

# 如何使用 PowerDesigner 进行数据库设计

## 一 . 所需软件

- 1. PowerDesigner16.5
- 2. mysql-installer-community-5.7.13.0.msi

PowerDesigner 是收费的，但从网上可下载到破解版。MySQL 开发版本是免费的，可以官网上直接下载。

## 二 . 软件安装

把上述下载的软件进行安装，除了 mysql 安装成开发用途外，安装没有特殊要求。

## 三 . 数据库设计实例

下面设计一个简单的图书管理系统，讲解 powerdesigner 的使用方法。

### 3.1 图书管理系统的总体功能需求

本系统实现馆室管理、管理员管理、读者管理、图书管理、借阅管理五项功能。

### 3.2 图书管理系统的具体功能需求

#### 1. 馆室管理

不同的馆室用于分类存储图书, 比如 101 室存储外文类图书, 102 室存储计算机类图书。不同类的图书放于不同馆室内存放管理。馆室管理完成馆室信息的添加、删除、修改和查询。

#### 2. 管理员管理

管理员是指对整个图书管理系统进行管理维护的人员, 系统允许有多位管理员参与系统管理。每位管理员必须属于一个馆室且只能属于一个馆室, 每个馆室要求有至少一位管理员进行管理。管理员管理完成管理人员的添加、删除、修改和查询。

#### 3. 读者管理

每位读者必须有一个唯一的编号来标识不同的读者。读者编号可选用学号、身份证号或工号等具有唯一性的字符串。读者管理完成读者信息的添加、删除、修改和查询。

#### 4. 图书管理

完成图书信息的添加, 删除, 修改和查询。

#### 5. 借阅管理

完成读者借书, 读者还书, 超期扣费等操作。

### 3.3 数据字典

#### 1. 数据结构 (#代表主码)

编号	数据结构名	含义说明	组成
1	ReadingRoom	馆室表	#馆室号, 馆室名称, 馆室地址
2	Admin	管理员表	#管理员编号, 管理员名, 密码, 馆室号, 电话
3	Privilege	借阅权限表	#借阅权限, 可借图书数量, 最大借阅期限
4	ReaderInfo	读者表	#读者编号, 读者姓名, 所属单位, 密码, 注册时间, 借阅权限
5	Catagory	图书类别表	#类别编号, 类别名称

6	BookInfo	图书表	#图书编号, %类别编号, 书名, 作者, 出版社, 出版日期, 单价, %馆室号, 入库时间, 管理员编号
7	BorrowInfo	借阅表	#借阅编号, 读者编号, 图书编号, 借阅时间, 应还时间, 归还时间, 处理状态, 管理员编号
8	Charge	罚款表	#编号, 借阅编号, 罚款金额, 交款时间, 管理员编号

## 2. 数据结构间的关系

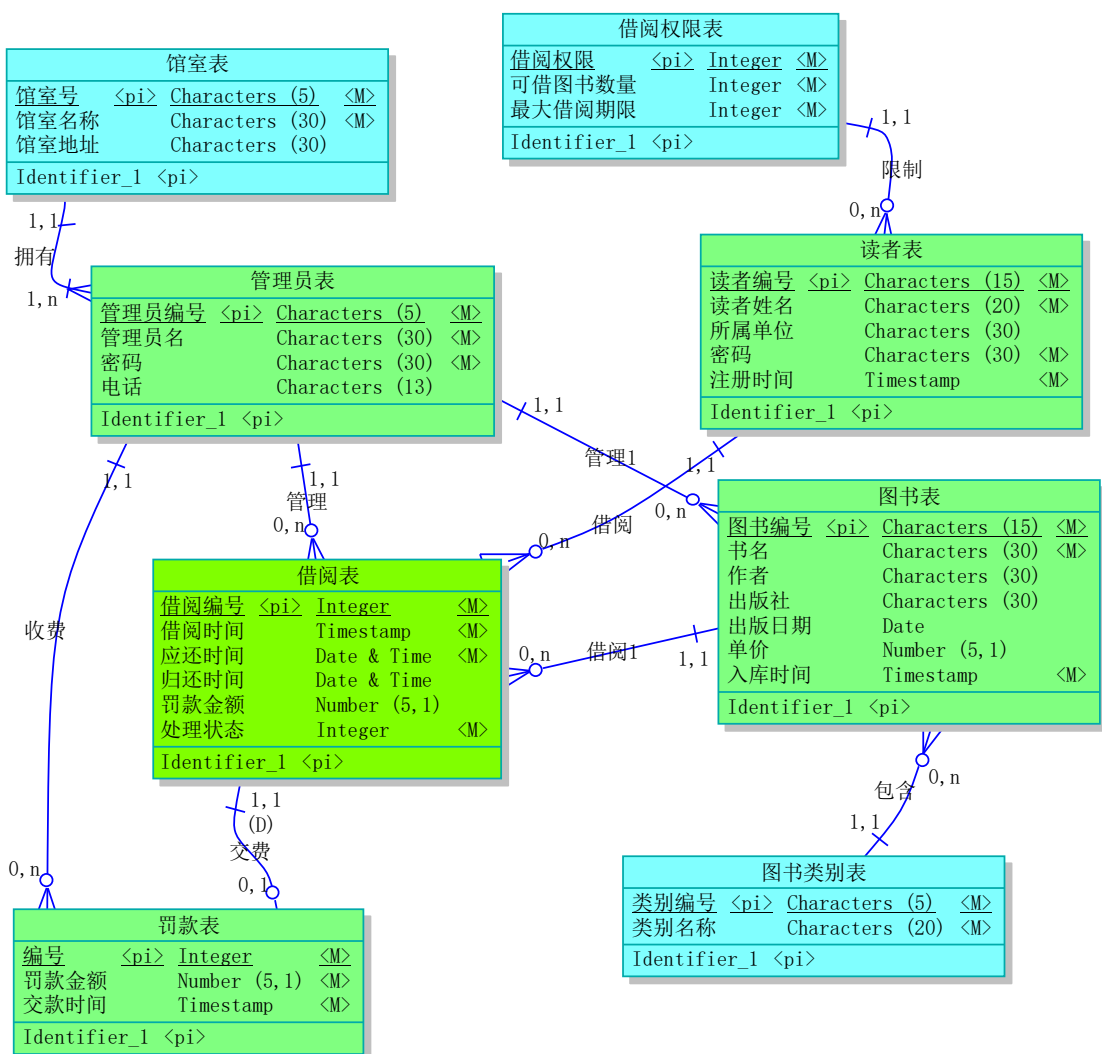
编号	数据结构 1	数据结构 2	关系类型
1	馆室表	管理员表	一对多
2	借阅权限表	读者表	一对多
3	管理员	图书表	一对多
4	管理员	借阅表	一对多
5	管理员	罚款表	一对多
6	读者表	图书表	多对多, 通过借阅表可拆分成两个一对多
7	图书类别表	图书表	一对多
8	借阅表	罚款表	一对一

## 3. 数据项

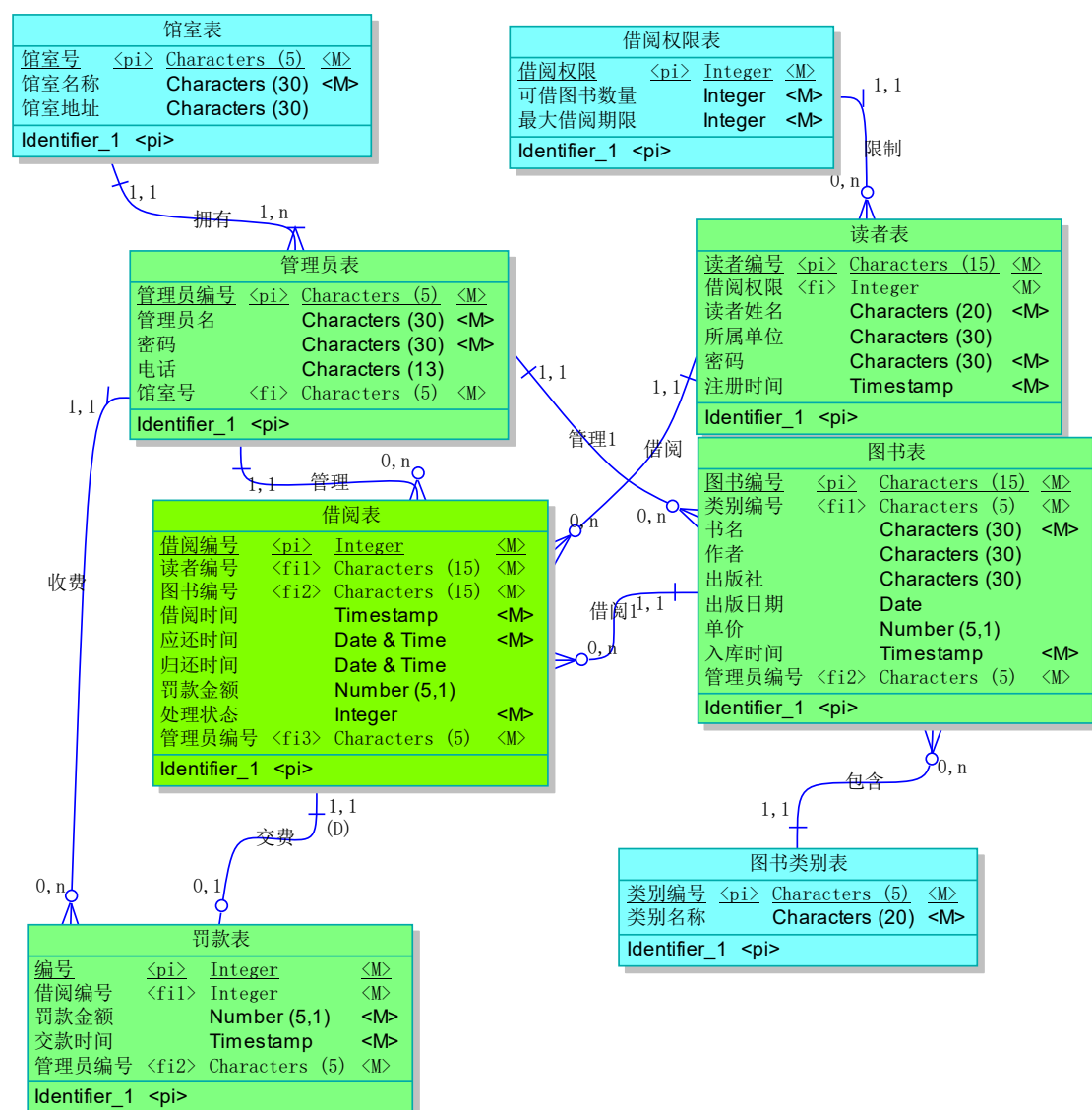
编号	别名	数据项名	存储结构	数据项含义
1	书名	BookName	char(30)	图书名称
2	交款时间	ExecTime	timestamp	图书超期交罚款的时间
3	作者	Author	char(30)	图书作者
4	借阅时间	Btime	timestamp	借书时间
5	借阅权限	Level	int	读者的借阅权限
6	借阅编号	BID	int	借阅表的编号
7	入库时间	EnterTime	timestamp	图书的入库时间
8	出版日期	Pubdate	date	图书的出版日期
9	出版社	Press	char(30)	图书的出版社
10	单价	Price	N(5,1)	图书的单价
11	可借图书数量	MaxBookNum	int	某权限的最大可借图书数量
12	图书编号	BookID	char(15)	图书编号
13	处理状态	State	int	借阅表中图书的处理状态
14	密码	ReaderPwd	char(30)	读者表中读者登录系统用的密码
15	密码	Pwd	char(30)	管理员登录系统用的密码
16	应还时间	Deadline	datetime	图书的要求归还时间
17	归还时间	Rtime	datetime	图书的实际归还时间
18	所属单位	Dept	char(30)	读者所属单位
19	最大借阅期限	MaxDays	int	某种借阅权限的最大借阅期限
20	注册时间	Regtime	timestamp	读者注册时间
21	电话	Phone	char(13)	管理员的联系电话

22	管理员名	AdminName	char(30)	管理员登录系统用的用户名
23	管理员编号	AdminID	char(5)	管理员编号
24	类别名称	CataName	char(20)	图书类别名称
25	类别编号	CatalD	char(5)	图书类别编号
26	编号	ID	int	罚款表编号
27	罚款金额	Amount	N(5,1)	罚款金额
28	读者姓名	ReaderName	char(20)	读者姓名
29	读者编号	ReaderID	char(15)	读者编号
30	馆室号	RRID	char(5)	馆室号
31	馆室名称	RRName	char(30)	馆室名称
32	馆室地址	Address	char(30)	馆室地址

#### 四．数据库的概念模型设计

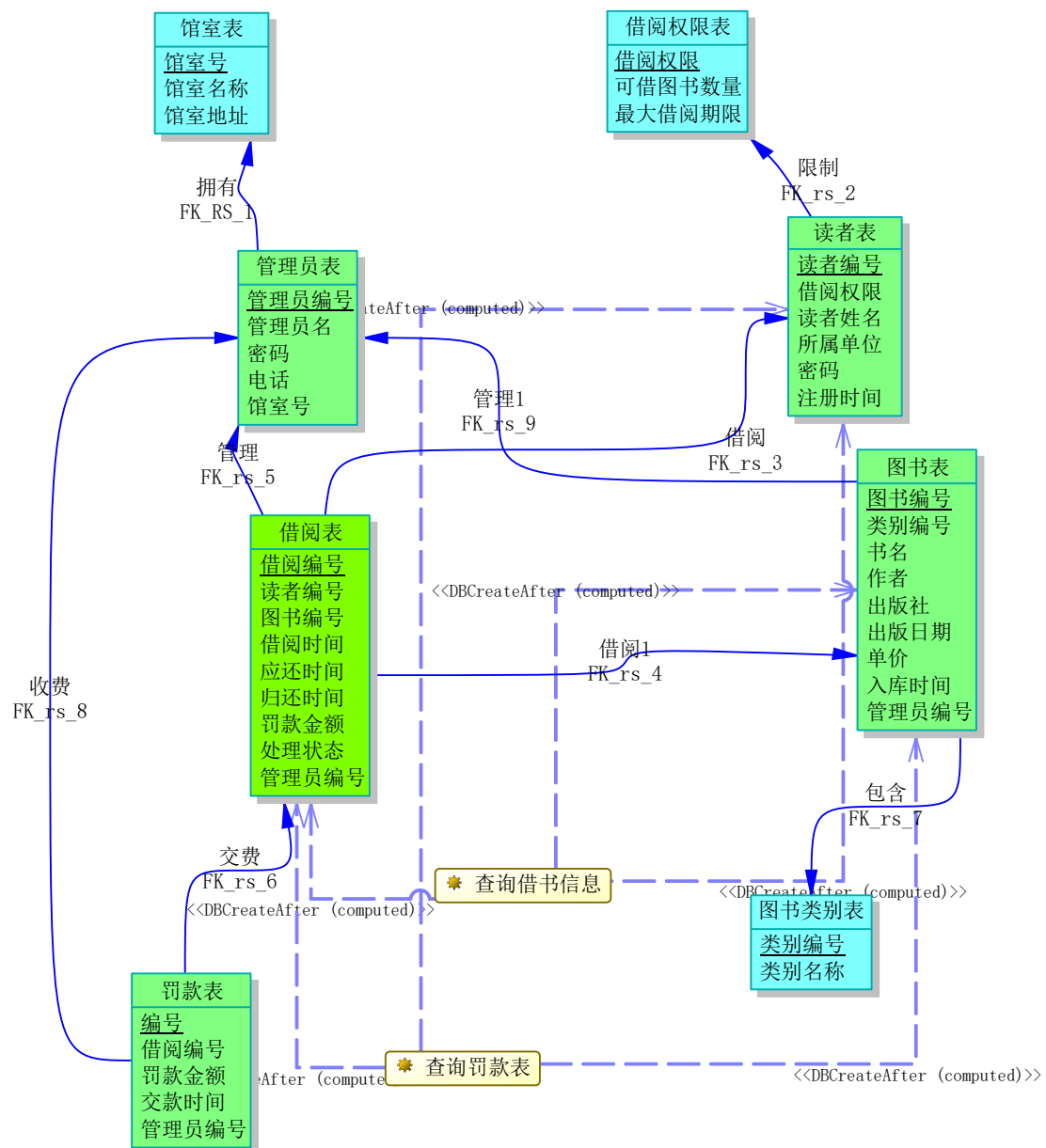


#### 五．数据库的逻辑模型设计



## 六．数据库物理模型设计

根据逻辑模型可以生成物理模型, 在生成物理模型之前, 需要选择所对应使用的 DBMS, 这里选择 mysql5.0。



为了简化编程，把借阅表中的“借阅编号”和罚款表中的“编号”字段设置为自动增加，另外为借阅表添加了一个删除触发器。为了演示如何添加存储过程，在图中添加了两个存储过程（查询借书信息和查询罚款表）。

## 七、数据库实施

根据上面生成的物理模型，可以直接生成数据库生成脚本文件。生成的脚本如下：

```
drop trigger Trigger_1;
```

```
drop procedure if exists Proc_BorrowInfo;
```

```
drop procedure if exists Proc_Penalty;
```

```
drop table if exists ADMIN;
```

drop table if exists BOOKINFO;

drop table if exists BORROWINFO;

drop table if exists CATAGORY;

drop table if exists CHARGE;

drop table if exists PRIVILEGE;

drop table if exists READERINFO;

drop table if exists READINGROOM;

```
/*=====
====*/
```

```
/* Table: ADMIN */
```

```
/*=====
====*/
```

create table ADMIN

```
(
    ADMINID          char(5) not null,
    ADMINNAME        char(30) not null,
    PWD              char(30) not null,
    PHONE            char(13),
    RRID             char(5) not null,
    primary key (ADMINID)
);
```

```
/*=====
====*/
```

```
/* Table: BOOKINFO */
```

```
/*=====
====*/
```

create table BOOKINFO

```
(
    BOOKID           char(15) not null,
    CATAID           char(5) not null,
    BOOKNAME         char(30) not null,
    AUTHOR           char(30),
    PRESS            char(30),
    PUBDATE          date,
    PRICE            numeric(5,1),
    ENTERTIME        timestamp not null,
```

```
        ADMINID          char(5) not null,  
        primary key (BOOKID)  
);
```

```
/*=====
```

```
====*/
```

```
/* Table: BORROWINFO                                     */
```

```
/*=====
```

```
====*/
```

```
create table BORROWINFO  
(  
    BID                int not null auto_increment,  
    READERID          char(15) not null,  
    BOOKID            char(15) not null,  
    BTIME             timestamp not null,  
    DEADLINE          datetime not null,  
    RTIME             datetime,  
    AMOUNT            numeric(5,1),  
    STATE             int not null,  
    ADMINID          char(5) not null,  
    primary key (BID)  
);
```

```
/*=====
```

```
====*/
```

```
/* Table: CATAGORY                                     */
```

```
/*=====
```

```
====*/
```

```
create table CATAGORY  
(  
    CATAID            char(5) not null,  
    CATANAME          char(20) not null,  
    primary key (CATAID)  
);
```

```
/*=====
```

```
====*/
```

```
/* Table: CHARGE                                     */
```

```
/*=====
```

```
====*/
```

```
create table CHARGE  
(  
    ID                int not null auto_increment,  
    BID              int not null,
```

```

        AMOUNT                numeric(5,1) not null,
        EXECTIME               timestamp not null,
        ADMINID                char(5) not null,
        primary key (ID)
    );

```

```

/*=====
====*/
/* Table: PRIVILEGE */
/*=====
====*/
create table PRIVILEGE
(
    LEVEL                int not null,
    MAXBOOKNUM           int not null,
    MAXDAYS              int not null,
    primary key (LEVEL)
);

```

alter table PRIVILEGE comment '注：借阅权限分为 1,2,3 三级，数值越大可借图书数量越多，借阅日期越长。';

```

/*=====
====*/
/* Table: READERINFO */
/*=====
====*/
create table READERINFO
(
    READERID             char(15) not null,
    LEVEL                int not null,
    READERNAME           char(20) not null,
    DEPT                 char(30),
    READERPWD            char(30) not null,
    REGTIME              timestamp not null,
    primary key (READERID)
);

```

```

/*=====
====*/
/* Table: READINGROOM */
/*=====
====*/
create table READINGROOM

```



```

(
    RRID                char(5) not null,
    RRNAME              char(30) not null,
    ADDRESS             char(30),
    primary key (RRID)
);

alter table ADMIN add constraint FK_RS_1 foreign key (RRID)
    references READINGROOM (RRID) on delete restrict on update restrict;

alter table BOOKINFO add constraint FK_rs_7 foreign key (CATAID)
    references CATAGORY (CATAID) on delete restrict on update restrict;

alter table BOOKINFO add constraint FK_rs_9 foreign key (ADMINID)
    references ADMIN (ADMINID) on delete restrict on update restrict;

alter table BORROWINFO add constraint FK_rs_3 foreign key (READERID)
    references READERINFO (READERID) on delete restrict on update restrict;

alter table BORROWINFO add constraint FK_rs_4 foreign key (BOOKID)
    references BOOKINFO (BOOKID) on delete restrict on update restrict;

alter table BORROWINFO add constraint FK_rs_5 foreign key (ADMINID)
    references ADMIN (ADMINID) on delete restrict on update restrict;

alter table CHARGE add constraint FK_rs_6 foreign key (BID)
    references BORROWINFO (BID) on delete restrict on update restrict;

alter table CHARGE add constraint FK_rs_8 foreign key (ADMINID)
    references ADMIN (ADMINID) on delete restrict on update restrict;

alter table READERINFO add constraint FK_rs_2 foreign key (LEVEL)
    references PRIVILEGE (LEVEL) on delete restrict on update restrict;

delimiter //
create procedure Proc_BorrowInfo (IN readerid char(15))
begin
select
a.readerid,b.readername,a.bookid,c.bookname,a.btime,a.deadline,a.rtime,a.amount,a.state
from borrowinfo a,readerinfo b,bookinfo c
where a.readerid=readerid and a.readerid=b.readerid and a.bookid=c.bookid and
isnull(a.rtime);
end//

```

```
delimiter //
create procedure Proc_Penalty (IN readerid char(15))
begin
select
a.readerid,b.readername,a.bookid,c.bookname,a.btime,a.deadline,a.rtime,a.amount,a.state
from borrowinfo a,readerinfo b,bookinfo c
where a.readerid=readerid and a.readerid=b.readerid and a.bookid=c.bookid and a.state=1;
end//
```

```
delimiter //
create trigger Trigger_1 before delete
on BORROWINFO for each row
begin
delete from charge where BID=old.BID;
end//
```

打开 mysql workbench 登进去，用命令 “create database test;” 命令创建一个测试数据库 test，然后用 “use test;” 命令把 test 数据库设置为当前数据库。再把上面生成的脚本复制到执行命令窗口中，然后执行生成所需的数据表、索引、存储过程和触发器。