [4]孙卫琴, 李洪成.Tomcat 与 Java Web 开发技术详解[M].北京: 电子工业出版社, 2004.198—206.
- [5]R Acerbis, A Bongio, S Butt. Architectural Issues and Solutions in the Development of Data-Intensive Web Applications[C]. 22100 Como, Italy. 7—8.
- [6]屈 霞.基于JSP的数据库应用程序安全访问控制设计[J].电脑开发与应用, 2005,18(6):15—16.
- [7]Oliver D,Morrison M.HTML 与 CSS 经典(第 7 版)[M].北京: 人民邮电出版社, 2007. 144—147.
- [8]解文明. 基于JSP技术的Web数据库系统的研究与开发[J]. 电脑与信息技术, 2005, 13 (4): 42—43.
- [9]谢 俊, 陈 明, 付 蝶. 基于 Java 的数据库连接池的研究与实现[J]. 中国科技信息, 2007,7(4): 135—137.
- [10]舒红平, 周定文, 何 嘉等.Web 数据库编程—Java[M].西安: 西安电子科技大学出版社.2005.168—179.
- [11]陈义彬,吴保国.基于 JSP 的 Web 应用中数据库访问技术及其比较[J].计算机与现代化,2005,12(6): 60—63.
- [12]陈欣, 缪天鹏. 基于JSP动态网站的建设[J]. 计算机与数字工程, 2004, 32(4): 94—96.
- [13]David H. Jsp, Servlet, and MySQL[M].Hungry Minds,Inc, 2001.41—44.
- [14]Hans B. Java Server Pages,third edition[M].OReilly.2003.75—76.
- [15]Hall M.Core Servlets and Java Server Pages[M].Prentice Hall PTR.2003.169—174.
- [16]Raghu.R,Johannes .G. Database Management Systems[M].北京: 清华大学出版社, 2003.76—79.
- [17]黄伟. 在 JSP 中使用连接池优化数据库访问效率[J]. 计算机应用, 2002, (4): 67—69.

致 谢

在此感谢我的毕设指导老师张坤龙副教授对我不厌其烦的指导和鼓励,在毕业设计和论文的完成过程中,我从张坤龙老师那里学到了很多,包括专业知识和专业知识之外的为人处事。从开始毕设的指导到后来的论文修改批注和最后的翻译校对,张老师付出了大量的精力和时间,这从论文中密密麻麻的批注和翻译里逐字逐句的勘误中能体会的到,这使我大为感动。感谢张坤龙老师的时间和精力的付出。

另外感谢与我同组进行毕业设计的李宇鹏同学,和多次就毕设相关问题与我进行交流的宫平和刘洋两位同学,谢谢你们对我的帮助和提示。

再次感谢张坤龙老师。