



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: 校园猫管理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: 计科 1 班
学生学号: 200110102
学生姓名: 王恩堂
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

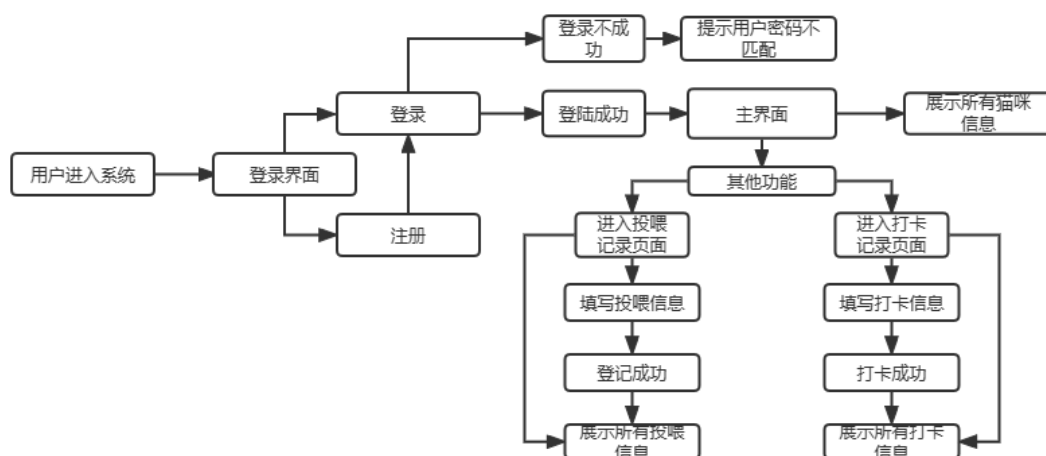
请填写用到的操作系统和主要开发工具。

本次实验使用 windows11 操作系统,主要开发工具为 IntelliJ IDEA 2021.3.2,MySQL, power designer

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。



1.登录注册功能:

用户第一次使用校园猫管理系统时需要进行注册,输入用户名和密码,点击注册,若系统提示“注册成功”则用户成功完成注册。

Login

哈尔滨工业大学(深圳)
Harbin Institute of Technology, Shenzhen

用户名

密码

登录 注册

localhost:8080 显示

注册成功!

确定

Login



12345

12345

登录

注册

输入正确的用户名和密码点击登录，服务器执行 sql 语句在数据库中查询是否存在相应的用户，若存在，则登录成功；若不存在，则提示账户与密码不一致。



2.主界面

登陆成功后进入主界面，服务器发送数据库链接申请，执行 sql 语句查询所有猫咪信息，并返回给浏览器，浏览器展示所有猫咪信息。



id	昵称	种类	颜色	性格	栖息地	打卡次数	操作	
1	皮卡丘	英国短毛	黑	勇敢	T5	3	修改	删除
2	妙蛙种子	阿比西尼亚	灰	调皮	T5	1	修改	删除
3	杰尼龟	伯曼	白	浮躁	主楼	1	修改	删除
4	小火龙	异国短毛	蓝	慎重	T2	4	修改	删除
5	伊布	苏格兰折耳	黄	实干	T3	2	修改	删除

点击喂食记录进入投喂界面，点击打卡记录进入打卡界面。

3.喂食界面

localhost:8080/database_war_exploded/feed?

名字:admin
编号:2

输入猫id 输入用户名 输入地址 输入时间 输入食物种类 登记 图鉴 打卡记录

id	猫id	用户名	喂食地址	记录时间	食物种类	更新时间
1	1	admin	T5	2022/12/7	鱼干	
2	2	admin	主楼	2022/12/10	猫粮	
3	4	admin	T2	2022/12/8	猫粮	
4	3	admin	T3	2022/12/13	猫粮	
5	5	admin	T5	2022/12/13	猫粮	
6	3	aaa	T5	2022/12/20	鱼干	
7	4	a1	主楼	2022/12/19	鱼干	2022-12-27 20:35:12
8	2	aaa	理学校	2022/12/18	鸡胸	2022-12-27 21:05:47
9	3	admin	G707	2022/12/21	三文鱼	2022-12-27 21:06:47
10	4	a1	T2	2022/12/22	鸡蛋	2022-12-27 21:10:03
11	1	admin	T4	2022/12/22	猫粮	2022-12-27 21:11:51
14	1	admin2	T5	2022/12/28	猫粮	2022-12-28 19:59:48
15	4	admin	T3	2022/12/28	鱼干	2022-12-28 21:07:56

进入喂食界面后，服务器向数据库发送链接请求，执行 sql 语句查询全部投喂信息并发送给浏览器，浏览器展示全部投喂信息。用户可以填写投喂信息并点击登记进行投喂记录新增。此外，用户可以点击图鉴返回主界面， 点击打卡记录进入打卡界面。

4.打卡界面

localhost:8080/database_war_exploded/record?

名字:admin
编号:2

输入猫id 输入用户名 输入地址 输入时间 打卡 图鉴 喂食记录

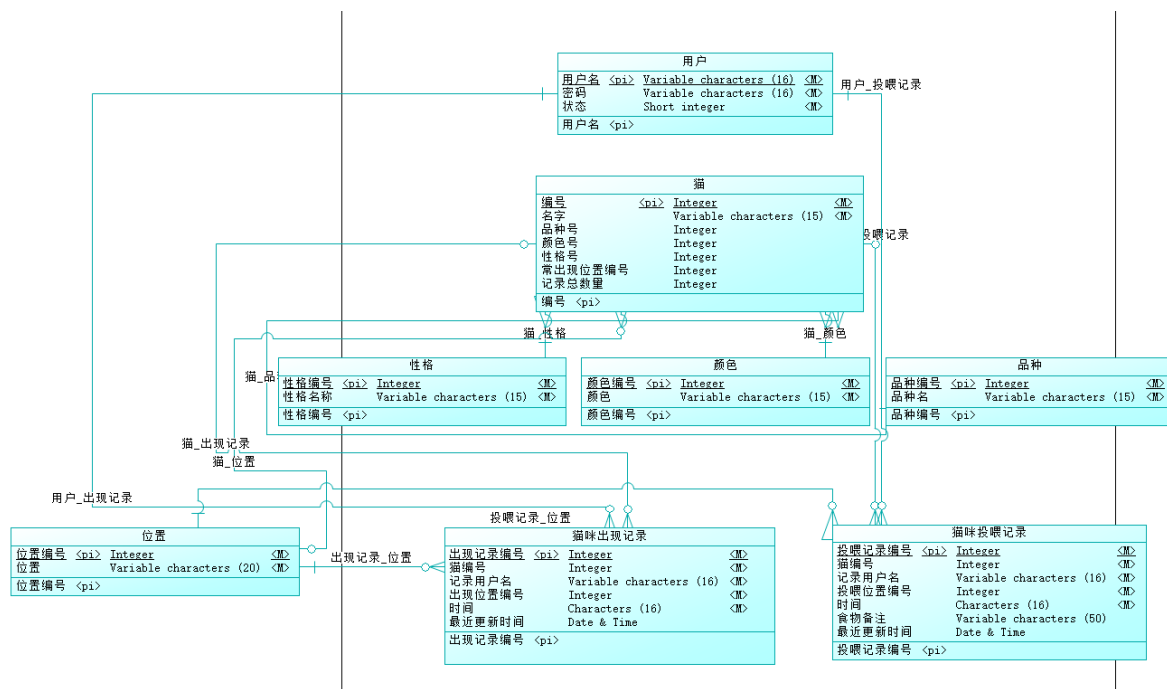
id	猫id	用户名	打卡地址	记录时间	更新时间
1	1	admin	T5	2022/12/7	
2	3	admin	T2	2022/12/8	
3	4	admin	主楼	2022/12/10	
4	2	admin	T5	2022/12/6	
5	3	admin	主楼	2022/12/11	
6	5	aaa	T5	2022/12/20	2022-12-27 20:27:53
7	4	a1	主楼	2022/12/19	2022-12-27 20:45:56
8	2	aaa	理学校	2022/12/18	2022-12-27 21:04:22
9	1	admin	G707	2022/12/21	2022-12-27 21:07:10
11	3	admin3	T5	2022/12/28	2022-12-28 19:55:22
12	1	admin2	T5	2022/12/28	2022-12-28 19:59:12

进入打卡界面后，服务器向数据库发送链接请求，执行 sql 语句查询全部打卡信息并发送给浏览器，浏览器展示全部打卡信息。用户可以填写打卡信息并点击打卡进行打卡记录新增。此外，用户可以点击图鉴返回主界面， 点击喂食记录进入喂食界面。

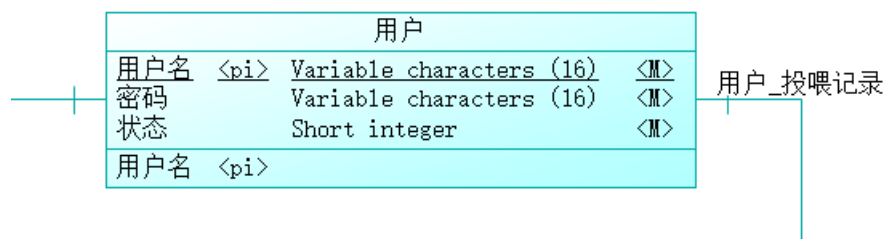
2.2 数据库设计

2.1.1 ER 图

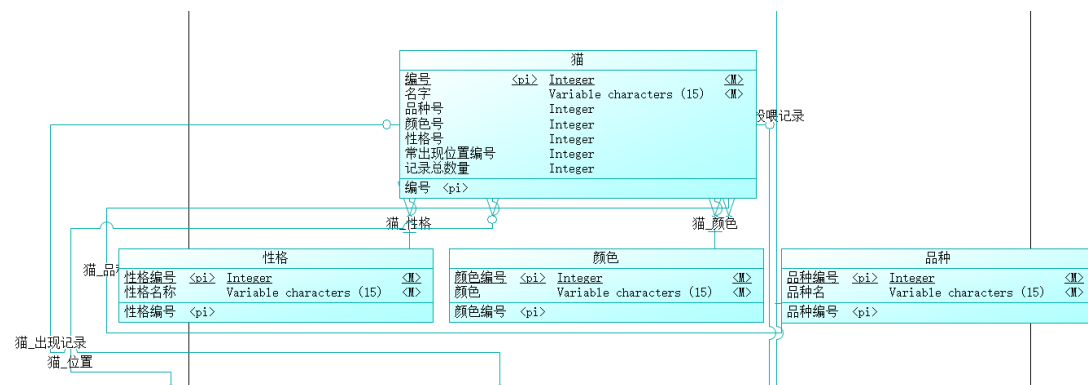
要求：截图务必清晰，如果图太大可截图一个总图，然后再分块截图。如果看不清截图会影响成绩。



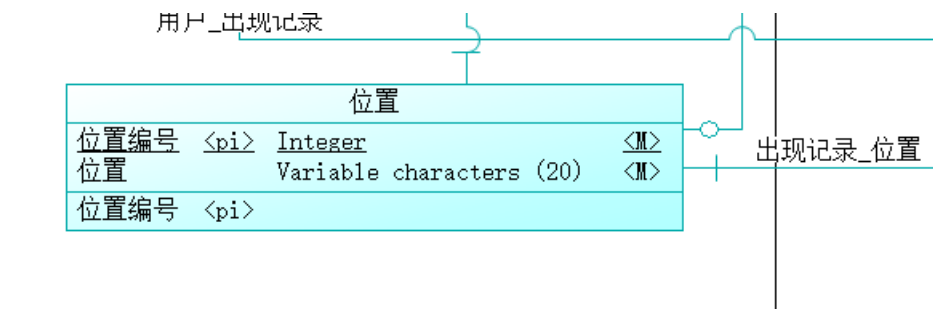
用户信息：



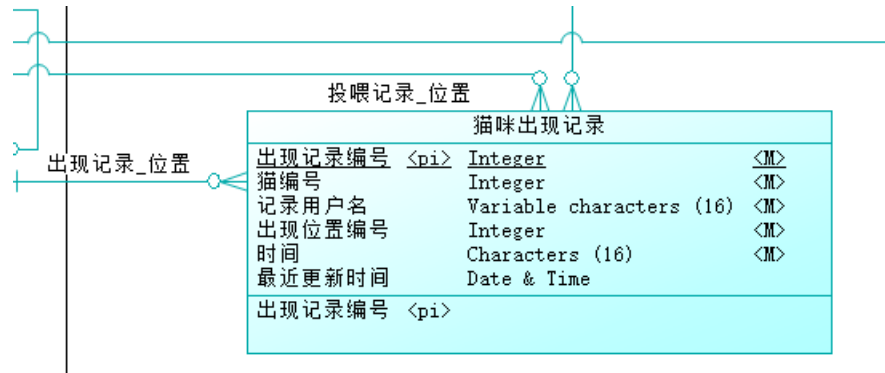
校园猫信息：



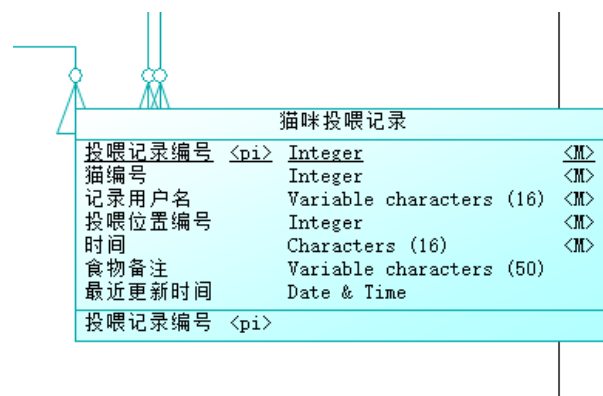
位置信息：



打卡信息:

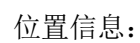
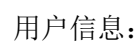


投喂信息:



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。



猫咪出现记录			
出现记录编号	<pi>	Integer	<
猫编号	<fi1>	Integer	<
记录用户名	<fi2>	Variable characters (16)	<
出现位置编号	<fi3>	Integer	<
时间		Characters (16)	<
最近更新时间		Date & Time	
出现记录编号 <pi>			

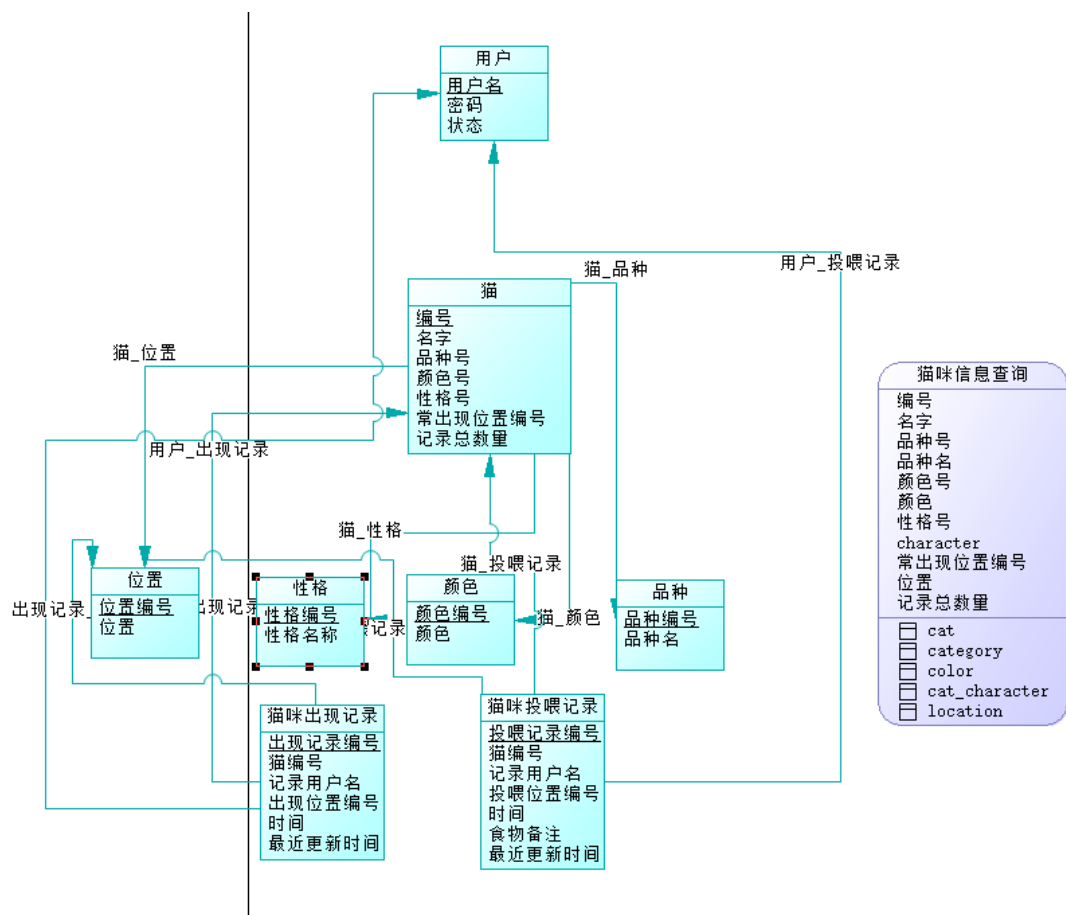
投喂信息：

位置

猫咪投喂记录			
投喂记录编号	<pi>	Integer	<
猫编号	<fi1>	Integer	<
记录用户名	<fi2>	Variable characters (16)	<
投喂位置编号	<fi3>	Integer	<
时间		Characters (16)	<
食物备注		Variable characters (50)	<
最近更新时间		Date & Time	
投喂记录编号 <pi>			

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。



2.1.4 数据库表结构

1、 表结构

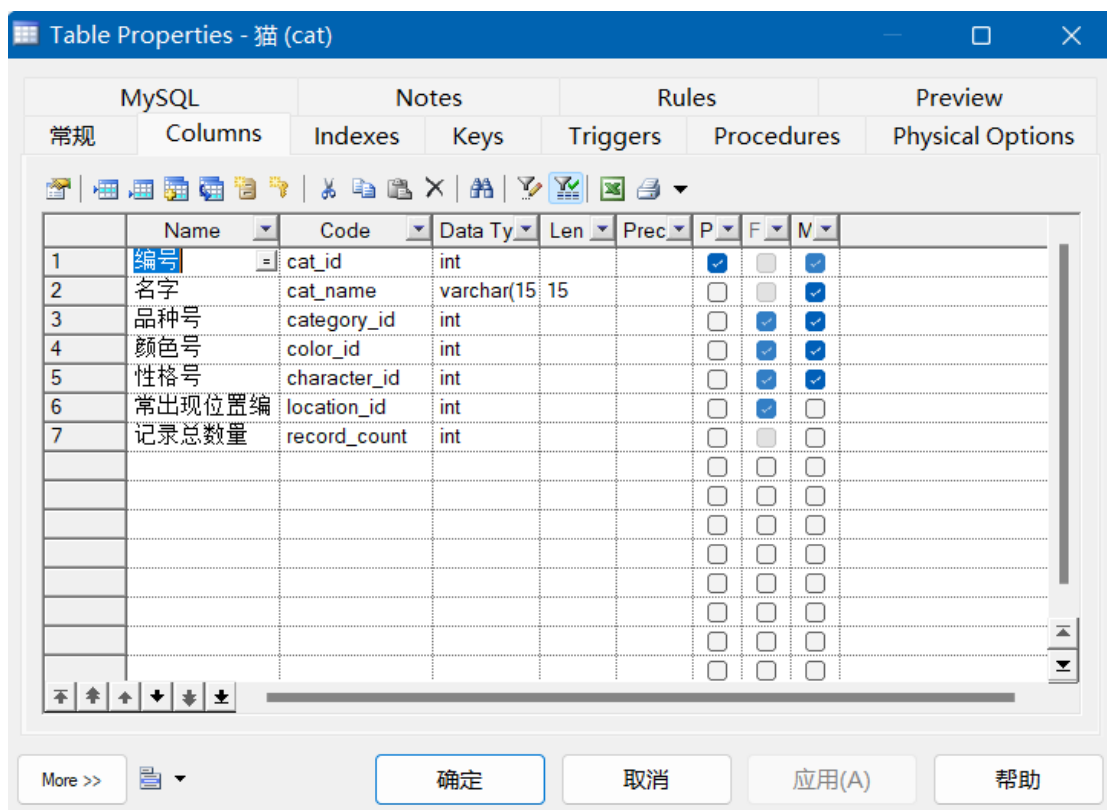
选取 2-3 个比较**有代表性的**的表结构截图，体现主键约束、外键约束、空值约束等。

1)

猫		
编号	<pi>	Integer
名字		Variable characters (15)
品种号	<fi1>	Integer
颜色号	<fi2>	Integer
性格号	<fi3>	Integer
常出现位置编号	<fi4>	Integer
记录总数量		Integer
编号	<pi>	

校园猫信息表

在表详细信息中可以看到校园猫信息表的基本属性和相关约束。

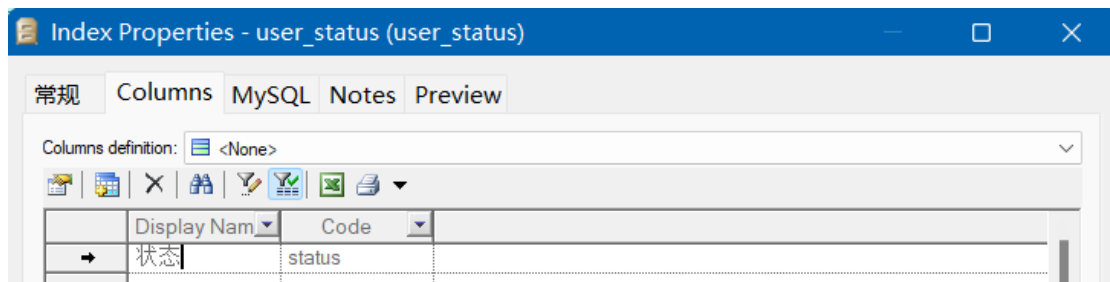


如图，三列选项分别代表是否为主键，是否为外键，是否强制。其中编号为校园猫信息表的主键，具有主键约束，品种号，颜色号，性格号，常出现位置编号为外键，具有外键约束，分别联系着品种表，颜色表，性格表和位置表。同时编号、名字、品种号、颜色号、性格号属性均存在空值约束，即均不能为空。可以通过校园猫信息中的各类编号到相应的表中查询文本信息。

2) 投喂记录表

猫咪投喂记录		
投喂记录编号	<pi>	Integer
猫编号	<fi1>	Integer
记录用户名	<fi2>	Variable characters (16)
投喂位置编号	<fi3>	Integer
时间		Characters (16)
食物备注		Variable characters (50)
最近更新时间		Date & Time
投喂记录编号	<pi>	

一条投喂记录信息由投喂记录编号、猫编号、记录用户名、投喂位置编号、记录时间、食物种类、最近更新时间构成。由投喂记录编号作为主键，猫编号，投喂位置编号作为外键。最近更新时间不存在空值约束，采用触发器的方式，当猫咪投喂记录表添加新的信息或修改历史信息时，使用触发器和 NOW() 函数为最近更新时间进行赋值。

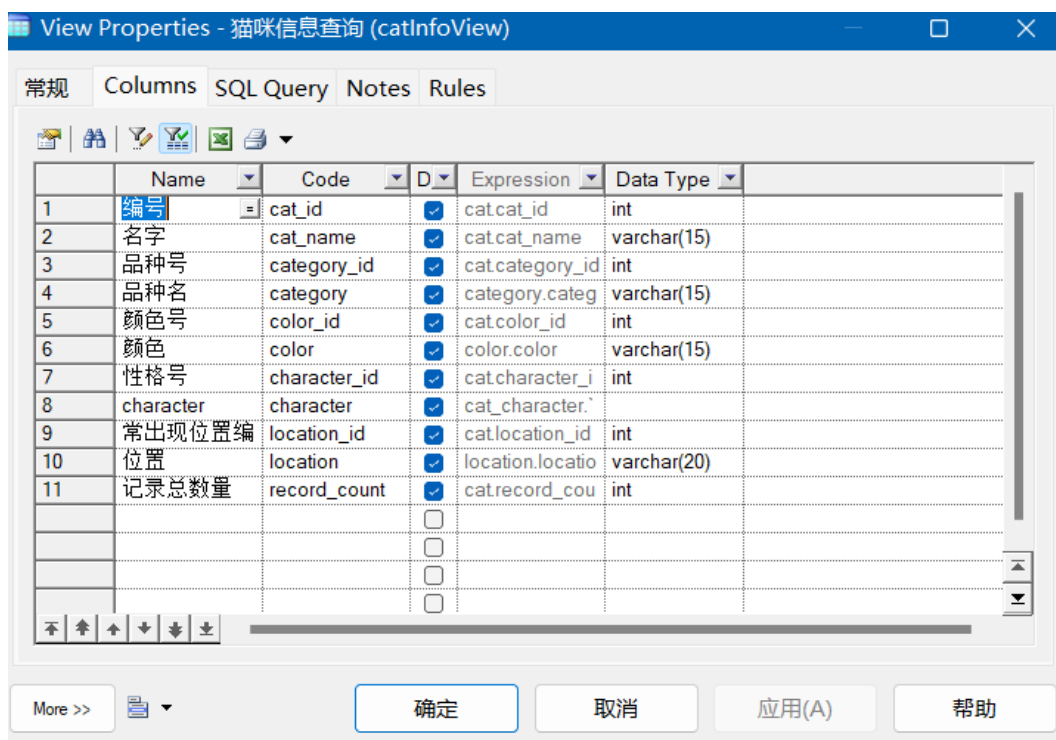
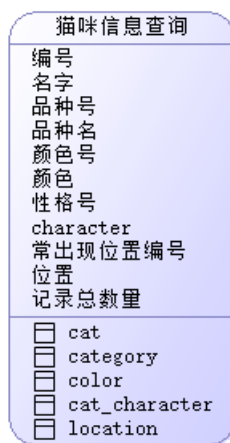


2) 使用场景（用途）

注册后的用户状态初始化时均设置为 1，当注销或冻结用户时，将用户状态设置为 0，设计索引可以快速定位用户状态为 0 的用户。

3、 视图

1) 视图截图



Query 1 feed_record - Table

Table Name: feed_record Schema: catdb

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci Engine: InnoDB

Comments:

▼ BEFORE INSERT
feed_record_BEFORE_INSERT
AFTER INSERT
BEFORE UPDATE
AFTER UPDATE
BEFORE DELETE
AFTER DELETE

```

1 • CREATE DEFINER='root'@'localhost' TRIGGER `feed_record_BEFORE_INSERT` BEFORE INSERT ON `feed_record`
2   FOR EACH ROW BEGIN
3     set new.last_update=now();
4   END

```

Columns Indexes Foreign Keys Triggers Partitioning Options

2) 使用场景（用途）

当投喂记录插入新记录时，触发器将调用 now()函数用系统时间更新最后更新时间属性。

3) 验证触发器

```

1 • INSERT INTO `catdb`.`feed_record` (cat_id,username,cat_position,record_time,feed_food)
2   VALUES (4, 'admin', 'T3','2022/12/28','鱼干');
3 • select * from feed_record;
4

```

record_id	cat_id	username	cat_position	record_time	feed_food	last_update
2	2	admin	主楼	2022/12/10	猫粮	NULL
3	4	admin	T2	2022/12/8	猫粮	NULL
4	3	admin	T3	2022/12/13	猫粮	NULL
5	5	admin	T5	2022/12/13	猫粮	NULL
6	3	aaa	T5	2022/12/20	鱼干	NULL
7	4	a1	主楼	2022/12/19	鱼干	2022-12-27 20:35:12
8	2	aaa	理号楼	2022/12/18	鸡胸	2022-12-27 21:05:47
9	3	admin	G707	2022/12/21	三文鱼	2022-12-27 21:06:47
10	4	a1	T2	2022/12/22	鸡蛋	2022-12-27 21:10:03
11	1	admin	T4	2022/12/22	猫粮	2022-12-27 21:11:51
14	1	admin2	T5	2022/12/28	猫粮	2022-12-28 19:59:48
15	4	admin	T3	2022/12/28	鱼干	2022-12-28 21:07:56
*	NULL	NULL	NULL	NULL	NULL	NULL

eed_record 1 x

Output

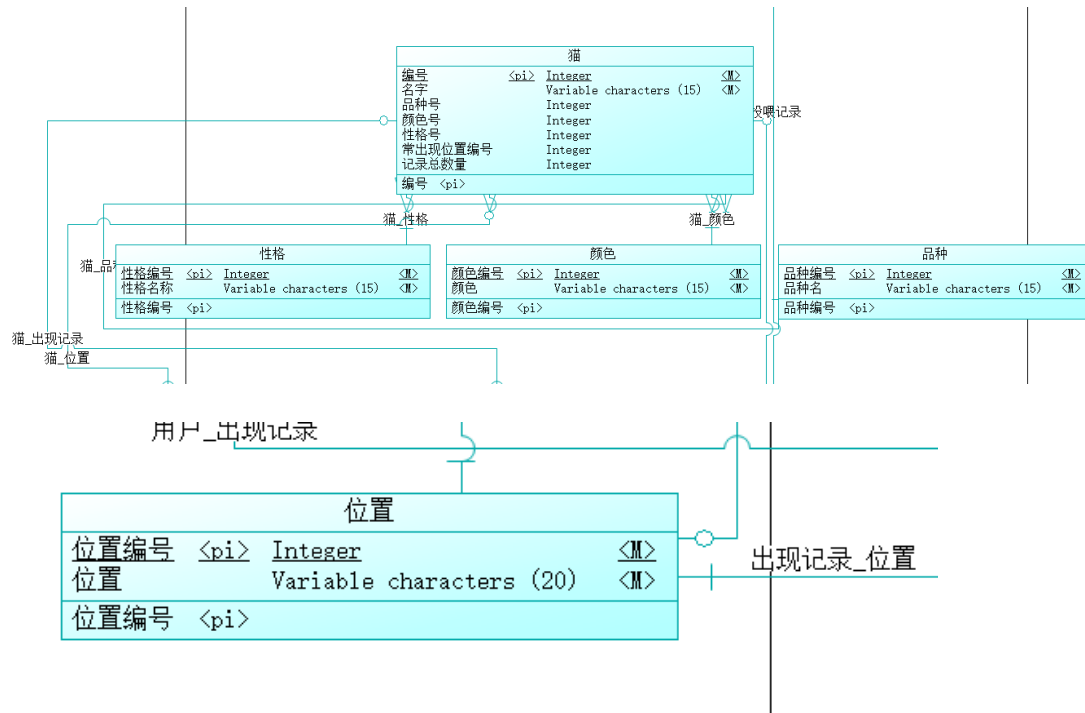
#	Time	Action	Message
✓ 1	21:07:56	INSERT INTO `catdb`.`feed_record` (cat_id,username,cat_position,record_time,feed_food) VALUES (4, 'admin', ...	1 row(s) affected
✓ 2	21:07:56	select * from feed_record LIMIT 0, 1000	13 row(s) returned

当插入一条新记录时，最后更新时间属性更新为系统时间。

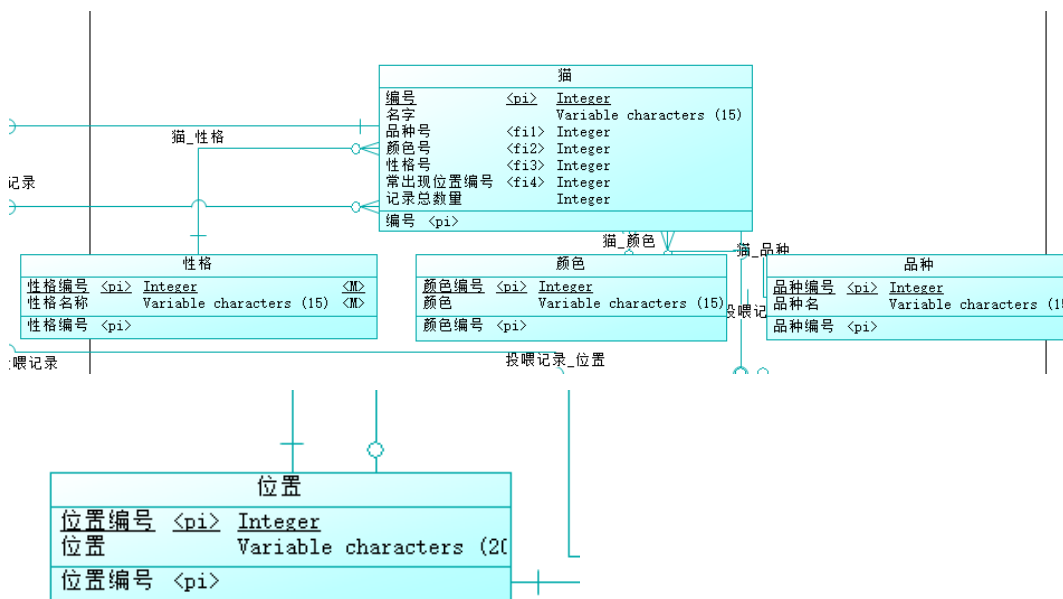
2.1.5 分析

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）。

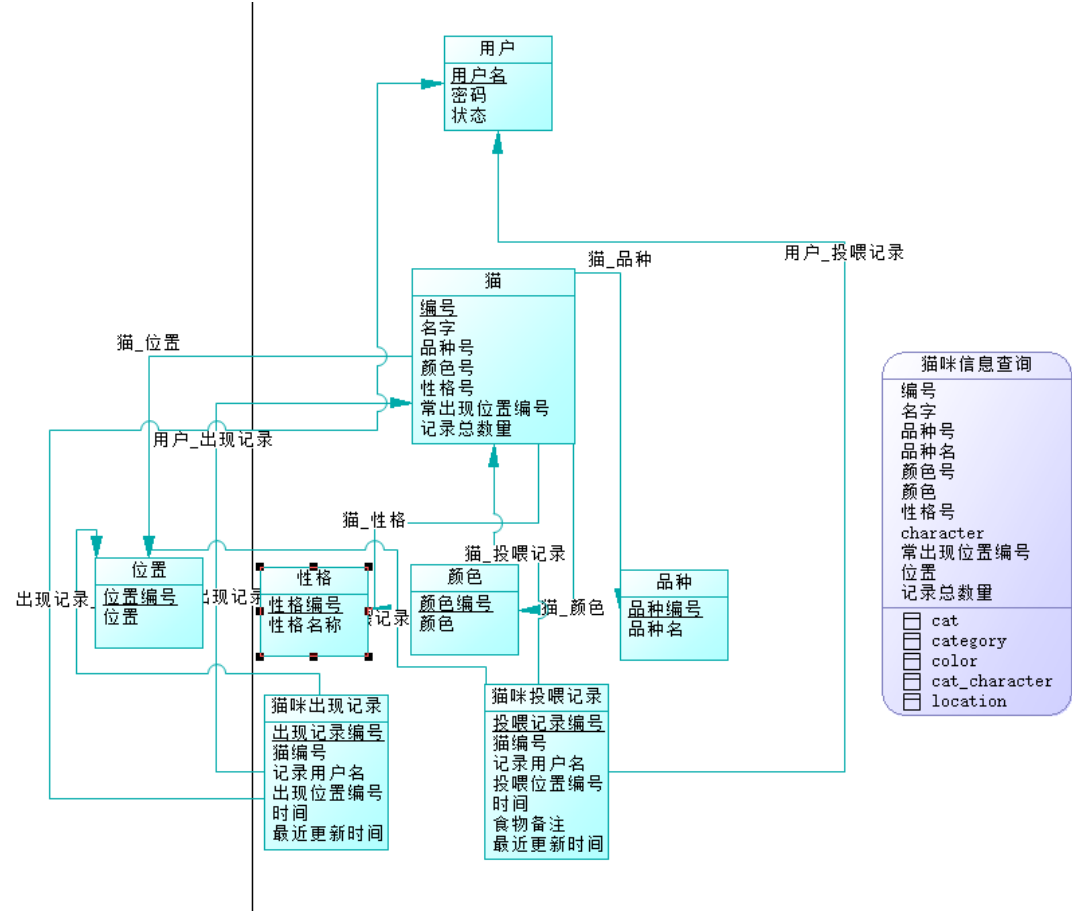
以猫猫信息为例，在 ER 图中，猫的属性由五张表校园猫信息表、性格表、颜色表、品种表、位置表构成，此时各个表都只有主键，而没有外键。



在 LDM 图中，在校园猫信息表中生成了四个外键，用于获取猫的品种、颜色、性格、位置信息。



在生成的 PDM 图中，表的内容更加简化，便于阅读。



最终得到的数据库表结构:

```
/*=====*/
/* Table: cat */
/*=====*/

create table cat
(
    cat_id          int auto_increment,
    cat_name        varchar(15) not null,
    category_id     int not null,
    color_id        int not null,
    character_id    int not null,
    location_id     int,
    record_count    int,
    primary key (cat_id)
);
```

```

/*=====*/
/* Table: cat_character */
/*=====*/
create table cat_character
(
    character_id      int auto_increment,
    `character`       varchar(15) not null,
    primary key (character_id)
);

/*=====*/
/* Table: category */
/*=====*/
create table category
(
    category_id      int auto_increment,
    category         varchar(15) not null,
    primary key (category_id)
);

create table color
(
    color_id         int auto_increment,
    color            varchar(15) not null,
    primary key (color_id)
);

create table location
(
    location_id      int auto_increment,
    location         varchar(20) not null,
    primary key (location_id)
);

```

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。

本次实验我选择使用 java 的 web 应用程序进行校园猫管理系统的开发，采用 web 程序可以使前后端分离，使项目结构更明确，但是对于之前没有 web 开发经验的我来说，是个不小的挑战。首先在前端开发上，没有编写 html,css 和 javascript 语言的经验，需要从头学起，通过博客和观看相关视频的方式获取相关信息。其次在后端的编写上，由于一开始不了解前后端通过调用 servlet 进行交互的方式，导致不能找出很多 bug 的位置。我一开始并没有采用先全部完成前端再编写后端的方式，而是前后端同时进行，通过增加页面的方式进行功能的扩充。在完成登录界面的编写后，我对于前后端交互的逻辑有了初步的认识，在编写剩下的界面的过程中，更加得心应手，同时在编写登录功

能时已经使用过与数据库的链接，这更加使得之后的工作大大简化。

在开发的初期，配置环境就耗费了我很大的精力，由于之前并没有接触过 **web** 开发，配置 **tomcat** 服务器和使用 **JDBC** 进行数据库链接就大费周章，但上手之后的开发便较为顺利，没有遇到太大的困难。

本次实验我独立编写了整个校园猫数据库管理系统，在一步步搭建系统的过程中，我学习了 **web** 前端开发的 **html,css** 和 **javascript**，以及后端调用 **servlet**，链接数据库的部分知识。在使用 **power designer** 编写数据库时，由 **ER** 图到 **LDM** 图再到 **PDM** 图的过程让我认识到了功能完备的数据库应当如何设计。在完成实验时，也收获了不少的成就感。