



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

实验报告

开课学期: 2023 秋季

课程名称: 数据库系统

实验名称: 校园志愿者招募平台系统设计与实现

实验性质: 设计型

实验学时: 8 地点: T2608

学生班级: 2021 级 8 班

学生学号: 210110812

学生姓名: 李春阳

评阅教师:

报告成绩:

实验与创新实践教育中心制

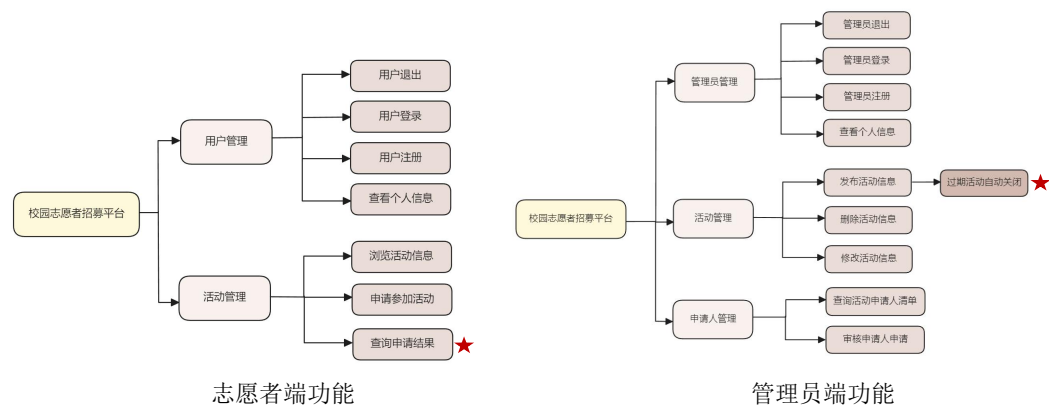
2023 年 10 月

1 实验环境

操作系统: Window 11
数据库设计: PowerDesigner
数据库管理系统: MySQL
前后端交互方案: 使用 Django 搭建 Web 应用
前后端交互编程语言: Python
前端开发所用的 IDE: PyCharm

2 实验过程

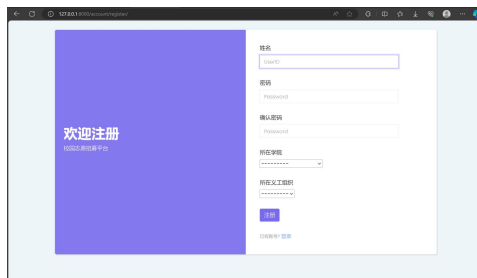
本实验在 4 条基础功能之外, 还完成了第 5 条功能“普通用户能看到自己提交的申请是待审核、通过还是拒绝状态”和第 6 条“过期的活动系统自动设置为关闭状态”(视频中无法演示, 详见报告第 3 页) 两个额外功能。志愿者端及管理员端功能设计图如下:



如果从整个系统的设计出发, 不难看出, 该校园志愿者招募平台的功能分为用户管理、活动管理和申请管理三大功能。

➤ 用户管理:
该平台有**志愿者**和**管理员**这两种类型的用户。使用者可根据自身情况完成这两种不同类型用户的**注册**、**登录**、**退出**操作, 并可在平台中**查询**到自己的姓名、编号、义工组织(仅志愿者包含这一信息)、学院等个人信息。





注册界面



个人信息查询

➤ 活动管理:

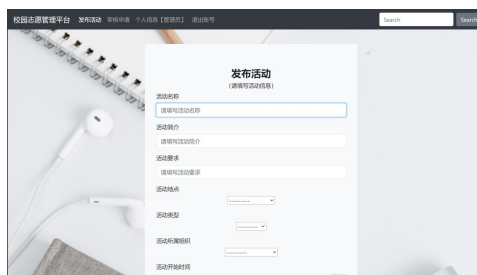
管理员可**发布新的志愿活动**（包括活动名称、时间、地点、人数、要求、介绍等活动信息），也可以**修改和删除已有的志愿活动**。此外，系统的活动展示界面会根据活动的开始时间升序进行展示，并会**自动关闭已经过期的活动**（仅志愿者端不再显示，管理员端仍可以对已过期活动进行编辑）。



志愿者活动列表展示界面



管理员修改及删除活动界面



管理员发布活动界面



由于“航天”活动过期，因此不再在活动列表中展示

➤ 申请管理:

志愿者可以**申请参加**正处于招募状态的活动。志愿者提交申请后，管理员可**查看自己所发布活动的申请人清单**，并对申请人进行**审核**（通过或拒绝）。此外，志愿者可以**查看自己所有历史申请的审核结果**（待审核、通过还是拒绝状态）。



活动申请界面



申请结果查询



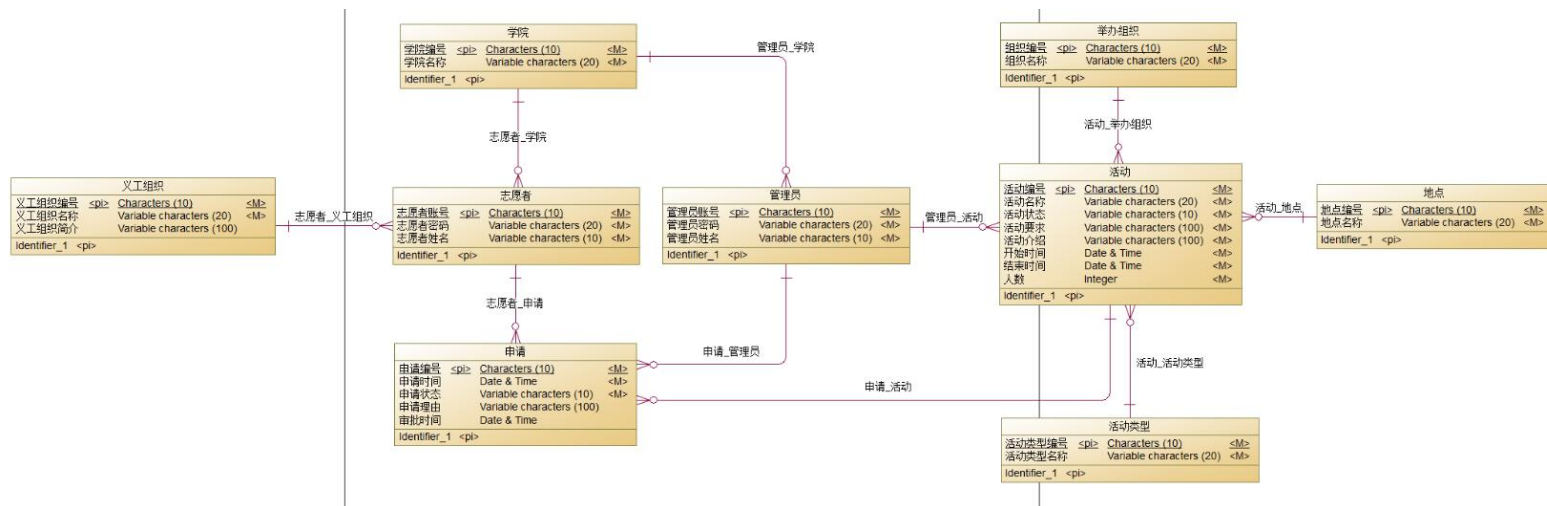
管理员查看自己的所有历史申请的审核结果

管理员活动审批界面

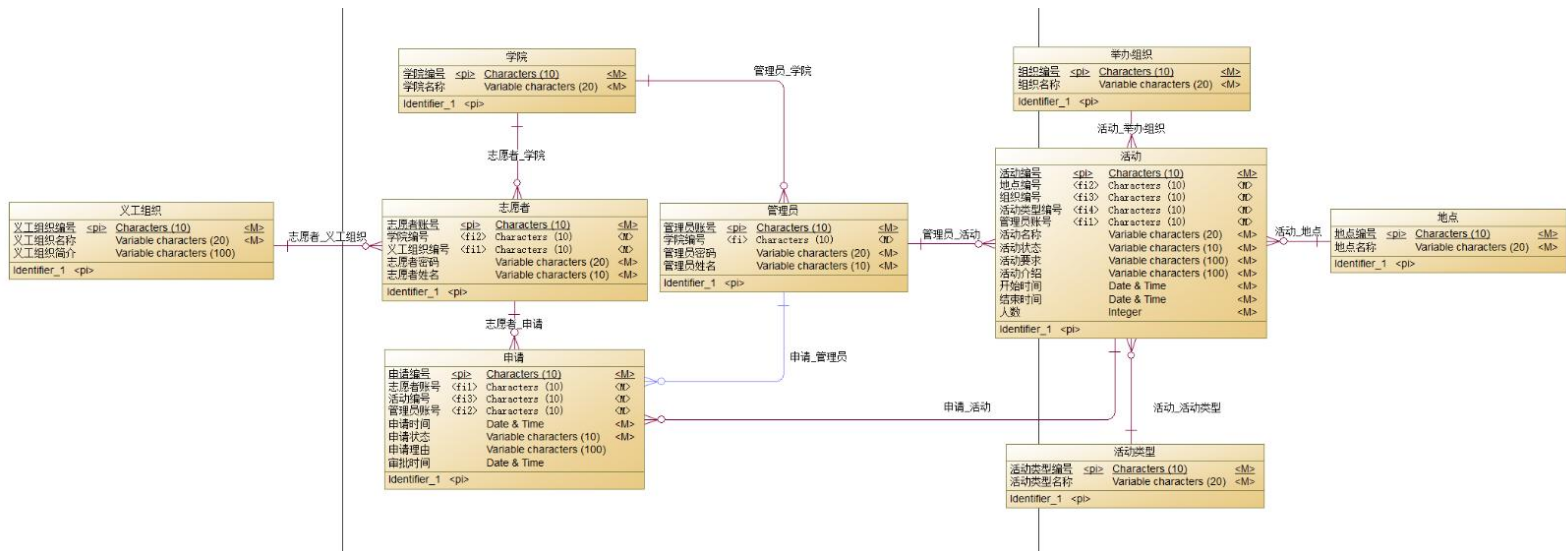
2.1 数据库设计

2.1.1 ER 图

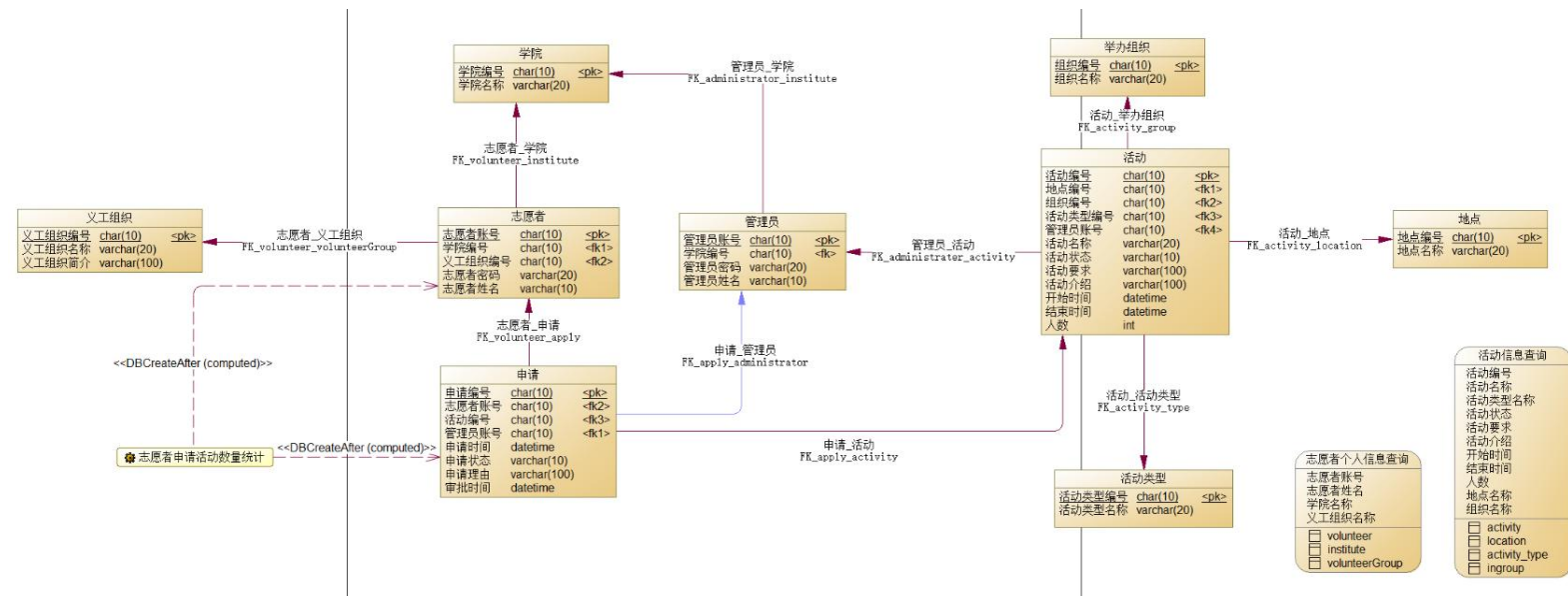
该 ER 图总共有地点、活动类型、活动、举办组织、申请、管理员、义工组织、志愿者、学院、管理员 9 个实体，并包含有 9 个联系。



2.1.2 LDM 图



2.1.3 PDM 图



2.1.4 数据库表结构

1、表结构

1) 志愿者表:

志愿者			
志愿者账号	<pi>	Characters (10)	<M>
志愿者密码		Variable characters (20)	<M>
志愿者姓名		Variable characters (10)	<M>
Identifier_1	<pi>		

ER 图

LDM 图

[illegible]

- 主键约束：志愿者表的主键为 `v_id`，即每个志愿者都会有一个唯一的编号。
- 外键约束：包括义工组织编号 `vg_id`、学院编号 `i_id`。因为义工组织名称、学院名称等信息被分别保存在义工组织表和学院表中，因此需要在志愿者表中使用义工组织表、学院表的主键作为外键。
- 空值约束：图中“NN”选项被勾选的属性均为非空属性（**not null**）。

2) 活动表:

活动			活动		
活动编号	<pi>	Characters (10)	活动编号	<pi>	Characters (10)
活动名称		Variable characters (20)	地点编号	<fi2>	Characters (10)
活动状态		Variable characters (10)	组织编号	<fi3>	Characters (10)
活动要求		Variable characters (100)	活动类型编号	<fi4>	Characters (10)
活动介绍		Variable characters (100)	管理员账号	<fi1>	Characters (10)
开始时间		Date & Time	活动名称		Variable characters (20)
结束时间		Date & Time	活动状态		Variable characters (10)
人数		Integer	活动要求		Variable characters (100)
Identifier_1	<pi>		活动介绍		Variable characters (100)
			开始时间		Date & Time
			结束时间		Date & Time
			人数		Integer
			Identifier_1	<pi>	

ER 图

LDM 图

Table Name: activity		Schema: volunteer	
Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci		Engine: InnoDB	
Comments:			
Column Name	Data type	PK	NN
av_id	CHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
l_id	CHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
g_id	CHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
t_id	CHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
a_id	CHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
av_title	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
av_state	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
av_request	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
av_content	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
av_starttime	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>
av_endtime	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>
av_number	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>

- 主键约束：活动表的主键为 av_id，即每个活动都会有一个唯一的编号。
- 外键约束：包括地点编号 l_id、活动类型编号 t_id、管理员编号 a_id、活动组织编号 g_id。因为地点名称、活动类型名称、管理员名称、活动组织名称等信息被分别保存在义工组织表和学院表中，因此需要在志愿者表中使用地点表、活动类型表、管理员表、活动组织表的主键作为外键。
- 空值约束：图中“NN”选项被勾选的属性均为非空属性（not null）。

2、索引

活动表中定义了 2 个索引 idx_start_time 和 idx_end_time（非主键、非外键索引），分别对应活动表中的活动开始时间 av_starttime 和活动结束时间 av_endtime 两个属性。

1) 索引截图

Table Properties - 活动 (activity)					
Physical Options		MySQL	Notes	Rules	Preview
General	Columns	Indexes	Keys	Triggers	Procedures
Name	Code	U	P	F	A
1 activity_PK	activity_PK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2 activity_location_FK	activity_location_FK	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3 activity_group_FK	activity_group_FK	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4 activity_type_FK	activity_type_FK	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5 administrater_activity	administrater_activity	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6 idx_start_time	idx_start_time	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7 idx_end_time	idx_end_time	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```

43  /*=====*/
44  /* Index: idx_start_time */
45  /*=====*/
46  • create index idx_start_time on activity
47  (
48      av_starttime
49  );
50
51  /*=====*/
52  /* Index: idx_end_time */
53  /*=====*/
54  • create index idx_end_time on activity
55  (
56      av_endtime
57  );

```

2) 使用场景（用途）

idx_start_time 和 idx_end_time 两个索引分别对应活动表中的活动开始时间 av_starttime 和活动结束时间 av_endtime。在按开始时间或结束时间升序查看志愿活动时，需要使用到 order by 对时间进行排序，而加入索引可以提升排序速度，进而提升查询速度。

3、 视图

1) 视图截图

志愿者视图：

志愿者个人信息查询

志愿者账号
 志愿者姓名
 学院名称
 义工组织名称

☐ volunteer
☐ institute
☐ volunteerGroup



活动视图：



2) 使用场景（用途）

志愿者视图：在该视图中，可以显示志愿者所有公开的个人信息（包括账号、姓名、所属学院、所属义工组织），但不包括密码等隐私信息。这在实际中是可能存在的。当需要查询志愿者的个人信息时，可直接查询该视图。

活动视图：在该视图中，可以显示活动的所有信息（包括活动编号、活动名称、活动类型、活动地点、活动状态、活动内容、活动要求、活动开始时间及结束时间、招募人数、举办组织等）。当需要查询活动相关信息时，可直接查询该视图。

4、 触发器

共定义了两个触发器，分别为自动设置申请时间和自动设置审批时间。

1) 触发器截图

Table Properties - 申请 (apply)

Physical Options		MySQL		Notes	Rules	Preview
General	Columns	Indexes	Keys	Triggers	Procedures	
Name	Code	C				
自动设置审批时间	approval_time	<input checked="" type="checkbox"/>				
2 自动设置申请时间	apply_time	<input checked="" type="checkbox"/>				
		<input type="checkbox"/>				
		<input type="checkbox"/>				
		<input type="checkbox"/>				
		<input type="checkbox"/>				

```

243 DELIMITER ;;
244 • create trigger apply_time
245     before insert on apply
246     for each row
247     begin
248         set new.ap_time = now();
249     end;;
250 DELIMITER ;;
251
252 DELIMITER ;;
253 • create trigger approval_time
254     before update on apply
255     for each row
256     begin
257         set new.ap_approvaltime = now();
258     end;;
259 DELIMITER ;;

```

2) 使用场景（用途）

- 自动设置申请时间触发器可以在志愿者添加申请记录时获取当前时间，并赋给 ap_time 属性，从而实现自动记录志愿者申请时间的功能。
- 自动设置审批时间触发器可以在管理员审批记录时获取当前时间，并赋给 ap_approvaltime 属性，从而实现自动记录管理员审批时间的功能。

3) 验证触发器

验证自动设置申请时间触发器：

Query 1 x apply volunteer_recruitment apply activity

```

1 • Insert Into apply(ap_id, ap_state, ap_reason, a_id, av_id, v_id)
2   values(123456789, 1, '无', 'A210110812', '01', 836691642)
3
4

```

Result Grid

ap_id	ap_time	ap_state	ap_reason	ap_approvaltime	a_id	av_id	v_id
01	2023-10-26 04:00:00.000000	1	无	2023-10-26 22:00:00.000000	210110806	01	210110812
123456789	2023-10-26 23:52:56.000000	1	无		A210110812	01	836691642
285593596	2023-10-25 13:44:59.496337	2	无		A210110812	01	836691642
304994700	2023-10-26 14:25:19.134999	1	无	2023-10-26 14:30:35.154156	210110806	122370227	210110812
490018862	2023-10-25 08:45:00.000000	1	无	2023-10-25 14:00:58.440216	210110806	02	210110812
577535585	2023-10-26 14:24:05.473144	2	无		582350494	154561600	210110812
689748104	2023-10-25 14:04:49.067027	1	无	2023-10-25 14:05:24.334783	210110806	564872888	210110812
781001747	2023-10-25 13:45:37.499279	2	无		A210110812	02	836691642

27 23:52:56 Insert into apply(ap_id, ap_state, ap_reason, a_id, av_id, v_id) values(123456789, 1, '无', 'A210110812', '01', 836691642... 1 row(s) affected

观察可知，当我们新增一条申请记录（即使用 Insert 语句插入）时，虽然没有指明 ap_time 的值，但这条新增记录的 ap_time 依旧被设置为了系统的当前时间，说明该触发器生效。

验证自动设置审批时间触发器：

Query 1 x apply volunteer_recruitment apply activity

```
1 • UPDATE apply
2 SET ap_state=2
3 WHERE ap_id='123456789' ;
```

Result Grid

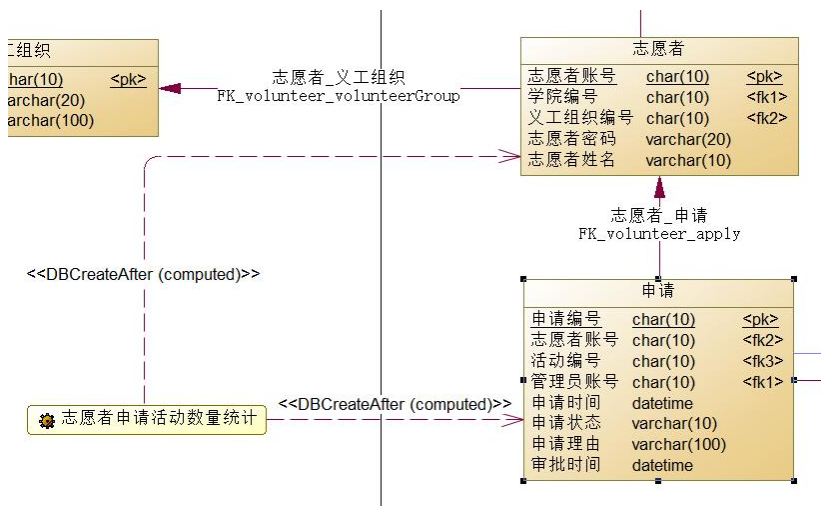
ap_id	ap_time	ap_state	ap_reason	ap_approvaltime	a_id	av_id	v_id
01	2023-10-26 04:00:00.000000	1	无	2023-10-26 22:00:00.000000	210110806	01	210110812
123456789	2023-10-26 23:59:41.000000	2	无	2023-10-27 00:02:37.000000	A210110812	01	836691642
285593596	2023-10-25 13:44:59.496337	2	无	NULL	A210110812	01	836691642
304994700	2023-10-26 14:25:19.134999	1	无	2023-10-26 14:30:35.154156	210110806	122370227	210110812
490018862	2023-10-25 08:45:00.000000	1	无	2023-10-25 14:00:58.440216	210110806	02	210110812
577535585	2023-10-26 14:24:05.473144	2	无	NULL	582350494	154561600	210110812
689748104	2023-10-25 14:04:49.067027	1	无	2023-10-25 14:05:24.334783	210110806	564872888	210110812
781001747	2023-10-25 13:45:37.499279	2	无	NULL	A210110812	02	836691642
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

36 00:02:37 UPDATE apply SET ap_state=2 WHERE ap_id='123456789' 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

观察可知，当我们新增一条审批记录（即使用 Update 语句更改 ap_state）时，原先 ap_approvaltime 的值为空的申请记录的 ap_approvaltime 被设置为了系统的当前时间，说明该触发器生效。

5、 存储过程或存储函数

1) 存储过程或存储函数截图



```

226 DELIMITER $$
227 • create procedure activities_number (IN p_volunteer_id char(10), OUT p_total_apply INT)
228 BEGIN
229     SELECT apply.ap_id
230     FROM apply, volunteer
231     WHERE volunteer.v_id = p_volunteer_id
232     AND volunteer.v_id = apply.v_id;
233
234     SELECT COUNT(apply.ap_id)
235     FROM apply, volunteer
236     WHERE volunteer.v_id = p_volunteer_id
237     AND volunteer.v_id = apply.v_id
238     group by volunteer.v_id
239     INTO p_total_apply;
240 END$$
241 DELIMITER ;

```

2) 使用场景（用途）

该存储过程用于统计某个志愿者申请的活动数量，同时返回其申请的活动列表及总共数量。当我们针对某一个志愿者进行查询时，可以更快速、更便捷的得到其申请的活动信息。

3 收获和反思

本次实验让我充分体验到一个小系统设计开发的过程，从中我增强了对数据库的理解、对 python 的掌握及前端开发相关技术的运用。最初是想使用 java Swing 进行开发，但由于 java Swing 的界面实在过于朴素(丑陋)，因此选择了使用 Django 搭建 Web 应用。Django 框架整体的后台管理非常便捷，因此在数据库管理方面并没有产生过多的问题。同时，我使用 Bootstrap 和网上其他开源的模板美化界面，也进一步提高了效率。在此感谢何为学长，其博客给予了我很大帮助。

ER 图的设计是本次实验的一个“地基”。在设计 ER 图时，需要不断考虑是否已经覆盖了全部数据需求和数据处理需求，避免出现需求遗漏的情况。而我就因为考虑不周，出现了推翻重改的情况。

总而言之，虽然过程中有痛苦、有迷茫，但计算机作为一门实践学科，亲身体验网页开发及前后端交互的过程都令我受益匪浅。最后，祝愿本课程越办越好！