

Sofu 数据库设计说明书

项目名称：Sofu（搜福）

项目团队：从 0 到 1 队

目录

1.引言	4
1.1 编写目的	4
1.2 项目背景	4
1.3 参考资料	4
2.外部设计	5
2.1 数据库名称与状态	5
2.2 数据库使用的软件	5
2.3 数据库命名约定	5
3.结构设计	5
3.1 概念结构设计	5
3.1.1 ER 图设计思路	5
3.1.2 ER 图	6
3.2 逻辑结构设计	6
4.应用设计	7
4.1 数据字典设计	7
4.1.1 sof_u_user	7
4.1.2 sof_u_question	7
4.1.3 sof_u_idea	8

4.1.4 sofuf_classify	8
4.1.5 sofuf_comment	9
4.1.6 sofuf_answer	9
4.1.7 sofuf_report	10
4.1.8 sofuf_admin	10
4.1.9 sofuf_usertaste	10
4.1.10 sofuf_like	11
4.1.11 sofuf_collect	11
4.1.12 sofuf_msg	11
4.1.13 sofuf_info	12
4.1.14 sofuf_concern	12
4.2 安全保密设计	12
4.2.1 防止 SQL 的注入	12
4.2.2 备份数据库	12
5.数据库验证验收标准	13
5.1 数据库数据体的验收	13
5.2 数据库安全性的验收	13

1.引言

1.1 编写目的

本说明书旨在对我们团队的作品——Sofu(搜福)进行较为系统的数据库设计说明。主要设计模块包括项目的外部设计、结构设计、应用设计以及数据库验证验收标准各大模块。本说明书能够简明体现出数据库设计的总体情况，有助于开发人员的开发与部署。

本说明书的读者为：系统开发、测试、维护人员。

1.2 项目背景

软件名称：Sofu(搜福)

项目开发者：从0到1队

背景介绍：

我们团队经过包括但不限于问卷调查、线上咨询等形式的调研，最终发现当前福大学子的求助类信息基本通过互助群、帮帮墙发布。而对于互助群，容易出现消息被覆盖、信息杂乱、搜索功能不便、人数上限低、而且时常会混入不明人员扰乱群聊等问题。而对于帮帮墙也是如此，运营者的精力有限，难以将同学们的提问及时且大量地发布出去，并且没有进行信息分类，大多以图片的形式进行发布，学生也无法进行搜索。

于是，我们想要开发出一款专属于福大的校园问答类的平台供大家使用，解决上述的所有问题，使得每位同学的提问都能得到及时的回应。除了核心功能（问答）以外，我们的产品还提供给用户分享想法的功能，同步校园动态，营造认真、专业、友善的社区氛围。

1.3 参考资料

《数据库设计说明书》国际规范文本

2.外部设计

2.1 数据库名称与状态

数据库名称: sofu

数据库状态: 暂停

2.2 数据库使用的软件

MySQL

2.3 数据库命名约定

数据库表名: 数据库名_表的含义名称(如:sofu_user)

数据库字段命名: 涉及到表需要连接的地方, 该字段若出现在两张表中且同名, 需要加上表名作为前缀(如 user_id)

3.结构设计

3.1 概念结构设计

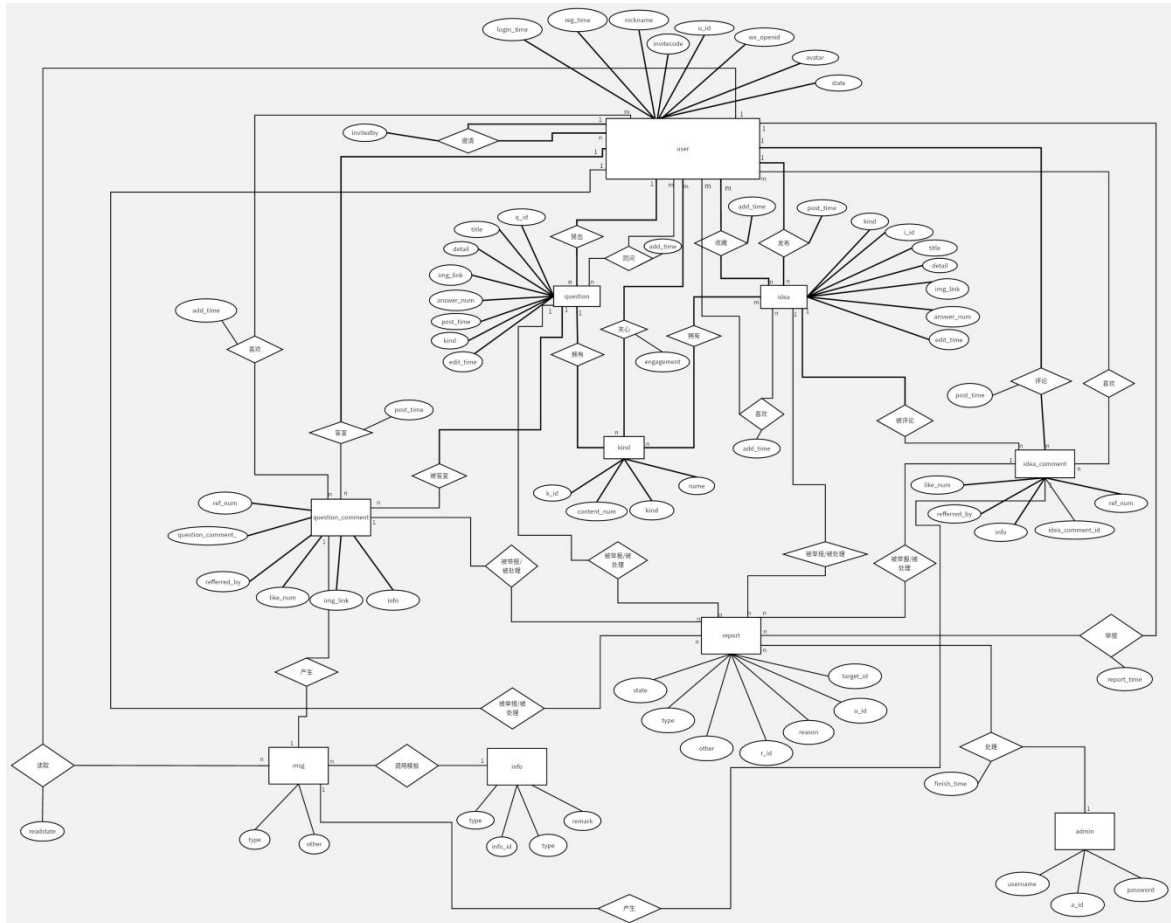
3.1.1 ER 图设计思路

- 1.确定实体: 识别系统中需要存储数据的实体。
- 2.确定属性: 确定每个实体所拥有的属性。
- 3.确定关系: 确定实体之间的关系。
- 4.绘制 ER 图: 使用实体、属性和关系构建 ER 图, 其中实体用矩形表示, 属性用椭圆形表示, 关系用菱形表示, 并用线连接它们。
- 5.优化 ER 图: 根据需求对 ER 图进行优化, 例如合并重复实体或简化关系。
- 6.实现数据库: 按照 ER 图设计的规范来创建数据库表格, 将实体转换为表格, 属性转换为列,

关系转换为外键。

7.测试和修改：测试数据库是否符合预期，并对其进行必要的修改。

3.1.2 ER 图



3.2 逻辑结构设计

表结构设计思路

- 1.明确需求：确定表需要存储哪些数据、如何查询数据以及数据的关系等。
- 2.设计范式：根据范式理论，规范化表结构，避免数据冗余和不一致性。
- 3.选择合适的数据类型：根据数据的实际情况选择合适的数据类型和长度。
- 4.建立主外键关系：通过建立主键和外键关系保证数据的完整性和一致性。
- 5.考虑索引：根据查询需求创建适当的索引，提高查询效率。
- 6.定义约束：定义适当的约束，保证数据的完整性和正确性。
- 7.优化性能：根据具体情况对表进行必要的优化，如分区、压缩等。

4.应用设计

4.1 数据字典设计

4.1.1 sofu_user

字段名称	数据类型	描述
user_id	int	用户 id 【KEY】
nickname	varchar(255)	昵称
avatar	varchar(255)	头像链接地址
login_time	datetime	登录时间
reg_time	datetime	注册时间
state	int	用户状态
invitedby	int	邀请该用户的用户 id[外键]
invitecode	varchar(255)	邀请码
wx_openid	varchar(255)	微信登录需要的 id

4.1.2 sofu_question

字段名称	数据类型	描述
q_id	int	问题 id 【KEY】
q_title	varchar(255)	问题标题
img_link	varchar(255)	问题图片的链接
answer_num	int	问题回答数
classify_id	int	类别 id[外键]
post_time	datetime	发布日期
concern_num	int	同问数
edit_time	datetime	修改时间

q_info	text(1023)	问题内容
user_id	int	用户 id[外键]

4.1.3 sofu_idea

字段名称	数据类型	描述
idea_id	int	想法 id【KEY】
idea_title	varchar(255)	想法标题
img_link	varchar(255)	想法图片的链接
comment_num	int	评论数
post_time	datetime	发布时间
like_num	int	喜欢数
edit_time	datetime	修改时间
classify_id	int	类别 id[外键]
idea_info	text(1023)	想法内容
user_id	int	用户 id[外键]
collect_num	int	收藏数

4.1.4 sofu_classify

字段名称	数据类型	描述
classify_id	int	分类 id【KEY】
name	varchar(255)	类别名称
type	int	类型
connect_num	int	该分类下所拥有资源数量

4.1.5 sofu_comment

字段名称	数据类型	描述
com_id	int	评论 id 【KEY】
idea_id	int	从属想法帖子的 id[外键]
com_info	varchar(255)	评论内容
like_num	int	喜欢数
ref_num	int	被其他评论@的数量
post_time	datetime	评论时间
user_id	int	用户 id
reply_com_id	int	@别人回答的评论 id(允许为空)

4.1.6 sofu_answer

字段名称	数据类型	描述
ans_id	int	回答 id 【KEY】
q_id	int	从属问题的 id[外键]
ans_info	varchar(255)	文本信息
img_link	varchar(255)	图片链接 Strign[]
like_num	int	喜欢数
ref_num	int	被其他回答@的数量
reply_ans_id	int	@别人回答的 id(允许为空)
user_id	int	回答者的 id
post_time	datetime	回答时间

4.1.7 sofu_report

字段名称	数据类型	描述
r_id	int	举报 id 【KEY】
type	int	举报对象类型
other	varchar(255)	补充信息
state	int	状态
user_id	int	举报者 id[外键]
targetid	int	被举报内容 id[外键]
report_time	datetime	举报时间
finish_time	datetime	处理结果反馈时间
admin_id	int	处理的管理员 id[外键]
reason	varchar(255)	举报原因

4.1.8 sofu_admin

字段名称	数据类型	描述
id	int	管理员 id 【KEY】
admin_acc	varchar(255)	管理员账号
admin_pwd	varchar(255)	管理员密码

4.1.9 sofu_usertaste

字段名称	数据类型	描述
id	int	项 id 【KEY】
classify_id	int	分类 id[外键]
user_id	int	用户 id[外键]
engagement	int	参与度

4.1.10 sofu_like

字段名称	数据类型	描述
id	int	项 id 【KEY】
match_id	int	对应贴的 id[外键]
user_id	int	用户 id[外键]
add_time	datetime	喜欢时间
type	int	类别区分(1 表示回答喜欢 2 表示想法评论喜欢)

4.1.11 sofu_collect

字段名称	数据类型	描述
id	int	项 id 【KEY】
idea_id	int	ideaId[外键]
user_id	int	用户 id[外键]
add_time	datetime	收藏时间

4.1.12 sofu_msg

字段名称	数据类型	描述
m_id	int	消息 id 【KEY】
type	int	类型(官方消息 回答被@ 评论被@)
user_id	int	用户 id[外键]
other	varchar(255)	其他项
readstate	int	被读状态
info	varchar(255)	消息内容

4.1.13 sof_u_info

字段名称	数据类型	描述
id	int	官方提示 id 【KEY】
text	varchar(255)	文本模板
remark	varchar(255)	备注
type	int	类型

4.1.14 sof_u_concern

字段名称	数据类型	描述
id	int	项 id 【KEY】
q_id	int	同问问题贴的 id[外键]
user_id	int	用户 id[外键]
add_time	datetime	同问时间

4.2 安全保密设计

4.2.1 防止 SQL 的注入

- 1.前端对输入框的特殊字符进行转义过滤。
- 2.对用户的操作进行权限控制，防止非法操作。

4.2.2 备份数据库

- 1.定期对数据库进行备份。
- 2.在更新版本迭代前进行数据库备份。

5.数据库验证验收标准

5.1 数据库数据体的验收

- 1.数据准确性检查：验证数据是否完整、准确、一致，并排除数据重复或错误。
- 2.数据质量评估：对数据进行质量评估，包括数据的完整性、精确性、一致性等方面的评估。
- 3.性能测试：测试数据库系统在高并发、大数据量情况下的运行性能，包括响应时间、吞吐量 4.等指标。
- 5.安全性检查：检查数据库系统的安全性能是否符合相关的安全要求和标准，如访问控制、数据加密等。
- 6.用户界面测试：测试数据库系统的用户界面是否易用、清晰明了，并验证用户所需功能是否都能正常使用。
- 7.兼容性测试：测试数据库系统与其他系统的兼容性，特别是与已有的系统集成时是否能够正常工作。
- 8.可靠性测试：测试数据库系统的可靠性和稳定性，包括系统崩溃后的恢复能力和数据丢失情况等。

5.2 数据库安全性的验收

- 1.访问控制：确保只有授权用户能够访问数据库，可以采用密码、加密等方式进行用户身份验证。
- 2.数据加密：对敏感数据进行加密处理，防止未经授权的访问和窃取，常见的加密算法有 AES、RSA 等。
- 3.数据备份与恢复：定期备份数据库，并测试恢复过程，以确保在数据库遭受灾难性损失时能够快速恢复。
- 4.安全审计：记录数据库的操作日志，及时发现和应对潜在的安全问题。
- 5.漏洞管理：持续跟踪并修复数据库软件存在的漏洞，确保数据库系统不会因为漏洞被攻击而受到损害。
- 6.数据库完整性：保证数据库中数据的完整性，防止误删、篡改等情况发生。