

# 第 29 章 数据库设计说明书

www.brophp.com

## 内容管理系统(BroCMS)

文件状态： [ ] 草稿 [ ✓ ] 正式发布 [ ] 正在修改	文件标识：	LAMP 兄弟连-BroCMS-02-DATABASE
	当前版本：	2.0
	作 者：	兄弟连
	完成日期：	2011-11-15

## 版本历史

版本/状态	作者	参与者	起止日期	备注
1.0	兄弟连	教学组成员	2009-11-05 2009-11-20	《细说 PHP》第一版
2.0	兄弟连	教学组成员	2012-03-15 2012-03-30	《细说 PHP》第二版

## 29.1 引言

在使用任何数据库之前，都必须设计好数据库，包括将要存储的数据的类型，数据之间的相互关系以及数据的组织形式。数据库设计是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库及其应用系统，使之能够有效地存储数据。在 CMS 项目中总是需要处理大量的数据资源，这正是内容管理系统的基础和核心，为了合理地组织和高效率地存取数据，目前最好的方式，就是建立数据库系统，因此在系统的总体设计阶段，数据库的建立与设计是一项十分重要的内容。由于数据库应用系统的复杂性，为了支持相关程序运行，数据库设计就变得异常复杂，因此最佳设计不可能一蹴而就，而只能是一种“反复探寻，逐步求精”的过程，也就是规划和结构化数据库中的数据对象以及这些数据对象之间关系的过程。

### 29.1.1 编写目的

一个成功的管理系统，是由：[50% 的业务 + 50% 的软件] 所组成，而 50% 的成功软件又有 [25% 的数据库 + 25% 的程序] 所组成，数据库设计的好坏是一个关键。如果把企业的数据库比做生命所必需的血液，那么数据库的设计就是应用中最重要的一部分，是一个系统的根基。在 BroCMS 内容管理系统的需求分析和系统概要设计的基础上，对数据进行分析 and 结构上进行设计。用于开发人员进行项目设计，以此作为编码的依据，同时也为后续的数据库维护工作提供了良好的使用说明，也可以作为未来版本升级时的重要参考资料。数据库设计的目标是建立一个合适的数据模型。这个数据模型应当是满足用户要求，既能合理地组织用户需要的所有数据，又能支持用户对数据的所有处理功能。也要满足 CMS 数据库管理系统的要求，又能够在数据库管理系统中实现。并且要具有较高的范式，数据完整性好，效益高，便于理解和维护，没有数据冲突。

### 29.1.2 背景

名称	说明
数据库名称	BroCMS（兄弟连内容管理系统）
数据库系统	MySQL5.0
客户端连接工具	MySQL Command Line Client
项目任务提出者	LAMP 兄弟连

<b>项目开发者</b>	兄弟连
<b>使用用户</b>	使用用户：《细说 PHP》读者及 LAMP 兄弟连学员

**注：**些数据库设计说明书文档范围只适用于内容管理系统 BroCMS V2.0，作为 WEB 程序员项目设计和学习的参考文档。

### 29.1.3 定义

**CMS：** Content Management System，内容管理系统

**E-R 图：** 实体关系图

### 29.1.4 参考资料

- A. 《细说 PHP》教程
- B. 《BroCMS 项目需求分析说明书》
- C. [www.brophp.com](http://www.brophp.com) 和 [bbs.lampbrother.net](http://bbs.lampbrother.net)
- D. 本项目相关的其他参考资料。

## 29.2 外部设计

外部设计是研究和考虑所要建立的数据库的信息环境,对数据库应用领域中各种信息要求 and 操作要求进行详细地分析,了解应用领域中数据项、数据项之间的关系和所有的数据操作的详细要求,了解哪些因素对响应时间、可用性和可靠性有较大的影响等各方面的因素。

### 29.2.1 标识符和状态

**数据库表前缀：**bro\_

**用户名：**root

**密码：**123456

**权限：**全部

**有效时间：**开发阶段

**说明：**系统正式发布后，可能更改数据库用户/密码，请在统一位置编写数据库连接字符串，在发行前请予以改正。

## 29.2.2 使用它的程序

本系统主要利用 PHP 作为前端的应用开发工具,使用 MySQL 作为后台的数据库, Linux 或 Windows 均可作为系统平台。

## 29.2.3 约定

- 所有命名一定要具有描述性,杜绝一切拼音、或拼音英文混杂的命名方式。
- 字符集采用 UTF-8, 请注意字符的转换。
- 所有数据表第一个字段都是系统内部使用主键列,自增字段,不可空,名称为: id, 确保不把此字段暴露给最终用户。
- 除特别说明外,所有日期格式都采用 int 格式,无时间值。
- 除特别说明外,所有字段默认都设置不允许为空, 需要设置默认值。
- 所有普通索引的命名都是表名加设置索引的字段名组合,例如用户表 User 中 name 字段设置普通索引,则索引名称命名方式为 user\_name;

## 29.2.4 支持软件

操作系统: Linux / Windows

数据库系统: MySQL

查询浏览工具: PHPMyAdmin

命令行工具: mysql

注意: mysql 命令行环境下对中文支持不好,可能无法书写带有中文的 SQL 语句,也不要使用 PHPMyAdmin 录入中文。

## 29.3 结构设计

数据库的结构设计中有许许多多需要考虑的因素,如数据库的背景、应用环境等方面都需要有深入的了解,只有一个对所有这些因素都很了解的数据库设计专家,他设计的数据库才能易于使用和维护,并且具有高效和一致的特征。虽然这样只对数据库设计过程有一个概要的了解,但是仍然有助于读者了解和掌握 SQL,使读者可以很好地分析数据间的相互关系,在使用 SQL 进行报表的生成、子查询及视图等操作时,可以更好地进行操作。

### 29.3.1 概念结构设计

概念数据库的设计是进行具体数据库设计的第一步,概念数据库设计的好坏直接影响到逻辑数据库的设计,影响到整个数据库的好坏。在 BroCMS 系统的分析阶段,我们已经到了系统的数据流程图和数据字典,现在就是要结合数据规范化的理论,用一种模型将用户

的数据要求明确地表示出来。概念数据库的设计应该极易于转换为逻辑数据库模式，又容易被用户所理解。概念数据库设计中最主要的就是采用实体-关系数据模型来确定数据库的结构。数据是表达信息的一种重要的量化符号，是信息存在的一种重要形式。数据模型则是数据特征的一种抽象。它描述的是数据的共性，而不是描述个别的数据。一般来说，数据模型包含两方面内容。

(1) 数据的静态特性：主要包括数据的基本结构、数据间的关系和数据之间的相互约束等特性。

(2) 数据的动态特性：主要包括对数据进行操作的方法。

在数据库系统设计中，建立反映客观信息的数据模型，是设计中最重要、最基本的步骤之一。数据模型是连接客观信息世界和数据库系统数据逻辑组织的桥梁，也是数据库设计人员与用户之间进行交流的共同基础。概念数据库中采用的实体-关系模型，与传统的数据模型有所不同。实体-关系模型是面向现实世界，而不是面向实现方法的，它主要是用于描述现实信息世界中数据的静态特性。而不涉及数据的处理过程。但由于它简单易学，且使用方便，因而在数据库系统应用的设计中，得到了广泛应用。实体-关系模型可以用来说明数据库中实体的等级和属性。以下是实体-关系模型中的重要标识：

- 在数据库中存在的实体
- 实体的属性
- 实体之间的关系

### 29.3.1.1 实体和属性的定义

按照定义的数据类型和属性创建实体和实体属性列表。实体形成表，如“用户”就是一个实体，属性则为表中的列，如对应于实体“用户”属性包含“用户名”、“用户 ID”等。

#### ➤ 实体

实体是实体-关系模型的基本对象，是现实世界中各种事物的抽象。凡是可以相互区别开并可以被识别的事、物、概念等对象均可认为是实体。在本书示例的简单的 BroCMS 数据库中，基本的实体列表如下：

- 相册
- 栏目
- 图片
- 文章
- 幻灯片
- 评论
- 用户组
- 用户
- 站内信
- 公告
- 友情链接

- 动态

在绘制实体-关系图(E-R 图)时，实体出现在矩形中。如图 29-1 所示。



图 29-1 表示实体的 E-R 图

一般来说，每个实体都相当于数据库中的一个表。上面介绍的实体都是强实体，每个实体都有自己的键。但是在实际领域中，经常存在一些实体，它们没有自己的键，这样的实体称为弱实体。弱实体中不同的记录有可能完全相同，难以区别，这些值依赖于另一个实体(强实体)的意义，必须与强实体联合使用。在创建了实体之后，就可以标识各个实体的属性了。

### ➤ 属性

每个实体都有一组特征或性质，称为实体的属性。实体的属性值是数据库中存储的主要数据，一个属性实际上相当于表中的一个列。下面来看看“文章”(article)实体。这个实体具有哪些属性呢？对于一篇文章来说，都具有文章标题、文章简介、添加时间、文章来源、文章内容、关键字、访问次数、推荐状态、审核状态。所以关于“文章”实体的属性如下：

- 文章标题 (title)
- 文章编号(id)
- 文章简介 (summary)
- 添加时间(posttime)
- 文章来源(comefrom)
- 文章内容(content)
- 关键字(keyword)
- 访问次数(views)
- 推荐状态(recommend)
- 审核状态(audit)

实体“栏目 (column)”包含的属性如下：

- 栏目标题(title)
- 栏目路径(path)
- 栏目描述(description)
- 排序编号(ord)

由于篇幅有限这里就不列出所有实体的属性了，在绘制 E-R 图中，属性由椭圆包围，在属性和它所属的实体间使用直线进行连接，以实体“文章”为例进行示例，如图 29-2 所示。

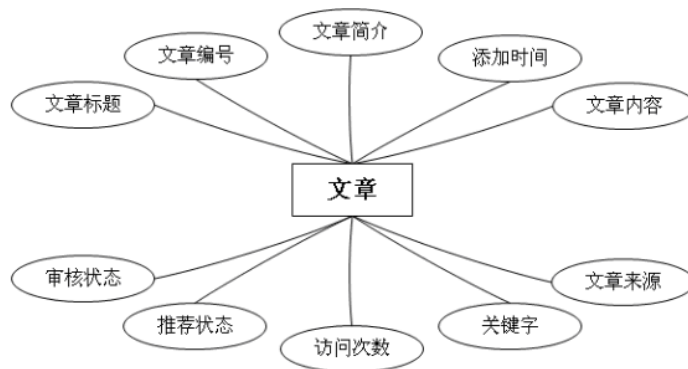


图 29-2 包含属性的 Department 的 E-R 图

对于每个实体，都有其确定的主属性(实体中的主属性实际上相当于表中的主键)，就可以惟一地确定实体的每个记录。最好是创建一个单独的属性作为主属性，在实体文章中可以选择“文章编号”作为主属性，在绘制 E-R 图中，主属性在属性下加下划线来说明。以实体“文章”为例进行示例，如图 29-3 所示。

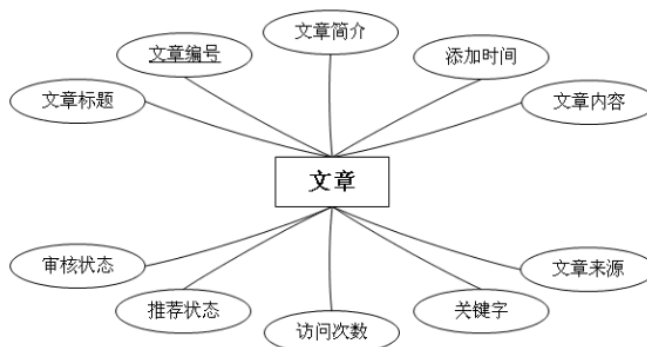


图 29-3 定义了主属性的“文章”的 E-R 图

**注意：**在数据库设计中，选择和设置列作为主键是一个关键步骤。

### 29.3.1.2 E-R 图的绘制

实体-关系图是表现实体-关系模型的图形工具，简称 E-R 图。这节会以 BroCMS 数据库为例，给出一个完整的数据库的 E-R 图设计示例。图 29-3 给出了在 E-R 图中使用的各种元素的图形符号。

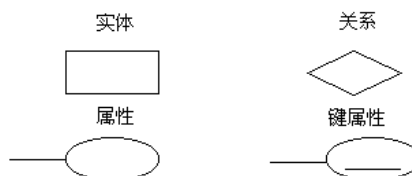


图 29-4 E-R 图中使用的各种元素的图形符号

在 E-R 图中，实体之间的关系以菱形表示，关系中各方面的表通过直线与菱形中的关系名称相连接。还要为每个关系命名一个“关系名称”，实体与关系相连的直线旁都根据关系的属性标注有“1”或“N”。

E-R 图为读者的数据库提供了一个不错的蓝图，可以分成三步进行：首先设计局部 E-R 图；然后合并各局部 E-R 图，并解决可能存在的冲突，得到初步 E-R 图；最后修改和重构初步 E-R 图，消除其中的冗余部分，得到最终的全局 E-R 图，即概念模式。设计全局 E-R 模式的目的在于把若干局部 E-R 模式形式上合并为一个 E-R 模式，而在于消除冲突使之成为能够被全系统中所有用户共同理解和接受的统一的概念模型。使设计人员仅从用户角度看待数据及处理要求和约束，产生一个反映用户观点的概念模式。

### 29.3.1.3 设计局部 E-R 模式

先设计局部 E-R 图，也称用户视图。在设计初步 E-R 图时，要尽量能充分地把组织中各部门对信息的要求集中起来，而不需要考虑数据的冗余问题。局部概念模型设计是从用户的观点出发，设计符合用户需求的概念结构。局部概念模型设计的就是组织、分类收集到的数据项，确定哪些数据项作为实体，哪些数据项作为属性，哪些数据项是同一实体的属性等。确定实体与属性的原则：

- 能作为属性的尽量作为属性而不要划为实体；
- 作为属性的数据元素与所描述的实体之间的联系只能是 1:n 的联系；
- 作为属性的数据项不能再用其他属性加以描述，也不能与其他实体或属性发生联系。

以下是 BroCMS 系统的部分局部 E-R 图的设计：

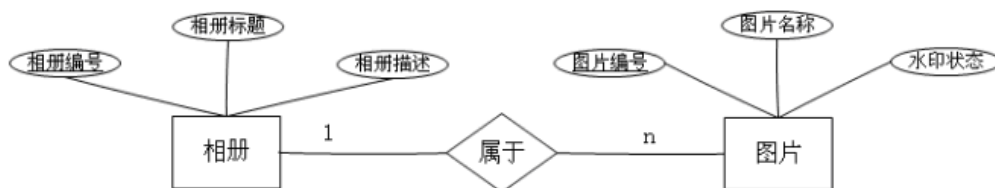


图 29-5 相册、图片的 E-R 图

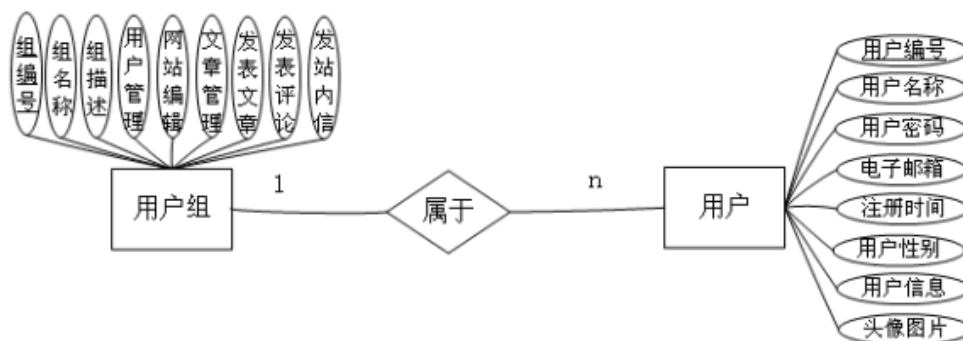


图 29-6 用户组、用户的 E-R 图



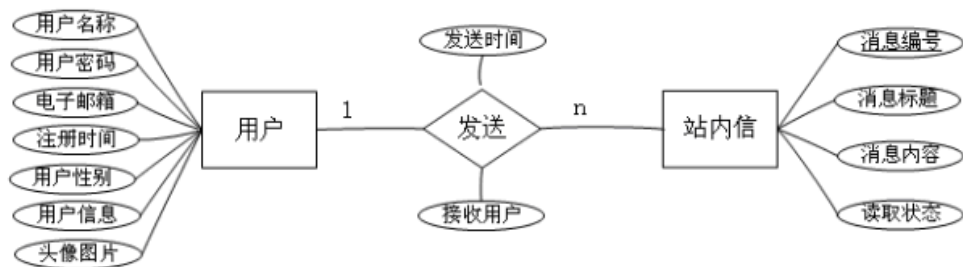


图 29-7 用户、站内信的 E-R 图

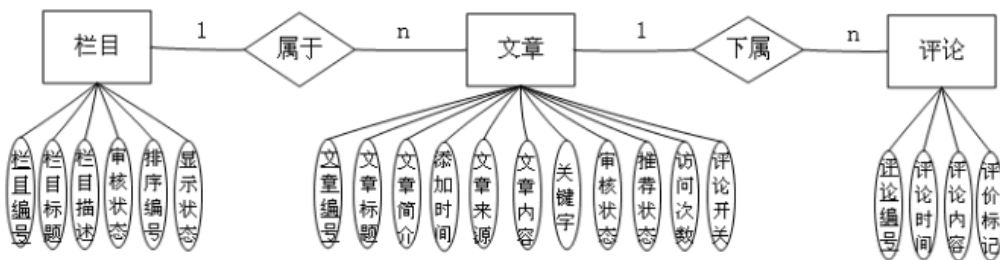


图 29-8 栏目、文章、评论的 E-R 图

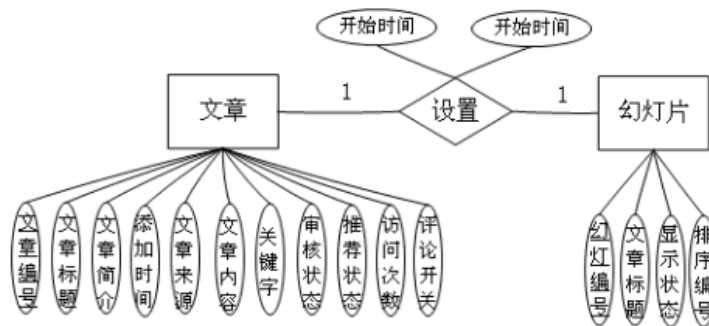


图 29-9 文章、幻灯片的 E-R 图

#### 29.3.1.4 设计全局 E-R 模式

综合各局部 E-R 图，形成总的 E-R 图，即用户视图的集成。所有局部 ER 模式都设计好了后，接下来就是把他们综合成单一的全局概念结构。全局概念结构不仅要支持所有局部 ER 模式，而且必须合理地表示一个完整、一致的数据库概念结构。

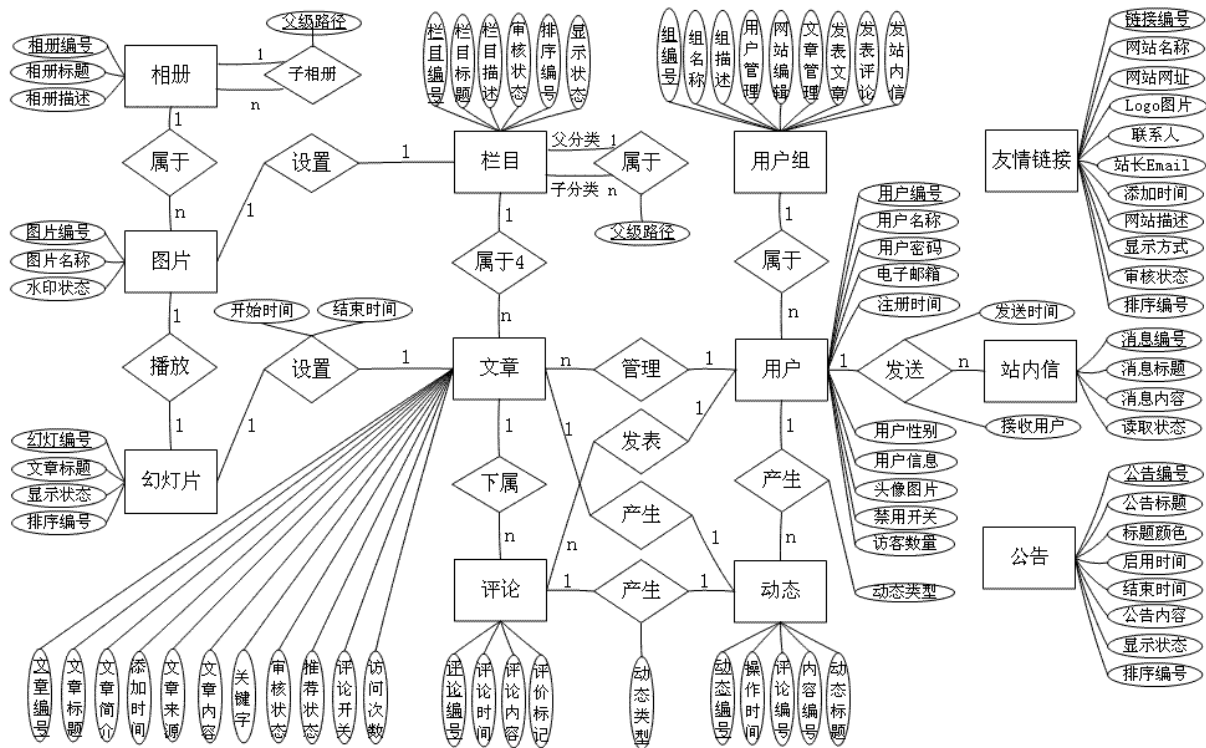


图 29-10 BroCMS 的全局 E-R 图

另外，在进入下一节之前，先回顾一下概念数据库的设计，其中主要是实体-关系模型的建立。简要总结一下实体-关系模型建立的步骤：

- (1) 对需求进行分析，从而确定系统中所包含的实体。
- (2) 分析得出每个实体所具有的属性。
- (3) 保证每个实体有一个主属性，该主属性可以是实体的一个属性或多个属性的组合。主属性必须能惟一地描述每个记录。
- (4) 确定实体之间的关系。

经过这些步骤后，读者就可以绘制出 E-R 图。之后可以再看看数据库的需要，判断是否获取了所需的信息，是否有遗漏信息等，读者可以再对 E-R 图进行修改，添加或删除实体与属性。

### 29.3.1.5 全局 ER 模式的优化

在得到全局 ER 模式后，为了提高数据库系统的效率，还应进一步依据处理需求对 ER 模式进行优化。一个好的全局 ER 模式，除能准确、全面地反映用户功能需求外，还应满足下列条件：

1. 实体类型的个数要尽可能的少
2. 实体类型所含属性个数尽可能少
3. 实体类型间联系无冗余

## 29.4 逻辑结构设计

逻辑结构设计的任务是把概念设计阶段建立的基本 E-R 图，按照选定的内容管理系统软件支持的数据模型，转化成相应的逻辑设计模型。也就是可以将实体、实体间的关系等模型结构转变为关系模式，即生成数据库中的表，并确定表的列。下述讨论由实体-关系模型生成表的方法。

### ➤ 任务：

将基本 E-R 图转换为与选用 DBMS 产品所支持的数据模型相符合的逻辑结构。

### ➤ 过程：

- 1) 将概念结构转换为现有 DBMS 支持的关系模型
- 2) 从功能和性能要求上对转换的模型进行评价，看它是否满足用户要求；
- 3) 对数据模型进行优化

### 29.4.1 ER 图向关系模型的转化

在上面实体之间的的基础上，将实体、实体的属性和实体之间的联系转换为关系模式。这种转换的原则是：

- 一个实体转换为一个关系，实体的属性就是关系的属性，实体的码就是关系的码。
- 一个联系也转化为一个关系，联系的属性及联系所连接的实体的码都转化为关系的属性，但是关系的码会根据关系的类型变化，如果是：
  - 1) 1: 1 联系，两端实体的码都成为关系的候选码
  - 2) 1: n 联系，n 端实体的码成为关系的码
  - 3) m: n 联系，两端的实体码的组成为关系的码

### 29.4.2 确定关系模式

根据转换算法，E-R 图中有 12 个实体类型，可以转换成 12 个关系模式：

- 1) 相册（相册编号，上级编号，父级路径，相册标题，相册描述）

- 2) 图片（图片编号，类别编号，图片名称，水印状态）
- 3) 分类栏目（栏目编号，上级编号，父级路径，栏目标题，栏目描述，图片编号，审核状态，排序编号，显示状态）
- 4) 文章（文章编号，文章标题，文章简介，添加时间，用户编号，文章来源，文章内容，关键字，类别编号，审核状态，推荐状态，评论开关，访问次数）
- 5) 幻灯片（幻灯编号，文章编号，文章标题，图片编号，开始时间，结束时间，显示状态，排序编号）
- 6) 评论（评论编号，用户编号，文章编号，评论时间，评论内容，评价标记）
- 7) 用户组（用户组编号，用户组名称，用户组描述，用户管理权限，文章管理权限，发表文章权限，发表评论权限，发站内信权限）
- 8) 用户（用户编号，组编号，用户名称，用户密码，电子邮箱，注册时间，用户性别，用户信息，头像图片，禁用开关，访客数量）
- 9) 站内信（消息编号，消息标题，用户编号，接收用户，发送时间，消息内容，读取状态）
- 10) 公告（公告编号，公告标题，标题颜色，启用时间，结束时间，公告内容，显示状态，排序编号）
- 11) 友情链接（链接编号，网站名称，网站网址，Logo 图片，联系人名字，站长 EMAIL，添加时间，网站描述，显示方式，审核状态，排序编号）
- 12) 动态信息（动态编号，用户编号，动态类型，操作时间，评论编号，内容编号，动态标题）

### 29.4.3 消除冗余

所谓冗余的数据是指可由基本数据导出的数据，冗余的联系是指可由其他联系导出的联系。冗余数据和冗余联系容易破坏数据库的完整性，给数据库的维护增加困难，应当予以消除。本系统的冗余数据和冗余关系已经在概念结构设计中处理过了，这里不再进行过多的叙述。

## 29.5 物理结构设计

数据库设计的最后阶段是确定数据库在物理设备上的存储结构和存取方法，也就是设计数据库的物理数据模型，主要是设计表结构。一般地，实体对应于表，实体的属性对应于表的列，实体之间的关系成为表的约束。逻辑设计中的实体大部分可以转换成物理设计中的表，但是它们并不一定是一一对应的。本次项目开发采用的是 MySQL 建立数据库。

### 29.5.1 设计数据表结构

在利用 MySQL 创建一个新的数据表以前，应当根据逻辑模型和数据字典先分析和设计数据表，描述出数据库中基本表的设计。需要确定数据表名称，所包含字段名称，数据类型，宽度以及建立的主键、外键等描述表的属性的内容。BroCMS 项目全部 12 个数据表结构设计如下所示：

表 29-1 相册表

表名	bro_album 用于保存相册记录，表引擎为 MyISAM 类型，字符集为 utf-8			
列名	数据类型	属性	约束条件	说明
id	SMALLINT (5)	无符号/非空/自动增涨	主键	相册编号
pid	SMALLINT (5)	无符号/非空/缺省 0	外键/普通索引 (album_pid)	上级编号
path	VARCHAR (100)	非空/缺省''	外键/普通索引 (album_path)	父级路径
title	VARCHAR (100)	非空/缺省''		相册标题
description	VARCHAR (200)	非空/缺省''		相册描述
补充说明	父级路径：保存了所有顶层父类的关系，使用“-”分隔的字符串			

表 29-2 图片表

表名	bro_image 用于保存图片记录，表引擎为 MyISAM 类型，字符集为 utf-8			
列名	数据类型	属性	约束条件	说明
id	INT (11)	无符号/非空/自动增涨	主键	图片编号
pid	SMALLINT (5)	无符号/非空/缺省 0	外键/普通索引 (image_pid)	类别编号
name	CHAR (24)	非空/缺省''		图片名称
water	TINYINT(1)	非空/缺省 0		水印状态
补充说明	水印状态：表示图片是否使用水印，0 表示没有水印，1 表示使用了水印			

表 29-3 分类栏目表

表名	bro_column 用于保存栏目分类记录，表引擎为 MyISAM 类型，字符集为 utf-8			
列名	数据类型	属性	约束条件	说明
id	SMALLINT (5)	无符号/非空/自动增涨	主键	栏目编号
pid	SMALLINT (5)	无符号/非空/缺省 0	外键/普通索引 (column_pid)	上级编号
path	VARCHAR (100)	非空/缺省''	外键/普通索引 (column_path)	父级路径
title	VARCHAR (100)	非空/缺省''		栏目标题
description	VARCHAR (200)	非空/缺省''		栏目描述
picid	SMALLINT (5)	无符号/非空/缺省 0	外键/普通索引(column_picid)	图片编号
audit	SMALLINT (1)	无符号/非空/缺省 1	普通索引 (column_audit)	审核状态
ord	SMALLINT (3)	无符号/非空/缺省 0	普通索引 (column_ord)	排序编号
display	SMALLINT (3)	无符号/非空/缺省 1	普通索引 (column_display)	显示状态
补充说明	父级路径：保存了所有顶层父类的关系，使用“-”分隔的字符串 审核状态：表示该栏目下的文章是否需要人工审核，1 为需要，0 为不需要 排序编号：设置顶层栏目在页面中的显示顺序，以数值的从小到大顺序排列 显示状态：用于设置栏目的显示或隐藏，值 1 为显示，值 0 为隐藏			

表 29-4 文章表

表名	bro_article 用于保存文章记录，表引擎为 MyISAM 类型，字符集为 utf-8			
列名	数据类型	属性	约束条件	说明
id	INT (11)	无符号/非空/自动增涨	主键	文章编号
title	VARCHAR (50)	非空/缺省''	普通索引 (article_title)	文章标题
summary	VARCHAR(200)	非空/缺省''		文章简介
posttime	INT(10)	无符号/非空/缺省 0		添加时间

uid	INT(11)	无符号/非空/缺省 0	外键/普通索引 (article_uid)	用户编号
comefrom	VARCHAR(50)	非空/缺省 ''		文章来源
content	TEXT	非空		文章内容
keyword	VARCHAR(20)	非空/缺省 ''	普通索引 (article_keyword)	关键字
pid	SMALLINT(5)	无符号/非空/缺省 0	外键/普通索引 (article_pid)	类别编号
audit	SMALLINT(1)	无符号/非空/缺省 0	普通索引 (article_audit)	审核状态
recommend	SMALLINT(1)	无符号/非空/缺省 0	普通索引 (article_recommend)	推荐状态
allow	SMALLINT(1)	无符号/非空/缺省 1	普通索引 (article_allow)	评论开关
views	SMALLINT(5)	无符号/非空/缺省 0		访问次数
补充说明	用户编号：使用这个字段关联用户 类别编号：使用这个字段关联文章所属的类别 审核状态：用于标记文章是否审核，值 1 为审核通过，值 0 表示还没审核 推荐状态：数值越大推荐的人就越多，用户每推荐一次数值增 1 评论开关：设置文章是否允许评论，值 1 为允许评论，值 0 则不允许评论			

表 29-5 幻灯片表

表名	bro_play 用于保存幻灯片记录，表引擎为 MyISAM 类型，字符集为 utf-8			
列名	数据类型	属性	约束条件	说明
id	SMALLINT (5)	无符号/非空/自动增涨	主键	幻灯编号
aid	INT(11)	无符号/非空/缺省 0	外键/普通索引 (play_aid)	文章编号
title	VARCHAR(80)	非空/缺省 ''		文章标题
picid	SMALLINT(5)	无符号/非空/缺省 0	外键/普通索引 (article_picid)	图片编号
starttime	INT(10)	无符号/非空/缺省 0	普通索引 (article_starttime)	开始时间
endtime	INT(10)	无符号/非空/缺省 0	普通索引 (article_endtime)	结束时间
display	SMALLINT(1)	无符号/非空/缺省 1	普通索引 (article_display)	显示状态
ord	SMALLINT(3)	无符号/非空/缺省 0	普通索引 (article_ord)	排序编号
补充说明	图片编号：用于设置幻灯片播放的图片			

表 29-6 评论表

表名	bro_comment 用于保存用户评论记录，表引擎为 MyISAM 类型，字符集为 utf-8			
列名	数据类型	属性	约束条件	说明
id	INT (11)	无符号/非空/自动增涨	主键	评论编号
uid	INT(11)	无符号/非空/缺省 0	外键/普通索引 (comment_picid)	用户编号
aid	INT(11)	无符号/非空/缺省 0	外键/普通索引 (comment_aid)	文章编号
ptime	INT(10)	无符号/非空/缺省 0		评论时间
content	TEXT	非空		评论内容
cmt	SMALLINT(5)	非空/缺省 0		评价标记
补充说明	用户编号：用于标记评论所属用户 文章编号：用于标记评论所属文章 评价标记：表示用户评价级别，包括：0 为中立、1 为好评、-1 为差评			

表 29-7 用户组表

表名	bro_group 用于保存用户组记录，表引擎为 MyISAM 类型，字符集为 utf-8			
列名	数据类型	属性	约束条件	说明
id	SMALLINT (4)	无符号/非空/自动增涨	主键	用户组编号
groupname	VARCHAR(20)	非空/缺省''		用户组名称
description	VARCHAR(200)	非空/缺省''		用户组描述
useradmin	TINYINT(1)	非空/缺省 0		用户管理权限
webadmin	TINYINT(1)	非空/缺省 0		网站编辑权限
articleadmin	TINYINT(1)	非空/缺省 0		文章管理权限
sendarticle	TINYINT(1)	非空/缺省 0		发表文章权限
sendcomment	TINYINT(1)	非空/缺省 0		发表评论权限
sendmessage	TINYINT(1)	非空/缺省 0		发站内信权限
补充说明	在用户组中可以设置 6 个权限，设置方式相同，值 1 为拥有权限，值 0 为没有权限。所属该用户组中的所有用户拥有该组设置的权限。			

表 29-8 用户表

表名	bro_user 用于保存用户记录，表引擎为 MyISAM 类型，字符集为 utf-8			
列名	数据类型	属性	约束条件	说明
id	INT (11)	无符号/非空/自动增涨	主键	用户编号
gid	SMALLINT(4)	无符号/非空/缺省 0	外键/普通索引 (user_gid)	组编号
username	VARCHAR(20)	非空/缺省''	普通索引 (user_username)	用户名称
userpwd	VARCHAR(40)	非空/缺省''	普通索引 (user_userpwd)	用户密码
email	VARCHAR(60)	非空/缺省''		电子邮箱
regtime	INT(10)	无符号/非空/缺省 0		注册时间
sex	SMALLINT(3)	非空/缺省 0		用户性别
info	VARCHAR(120)	非空/缺省''		用户信息
upic	CHAR(24)	非空/缺省''		头像图片
disable	SMALLINT(3)	无符号/非空/缺省 0	普通索引 (user_disable)	禁用开关
views	SMALLINT(5)	无符号/非空/缺省 0		访客数量
补充说明	用户密码：使用 md5 加密 用户性别：有三个值，1 为男，2 为女，0 为保密。可以用来设置默认的用户头像 禁用开关：0 为开启，1 为禁用用户			

表 29-9 站内信表

表名	bro_message 用于保存发表的站内信记录，表引擎为 MyISAM 类型，字符集为 utf-8			
列名	数据类型	属性	约束条件	说明
id	INT (11)	无符号/非空/自动增涨	主键	消息编号
title	VARCHAR(80)	非空/缺省''		消息标题
uid	INT(11)	无符号/非空/缺省 0	外键/普通索引 (message_uid)	用户编号
revicename	VARCHAR(30)	非空/缺省''	普通索引 (message_revicename)	接收用户
ptime	INT(10)	无符号/非空/缺省 0		发送时间
content	TEXT	缺省''		消息内容
stutas	SMALLINT(1)	非空/缺省 0	普通索引 (message_stauts)	读取状态

<b>补充说明</b>	用户编号：指发送站内信的用户编号（关联发送者） 接收用户：指接收站内信的用户名称（关联接收者） 读取状态：有两个值 0 表示未读消息，1 表示已读
-------------	---

表 29-10 公告表

<b>表名</b>	bro_notice 用于保存公告记录，表引擎为 MyISAM 类型，字符集为 utf-8			
<b>列名</b>	<b>数据类型</b>	<b>属性</b>	<b>约束条件</b>	<b>说明</b>
id	SMALLINT (5)	无符号/非空/自动增涨	主键	公告编号
title	VARCHAR(80)	非空/缺省''		公告标题
color	CHAR(6)	非空/缺省'000000'		标题颜色
starttime	INT(10)	无符号/非空/缺省 0	普通索引 (notice_starttime)	启用时间
endtime	INT(10)	无符号/非空/缺省 0	普通索引 (notice_endtime)	结束时间
content	TEXT	缺省''		公告内容
display	SMALLINT(1)	无符号/非空/缺省 1	普通索引 (notice_display)	显示状态
ord	SMALLINT(3)	无符号/非空/缺省 0	普通索引 (notice_ord)	排序编号
<b>补充说明</b>	公告颜色：用于设置公告标题的高亮显示颜色，使用 16 进制 RGB 值 显示状态：有两个可用值，1 为显示，0 为不显示			

表 29-11 友情链接表

<b>表名</b>	bro_flink 用于保存友情链接记录，表引擎为 MyISAM 类型，字符集为 utf-8			
<b>列名</b>	<b>数据类型</b>	<b>属性</b>	<b>约束条件</b>	<b>说明</b>
id	SMALLINT (5)	无符号/非空/自动增涨	主键	链接编号
webname	VARCHAR(30)	非空/缺省''		网站名称
url	VARCHAR(60)	非空/缺省''		网站网址
logo	VARCHAR(60)	非空/缺省''		Logo 图片
rname	VARCHAR(30)	非空/缺省''		联系人名字
email	VARCHAR(50)	非空/缺省''		站长 EMAIL
dtime	INT(10)	无符号/非空/缺省 0		添加时间
msg	VARCHAR(200)	非空/缺省''		网站描述
list	SMALLINT(1)	无符号/非空/缺省 0	普通索引 (flink_list)	显示方式
audit	SMALLINT(1)	无符号/非空/缺省 0	普通索引 (flink_audit)	审核状态
ord	SMALLINT(3)	无符号/非空/缺省 0	普通索引 (flink_list)	排序编号
<b>补充说明</b>	Logo 图片：为需要链接的网站 Logo 图片地址 显示方式：有两种显示方式，0 为显示网站名称，1 为显示网站的 Logo 图片 审核状态：有两个状态，0 值为没开通，值 1 为可以显示			

表 29-12 动态信息表

<b>表名</b>	bro_dynamic 用于保存用户动态信息记录，表引擎为 MyISAM 类型，字符集为 utf-8			
<b>列名</b>	<b>数据类型</b>	<b>属性</b>	<b>约束条件</b>	<b>说明</b>
id	INT (11)	无符号/非空/自动增涨	主键	动态编号
uid	INT(11)	无符号/非空/缺省 0	外键/普通索引 (dynamic_uid)	用户编号



otype	SMALLINT(1)	无符号/非空/缺省 0	普通索引 (dynamic_otype)	动态类型
ptime	INT(11)	无符号/非空/缺省 0		操作时间
pid	INT(11)	无符号/非空/缺省 0	外键/普通索引 (dynamic_pid)	评论编号
cid	INT(11)	无符号/非空/缺省 0	外键/普通索引 (dynamic_cid)	内容编号
title	VARCHAR(100)	非空/缺省''		动态标题
补充说明	用户编号：标识具体用户的动态信息 动态类型：共 5 种类型，以 1-5 的整数表示。包括：1 表示用户发表文章、2 表示用户发表评论、3 表示用户的收藏、4 表示用户的推荐、5 表示添加关注好友。 评论编号：如果动态类型值为 2 时，用来保存评论的编号 内容编号：内容编号会根据动态类型的值进行设置，例如：类型为 1、2、3、4 时内容编号为发表文章的编号，类型值为 5 时内容编号则为关注的用户编号。 动态标题：动态标题也是根据动态类型的值进行设置，例如：类型为 1、2、3、4 时动态标题为发表文章的标题，类型值为 5 时动态标题则为关注的用户名称。			

**注意：**上述数据字典为在 MySQL 中呈现的方式，数据类型在其它数据库产品中部分需要改动。

## 29.5.2 创建数据表

通过数据表结构的详细设计，再结合 MySQL 的创建数据表的语法，BroCMS 项目的 12 个数据表的完整建表 SQL 语句如下所示，默认表前缀为 “bro\_”：

创建相册表 **bro\_album** 的语句如下所示：

```
DROP TABLE IF EXISTS bro_album;
CREATE TABLE bro_album (
  id SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
  pid SMALLINT(5) UNSIGNED NOT NULL DEFAULT 0,
  path VARCHAR(100) NOT NULL DEFAULT "",
  title VARCHAR(100) NOT NULL DEFAULT "",
  description VARCHAR(200) NOT NULL DEFAULT "",
  INDEX album_pid(pid),
  INDEX album_path(path),
  PRIMARY KEY (id)
) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

创建图片表 **bro\_image** 的语句如下所示：

```
DROP TABLE IF EXISTS bro_image;
CREATE TABLE bro_image (
  id INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  pid SMALLINT(4) UNSIGNED NOT NULL DEFAULT 0,
  name CHAR(24) NOT NULL DEFAULT "",
  water TINYINT(1) NOT NULL DEFAULT 0,
```

```
PRIMARY KEY (id),
KEY image_pid (pid)
) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

创建栏目表 **bro\_column** 的语句如下所示:

```
DROP TABLE IF EXISTS bro_column;
CREATE TABLE bro_column (
    id SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
    pid SMALLINT(5) UNSIGNED NOT NULL DEFAULT 0,
    path VARCHAR(100) NOT NULL DEFAULT "",
    title VARCHAR(100) NOT NULL DEFAULT "",
    description VARCHAR(200) NOT NULL default "",
    picid SMALLINT(5) UNSIGNED NOT NULL DEFAULT 0,
    audit SMALLINT(1) UNSIGNED NOT NULL DEFAULT '1',
    ord SMALLINT(3) UNSIGNED NOT NULL DEFAULT 0,
    display SMALLINT(3) UNSIGNED NOT NULL DEFAULT 1,
    PRIMARY KEY (id),
    INDEX column_pid(pid),
    INDEX column_path(path),
    INDEX column_audit(audit),
    INDEX column_ord(ord),
    INDEX column_display(display),
    INDEX column_picid(picid)
) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

创建文章表 **bro\_article** 的语句如下所示:

```
DROP TABLE IF EXISTS bro_article;
CREATE TABLE bro_article (
    id INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
    title VARCHAR(50) NOT NULL DEFAULT "",
    summary VARCHAR(200) NOT NULL DEFAULT "",
    posttime INT(10) UNSIGNED NOT NULL DEFAULT '0',
    uid INT(11) UNSIGNED NOT NULL DEFAULT 0,
    comefrom VARCHAR(50) NOT NULL DEFAULT "",
    content TEXT NOT NULL,
    keyword VARCHAR(20) NOT NULL DEFAULT "",
    pid SMALLINT(5) UNSIGNED NOT NULL DEFAULT 0,
    audit SMALLINT(1) UNSIGNED NOT NULL DEFAULT '0',
    recommend SMALLINT(1) UNSIGNED NOT NULL DEFAULT '0',
    allow SMALLINT(1) UNSIGNED NOT NULL DEFAULT '1',
    views SMALLINT(5) UNSIGNED NOT NULL DEFAULT 0,
    PRIMARY KEY (id),
    INDEX article_title(title),
    INDEX article_uid(uid),
```

```

        INDEX article_pid(pid),
        INDEX article_audit(audit),
        INDEX article_recommend(recommend),
        INDEX article_allow(allow),
        INDEX article_keyword(keyword)
    ) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

```

创建幻灯片表 **bro\_play** 的语句如下所示:

```

DROP TABLE IF EXISTS bro_play;
CREATE TABLE bro_play (
    id SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
    aid INT(11) UNSIGNED NOT NULL DEFAULT 0,
    title VARCHAR(80) NOT NULL DEFAULT "",
    picid SMALLINT(5) UNSIGNED NOT NULL DEFAULT 0,
    starttime INT(10) UNSIGNED NOT NULL DEFAULT '0',
    endtime INT(10) UNSIGNED NOT NULL DEFAULT '0',
    display SMALLINT(1) UNSIGNED NOT NULL DEFAULT '1',
    ord SMALLINT(3) UNSIGNED NOT NULL DEFAULT 0,
    PRIMARY KEY (id),
    INDEX play_aid(aid),
    INDEX play_picid(picid),
    INDEX play_starttime(starttime),
    INDEX play_endtime(endtime),
    INDEX play_ord(ord),
    INDEX play_display(display)
) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

```

创建评论表 **bro\_comment** 的语句如下所示:

```

DROP TABLE IF EXISTS bro_comment;
CREATE TABLE bro_comment (
    id INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
    uid INT(11) UNSIGNED NOT NULL DEFAULT '0',
    aid INT(11) UNSIGNED NOT NULL DEFAULT '0',
    ptime INT(10) UNSIGNED NOT NULL DEFAULT '0',
    content TEXT NOT NULL,
    cmt SMALLINT(5) NOT NULL DEFAULT 0,
    PRIMARY KEY (id),
    INDEX comment_uid(uid),
    INDEX comment_aid(aid)
) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

```

创建用户组表 **bro\_group** 的语句如下所示:

```

DROP TABLE IF EXISTS bro_group;
CREATE TABLE bro_group (

```

```

    id SMALLINT(4) UNSIGNED NOT NULL AUTO_INCREMENT,
    groupname VARCHAR(20) NOT NULL DEFAULT "",
    description VARCHAR(200) NOT NULL DEFAULT "",
    useradmin TINYINT(1) NOT NULL DEFAULT 0,
    webadmin TINYINT(1) NOT NULL DEFAULT 0,
    articleadmin TINYINT(1) NOT NULL DEFAULT 0,
    sendarticle TINYINT(1) NOT NULL DEFAULT 0,
    sendcomment TINYINT(1) NOT NULL DEFAULT 0,
    sendmessage TINYINT(1) NOT NULL DEFAULT 0,
    PRIMARY KEY (id)
) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

```

创建用户表 **bro\_user** 的语句如下所示:

```

DROP TABLE IF EXISTS bro_user;
CREATE TABLE bro_user (
    id INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
    gid SMALLINT(4) UNSIGNED NOT NULL,
    username VARCHAR(20) NOT NULL DEFAULT "",
    userpwd VARCHAR(40) NOT NULL DEFAULT "",
    email VARCHAR(60) NOT NULL DEFAULT "",
    regtime INT(10) UNSIGNED NOT NULL DEFAULT '0',
    sex SMALLINT(3) NOT NULL DEFAULT 0,
    info VARCHAR(120) NOT NULL DEFAULT "",
    upic CHAR(24) NOT NULL DEFAULT "",
    disable SMALLINT(3) UNSIGNED NOT NULL DEFAULT 0,
    views SMALLINT(5) UNSIGNED NOT NULL DEFAULT 0,
    PRIMARY KEY (id),
    INDEX user_gid(gid),
    INDEX user_username(username),
    INDEX user_userpwd(userpwd),
    INDEX user_disable(disable)
) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

```

创建站内消息表 **bro\_message** 的语句如下所示:

```

DROP TABLE IF EXISTS bro_message;
CREATE TABLE bro_message (
    id INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
    title VARCHAR(80) NOT NULL DEFAULT "",
    uid INT(11) UNSIGNED NOT NULL DEFAULT 0,
    revicename VARCHAR(30) NOT NULL DEFAULT "",
    ptime INT(10) UNSIGNED NOT NULL DEFAULT '0',
    content TEXT NOT NULL,
    stutas SMALLINT NOT NULL DEFAULT 0,
    PRIMARY KEY(id),

```

```

        INDEX message_uid(uid),
        INDEX message_revicename(revicename),
        INDEX message_stutas(stutas)
) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

```

创建公告表 **bro\_notice** 的语句如下所示:

```

DROP TABLE IF EXISTS bro_notice;
CREATE TABLE bro_notice (
    id SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
    title VARCHAR(80) NOT NULL DEFAULT "",
    color CHAR(6) NOT NULL DEFAULT '000000',
    starttime INT(10) UNSIGNED NOT NULL DEFAULT '0',
    endtime INT(10) UNSIGNED NOT NULL DEFAULT '0',
    content TEXT NOT NULL,
    display SMALLINT(1) UNSIGNED NOT NULL DEFAULT '1',
    ord SMALLINT(3) UNSIGNED NOT NULL DEFAULT 0,
    PRIMARY KEY (id),
    INDEX notice_starttime(starttime),
    INDEX notice_endtime(endtime),
    INDEX notice_display(display),
    INDEX notice_ord(ord)
) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

```

创建友情链接表 **bro\_flink** 的语句如下所示:

```

DROP TABLE IF EXISTS bro_flink;
CREATE TABLE bro_flink (
    id SMALLINT(5) UNSIGNED NOT NULL AUTO_INCREMENT,
    webname VARCHAR(30) NOT NULL DEFAULT "",
    url VARCHAR(60) NOT NULL DEFAULT "",
    logo VARCHAR(60) NOT NULL DEFAULT "",
    rname VARCHAR(30) NOT NULL DEFAULT "",
    email VARCHAR(50) NOT NULL DEFAULT "",
    dtime INT(10) UNSIGNED NOT NULL DEFAULT '0',
    msg VARCHAR(200) NOT NULL DEFAULT "",
    list SMALLINT(1) UNSIGNED NOT NULL DEFAULT '0',
    audit SMALLINT(1) UNSIGNED NOT NULL DEFAULT 0,
    ord SMALLINT(3) UNSIGNED NOT NULL DEFAULT 0,
    PRIMARY KEY (id),
    INDEX flink_list(list),
    INDEX flink_audit(audit),
    INDEX flink_ord(ord)
) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

```

创建用户动态表 **bro\_dynamic** 的语句如下所示:

```

DROP TABLE IF EXISTS bro_dynamic;
CREATE TABLE bro_dynamic (
    id INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
    uid INT(11) UNSIGNED NOT NULL DEFAULT 0,
    otype SMALLINT(1) UNSIGNED NOT NULL DEFAULT 0,
    ptime INT(11) UNSIGNED NOT NULL DEFAULT 0,
    pid INT(11) UNSIGNED NOT NULL DEFAULT 0,
    cid INT(11) UNSIGNED NOT NULL DEFAULT 0,
    title VARCHAR(100) NOT NULL DEFAULT "",
    PRIMARY KEY(id),
    INDEX dynamic_uid(uid),
    INDEX dynamic_otype(otype),
    INDEX dynamic_pid(pid),
    INDEX dynamic_cid(cid)
) DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;

```

### 29.5.3 数据表记录的输入

在创建数据表的时候可以根据系统提示直接输入记录，但是也可以暂时不输入记录。没有记录的数据表叫做空表，可以随时向数据表中追加记录，也可以向已经存在的记录的数据表追加记录。但根据 BroCMS 的需求，必须为用户组插入两条记录，作为默认的两个用户组。默认两条插入的 SQL 语句及插入的内容数据如下所示：

```

INSERT INTO bro_group(id,groupname,description, useradmin, webadmin, articleadmin, sendarticle,
sendcomment, sendmessage) VALUES('1','超级管理员','超级管理员用户组，该组的用户具有全部的权限
','1','1','1','1','1','1');

INSERT INTO bro_group(groupname, description, useradmin, webadmin, articleadmin, sendarticle,
sendcomment, sendmessage) VALUES('普通会员','新注册用户默认属于该组，只具有发评论和短消息的
权限','0','0','0','0','1','1');

```

## 29.6 安全保密设计

本数据库系统采用安全的用户名加口令方式登录。用户名的权限限制为只能进行基本的增删改查数据功能。

## 29.6.1 完整性

数据库应用对数据一般都具有一定的限制，这种限制称为完整性。数据库的完整性是保证数据库正确的关键。广义上说，数据库完整性包括数据库中数据的准确性和一致性。理想情况下，数据库软件提供了检查数据完整性的各种方法，但遗憾的是，目前大多数关系数据库系统对数据库的完整性支持并不够充分。

关系数据库系统应该保证输入的值符合其规定的数据类型，并保证值在系统支持的范围内。不同的关系数据库系统可能提供了不同的数据类型，但所有的关系数据库系统都检查输入的值，并且拒绝不符合定义的数据类型的值或者不在系统支持范围内的值。例如，不希望将一个工作人员的生日输入为“1981.11.5”而将另一个工作人员的生日输入为“11/5/1981”。数据必须准确且在数据库中一致存储，这样才能从数据库中检索出正确的数据，才能在数据间作出正确比较。一般来说，关系数据库系统都支持 3 种完整性：

- 域约束
- 实体完整性约束
- 关联完整性约束

## 29.6.2 数据库设计的其他问题

上面介绍了数据库设计的主要部分，这些可以帮助用户设计出基本符合要求的数据库，但是在数据库的设计中还有许多其他的问题需要考虑。

- 索引：索引是数据库实现的重要组成部分，对于数据的检索、存储等有很大的影响。关于索引的创建将在第 3 章创建数据库中介绍。
- 安全性：在数据库中安全性是一个非常重要的问题。必须保证数据库数据的安全性，特别是在 Internet 迅速发展的今天，有效地保障数据库数据的安全对用户来说是非常重要的。
- 数据字典：数据字典用来存储数据库中存储的数据的描述信息和数据库管理系统需要的控制信息，数据字典与数据库管理软件密切相关。
- 物理数据库设计：每个数据库管理系统都提供很多种存储结构和存储方法供数据库设计人员选择，物理数据库设计的好坏对数据库的性能有很大的影响。