

《校易》数据库设计 说明书

所属学院：至诚学院
团队名称：码到成功
指导老师：张栋老师

项目时间：2019-2020 学年第二学期

目录

版本信息.....	1
第一章 引言.....	1
1.1 编写目的.....	1
1.2 背景.....	1
1.3 参考资料.....	1
第二章 外部设计.....	2
2.1 标识符和状态.....	2
2.2 命名约定.....	2
2.3 设计约定.....	2
第三章 结构设计.....	3
3.1 概念结构设计.....	3
3.1.1 实体和属性的定义.....	3
3.1.2 局部 E-R.....	5
3.1.3 全局 E-R.....	9
3.2 逻辑结构设计.....	9
3.2.1 模式.....	9
3.2.2 外模式.....	10
3.3 物理结构设计.....	10
第四章 运用设计.....	11
4.1 数据字典设计.....	11
4.2 安全保密设计.....	11
4.3 数据库设计.....	11
4.3.1 创建表.....	11

版本信息

版本/状态	修订人	修改日期	备注
第 1 版	码到成功	2020/4/18	
第 2 版	码到成功	2020/4/19	修改表结构

第一章 引言

1.1 编写目的

本数据库设计说明书是关于网络二手交易平台系统数据库设计，主要包括数据逻辑结构设计、数据字典以及运行环境、安全设计等。

本数据库设计说明书读者：用户，系统设计人员，系统测试人员，系统维护人。

本数据库设计说明书是根据系统需求分析设计所编写的。

本系统说明书为开发软件提供了一定基础。

1.2 背景

21 世纪代言人无疑是互联网，从上个世纪末出现的这项划时代技术，经过几十年的发展已臻成熟。现如今它渗透于我们生活中的点点滴滴，其强大的功能已让人们深刻认识。

随着现在诸多高校的扩招与校园信息化的开展，校园网络的建设将进入一个更加高速发展的时期。这对于我们来说也将有一个更好的环境去发展校园电子商务，建立属于高校大学生自己的交易平台。

当今社会大学生的人口基数已不少于任何一群体，高校学生同时也是一个高消耗人群，对于商品资源的使用自然不会少，要做到资源充分利用，同时也响应党十七大“节能减排”的号召，立足于满足高校学生二手交易的需要，二手交易平台显得更加必不可少。同为大学生，大家都具有更相似的消费心理，共同的生活环境，生活方式，所以之间交易的商品也都是各自需要的，从这一角度来看，二手交易平台的构建更具可行性和必要性。

1.3 参考资料

[1] 《数据库设计说明书-完整版》. <https://wenku.baidu.com/view/1b32448f26284b73f242336c1eb91a37f011325c.html>

[2] 《数据库表结构设计的几条准则》. <https://www.cnblogs.com/wyq178/p/8549715.html>

[3] 《数据库表设计（一对多、多对多）》. <https://blog.csdn.net/fighteryang/article/details/82848505>

[4] 《完整的开发文档数据库设计说明书》. <https://wenku.baidu.com/view/0176e7eb856a561252d36f56.html>

[5] 《软件需求规格说明书》. <https://github.com/CourseManagement/Data/blob/master/%E8%BD%AF%E4%BB%B6%E9%9C%80%E6%B1%82%E8%A7%84%E6%A0%BC%E8%AF%B4%E6%98%8E%E4%B9%A6.docx>

第二章 外部设计

2.1 标识符和状态

数据库软件的名称: MySQL 5.0

数据库的名称为: CAMPUS_TRANSACTION_SQL

2.2 命名约定

所有的数据库命名都是以模块的英文名组成, 英文单词之间以下划线分开, 这样能够统一数据库表的命名, 也能够更好的规范数据库表命名。

2.3 设计约定

在本系统中, 数据库的设计采用 PowerDesigner 进行, 并且采用面向对象的设计方法, 首先进行对象实体的设计, 最后将对象持久化到数据库中, 所有的表和表之间的关联(ER 图)都采用标准的 PowerDesigner 设计工具进行, 这样能够将整个系统的设计和数据库设计有机的结合起来。

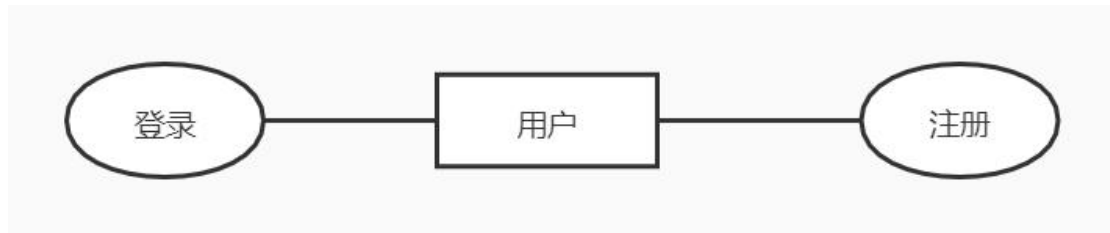
第三章 结构设计

3.1 概念结构设计

3.1.1 实体和属性的定义

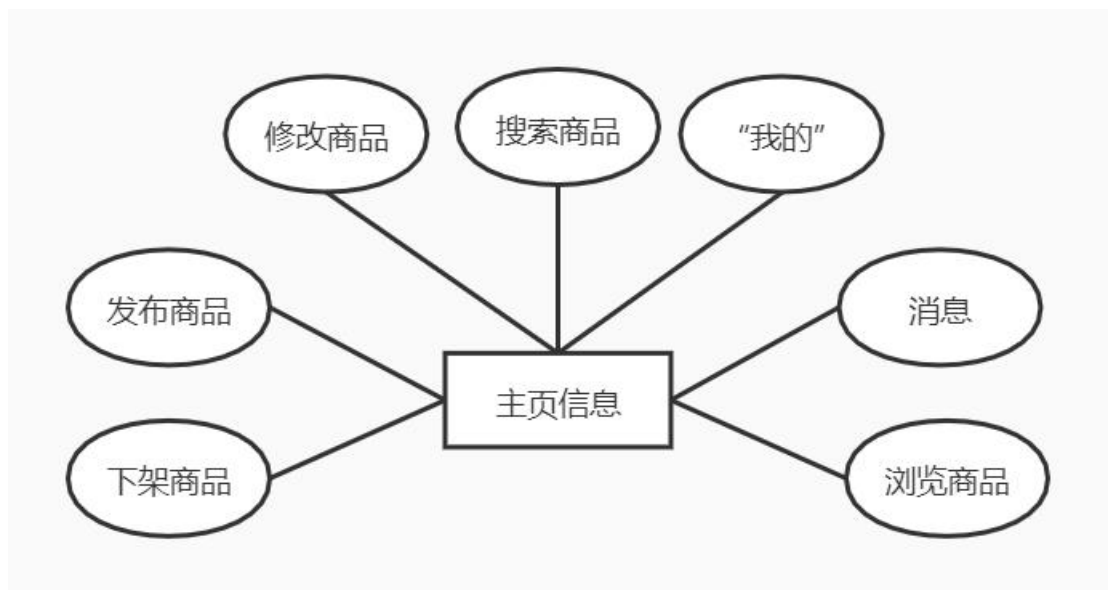
3.1.1.1 用户模块

小程序开始界面

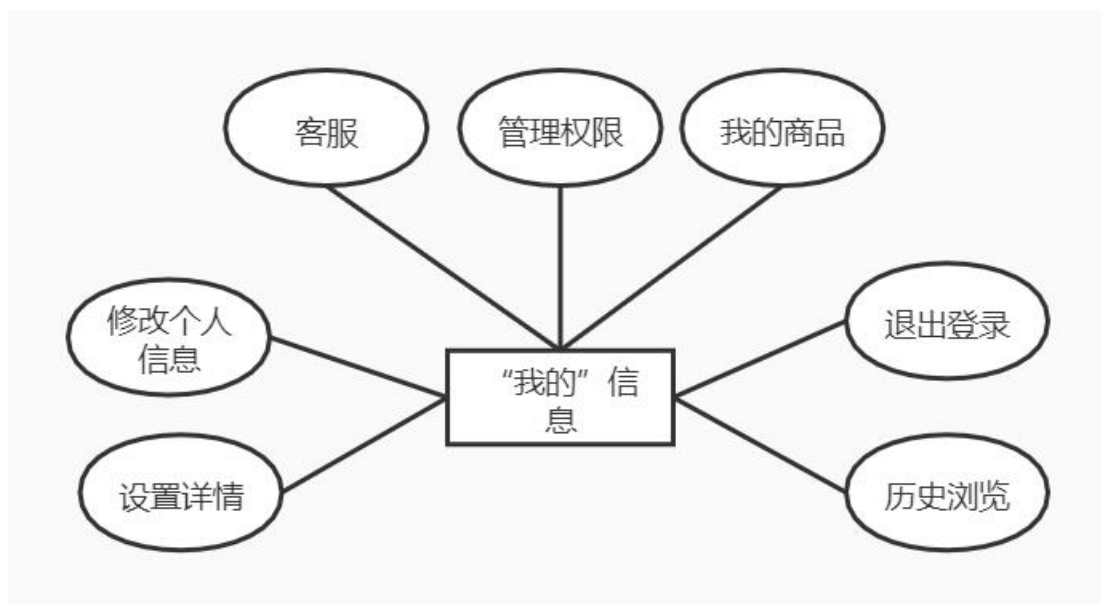


3.1.1.2 程序详情模块

主页界面

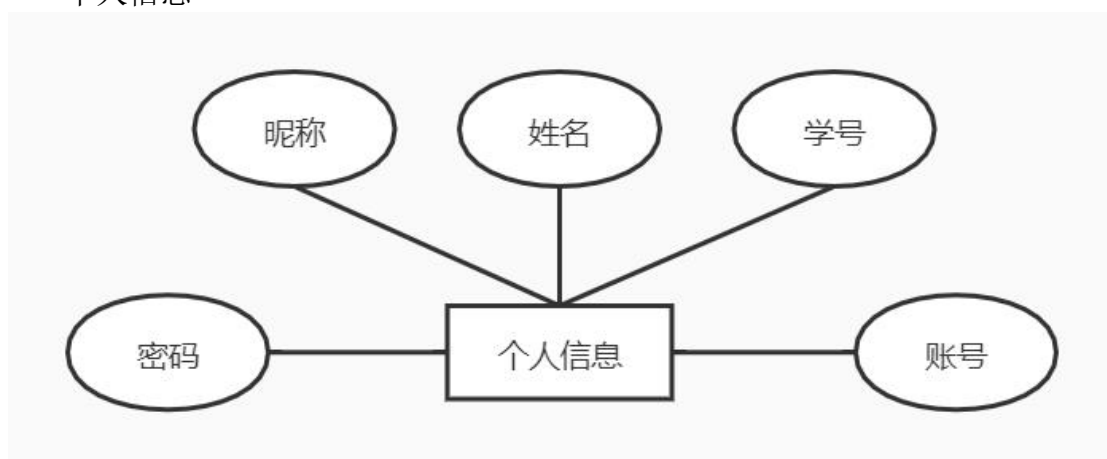


“我的”

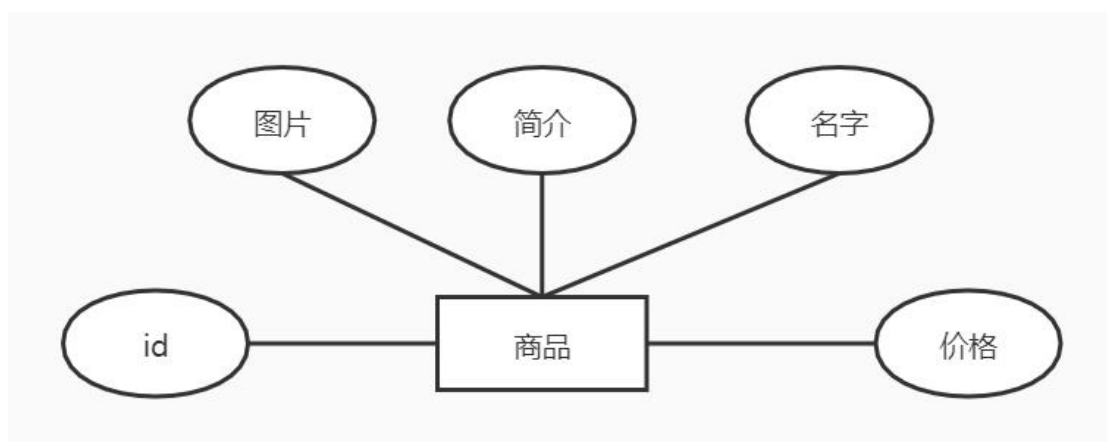


3.1.1.3 信息模块

个人信息

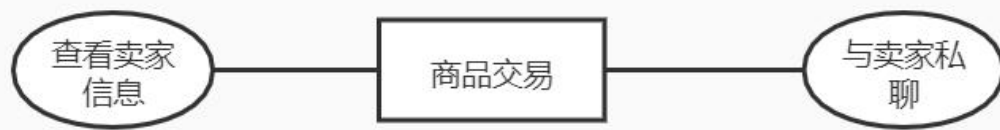


商品信息



3.1.1.4 交易模块

商品交易

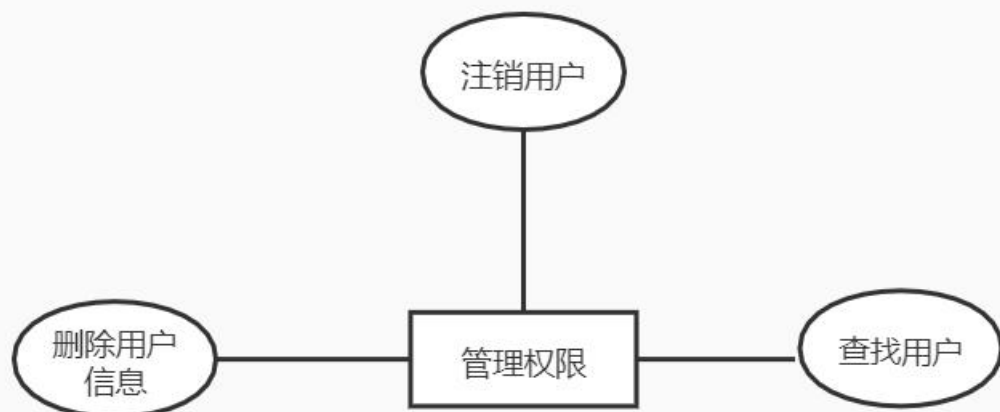


信息



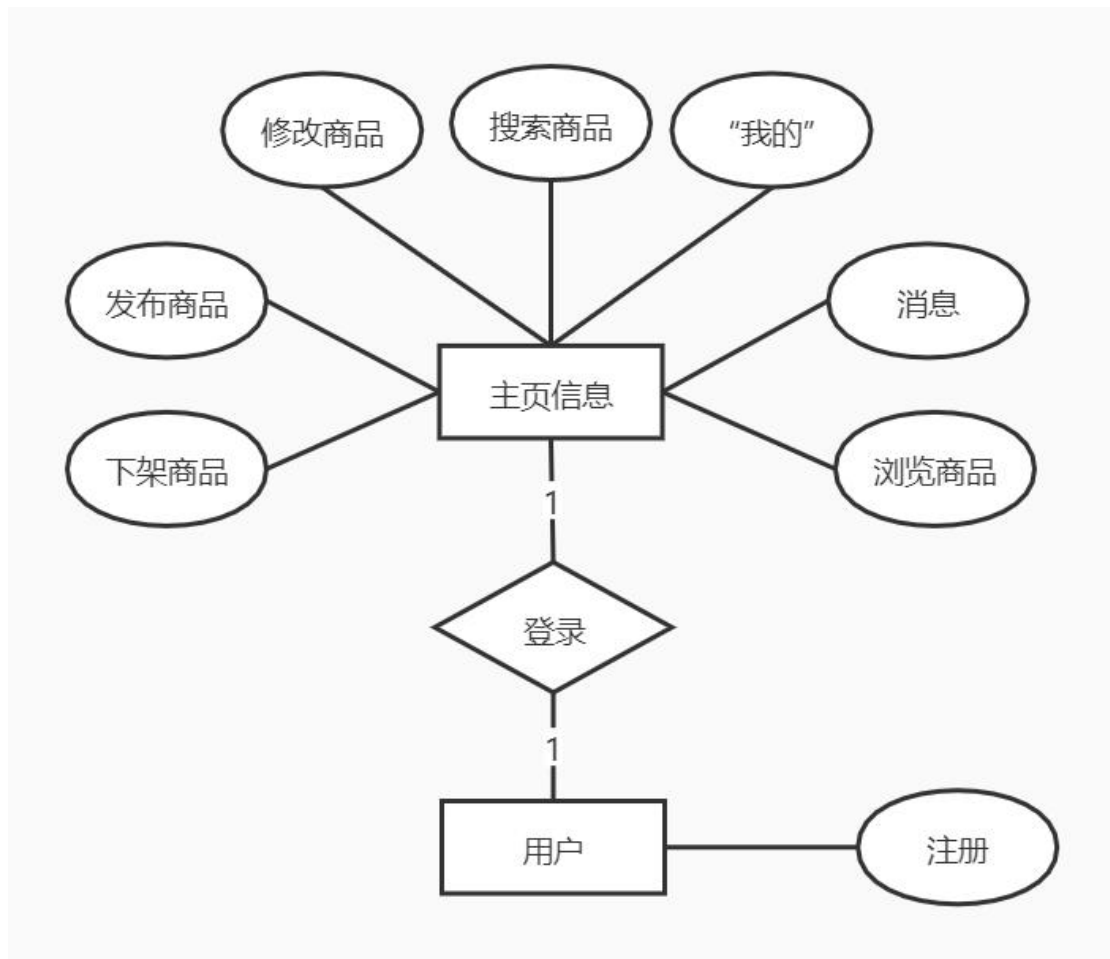
3.1.1.5 管理模块

管理权限

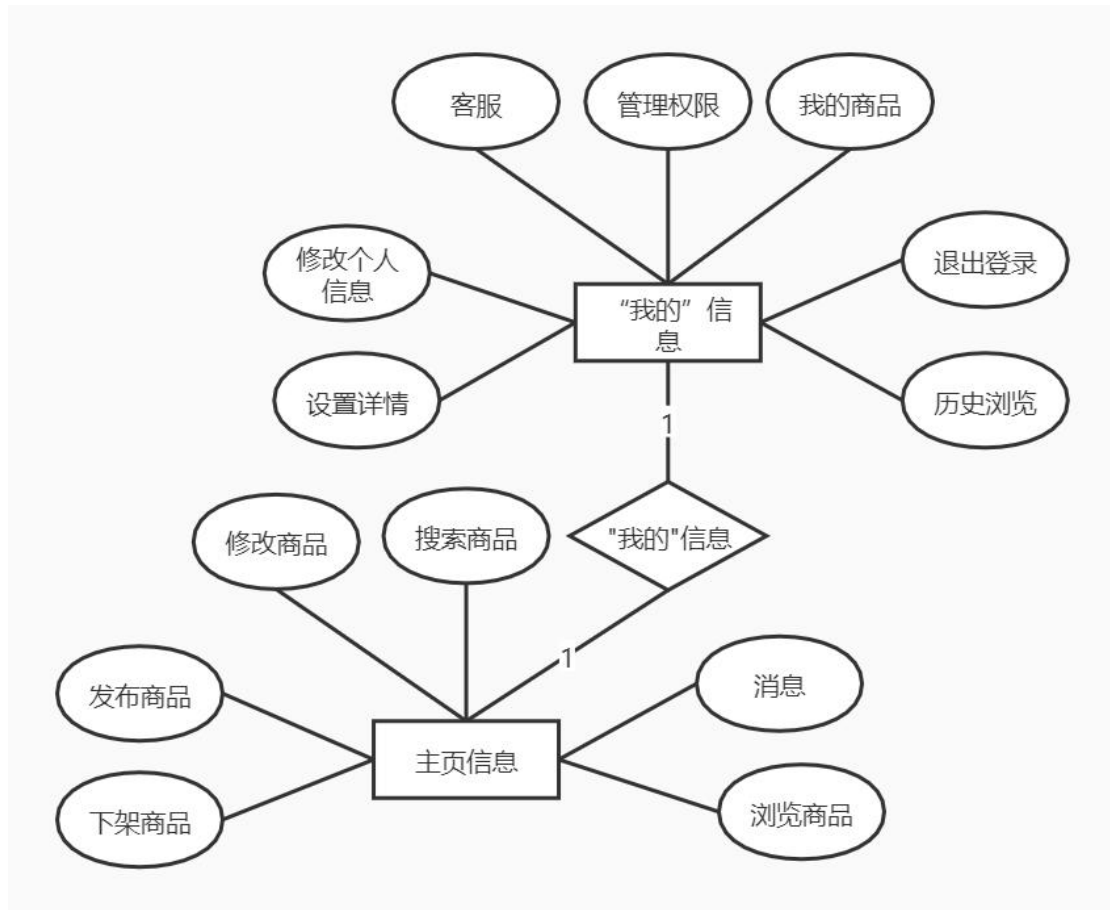


3.1.2 局部 E-R

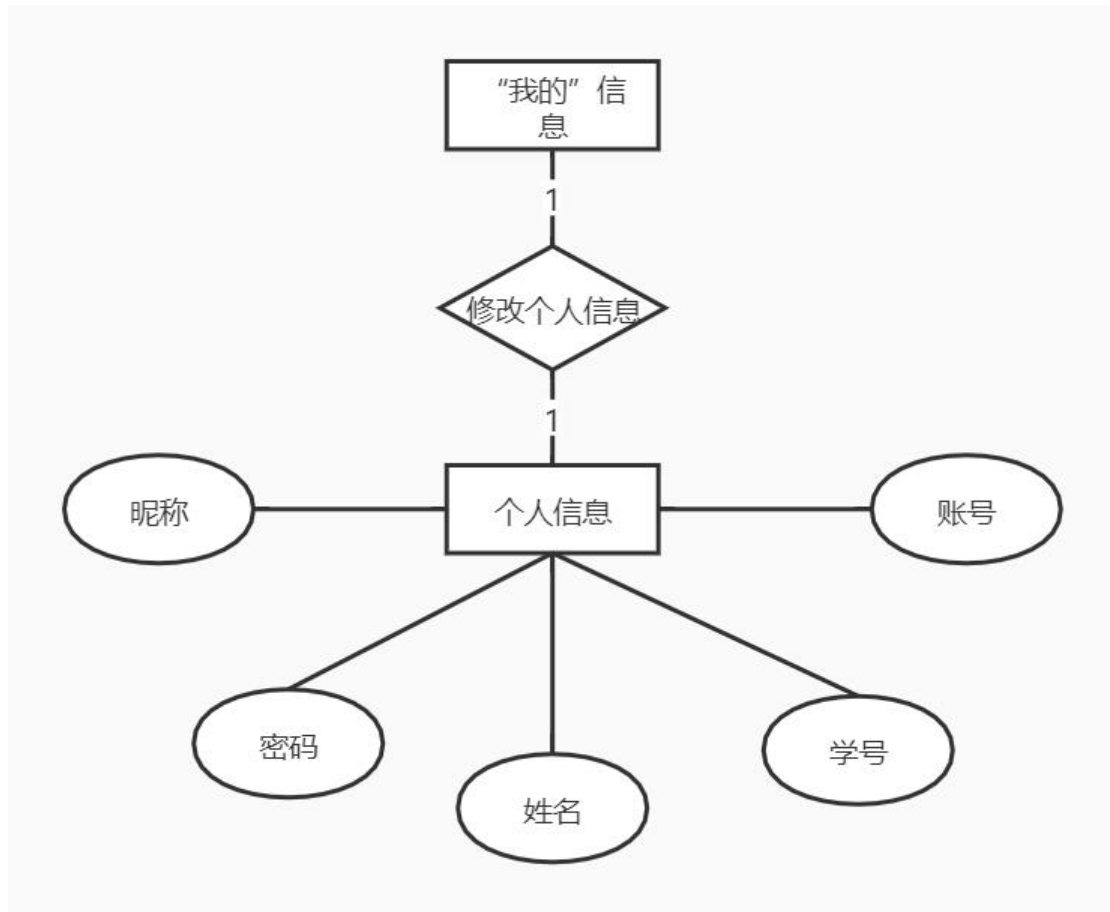
3.1.2.1 登录局部 E-R



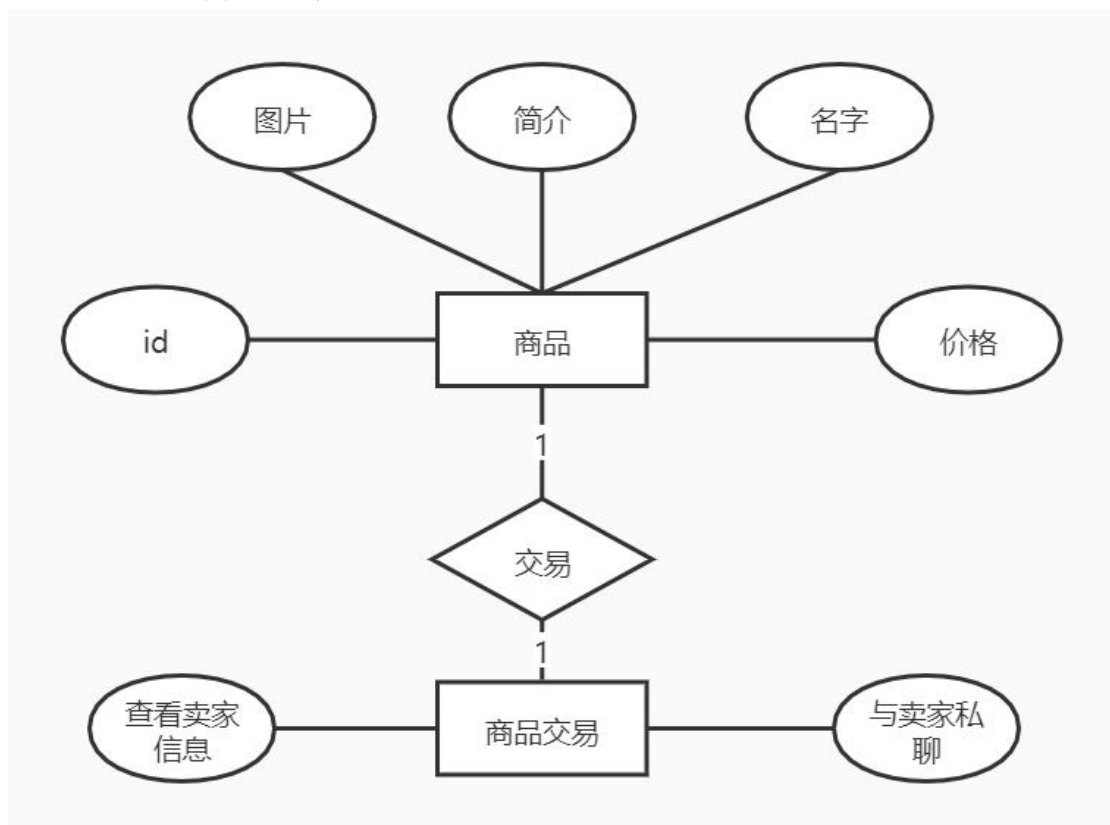
3.1.2.2 界面信息局部 E-R



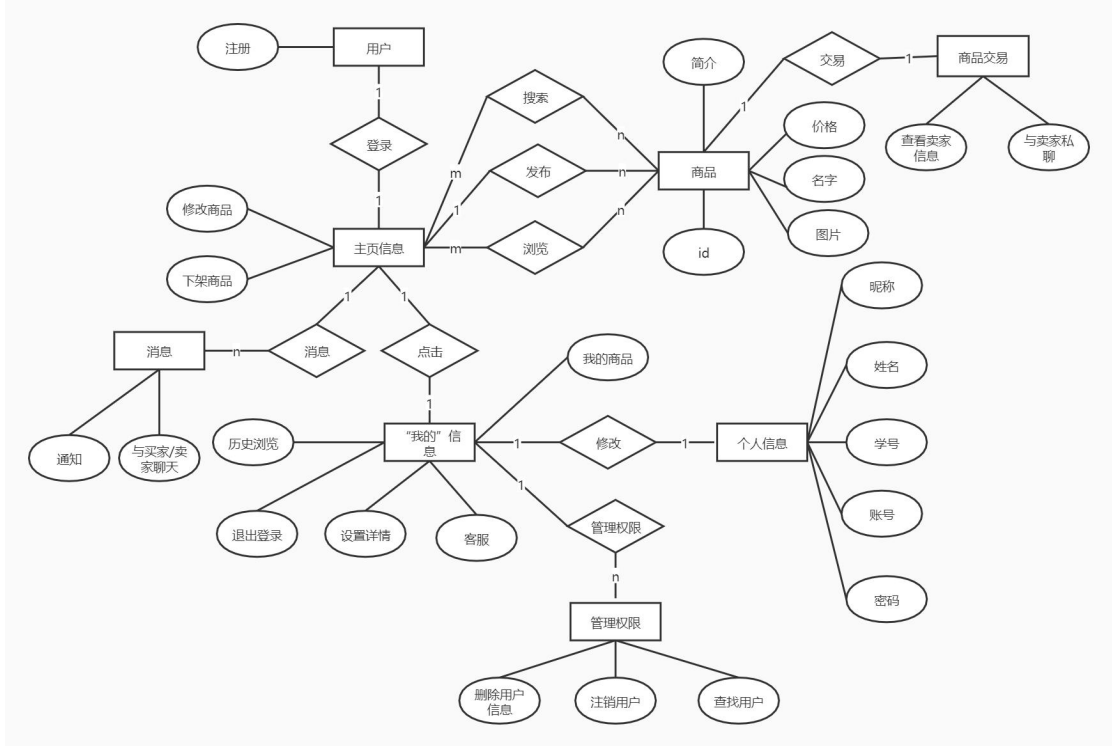
3.1.2.3 修改个人信息局部 E-R



3.1.2.4 商品局部 E-R



3.1.3 全局 E-R



3.2 逻辑结构设计

3.2.1 模式

1. 具体设计

表名	功能说明
STUDENT	学生表，用于验证是否是学生
USER_INFO	用户信息表
COMMODITY	商品表，存放商品信息

1.1 STUDENT 表（学生表）

表名	STUDENT			
列名	数据类型（精度范围）	空/非空	约束条件	其他说明
STUDENT_ID	VARCHAR(22)	非空	PRIMARY KEY	学号
STUDENT_NAME	VARCHAR(40)	非空		学生姓名
IS_REGISTER	VARCHAR(22)			是否已注册（没注册为空，已注册为用户账号）

1.2 USER_INFO 表（用户信息表）

表名	USER_INFO			
列名	数据类型（精度范围）	空/非空	约束条件	其他说明

	围)			
USER_ID	VARCHAR(22)	非空	PRIMARY KEY	账号
USER_NAME	VARCHAR(40)	非空	UNIQUE	用户名
STUDENT_ID	VARCHAR(40)	非空		学号
USER_PASSWORD	VARCHAR(40)	非空		密码
USER_PICTURE	IMAGE			头像(空的话, 用默认头像)

1.3 COMMODITY 表 (商品表)

表名	COMMODITY			
列名	数据类型 (精度范围)	空/非空	约束条件	其他说明
COMMODITY_ID	VARCHAR(20)	非空	PRIMARY KEY	商品号
USER_ID	VARCHAR(22)	非空		用户账号
COMMODITY_NAME	VARCHAR(40)	非空		商品名
COMMODITY_INFO	VARCHAR(400)	非空		商品信息
COMMODITY_PRICE	INT	非空		商品价格
COMMODITY_PICTURE	IMAGE	非空		商品图片

3.2.2 外模式

3.3 物理结构设计

数据库名称: CAMPUS_TRANSACTION_SQL

存储位置: 默认位置

建立系统程序员视图, 包括:

- 数据在内存中的安排, 包括对索引区、缓冲区的设计;
- 所使用的外存设备及外存空间的组织, 包括索引区、数据块的组织与划分;
- 访问数据的方式方法

第四章 运用设计

4.1 数据字典设计

无说明。

4.2 安全保密设计

通过区分不同的访问者、不同的访问类型和不同的数据对象，进行分别对待而获得的数据库安全保密设计考虑。对数据库设计中涉及到的各种项目，如数据项、记录、系、文卷、模式、子模式等一般要建立起数据保护措施，以说明它的标识符、同义名及有关信息。数据库由专门数据库管理用员对数据库操作，需要注意以下几项安全问题：

访问安全 、网络安全 、传输安全 、备份安全 、数据安全

4.3 数据库设计

4.3.1 创建表

```
/*=====*/
/* DBMS name:      MySQL 5.0                      */
/* Created on:      2020/4/19 23:26:26              */
/*=====*/

drop table if exists COMMODITY;

drop table if exists STUDENT;

drop table if exists USER_INFO;

/*=====*/
/* Table: COMMODITY                                */
/*=====*/
create table COMMODITY
(
    COMMODITY_ID      VARCHAR(20) not null,
    USER_ID           VARCHAR(22) not null,
    COMMODITY_NAME     VARCHAR(40) not null,
    COMMODITY_INFO     VARCHAR(400) not null,
    COMMODITY_PRICE    INT not null,
    COMMODITY_PICTRUE  IMAGE not null,
    primary key (COMMODITY_ID)
);
```

```

/*=====*/
/* Table: STUDENT */
/*=====*/
create table STUDENT
(
    STUDENT_ID          VARCHAR(22) not null,
    STUDENT_NAME        VARCHAR(40) not null,
    IS_REGISTER         VARCHAR(22),
    primary key (STUDENT_ID)
);

/*=====*/
/* Table: USER_INFO */
/*=====*/
create table USER_INFO
(
    USER_ID            VARCHAR(22) not null,
    USER_NAME          VARCHAR(40) not null,
    USER_PASSWORD      VARCHAR(40) not null,
    USER_PICTURE       IMAGE,
    primary key (USER_ID),
    unique key UNQ_USER_INFO_USER_NAME (USER_NAME)
);

alter table COMMODITY add constraint FK_USERINFO_COMMODITY foreign key (USER_ID)
references USER_INFO (USER_ID) on delete restrict on update restrict;

```