

MyBookDB

Project for 数据库及实现。

Github: <https://github.com/Super-Dainiu/DATA130039.01-MyBookDB>

MyBookDB

Versions

Compatibility

Installation

Contributors

Change Log

- Beta 0.1.0 User Login (May 29, 2021)
- Beta 0.1.1 Fixes (May 30, 2021)
- Beta 0.2.0 New Pages (June 1, 2021)
- Beta 0.2.1 User Management (June 2, 2021)
- Beta 0.2.2 User Edit (June 4, 2021)
- Beta 0.2.3 Important Fixes (June 5, 2021)
- Beta 0.3.0 Publisher Management (June 9, 2021)
- Beta 0.3.1 Writer Management (June 8, 2021)
- Beta 0.3.2 Data.py (June 16, 2021)
- Beta 0.3.3 Book Management (June 18, 2021)
- V 1.0.0 Launch Project (June 19, 2021)
- V 1.0.1 Add Index to Books (June 20, 2021)
- V 1.1.0 Orders Simulation (June 22, 2021)
- V 1.1.1 Orders Management (June 23, 2021)

Database Schema

- Login_user
- Users_user
- Publishers_publishers
- Writers_writers
- Books_books
- Books_books_writers
- Books_classification
- Books_classificationsub
- Orders_orders
- Orders_details

URL paths

Design Documents

- 1 系统需求分析
 - 1.1 用户需求分析
- 2 模型图
 - 2.1 数据流图
 - 2.2 数据字典
 - 2.3 E-R图
 - 2.3.1 会员信息
 - 2.3.2 图书信息
 - 2.3.3 订单信息
 - 2.3.4 整体E-R图
- 3 功能划分
 - 3.1 管理员登录、注册功能
 - 3.2 检索功能
 - 3.3 修改/删除功能
 - 3.4 订单操作

Versions

Artifact	Version
MySQL	8.0.23
Python	3.7.10
Navicat Premium	15.0.023
Django	3.2.3
Pymysql	1.0.2
Bootstrap	4.5

Compatibility

MyBookDB works on Windows 10. Chrome is recommended to have the best experience.

Installation

1. Create virtualenv (Recommended), choose the corresponding interpreter.
2. Git clone through Git CMD or download the project through Github Desktop to your direction:

```
git clone https://github.com/Super-Dainiu/DATA130039.01-MyBookDB
# Following message will occur.
```

```
Cloning into 'DATA130039.01-MyBookDB'...
remote: Enumerating objects: 846, done.
remote: Counting objects: 100% (846/846), done.
remote: Compressing objects: 100% (551/551), done.
Receiving objects: 100% (846/846), 91.00 MiB | 2.43 MiB/s, done.
remote: Total 846 (delta 453), reused 661 (delta 281), pack-reused 0
Resolving deltas: 100% (453/453), done.
```

3. Enter the root file "DATA130039.01-MyBookDB/" and install dependencies:

```
pip install -r requirements.txt
```

4. Set up database and replace my username and password in project root file "MyBookDB/settings.py" containing something like:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'MyBookDB',
        'USER': username,
        'PASSWORD': password,
        'HOST': '127.0.0.1',
        'PORT': '3306'
    }
}
```

5. Apply migrations and create superuser:

```
python manage.py migrate
python manage.py createsuperuser # Not necessarily required.
```

6. To avoid starting the system with empty database, run in command line:

```
python data.py
```

7. Start server at local host:

```
python manage.py runserver # Following message will occur.
```

```
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 25, 2021 - 00:47:23
Django version 3.2.3, using settings 'MyBookDB.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

8. Enter server: <http://127.0.0.1:8000/>

9. Database being polluted, you may:

```
python manage.py flush
python data.py # This simply reset everything including your login info.
```

Contributors

- 邵彦骏：前端、后端、逻辑设计，完成了主页、管理员登录与注册、用户管理以及图书管理等页面，撰写期末报告。
- 张涵秋：完成了订单管理系统，负责期中报告视频。
- 刘洛侃：爬取豆瓣书籍，负责期中报告ppt。

Change Log

Beta 0.1.0 User Login (May 29, 2021)

- 使用Bootstrap框架和HTML完成了主页和管理员登录的前端设计。
- 使用Django框架基本实现了管理员注册、登录功能：
 - 管理员注册需要输入两次相同的密码，前端页面将会显示用户已注册/密码错误/不存在相应用户等错误提示。

Beta 0.1.1 Fixes (May 30, 2021)

- 修复了管理员输错密码后需要重新输入所有信息的BUG。
- 增加了管理员使用Email和用户名双重登录的可能性。
- 美化页面，为注册界面增加新配色。
- 添加Footer，提供Github项目的链接和作者信息。

Beta 0.2.0 New Pages (June 1, 2021)

- 使用Bootstrap框架和HTML完成了base.html的前端设计：
 - 添加顶部Navbar，链接到Users, Writers, Publishers, Books, Orders五个功能页面。
 - 添加表格模板，预备实现信息检索与修改功能。
 - 添加Footer，提供Github项目的链接和作者信息。
- 设计MyBookDB的LOGO，并将其显示在顶部状态栏中。
- 使用babies-first-names-top-100-boys.csv生成了一些随机用户，以测试数据库后端模型。
- 加入了管理员登出的选项。

Beta 0.2.1 User Management (June 2, 2021)

- 基本实现管理员对当前系统中的搜索和展示：
 - 可以按任意单项排序，最多支持两个单项上升/下降排序。
 - 可以使用姓名、邮箱、手机号码进行模糊搜索，或者按照ID进行精确搜索。
- 提升了系统的安全性，使非管理员不得通过修改URL的形式直接进入仅管理员可见页面。
- 提供了对用户的删除操作。

Beta 0.2.2 User Edit (June 4, 2021)

- 修复了Filter栏中用户无法清空选择的BUG。
- 使用Bootstrap框架和HTML完成了客户信息修改的前端设计。
- 基本实现管理员新建/编辑客户的功能：
 - 管理员必须键入某用户的全部信息，并且需要满足完整性约束。
 - 管理员输入用户信息，前端页面将会显示用户名/手机号/Email已存在等错误提示。
 - 管理员输入错误信息后，前端强制将输入信息改为修改前的信息。

Beta 0.2.3 Important Fixes (June 5, 2021)

- 修复了修改用户时唯一性约束施加在本用户上的严重BUG。
- 对管理员登录权限作了进一步提升，现在将记录管理员登录时的IP地址。

Beta 0.3.0 Publisher Management (June 9, 2021)

- 使用Bootstrap框架和HTML完成了出版商信息修改的前端设计。
- 基本实现管理员对当前系统中的搜索和展示：
 - 可以按任意单项排序，最多支持两个单项上升/下降排序。
 - 可以使用出版商名称、邮箱、手机号码进行模糊搜索，或者按照ID进行精确搜索。
- 提供了对出版商的删除操作。
- 基本实现管理员新建/编辑出版商的功能：
 - 管理员必须键入某出版商的全部信息，并且需要满足完整性约束。
 - 管理员输入出版商信息，前端页面将会显示出版商名称/手机号/Email已存在等错误提示。
 - 管理员输入错误信息后，前端强制将输入信息改为修改前的信息。

Beta 0.3.1 Writer Management (June 8, 2021)

- 使用Bootstrap框架和HTML完成了作者信息修改的前端设计。
- 基本实现管理员对当前系统中的搜索和展示：
 - 可以按任意单项排序，最多支持两个单项上升/下降排序。
 - 可以使用作者名称、作者/译者分类进行模糊搜索，或者按照ID进行精确搜索。
- 提供了对作者的删除操作。
- 基本实现管理员新建/编辑作者的功能：
 - 管理员必须键入某作者的全部信息，并且需要满足完整性约束。
 - 管理员输入作者信息，前端页面将会显示作者/译者已存在等错误提示。
 - 同一个名称只能存在最多一位作者和一位译者。
 - 管理员输入错误信息后，前端强制将输入信息改为修改前的信息。

Beta 0.3.2 Data.py (June 16, 2021)

- 获取并整理豆瓣网爬虫数据，并处理图书分类信息。
- 建立图书-作者/译者 Many-to-Many Relationship，图书-类别，图书-子类，图书-出版社，子类-类别 Many-to-One Relationship。
- 创建data.py脚本，实现用户、作者、出版社、图书信息自动导入操作。

Beta 0.3.3 Book Management (June 18, 2021)

- 使用Bootstrap框架和HTML完成了图书信息修改的前端设计。
- 基本实现管理员对当前系统中的搜索和展示：
 - 可以按任意单项排序，最多支持两个单项上升/下降排序。
 - 可以使用图书名称、作者名称、分类信息、出版社信息进行模糊搜索，或者按照ID进行精确搜索。
- 提供了对图书的删除操作。
- 基本实现管理员新建/编辑图书的功能：
 - 管理员必须键入某图书的全部信息，并且需要满足完整性约束。
 - 管理员输入图书信息，前端页面将会显示图书已存在/作者/出版商不存在等错误提示。
 - 图书的存在当且仅当其作者/译者以及出版商存在。
 - 版本、出版商相同的同名图书将被认为是重复的错误信息。
 - 管理员输入错误信息后，前端强制将输入信息改为修改前的信息。
- 创建订单格式，建立图书-订单 Many-to-Many Relationship，订单-用户 Many-to-One Relationship。

V 1.0.0 Launch Project (June 19, 2021)

- 修复几处链接失效、显示错误、访问错误的BUG。
- 添加几处超链接，允许从Books界面跳转到Publishers和Writers界面对应的信息处。
- 添加home.html，提供模拟生成订单的接口。

V 1.0.1 Add Index to Books (June 20, 2021)

- 为Books_books表添加若干索引，依然不能解决/books页面访问较慢的问题。

V 1.1.0 Orders Simulation (June 22, 2021)

- 在/home中模拟生成订单：
 - 模拟生成若干订单，对应系统中的随机用户。
 - 订单号对应当前生成时间戳。
- 使用Bootstrap框架和HTML完成了/home的前端设计

V 1.1.1 Orders Management (June 23, 2021)

- 使用Bootstrap框架和HTML完成了订单信息修改的前端设计。
- 基本实现管理员对当前系统中的搜索和展示：
 - 可以按任意单项排序，最多支持两个单项上升/下降排序。
 - 可以使用管理员名称、完成情况、用户名称、创建时间进行模糊搜索，或者按照ID进行精确搜索。
- 基本实现管理员确认订单的功能：
 - 管理员确认订单当且仅当订单对应图书库存大于需求，如果不足会跳出alert(message)。
 - 确认订单后图书库存相应减少。
 - 管理员的姓名将被记录。

Database Schema

Login_user

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
username	varchar(40)	NO		NULL	
email	varchar(40)	NO		NULL	
pwd	varchar(128)	NO		NULL	
ip	varchar(40)	NO		NULL	

Users_user

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
name	varchar(40)	NO		NULL	
sex	varchar(6)	NO		NULL	
phone	varchar(20)	NO		NULL	
email	varchar(40)	NO		NULL	
vip	tinyint(1)	NO		NULL	
address	varchar(100)	NO		NULL	

Publishers_publishers

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
name	varchar(100)	NO		NULL	
phone_number	varchar(20)	NO		NULL	
email	varchar(254)	NO		NULL	
contacts	varchar(40)	NO		NULL	
address	varchar(60)	NO		NULL	

Writers_writers

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
name	varchar(40)	NO		NULL	
author_type	varchar(20)	NO		NULL	

Books_books

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
title	varchar(64)	NO	MUL	NULL	
price	decimal(6,2)	NO		NULL	
price_vip	decimal(6,2)	NO		NULL	
publish_date	date	NO		NULL	
edition	longtext	YES		NULL	
storage	int unsigned	NO		NULL	
classification_id	bigint	NO	MUL	NULL	
publishers_id	bigint	NO	MUL	NULL	
sub_classification_id	bigint	NO	MUL	NULL	

Books_books_writers

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
books_id	bigint	NO	MUL	NULL	
writers_id	bigint	NO	MUL	NULL	

Books_classification

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
class_name	varchar(20)	NO		NULL	

Books_classificationsub

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
class_name	varchar(20)	NO		NULL	
ancestor_class_name_id	bigint	NO	MUL	NULL	

Orders_orders

Field	Type	Null	Key	Default	Extra
id	varchar(64)	NO	PRI	NULL	
last_edit	datetime(6)	NO		NULL	
date	date	NO		NULL	
user_id	bigint	NO	MUL	NULL	

Orders_details

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
price	decimal(6,2)	NO		NULL	
count	int unsigned	NO		NULL	
book_id	bigint	NO	MUL	NULL	
order_id	varchar(64)	NO	MUL	NULL	

URL paths

The URL tree should be constructed as follows:

1. admin/
2. accounts/
3. home/
- 4.
5. users/
6. books/
7. publishers/
8. writers/
9. orders/

Design Documents

1 系统需求分析

1.1 用户需求分析

随着互联网技术的发展与普及，传统的线下图书销售已经无法满足当今人类快节奏、低成本的消费需求，诸如“当当网”、“孔夫子旧书网”等新模式的在线图书网站代替书城、书局等传统销售方式，逐渐步入现代市民的生活。在这样一个电子商务盛行的时代，如何扩大网上书城的知名度，提高客户量，创造可观的经济效益，不仅需要人性化的前端外观设计，同时也与后端高并发、系统化、模块化的数据库系统设计密不可分。因此，本数据库课程团队将会设计网上书店数据库MyBookDB，作为课程项目提交。

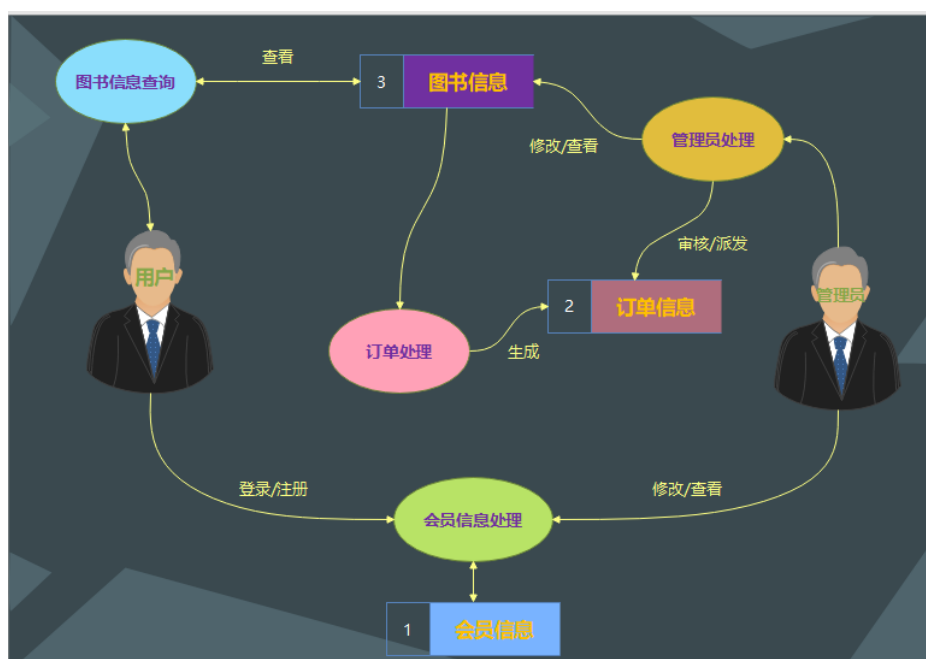
注意到作为信息管理系统，首先MyBookDB将会包含各类正式出版的图书信息（书号、书名、作者、定价、出版社、出版时间、版本号），其中，由于作者和译者可能由多人组成，因而还需要额外的作者-译者信息。为便于管理，图书还将被进行分类（如数学、外语、计算机等等），每一类图书下设子类（如计算机类又可以分编程语言、算法、网络等等）。此外，出版社的信息（编号、出版社名称、联系电话、联系人、e-mail、地址）也将作为重要信息储存。

为了便于后台程序管理，将会提供仅管理员可见的订单信息、销售记录（流水号、日期、会员编号、书号、价格、数量）等，还有独立的管理员账户，可以有权限对图书销售信息（定价、库存、会员）等进行修改。管理员可以审核处理订单信息，并且增删修改当前图书信息、库存信息。

用户具有独立的信息（用户名、联系电话、e-mail、地址），并且分为会员与非会员两种类型，以便会员能够享受其独有的优惠政策。用户提交的订单将交由后台管理员处理，他们只能购买有剩余库存的书目。

2 模型图

2.1 数据流图



2.2 数据字典

1. 会员

1. {用户名, 用户的唯一编号, char(30)}
2. {会员姓名, varchar(10)}
3. {性别, ENUM("男", "女", "保密")}
4. {手机号码, varchar(20)}
5. {email, varchar(30)}
6. {地址, varchar(50)}
7. {会员, ENUM("是", "否")}

2. 管理员

1. {用户名, 管理员的唯一编号, char(30)}
2. {email, varchar(30)}
3. {登陆密码, char(128)}

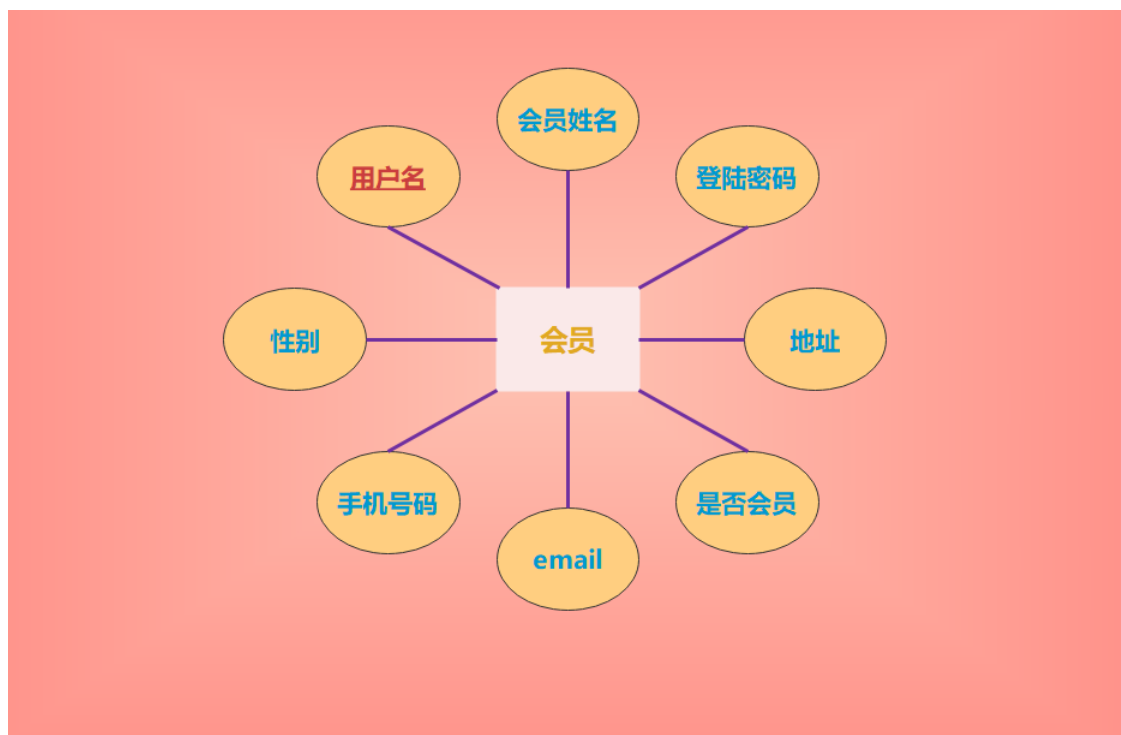
3. 订单信息

1. {订单号, 唯一确定订单的编号, char(10)}
2. {订单日期, DATETIME}
3. {收货人姓名, varchar(10)}

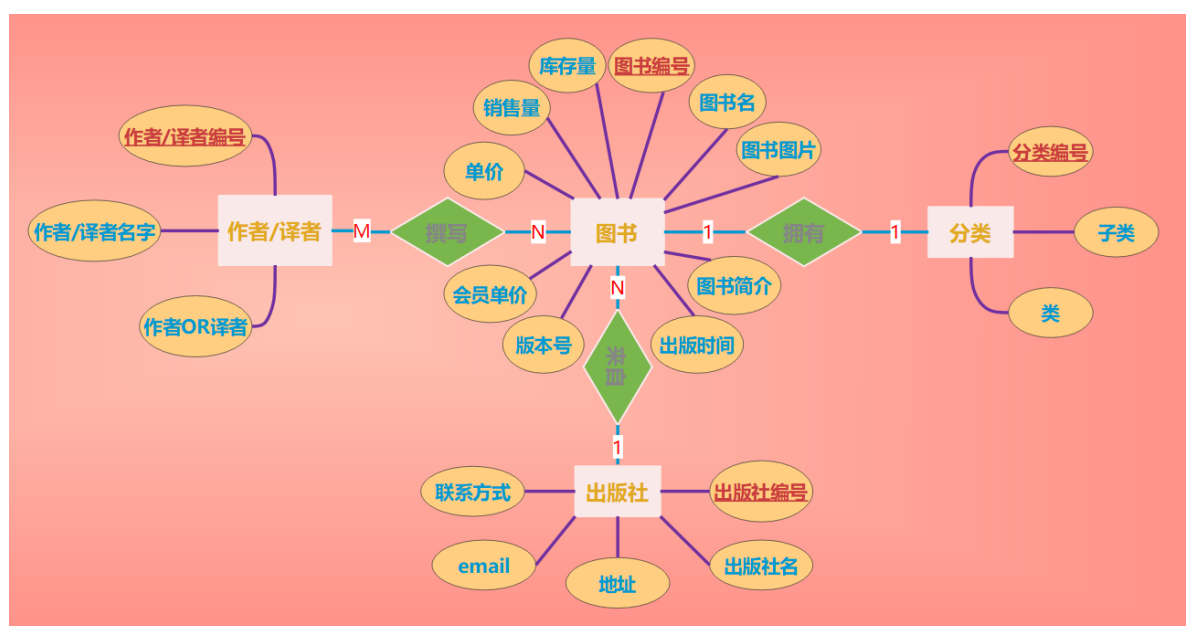
4. {发货状态, ENUM("未发货", "已发货")}
 5. {图书编号, char(20)}
 6. {订购数量, int(4)}
 7. {单价, int(4)}
4. 图书信息
1. {图书编号, 唯一标识图书的编号, char(20)}
 2. {图书名, varchar(30)}
 3. {出版时间, DATETIME}
 4. {版本号, char(30)}
 5. {出版社编号, int(4)}
 6. {作者/译者编号, int(4)}
 7. {图书类别编号, int(4)}
 8. {图书子类编号, int(4)}
 9. {库存量, int(4)}
 10. {单价, int(4)}
 11. {会员单价, int(4)}
5. 作者/译者
1. {作者/译者编号, 唯一标识作者/译者的编号, int(4), AUTO_INCREMENT}
 2. {作者/译者名字, varchar(30)}
 3. {作者/译者, SET("作者", "译者")}
6. 出版社信息
1. {出版社编号, 唯一标识出版社的编号, int(4), AUTO_INCREMENT}
 2. {出版社, varchar(30)}
 3. {出版社地址, varchar(50)}
 4. {出版社email, char(30)}
 5. {出版社联系方式, varchar(20)}
7. 分类信息1
1. {图书类别编号, 唯一标识分类的编号, int(4), AUTO_INCREMENT}
 2. {类别, varchar(10)}
8. 分类信息2
1. {图书子类编号, 唯一标识子类的编号, int(4), AUTO_INCREMENT}
 2. {子类, varchar(10)}

2.3 E-R图

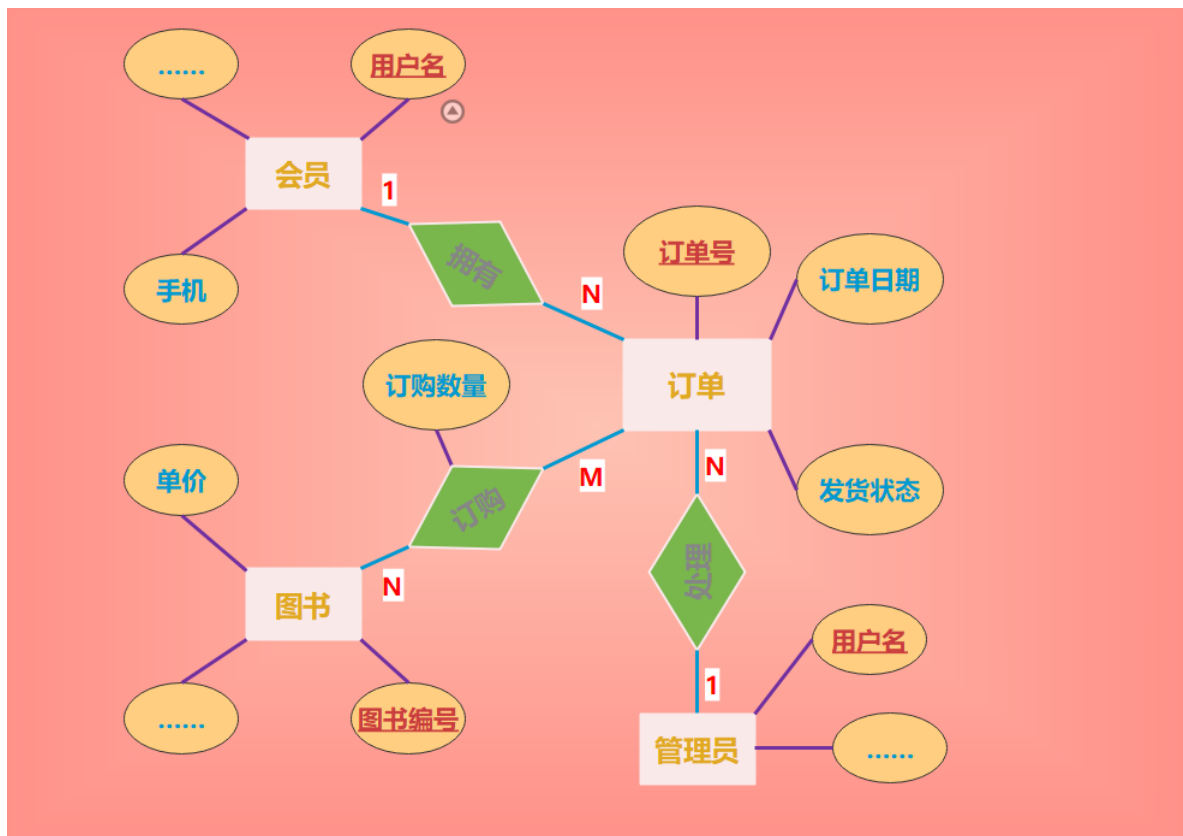
2.3.1 会员信息



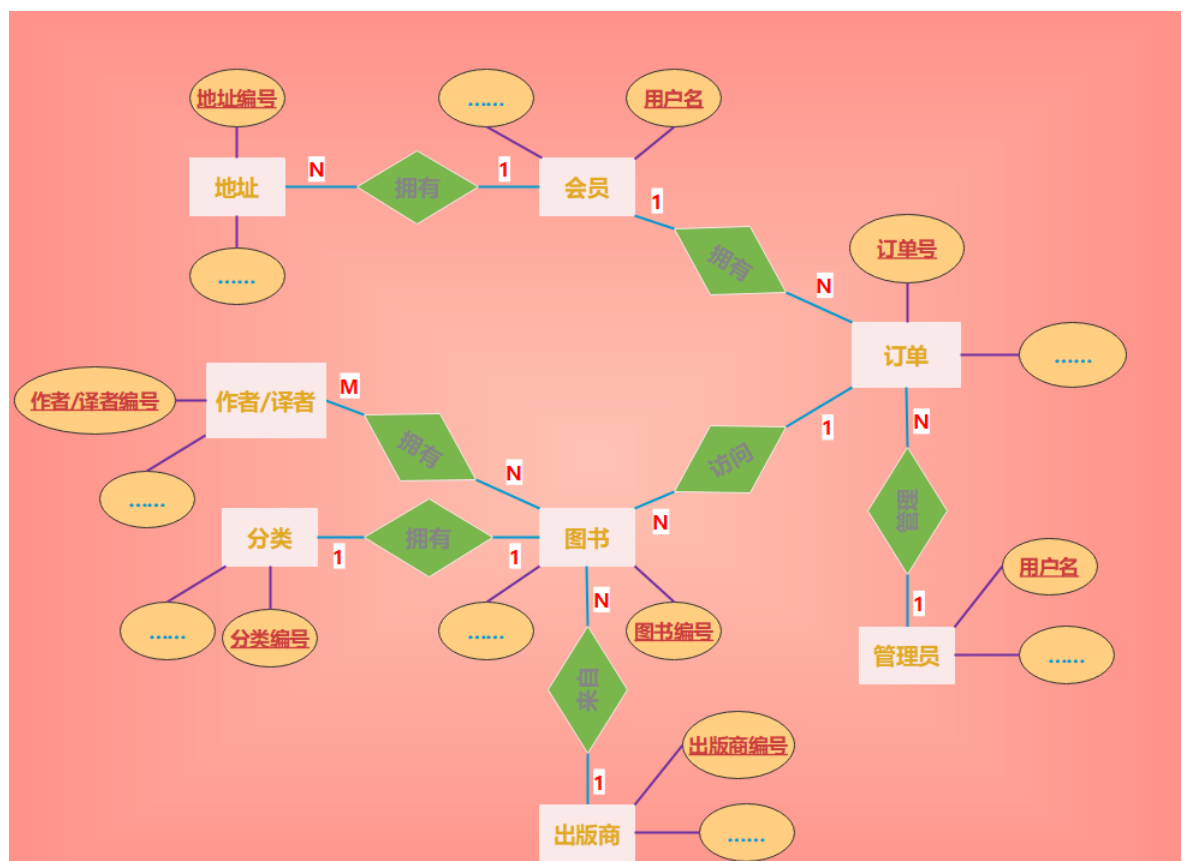
2.3.2 图书信息



2.3.3 订单信息



2.3.4 整体E-R图



3 功能划分

3.1 管理员登录、注册功能

管理员将可以自行录入，设置密码、用户名、email等个人信息，并且密码将使用加密算法，以提高系统的安全性等级。

3.2 检索功能

管理员将被允许使用过滤器、搜索栏，按照分类、关键词、作者、出版社等重要信息对图书进行检索，同时还可以按照销售量、访问量、价格等参数升序或降序排列。管理员还可以单独访问作者、译者以及出版社的具体信息，以便更好地查询信息。

3.3 修改/删除功能

管理员将被允许新增、删除、修改客户、图书、作者、出版社等信息。

3.4 订单操作

订单将会成交当且仅当管理员审核并通过订单，否则可能会导致恶意订单的提交。管理员还被允许根据实际情况增删库存信息以及图书信息。

