



哈爾濱工業大學 (深圳)
HARBIN INSTITUTE OF TECHNOLOGY

实验报告

开课学期: 2022 秋季
课程名称: 数据库系统
实验名称: **** 系统设计与实现
实验性质: 设计型
实验学时: 6 地点: T2 205
学生班级: 5
学生学号: 200110529
学生姓名: 梁爽
评阅教师:
报告成绩:

实验与创新实践教育中心制

2022 年 11 月

1 实验环境

请填写用到的操作系统和主要开发工具。

windows 系统，开发语言：java,mysql

开发工具:datagrip

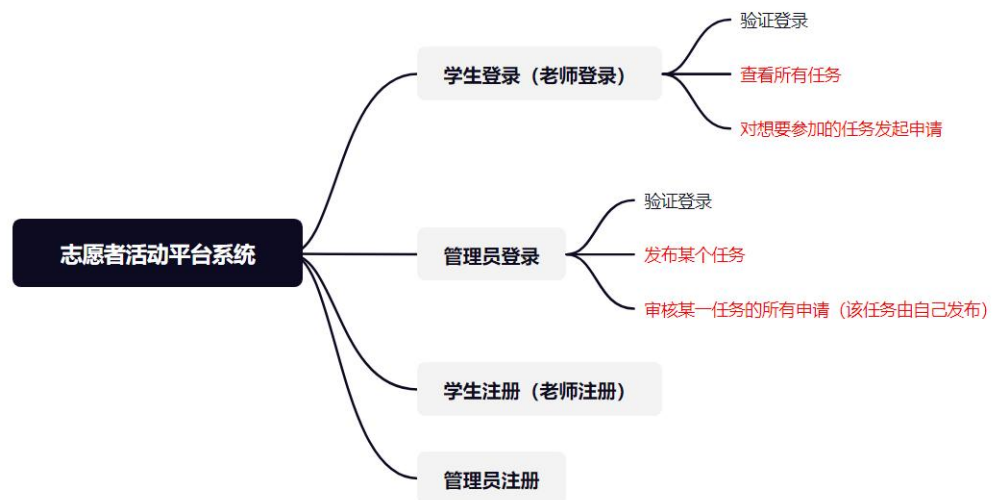
intellij IDEA

powerdesigner

2 实验过程

2.1 系统功能

请结合文字、图表等方式清晰描述系统的功能。如有亮点功能请用*标志。



总体的界面：



用户注册：

学号:5478

确认

密码:123

姓名:小小

消息

欢迎您,您的用户编号是11,您的姓名是:小小,您已经注册成功

确定

学生登录:

登录帐号:5478

确认

登录密码:123

消息

登录成功

确定

密码正确就会

消息

你的密码貌似错了噢, 请重新输入

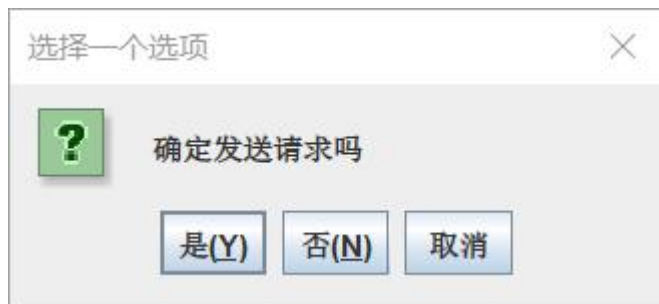
确定

密码失败就会

*登录成功后可以享受的功能:

活动编号	管理员编号	活动时间	活动地点	活动需募人数	活动发布时间	活动名称
1	1	2012.8.9 12点	点津镇	10	2012.8.5 11点	捉鬼大队行动
2	2	2012.8.9 12点	点津镇	10	2012.8.5 11点	捉鬼大队行动
3	3	明天	点津镇	5	昨天	捉鬼
4	3	2022年12月25日晚7点	教学楼T4 502	12	2022年12月1日早上8点	圣诞节派对

查看所有活动的编号，时间，需要人数，发布时间等等，并且发送关于此活动的请求。



我们的管理员会得到这个请求的噢。

管理员登录，一样的



*管理员功能一：发布任务：

public_acti...

确认录入并发布活动信息

活动时间: 2022年12月30日中午五点

活动名称: 吃饭

活动人数: 15

活动地点: 荔园

活动发布时间: 2022年12月27日

消息

已经录入成功

确定

管理员功能二：审核任务（审核的是自己发布的任务对应的申请，就是有用户申请参加自己发布的活动，就能看到）

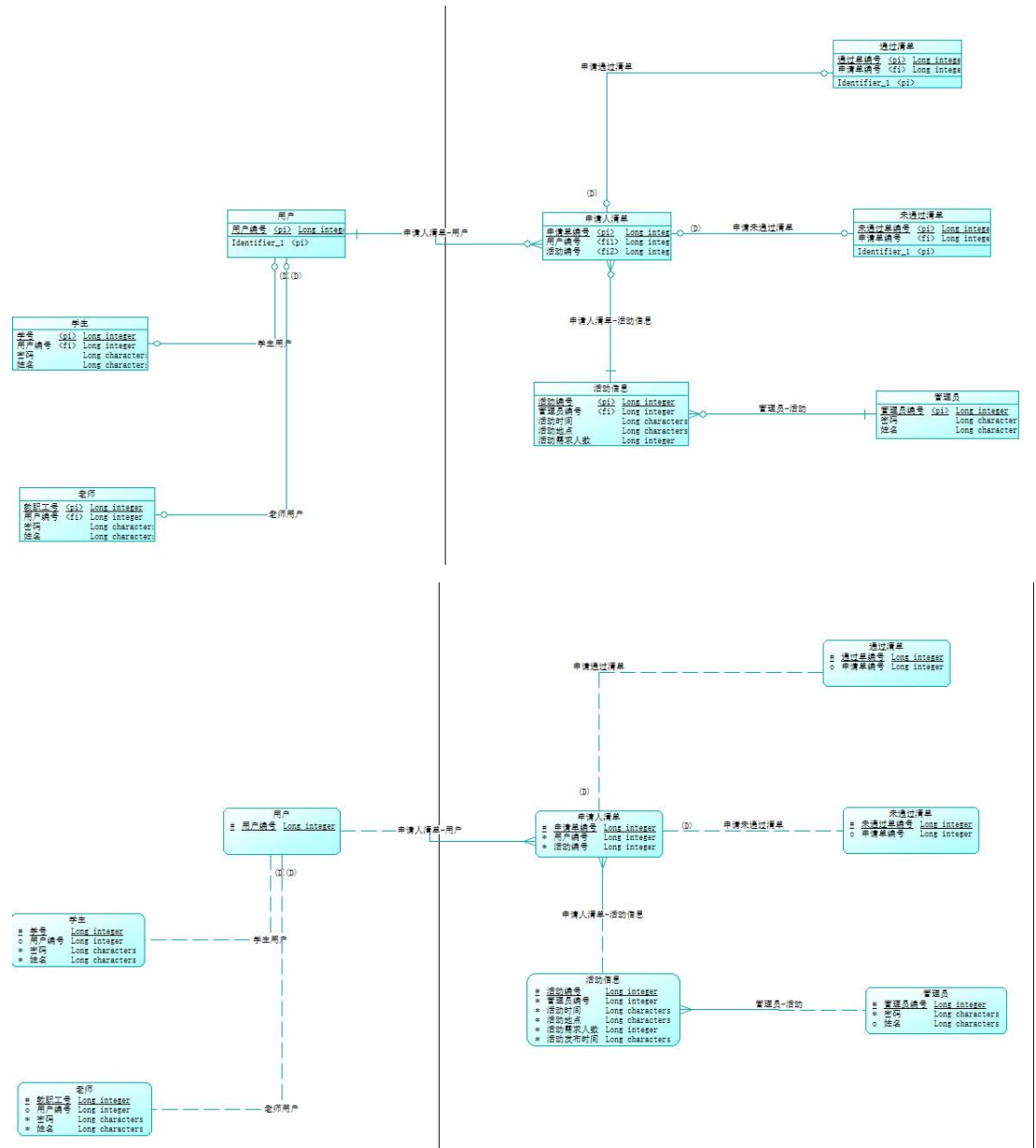
申请号	用户编号	申请的活动编号	申请时间	负责的管理员编号	活动时间	活动地点	需要人数	活动名称
8	8	1	2022-12-29 19:28:27	1	2012.8.9 12点	荔湾镇	10	捉鬼大队行动
10	11	1	2022-12-29 20:13:00	1	2012.8.9 12点	荔湾镇	10	捉鬼大队行动

审核通过 审核不通过

管理员可以自主选择通过或者不通过，在下方有两个按钮

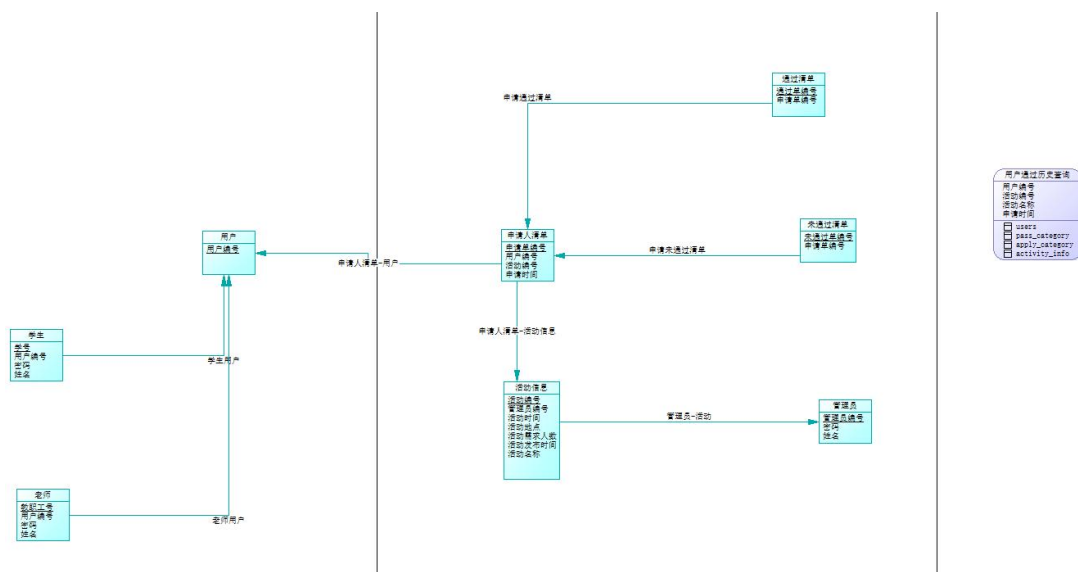
2.1.2 LDM 图

要求：截图务必清晰，如果图太大可截图一个总图，然后再分块截图。如果看不清截图会影响成绩。



2.1.3 PDM 图

要求：截图务必清晰，如果图太大可截图一个总图，然后再分块截图。如果看不清截图会影响成绩。



唯一约束：活动的名字是唯一的

外键约束:manager_id，在进行活动建立时，所使用的 manager_id 必须在 manager 表中已有，必须有已经存在的管理员才行。

Apply_category 表（即用户申请参与活动的申请表）：

	apply_id	user_id	activity_id	apply_time
1	1	6	2	2022-12-27 21:05:48
2	2	6	2	2022-12-27 21:07:21
3	3	6	2	2022-12-27 21:09:19
4	4	6	1	2022-12-27 22:22:43
5	5	8	4	2022-12-28 18:31:51
6	6	8	2	2022-12-28 18:31:55
7	7	8	3	2022-12-28 18:32:00

```
-- auto-generated definition
create table apply_category
(
    apply_id    int            not null
               primary key,
    user_id     int            not null,
    activity_id int            not null,
    apply_time  varchar(100) not null,
    constraint apply_category_users_null_fk
        foreign key (user_id) references users (user_id)
);
```

主键约束：apply_id,代表申请的编号

非空约束：所有属性，user_id,activity_id 等，这些皆不能为空

外键约束：user_id，申请表中的所有申请，其发起人的 user_id 必定是 user 表中已有的 user_id 记录。

2、索引

1) 索引截图

indexes 3
i activity_name (activity_name) UNIQUE
i PRIMARY (activity_id) UNIQUE
i activity_info_manager_null_fk (mana

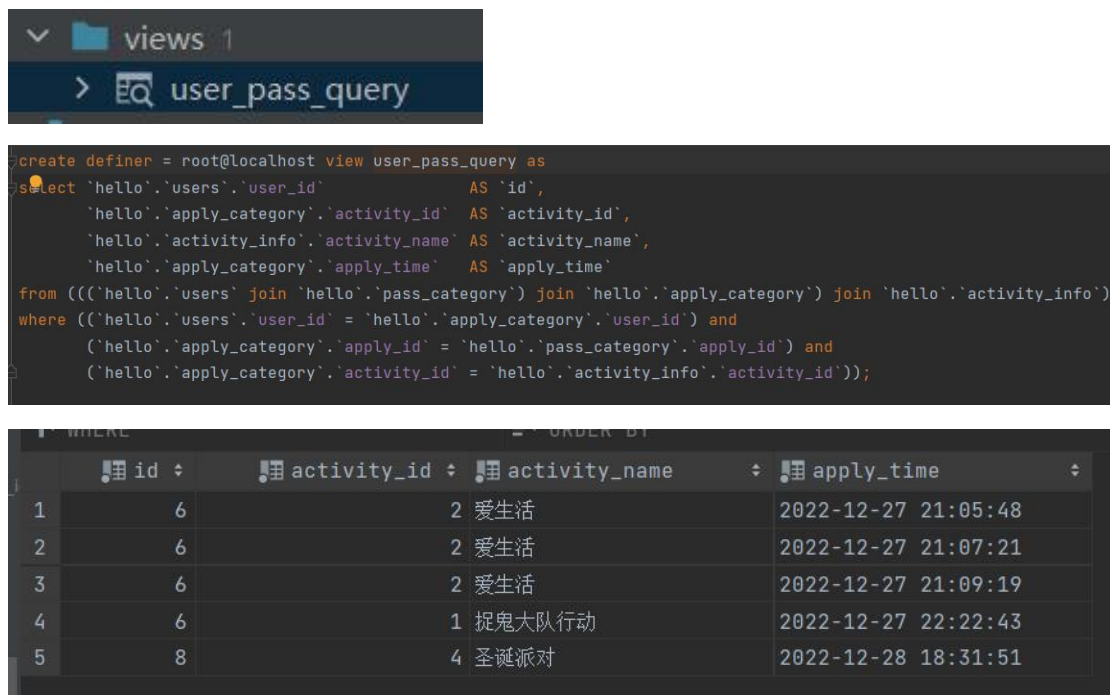
activity_name 是我设计的除了主键，外键索引以外的索引，他是一个唯一索引

2) 使用场景（用途）

按照真实场景，所有活动应该避免重名造成歧义，所以活动的名字应该是唯一的，并且简明扼要提示活动信息。这个索引的使用用途就是如此，因为名字的唯一性，和活动名字经常提起的特性，这个索引设计的也有一点意义。

3、 视图

1) 视图截图



The screenshot displays a database management tool interface. At the top, a folder icon labeled 'views' is expanded, showing a view named 'user_pass_query'. Below this, the SQL code for creating the view is shown in a dark-themed editor. The code defines a view named 'user_pass_query' that selects data from several tables: 'users', 'pass_category', 'apply_category', and 'activity_info'. The selection criteria include user_id, activity_id, activity_name, and apply_time. The view is created by a user named 'root' on a 'localhost' machine.

```
create definer = root@localhost view user_pass_query as
select 'hello`.`users`.`user_id` AS `id`,
'hello`.`apply_category`.`activity_id` AS `activity_id`,
'hello`.`activity_info`.`activity_name` AS `activity_name`,
'hello`.`apply_category`.`apply_time` AS `apply_time`
from (('hello`.`users` join `hello`.`pass_category`) join `hello`.`apply_category`) join `hello`.`activity_info`)
where (('hello`.`users`.`user_id` = `hello`.`apply_category`.`user_id`) and
('hello`.`apply_category`.`apply_id` = `hello`.`pass_category`.`apply_id`) and
('hello`.`apply_category`.`activity_id` = `hello`.`activity_info`.`activity_id`));
```

Below the SQL code, a table view of the data is shown. The table has four columns: 'id', 'activity_id', 'activity_name', and 'apply_time'. It contains five rows of data, representing different activities and their application times.

	id	activity_id	activity_name	apply_time
1	6	2	爱生活	2022-12-27 21:05:48
2	6	2	爱生活	2022-12-27 21:07:21
3	6	2	爱生活	2022-12-27 21:09:19
4	6	1	捉鬼大队行动	2022-12-27 22:22:43
5	8	4	圣诞派对	2022-12-28 18:31:51

2) 使用场景（用途）

该视图展示的是所有通过的活动的详细信息，能让人清楚直观的看见申请人的编号，通过活动的编号，通过活动的名字和其申请的时间。当想看某个人的通过活动记录时，就可以很直观地看见，另外想看整体时，也十分方便。

4、 触发器

1) 触发器截图



The screenshot shows a database management tool interface with a SQL editor. The code defines a trigger named 'users_makeapply_time' that is triggered before an insert operation on the 'apply_category' table. The trigger sets the 'apply_time' column of the new record to the current time using the 'NOW()' function.

```
create definer = root@localhost trigger users_makeapply_time
before insert
on apply_category
for each row
SET NEW.apply_time = NOW();
```

2) 使用场景（用途）

用户申请活动时，没必要填写自己申请的时间，系统会根据创建申请当下的时间利用触发器及时获取申请的时间

3) 验证触发器

```
hello> insert into apply_category (apply_id, user_id, activity_id, apply_time) VALUES (8,8,1, '')
[2022-12-29 19:28:27] 1 row affected in 12 ms
```

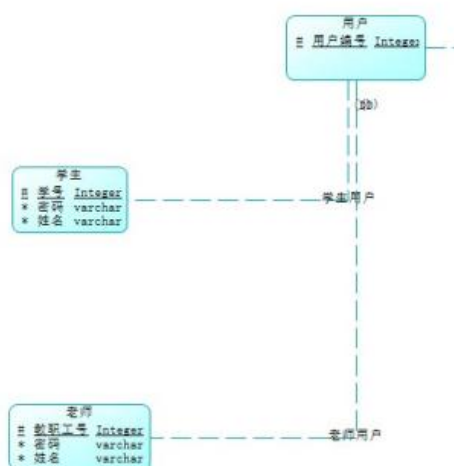
作用成功

表中出现了内容

8	8	8	1	2022-12-29 19:28:27
---	---	---	---	---------------------

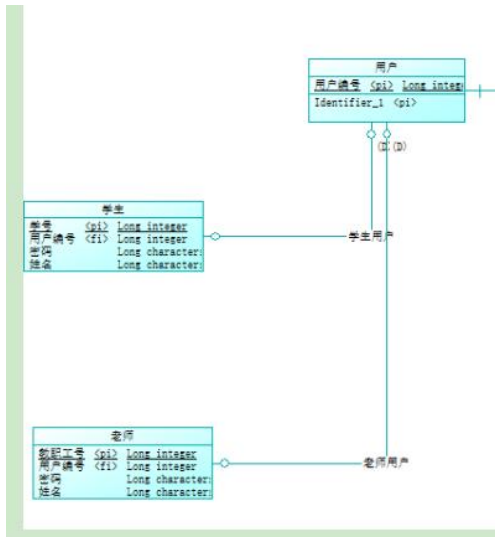
2.1.5 分析

选择较为有代表性的实体和联系进行分析，图文讲解如何从ER图到LDM图、再到PDM图，最后变成数据库表结构的转换过程（参考PPT里“补充内容ER图→LDM图→PDM图”中老师讲的例子）。



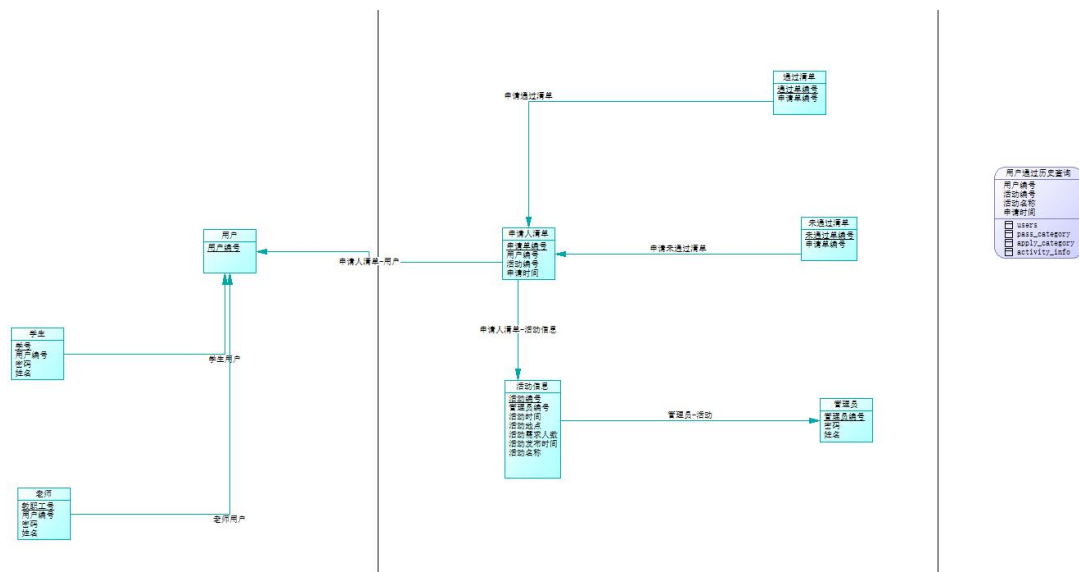
E-R图，只是粗略地从大概角度定义了实体之间的联系，比如一对一，一对多等，学生和用户，老师和用户都是一对一的关系，属性之间的具体关系并未定义，我们为每一个实体定义了一些属性，并设置了主键，非空的一系列定义

E-R图的基础上，我们生成了LDM图，软件根据我们设计的关系，和我们自己的调整，生成了联系几个实体的，具有关系的属性，使整个系统结构更加充实



由于我们的设置，用户里的属性“用户编号”在学生和老师中也进行了生成，补全了结构。

LDM 的基础上，我们生成了 PDM 图，具体展示了逻辑结构，将这个系统的结构化繁为简，着眼于逻辑上的定义，并补充了视图（紫色部分），具体定义了箭头和依赖关系，在最后的数据库系统生成中，这部分将作为外键存在



3 收获和反思

请填写本次实验的收获，记录实验过程中出现的值得反思的问题及你的思考。

生成 CDM,LDM,PDM 的过程中，一帆风顺，后来生成数据库的时候却发现出现了一些小问题，一开始是类型选错了，没选到 **mysql**，后来发现属性之间的联系有点问题，明明该作为外键的，最后定义反了，比如学生和用户之间，学生就是用户的一部分，学生具有的属性：用户编号，必须首先保证在用户中有这样的用户编号存在，这样的约束

这个实验也锻炼了我前后端交互的能力，比如查询语句，因为主体不同，我在写语句的时候就用变量代替了主语，主语不同，变量不同，最后对应的查询语句就不同，这样有一个麻烦的地方就是标点符号很难表达。最后克服成功。整个实验令我很有成就感。

```
sql=insert into hello.students (student_number, user_id, student_code, student_name) VALUES ("Id+", "+mx+", "+ "+Code+" "+" +"
```