**Iaecpedia**

**数据库设计说明书**

编写人员：何徐昊

目录

[1.概述 2](#_Toc471219040)

[2.数据库概念设计 3](#_Toc471219041)

[2.1 概述 3](#_Toc471219042)

[2.2 实体和属性的定义 3](#_Toc471219043)

[2.3 ER图 3](#_Toc471219044)

[3.数据库逻辑结构设计 4](#_Toc471219045)

[3.1概述 4](#_Toc471219046)

[3.2关系模式 5](#_Toc471219047)

[3.3 数据表图 5](#_Toc471219048)

[3.4 数据表详细内容 6](#_Toc471219049)

[4.数据库物理结构设计 9](#_Toc471219050)

[4.1概述 9](#_Toc471219051)

[4.2 数据库服务器 9](#_Toc471219052)

[4.3 网站服务器 9](#_Toc471219053)

1.概述

本数据库设计说明书是对laecpedia的数据库设计的定义，包含本系统数据库概念设计，数据库逻辑结构设计，数据库物理结构设计，数据字典。

本数据库设计说明书适合以下读者：

系统设计人员

系统开发人员

系统测试人员

系统维护人员

本数据库说明书用于以下开发活动的依据之一：

详细设计

用户验收

2.数据库概念设计

2.1 概述

经过需求分析后，我们将需求分析得到的用户需求抽象为信息结构(即概念模型)的过程就是概念结构设计。概念结构是各种数据 模型的共同基础，它比数据模型更独立于机器，更抽象，从而更加稳定，它是整个数据库设计的关键。它通过ER图的形式来展现。

2.2 实体和属性的定义

用户基本信息（账号、密码、用户名、角色、等级、经验值）

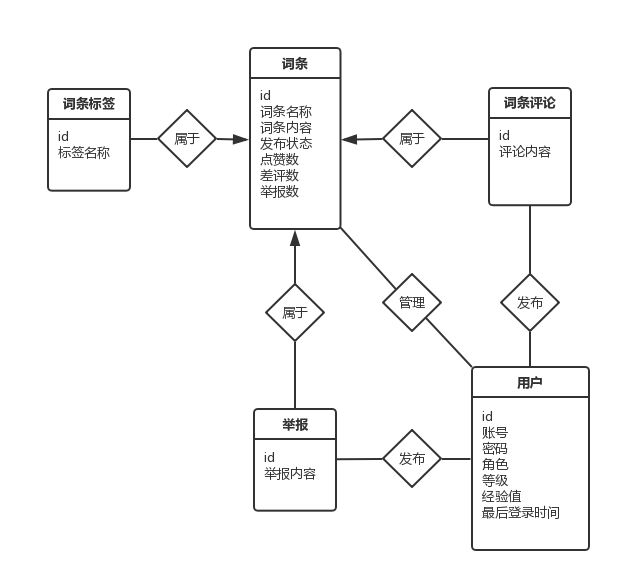
词条基本信息（名称、内容、状态、标签（多值）、点赞数、差评数、举报数）

评论基本信息（评论内容）

举报基本信息（举报理由）

2.3 ER图

如下：



3.数据库逻辑结构设计

3.1概述

逻辑结构设计需要先将概念结构转化为一般的关系， 网状，层次模型，然后将这些模型向特定DBMS支持下的数据模型转换，最后对数据模型进行优化。将概念模型转换为逻辑模型通常是将E-R模型转换为关系模型。转换的原则是：1.一个实体转换为一个关系模式；2.一个m:n联系转换为一个关系模式；3.一个1:n联系可以转换为一个独立的关系模式，也可以与n对应的关系模式合并；4.一个1:1联系可以转换为一个独立的关系模式，也可以与 任意一端的关系模式合并；5.三个或三个以上实体间的一个多元联系转换为一个关系模式；6.同一实体集的实体间的联系，即自联系，也可按上述1:1,1:n和m:n三种情况分别处理。在数据库中有以下实体：用户、词条、举报、评论、标签。再结合各个实体之间的关系，可得出以下关系模式（下划线表示主码）:

3.2关系模式

User(uid,account,password,username,role,level,exp,lastLoginTime);

Entry(eid,uid,entryName,entryContent,status,praiseTimes,badReviewTimes

, reportTimes,pictureAddr,publisherTime);

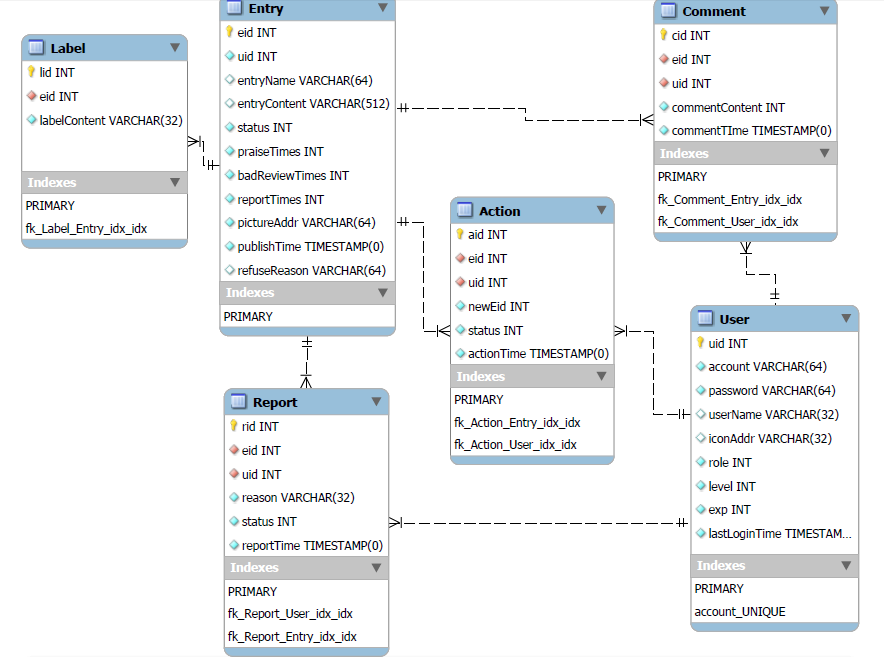
Report(rid,eid,uid.reason,status);

Comment(cid,eid,uid,commentContent);

Label(lid,eid,labelContent);

Action(aid,eid,uid,status,neweid,actionTime);

3.3 数据表图



3.4 数据表详细内容

User表 存放用户的基本信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键 | 非空 | 描述 |
| uid | Int | 11 | 是 | 是 | 自增id |
| account | Varchar | 64 |  | 是 | 用户账号 |
| password | Varchar | 64 |  | 是 | 用户密码 |
| userName | Varchar | 32 |  |  | 用户名(默认为account) |
| iconAddr | Varchar | 32 |  |  | 头像地址 |
| role | Int | 11 |  | 是 | 角色（1=普通用户 2=管理员） |
| Level | Int | 11 |  | 是 | 等级（默认为1） |
| Exp | Int | 11 |  | 是 | 经验值（默认为0） |
| lastLoginTime | Timestamp | Yyyy-mm--dd |  | 是 | 最后登录时间 |

Entry表 存放词条的基本信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键 | 非空 | 描述 |
| eid | Int | 11 | 是 | 是 | 词条自增id |
| uid | Int | 11 |  | 是 | 发布词条者的id |
| entryName | Varchar | 64 |  | 是 | 词条名称 |
| entryContent | Varchar | 512 |  | 是 | 词条具体内容 |
| Status | Int | 11 |  | 是 | 词条状态（  1待审核  2已发布  3被删除  4被替换  5审核不通过） |
| PraiseTimes | Int | 11 |  | 是 | 点赞数（默认为0） |
| badReviewTImes | Int | 11 |  | 是 | 差评数（默认为0） |
| reportTimes | Int | 11 |  | 是 | 举报数（默认为0） |
| publishedTime | Timestamp | Yyyy-mm--dd |  | 是 | 词条发布时间 |
| refuseReason | Varchar | 64 |  | 是 | 词条不通过理由 |

Label表 存放词条的标签

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键 | 非空 | 描述 |
| lid | Int | 11 | 是 | 是 | 标签自增id |
| eid | Int | 11 |  | 是 | 词条的id |
| labelContent | Varchar | 16 |  | 是 | 标签内容 |

Report表 存放举报信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键 | 非空 | 描述 |
| rid | Int | 11 | 是 | 是 | 举报的自增id |
| eid | Int | 11 |  | 是 | 被举报的词条的id |
| uid | Int | 11 |  | 是 | 举报者的id |
| reason | Varchar | 32 |  | 是 | 举报理由 |
| Status | Varchar | 32 |  | 是 | 举报处理的状态（  1 待审核  2 通过  3不通过） |
| reportTime | Timestamp | Yyyy-mm--dd |  | 是 | 举报时间 |

Comment表 存放对词条的评论

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键 | 非空 | 描述 |
| cid | Int | 11 | 是 | 是 | 评论的自增id |
| eid | Int | 11 |  | 是 | 被评论词条的id |
| Uid | Int | 11 |  | 是 | 评论用户的id |
| commentContent | Varchar | 64 |  |  | 评论内容 |
| commentTIme | Timestamp | Yyyy-mm-dd |  | 是 | 评论提交时间 |

Action表 存放词条修改的信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键 | 非空 | 描述 |
| Aid | Int | 11 | 是 | 是 | 修改的自增id |
| eid | Int | 11 |  | 是 | 原词条的id |
| neweid | Int | 11 |  | 是 | 新词条的id |
| Uid | Int | 11 |  |  | 修改者的id |
| Status | Int | 11 |  |  | 修改审核的状态  1未审核  2已通过  3不通过 |
| actionTime | Timestamp | Yyyy-mm-dd |  | 是 | 修改提交时间 |

4.数据库物理结构设计

4.1概述

数据库在物理设备上的存储结构和存取方法称为数据库的物理结构，它依赖于给定的计算机系统，为一个给定的逻辑数据模型选取一个最适合应用环境的物理结构的过程就是数据库的物理设计。物理设计要做的就是为关系模式选择存取方法（建立存取路径），设计关系、索引等数据库文件的物理存储结构。

4.2 数据库服务器

* 服务器: Localhost via UNIX socket
* 服务器类型: MySQL
* 服务器版本: 5.5.49-0ubuntu0.14.04.1 - (Ubuntu)
* 协议版本: 10
* 用户: root@localhost
* 服务器字符集: UTF-8 Unicode (utf8)

4.3 网站服务器

* Apache/2.4.7 (Ubuntu)
* 数据库客户端版本: libmysql - 5.5.49

**5.系统数据字典**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据名称 | 数据业务描述 | 业务数据类型 | 所属表格 | 对应的字段 | 字段类型 |
| 账号 | 用户账号 | 可变字符串(10~40) | User | account | VarChar(64) |
| 密码 | 用户密码 | 可变字符串(10~40) | User | password | Varchar(64) |
| 用户名 | 用户名 | 可变字符串(4~25) | User | userName | Varchar(32) |
| 头像 | 用户头像(数据库只存放地址，图片另存在硬盘) | 可变字符串(30~70) | User | iconAddr | Varchar(128) |
| 角色 | 用户身份(  1为普通用户  2为管理员) | 整数 | User | role | Int |
| 等级 | 用户等级 | 整数 | User | revel | Int |
| 经验值 | 用户当前经验值 | 整数 | User | exp | Int |
| 最后登录时间 | 用户最后一次登录的时间 | 时间(yyyy-mm-dd ) | User | lastLoginTime | Timestamp |
| 词条名字 | 词条的名称 | 可变字符串（1~30） | Entry | entryName | Varchar(32) |
| 词条标签 | 用户自定义词条的标签 | 可变字符串（1~10） | Label | labelContent | Varchar(16) |
| 词条内容 | 对词条详细的描述 | 可变字符串（1~500） | Entry | entryContent | Varchar（512） |
| 词条图片地址 | 词条图片的地址 | 可变字符串(30~70) | Entry | prictureAddr | VarChar(128) |
| 词条状态 | 词条的状态  1待审核  2已发布  3被删除  4被替换  5审核不通过 | 整数 | Entry | Status | Int |
| 发布时间 | 发布时间 | 时间(yyyy-mm-dd) | Entry | publishTime | Timestamp |
| 点赞数 | 词条受到点赞的次数 | 整数 | Entry | praiseTimes | Int |
| 差评数 | 词条受到差评的次数 | 整数 | Entry | badreviewTimes | Int |
| 举报次数 | 词条被举报的次数 | 整数 | Entry | reportTimes | Int |
| 词条评论 | 用户对词条的评论 | 可变字符串（1~50） | Comments | commentContent | Varchar(64) |