**实验一：数据库系统开发**

**1. 实验目的**

在熟练掌握MySQL基本命令、SQL语言以及用C语言编写MySQL操作程序的基础上，学习简单数据库系统的设计方法，包括数据库概要设计、逻辑设计。

**2. 实验环境**

Windows XP操作系统、MySQL 关系数据库管理系统、MinGW编译器或Microsoft Visual C++编译器。

本次实验可使用C，C++，JAVA，PHP或其他语言均可。

**3. 实验内容**

开发一个数据库系统，可以参考教材的例子。

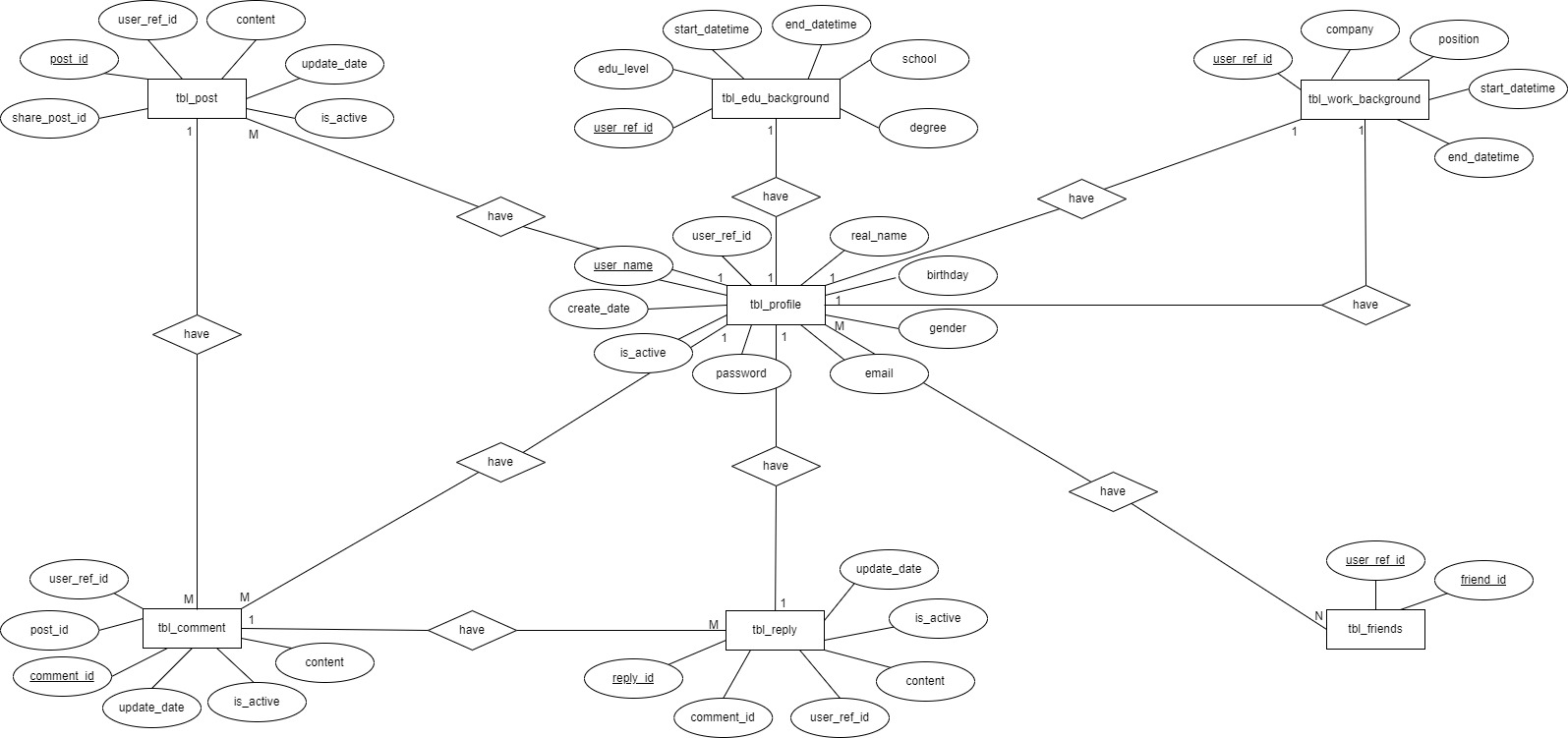
**3.1 要求**

1. 该系统的E-R图至少包括8个实体和7个联系（必须有一对一联系、一对多联系、多对一联系）。
2. 在设计的关系中需要体现关系完整性约束：主键约束、外键约束，空值约束。
3. 对几个常用的查询创建视图、并且在数据库中为常用的属性（非主键）建立索引。
4. 该系统功能必须包括：插入、删除、连接查询、嵌套查询、分组查询。其中插入，删除操作需体现关系表的完整性约束，例如插入空值、重复值时需给予提示或警告等。
5. 加分项：界面友好、包含事务管理、触发器等功能。

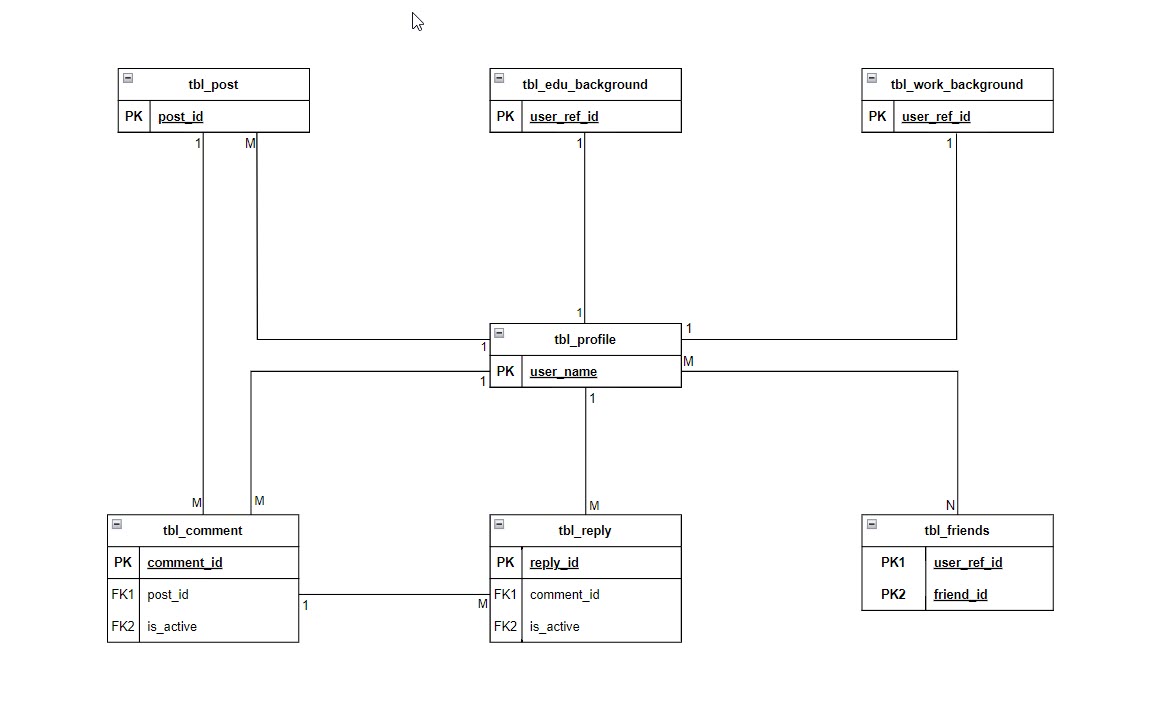
**3.2数据库设计**

社会网络数据库的模式一共有7个：

1. 用户信息 (用户号，用户名，实名，生日，性别，电子邮箱，密码， 有效，注册日期)
2. 学习经历（用户号，年纪，开始，结束，学校，学位）
3. 工作经验（用户号，公司名，位置，开始，结束）
4. 日志（日志号，用户号，内容，修改日期，有效，分享日志）
5. 日志的回复（日志回复号，日志号，用户号，内容，有效，修改日期）
6. 回复日志的回复（回复号码，日志回复号，用户号，内容，有效，修 改日期）
7. 好友（用户号，好友号）

**3.3 ER图**

**3.4 关系表**



**3.5创建的视图**

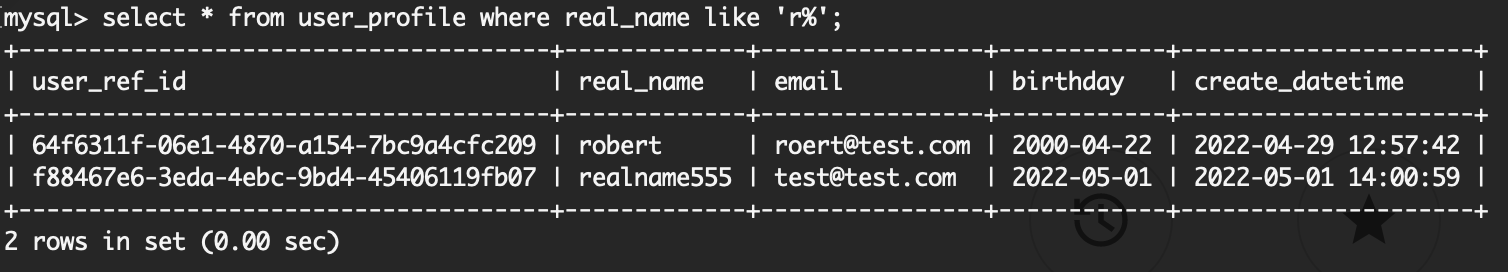
**创建某用户的视图:**

create view

user\_profile(user\_ref\_id, real\_name, email, birthday, create\_datetime)

as select user\_ref\_id, real\_name, email, birthday, create\_datetime

from tbl\_profile;

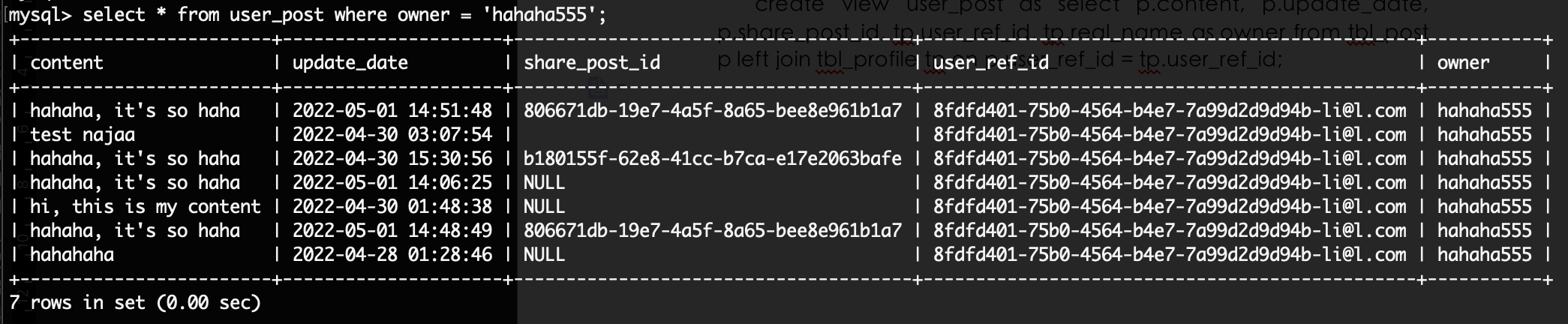
****

**创建某用户日志的视图:**

create view user\_post

as select p.content, p.update\_date, p.share\_post\_id, tp.user\_ref\_id, tp.real\_name as owner

from tbl\_post p left join tbl\_profile tp on p.user\_ref\_id = tp.user\_ref\_id;

****

**创建某用户信息的视图:**

create view

user\_info as

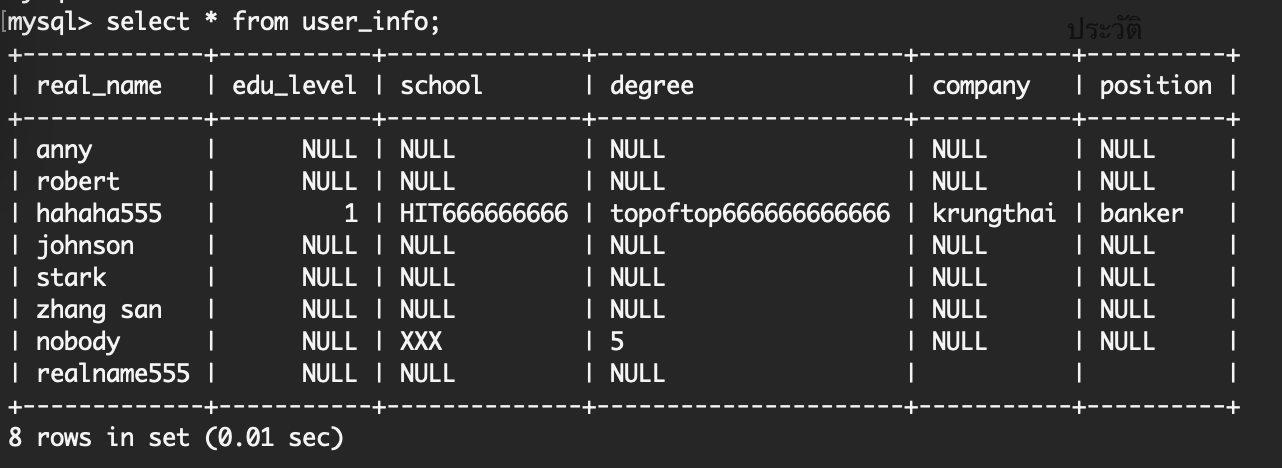
select p.real\_name,edu.edu\_level, edu.school, edu.degree,

w.company, w.position

from tbl\_edu\_background edu

inner join tbl\_work\_background w on edu.user\_ref\_id = w.user\_ref\_id

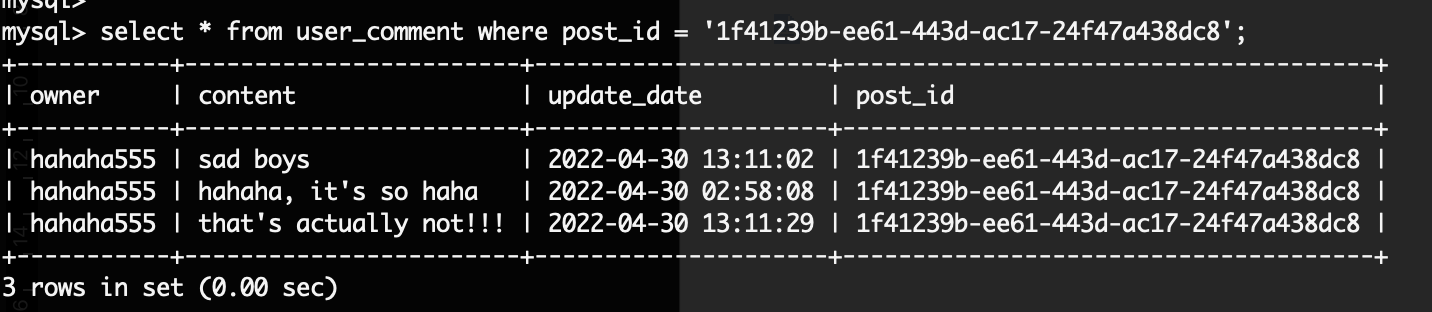
left join tbl\_profile p on edu.user\_ref\_id = p.user\_ref\_id;



**创建某用户日志回复的视图:**

create view user\_comment as

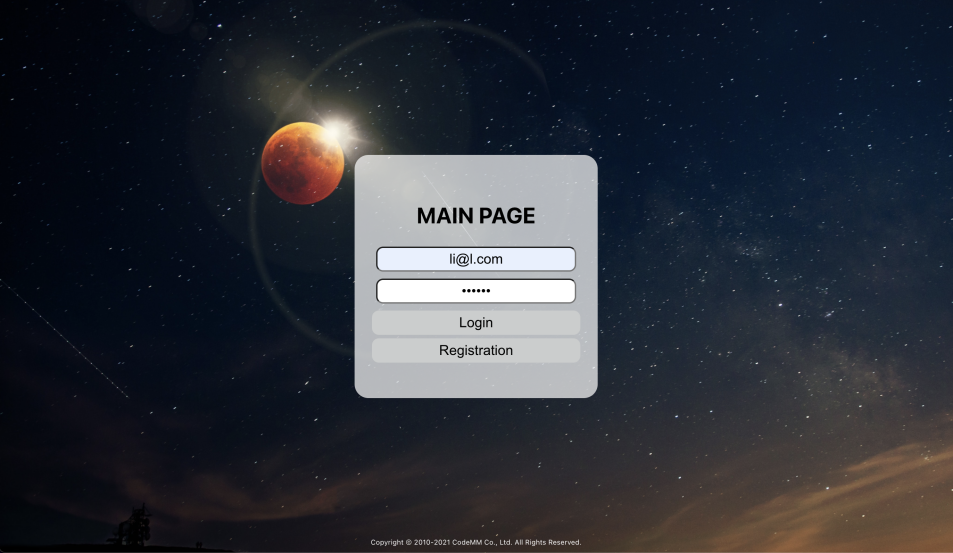
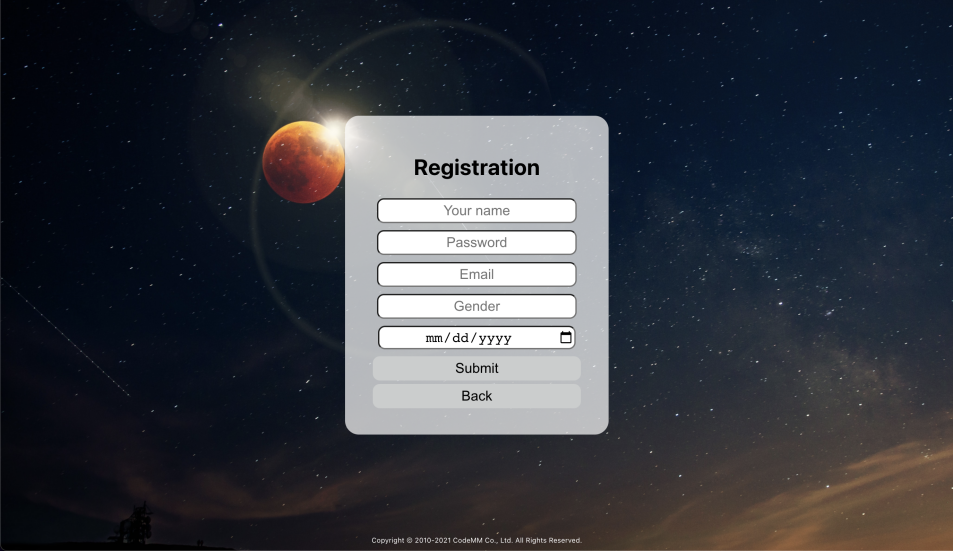
select tp.real\_name as owner, c.content, c.update\_date, c.post\_id from tbl\_comment c left join tbl\_profile tp on c.user\_ref\_id = tp.user\_ref\_id;



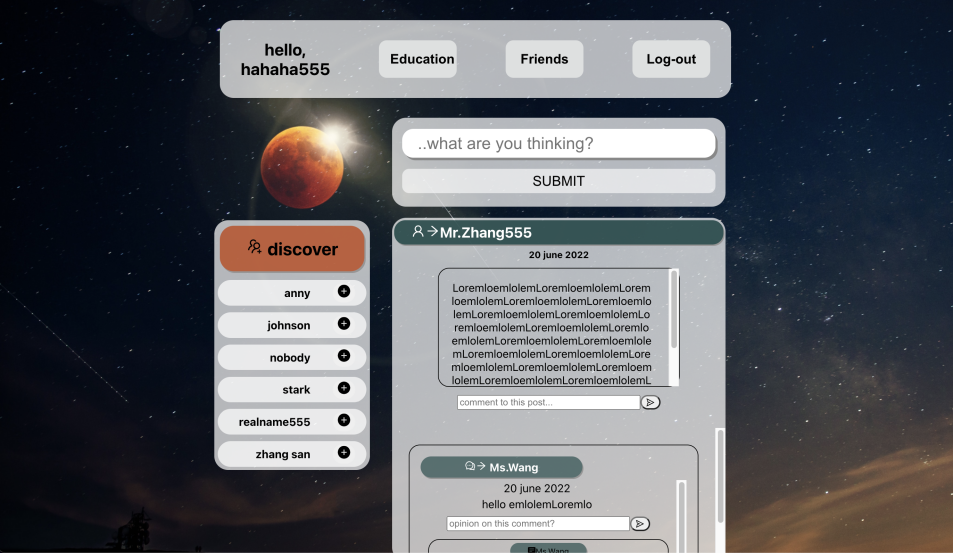
1. **运行界面**

用户注册输入： username，password, email, birthday,gender

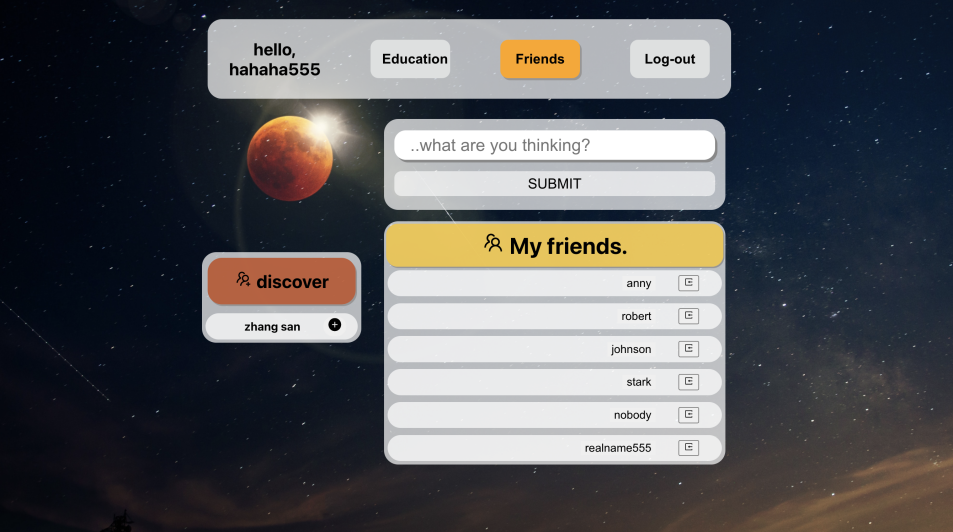
用户登入：username或者emaili，password



主页：



我的好友：



**遇到的问题**

本次实验要把数据显示到浏览器，我本人对Java脚本不太了解，所以就尽量完成了前端。主要系统可以通过后端，apiTestCase.json。

