

（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： 校园猫猫管理系统设计与实现

实验性质： 设计型

实验学时： 6 地点：

学生班级： 计算机6班

学生学号： 200110619

学生姓名： 梁鑫嵘

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：Arch Linux + Windows 11
* 开发工具：JetBrains IntelliJ CLion
* 开发语言：Rust、SQL

# 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。*

系统架构特点和系统功能

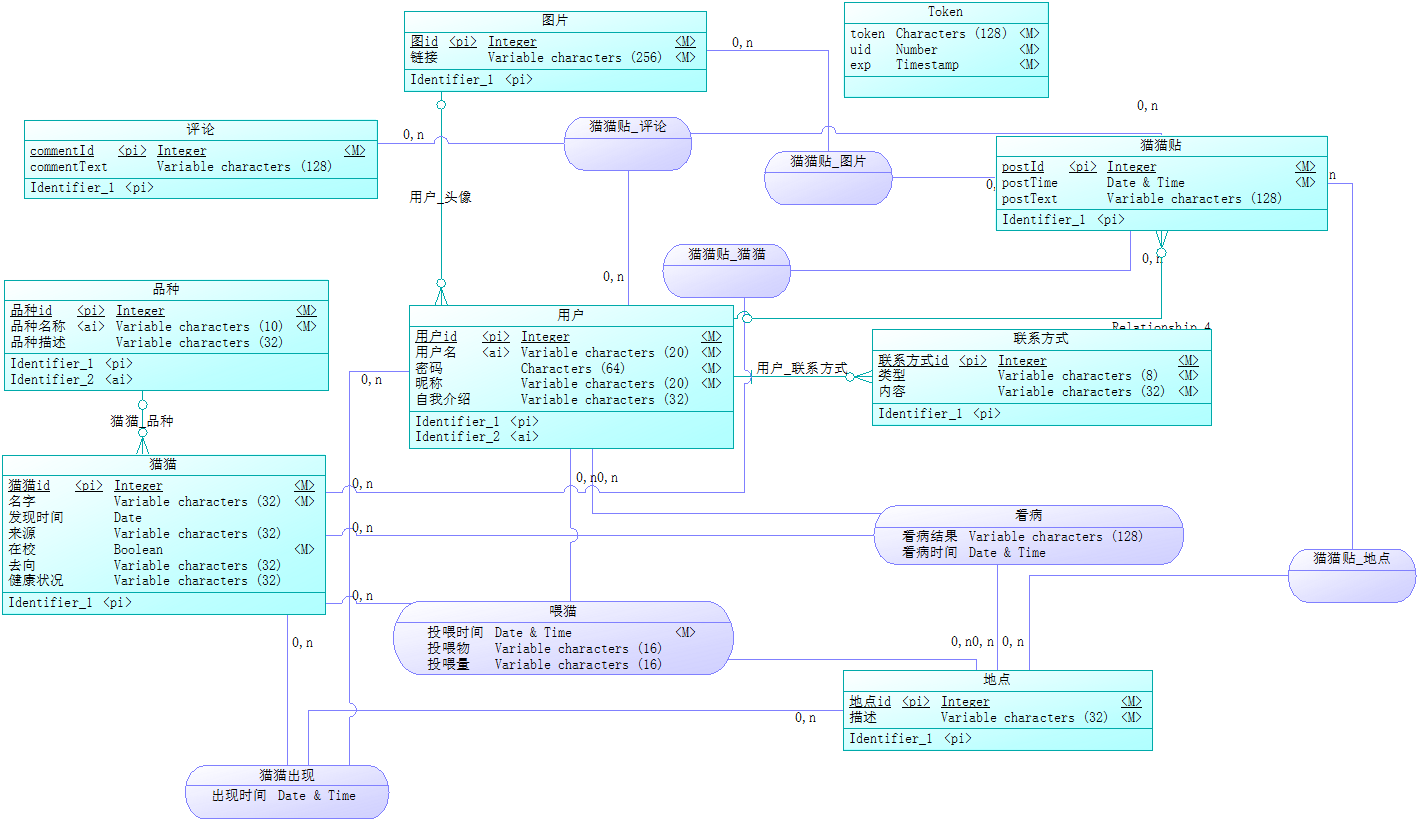
* 前后端分离架构
  + 系统使用基于网页的前后端分离架构
  + \* 系统API设计基于Restful风格
  + \* 系统使用统一的序列化、反序列化API
* 前端
  + \* 前端使用纯Rust语言构建
  + \* 首先将Rust编译到Web Assembly平台运行
  + \* 再通过中间组件交互HTML DOM和发起fetch请求
* 后端
  + \* 后端基于tokio和wrap，基于Actor的高性能调度方式
  + 数据库使用MySQL
* 用户系统
  + \* 使用JWT认证系统
    - \* 前端持有两个Token，短Token时常刷新，长Token用于保持登录状态
    - \* 后端通过Token辨识用户信息
  + 用户可以注册、登录
  + \* 用户可以编辑头像、自我介绍等信息
* 猫猫登记
  + 添加猫猫品种
  + 登记新的猫猫，记录名字、品种、发现时间、来源等
* \* 猫猫贴社区
  + \* 用户可以在社区中发表猫猫贴，包含文字、时间、地点、猫猫、图片等信息
  + \* 同时可以标记多个位置
  + \* 同时可以关联多个猫猫
  + \* 同时可以在帖子中附上多个图片
  + \* 猫猫贴可以被多人回复评论
* 投喂记录
  + 可以登记某一次投喂的猫猫
  + 以及对应的时间、食物、投喂量、地点
  + \* 可以检索最近的投喂记录，了解哪些猫猫还未被投喂
* 出现地点统计
  + 统计猫猫最近出现地点
* \* 就医统计
  + 登记就医情况
  + 更新关联猫猫的健康状态

## 数据库设计

### ER图

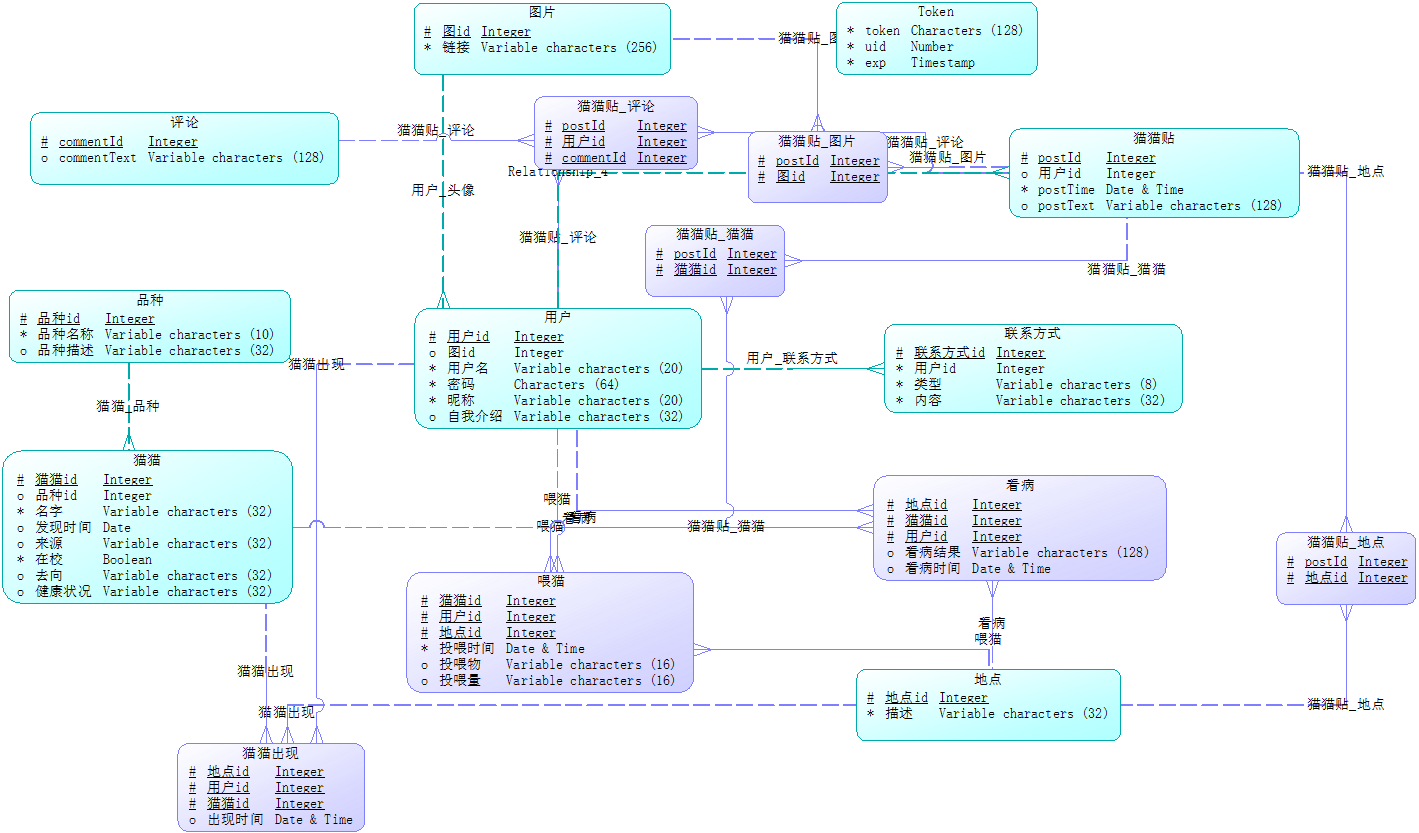
*要求：截图务必清晰，如果图太大可截图一个总图，然后再分块截图。如果看不清截图会影响成绩。*

*\* 请放大查看图片*



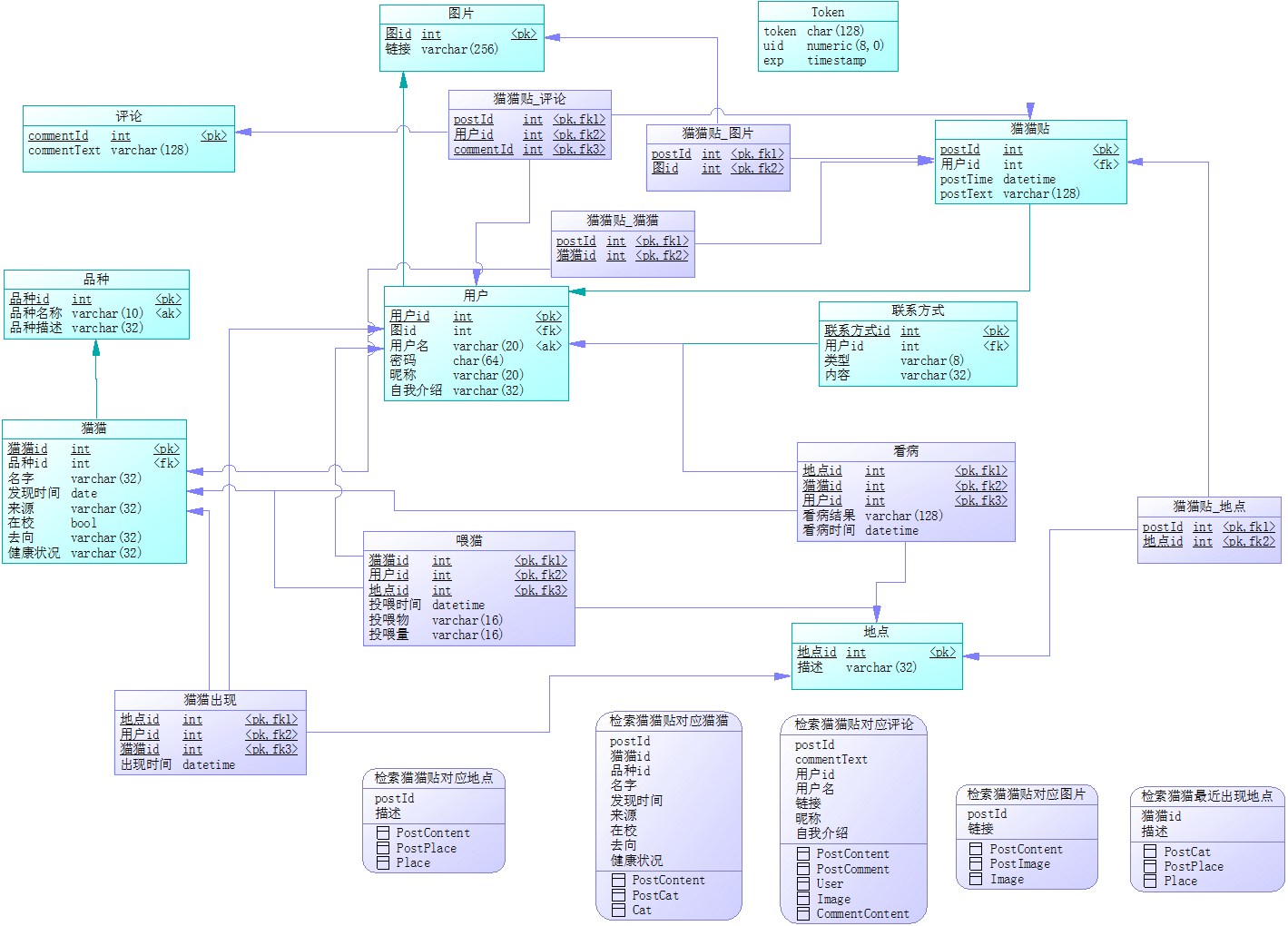
### LDM图

*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。*

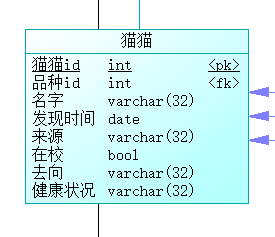


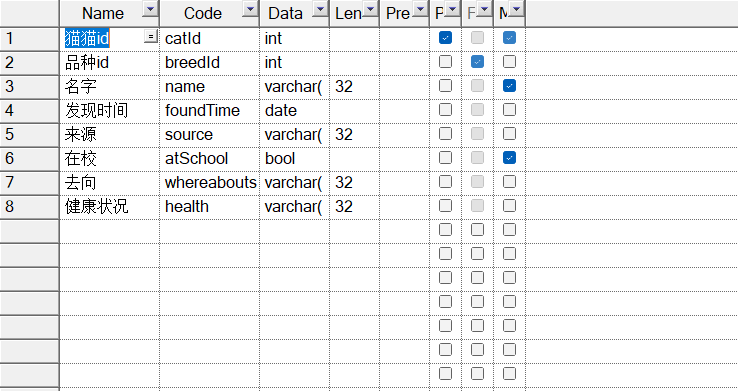
### 数据库表结构

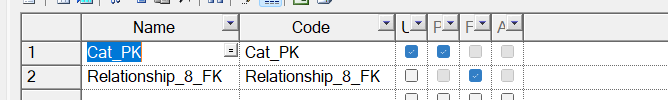
1. **表结构**

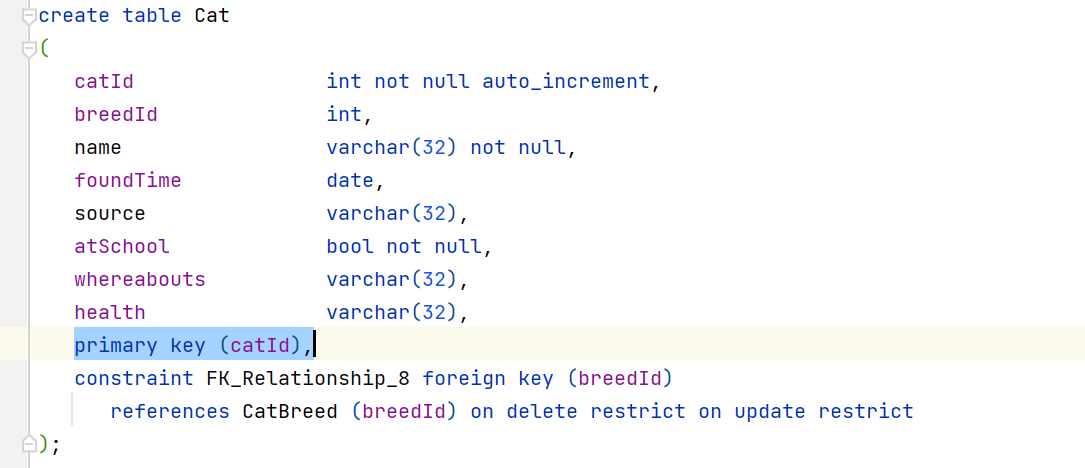
*选取2-3个比较有代表性的表结构截图，体现主键约束、外键约束、空值约束等。*

主键约束

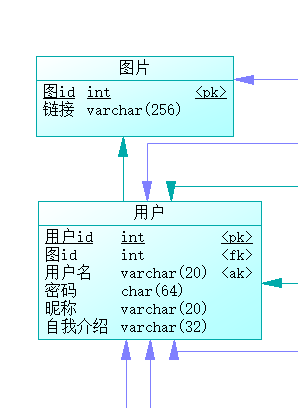


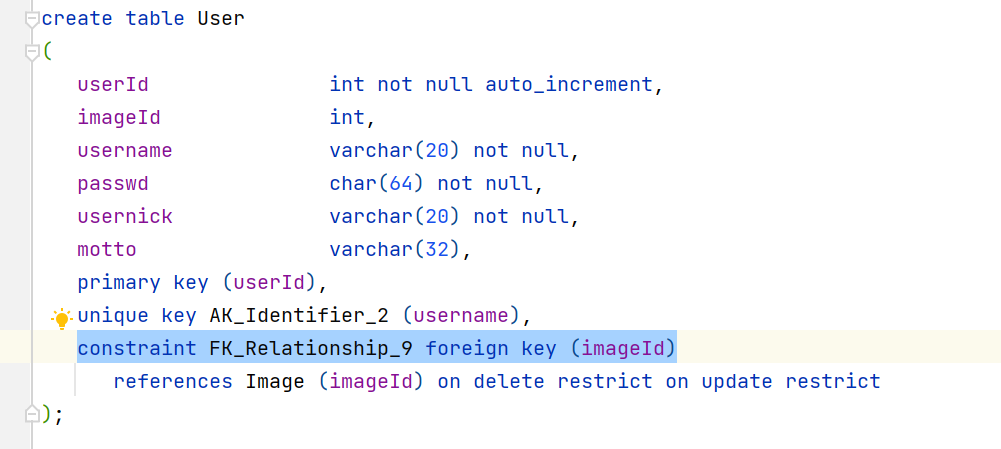




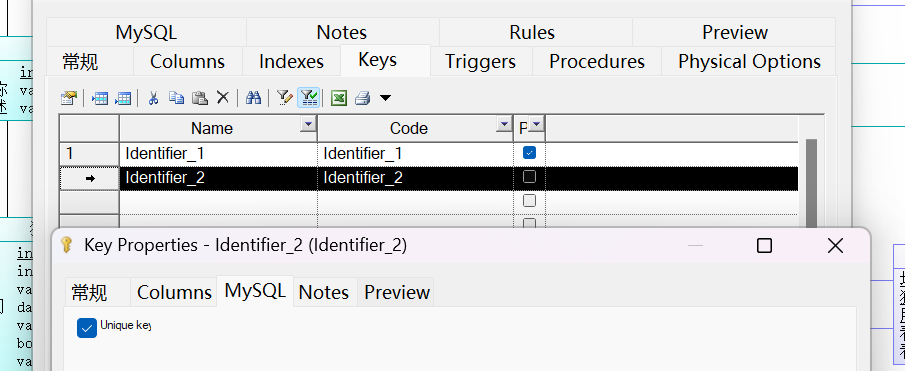


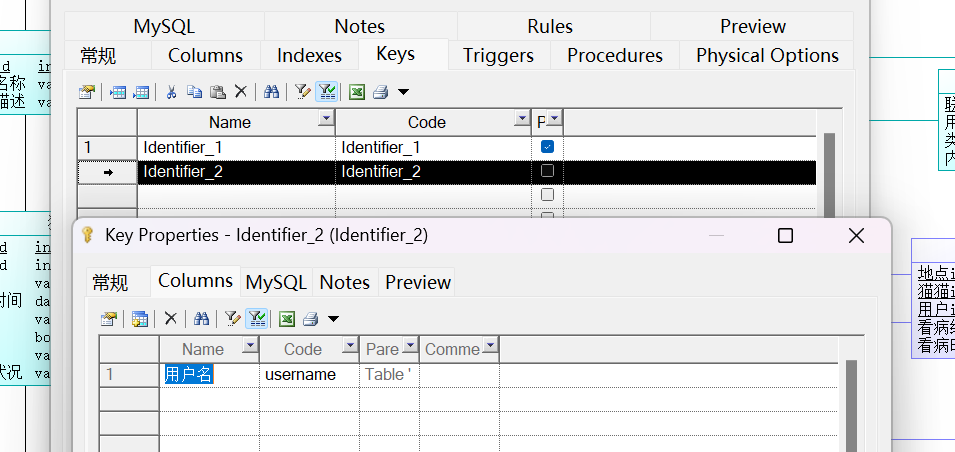
外键约束：用户头像，每个用户都有一个头像图片。

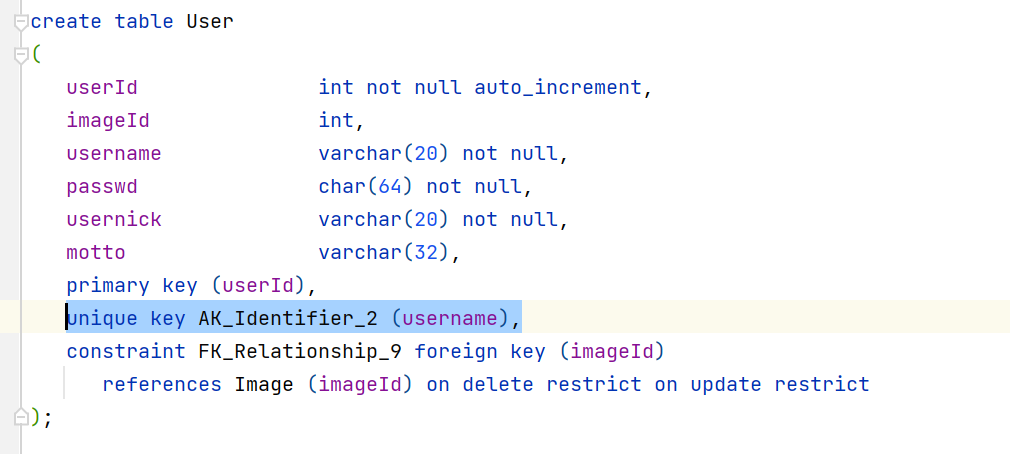




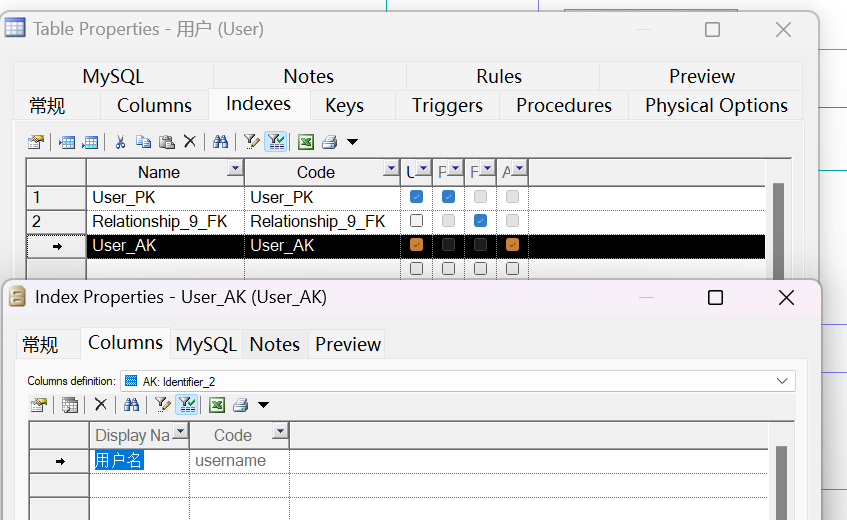
唯一约束：用户名必须唯一







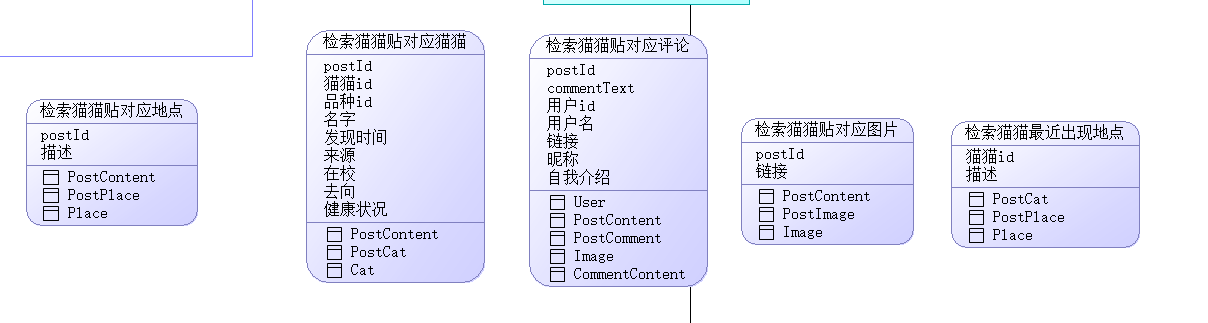
1. **索引**
2. 索引截图



1. 使用场景（用途）

唯一索引，保证用户名唯一

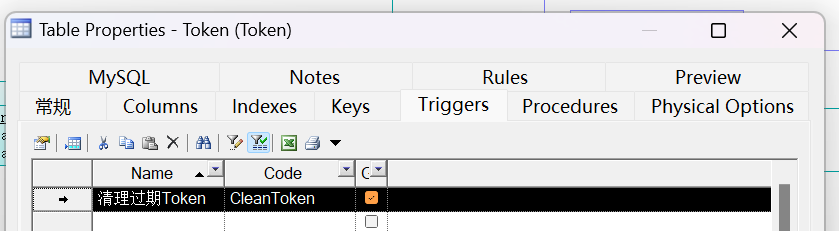
1. **视图**
2. 视图截图

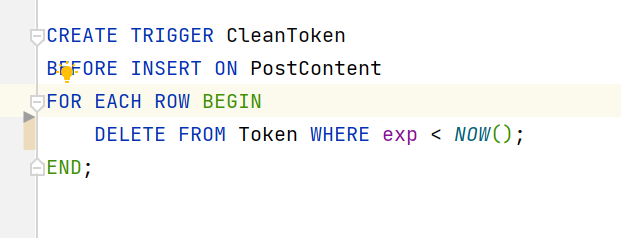


1. 使用场景（用途）

（用途见图片中每个视图的标题）

1. **触发器**
2. 触发器截图



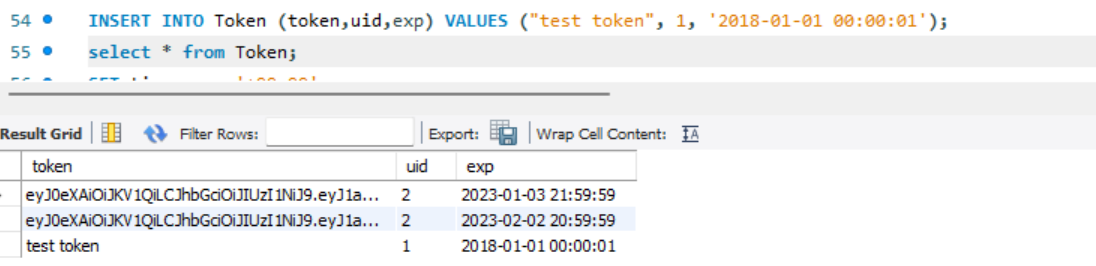


1. 使用场景（用途）

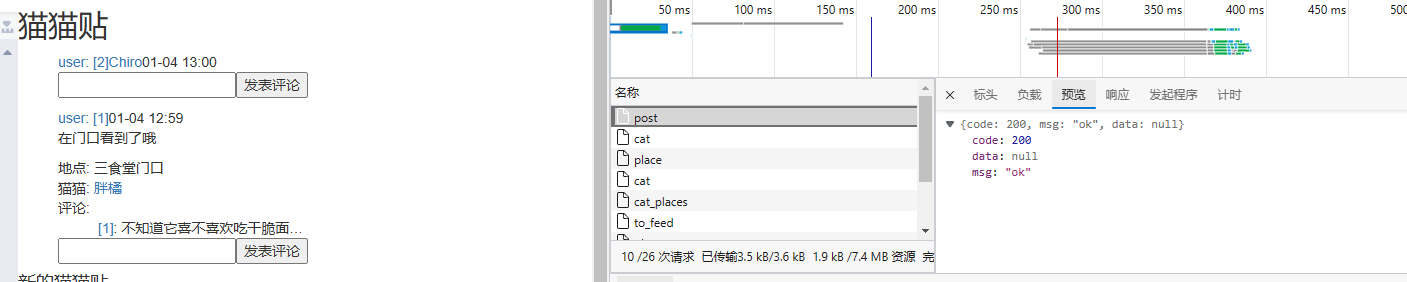
每次插入Token前清理过期的Token，节约储存空间。

1. 验证触发器

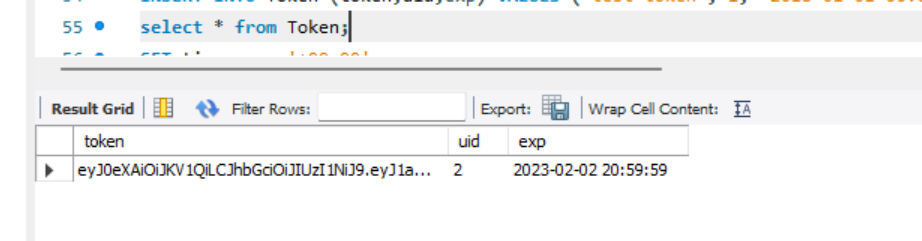
插入test token



发布新的猫猫贴



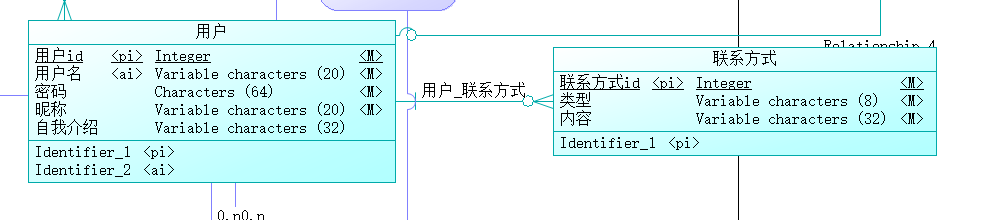
过期token已经被删除



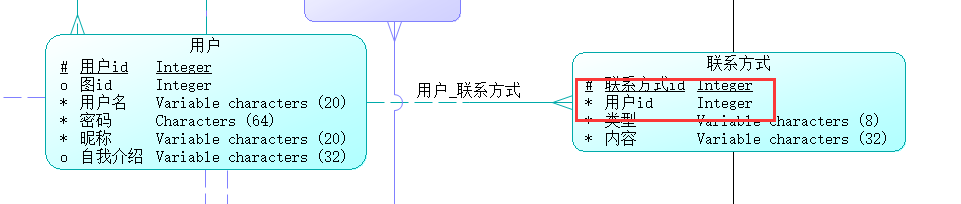
### 分析

*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）。*

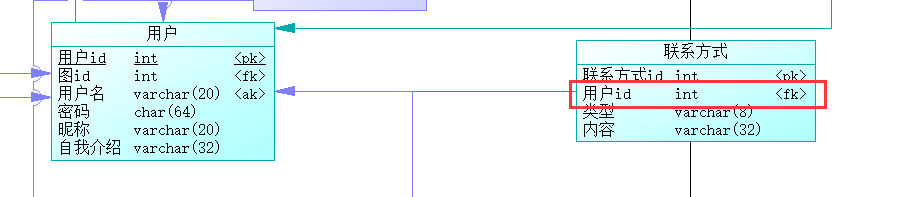
ER:



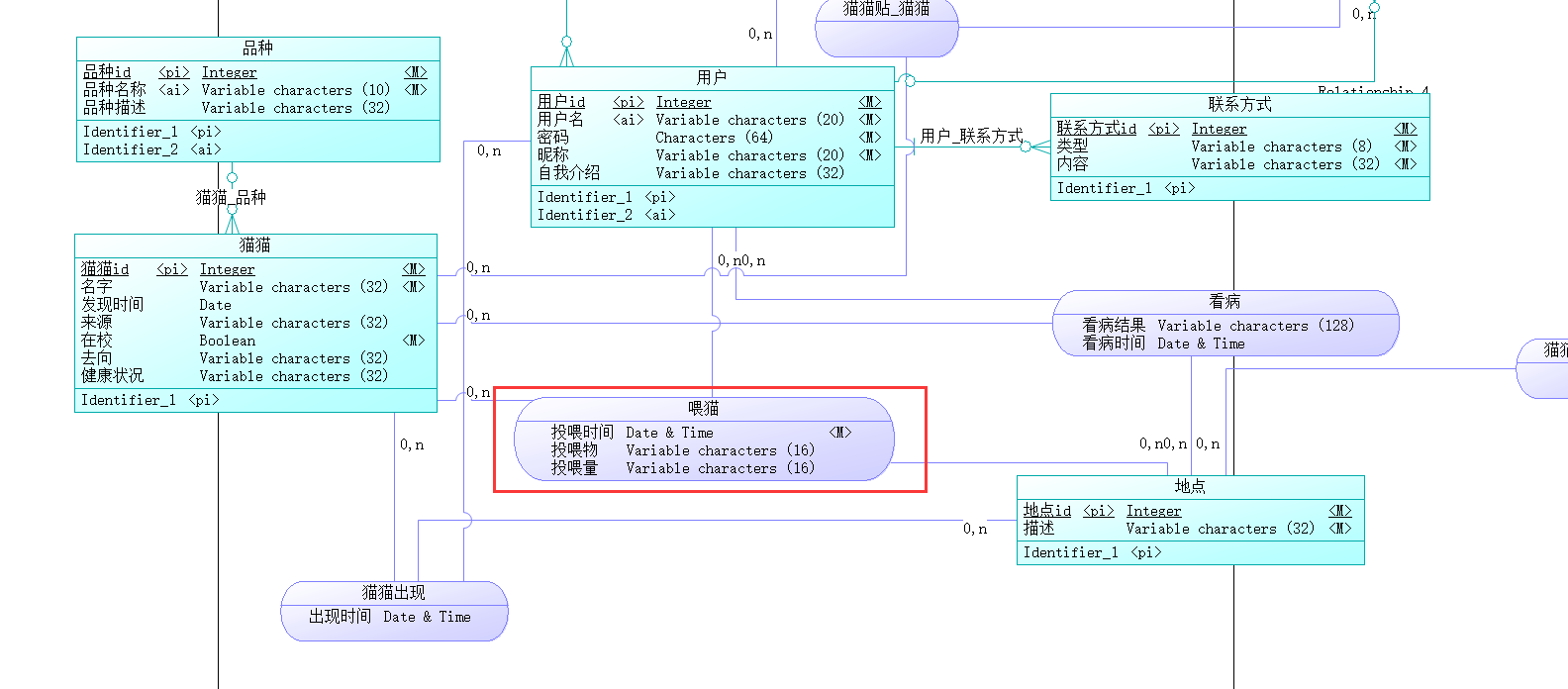
LDM:



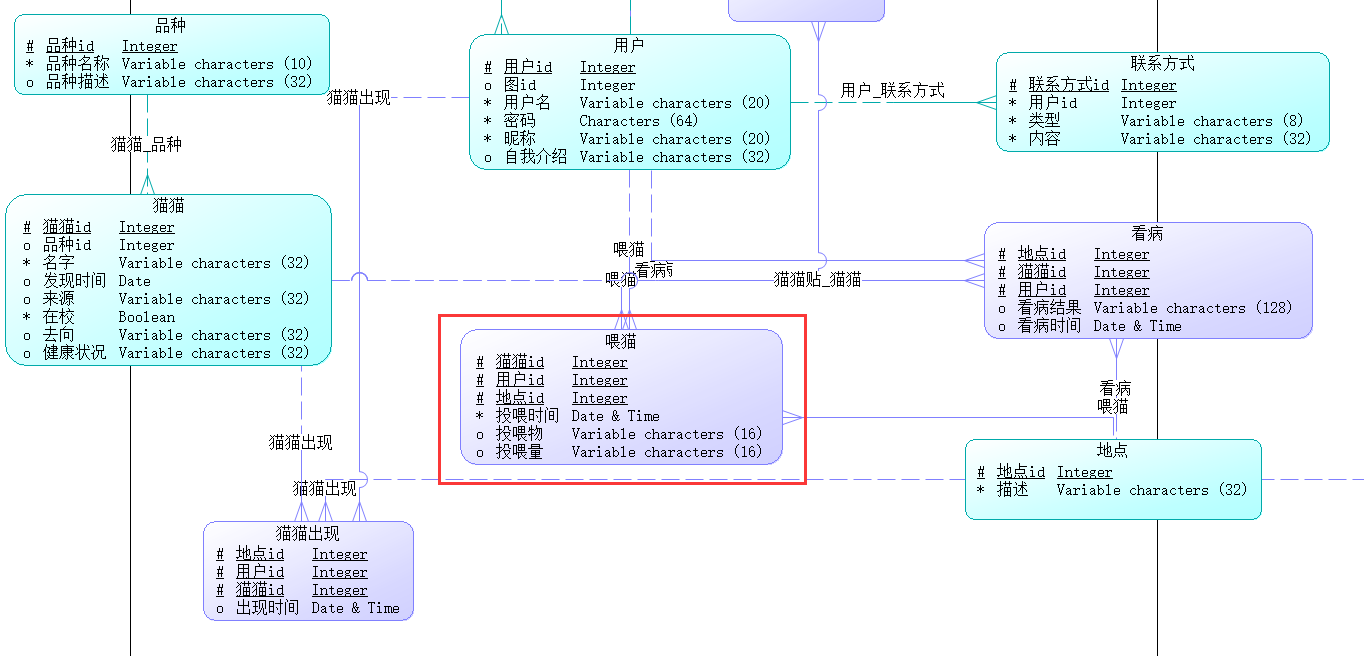
PDM:



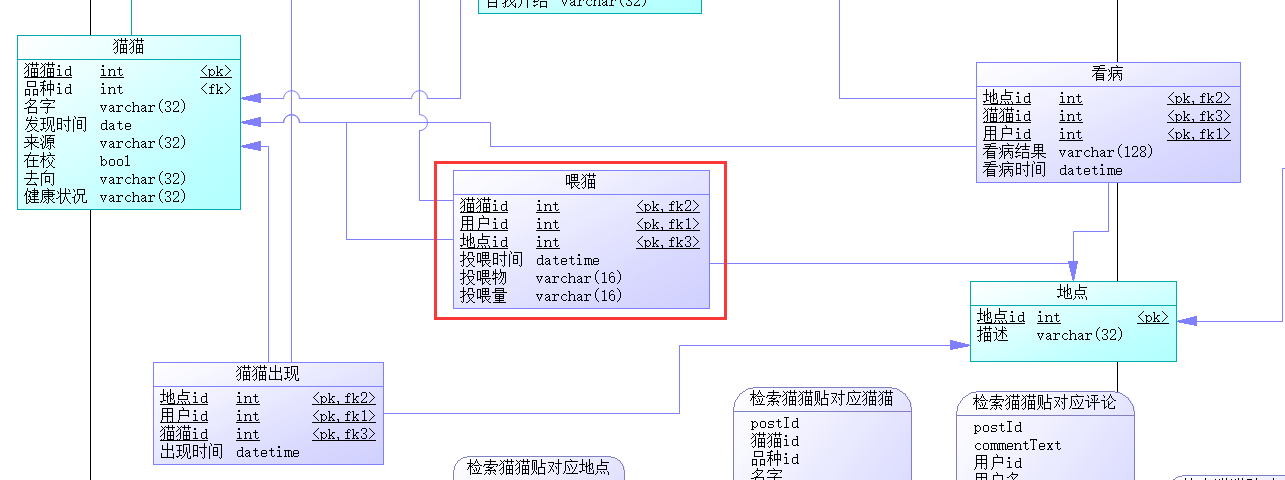
ER:



LDM:



PDM:



可以看出，ER->LDM时为联系添加了依赖的字段，LDM->PDM时将这些字段设置了外键约束。

# 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。

本次实验中实现了一个围绕着数据库的系统，考验了我们的系统构建能力。

出现的问题：

1. 数据库应该如何设计才能符合复杂功能的需求
2. 后端应该如何沟通数据库
3. 前端后端之间的交互如何以数据模型为中心
4. 前后端内存中的数据模型和数据库中数据模型的差异

本次实验中实验了一些期望解决上面问题的方案，如

1. 使用相同的语言构建前端后端，将整个前后端系统放在一个整体的项目中，利于将数据模型同时应用在前后端，减少序列化/反序列化等的代码量
2. 设计时首先从需求开始分析，并分析出需要实现的功能，再依据功能设计数据库，然后根据数据库模型设计前后端沟通API
3. 将模型直接抽象为Rust struct结构，能够方便读写数据库的时候的类型转换

同时，本次实验中还实验了许多比较新颖的技术，如Web Assembly、Rust等，检验了这些技术在构建系统方面的能力。