

# 数据库设计说明书

# 第一章 引言

## 1.1 编写目的

本数据库设计说明书是对失物招领系统数据库设计的定义，包括本系统数据 逻辑结构设计、数据字典以及运行环境、安全保密设计等。 本数据库设计说明书适合以下读者：

1. 用户
2. 系统设计人员
3. 质量控制人员
4. 系统确认测试人员
5. 系统维护人员

本数据库设计说明书是以下开发活动的依据之一：

6. 系统详细设计
7. 用户验收

# 第二章 外部设计

## 2.1 标识符和状态

数据库软件的名称：xampp  
管理数据库软件的名称：navicat  
数据库的名称为：lostandfound

## 2.2 使用它的程序

本数据库使用于 “失物招领系统”

## 2.3 命名约定

所有的数据库命名都是以模块的缩写加上具体表的英文词汇组成，这样能够统一数据库表的命名，也能够更好的规范数据库表命名。

## 2.4 设计约定

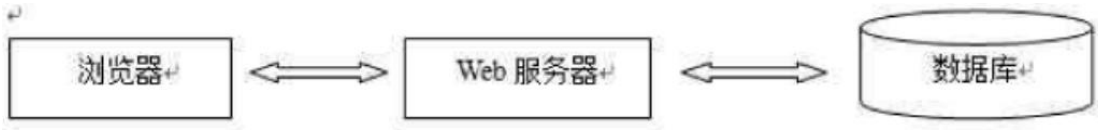
在本系统中，数据库的设计采用 navicat 进行，并且采用面向对象的设计方法，首先进行对象实体的设计，最后将对象持久化到数据库中，所有的表和表之间的关联(ER 图)都采用标准的 navicat 设计工具进行，这样能够将整个系统的设计和 数据库设计有机的结合起来。

# 第三章 结构设计

## 3.1 数据库的配置

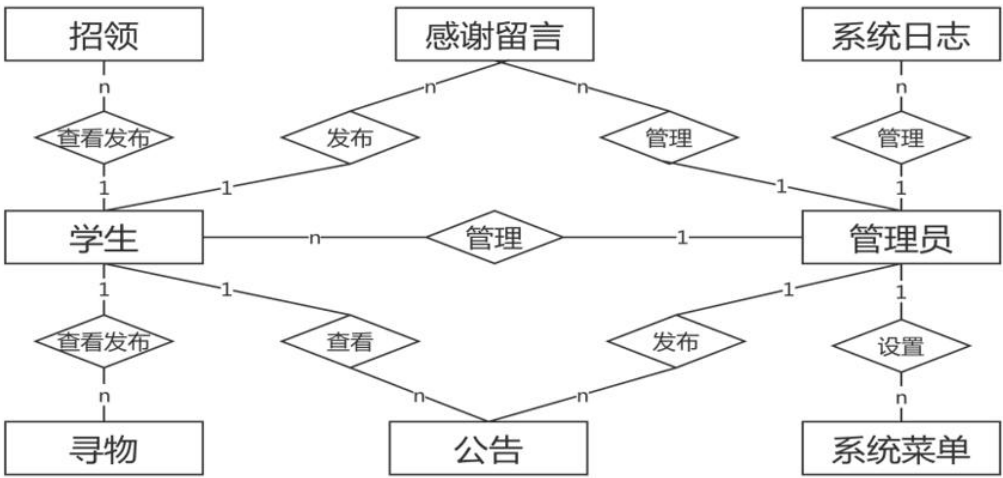
本校园失物招领系统拟按通用的 B/S（浏览器、服务器）模式进行设计，数据库在一个信息管理系统中占有非常重要的地位，数据构设计的好坏将直接对应用系统的效率以及实现的效生影响。合理的数据库结构设计可以提高数据存储的效率，保证数据的完整和一一致。在本校园失物招领系统当中，采用有着免费的开放源代码的 Web 应用服务器，属于轻量级应用服务器的 Tomcat, 以及 MySQL 数据库作为本校园失物招领系统的数据库。

### 3.2 概念模型设计



概念模型设计的典型方法是用 E-R 图方法，即用实体-联系模型表示。E-R 方法是用 E-R 图来描述显示世界，E-R 图包含三个基本成分:实体、联系、属性。它直观易懂，能够比较准确地反映现实世界的信息联系，从概念上表示一个数据库的信息组织情况。如下图中我们所看到的是本校园失物招领系统的 E-R 图，根据 E-R 图所示，我们可以直观准确的了解到本校园失物招领系统的图表之间的联系。

**E-R 图:**



### 3.3 数据库逻辑模型

本校园失物招领系统主要有五个表，分别如下：

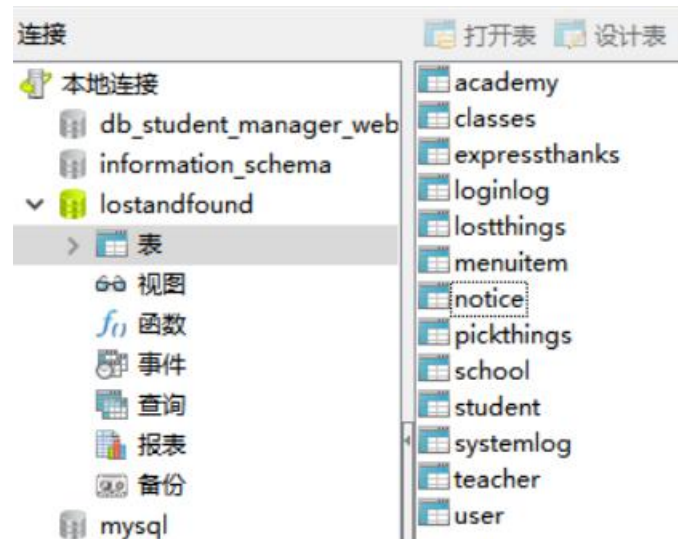
1 user 表：(主键、用户 ID、用户名、密码、年纪、姓名、昵称、联系电话、邮箱、QQ、状态、创建时间、最后一次登录)

2 pickthings 表：(主键、用户 ID、物品名称、拾取地点、拾取时间、物品类型、物品编号、物品图片、物品描述、存放地址、电话、发布时间、状态、用户名称、物品个数)

3 lostthings 表：（主键、物品名称、拾取地点、物品类型、物品编号、物品图片、物品描述、存放地址、电话、发布时间、状态、用户名称、物品个数）

4 loginlog 表：（主键、用户 ID、用户名称、系统版本、浏览器名称、登录 IP、登陆时间、物品个数）

5 notice 表：（主键、标题、内容、状态、发布时间）



## 3.4 数据库表结构

### 数据库设计

数据库的设计关系到整个应用系统的运行效率,数据库设计得好,不仅有利于日常数据的维护更新,而且可以提高系统的运行效率,缩短数据查询响应周期,增加网站的流量<sup>[9]</sup>。合理的数据库设计可以使围绕它支持的 Web 页面的 Java 代码简单化,易于实现,并且可以提高数据存储的效率,保证数据的完整一致。校园失物招领管理系统采用 MySQL 作为后台数据库开发工具。

### 数据库概念设计

概念模型用于信息世界的建模,与具体的 DBMS 无关。为了把现实世界中的具体事物抽象、组织为某一 DBMS 支持的数据模型。人们常常首先将现实世界抽象为信息世界,然后再将信息世界转换为机器世界。也就是说,首先把现实世界中的客观对象抽象为某一种信息结构,这种信息结构并不依赖于具体的计算机系统和具体的 DBMS,而是概念级的模型,然后再把模型转换为计算机上某一个 DBMS 支持的数据模型。实际上,概念模型是现实世界到机器世界的一个中间层次。

信息世界中包含的基本概念有实体和联系。

#### (1) 实体 (entity)

客观存在并可相互区别的事物称为实体。实体可以是具体的人、事、物,也可以是抽象的概念或联系。例如,一个学生、一门课、一个供应商、一个部门、一本书、一位读者等都是实体。

#### (2) 联系 (relationship)

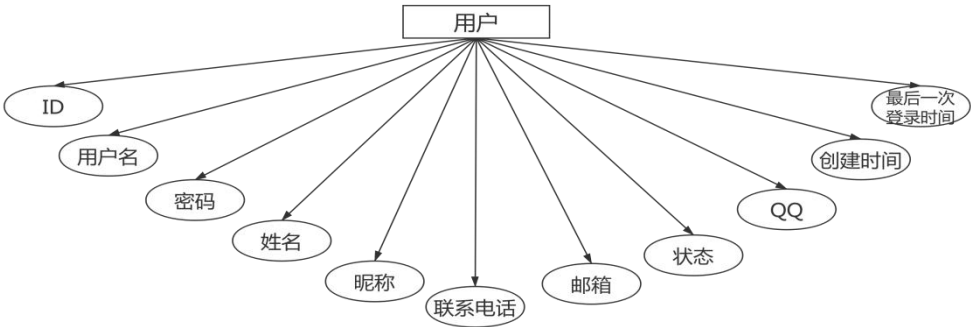
在现实世界中,事物内部以及事物之间是有联系的,这些联系在信息世界中反映为实体内部的联系和实体之间的联系。实体内部的联系通常是组成实体的各属性之间的联系。两个实体型之间的联系可以分为 3 类,一对一联系, (1:1); 一对多联系 (1 : n); 多对多联系 (m : n)。

概念模型是对信息世界建模,所以概念模型应该能够方便、准确地表示信息世界中的常用概念。概念模型的表示方法很多,其中最为常用的是 P. P. S. Chen 于 1976 年提出的实体,联系方法 (Entity-Relationship Approach) 简记为 E-R 表示法)。该方法用 E-R 图来描述现实世界的概念模型,称为实体-联系模型,简称 E-R 模型。根据数据流程分析,绘制校园失物招领管理系统的全局 E-R 模型如图 4-7 所示。

根据系统分析的主要实体有:用户(学生和管理员,用系统权限来区分)、

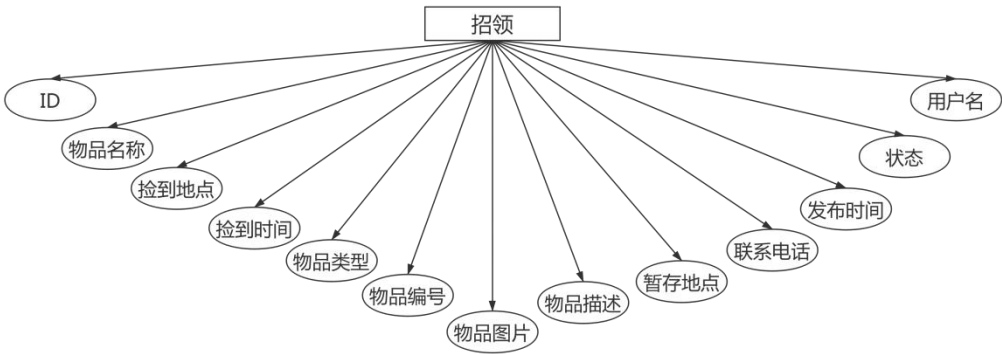
招领、寻物、感谢留言、公告、系统日志和系统菜单。各个实体具体的描述属性图如下（实体属性在下图中并没有全部给出，因为属性过多的原因）：

1. 用户实体



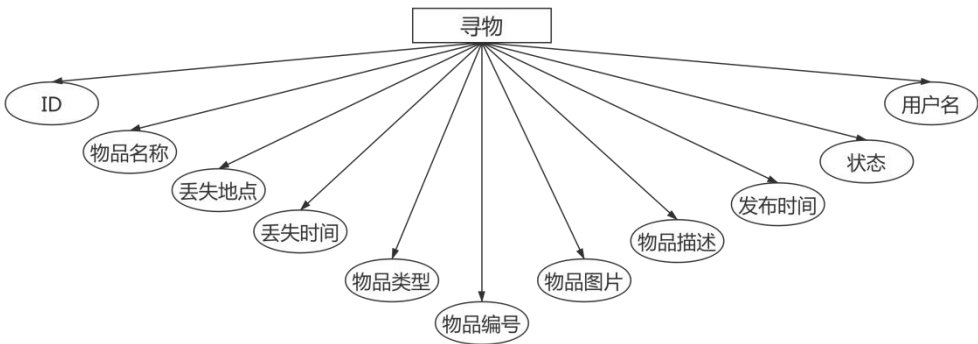
4-9 用户实体图

2. 招领实体



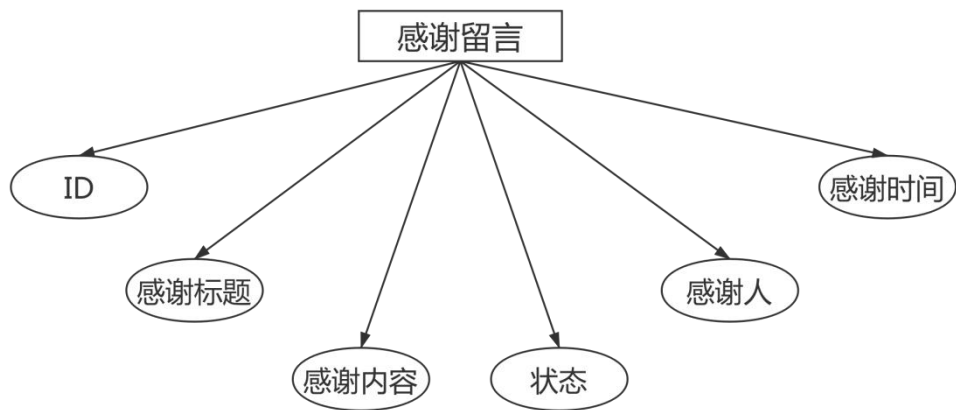
4-10 招领实体图

3. 寻物实体



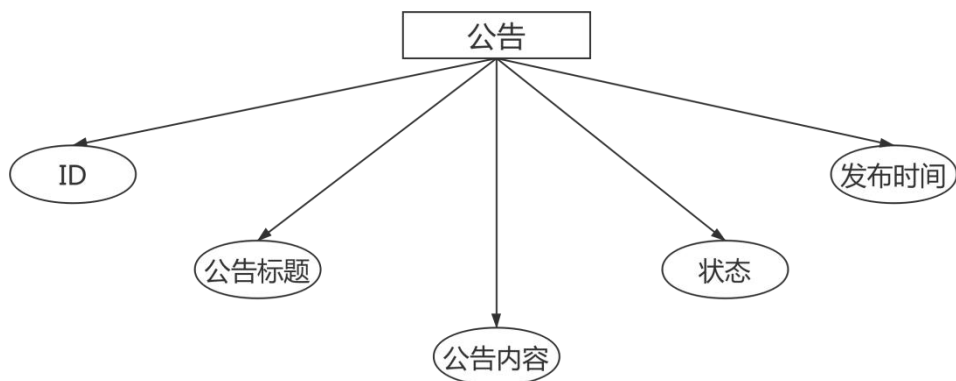
4-11 寻物实体图

4. 感谢留言实体



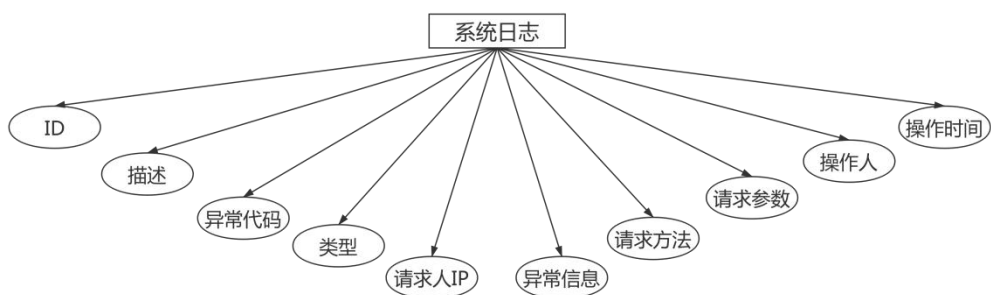
4-12 感谢留言实体图

## 5. 公告实体



4-13 公告实体图

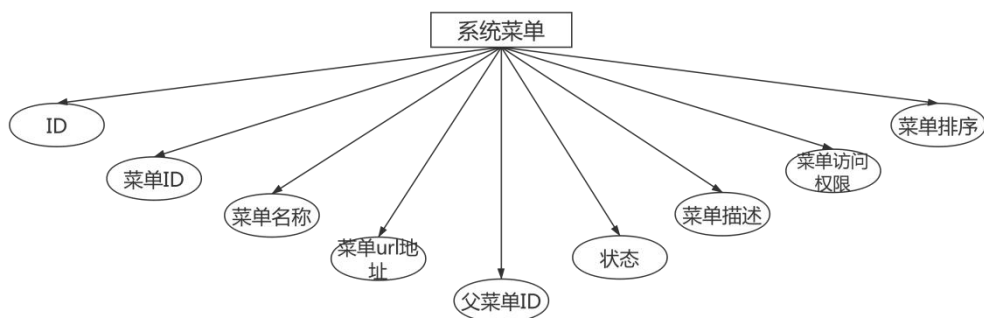
## 6. 系统日志实体



4-14 系统日志实体图

## 7. 系统菜单实体





4-15 系统菜单实体图

#### 4.4.2 数据库逻辑设计

数据库逻辑设计主要是把数据库概念设计时设计好的基本 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。它包括数据项、记录及记录间的联系、安全性和一致性约束等等。导出的逻辑结构是否与概念模式一致，从功能和性能上是否满足用户的要求，要进行模式评价。

本系统数据库名称为 lostandfound，数据库中包括：1) 用户表 (user) 2) 招领表 (pickthings) 3) 寻物表 (lostthings) 4) 感谢留言表 (expressthanks) 5) 公告表 (notice) 6) 系统日志表 (systemlog) 7) 系统菜单表 (menuitem)。各表数据结构如下：

(1) 用户表 (user)，存储用户信息

字段名称	数据类型	主键	是否空	说明
ID	bigint(20)	Y	N	用户 ID
UserName	varchar(20)	N	N	用户名
Password	varchar(50)	N	N	密码
Grade	varchar(10)	N	N	年级
Name	varchar(20)	N	Y	姓名
NickName	varchar(20)	N	Y	昵称
TelPhone	varchar(12)	N	N	联系电话
Email	varchar(30)	N	Y	邮箱
QQ	varchar(15)	N	Y	qq
State	int(11)	N	N	状态
CreateTime	date	N	N	创建时间
LastLoginTime	datetime	N	Y	最后一次登录时间

(2) 招领表 (pickthings)，存储招领信息

字段名称	数据类型	主键	是否空	说明
ID	bigint(11)	Y	N	招领 ID
ThingsName	varchar(30)	N	N	物品名称
PickPlace	varchar(30)	N	N	捡到地点
PickTime	date	N	N	捡到时间
ThingsType	varchar(15)	N	N	物品类型
ThingsNo	varchar(10)	N	Y	物品编号
ThingsImg	text	N	Y	物品图片
ThingsDes	text	N	N	物品描述
StoragePlace	text	N	N	暂存地点
TelPhone	varchar(30)	N	Y	联系电话
PublishTime	datetime	N	Y	发布时间
Status	int(11)	N	N	状态
UserName	varchar(20)	N	N	用户名

(3) 寻物表(lostthings)，存储寻物信息

字段名称	数据类型	主键	是否空	说明
ID	bigint(11)	Y	N	寻物 ID
ThingsName	varchar(30)	N	N	物品名称
LostPlace	varchar(30)	N	N	丢失地点
LostTime	date	N	N	丢失时间
ThingsType	varchar(15)	N	N	物品类型
ThingsNo	varchar(10)	N	Y	物品编号
ThingsImg	text	N	Y	物品图片
ThingsDes	text	N	N	物品描述
PublishTime	datetime	N	Y	发布时间
Status	int(11)	N	N	状态
UserName	varchar(20)	N	N	用户名

(4) 感谢留言表(expressthanks)，存储感谢留言信息

字段名称	数据类型	主键	是否空	说明
ID	bigint(20)	Y	N	感谢留言 ID
Title	varchar(30)	N	N	感谢标题
Substance	text	N	N	感谢内容
Status	int(11)	N	N	状态

UserName	varchar(20)	N	N	感谢人
LeaveTime	datetime	N	N	感谢时间

(5) 公告表 (notice)，存储系统公告信息

字段名称	数据类型	主键	是否空	说明
ID	int(11)	Y	N	公告 ID
title	varchar(50)	N	Y	标题
Substance	longtext	N	N	公告内容
Status	int(11)	N	N	状态
PublishTime	datetime	N	Y	发布时间

(6) 系统日志表 (systemlog)，存储系统日志比信息

字段名称	数据类型	主键	是否空	说明
ID	bigint(20)	Y	N	系统日志 ID
Description	longtext	N	N	描述
ExceptionCode	longtext	N	N	异常代码
Type	varchar(10)	N	Y	类型
RequestIp	varchar(50)	N	N	请求人 IP
ExceptionDetail	longtext	N	N	异常详细信息
Method	longtext	N	N	请求方法
Params	longtext	N	N	请求参数
CreateBy	longtext	N	N	操作人
CreateDate	datetime	N	N	操作时间

(7) 系统菜单表 (menuitem)，存储系统菜单信息

字段名称	数据类型	主键	是否空	说明
ID	int(20)	Y	N	系统菜单 ID
MenuItemName	varchar(20)	N	N	菜单名称
MenuItemUrl	varchar(30)	N	Y	菜单 url 地址
ParentId	varchar(20)	N	Y	父菜单 ID
Status	int(11)	N	N	状态
MenuItemDes	varchar(30)	N	Y	菜单描述
Authority	int(11)	N	N	菜单访问权限
Sort	int(11)	N	Y	菜单排序

## 第四章 运用设计

### 4.1 数据字典设计：无说明

### 4.2 安全保密设计

通过区分不同的访问者、不同的访问类型和不同的数据对象，进行分别对待而获得的数据库安全保密设计考虑。对数据库设计中涉及到的各种项目，如数据项、记录、系、文卷、模式、子模式等一般要建立起数据保护措施，以说明它的标识符、同义名及有关信息。数据库由专门数据库管理用员对数据库操作，需要注意以下几项安全问题：访问安全 、网络安全 、传输安全 、备份安全 、数据安全 。

### 4.3 数据库实施

#### 4.3.1 创建数据库

```
Create database public;
```

```
Create database lostandfound;
```

```
Create database users;
```

#### 4.3.2 创建表

##### 1. 数据库基本信息

```
/*
```

```
Navicat MySQL Data Transfer
```

```
Source Server          : 本地连接
```

```
Source Server Version : 50505
```

```
Source Host            : localhost:3306
```

```
Source Database        : lostandfound
```

```
Target Server Type     : MYSQL
```

```
Target Server Version : 50505
```

```
File Encoding          : 65001
```

\*/

## 2. user 表

-- Table structure for `user`

-- -----

DROP TABLE IF EXISTS `user`;

CREATE TABLE `user` (

  `ID` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT 'ID 自增',

  `UserName` varchar(20) NOT NULL COMMENT '用户名',

  `Password` varchar(50) NOT NULL COMMENT '密码',

  `Grade` varchar(10) NOT NULL COMMENT '年级',

  `Name` varchar(20) DEFAULT NULL COMMENT '姓名',

  `NickName` varchar(20) DEFAULT NULL COMMENT '昵称',

  `TelPhone` varchar(12) NOT NULL COMMENT '联系电话',

  `Email` varchar(30) DEFAULT NULL COMMENT '邮箱',

  `QQ` varchar(15) DEFAULT NULL COMMENT 'qq',

  `State` int(11) NOT NULL COMMENT '状态',

  `CreateTime` date NOT NULL COMMENT '创建时间',

  `IsNew` int(11) NOT NULL DEFAULT '0' COMMENT '原始密码 (0)

修改后 (1)',

  `LastLoginTime` datetime DEFAULT NULL COMMENT '最后一次登录时间',

PRIMARY KEY (`ID`) USING BTREE,

```
    UNIQUE KEY `UserName` (`UserName`) USING BTREE
) ENGINE=InnoDB AUTO_INCREMENT=850 DEFAULT CHARSET=utf8
ROW_FORMAT=COMPACT;
```

---

### 3 lostthings 表

---

-- Table structure for `lostthings`

---

```
DROP TABLE IF EXISTS `lostthings`;

CREATE TABLE `lostthings` (
  `ID` bigint(11) NOT NULL AUTO_INCREMENT COMMENT 'LostThings
ID 自增',
  `ThingsName` varchar(30) NOT NULL COMMENT '物品名称',
  `LostPlace` varchar(30) NOT NULL COMMENT '丢失地点',
  `LostTime` date NOT NULL COMMENT '丢失时间',
  `ThingsType` varchar(15) NOT NULL COMMENT '物品类型',
  `ThingsNo` varchar(10) DEFAULT NULL COMMENT '物品编号',
  `ThingsImg` text COMMENT '物品图片 可 null',
  `ThingsDes` text NOT NULL COMMENT '物品描述',
  `PublishTime` datetime NOT NULL COMMENT '发布时间',
  `Status` int(11) NOT NULL COMMENT '状态',
```

```

`UserName` varchar(20) NOT NULL COMMENT '用户名',
`UID` bigint(20) NOT NULL COMMENT '用户表 ID 外键',
PRIMARY KEY (`ID`) USING BTREE,
KEY `UID` (`UID`) USING BTREE,
CONSTRAINT `lostthings_ibfk_1` FOREIGN KEY (`UID`)
REFERENCES `user` (`ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=212 DEFAULT CHARSET=utf8
ROW_FORMAT=COMPACT;

```

-- -----

#### 4.pickthings 表

```
-- Table structure for `pickthings`
```

-- -----

```

DROP TABLE IF EXISTS `pickthings`;
CREATE TABLE `pickthings` (
  `ID` bigint(11) NOT NULL AUTO_INCREMENT COMMENT 'LostThings
ID 自增',
  `ThingsName` varchar(30) NOT NULL COMMENT '物品名称',
  `PickPlace` varchar(30) NOT NULL COMMENT '丢失地点',
  `PickTime` date NOT NULL COMMENT '丢失时间',
  `ThingsType` varchar(15) NOT NULL COMMENT '物品类型',

```

```
`ThingsNo` varchar(10) DEFAULT NULL COMMENT '物品编号',
`ThingsImg` text COMMENT '物品图片',
`ThingsDes` text NOT NULL COMMENT '物品描述',
`StoragePlace` text COMMENT '暂存地点',
`TelPhone` varchar(30) DEFAULT NULL COMMENT '联系电话',
`PublishTime` datetime DEFAULT NULL COMMENT '发布时间',
`Status` int(11) NOT NULL COMMENT '状态',
`UserName` varchar(20) NOT NULL COMMENT '用户名',
`UID` bigint(20) NOT NULL COMMENT '用户表 ID 外键',
PRIMARY KEY (`ID`) USING BTREE,
KEY `UID` (`UID`) USING BTREE,
CONSTRAINT `pickthings_ibfk_1` FOREIGN KEY (`UID`)
REFERENCES `user` (`ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=272 DEFAULT CHARSET=utf8
ROW_FORMAT=COMPACT;
```

-- -----