编写本功能说明书是为了重点说明本网站系统软件的结构,数据库建设及实施方案等内容.本说明书将作为系统后续开发工作的指南,也将是系统维护的重要参考资料.

数据库设计

根据对系统所做的需求分析、系统设计、规划出本站相关数据表有：管理员表、分类表、商品表、会员表、相册表等：

注册信息表 **uers\_registry**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 允许非空 | 自动递增 | 备注 |
| user\_id | tinyint unsigned | NO | YES | 主键 |
| username | varchar(30) | NO |  | 用户名称，由2-8字符的中英文组成 |
| password | varchar(32) | NO |  | 管理员密码 |
| email | varchar(60) | NO |  | 邮箱 |

分类表imooc\_cate (以后设计成无限极分类)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 默认值 | 允许非空 | 自动递增 | 备注 |
| id | int unsigned |  | NO | YES | 主键 |
| cName | varchar(30) |  | NO |  | 分类名称 |

商品表imooc\_pro

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 默认值 | 允许非空 | 自动递增 | 备注 |
| id | smallint unsigned |  | NO | YES | 主键 |
| pName | varchar(255) |  | NO |  | 商品名称 |
| cId | int unsigned |  | NO |  | 所属分类ID |
| pSn | varchar(50) |  | NO |  | 商品货号 |
| pNum | int unsigned | 0 | NO |  | 商品库存 |
| mPrice | decimal(10,2) |  | NO |  | 市场价 |
| iPrice | decimal(10,2) |  | NO |  | 网站价 |
| pDesc | mediumtext |  | YES |  | 商品简介 |
| pImg | varchar(255) |  | NO |  | 商品图片 |
| pubTime | int unsigned |  | NO |  | 商品上架时间 |
| isShow | tinyint(1) | 1 | NO |  | 商品是否上架 |
| isHot | tingyint(1) | 0 | NO |  | 商品是否热卖 |

会员表：imooc\_user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 默认值 | 允许非空 | 自动递增 | 备注 |
| id | int unsigned |  | NO | YES | 主键 |
| username | varchar(30) |  | NO |  | 会员名称 |
| password | char(32) |  | NO |  | 密码 |
| sex | enum(“男”,”女”,”保密”) | 男 | NO |  | 性别 |
| email | varchar(60) |  | NO |  | 邮箱 |
| face | varchar(50) |  | NO |  | 用户头像 |
| regTime | int unsigned |  | NO |  | 注册时间 |
| activeFlag | tinyint(1) | 0 | NO |  | 是否激活 |

相册表imooc\_album

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名称 | 数据类型 | 默认值 | 允许非空 | 自动递增 | 备注 |
| id | int unsigned |  | NO | YES | 主键 |
| Pid | int unsigned |  | NO |  | 对应商品id |
| albumPath | varchar(50) |  | NO |  | 商品图片 |

社交网的一个显著特点是支持巨大用户数,例如Facebook支持超过3亿的用户,其数据中心运行着超过万台的服务器,为遍布全球的用户提供信息通讯服务。另外,任何两个社交网用户都可能交互,也就是必须支持任何两个数据库用户的数据关联操作。这对于服务端的数据库管理提出了极大的挑战。

关系数据库与 NoSQL 数据库

关系数据库使用者遵循一些数据库范式,这些范式是数据库设计中的一系列原理和技术,目的是为了减少数据库中数据冗余和增进数据的一致性。结构化查询语言SQL大量使用多表连接操作,SQL的通用性可以为使用者带来很多方便。随着越来越多大规模工作负荷应用的发行,对可伸缩性的需求,可能会变得非常迅速和无比庞大。关系数据库的确能伸缩自如,但通常只能在单台服务器节点上进行。例如采用表分区技术,一个表格可以由多个物理文件组成,虽然表格的容量增大了,但该表格仍然只能由一数据库引擎管理;另外增加一物理文件时,表格Schema得做改动,也就是还不能支持动态扩容。 一旦单节点的能力抵达上限,就得通过多服务器节点来扩展来分发负载。这时关系数据库的复杂性就开始影响其潜在的扩展规模了。RDBMS支持分区视图(Partition View) 技术,也就是支持联合数据库(Federated Database)(如图1)。一个分区视图可以由多个分布在不同数据库节点服务器上的表格组合而成,数据库用户看到的只是该视图,不必关心物理表格。通过数据水平分割技术,分区视图把负载分担到多个数据库节点服务器上。扩容时,该方法除了需改动视图定义外,分区视图成为分布式数据库系统的中心,存在单点故障问题。另外,跨数据库节点之间多表格间连接操作的支持出现极大困难。

当试图扩展数据库系统到成百上千个节点时,将导致不堪复杂性之重负,这一特点使得RDBMS在大型分布式系统平台市场里的生存能力被大幅削减。

为了能向客户提供一个伸缩自如的空间存放应用数据,供应商实际上只有一种真正的选择——实现一种新型的专注于可扩展性的数据库系统,而牺牲掉关系数据库所带来的其他好处。NoSQL是非关系型数据存储的广义定义,它打破了长久以来关系型数据库与ACID理论大一统的局面。NoSQL数据存储不需要固定的表结构,通常也不存在连接操作,在超大型数据存取上具备关系型数据库无法比拟的性能优势。

1. 环境设计

2.1 硬件环境

WEB服务器：用于发布网站、存放网站文件；

数据库服务：用于安装ORACLE数据库

应用服务器：用作后台数据库连接（出于数据安全考虑）以及中间件应用等。

备份服务器：用于进行网站内容的备份

2.2 软件环境

服务器端操作系统：

CentOS 7.0 ;

网站发布平台：

Tomact 7.0;

数据库系统：

Oracle 11g或MongoDB;

网站开发语言：

HTML、CSS、JavaScript、Java、PHP、SQL、Shell；

安全措施：防火墙及安全系统

浏览工具：

Chrome、FireFox、Internet Explorer、Opera

3.产品技术线路

为了确保系统整体运行的效率和稳定性，发布系统采用在RedHat Linux 平台之上，通过Java做为底层的研发平台编制而成。各模块之间通过标准的JSON协议，采用Socket通讯方式进行通讯。

系统的界面均采用B/S结构实现，便于系统的操作员管理员对系统的操作、维护与管理。

详细列表如下：

|  |  |  |
| --- | --- | --- |
| 技术环节 | 采用技术 | 备注 |
| 操作系统 | Red Hat linux 9.0 | * 运行比较稳定，受病毒侵害极小，费用较低。 |
| 底层开发技术 | JAVA语言 | 运行效率极高，可长时间稳定运行 |
| 界面开发技术 | J2EE | 目前常用的编程技术 |
| CGI |  |
| 中间件 | Tomcat |  |
| Web服务器 | Apache |  |
| 数据库 | ORCALE9.2 |  |
| 服务器 | INTEL系列服务器  SUN系列服务器 | 2PIII400CPU、1G内存、40G硬盘  2SPARC400CPU、1G内存、40G硬盘 |

内容发布一般情况下是直接到Web服务器，针对某些特殊情况，提供组件支持移动设备的内容发布。

系统底层通过JDBC/ODBC接口，可在任何数据库上运行。一般情况下，系统支持Oracle和SQL Server。从整体流程上，可以用下图来表示：

由于工作性质的不同，数据来源是非常多样化的。从简单的格式化数据、到复杂的电子文档、视频音频流，甚至有可能是某一种序列。通过数据处理，对各种来源进行整理、加工、分析挖掘，最后通过发布服务器在Intranet、Internet、移动网络或者工控总线中发送各种格式化的信息。



数据来源



数据处理



发布服务器



电子文档

图文信息

视频、音频

纸质报表

为了满足各种数据格式，系统基于XML/XSL，让信息内容和显示形式分离。具体结构如下图所示：



从图示的模型可以看出，系统能够获取不同的数据源并且能够与其它系统接口，整个信息系统都可以基于XML交互。在此之后辅以XSL，就可以按照不同的方式进行发布。从而实现Any Content from Anywhere to Everywhere。

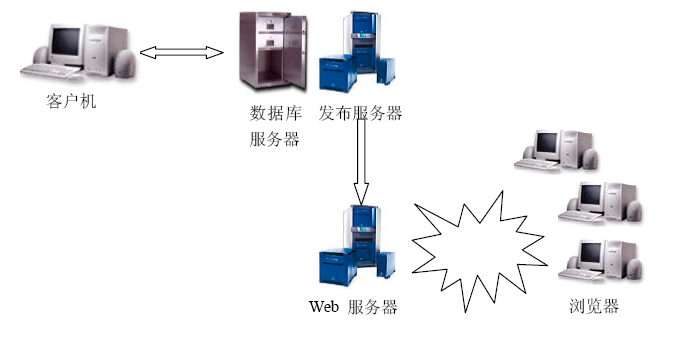
XML、XSQL、XSL是整个9系统构架的基础，它使系统拥有非常强大的灵活性和扩展性。由于基于XML，系统可以非常方便地把数据发布到整个Web中，包括通常的Intranet、Internet，甚至还可以把信息发送到PDA、WAP等其他形式的网络中。正是这样，能够实现通用内容的不同发布。如下图



三、产品部署方式

本系统采用Java语言编写，不论何种服务器配置何种操作系统，只要有Java虚拟机就可以使系统运行。也就是说，服务器和操作系统没有限制。

由于Java采用JDBC数据库接口，系统对数据库并没有限制。但为了充分发挥系统的性能，数据库采用了Oracle9i。用户可以提出数据库移植的要求，如SQL Server、Sybase等。但我们推荐使用SQL Server或者Oracle9i。

系统的各个组件可以分布式运行也可以集中在一台服务器上，具体情况要是看用户的要求，尤其是特殊领域的内容发布的时候。一般情况下，数据库服务器的配置较高一些。下图是该系统的一种网站拓扑模型：

在这个系统中：

对工作人员的机器要求不高，只要能够使用浏览器就可以了。如果有额外的使用要求，由所安装的客户端软件决定。

服务器1是运行系统的主要运行机器，在这台机器上执行几乎所有的功能。为了让系统稳定运行，服务器至少要达到PⅡ400、256兆内存的PC机配置。

数据库1是整个系统的主数据库，该数据库不仅要保存数据，还要执行数据快照、数据备份等操作。它的好坏将决定整个系统的性能，推荐使用双XeonCPU+1G内存机器以上配置的服务器。如果使用Oracle数据库，可以考虑使用SUN服务器。

服务器2是网站的WEB服务器，一般情况下根据访问量决定其配置。如果访问量很大，就需要是专门的服务器甚至是服务器阵列。

数据库2是网站供外部访问的数据库，主要提供查询、流量统计等数据的存储。其配置在满足数据库的一般配置情况下，根据访问量决定。

需要说明的是，有的时候为了优化速度，主数据库甚至后台管理的服务器都可以放在网站上，这样做会对后台管理有一定的负面影响——主要是受网络带宽的影响，但信息发布的速度，尤其是HTML页的生成会有很大的提高。

投标商代表签字

投标单位盖章：

注：投标商应尽可能提供详细说明，以便评委会对投标方案有详细了解。如因提供内容不详导致投标商案被否决，责任由投标商自负。