

（深圳）

实验报告

开课学期： 2022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： 6 地点： T2210

学生班级： 20级5班

学生学号： 200110503

学生姓名： 孙铎

评阅教师：

报告成绩：

实验与创新实践教育中心制

2022年11月

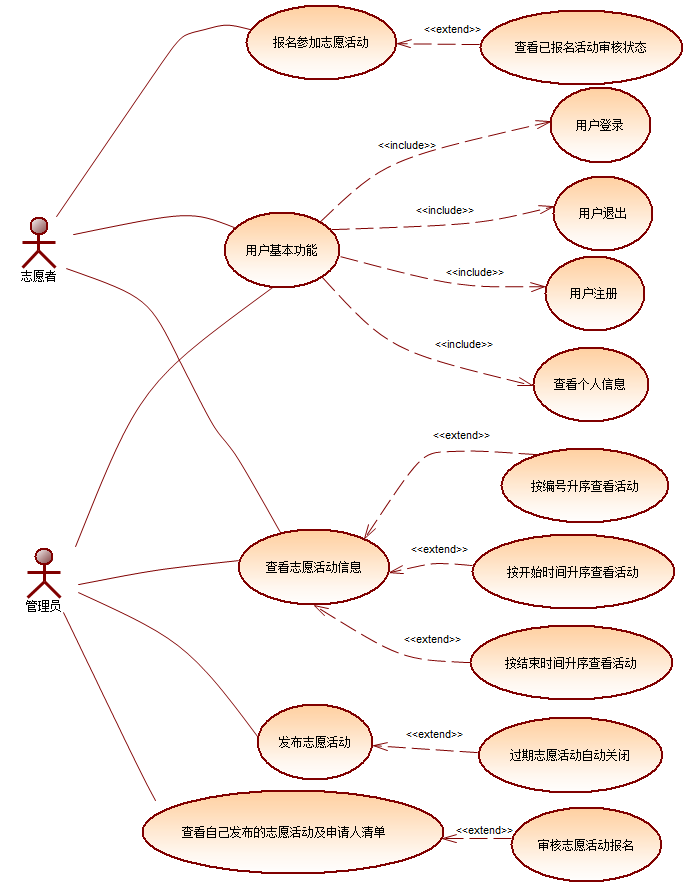
# 实验环境

*请填写用到的操作系统和主要开发工具。*

* 操作系统：Windows操作系统
* 开发语言：
  + 后端：Java
  + 前端：HTML、CSS、Javascript
* 环境与依赖：
  + JDK 8
  + MySQL JDBC: mysql-connector-j-8.0.31.jar
  + Tomcat 10.0.14
  + Jakarta Servlet: servlet-api.jar
  + MySQL 8.0
  + Jackson: jackson-core-2.13.0.jar, jackson-databind-2.13.0.jar, jackson-annotations-2.13.0.jar
  + Chrome浏览器（浏览器需要支持cookie）
* IDE：
  + PowerDesigner
  + IDEA

# 实验过程

## 系统功能

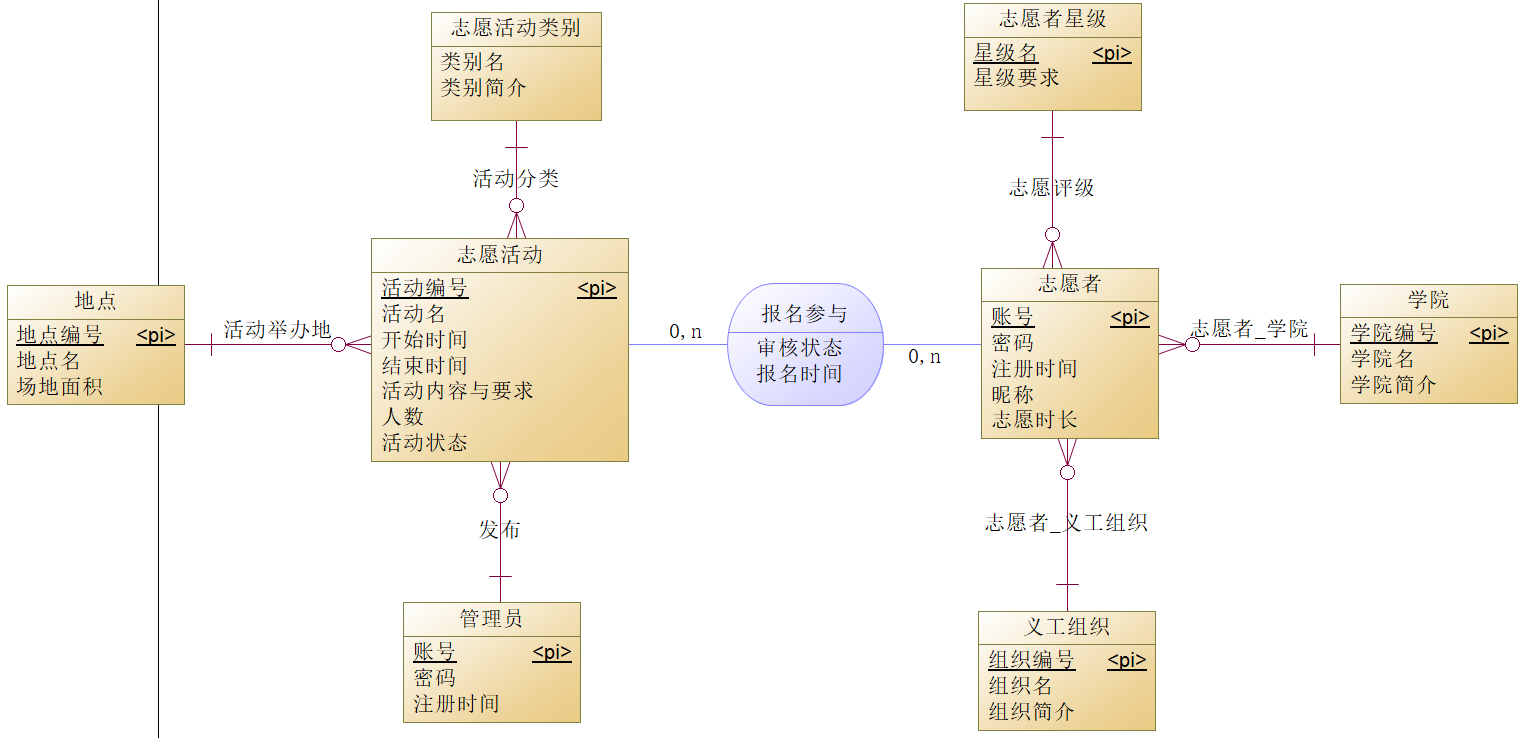
*请结合文字、图表等方式清晰描述系统的功能。如有亮点功能请用\*标志。*

* **用户基本**功能：
  + 用户注册：**普通用户/志愿者**注册、**管理员**注册。
  + 用户登录。
  + 用户退出。
  + 查看个人信息：**志愿者**可以查看账号、昵称、所属组织、志愿时长、志愿星级等信息。**管理员**可以查看账号、注册时间的信息。
* **志愿活动管理**功能**：**
  + **发布活动**：**管理员**可以发布志愿活动信息，包括活动时间、活动地点、所需人数、具体要求、志愿时长等信息。
  + **\*查看活动信息**：**志愿者和管理员**可以查看已经发布的志愿活动详细信息，并可以**按活动编号、开始时间、结束时间三种方式升序**查看。
  + **关闭志愿活动**：过期的志愿活动会自动设置为关闭状态。
* **志愿活动报名审核**功能：
  + **申请参加活动**：**志愿者**可以申请参加志愿活动。
  + **查看活动申请列表**：**管理员**可查看自己发布的志愿活动申请人清单。
  + **审核活动报名**：**管理员**可以审核报名参加志愿活动的申请人。
  + **查看报名状态**：**志愿者**可以查看自己的活动报名是待审核、通过还是拒绝。
  + **\*志愿者星级：志愿者**参加志愿活动累计到达一定时长时，会分别被评级为**一星级志愿者、二星级志愿者和三星级志愿者**。

## 数据库设计

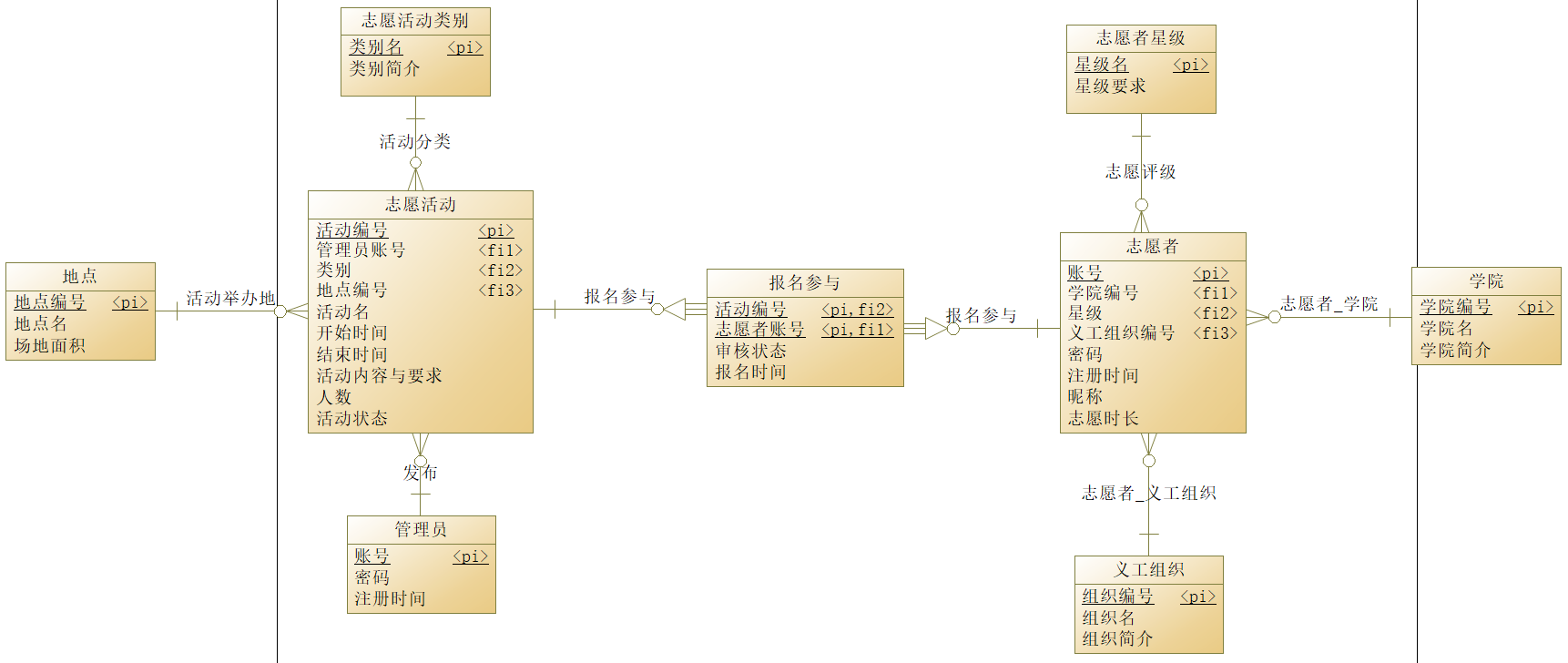
### ER图

*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。*



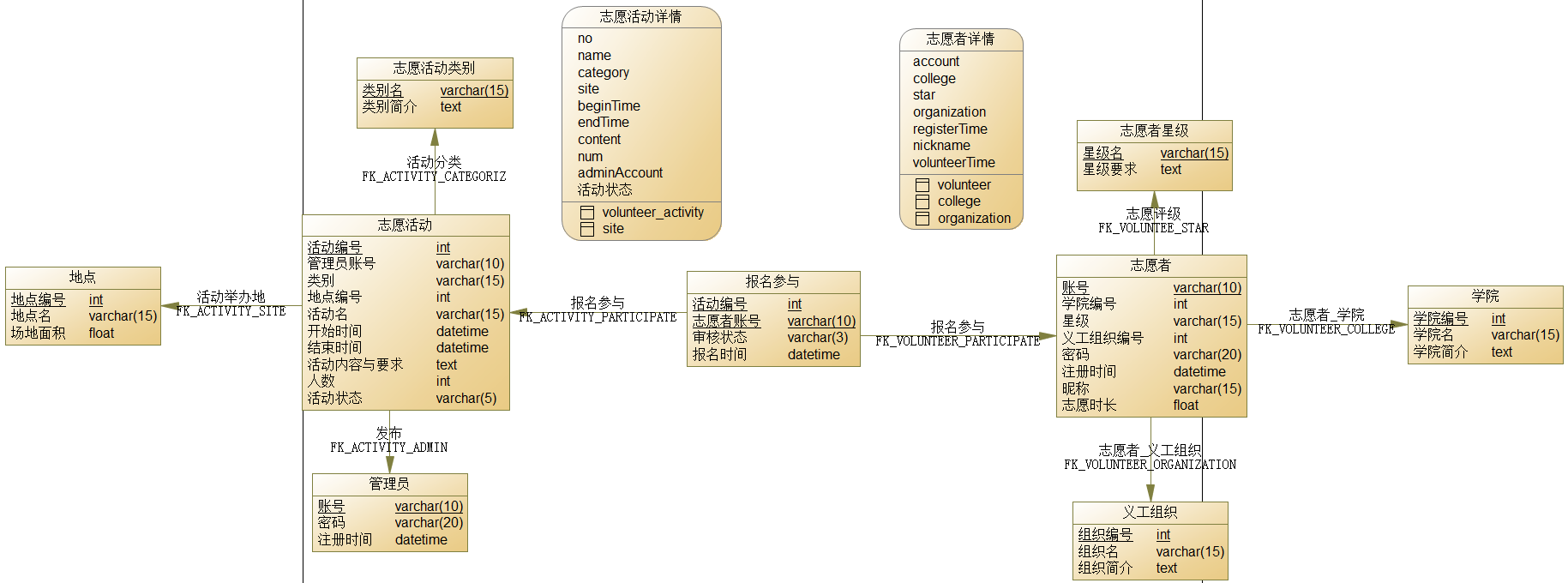
### LDM图

*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。*



### 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。*



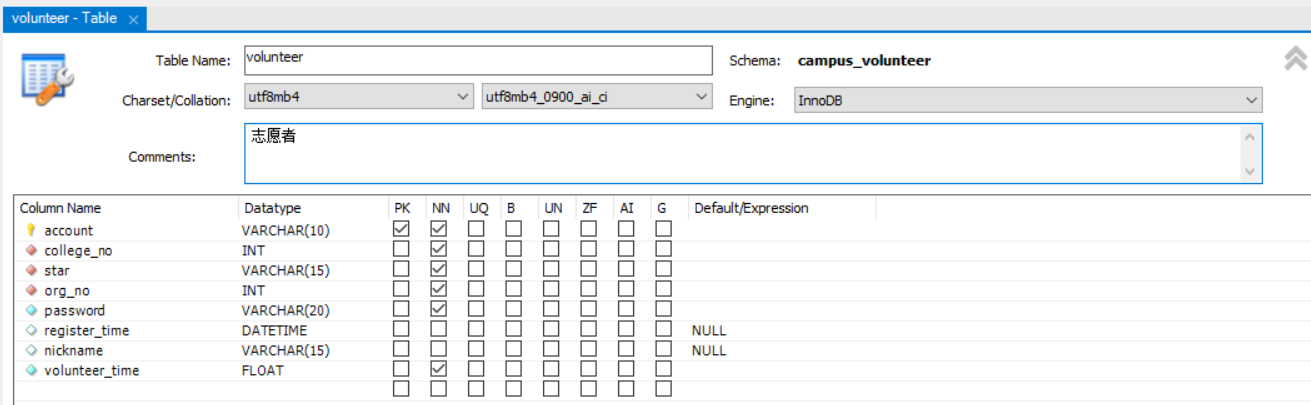
### 数据库表结构

1. **表结构**

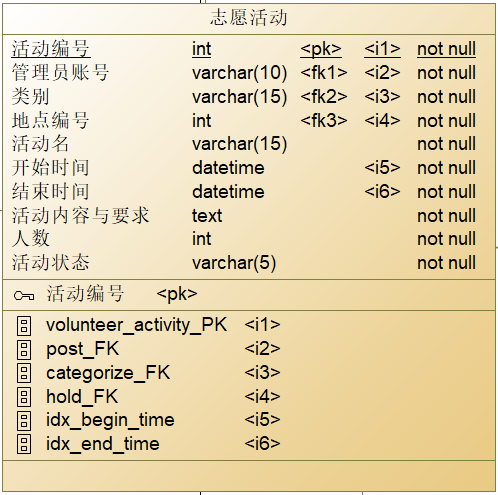
*选取2-3个比较有代表性的表结构截图，体现主键约束、外键约束、空值约束等。*

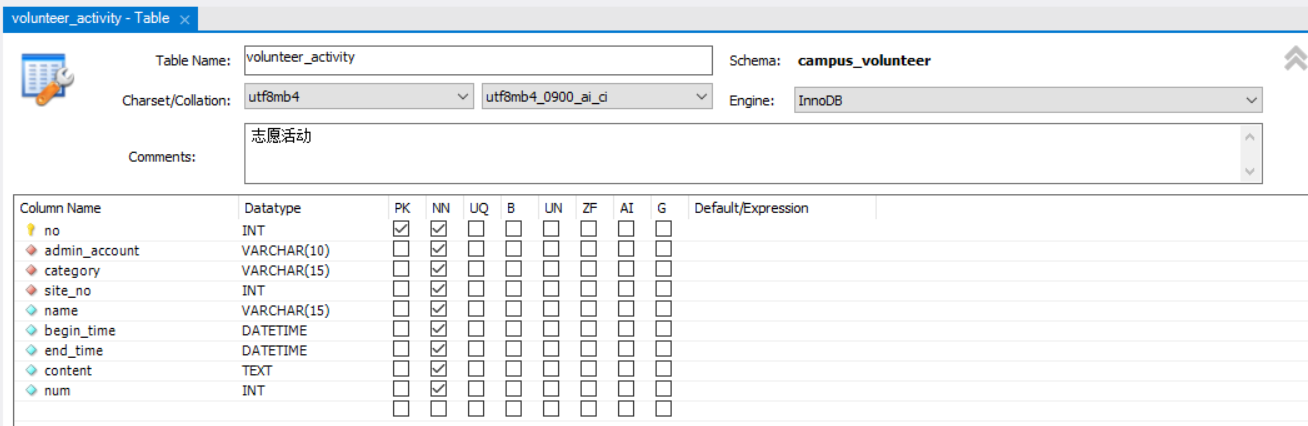
志愿者表：





志愿活动表：



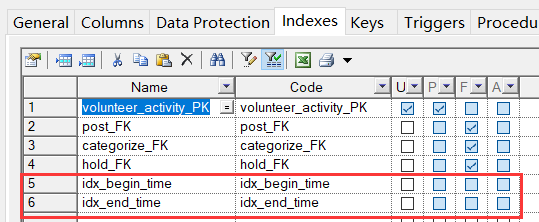


1. **索引**
2. 索引截图

志愿活动表，**开始时间**和**结束时间**两个字段上的两个**普通索引**：

手机屏幕截图

描述已自动生成



（不再说明其他主键索引和外键索引）

1. 使用场景（用途）

按开始时间或结束时间升序查看志愿活动时，需要使用到 ordey by 进行排序。在开始时间和结束时间两个字段上添加普通索引，可以加快排序操作，提升查询速度。

1. **视图**
2. 视图截图

志愿者详情视图：



志愿活动详情视图：

手机屏幕截图

描述已自动生成

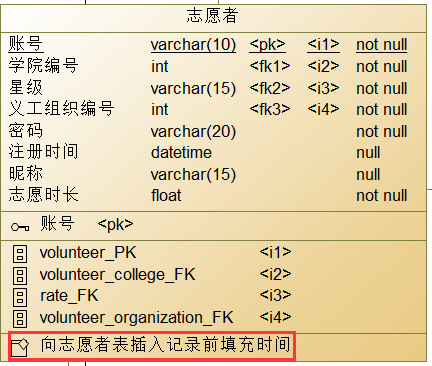
1. 使用场景（用途）

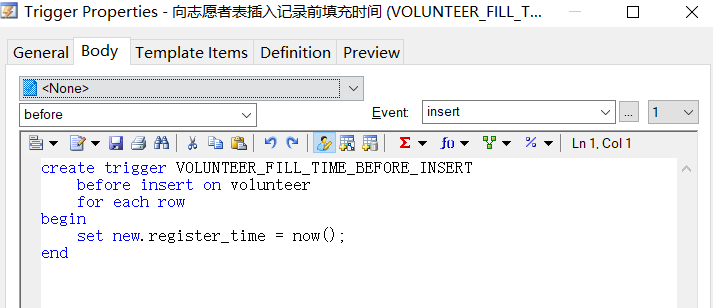
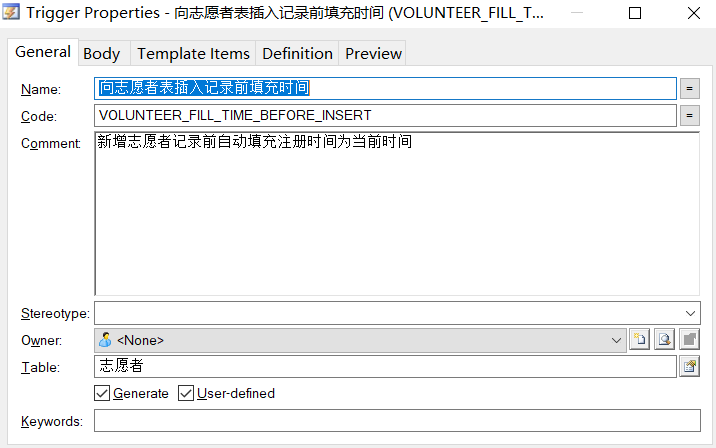
志愿者详情视图：展示志愿者包括所在**学院名、组织名**在内的详细信息，但不需要展示**密码**等隐私信息时，可以直接查询此视图。

志愿活动详情视图：可以展示包括所在**活动地点名**在内的志愿活动详细信息，并可以实时计算志愿活动是否过期关闭（当前时间是否大于志愿活动的开始时间），封装成视图，方便直接查询使用。

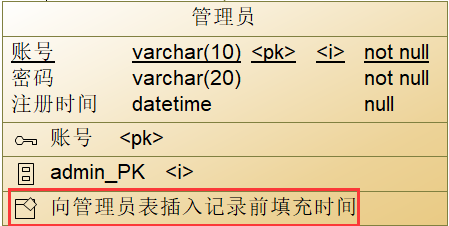
1. **触发器**
2. 触发器截图

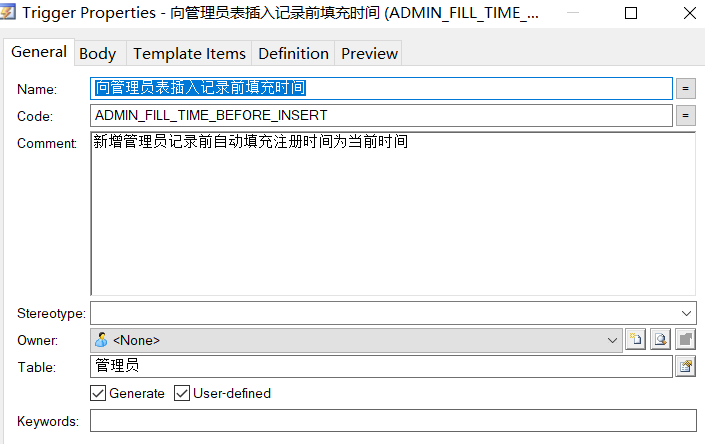
志愿者表填充注册时间的触发器：

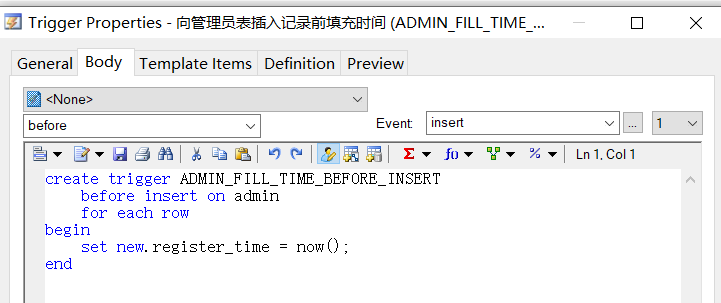




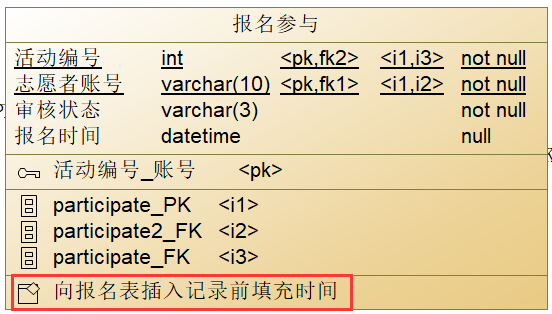
管理员表填充注册时间的触发器：

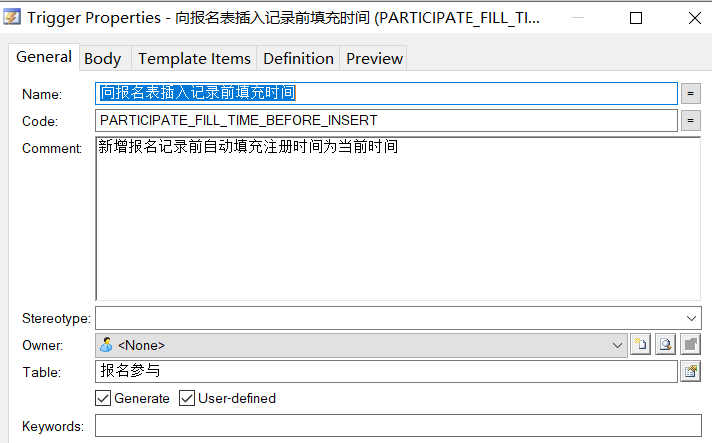


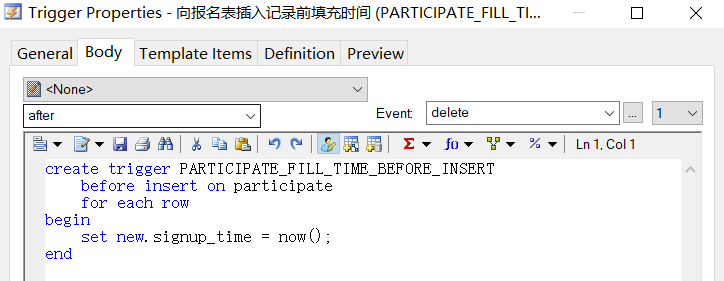




报名参与表填充报名时间的触发器：







1. 使用场景（用途）

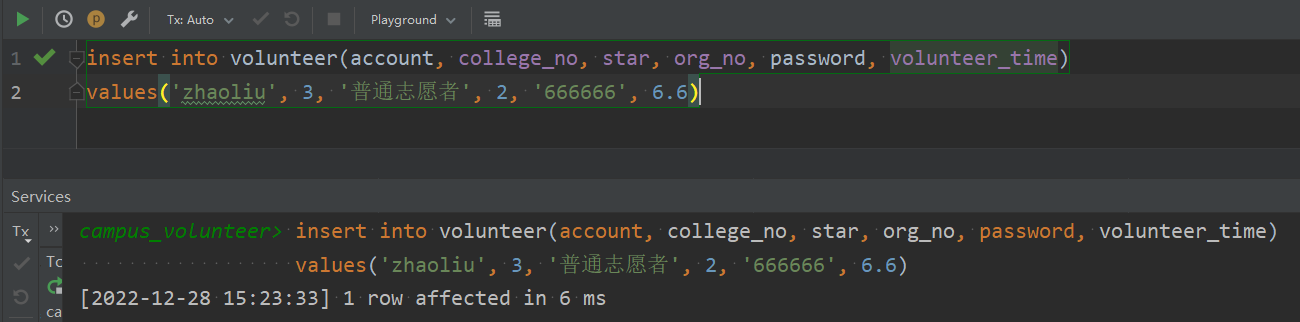
志愿者和管理员在注册时，会向志愿者表和管理员表插入一条新记录，触发器可以在插入新记录前获取当前时间，设置为新记录的注册时间，从而实现自动填充注册时间。

志愿者在报名时，会向报名参与表插入一条新记录，触发器可以在插入新记录前获取当前时间，设置为新记录的报名时间，从而实现自动填充报名时间。

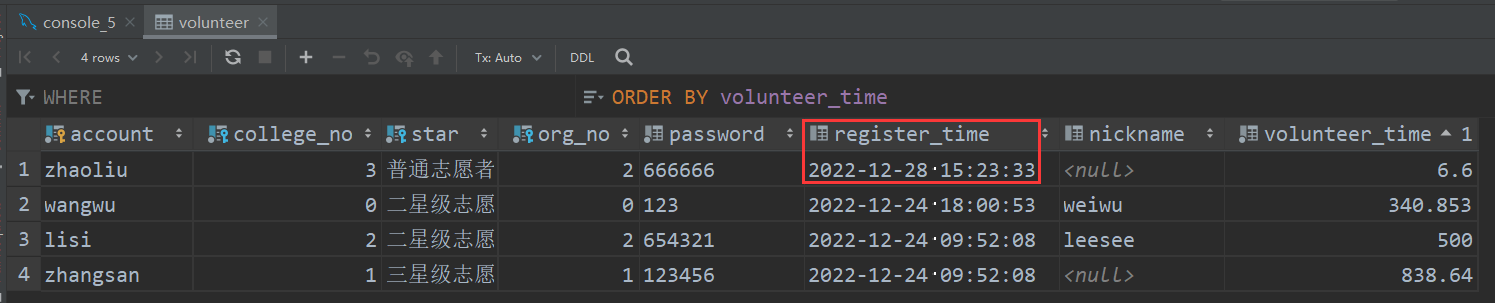
1. 验证触发器

验证志愿者表填充注册时间的触发器：

1. 插入记录，但不指定注册时间：

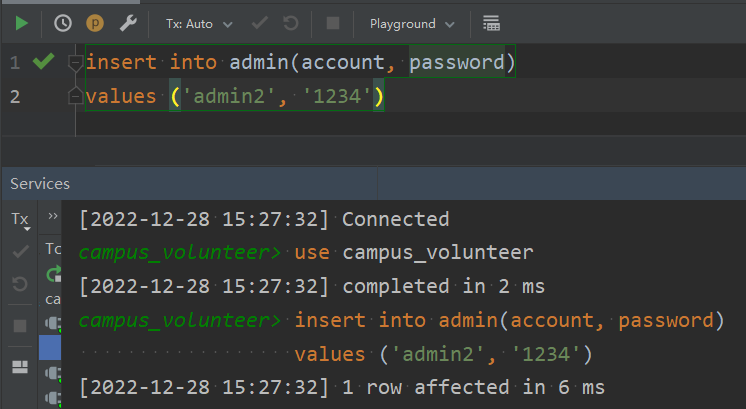


1. 查看表，新增记录自动填充了当前时间：

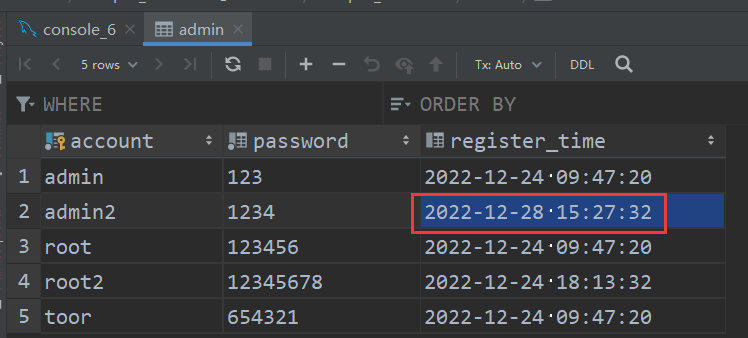


验证管理员表填充注册时间的触发器：

1. 插入记录，但不指定注册时间：

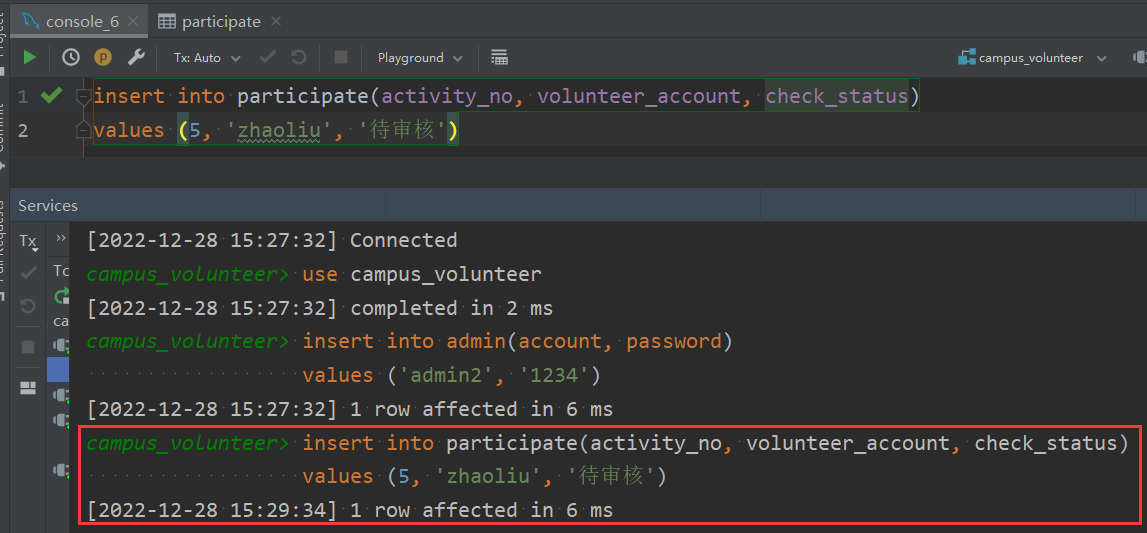


1. 查看表，新增记录自动填充了当前时间：

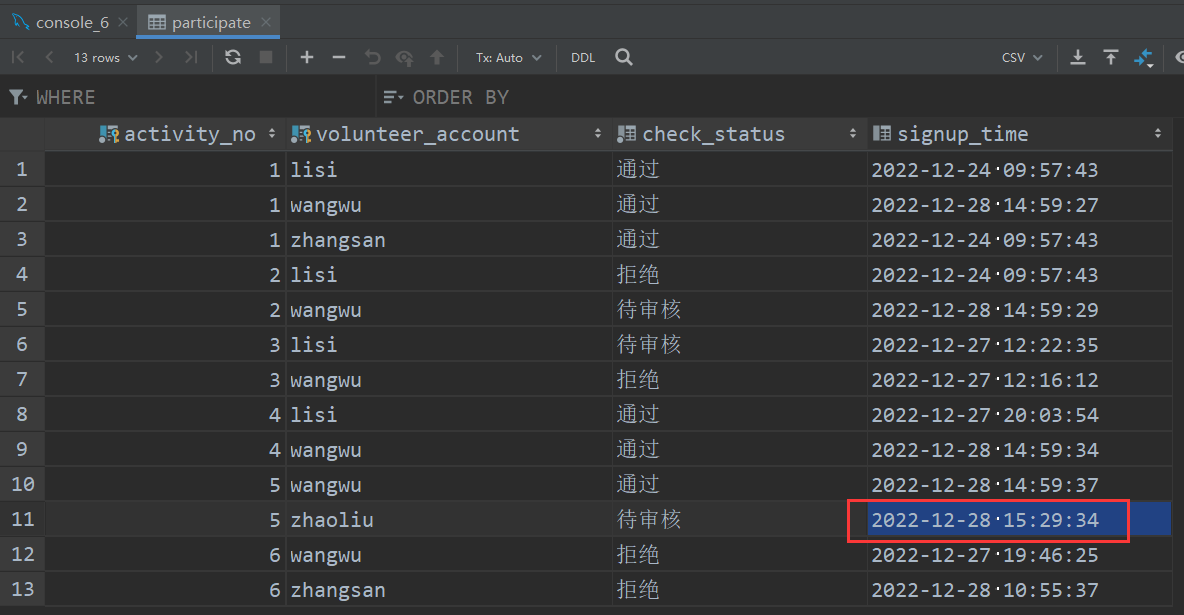


验证报名参与表填充报名时间的触发器：

1. 插入记录，但不指定报名时间：



1. 查看表，新增记录自动填充了当前时间：

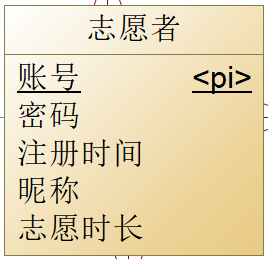


### 分析

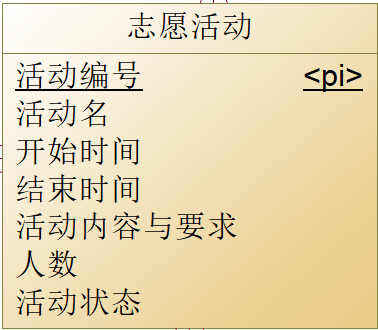
*选择较为有代表性的实体和联系进行分析，图文讲解如何从ER图到LDM图、再到PDM图，最后变成数据库表结构的转换过程（参考PPT里“补充内容 ER图🡪LDM图🡪PDM图”中老师讲的例子）。*

ER图实体：

* 普通用户/志愿者：

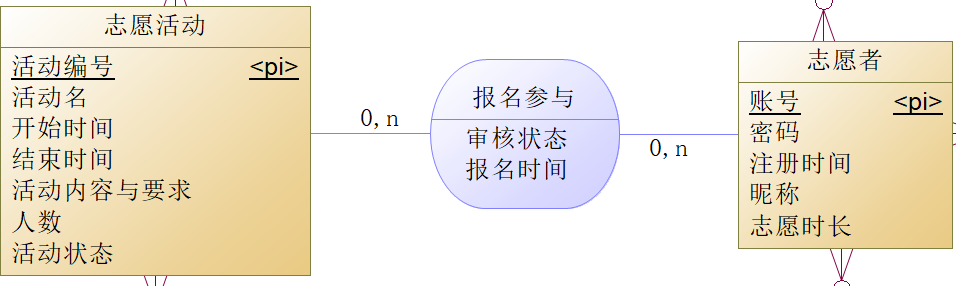


* 志愿活动：



ER图联系：

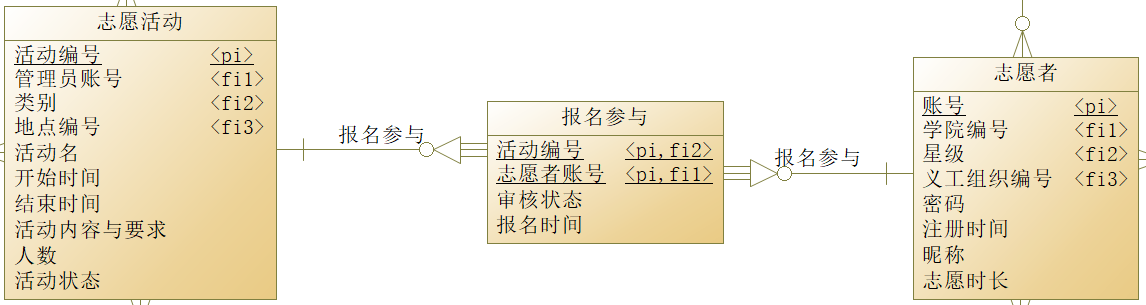
* 志愿者报名参与志愿活动：



可以看出，这是一个**多对多**的联系，且双方均为**部分参与**。这个联系本身还具有两个属性，分别是报名时间和管理员的此次报名的审核情况。

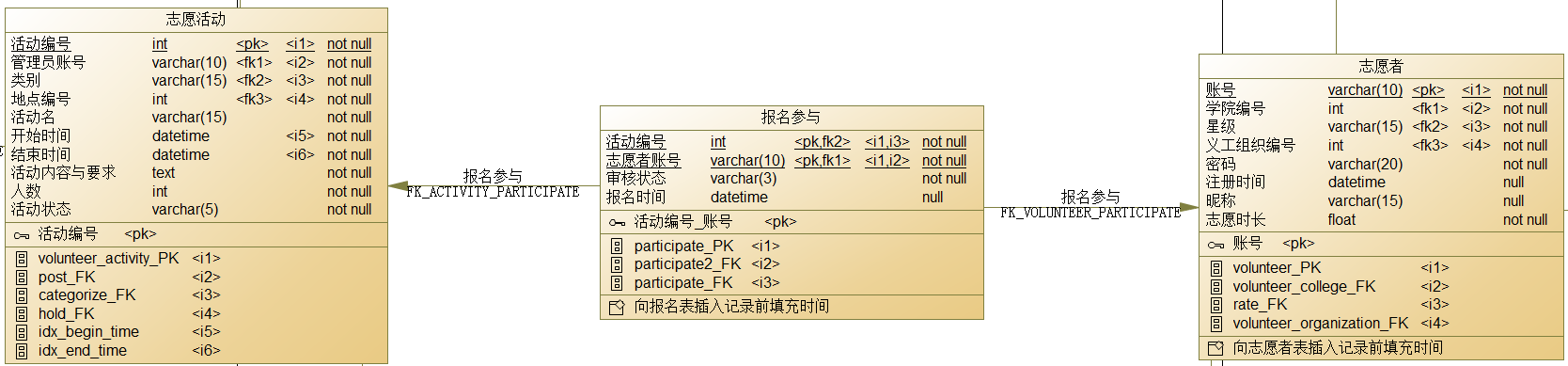
（注：更严格来讲，管理员也参与了“报名参与”这个联系，因为活动相关的管理员需要去审核此次报名。如果也画在ER图中，就出现了一个三元联系，会变得复杂。而实际开发中，根据志愿活动的编号就可以确定需要审核其报名的管理员，从而实现审核功能，而不需要让管理员也直接参与到该联系中，所以为简化设计，这里只使用志愿者和志愿活动的二元联系）

从ER图到LDM图：



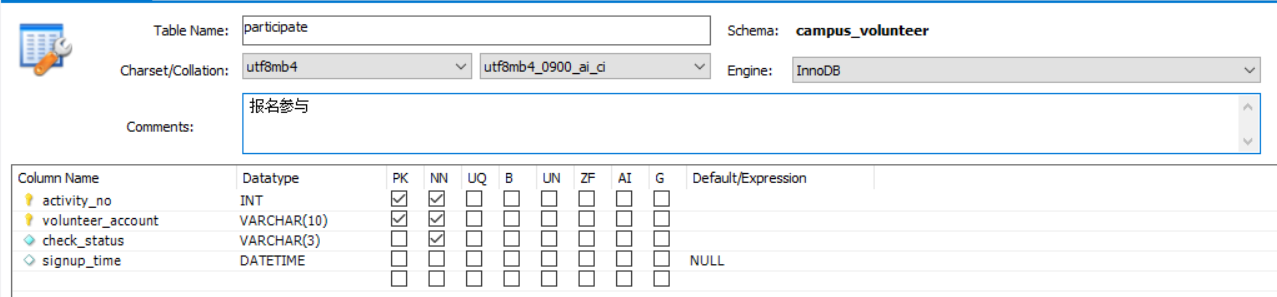
“报名参与活动”这个行为需要确定报名的志愿者和要报名的活动，可以通过这两个实体的主键加以确定，因此除了联系本身的属性，LDM图中还额外增加了联系双方实体外键作为联合主键，以确定一条报名记录。其他实体的主键和普通属性保持相同。

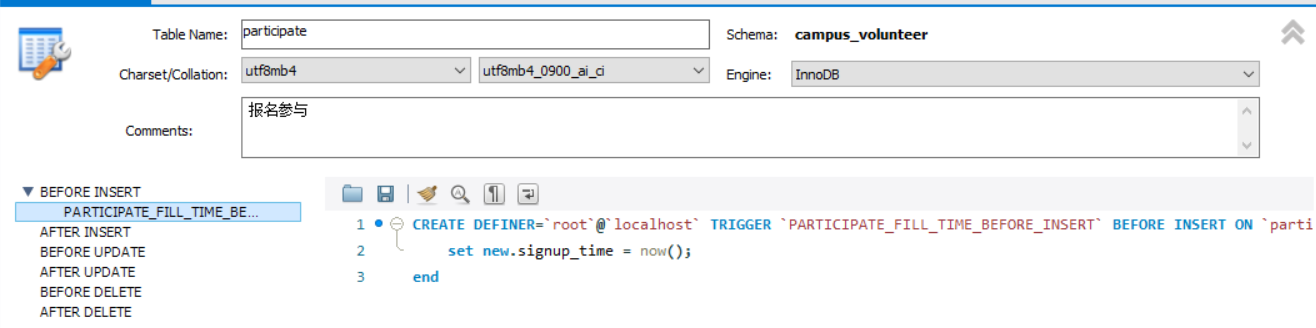
（注：志愿活动和志愿者表新增的外键来自于其他实体，在本例中没有体现）

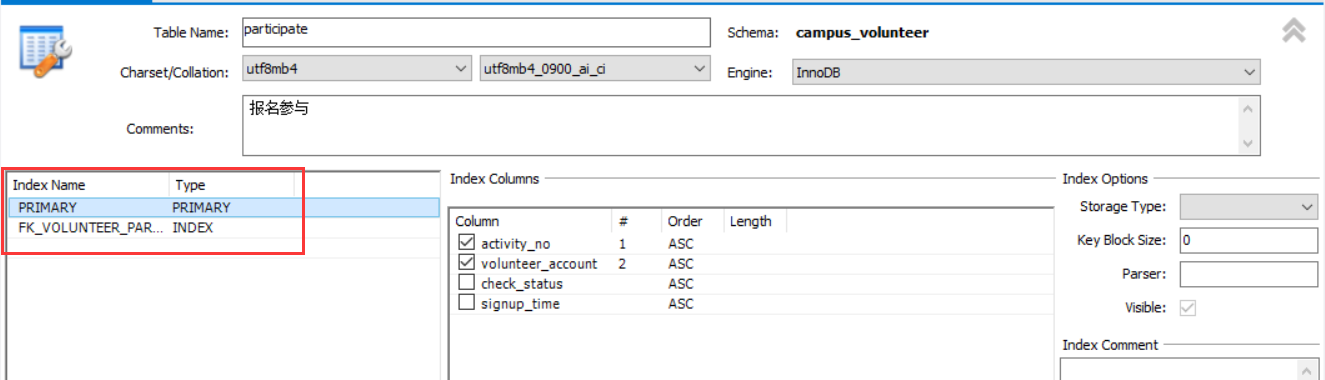
 从LDM图到PDM图：

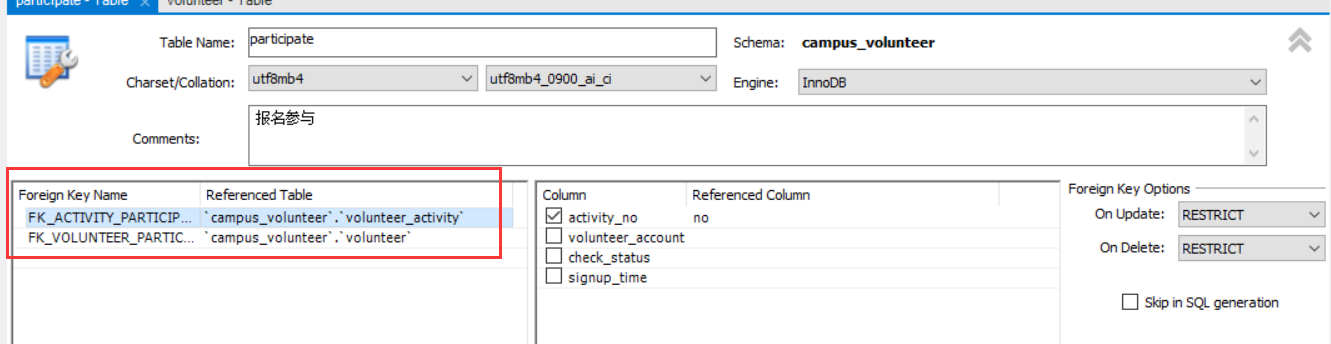
PDM图与LDM图的总体框架结构差别不大，PDM图更接近物理数据库表，明确了联合主键、数据类型、约束，建立了主键和外键索引，通过外键约束保证报名参与的志愿者和活动编号都是在志愿者表和志愿活动表中真实存在的。同时还建立了触发器，在新增报名记录时自动填充当前时间作为报名时间。

从PDM图到数据库表结构：









数据库表结构与PDM图上的细节几乎是一一对应的，使用MySQL实现PDM图，根据PowerDesign生成的SQL脚本（进行了一定修改），创建数据表，采用InnoDB存储引擎，将数据类型、约束、索引、触发器等要素以MySQL支持的方式创建了出来。

# 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。

* 对需求分析和E-R图建模的理解：从开发者的视角看待用户需求，应当注重从需求中**提取必要的实体**，并发掘**实体之间的必要联系**。设计好E-R图之后，还需要反思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。
* 对数据库设计过程的理解：数据库的设计不是一蹴而就的，而是从现实世界出发，经过**一步步抽象**得来的。最初进行顶层设计时，一定要做到认真细致，不能遗漏细节，如果顶层设计中出现了错误，那么接下来所有基于该设计的工作都需要做出修改，可能会很麻烦，所以最好保证**在最初设计时就做到简约、不繁琐、不易出错**。
* 对软件开发的理解：从后端的角度讲，开发功能的本质就是操作数据库，数据库设计得好，开发功能时就会变得顺畅。当进行前后端数据交互的开发时，一个较为便捷的思路就是，**从前端用户出发**，用户能进行的某个操作其实就对应着一个功能，思考该操作涉及到前端哪些组件，会触发什么逻辑，**然后从前端向后端思考**，前端应当向后端请求什么数据，后端应该如何操纵数据库，或进行其他操作，并思考如何将该操作的结果反馈给用户，后端应当向前端响应什么数据，前端应当如何向用户展示。当这样一条流程打通时，一个功能就基本上开发成功了。