



华中科技大学

数据库系统原理实践报告

综合设计：图书管理系统

姓 名：张鑫

学 院：计算机科学与技术

专 业：计算机科学与技术

班 级：1508

学 号：U201514699

指导教师：潘鹏

分数	
教师签名	

2018 年 6 月 28 日

任务书

图书管理系统：

假设图书馆的工作人员要处理下列日常工作：

- 借书：核实读者身份并检查是否存在下述情况：
 - 该读者借书的数额超标；
 - 该读者所借的书过期未还；
 - 该读者曾因借书过期被罚款而未交；如不存在上述情况，则登记借书信息；
- 还书：检查所还图书是否损坏或过期，是则登记罚单信息并打印罚单，在交纳罚金前，不允许该读者继续借书。若图书损坏，注销该图书信息，否则进行还书登记。
- 罚款：根据罚单收取罚金，同时取消该读者的借书限制。
- 图书信息维护：新书上架、旧书下架及图书信息查询。
- 读者信息维护：录入、注销、修改及查询读者信息。

此外，图书馆还应向读者提供下列基本功能：

- 查询图书信息；
- 查询自己的基本信息和借书记录；
- 续借；

设计一个 B/S 或 C/S 模式的系统实现上述功能。

目 录

目录

1	课程任务概述	1
2	软件功能学习部分	2
2.1	任务要求.....	2
2.2	完成过程.....	2
2.2.1	脱机备份.....	2
2.2.2	系统备份及恢复.....	4
2.2.3	新增用户.....	5
2.3	任务总结.....	8
3	SQL 练习部分	10
3.1	任务要求.....	10
3.1.1	建表.....	10
3.1.2	数据更新.....	11
3.1.3	查询.....	11
3.2	完成过程.....	12
3.2.1	建表.....	12
3.2.2	数据准备.....	13
3.2.3	数据更新.....	15
3.2.4	数据查询.....	19
3.3	任务总结.....	26
4	综合实践任务	27
4.1	系统设计目标.....	27
4.2	需求分析.....	27
4.3	总体设计.....	28
4.4	数据库设计.....	29
4.5	详细设计与实现.....	32
4.6	系统测试.....	42
4.7	系统设计与实现总结.....	56
5	课程总结	57
	附录.....	58

1 课程任务概述

实验主要分成三部分：软件功能学习部分、SQL 联系部分和综合实践部分。

- 1) 软件功能学习部分：实验制定使用 SQL server 数据库，该部分主要是熟悉该数据库的操作，包括创建用户，设置权限，备份数据和恢复数据等。
- 2) SQL 联系部分：用于熟悉 sql 语句的使用，所有的查询都基于与电影相关的关系，共 20 条查询语句，涵盖了大部分 sql 语法。
- 3) 综合实践部分：提供若干个题目供选择，如工资管理系统、航班管理系统等，每个系统都需要根据实际情况设计并创建若干个表，选用一种编程语言实现一个客户端/网页，要求数据库关系设计合理，客户端要有良好的人机交互界面。

2 软件功能学习部分

2.1 任务要求

- 1) 练习 sqlserver 的两种完全备份方式：数据和日志文件的脱机备份、系统的备份功能。
- 2) 练习在新增的数据库上增加用户并配置权限的操作。

2.2 完成过程

实验环境为 Windows10 系统下的 Microsoft SQL server2017 版本，使用 SQL server management studio 图形界面进行操作。

2.2.1 脱机备份

- 1) 要实现脱机备份，先要对数据库进行脱机，如图 2-1 所示，选择要备份的数据库，右键-任务-脱机，该数据库就和服务器断开了连接。

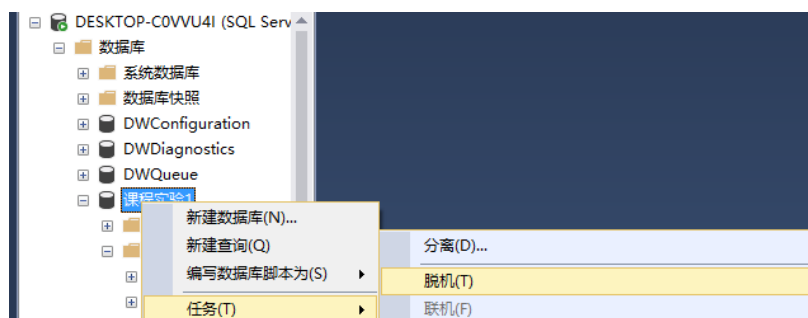


图 2-1 数据库脱机

脱机后的数据库图标上有红色的“×”，如图 2-2 所示。

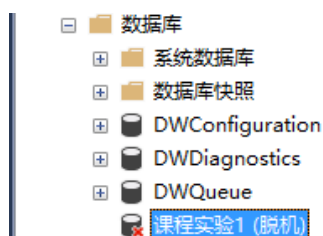


图 2-2 脱机的数据库

脱机以后，找到数据库所在的文件夹，将数据文件和日志文件拷贝，放到想要备份的位置，即实现了脱机备份，如图 2-3 所示。

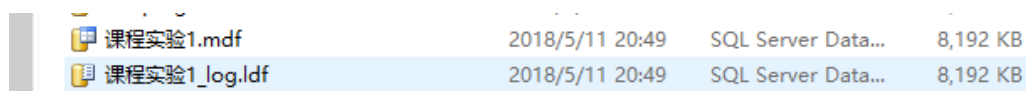


图 2-3 拷贝数据库

如果在复制前不脱机，则数据库是不可以拷贝的，如图 2-4 所示。

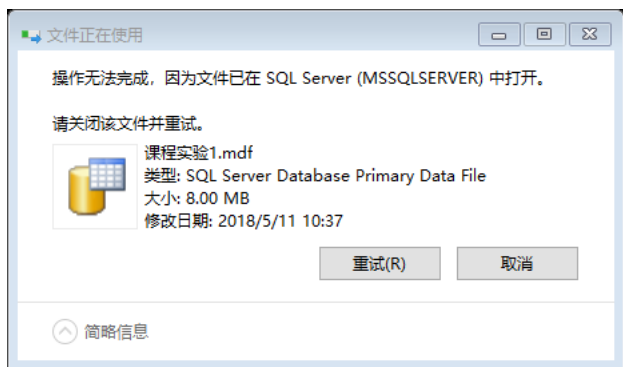


图 2-4 未脱机备份失败

2) 脱机备份以后的数据还可以恢复，如图 2-5 所示，在“数据库”选项上右键-附加，在弹出窗口中选择要恢复的数据库，也就是刚刚脱机备份的目标文件夹中的数据库。

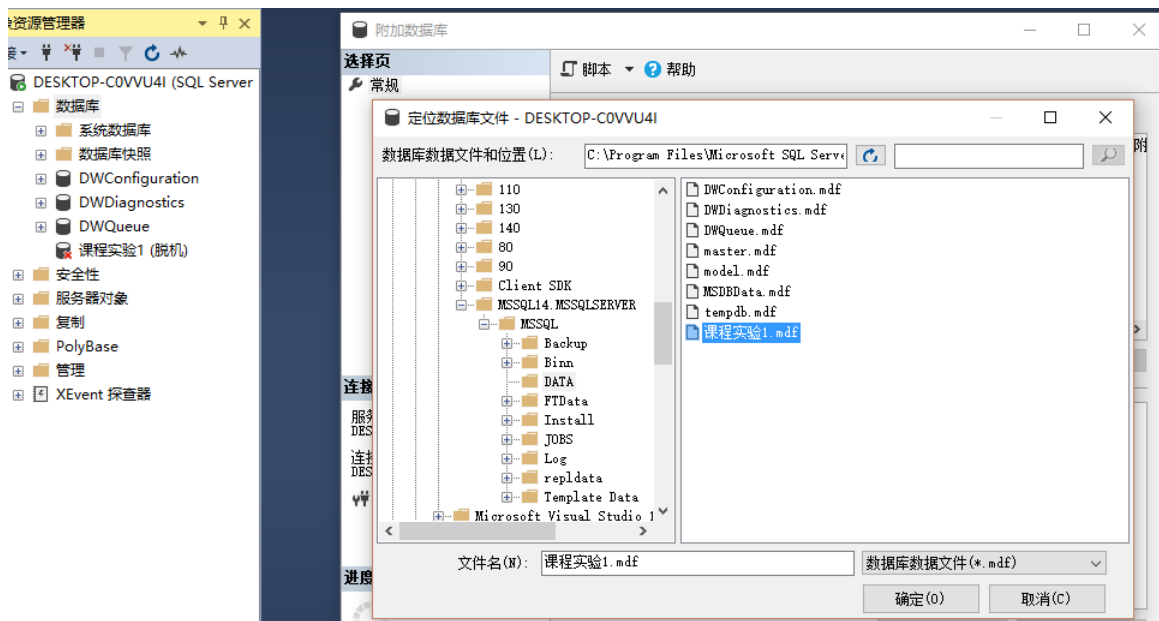


图 2-5 恢复脱机备份的数据库

选择数据库后，如图 2-6 所示，可以选择“恢复为……”，如果不修改名字，则会覆盖原有的数据库，这里我们修改一下，改为“课程实验 1-脱机恢复”。

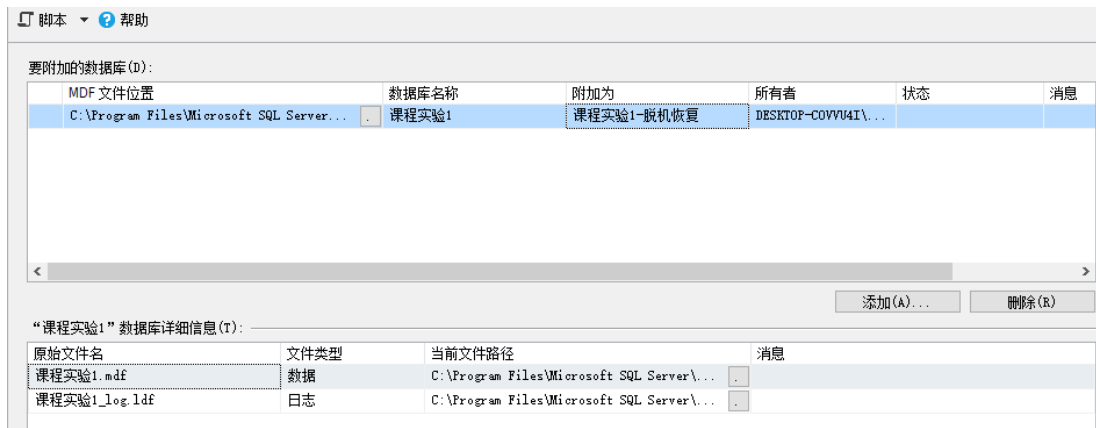


图 2-6 恢复选项

如图 2-7 所示，恢复以后的数据库和原有数据库并存。

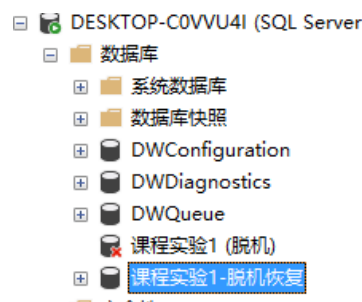


图 2-7 恢复数据库成功

2.2.2 系统备份及恢复

1) 选择“课程实验 1”数据库，右键-任务-备份，如图 2-8 所示。



图 2-8 开始备份数据库

弹出窗口如图 2-9 所示，默认备份路径在 C 盘，可以手动修改，也可以直接用该路径。点击确定，备份完成。



图 2-9 备份选项

备份完成后，在目标路径中能找到备份的数据库，如图 2-10 所示。



图 2-10 磁盘中的备份文件

- 2) 备份完成后还要恢复，选择该数据库，右键-任务-还原，弹出窗口如图 2-11 所示，一切都采用默认选项，直接点击确定，即可恢复数据。

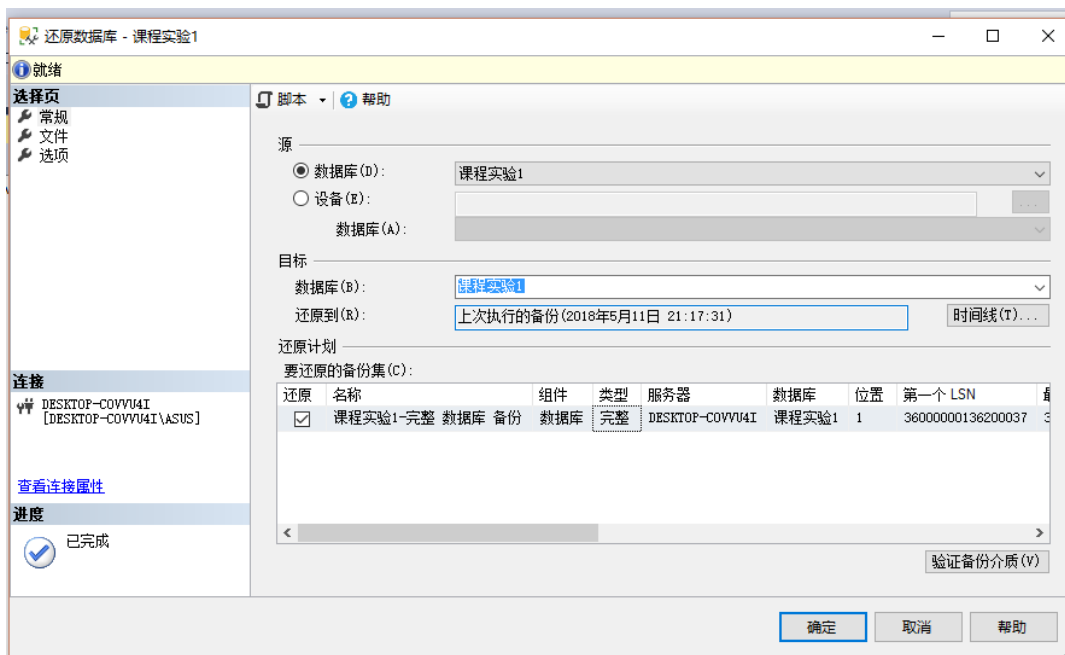


图 2-11 恢复数据库

2.2.3 新增用户

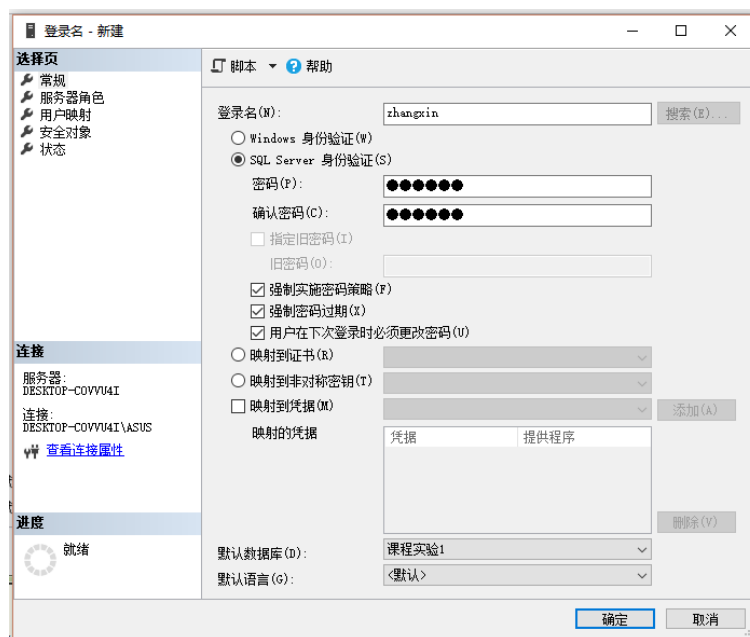


图 2-12 新增登录名 1

- 1) 用 Windows 身份验证方式登录，在 ssms 左侧的“安全性”选项右键->新建->登录名，即可新建一个登录用户，如图 2-12 所示。在“常规”选项卡中，

填写登录名，如“zhangxin”，验证方式选为 SQL server 身份验证，密码自设。在最下方的默认数据库一项，选择已经创建好的数据库“课程实验 1”，当然也可以选择别的数据库。

“服务器角色”选项卡采用默认设置，切换到“用户映射选项卡”，在“映射到此登录名的用户”中勾选“课程实验”数据库，默认架构为 dbo。其他设置不修改，如图 2-13 所示。

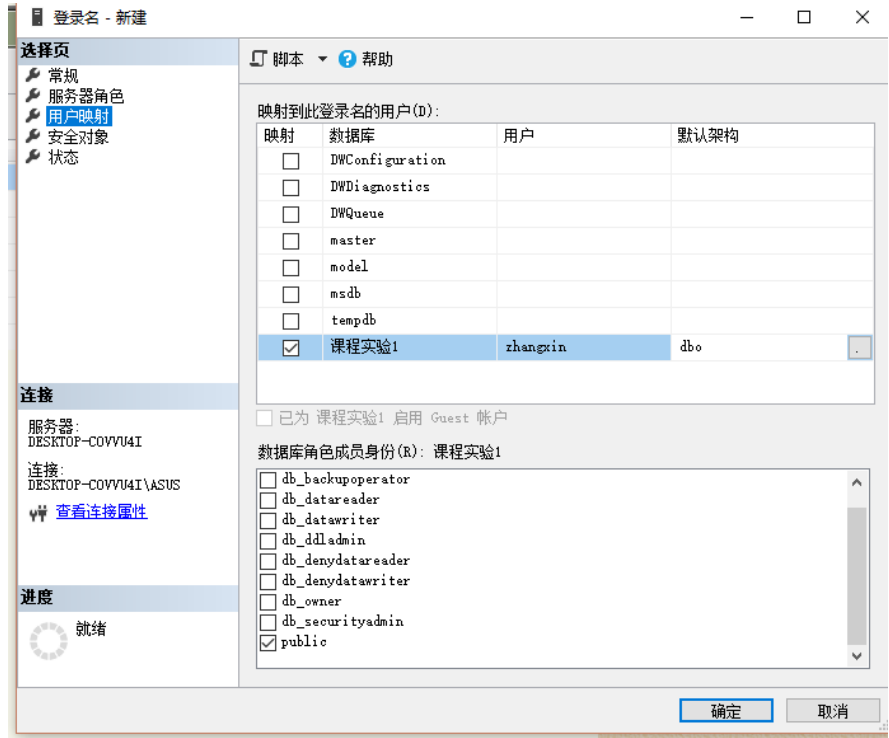


图 2-13 新增登录名 2

2) 创建完成后，断开连接，重新连接，如图 2-14 所示，用 SQL server 身份验证，填写刚刚新建的登录名和密码。

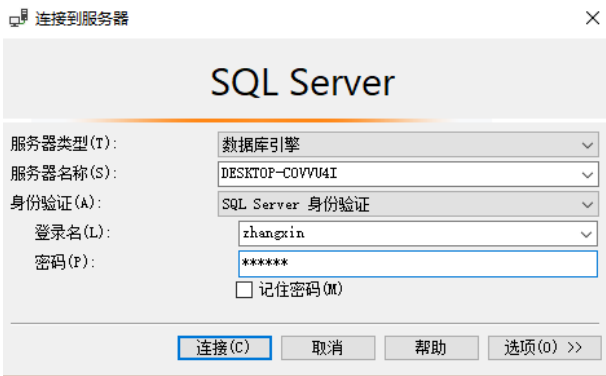


图 2-14 用新登录名登录

进入到“课程实验”数据库的“表”选项，发现没有可见的表，如图 2-15 所示。

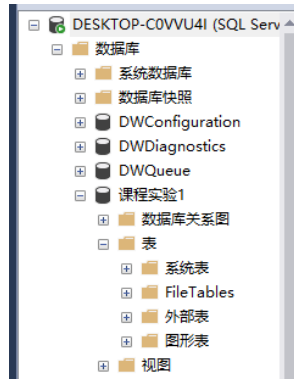
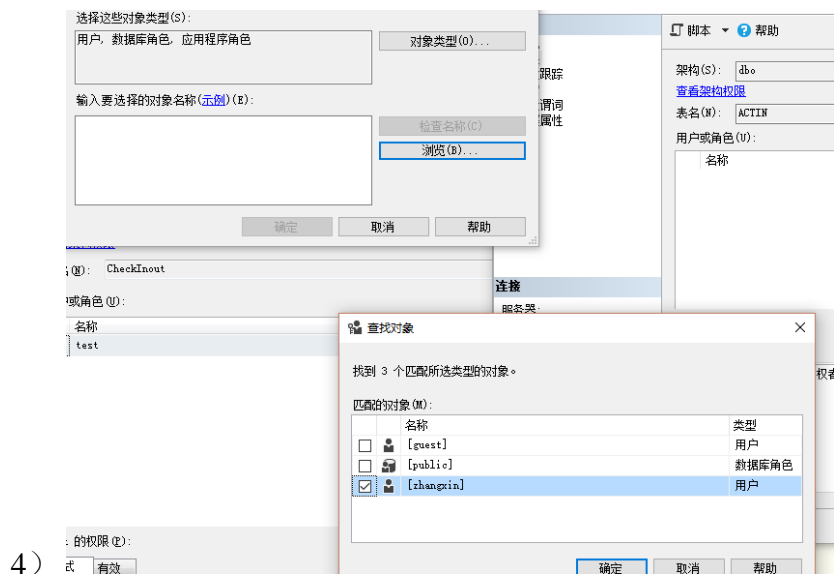


图 2-15 新用户不可查看数据库

- 3) 图 2-15 中没有可见的数据库是因为只在新建的用户一端设置了查看的数据库，没有在数据库的表上进行设置，默认是不可见的。



5) 图 2-16 在表上设置权限

如图 2-16 所示，用 Windows 验证方式重新连接，在需要查看的表上右键->属性,选择“权限”选项卡,在“用户或角色”一栏选择新建的登录名“zhangxin”。选择了用户以后，在改用户“zhangxin 的权限”一栏，勾选允许该用户进行的操作，如图 2-17 所示，允许该用户插入、修改、更新、删除和选择表。

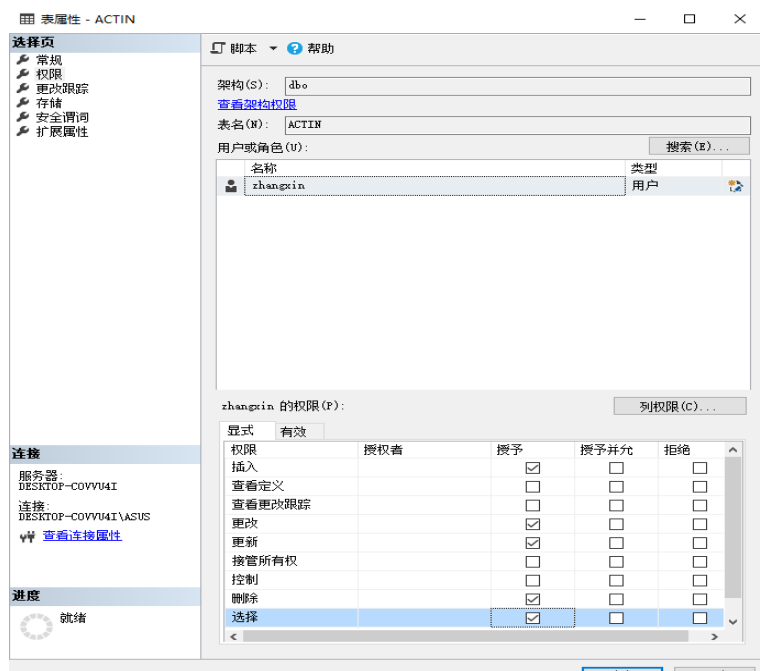


图 2-17 设置用户权限

设置完成后再用“zhangxin”进行连接，如图 2-18 所示，在“课程实验”数据库的“表”选项中多了一个表，就是我们刚刚设置为对该用户可见的表。

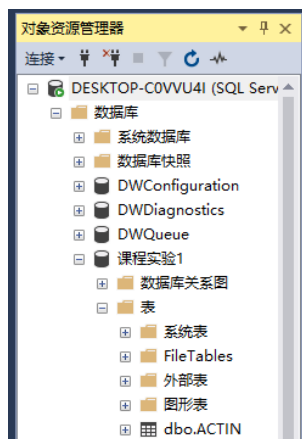


图 2-18 用户查看表

2.3 任务总结

1) 在创建完新的登录名后试图进行连接，出现错误，错误信息如图 2-19 所示。

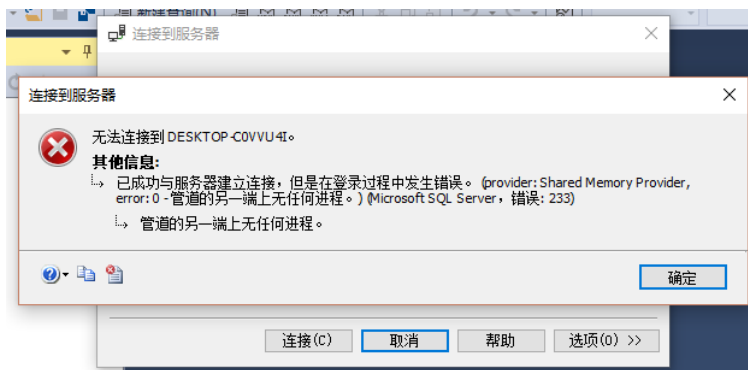


图 2-19 连接报错

查找资料得知这是因为没有启用数据库实例的 tcp/ip 协议，进入 SQL server configuration manager，如图 2-20 所示，在 mssqlserver 实例上右键启用该协议。



图 2-20 启用 tcp/ip 协议

- 2) 数据脱机备份时，将数据复制到了桌面，在恢复数据时，发现不能选择进入桌面，所以找不到该数据，原因是 SQL server 没有访问部分文件夹的权限。一开始想要获取权限，但是始终没有找到好的方法，最后只好将数据移动到可以访问的文件夹，才得以恢复。

3 SQL 练习部分

3.1 任务要求

3.1.1 建表

1) 创建下列跟电影相关的关系，包括主码和外码的说明

电影表【电影编号，电影名称，电影类型，导演姓名，电影时长（以分钟计），是否 3D，用户评分】

FILM(FID int, FNAME char(30), FTYPE char(10), DNAME char(30), length int, IS3D char(1), GRADE int)。

主码为电影编号，IS3D 取值为'Y'表示是 3D 电影，'N'表示不是，用户评分规定为 0~100 分之间或者为空值。

演员表【演员编号，演员姓名，性别，出生年份】

ACTOR(ACTID int, ANAME char(30), SEX char(2), BYEAR int)

主码为演员编号

参演表【演员编号，电影编号，是否主角，用户对该演员在该电影中的评分】

ACTIN(ACTID int, FID int, ISLEADING char(1), GRADE int)

主码、外码请依据应用背景合理定义。ISLEADING 取值为'Y'表示是，'N'表示不是主角，也可能取空值，表示不太确定该演员在该电影中是否主角。GRADE 规定为 0~100 分之间或者为空值。

电影院表【电影院编号，电影院名字，影院所在行政区，影院地址】

THEATER (TID int, TNAME char(20), TAREA char(20), ADDRESS char(30))

主码为电影院编号，影院所在行政区取值如“洪山区”、“武昌区”等等。

上映表【电影编号，影院编号，上映年份，上映月份】

SHOW(FID int, TID int, PRICE int, YEAR int, MONTH int)

假定一部电影在一家影院只上映一次，主码、外码请依据应用背景合理定义。

2) 观察性实验

验证在建立外码时是否一定要参考被参照关系的主码，并在实验报告中简述过程和结果。

3) 数据准备

依据后续实验的要求，向上述表格中录入适当数量的实验数据，从而对相关的实验任务能够起到验证的作用。

3.1.2 数据更新

- 1) 分别用一条 sql 语句完成对电影表基本的增、删、改的操作;
- 2) 批处理操作

将演员表中的 90 后演员记录插入到一个新表 YOUNG_ACTOR 中。

- 3) 数据导入导出

通过查阅 DBMS 资料学习数据导入导出功能, 并将任务 2.1 所建表格的数据导出到操作系统文件, 然后再将这些文件的数据导入到相应空表。

- 4) 观察性实验

建立一个关系, 但是不设置主码, 然后向该关系中插入重复元组, 然后观察在图形化交互界面中对已有数据进行删除和修改时所发生的现象。

- 5) 创建视图

创建一个有 80 后演员作主角的参演记录视图, 其中的属性包括: 演员编号、演员姓名、出生年份、作为主角参演的电影数量、这些电影的用户评分的最高分。

- 6) 触发器实验

编写一个触发器, 用于实现对**电影表**的完整性控制规则: 当增加一部电影时, 若导演的姓名为周星驰, 则电影类型自动设置为“喜剧”。

3.1.3 查询

请分别用一条 SQL 语句完成下列各个小题的查询需求:

- 1) 查询“战狼”这部电影在洪山区各家影院的 2017 年的上映情况, 并按照上映的月份的降序排列;
- 2) 查询所有无参演演员信息的电影的基本信息, 并且将结果按照电影类型的升序排列, 相同类型的电影则按照用户评分的降序排列;
- 3) 查询所有直到 2017 年仍未上映的电影编号、电影名称、导演姓名;
- 4) 查询在各家电影院均上映过的电影编号;
- 5) 查询所有用户评分低于 80 分或者高于 89 分的电影编号、电影名称、导演姓名及其用户评分, 要求 where 子句中只能有一个条件表达式;
- 6) 查询每个导演所执导的全部影片的最低和最高用户评分;
- 7) 查询至少执导过 2 部电影的导演姓名、执导电影数量;
- 8) 查询至少 2 部电影的用户评分超过 80 分的导演及其执导过的影片数量、平均用户评分;
- 9) 查询至少执导过 2 部电影的导演姓名以及跟这些导演合作过的演员编号、姓名;
- 10) 查询每个演员担任主角的电影中的平均用户评分;
- 11) 查询用户评分超过 90 分的电影的最早上映年月;

- 12) 查询用户评分超过 90 分的电影的最早上映年月及其相应的上映影院编号;
- 13) 查询每个电影的上映总次数;
- 14) 查询执导过动作片,或者警匪片,或者枪战片的导演的姓名,要求 where 子句中只能有一个条件表达式;
- 15) 查询所有“战狼”系列的电影的编号、电影名称、上映电影院名称及其上映年月,结果按照电影名称的升序排列;
- 16) 查询在同一个年月上映 1 号和 2 号电影的影院编号;
- 17) 查询所有没参演过用户评分 85 分以下电影的演员的编号、姓名;
- 18) 查询参演过“吴宇森”执导过的所有电影的演员姓名;
- 19) 查询所有的演员的编号、姓名及其参演过的电影名称,要求即使该演员未参演过任何电影也要能够输出其编号、姓名;
- 20) 查询所有上映超过 3 次但没有用户评分的电影编号、名称。

3.2 完成过程

3.2.1 建表

1) 电影表如图 3-1 所示;


 FID	int	<input type="checkbox"/>
FNAME	char(30)	<input checked="" type="checkbox"/>
FTYPE	char(10)	<input checked="" type="checkbox"/>
DNAME	char(30)	<input checked="" type="checkbox"/>
length	int	<input checked="" type="checkbox"/>
IS3D	char(1)	<input checked="" type="checkbox"/>
GRADE	int	<input checked="" type="checkbox"/>

图 3-1 电影表

2) 演员表如图 3-2 所示;


 ACTID	int	<input type="checkbox"/>
ANAME	char(30)	<input checked="" type="checkbox"/>
SEX	char(2)	<input checked="" type="checkbox"/>
BYEAR	int	<input checked="" type="checkbox"/>

图 3-2 演员表

3) 参演表如图 3-3 所示;



 ACTID	int	<input type="checkbox"/>
 FID	int	<input type="checkbox"/>
ISLEADING	char(1)	<input checked="" type="checkbox"/>
GRADE	int	<input checked="" type="checkbox"/>

图 3-3 参演表

4) 电影院表如图 3-4 所示;


 TID	int	<input type="checkbox"/>
TNAME	char(20)	<input checked="" type="checkbox"/>
TAREA	char(20)	<input checked="" type="checkbox"/>
ADDRESS	char(30)	<input checked="" type="checkbox"/>

图 3-4 电影院表

5) 上映表如图 3-5 所示。

FID	int	<input type="checkbox"/>
TID	int	<input type="checkbox"/>
PRICE	int	<input checked="" type="checkbox"/>
YEAR	int	<input checked="" type="checkbox"/>
MONTH	int	<input checked="" type="checkbox"/>

图 3-5 上映表

6) 观察性实验。在建表时，建立外码时不参照外表的主码，点击确定，如图 3-6 所示，提示约束不匹配，所以建立外码时必须引用某个表的主码。

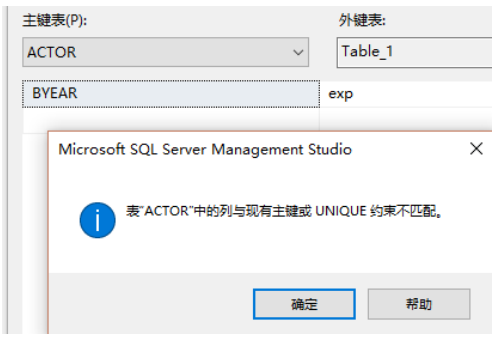


图 3-6 观察性实验

3.2.2 数据准备

1) 录入电影信息如图 3-7 所示；

	FID	FNAME	FTYPE	DNAME	length	IS3D	GRADE
▶	1	战狼2	... 枪战	吴京	... 155	Y	98
	2	猛虎英雄传	... 枪战	印度阿三	... 133	N	87
	3	美丽人生	... 战争	犹太	... 118	N	91
	4	未知死亡	... 爱情	印度阿三	... 185	Y	93
	5	爱丽丝梦游仙...	... 科幻	疯帽子	... 112	Y	89
	6	唐人街探案2	... 悬疑	陈思诚	... 164	Y	NULL
	7	钢铁侠	... 科幻	斯皮尔伯格	... 134	Y	96
	8	长江七号	... 喜剧	周星驰	... 105	N	88
	9	逃学威龙	... 喜剧	周星驰	... 113	N	72
	10	赌圣	... 喜剧	周星驰	... 132	N	99
	11	尖峰时刻	... 动作	拉特纳	... 128	N	94
	12	小萝莉的猴神...	... 宗教	卡比尔汗	... 170	Y	100
	13	战狼	... 枪战	吴京	... 112	Y	NULL
	14	唐人街探案	... 悬疑	陈思诚	... 109	Y	90
	15	铁汉柔情	... 动作	吴宇森	... 110	N	77
	16	谍中谍	... 悬疑	吴宇森	... 120	N	83
	17	sdaf	... 喜剧	周星驰	... NULL	NULL	NULL

图 3-7 电影信息数据

2) 录入演员信息如图 3-8 所示；

	ACTID	ANAME	SEX	BYEAR
▶	1	阿米尔汗	男	1965
	2	沙鲁克汗	男	1965
	3	萨尔曼汗	男	1965
	4	卢靖姗	女	1885
	5	刘承羽	女	1994
	6	吴京	男	1974
	7	余男	女	1976
	8	王宝强	男	1984
	9	刘昊然	男	1997
	10	吉雅可汗	女	1988
	11	成龙	男	1954
	12	李连杰	男	1963
	13	甄子丹	男	1963
	14	范冰冰	女	1981
	15	迪丽热巴	女	1992
	16	古力娜扎	女	1992

图 3-8 演员信息数据

3) 录入参演信息如图 3-9 所示；

	ACTID	FID	ISLEADING	GRADE
▶	1	3	N	68
	1	4	Y	96
	2	9	N	90
	3	2	Y	80
	3	9	N	90
	4	1	Y	99
	5	6	Y	90
	6	1	Y	85
	7	1	N	78
	8	6	Y	88
	8	15	NULL	NULL
	9	6	N	92
	10	4	Y	79
	10	5	Y	73
	11	15	NULL	NULL
	11	16	NULL	NULL
	12	15	NULL	NULL
	12	16	NULL	NULL

图 3-9 参演信息数据

4) 录入电影院信息如图 3-10 所示；

	TID	TNAME	TAREA	ADDRESS
▶	1	中影星美	宁乡	花明路
	2	巨幕	洪山区	光谷广场
	3	万达	开福区	八一路
	4	CGV	武昌	珞喻路
	5	华夏	望城	斑马路
	6	成龙	岳麓区	大学城
	7	天河	洪山区	鲁磨路
	8	格瓦拉	洪山区	民族大道
*	NULL	NULL	NULL	NULL

图 3-10 电影院信息数据

5) 录入上映信息如图 3-11 所示。

	FID	TID	PRICE	YEAR	MONTH
▶	1	1	38	2017	5
	1	2	38	2017	7
	1	3	38	2017	7
	1	4	45	2017	8
	1	5	60	2017	7
	1	6	38	2017	6
	1	7	27	2017	9
	1	8	33	2017	6
	2	3	8	2013	3
	2	4	40	2017	8
	2	8	14	2017	6
	3	2	25	2010	9
	3	6	30	2010	6
	4	1	20	2014	1
	6	1	35	2018	2
	6	2	35	2018	2
	6	3	37	2018	1
	6	4	28	2018	2
	12	5	24	2018	3
	13	2	22	2011	5
	13	4	27	2015	9
	13	6	30	2015	6
*	NULL	NULL	NULL	NULL	NULL

图 3-11 上映信息数据

3.2.3 数据更新

- 1) 对电影表进行增删改。
 - a) 增加一条信息,如图 3-12 所示,插入以后在电影表中可以查看该条信息,如图 3-13 所示。

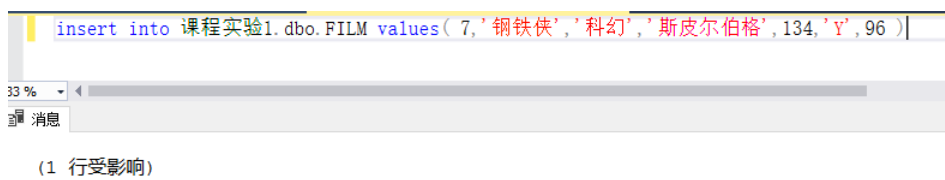


图 3-12 插入电影信息

6	唐人街探案	喜剧	陈思诚	164	Y	92
7	钢铁侠	科幻	斯皮尔伯格	134	Y	96

图 3-13 插入电影信息结果

- b) 修改电影信息,如图 3-14 所示,将钢铁侠的评分改成 100 分;修改以后查看该条信息,如图 3-15 所示,修改成功。

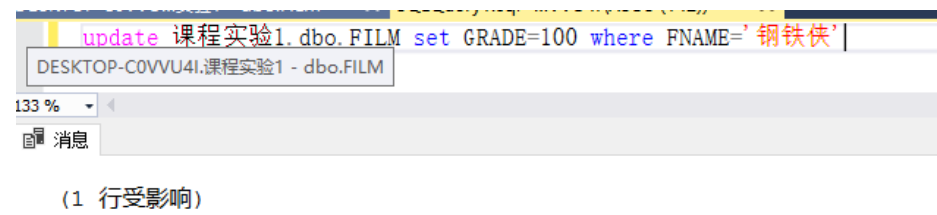


图 3-14 修改电影信息

7	钢铁侠	...	科幻	斯皮尔伯格	...	134	Y	100
NULL	NULL	...	NULL	NULL	...	NULL	NULL	NULL

图 3-15 修改电影信息结果

- c) 删除电影信息，如图 3-16 所示，删除新插入的钢铁侠电影信息；如图 3-17 所示，在电影表中已经找不到该信息。

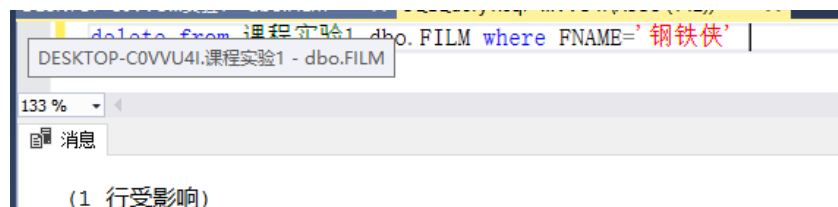


图 3-16 删除电影信息

1	战狼	...	战争	吴京	...	150	Y	98
2	猛虎英雄传	...	动作	猛虎	...	133	N	87
3	美丽人生	...	战争	犹太	...	118	N	91
4	未知死亡	...	爱情	印度阿三	...	185	Y	93
5	爱丽丝梦游仙...	...	科幻	疯帽子	...	112	Y	89
6	唐人街探案	...	喜剧	陈思诚	...	164	Y	92
NULL	NULL	...	NULL	NULL	...	NULL	NULL	NULL

图 3-17 删除电影信息结果

- 2) 批处理操作。将 90 后演员的信息插入新表中，如图 3-18 所示；执行语句，出现一个新表，如图 3-19 所示，表中有两位 90 后演员。

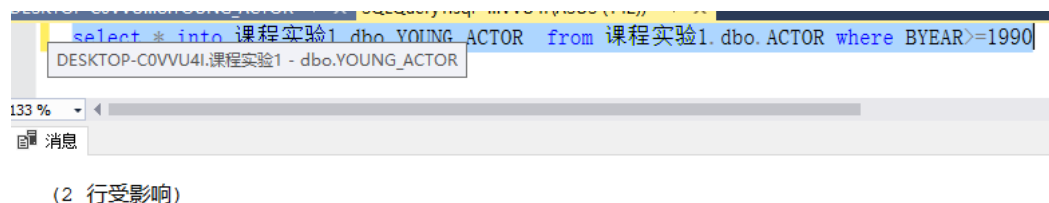


图 3-18 创建年轻演员表

ACTID	ANAME	SEX	BYEAR
9	刘昊然	男	1997
NULL	NULL	NULL	NULL

图 3-19 年轻演员表

- 3) 数据导入导出。在第一部分实验中已经测试了两种数据备份方式，其中包括导入和导出，不再重复。
- 4) 观察性实验。先创建一个没有主键的表，向其中录入两条重复的数据，如图 3-20 所示，第二条数据上有一个红色感叹号“!”，将光标移至该处，有提示信息说该行是只读的。

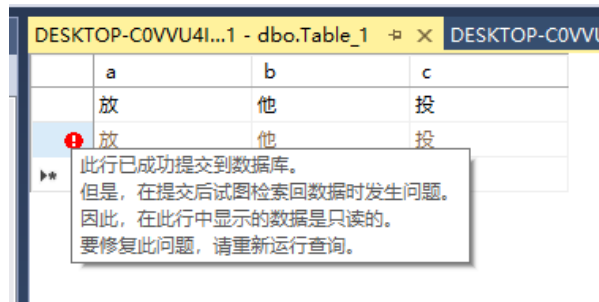


图 3-20 向没有主键的表插入数据

关闭该表，重新打开，如图 3-21 所示，此时已经没有了红色感叹号，表也是可以编辑的。

	a	b	c
	放	他	投
	放	他	投
	NULL	NULL	NULL

图 3-21 无主键表插入结果

对该表进行查询，如图 3-22 所示，两条重复信息都被查出，由此可见，没有主键的表是可以存在的，而且可以向这样的表中插入两条完全相同的信息。

select * from 课程实验1.dbo.Table_1 where a='放'

	a	b	c
1	放	他	投
2	放	他	投

图 3-22 无主键表查询结果

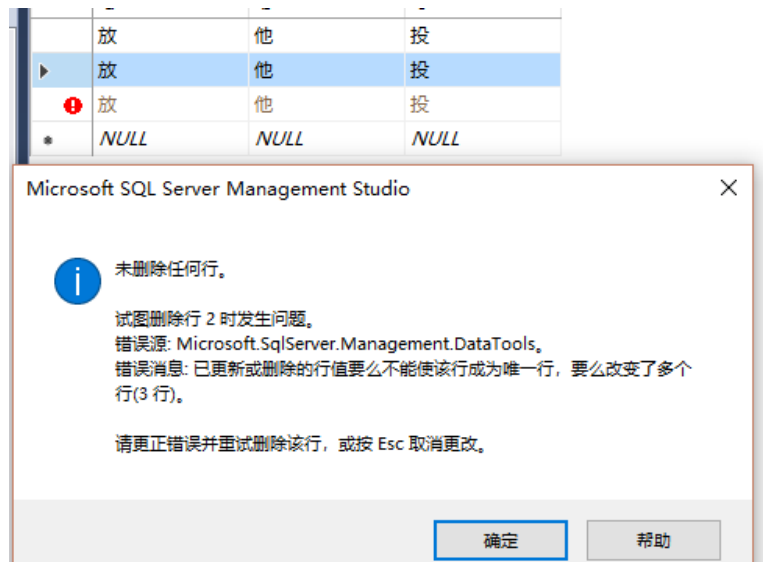


图 3-23 删除无主键表信息

继续向表中插入第三条相同信息，仍然有前述的感叹号出现。再试图删除第二行数据，弹出窗口如图 3-23 所示，删除失败。分析原因，应该是执行删除时需要用到 where 条件进行查询，查找出来的信息有三条，系统不知道该删除哪一

条。对表进行修改，同样弹出这个窗口提示，原因与之相同，系统不确定该修改哪一条。

5) 创建视图，执行创建视图语句，如图 3-24 所示，将 80 后演员的信息、主演的电影数等创建为新的视图。创建的视图如图 3-25 所示。

```
create view dbo.LEADER_80 (ACTID, ANAME, BYEAR, MCOUNT, MGRADE) as
select actor.ACTID, actor.ANAME, actor.BYEAR,
count(*), max(activ.GRADE) from 课程实验1.dbo.ACTOR actor,
课程实验1.dbo.ACTIN activ where actor.ACTID=activ.ACTID and activ.ISLEADING='Y'
and actor.BYEAR>=1980 group by actor.ACTID, actor.BYEAR, actor.ANAME
```

命令已成功完成。

图 3-24 创建视图

	ACTID	ANAME	BYEAR	MCOUNT	MGRADE
▶	5	刘承羽	1994	1	90
	8	王宝强	1984	1	88
	10	吉雅可汗	1988	2	79
*	NULL	NULL	NULL	NULL	NULL

图 3-25 视图信息

6) 触发器实验。

如图 3-26 所示，创建触发器并执行。

```
ALTER TRIGGER [dbo].[FILM_TRIGGER]
ON [dbo].[FILM]
AFTER INSERT, UPDATE
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
update FILM set FTYPE='喜剧' where exists
(select * from inserted where inserted.FID=FILM.FID and inserted.DNAME='周星驰')
END
```

命令已成功完成。

图 3-26 创建触发器

如图 3-27 所示，向电影表中插入一条新的信息，导演为周星驰，但是电影类型填写为“战争”。

FID	FNAME	FTYPE	DNAME	length	IS3D	GRADE
1	战狼	战争	吴京	150	Y	98
2	猛虎英雄传	动作	猛虎	133	N	87
3	美丽人生	战争	犹太	118	N	91
4	未知死亡	爱情	印度阿三	185	Y	93
5	爱丽丝梦游仙	科幻	疯帽子	112	Y	89
6	唐人街探案	喜剧	陈思诚	164	Y	92
7	钢铁侠	科幻	斯皮尔伯格	134	Y	96
8	❗ 长江七号	❗ 战争	❗ 周星驰	❗ 105	❗ N	❗ 88

图 3-27 插入信息验证触发器功能

如图 3-28 所示，确定插入，可以看到电影类型立马被修改为“喜剧”，可见触发器功能正确。

FID	FNAME	FTYPE	DNAME	length	IS3D	GRADE
1	战狼	战争	吴京	150	Y	98
2	猛虎英雄传	动作	猛虎	133	N	87
3	美丽人生	战争	犹太	118	N	91
4	未知死亡	爱情	印度阿三	185	Y	93
5	爱丽丝梦游仙...	科幻	疯帽子	112	Y	89
6	唐人街探案	喜剧	陈思诚	164	Y	92
7	钢铁侠	科幻	斯皮尔伯格	134	Y	96
8	长江七号	喜剧	周星驰	105	N	88

图 3-28 触发器功能验证

3.2.4 数据查询

- 1) 查询“战狼”这部电影在洪山区各家影院的 2017 年的上映情况，并按照上映的月份的降序排列。

要对电影表、上映表和电影院表进行连接，前两者连接条件是电影编号相同，后两者连接条件是电影院编号相同，此外还要添加约束条件，年份为 2017，电影名为战狼，电影院在洪山区。查询结果如图 3-29 所示：

```

select s.* from dbo.SHOW s, dbo.THEATER t
where t.TAREA='洪山区' and s.YEAR=2017
and t.TID=s.TID
and s.FID in (select FID from dbo.FILM where FNAME='战狼')
order by s.MONTH desc

```

FID	TID	PRICE	YEAR	MONTH
1	7	27	2017	9
2	2	38	2017	7
3	8	33	2017	6

图 3-29 查询 1

- 2) 查询所有无参演演员信息的电影的基本信息，并且将结果按照电影类型的升序排列，相同类型的电影则按照用户评分的降序排列。

要用 exists 谓词进行查询，要查找的电影是没有参演演员信息的，也就是 ACTIN 表中没有任何一项与该电影有关，用 not exists 即可解决。结果如图 3-30 所示：

```

select f.* from dbo.FILM f
where not exists (select * from dbo.ACTIN tmp where tmp.FID=f.FID)
order by f.FTYPE, f.GRADE desc

```

FID	FNAME	FTYPE	DNAME	length	IS3D	GRADE
11	尖峰时刻	动作	拉特纳	129	N	94
10	赌圣	喜剧	周星驰	132	N	99
7	钢铁侠	喜剧	斯皮尔伯格	134	Y	96
8	长江七号	喜剧	周星驰	105	N	88
9	逃学威龙	喜剧	周星驰	113	N	72

图 3-30 查询 2

- 3) 查询所有直到 2017 年仍未上映的电影编号、电影名称、导演姓名。

直到 2017 年还未上映，也就是该电影的上映时间全都大于 2017，所以可以用 2017<all（上映时间）来查询。

结果如图 3-31 所示，有些电影是 2018 上映的，比如小萝莉的猴神叔叔，有些电影则没有录入上映信息，比如周星驰的电影；

```

select f.FID, f.FNAME, f.DNAME from dbo.FILM f
where 2017<all (select s.YEAR from dbo.SHOW s where s.FID=f.FID)

```

	FID	FNAME	DNAME
1	5	爱丽丝梦游仙境	疯帽子
2	6	唐人街探案	陈思诚
3	7	钢铁侠	斯皮尔伯格
4	8	长江七号	周星驰
5	9	逃学威龙	周星驰
6	10	赌圣	周星驰
7	11	尖峰时刻	拉特纳
8	12	小萝莉的猴神叔叔	卡比尔汗

图 3-31 查询 3

- 4) 查询在每家电影院均上映过的电影编号。

我们要查询的电影编号满足：不存在这样一家电影院，它的影院编号不在所有上映了该电影的影院的列表中，用 not exists 和 not in 嵌套即可。

结果如图 3-32 所示，只有一号电影《战狼》在所有的影院都上映过；

```

select f.FID from dbo.FILM f where not exists
(select t.TID from dbo.THEATER t where t.TID not in
(select s.TID from dbo.SHOW s where s.FID=f.FID)
)

```

FID
1

图 3-32 查询 4

- 5) 查询所有用户评分低于 80 分或者高于 89 分的电影编号、电影名称、导演姓名及其用户评分，要求 where 子句中只能有一个条件表达式。

where 子句中只能有一个表达式，所以低于 80 或者高于 89 要用 not between 来表达。结果如图 3-33 所示；

```

select f.FID, f.FNAME, f.DNAME, f.GRADE from dbo.FILM f
where f.GRADE not between 80 and 89

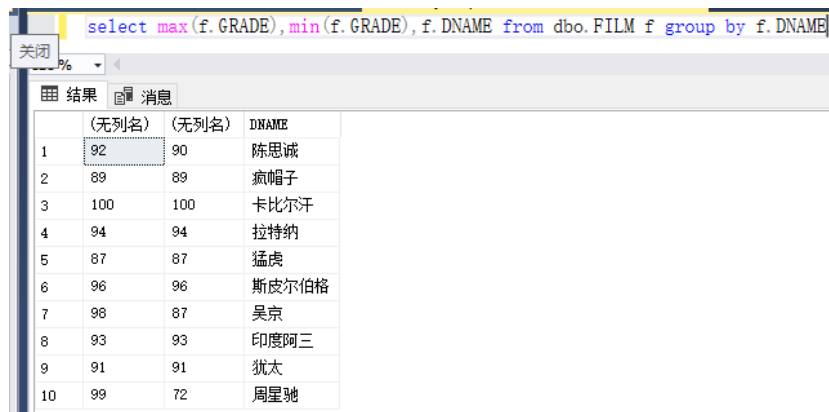
```

	FID	FNAME	DNAME	GRADE
1	1	战狼	吴京	98
2	3	美丽人生	犹太	91
3	4	未知死亡	印度阿三	93
4	6	唐人街探案	陈思诚	92
5	7	钢铁侠	斯皮尔伯格	96
6	9	逃学威龙	周星驰	72
7	10	赌圣	周星驰	99
8	11	尖峰时刻	拉特纳	94
9	12	小萝莉的猴神叔叔	卡比尔汗	100

图 3-33 查询 5

- 6) 查询每个导演所执导的全部影片的最低和最高用户评分。

查询极值，用 max 和 min 实现。结果如图 3-34 所示。



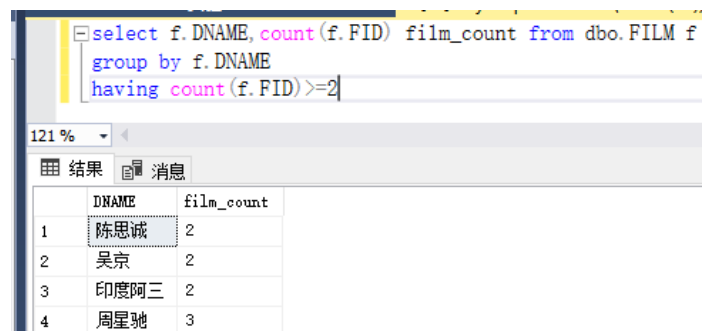
The screenshot shows a SQL query: `select max(f.GRADE), min(f.GRADE), f.DNAME from dbo.FILM f group by f.DNAME`. The result table has three columns: (无列名), (无列名), and DNAME. It lists 10 directors with their corresponding maximum and minimum grades.

	(无列名)	(无列名)	DNAME
1	92	90	陈思诚
2	89	89	疯帽子
3	100	100	卡比尔汗
4	94	94	拉特纳
5	87	87	猛虎
6	96	96	斯皮尔伯格
7	98	87	吴京
8	93	93	印度阿三
9	91	91	犹太
10	99	72	周星驰

图 3-34 查询 6

- 7) 查询至少执导过 2 部电影的导演姓名、执导电影数量。

执导电影数量用 count 进行查询，要求改导演至少执导过 2 部电影，在查询语句最后加上 having 约束，要求 count () >=2。结果如图 3-35 所示；

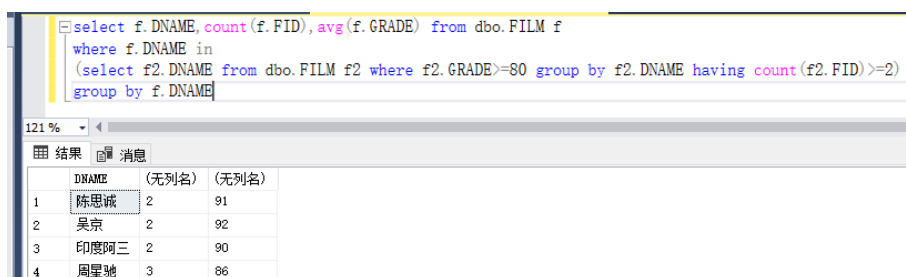


The screenshot shows a SQL query: `select f.DNAME, count(f.FID) film_count from dbo.FILM f group by f.DNAME having count(f.FID) >=2`. The result table has two columns: DNAME and film_count. It lists 4 directors: 陈思诚 (2), 吴京 (2), 印度阿三 (2), and 周星驰 (3).

	DNAME	film_count
1	陈思诚	2
2	吴京	2
3	印度阿三	2
4	周星驰	3

图 3-35 查询 7

- 8) 查询至少 2 部电影的用户评分超过 80 分的导演及其执导过的影片数量、平均用户评分。结果如图 3-36 所示，其中前三位导演都导演了两部作品，均达到了 80 分，周星驰导演了三部电影，两部达到 80 分；



The screenshot shows a SQL query: `select f.DNAME, count(f.FID), avg(f.GRADE) from dbo.FILM f where f.DNAME in (select f2.DNAME from dbo.FILM f2 where f2.GRADE >=80 group by f2.DNAME having count(f2.FID) >=2) group by f.DNAME`. The result table has three columns: DNAME, (无列名), and (无列名). It lists 4 directors with their movie counts and average ratings.

	DNAME	(无列名)	(无列名)
1	陈思诚	2	91
2	吴京	2	92
3	印度阿三	2	90
4	周星驰	3	86

图 3-36 查询 8

- 9) 查询至少执导过 2 部电影的导演姓名以及跟这些导演合作过的演员编号、姓名，结果如图 3-37 所示。

<pre> select f.DNAME, a.ACTID, a.ANAME from dbo.ACTOR a, dbo.ACTIN ai, dbo.FILM f where ai.FID=f.FID and ai.ACTID=a.ACTID and f.DNAME in (select f.DNAME from dbo.FILM f group by f.DNAME having count(*)>=2) group by f.DNAME, a.ACTID, a.ANAME </pre>			
121 %			
结果 消息			
	DNAME	ACTID	ANAME
1	陈思诚	5	刘承羽
2	陈思诚	8	王宝强
3	陈思诚	9	刘昊然
4	吴京	4	卢靖姗
5	吴京	6	吴京
6	吴京	7	余男
7	吴宇森	8	王宝强
8	吴宇森	11	成龙
9	吴宇森	12	李连杰
10	印度阿三	1	阿米尔汗
11	印度阿三	3	萨尔曼汗
12	印度阿三	10	吉雅可汗
13	周星驰	2	沙鲁克汗
14	周星驰	3	萨尔曼汗

图 3-37 查询 9

- 10) 查询每个演员担任主角的电影中的平均用户评分。

该查询要对电影表、参演表和演员表进行连接，且要求 ISLEADING 字段为‘Y’，表示该演员是主演，平均用户评分用 avg(GRADE)进行查询。结果如图 3-38 所示；

<pre> select a.ANAME, avg(ai.GRADE) from dbo.FILM f, dbo.ACTIN ai, dbo.ACTOR a where f.FID=ai.FID and ai.ACTID=a.ACTID and ai.ISLEADING='Y' group by ai.ACTID, a.ANAME </pre>			
121 %			
结果 消息			
	ANAME	(无列名)	
1	阿米尔汗	96	
2	萨尔曼汗	80	
3	卢靖姗	99	
4	刘承羽	90	
5	吴京	85	
6	王宝强	88	
7	吉雅可汗	76	

图 3-38 查询 10

- 11) 查询用户评分超过 90 分的电影的最早上映年月。结果如图 3-39 所示；

<pre> select f.FNAME, s.YEAR, s.MONTH from dbo.SHOW s, dbo.FILM f where s.YEAR in (select min(s1.YEAR) from dbo.SHOW s1 where s1.FID=f.FID) and s.MONTH in (select min(s2.MONTH) from dbo.SHOW s2 where s2.FID=f.FID and s2.YEAR in (select min(s1.YEAR) from dbo.SHOW s1 where s1.FID=f.FID))and f.GRADE>90 </pre>			
121 %			
结果 消息			
	FNAME	YEAR	MONTH
1	战狼2	2017	5
2	美丽人生	2010	6
3	未知死亡	2014	1
4	小萝莉的猴神叔叔	2018	3

图 3-39 查询 11

- 12) 查询用户评分超过 90 分的电影的最早上映年月及其相应的上映影院编号。结果如图 3-40 所示，与查询 11 相比，只是加了上映影院的编号；

```

select f.FNAME, s.YEAR, s.MONTH, s.TID from dbo.SHOW s, dbo.FILM f
where s.YEAR in (select min(s1.YEAR) from dbo.SHOW s1 where s1.FID=f.FID)
and s.MONTH in (select min(s2.MONTH) from dbo.SHOW s2 where s2.FID=f.FID and s2.YEAR in
(select min(s1.YEAR) from dbo.SHOW s1 where s1.FID=f.FID)
)and f.GRADE>90

```

	FNAME	YEAR	MONTH	TID
1	战狼2	2017	5	1
2	美丽人生	2010	6	6
3	未知死亡	2014	1	1
4	小萝莉的猴神叔叔	2018	3	5

图 3-40 查询 12

- 13) 查询每个电影的上映总次数。对电影表和上映表进行连接，然后用 count () 即可查出每个电影的上映次数。结果如图 3-41 所示；

```

select f.FNAME, count(f.FID) from dbo.FILM f, dbo.SHOW s
where f.FID=s.FID
group by f.FID, f.FNAME

```

	FNAME	(无列名)
1	战狼2	8
2	猛虎英雄传	1
3	美丽人生	2
4	未知死亡	1
5	唐人街探案2	4
6	小萝莉的猴神叔叔	1

图 3-41 查询 13

- 14) 查询执导过动作片，或者警匪片，或者枪战片的导演的姓名，要求 where 子句中只能有一个条件表达式。查询要求满足多个条件之一即可，这样的情况要用 where * in () 作为 where 子句，括号中列出多个条件。结果如图 3-42 所示；为了验证查询的正确性，还在下方查询了整个电影表。

```

select f.DNAME from dbo.FILM f
where f.FTYPE in ('警匪', '枪战', '动作')
select * from dbo.FILM

```

	DNAME
1	吴京
2	印度阿三
3	拉特纳
4	吴京

	FID	FNAME	FTYPE	DNAME	length	IS3D	GRADE
1	1	战狼2	枪战	吴京	152	Y	98
2	2	猛虎英雄传	枪战	印度阿三	133	N	87
3	3	美丽人生	战争	犹太	118	N	91
4	4	未知死亡	爱情	印度阿三	185	Y	93
5	5	爱丽丝梦游仙境	科幻	疯帽子	112	Y	89
6	6	唐人街探案2	悬疑	陈思诚	164	Y	92
7	7	钢铁侠	科幻	斯皮尔伯格	134	Y	96
8	8	长江七号	喜剧	周星驰	105	N	88
9	9	逃学威龙	喜剧	周星驰	113	N	72
10	10	赌圣	喜剧	周星驰	132	N	99
11	11	尖峰时刻	动作	拉特纳	128	N	94
12	12	小萝莉的猴	宗教	卡比尔汗	170	Y	100
13	13	战狼	枪战	吴京	112	Y	87
14	14	唐人街探案	悬疑	陈思诚	105	Y	90

图 3-42 查询 14

- 15) 查询所有“战狼”系列的电影的编号、电影名称、上映电影院名称及其上映年月，结果按照电影名称的升序排列。战狼系列已经有战狼、战狼 2，将来还可能有战狼 3，它们的共同特点是名称以“战狼”开头，而以什么结束不重要，所以可以在 where 子句中不用“=”进行限制，而用“like”进行限制，即可查出该系列的所有电影。结果如图 3-43 所示，

```

select f.FID, f.FNAME, t.TNAME, s.YEAR, s.MONTH from dbo.FILM f, dbo.SHOW s, dbo.THEATER t
where f.FID=s.FID and s.TID=t.TID and f.FNAME like '战狼%'
order by f.FNAME

```

	FID	FNAME	TNAME	YEAR	MONTH
1	13	战狼	CGV	2015	9
2	13	战狼	成龙	2015	6
3	1	战狼2	中影星美	2017	5
4	1	战狼2	巨幕	2017	7
5	1	战狼2	万达	2017	7
6	1	战狼2	CGV	2017	8
7	1	战狼2	华夏	2017	7
8	1	战狼2	成龙	2017	6
9	1	战狼2	天河	2017	9
10	1	战狼2	格瓦拉	2017	6

图 3-43 查询 15

- 16) 查询在同一年月上映 1 号和 2 号电影的影院编号。结果如图 3-44 所示。

```

select t.TID, t.TNAME from dbo.THEATER t
where exists
(select * from dbo.SHOW s where s.YEAR in (select s2.YEAR from dbo.SHOW s2 where s2.FID=2 and s2.TID=t.TID)
and s.MONTH in (select s3.MONTH from dbo.SHOW s3 where s3.FID=2 and s3.TID=t.TID)
and s.FID=1)
group by t.TID, t.TNAME

```

	TID	TNAME
1	4	CGV
2	8	格瓦拉

图 3-44 查询 16

- 17) 查询所有没参演过用户评分 85 分以下电影的演员的编号、姓名。
用嵌套查询实现，在内层查询中查出一个演员参演的电影的评分，外层循环中用 $85 \leq \text{all}$ 作为 where 子句的条件。结果如图 3-45 所示。

```

select a.ACTID, a.ANAME from dbo.ACTIN ai, dbo.ACTOR a
where 85 <= all
(select f.GRADE from dbo.FILM f where f.FID in
(select FID from dbo.ACTIN ai where ai.ACTID=a.ACTID)
)
group by a.ACTID, a.ANAME

```

	ACTID	ANAME
1	1	阿米尔汗
2	4	卢靖姗
3	5	刘承羽
4	6	吴京
5	7	余男
6	8	王宝强
7	9	刘昊然
8	10	吉雅可汗

图 3-45 查询 17

- 18) 查询参演过“吴宇森”执导过的所有电影的演员姓名。

我们要查询的演员满足：不存在这样一部电影，它是吴宇森导演的，但它的电影编号不在该演员参演的电影编号列表中。结果如图 3-46 所示

```

select a. ANAME from dbo. ACTOR a
where not exists
(
select * from dbo. FILM f where f. DNAME=' 吴宇森' and f. FID not in
(
select ai. FID from dbo. ACTIN ai where ai. ACTID=a. ACTID )
)

```

	ANAME
1	成龙
2	李连杰

图 3-46 查询 18

- 19) 查询所有的演员的编号、姓名及其参演过的电影名称，要求即使该演员未参演过任何电影也要能够输出其编号、姓名。前面的查询用的是自然连接，查询结果中只有多个表都不为空的部分，先要求没有任何参演记录的演员也要查出，应该用外连接，left join 即可。结果如图 3-47 所示，多数演员都有参演记录，最后几名演员没有参演，电影名一栏为空。

```

select a. ACTID, a. ANAME, f. FNAME from (dbo. ACTOR a left join dbo. ACTIN ai
on a. ACTID=ai. ACTID) left join dbo. FILM f on f. FID=ai. FID
group by a. ACTID, a. ANAME, f. FNAME

```

	ACTID	ANAME	FNAME
1	1	阿米尔汗	美丽人生
2	1	阿米尔汗	未知死亡
3	2	沙鲁克汗	逃学威龙
4	3	萨尔曼汗	猛虎英雄传
5	3	萨尔曼汗	逃学威龙
6	4	卢靖姗	战狼2
7	5	刘承羽	唐人街探案2
8	6	吴京	战狼2
9	7	余男	战狼2
10	8	王宝强	唐人街探案2
11	8	王宝强	铁汉柔情
12	9	刘昊然	唐人街探案2
13	10	吉雅可汗	爱丽丝梦游仙境
14	10	吉雅可汗	未知死亡
15	11	成龙	谍中谍
16	11	成龙	铁汉柔情
17	12	李连杰	谍中谍
18	12	李连杰	铁汉柔情
19	13	甄子丹	NULL
20	14	范冰冰	NULL
21	15	迪丽热巴	NULL

图 3-47 查询 19

- 20) 查询所有上映超过 3 次但没有用户评分的电影编号、名称。

没有人评分，where 子句表示为 GRADE is null；上映超过 3 次，用 having count>3 限制。

结果如图 3-48 所示。没有评分的电影有《唐人街探案》和《战狼》，其中《战狼》上映没有超过三次，所以满足条件的结果只有一个。

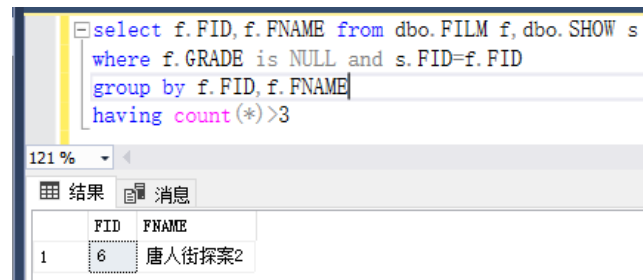


图 3-48 查询 20

3.3 任务总结

- 1) 查询部分基本上不存在技术性的问题，主要考察的是对表的连接、投影等运用的熟练程度，有部分查询语句较难实现，花费了不少时间，但是都是可实现的。
- 2) 书本上给出的触发器模型是针对 Oracle 数据库的，在 SQL server 中创建这样的触发器会报语法错误，在查阅了有关 SQL server 的触发器用法以后解决了这一问题。

4 综合实践任务

4.1 系统设计目标

1) 应用背景:

图书管理系统是各图书馆都需要使用的系统,它能帮助图书管理员更好地管理海量图书,尤其是在图书馆藏数量和读者数量都很大的图书馆,我们无法用手写的方式录入、管理图书信息和借阅信息,必须借助图书管理系统来完成这些工作。

2) 总体目标:

系统的使用者分为管理员和读者。管理员负责新书的录入、旧书的清理、借书审核、还书审核,以及日常的维护,有权限增删图书的信息;读者要能够根据书名查找图书,并选择中意的图书进行借阅,读完以后还可以还书;读者没有权限修改图书的信息,他对图书的操作仅限于借、还。这就是一个简单的图书管理系统的大致功能。

4.2 需求分析

1) 系统总体功能需求列表如表 4.1 所示。

表 4.1 系统总体需求

功能编号	功能名称	功能描述	权限
1	录入图书信息	将新的书籍信息录入系统	管理员
2	删除图书信息	将损坏的图书从系统删除	管理员
3	修改图书信息	更新图书信息	管理员
4	审核还书申请	审核读者的还书请求,检查图书损坏情况并开出罚单	管理员
5	审核借书申请	审核读者是否有罚金未交,确定读者是否可借阅书籍	管理员
6	查找书籍	通过书名查找书籍信息	读者
7	查询个人信息	读者查询个人信息	读者
8	查询借书记录	读者查询借书记录	读者
9	借书	读者选择书籍进行借阅	读者
10	还书	归还借阅的书籍	读者
11	续借	继续借阅到期的书籍	读者
12	交款	缴纳罚金	读者

2) 性能需求上,要求所有的操作都没有明显的延迟,这对于现在的计算机来说不算难事,所以是不用刻意考虑性能方面的因素的。

- 3) 数据完整性，具体在数据库设计中说明。
- 4) 数据流图，系统的数据流图如图 4-1 所示。

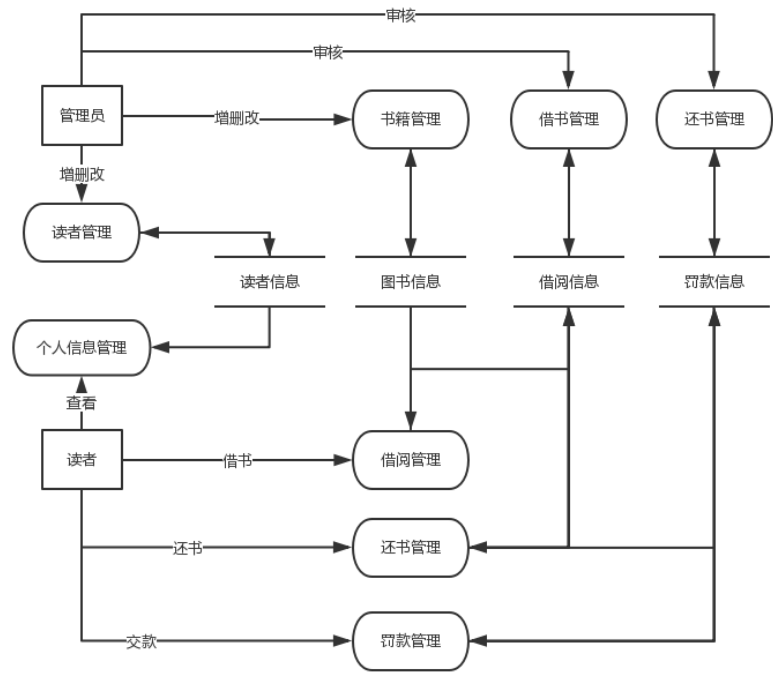


图 4-1 系统数据流图

4.3 总体设计

系统采用 C/S 模式实现，没有为服务器设计专门的客户端，实际上服务器就是数据库，用户是读者和管理员。主要分为两大功能板块：管理员和读者。

实际的图书管理系统要配合扫描仪等设备进行使用，我们的系统显然不能连接外设，适当信息需要手动输入。

系统的总体结构图如图 4-2 所示。

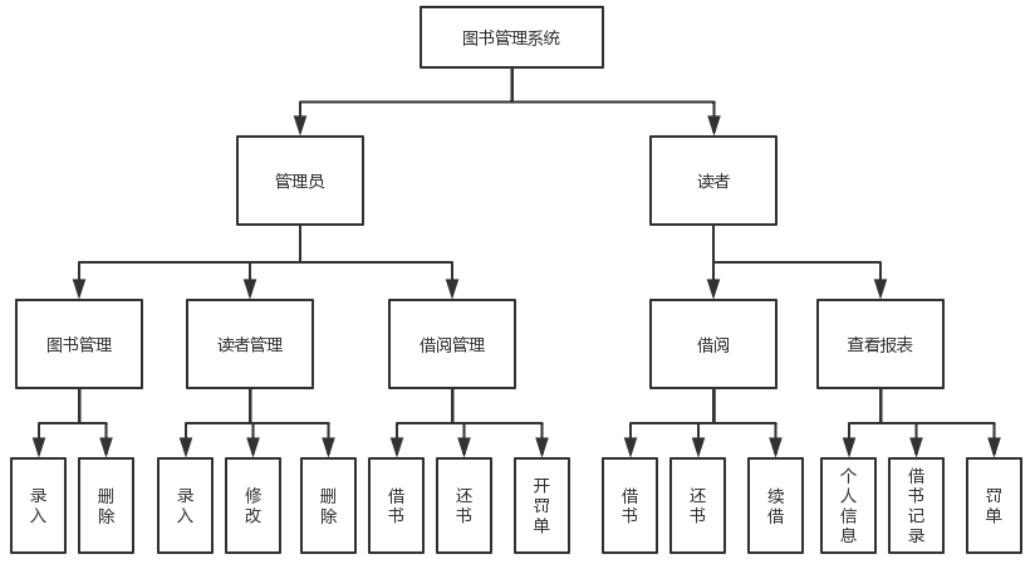


图 4-2 总体设计图

1) 管理员

a) 图书信息管理

具体又分为两个部分：新书上架，也就是将新书的各项信息录入到数据库；查询维护，通过索书号、书名、作者等信息进行查询，得到查询结果后，可以选中并删除图书，这主要是方便管理员查找图书，找到以后还可以精准地删除图书。

b) 读者信息管理

与图书信息管理类似，可以录入、查找并修改或注销读者信息。

c) 借阅管理

分为借书和还书两部分。读者想要借书时，发出一个借书请求，管理员收到该请求后，查看该读者是否还存在有罚金未交的情况，若是，则拒绝读者的借书请求；否则，同意借书。读者还书时也发出还书请求，管理员收到后，查看该读者是否延期还书，或者图书损坏，若是，则开出罚单，等待读者缴费，若图书损坏，则将图书标记为损坏，否则，进行还书登记。

2) 读者

a) 借阅

分为借书、还书和续借。借书时，可以根据关键字在数据库中进行查找，选择借书，此时发出借书申请，等图书管理员审核通过后才能成功借书；还书时，从借阅列表中选择尚未归还的书籍，发出还书请求，等待管理员审核，若审核发现图书有损坏，则会收到罚单；续借是将当前正在借阅的书籍再延长一个借书周期。

b) 查看报表，读者可以查看个人信息、借阅信息、罚单。

4.4 数据库设计

1) ER 图

用 Microsoft Visio2010 与 SQL server 连接，生成的实体关系图如图 4-3 所示。

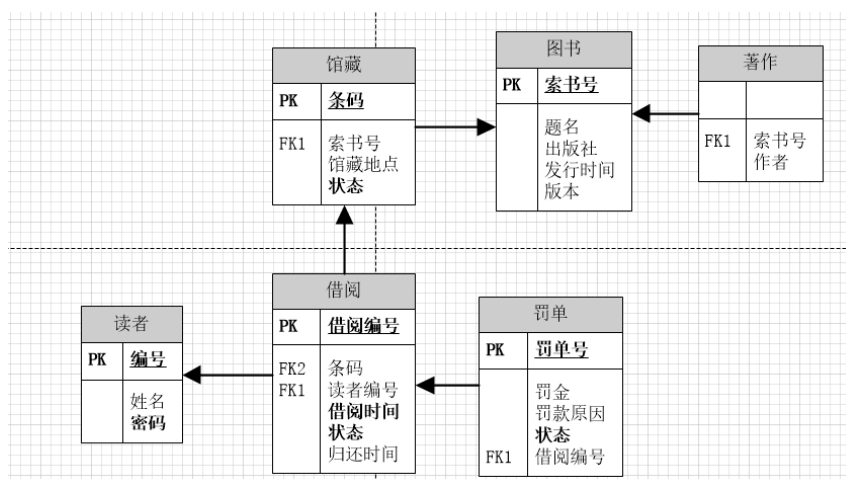


图 4-3 实体关系图

2) 权限控制

图书馆的访问者主要是管理员和读者，所以设置两个用户。管理员有所有表的所有权限，读者的权限则要受到限制。

在图 4-3 的 6 个表中，读者有权限访问馆藏表、图书表和作者表，因为读者需要查询图书；读者有权限访问读者信息表，因为他需要查看自己的个人信息；读者有权限查询和更新借阅表，因为他需要查看自己的借书记录，此外，发起借书请求时会向借阅表中增加一条记录，所以需要更新权限；读者有权限查询和更新罚单表，与前者同理，读者需要查看罚单，并且缴纳罚金时会修改罚单状态，所以需要更新权限。读者没有删除任何一个表的权限，有查询所有表的权限和修改部分表的权限。

3) 数据库逻辑结构设计

数据库主要分成三大部分：图书信息、借阅信息和读者信息，其中借阅信息和读者信息分别可用一个表实现，但是图书信息相对较复杂。在了解了学校图书馆的图书管理系统以后，发现每一本书都有一个索书号，一个索书号可以对应该书的多本实体，每个实体只用索书号是无法区分的，图书馆采用条码进行区分，所以需要增设馆藏信息表来存放每本实体书的条码号和馆藏地点、馆藏状态等。

同样的，一本书也可能有多个作者，所以也需要用一个单独的表来存放作者，这里我们暂不考虑作者同名的情况，所以没有给作者设置编号。出版社在一般情况下不会同名，也不设置单独的表。

借阅信息用于存放读者的借阅记录，由于借阅过程存在罚款，所以还有一个罚款表。

最终图书管理系统的数据库共有 6 个表：图书信息表、作者信息表、馆藏信息表、借阅信息表、罚款信息表、读者信息表和罚单表，表项分别如下：

- i. 图书信息表：题名，索书号，出版社，发行时间，版本。主码为索书号，题名也不能为空。
- ii. 馆藏信息表：条码，索书号，馆藏地点，状态。主码为条码，索书号为外码，来自图书信息表，馆藏状态不能为空。
- iii. 作者信息表：索书号，作者。索书号为外码，来自图书信息表。
- iv. 借阅信息表：条码，读者编号，借阅日期，归还日期，借阅编号，状态。主码为借阅编号，条码为来自馆藏信息表的外码，读者编号为来自读者信息表的外码。
- v. 读者信息表：编号，姓名。编号为主码。
- vi. 罚款信息表：罚单号，罚金，罚款原因，状态，借阅编号。主码为罚单号，借阅编号为来自借阅信息表的外码。

除了基本表外，还需要若干个视图来实现系统的功能。

- i. 图书视图：题名，索书号，作者，出版社，发行时间，版次，条码，馆藏

地点，状态。该视图用于管理员和读者查询图书的结果显示，是图书信息表、馆藏信息表和作者表三者的连接。

- ii. 借阅记录视图：借阅编号，条码，书名，读者编号，读者姓名，借阅时间。该视图用于管理员对读者的借书请求进行处理，视图中包含所有读者的所有借阅记录，是借阅信息表、图书信息表和读者信息表的连接。
- iii. 借阅统计：借阅申请次数、借阅成功次数、还书申请次数、还书破损册数。该视图用于管理员的统计查询，对一段时间内的借书、还书信息进行统计。
- iv. 借阅排行：索书号，书名，借阅次数。用于管理员统计一段时间内借阅次数排名前列的书籍，了解图书馆最受欢迎的书籍情况。

4) 数据库物理设计

图书表设计如表 4.2 所示：

表 4.2 图书信息

字段名	类型	主键/外键	为空	索引	取值	备注
索书号	char(20)	主键	否	是		
书名	char(100)		否	是		
出版社	char(30)					
发行时间	date				合法日期	
版本	tinyint				正整数	

馆藏信息表设计如表 4.3 所示：

表 4.3 馆藏表

字段名	类型	主键/外键	为空	索引	取值	备注
条码	int	主键	否	是	从 1 起自增	
索书号	char(20)	外键	否	是	来自图书表	
馆藏地点	char(100)		否			
状态	tinyint		否		0、1、2	1-在馆 2-借出 0-损坏

著作表设计如表 4.4 所示：

表 4.4 著作表

字段名	类型	主键/外键	为空	索引	取值	备注
索书号	char(20)	外键	否	是	来自图书表	
作者	char(20)		否	是		

读者表设计如表 4.5 所示：

表 4.5 读者表

字段名	类型	主键/外键	为空	索引	取值	备注
编号	int	主键	否	是	从 1 自增	
姓名	char(10)		否	是		
密码	char(20)		否			

借阅表设计如表 4.6 所示：

表 4.6 借阅表

字段名	类型	主键/外键	为空	索引	取值	备注
借阅编号	int	主键	否	是	从 1 自增	
条码	int	外键	否	是	来自馆藏表	
读者编号	int	外键	否	是	来自读者表	
借阅日期	datetime		否			
归还日期	datetime					
状态	tinyint		否		0-4	1-申请借阅 2-在借 3-申请归还 4-已归还 0-失效

罚单表设计如表 4.7 所示：

表 4.7 罚单表

字段名	类型	主键/外键	为空	索引	取值	备注
罚单号	int	主键	否	是	从 1 自增	
借阅编号	int	外键	否	是	来自借阅表	
罚金	int		否			
罚款原因	bit		否		0、1	0-过期归还罚款 1-损坏图书罚款
状态	bit		否		0、1	0-用户未交罚金 1-罚金已付

4.5 详细设计与实现

系统用 Java 语言实现，通过 jdbc 与 SQL server 进行连接，获取数据库中的数据。对管理员和用户分别提供了客户端，图形界面用 Javafx 实现。

1) 读者-搜索图书

a) 流程图

算法的业务流程图如图 4-4 所示。借书功能与搜索功能是密不可分的，一般都是先搜索再借书。程序初始化时就从数据库中查找出所有的图书信息，

并显示到表格上，用户可以直接选择从表中选取某一项，然后进行借阅；如果图书信息过多，直接选取较为困难，可以在输入框中输入关键字搜索，再选择图书进行操作。

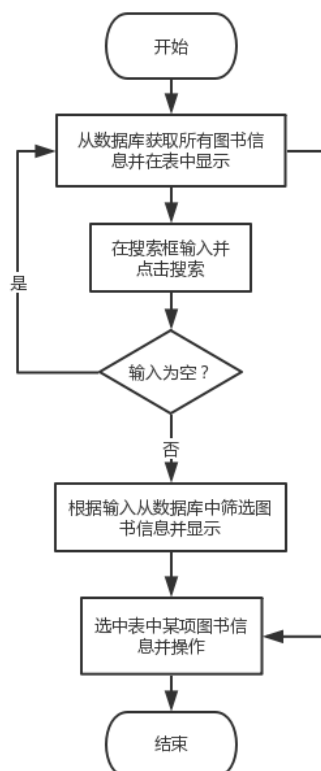


图 4-4 查找图书流程

b) 算法说明

图书的状态，如在馆、借出等，在数据库中是用 `int` 型存储的，所以向管理员展示时，必须用汉字直观地打出，所以要将 `int` 转换成字符串；

由于每本书都可能有多作者，所以要对作者进行特别处理，当多条记录的索书号相同而作者不同时，要合并为一条记录，把各个作者拼接成一个字符串，中间用分号隔开；

此外，在数据库中，如果某个字段是长度为 20 的字符类型，而实际只用了 8 个字符，那么后面 12 个字符会用空格填充，这虽然不影响查询的结果，但是在表格中显示时，空格也会占用空间，所以需要用 Java 中 `String.trim()` 去除多余的空格。

c) 数据库实现

sql 语句如下，其中 `index` 是搜索的关键字，关键字可以是图书的索书号的一部分、书名的一部分或者图书作者的一部分，即只要有部分字段匹配即可搜索出结果，所以 `where` 子句中用“`like`”进行匹配。

```

sql  "select * from dbo.图书 t, dbo.馆藏 t2, dbo.著作 t3 where t.索书号=t2.索书号 "
    " and t3.索书号=t.索书号 and (t.索书号=' " index
    "' or t.题名 like '%" index "%' or t2.条码=' " index "' or t3.作者 like "
    "'%" index "%') order by t2.条码,t.索书号,t3.作者"
  
```

2) 读者-借书

a) 流程图

图书搜索功能中，在选中了图书并申请借阅以后，要进行一系列判断审核，流程图如图 4-5 所示。

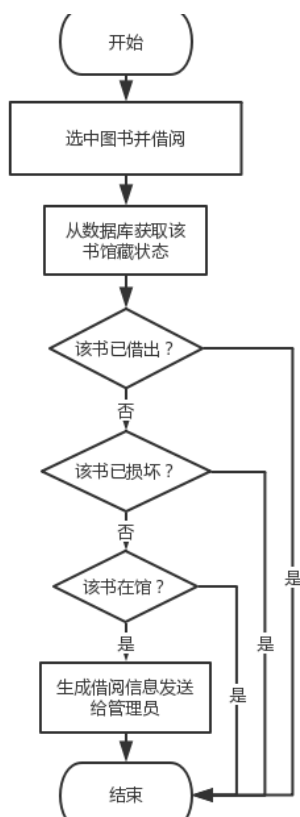


图 4-5 借书流程

b) 算法说明

从选中表项中获取该的条码号，实际上从表中可以直接获取图书的馆藏状态，但是考虑到表中数据可能未及时更新，该书在数据库中可能已经改变状态，所以不能用表中的状态。

根据条码号从数据库中查询该书状态，如果已经借出或者已经损坏，则提示用户借书失败；否则，可以生成借阅信息，插入到借阅信息表中，并提示用户借书请求成功。这不意味着真正地借书成功，还要等待管理员审核。

c) 数据库实现

插入借阅信息 sql 语句如下，借阅时间是当前时间，借阅编号自动生成。

```
sql  "insert into dbo.借阅(条码,读者编号,借阅时间,状态,"
      "借阅编号) values(" barcode ",'" Main getUser() "',"
      "'" df format new Date()) "',1," (count 1) ")"
```

3) 读者-续借/还书

a) 流程图

算法流程图如图 4-6 所示。续借和还书都是以借阅记录表为基础的。虽然在需求分析中，这些功能是相互独立的，但在实际实现时，我们要把他们充分联系起来，这也是为了用户的使用方便。

从数据库中获取该读者所有的借阅记录，读者可以选中任意一条记录，选择续借或者归还该书籍。

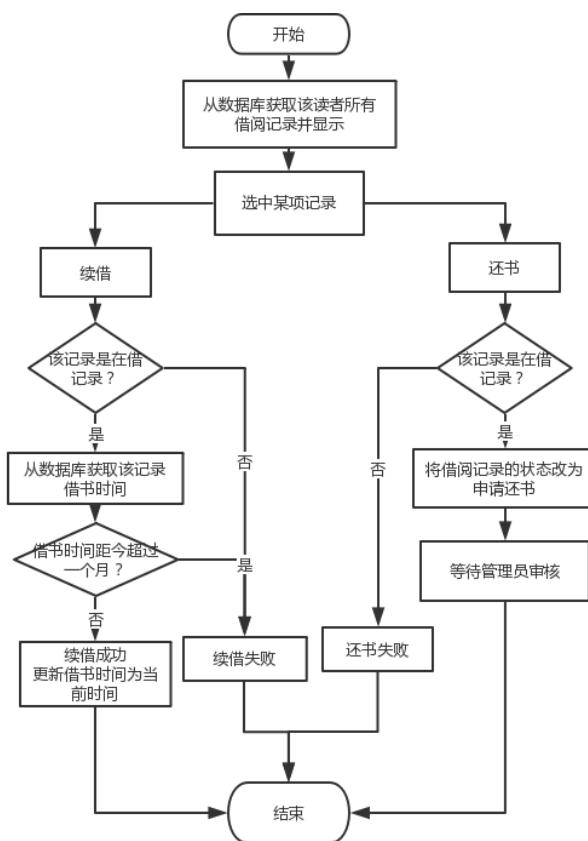


图 4-6 续借-还书流程

b) 算法说明

读者查看借阅记录时，借阅记录表已经初始化好，从数据库中选取了该读者的所有历史借阅记录，记录的状态分为在借、已归还。

读者选中某条记录后，若选择续借，则首先要判断是不是在借记录，不是则无法续借，然后还要判断该书是否逾期未还，借书期限为一个月，超过期限的书是不可以续借的，若上述条件都满足，则续借成功，将数据库中的借阅时间更新为当前时间，还书期限重新计时。

若读者选择归还，则先判断该记录是否为在借记录，若不是，则无法归还，若是，则直接申请归还，这不需要新增一条借阅记录，只需要将借阅记录中的状态从在借修改为申请还书，管理员看到以后会进行审核。

4) 读者-查看个人信息

该选项只能查看信息，不可进行其他操作，功能简单。

获取读者在借册数的 sql 语句如下：

```
sql  "select count(*) total from dbo.借阅 t where t.读者编号"
      "='" usernum "' and t.状态=2"
```

获取读者总罚金数的 sql 语句如下：

```
sql  "select sum(t.罚金) total from dbo.罚单 t, dbo.借阅 t2 "
      "where t.借阅编号=t2.借阅编号 and t2.读者编号='" usernum "' and t.状态=1"
```

5) 读者-查看罚单

该选项只能查看信息，功能简单。获取罚单的 sql 语句如下：

```
sql "select t.罚单号,t.罚金,t.罚款原因,t.状态,t.借阅编号,t4.题名,"  
    "t3.借阅时间,t3.归还时间"  
    " from dbo.罚单 t,dbo.馆藏 t2,dbo.借阅 t3,dbo.图书 t4 "  
    "where t.借阅编号=t3.借阅编号 and t3.条码=t2.条码 and "  
    "t2.索书号=t4.索书号 and t3.读者编号='" Main getUser() "'"
```

6) 管理员-查询图书

管理员查询图书与读者查询图书的流程基本相同，不再赘述。

7) 管理员-下架旧书

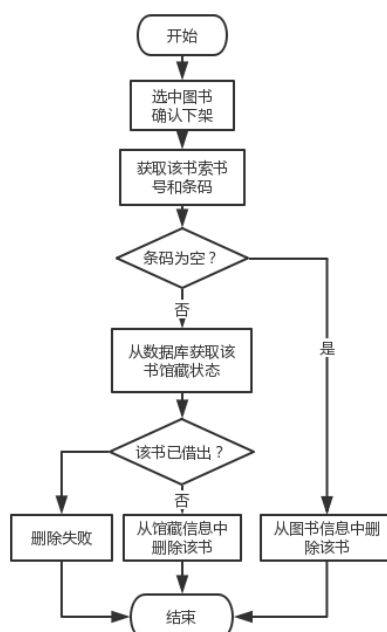


图 4-7 下架流程

a) 流程图

算法流程图如图 4-7 所示。

b) 算法说明

由于管理员看到的图书信息实际上是由图书信息表和馆藏信息表连接生成的，所以在进行图书下架时，既涉及到馆藏信息的删除，又有图书信息的删除。有些书籍没有馆藏信息，只有图书信息，则只需要删除图书信息表，否则只要删除馆藏信息表，不能删除图书信息表，因为该图书可能存在其他实体，我们只能一次删除馆藏信息中的一个实体，如果把图书信息也一并删除，则馆藏信息的外键引用会出错。

8) 管理员-录入图书

a) 流程图

算法流程图如图 4-8 所示。

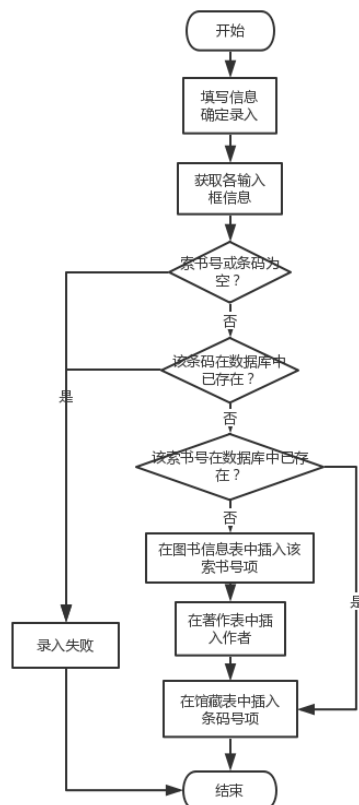


图 4-8 录入流程

b) 算法说明

录入图书时，索书号和条码号绝对不能为空，须在所有操作之前检查。录入的情况分两种：首先是索书号已经存在的，也就是说，这本书以前就在图书馆中存在，现在又购置了该种书的一批新书，所以图书信息不需要再插入，直接插入馆藏表即可；

第二种是该书原本不存在的情况，图书馆第一次购置该种书，则需要先插入图书信息表，然后再插入馆藏信息表，不可改变插入顺序，否则外键引用出错。

c) 数据库实现

插入图书信息：

```
sql "insert into dbo.图书(题名,索书号,出版社,"
    "发行时间,版本) "
    "values (" bname "," isbn "," press "," ymd "," version ")"
```

插入馆藏信息：

```
sql "insert into dbo.馆藏(条码,索书号,馆藏地点,"
    "状态) "
    "values (" barcode "," isbn "," store ",1)"
```

9) 管理员-查询读者

a) 流程图

算法流程图如图 4-9 所示：

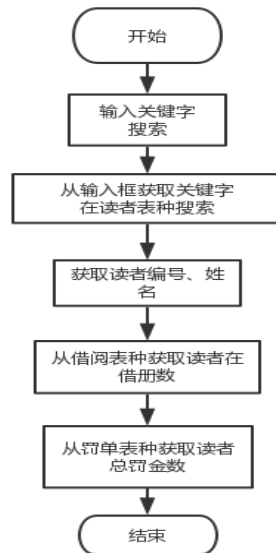


图 4-9 查询读者流程

b) 算法说明

搜索框的输入可以是读者编号、读者姓名的一部分，在数据库中的读者信息表中查找符合要求的项，这样的项可能有多个，循环取出，依次查找他们的在借册数和罚金数额，都添加到结果表格中显示。

10) 管理员-修改读者

a) 算法说明

管理员选中读者表中的某个以后，选择修改该读者的信息，由于 Javafx 中的表格直接修改不太方便，所以用弹窗的方式修改。将原来的读者信息在弹窗中显示，管理员选择修改，点击确认以后，将修改后的信息写入到数据库中该读者对应的条目。

11) 管理员-注销读者

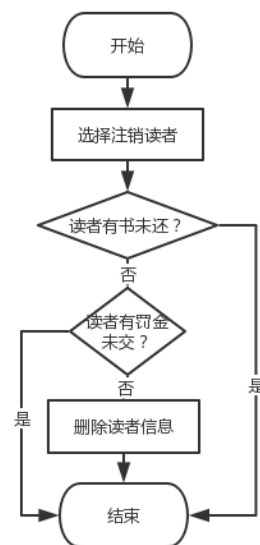


图 4-10 注销读者流程

a) 流程图

算法流程图如图 4-10 所示。

b) 算法说明

管理员选中某个读者，选择注销该读者，先判断读者是否有在借书籍，如果有，则不能注销，否则该读者无法还书；再判断该读者是否有罚金未交，如果有，则不可注销，否则会造成图书馆的亏损。若上述情况都没有，则直接将其从读者信息表中删除，其他在外键中引用了该读者的表项也会被触发器一并删除。

12) 管理员-审核借书

a) 流程图

算法流程图如图 4-11 所示：

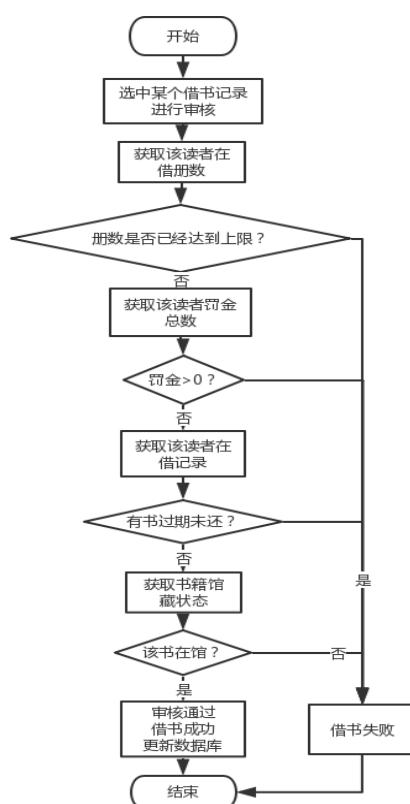


图 4-11 审核借书流程

b) 算法说明

审核读者的借书申请时，首先要检查读者已经借了多少册书，若达到上限，则不允许继续借书；然后检查该读者的罚单，如果有罚金未交，也不允许借书；再检查该读者在借记录，若有书过期未还，不允许借书。

以上情况均不存在，则读者有借书资格，然后检查该书是否在馆，因为该书可能被前面的读者借走了，若在馆则允许借书，审核通过，否则失败。

13) 管理员-审核还书

a) 流程图

算法流程图如图 4-12 所示：

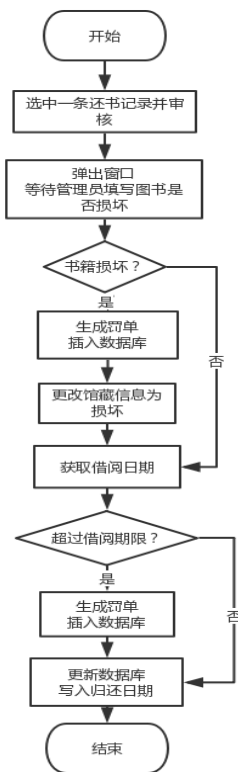


图 4-12 审核还书流程

b) 算法说明

管理员审核还书时，先要看图书是否损坏，判断以后填写，如果损坏，则需要生成书籍损坏的罚单，还要将该图书的馆藏信息状态修改为损坏。然后再判断读者是否未按时归还该书籍，若是，则需要生成延期的罚单。最后将归还日期写入数据库，审核完成。

14) 数据库事务实现

在进行书籍的录入时，先要向图书信息表添加数据，然后再向馆藏信息表添加数据，这两者是分两步进行的。在一般情况下，两步都会成功。但是，特殊情况下，可能插入图书信息成功，而插入馆藏信息失败，这是不应该的。为此我们要添加数据库事务，目的是，对于连续的几个数据库操作，要使其成为原子操作，也就是，要么全都执行成功，要么全都不执行。假如第一条成功了，而第二条失败了，那么要将第一条的结果回滚，部分代码如下所示：

```

stmt.executeUpdate(sql_1);
stmt.executeUpdate(sql_2);
conn.commit();
//若所有结果执行完成，则执行提交事务
ps.close();
conn.close()
catch(ClassNotFoundException e){
    e.printStackTrace();
}catch(SQLException e){
    e.printStackTrace();
    try{
        //只要有一个 sql 语句错误，则事务回滚
        conn.rollback();
    }catch(SQLException e){

```

```

        e.printStackTrace();
    }
}

```

15) 触发器-删除空白书籍

a) 触发器说明

书籍的信息是存放在图书信息、馆藏信息两个表中的，在录入某种书籍时，先在图书信息表中录入一条数据，再在馆藏信息表中录入多条数据，表示该种书的多个实体。假如该种书全都损坏，则馆藏表中关于该书的条目全都被删除，此时图书信息表中还残留有该书的信息，这条信息已经无效了，需要删除。为了避免手动删除，使用触发器来实现，每次更新或者删除馆藏信息后，检测图书信息表中是否有某条记录已经没有馆藏书籍，将它从图书信息表中删除。

b) 触发器实现

触发器功能部分如下：

```

ALTER TRIGGER [dbo].[delete_emptybook]
ON [dbo].[馆藏]
AFTER DELETE, UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    delete from 图书 where 0=(select count(*) from 馆藏 where 馆藏.索书号=图书.索书号)
END

```

16) 触发器-设置图书损坏

a) 触发器说明

当读者还书时，管理员需要判定该书是否损坏，如果鉴定为损坏，则需要开具罚单，罚单中有一项表示罚款的原因是书籍损坏还是迟还，如果该项填写为书籍损坏，则自动将馆藏表中对应的书籍馆藏状态设置为损坏，省去了手动设置。

b) 触发器实现

触发器功能部分实现如下：

```

ALTER TRIGGER [dbo].[set_bookbroken]
ON [dbo].[罚单]
AFTER INSERT, DELETE, UPDATE
AS
BEGIN
    update 馆藏 set 状态=0 where exists
    (select * from inserted, (select * from 借阅)t where inserted.罚款原因=1 and
    inserted.借阅编号=t.借阅编号
    and t.条码=馆藏.条码)
END

```

17) 触发器-设置罚金

c) 触发器说明

在开具罚单时，管理员只需要填写罚款原因，而罚款金额有数据库生

成，规定过期还书罚款 1 元，书籍损坏罚款 5 元，自动设置罚款的触发器根据罚款原因进行罚金的设置。

d) 触发器实现

触发器功能部分实现如下：

```
ALTER TRIGGER [dbo].[set_fine]
ON [dbo].[罚单]
AFTER INSERT,DELETE,UPDATE
AS
    declare @sake bit;
BEGIN
    select @sake = 罚款原因 from 罚单;
    if @sake=0 begin
        update 罚单 set 罚金=1 where exists(select * from inserted where 罚单.罚单号
=inserted.罚单号)
    end
    else begin
        update 罚单 set 罚金=5 where exists(select * from inserted where 罚单.罚单号
=inserted.罚单号)
    end
END
```

18) 触发器-修改读者信息

e) 触发器说明

管理员有一项功能是修改读者的信息，如果只是修改了读者姓名、密码等，那么只需要修改读者表这一个表，而如果修改了读者编号，即修改了主码，那么与该读者有关的借阅信息表中存有的该读者的编号也要进行修改，这一操作不能让管理员执行，需要用触发器实现。

f) 触发器实现

触发器功能部分实现如下：

```
ALTER TRIGGER [dbo].[update_aboutuser]
ON [dbo].[读者]
AFTER UPDATE
AS
    declare @old char(10)
    declare @newuser char(10)
BEGIN
    select @old=t.编号 from deleted t
    select @newuser =t.编号 from inserted t
    update dbo.借阅 set 读者编号=@newuser where 借阅.读者编号=@old
END
```

4.6 系统测试

1) 管理员-图书查询测试

a) 测试说明

可以在选择框输入索书号、作者或者是书名，均可查询出结果。

b) 测试过程

管理员的图书查询维护界面如图 4-13 所示，未执行查询时会显示所有的

图书信息。

图书录入	图书查询维护	读者录入	读者查询维护	借书管理	还书管理			
所有记录				搜索		下架		
题名	索书号	作者	出版社	发行时间	版	条码	地点	状态
数据结构	cs1	秦嘉华;袁玲	高等教育出版社	2016-02-28	1	1	新馆1楼	在馆
数据结构	cs1	秦嘉华;袁玲	高等教育出版社	2016-02-28	1	162	新馆2楼	损坏
数据结构	cs1	秦嘉华;袁玲	高等教育出版社	2016-02-28	1	2	新馆1楼	损坏
数据结构	cs1	秦嘉华;袁玲	高等教育出版社	2016-02-28	1	3	新馆1楼	在馆
数据结构	cs2	秦嘉华;王多...	高等教育出版社	2018-03-10	2	4	新馆1楼	在馆
数据结构	cs2	秦嘉华;王多...	高等教育出版社	2018-03-10	2	5	新馆1楼	在馆
夏浩的网	et1	拉夫罗夫	儿童文学出版社	2015-06-14	1	6	老馆2楼	在馆
水浒传	gd1	施耐庵;吴承恩	古籍出版社	2009-12-13	1	7	新馆4楼	在馆
水浒传	gd1	施耐庵;吴承恩	古籍出版社	2009-12-13	1	8	新馆4楼	在馆
撒哈拉的故事	sw1	三毛	长江出版社	2014-03-24	1	20	老馆5楼	在馆
撒哈拉的故事	sw1	三毛	长江出版社	2014-03-24	1	9	老馆5楼	在馆
万水千山走遍	sw2	三毛	人民出版社	2015-12-08	1	10	老馆5楼	在馆

图 4-13 所有图书信息

查询可以根据索书号、书名、作者这三个关键字的全部或者部分进行查询，根据索书号查询结果如图 4-14 所示。

图书录入	图书查询维护	读者录入	读者查询维护	借书管理	还书管理			
所有记录		cs			搜索		下架	
题名	索书号	作者	出版社	发行时间	版	条码	地点	状态
数据结构	cs1	袁玲	高等教育出版社	2016-02-28	1	1	新馆1楼	在馆
数据结构	cs1	袁玲;袁玲	高等教育出版社	2016-02-28	1	162	新馆2楼	损坏
数据结构	cs1	袁玲;袁玲;袁玲	高等教育出版社	2016-02-28	1	2	新馆1楼	损坏
数据结构	cs1	袁玲;袁玲;袁...	高等教育出版社	2016-02-28	1	3	新馆1楼	在馆
数据结构	cs2	袁玲;袁玲;袁...	高等教育出版社	2018-03-10	2	4	新馆1楼	在馆
数据结构	cs2	袁玲;袁玲;袁...	高等教育出版社	2018-03-10	2	5	新馆1楼	在馆

图 4-14 按索书号查找

根据书名查找结果如图 4-15 所示。

图书录入		图书查询维护		读者录入		读者查询维护		借书管理		还书管理	
所有记录		科波菲尔				搜索		下架			
题名		索书号	作者	出版社		发行时间		版 条	码	地点	状态
大卫·科波菲尔		wy1	查尔斯·狄更...	译林出版社		2012-08-04		1	11	老馆6楼	借出
大卫·科波菲尔		wy1	查尔斯·狄更...	译林出版社		2012-08-04		1	19	老馆6楼	损坏

图 4-15 按书名查找

根据作者查询结果如图 4-16 所示。

图书录入	图书查询维护	读者录入	读者查询维护	借书管理	还书管理		
所有记录		三毛			搜索	下架	
题名	索书号	作者	出版社	发行时间	版 条码	地点	状态
万水千山走遍	sw2	三毛	人民出版社	2015-12-08	1 10	老馆5楼	在馆
撒哈拉的故事	sw1	三毛	长江出版社	2014-03-24	1 20	老馆5楼	在馆
撒哈拉的故事	sw1	三毛	长江出版社	2014-03-24	1 9	老馆5楼	在馆

图 4-16 按作者查找

查询失败结果如图 4-17 所示，书库中暂无该书。

图书录入	图书查询维护	读者录入	读者查询维护	借书管理	还书管理		
所有记录		数据库系统概论		搜索	下架		
题名	索书号	作者	出版社	发行时间	版 条码	地点	状态

图 4-17 查询失败

若输入框为空，则查询结果为所有书籍。

2) 管理员-图书下架测试

a) 测试说明

任务书中要求管理员可以进行旧书下架，旧这个概念比较模糊，没法严格定义，所以我们把一本书是否已经“旧”了的决策权交给管理员，也就是说，管理员可以删除任何他认为算作旧书的书籍，所以管理员可以选择任何书籍进行删除，除非该书已借出，不在馆。

b) 测试过程

选中已经借出的书籍《大卫·科波菲尔》，选择下架，弹出窗口警告该书已借出，不可删除，如图 4-18 所示。

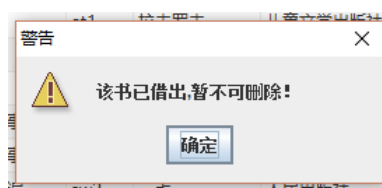


图 4-18 删除失败

选择在馆图书《夏洛的网》，点击下架，结果如图 4-19 所示，删除成功。



图 4-19 下架成功

《夏洛的网》只有一本实体书，删除以后，馆藏信息表中已经没有该书，而图书信息表中还有该书，此时触发器检测到该书没有馆藏信息，在图书信息表中将其自动删除。

3) 管理员-新书上架测试

a) 测试说明

新书上架界面如图 4-20 所示，其中索书号一栏用的是下拉框，既可以选择又可以输入，这样设计是因为，录入图书时，有两种情况：其一，图书馆购置了一批最新的书籍，馆中没有这种书，那么索书号必须要新分配，所以要输入索书号；其二，馆中已经有的书，最近又增购了一些，录入这些书籍时，就可以直接从列表中选择索书号了，我们的测试也分这两种情况进行。书籍的条码号不需要手动填写，数据库会自增生成。

图 4-20 上架界面

b) 测试过程

首先选择录入已经存在的书籍，从索书号列表选择一个索书号，如图 4-21 所示，书名、作者等项会自动填充，且不可编辑，只有条码号和馆藏地点可以填写。

图 4-21 上架已有书籍

填写好以后，点击录入，录入成功，结果如图 4-22 所示，最后一行是《兄弟》新录入的馆藏本，条码号为 168。

兄弟	xs1	余华	岳麓出版社	2017-07-01	1	17	新馆5楼	借出
兄弟	xs1	余华	岳麓出版社	2017-07-01	1	18	新馆5楼	借出
兄弟	xs1	余华	岳麓出版社	2017-07-01	1	167	老馆3楼	在馆
兄弟	xs1	余华	岳麓出版社	2017-07-01	1	168	老馆3楼	在馆

图 4-22 上架成功

图 4-23 上架新书

再测试录入馆中没有的书籍，我们录入一本计算机丛书《编译原理》，索

书号为 cs3，其他各项信息如图 4-23 所示。

确定录入，录入成功，结果如图 4-24 所示，最后一行是刚录入的书籍。

数据结构	cs1	秦磊华袁玲	高等教育出版社	2016-02-28	1	2	新馆1楼	损坏
数据结构	cs1	秦磊华袁玲	高等教育出版社	2016-02-28	1	3	新馆1楼	在馆
数据结构	cs2	秦磊华王多...	高等教育出版社	2018-03-10	2	4	新馆1楼	损坏
数据结构	cs2	秦磊华王多...	高等教育出版社	2018-03-10	2	5	新馆1楼	在馆
编译原理	cs3	刘汝佳	清华大学出版社	2017-03-21	1	169	老馆5楼	在馆

图 4-24 上架成功

测试异常情况,如图 4-25 所示,不填写索书号直接录入,弹出窗口警告。



图 4-25 上架异常

如图 4-26 所示,不填写馆藏地点直接录入,也提示错误。

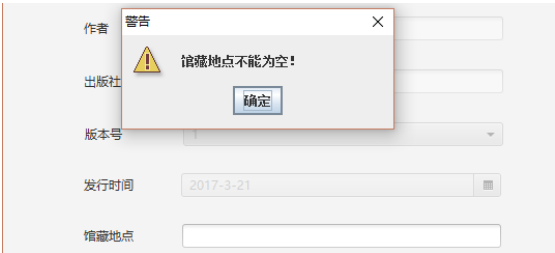


图 4-26 上架异常

4) 管理员-读者查询

a) 测试说明

读者信息查询可以根据读者编号、读者姓名的一部分或全部进行,但不能通过用户密码进行查询,因为涉及到密码的操作不太安全,实际上这里将用户密码直接显示也是不太好的,只不过我们认为管理员有一切权限,所以可以查看用户密码。

b) 测试过程

读者查询维护界面初始化如图 4-27 所示,表中显示了所有的读者。

图书录入 图书查询维护 读者录入 读者查询维护 借书管理 还书管理					
		搜索	修改	删除	
读者编号	读者姓名	密码	在借册数	待付罚金	
1	王尼玛	wang	0	0	
10	张三	zhangsan	0	0	
2	陈根	chen	0	0	
201514699	张鑫	zhang	0	0	
3	蔡韬	cai	0	0	
4	陈胜强	chen	0	0	
5	杜兰特	durant	0	0	

图 4-27 查看读者

根据读者姓名查询,在搜索框输入“张”,查询结果如图 4-28 所示。

张		搜索	修改	删除	
读者编号	读者姓名	密码	在借册数	待付罚金	
10	张三	zhangsan	0	0	
201514699	张鑫	zhang	0	0	

图 4-28 按姓名查询读者

根据读者编号进行查询，结果如图 4-29 所示。

图书录入	图书查询维护	读者录入	读者查询维护	借书管理	还书管理
2015		搜索		修改	删除
读者编号	读者姓名		密码	在借册数	待付罚金
201514699	张鑫		zhang	0	0

图 4-29 按编号查询读者

查询异常情况，输入的关键字不匹配，查询结果为空，如图 4-30 所示。

图书录入	图书查询维护	读者录入	读者查询维护	借书管理	还书管理
201513		搜索	修改	删除	
读者编号		读者姓名	密码	在借册数	待付罚金

图 4-30 查询读者失败

5) 管理员-录入读者信息

a) 测试说明

读者录入分为正常录入、读者编号重复异常、信息不全异常。

b) 测试过程

录入一名读者，如图 4-31 所示。

图书录入	图书查询维护	读者录入	读者查询维护	借书管理	还书管理
<div>读者姓名 <input type="text" value="高泽文"/></div> <div>登录密码 <input type="text" value="gao"/></div> <div>录入</div>					

图 4-31 录入读者

录入以后，在读者列表中可以看到刚刚录入的读者信息，如图 4-32 所示。

7	李四	lisi	0	0	
8	高泽文	gao	0	0	

图 4-32 录入读者成功

将读者姓名、密码之中的任意一项空缺，点击录入，将提示信息填写不全，如图 4-33 所示。

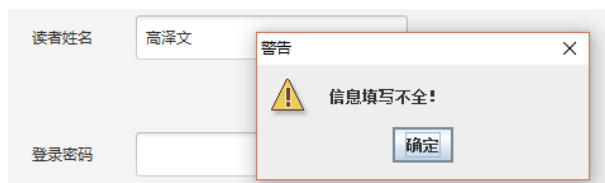


图 4-33 录入读者失败

6) 管理员-修改读者信息

a) 测试说明

管理员可以选择一条用户信息，选择修改。修改后点击确定，即可修改成功。不允许修改主码读者编号，因为该项是自增的，不可编辑，可以任意修改读者姓名和登录密码。

b) 测试过程

选择读者“张三”，点击修改，弹出窗口如图 4-34 所示。将读者姓名修改为“李四”，密码改为“lisi”，点击确定。

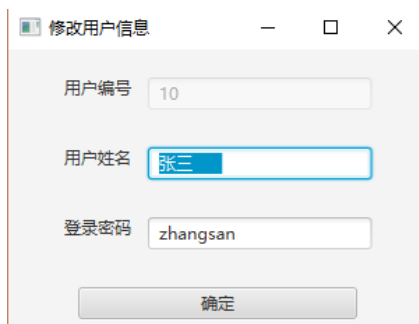


图 4-34 修改读者信息

修改成功后的读者信息如图 4-35 所示。

1	王尼玛	wang	0	0
10	李四	lisi	0	0
2	陈根	chen	0	0

图 4-35 修改读者成功

7) 管理员-注销用户

a) 测试说明

管理员有权限删除任何一个用户，只是也要受一些限制。如果读者有借书还未归还，则不能删除；如果读者有罚金未交，则不能删除。只有读者与图书馆不存在利益关系时才可注销账户。

b) 测试过程

选择 3 号用户蔡韬，点击删除，如图 4-36 所示，注销成功。



图 4-36 注销读者

如图 4-37 所示，注销以后该用户不存在。

1	王尼玛	wang	1	0	
10	李四	lisi	0	5	
2	陈根	chen	0	0	
201514699	张鑫	zhang	0	0	
4	陈胜强	chen	0	0	
5	杜兰特	durant	0	0	

图 4-37 注销读者成功

选择用户“王尼玛”，点击注销，如图 4-38 所示，可见该用户还有一本书未归还，所以不可注销。



图 4-38 注销读者失败

如图 4-39 所示，选择用户“李四”，点击注销，由于该用户还有 5 元罚金未交，所以也不可注销。

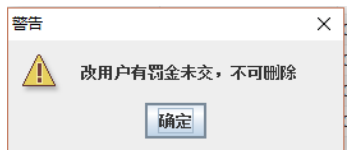


图 4-39 注销读者失败

8) 管理员-查看报表

统计报表的功能较为简单，如图 4-40 所示是借阅统计报表，可以选择开始时间和结束时间，统计该段时间内的总借阅申请、借阅成功、还书申请、还书成功数目。

开始时间	2018-5-27	结束时间	2018-6-28
借还统计	借阅排行		
借阅申请	借阅成功	还书申请	还书成功
7	2	2	2

图 4-40 借阅统计

如图 4-41 所示是借阅排行榜，可以选择开始时间和结束时间，统计该段时间内，图书馆的书籍借出次数，降序排列，便于管理员掌握受欢迎的书籍种类。

开始时间	2018-5-27	结束时间	2018-6-28
借还统计	借阅排行		
书名	索书号	借阅次数	
数据结构 ...	cs1	3	
水浒传 ...	gd1	2	
撒哈拉的故事 ...	sw1	1	
数据结构 ...	cs2	1	

图 4-41 借阅排行

9) 借书/还书综合测试

a) 测试说明

由于读者的借书和管理员的借书审核、读者的还书和管理员的还书审核，互相之间是密不可分的，所以测试也不能孤立地进行，我们将读者和管理员的借书、还书操作混合交替进行，以达到测试借书、还书的目的。

管理员审核读者的借出请求时，首先要判断读者在借册数是否超标，再判断是否有在借的书逾期未还，最后判断是否有罚金还未交，以上三种情况都拒绝借书，否则审核通过，借书成功。

借书测试计划是，先让一个用户连续申请借书 4 次，管理员进行审核，由于用户借书上限是 3 本，所以该用户的前三次请求均能通过，属于正常情况，第 4 次被拒绝，属于异常中的借书册数超标。然后读者申请还书，管理员很合还书时将该书鉴定为损坏，对读者开罚单，读者收到罚单后再次借书会失败，这是借书异常中的罚款未交异常。再将读者借书日期修改为超过 1 个月以前，这样读者再次借书也会失败，属于借书异常中的有书逾期未还异常。

b) 测试过程

读者请求借书，连续申请 4 次，如图 4-42 所示。



图 4-42 借书请求

在管理员一端可以看到读者“张鑫”的 4 次借书请求，如图 4-43 所示。

7	12	羊脂球	10	李四	2018-06-12 23:55:32.0
8	14	兄弟	10	李四	2018-06-12 23:55:37.0
9	12	羊脂球	10	李四	2018-06-12 23:56:08.0
10	3	数据结构	10	李四	2018-06-12 23:56:17.0
11	10	万水千山走遍	201514699	张鑫	2018-06-13 13:00:25.0
12	20	撒哈拉的故事	201514699	张鑫	2018-06-13 13:00:32.0
13	23	编译原理	201514699	张鑫	2018-06-13 13:00:35.0
14	3	数据结构	201514699	张鑫	2018-06-13 13:04:11.0

图 4-43 借书请求

管理员进行借书审核，如图 4-44 所示，读者借书成功。



图 4-44 审核借书

审核到第 4 本书时，系统检测到该读者已经借书 3 本，不可继续借书，所以弹窗警告，如图 4-45 所示，借书失败。

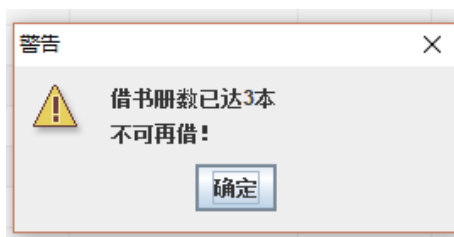


图 4-45 借书失败

紧接着，读者申请还书，如图 4-46 所示。



图 4-46 申请还书

管理员审核还书，如图 4-47 所示，管理员填写鉴定图书是否损坏，默认为正常，可以下拉选择损坏。

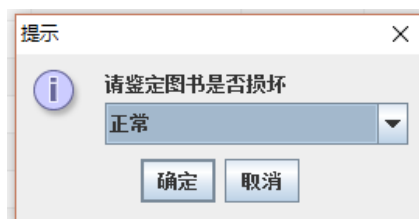


图 4-47 审核还书

选择损坏并点击确定，图书损坏需要罚款 5 元，如图 4-48 所示。

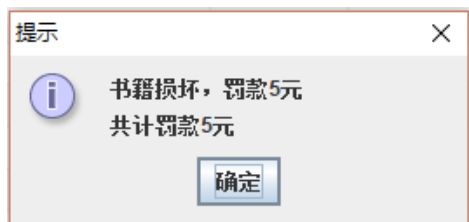


图 4-48 开罚单

由于该读者还书时被鉴定为损坏图书，所以他已经被取消了借阅资格，只有缴纳了罚金以后才会被取消限制，现在该读者再次尝试借书，如图 4-49 所示，管理员审核时检测到他有罚金未交，拒绝了借书请求。

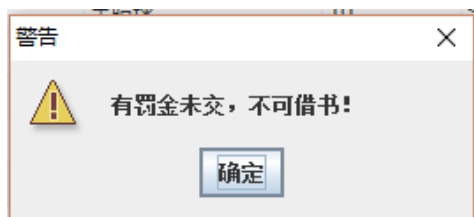


图 4-49 拒绝借书

要恢复还书资格，读者必须缴纳罚款，如图 4-50 所示，读者成功缴纳 5 元罚金，重获借书资格并再次申请借书。



图 4-50 交罚金

管理员再次审核读者的借书请求，如图 4-51 所示，借书成功。



图 4-51 借书成功

最后一种异常情况，就是读者借书过期未还时，不可继续借书。我们设置的借书期限是一个月，考虑到测试时生成的借阅记录都是今天的，所以短时间内不会超过借书期限，为了完成测试，我们直接到 SQL server 中将借阅事件修改到两个月以前，这样一来，这条借阅记录就一定过期了。如图 4-52 所示，最后一条借阅记录时间为 4 月 12 日。

借阅编号	条码	书名	借阅时间	状态
1	3	数据结构	2018-06-12 23:32:05.0	借阅失败
2	5	数据结构	2018-06-12 23:33:36.0	借阅失败
3	3	数据结构	2018-06-12 23:36:26.0	借阅失败
4	4	数据结构	2018-04-12 23:38:57.0	在借

图 4-52 借阅记录

该读者有书过期未还，现在再次申请借书，如图 4-53 所示，管理员审核借书信息时，检测到他有书过期未还，拒绝借书。



图 4-53 借书失败

读者将过期未还的书申请还书，如图 4-54 所示，管理员审核时，系统检测到该书过期，收取 1 元罚金，该读者被取消借书资格，只有缴纳罚金以后才会取消限制。

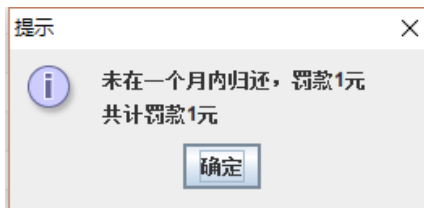


图 4-54 开罚单

c) 结果分析

以上测试过程涵盖了读者借书、读者还书、管理员审核借书、管理员审

核还书。读者不管有没有借书资格，都是可以发出借书申请的，最终借书是否通过，在管理员一方决定。读者所借书籍不管是不是已经过期、是不是损坏，都是可以申请还书的，最终是否罚款，在管理员一方判断。所以说，读者借书、还书都没有异常情况。

管理员审核借书，有三种异常情况：该读者已经有 3 本书在借未还；该读者有罚金未交；该读者有至少一本书过期未还。这三种情况在测试都进行了处理。

管理员审核还书，有正常情况和两种异常情况。不论如何，都会弹出窗口要求管理员填写图书是否损坏，如果填写正常，那么直接审核通过；如果填写损坏，那么罚款 5 元；如果读者还书时已经超过借书期限，则罚款 1 元。在测试用例中上述情况都进行了测试。

综上，读者、管理员的借书、还书综合测试成功，系统功能达到设计目标。

10) 读者-图书查询

a) 测试说明

读者的图书查询与管理员稍有不同，管理员的查询是单条件查询，而为了使读者的使用体验更好，向读者提供了多条件查询。

b) 测试过程

如图 4-55 和图 4-56 所示，可以根据书名、作者或者出版社进行单项查找，不需要的条件直接不填即可。

书名	兄弟	作者		出版社		搜索	借阅	
题名	索书号	条码	作者	出版社	发行时间	版	地点	状态
兄弟	... xs1	... 13	路遥;余华	岳麓出版社	... 2017-07-01	1	新馆5楼	借出
兄弟	... xs1	... 14	路遥;余华...	岳麓出版社	... 2017-07-01	1	新馆5楼	在馆
兄弟	... xs1	... 15	路遥;余华...	岳麓出版社	... 2017-07-01	1	新馆5楼	借出
兄弟	... xs1	... 16	路遥;余华...	岳麓出版社	... 2017-07-01	1	新馆5楼	损坏
兄弟	... xs1	... 17	路遥;余华...	岳麓出版社	... 2017-07-01	1	新馆5楼	借出
兄弟	... xs1	... 18	路遥;余华...	岳麓出版社	... 2017-07-01	1	新馆5楼	借出
兄弟	... xs1	... 167	路遥;余华...	岳麓出版社	... 2017-07-01	1	老馆3楼	在馆
兄弟	... xs1	... 168	路遥;余华...	岳麓出版社	... 2017-07-01	1	老馆3楼	在馆

图 4-55 按书名查询

书名		作者	三毛	出版社		搜索	借阅	
题名	索书号	条码	作者	出版社	发行时间	版	地点	状态
撒哈拉的故事	... sw1	... 9	三毛	长江出版社	... 2014-03-24	1	老馆5楼	在馆
万水千山走遍	... sw2	... 10	三毛	人民出版社	... 2015-12-08	1	老馆5楼	在馆

图 4-56 按作者查询

如图 4-57 所示，可以根据作者和出版社进行多条件查找。

书名		作者	三毛	出版社	人民出版社	搜索	借阅	
题名	索书号	条码	作者	出版社	发行时间	版	地点	状态
万水千山走遍	... sw2	... 10	三毛	人民出版社	... 2015-12-08	1	老馆5楼	在馆

图 4-57 多条件查找

如图 4-58 所示，还可以三个条件同时查找。

借书	借阅记录	罚单	个人信息										
书名	万水千山走遍	作者	三毛	出版社	人民出版社	搜索	借阅						
题名	索书号	条码	作者	出版社	发行时间	版	地点	状态					
万水千山走遍	...	sw2	...	10	三毛	...	人民出版社	...	2015-12-08	1	老馆5楼	...	在馆

图 4-58 多条件查找

11) 读者-查看借书记录

a) 测试过程

如图 4-59 所示是该读者的借书记录，最后一列“在借”表示当前正在读者手中的书籍，“借阅失败”可能是因为读者有罚金未交等异常情况引起的，“已还”表示读者已经成功归还书籍。

借书	借阅记录	罚单	个人信息	
还书		续借		
借阅编号	条码	书名	借阅时间	状态
1	3	数据结构	2018-06-12 23:32:05.0	借阅失败
2	5	数据结构	2018-06-12 23:33:36.0	借阅失败
3	3	数据结构	2018-06-12 23:36:26.0	借阅失败
4	4	数据结构	2018-04-12 23:38:57.0	已还
19	3	数据结构	2018-06-13 13:17:23.0	借阅失败
20	9	撒哈拉的故事	2018-06-13 14:47:36.0	借阅失败
21	8	水浒传	2018-06-13 14:47:58.0	在借

图 4-59 借书记录

12) 读者-查看基本信息

a) 测试过程

如图 4-60 所示是当前读者的基本信息。

借书	借阅记录	罚单	个人信息
读者编号	1		
姓名	王尼玛		
在借册数	1		
待处罚金	0		

图 4-60 读者个人信息

13) 读者-续借

a) 测试说明

一本书籍在借，且为过期时，读者可以选择续借，续借成功后，借阅时间更新为当前时间，借阅期限重新计算。当书籍过期未还时不可续借，不是在借的书籍也不可续借。以下测试将对上述三种情况进行测试。

b) 测试过程

当前读者的借阅记录如图 4-61 所示，有两本在借书籍，一本是 4 月 12

日借阅的，已经过期，另一本是 6 月 13 日借阅的，还未过期。

借书 借阅记录 罚单 个人信息				
		还书	续借	
借阅编号	条码	书名	借阅时间	状态
1	3	数据结构	2018-06-12 23:32:05.0	借阅失败
2	5	数据结构	2018-06-12 23:33:36.0	借阅失败
3	3	数据结构	2018-06-12 23:36:26.0	借阅失败
4	4	数据结构	2018-04-12 23:38:57.0	在借
19	3	数据结构	2018-06-13 13:17:23.0	借阅失败
20	9	撒哈拉的故事	2018-06-13 14:47:36.0	借阅失败
21	8	水浒传	2018-06-13 14:47:58.0	在借

图 4-61 读者借阅记录

选择续借 6 月 13 日借阅的书籍，如图 4-62 所示，续借成功。

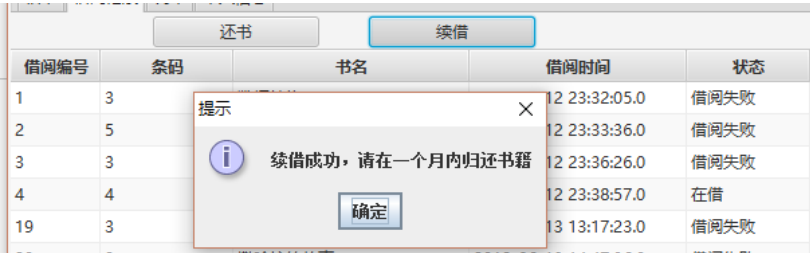


图 4-62 续借

如图 4-63 所示，续借成功后，书籍的借阅时间进行了更新。

借书 借阅记录 罚单 个人信息				
		还书	续借	
借阅编号	条码	书名	借阅时间	状态
1	3	数据结构	2018-06-12 23:32:05.0	借阅失败
2	5	数据结构	2018-06-12 23:33:36.0	借阅失败
3	3	数据结构	2018-06-12 23:36:26.0	借阅失败
4	4	数据结构	2018-04-12 23:38:57.0	在借
19	3	数据结构	2018-06-13 13:17:23.0	借阅失败
20	9	撒哈拉的故事	2018-06-13 14:47:36.0	借阅失败
21	8	水浒传	2018-06-13 15:00:22.0	在借

图 4-63 续借成功

如图 4-64 所示，选择续借 4 月 12 日借阅的书籍，系统检测到该书已过期，拒绝了续借。



图 4-64 续借失败

如图 4-65 所示，选择一条借阅失败的记录进行续借，系统提示该条信息不可续借。

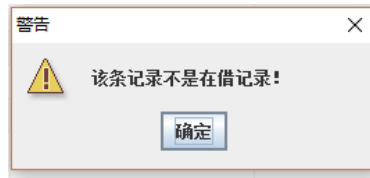


图 4-65 续借失败

4.7 系统设计与实现总结

- 1) 根据任务书, 结合学校图书馆的实际情况, 进行了需求分析。对于任务书中的各项功能, 可以合并的进行合并, 方便用户使用系统。
- 2) 根据各实体间的关系, 绘制了 ER 图, 由此设计出了各个数据库表。
- 3) 根据需求分析的结果, 对系统进行了总体设计, 确定系统的大致功能。
- 4) 将总体设计细化, 划分出子模块, 用 Java 语言配合 SQL server 数据库, 逐个具体实现了功能模块。
- 5) 设计一定量的触发器, 将自动更新操作等人为操作较为繁琐的事务交给数据库自动执行。
- 6) 向数据库中录入一定数据, 进行功能测试, 适当修改代码, 逐步完善系统。

5 课程总结

- 1) 主要熟悉了 SQL server 的使用,学会了新增用户并设置用户权限;掌握了 sql 查询语句的多种用法以及较为复杂的查询逻辑的实现;掌握了触发器的简单使用;将数据库知识和实际结合,实现了一个功能比较完备的管理系统。
- 2) 实验过程中使用了 MySQL 和 SQL server 两个数据库,两者各有优劣,基本的操作都已经掌握,在以后的使用了可以灵活选择。只是对于命令行下操作数据库还不够熟悉。
- 3) 触发器的实现花费了较多时间,因为触发器的语句写法很灵活,而且 Oracle、SQL server 等不同数据库的触发器语法还大不相同。在编程的过程中发现,很多能够在高级语言中实现的功能,其实也可以在数据库中用触发器实现,两者对比,选择用触发器实现更好,可以减少编码量。
- 4) 在系统生成报表时,是用 sql 语句对多个表进行连接生成的,而没有使用视图,这方面还有待改进,用视图生成报表更加方便。
- 5) 软件功能学习部分的选做题没有完成,有时间还需要实现,加深对数据库的理解和操作的熟练程度。

附录

第二部分：

```
insert into 课程实验 1.dbo.FILM values( 7,'钢铁侠','科幻','斯皮尔伯格',134,'Y',96 )
update 课程实验 1.dbo.FILM set GRADE=100 where FNAME='钢铁侠'
delete from 课程实验 1.dbo.FILM where FNAME='钢铁侠'
select * into 课程实验 1.dbo.YOUNG_ACTOR  from 课程实验 1.dbo.ACTOR where BYEAR>=1990
select * from 课程实验 1.dbo.Table_1 where a='放'
create view dbo.LEADER_80(ACTID,ANAME,BYEAR,MCOUNT,MGRADE) as
    select actor.ACTID,actor.ANAME,actor.BYEAR,
    count(*),max(actin.GRADE) from 课程实验 1.dbo.ACTOR actor,
    课程实验 1.dbo.ACTIN actin where actor.ACTID=actin.ACTID and actin.ISLEADING='Y'
    and actor.BYEAR>=1980 group by actor.ACTID,actor.BYEAR,actor.ANAME

USE [课程实验 1]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[FILM_TRIGGER]
    ON [dbo].[FILM]
    AFTER INSERT,UPDATE
AS
BEGIN
    update FILM set FTYPE='喜剧' where exists
    (select * from inserted where inserted.FID=FILM.FID and inserted.DNAME='周星驰')
END
```

第三部分：

```
1)
select s.* from dbo.SHOW s,dbo.THEATER t
where t.TAREA='洪山区' and s.YEAR=2017
and t.TID=s.TID
and s.FID in (select FID from dbo.FILM where FNAME='战狼')
order by s.MONTH desc
2)
select f.* from dbo.FILM f
where not exists (select * from dbo.ACTIN tmp where tmp.FID=f.FID)
order by f.FTYPE,f.GRADE desc
3)
select f.FID,f.FNAME,f.DNAME from dbo.FILM f
where 2017<all (select s.YEAR from dbo.SHOW s where s.FID=f.FID)
4)
select f.FID from dbo.FILM f where not exists
(select t.TID from dbo.THEATER t where t.TID not in
    (select s.TID from dbo.SHOW s where s.FID=f.FID)
)
5)
select f.FID,f.FNAME,f.DNAME,f.GRADE from dbo.FILM f
where f.GRADE not between 80 and 89
6)
select max(f.GRADE),min(f.GRADE),f.DNAME from dbo.FILM f group by f.DNAME
7)
select f.DNAME,count(f.FID) film_count from dbo.FILM f
group by f.DNAME
```

```

having count(f.FID)>=2
8)
select f.DNAME,count(f.FID),avg(f.GRADE) from dbo.FILM f
where f.DNAME in
(select f2.DNAME from dbo.FILM f2 where f2.GRADE>=80 group by f2.DNAME having count(f2.FID)>=2)
group by f.DNAME
9)
select f.DNAME,a.ACTID,a.ANAME from dbo.ACTOR a,dbo.ACTIN ai,dbo.FILM f
where ai.FID=f.FID and ai.ACTID=a.ACTID and f.DNAME in
(select f.DNAME from dbo.FILM f
group by f.DNAME
having count(*)>=2)
group by f.DNAME,a.ACTID,a.ANAME
10)
select a.ANAME, avg(ai.GRADE) from dbo.FILM f,dbo.ACTIN ai,dbo.ACTOR a
where f.FID=ai.FID and ai.ACTID=a.ACTID and ai.ISLEADING='Y'
group by ai.ACTID,a.ANAME
11)
select f.FNAME,s.YEAR,s.MONTH from dbo.SHOW s,dbo.FILM f
where s.YEAR in (select min(s1.YEAR) from dbo.SHOW s1 where s1.FID=f.FID)
and s.MONTH in (select min(s2.MONTH) from dbo.SHOW s2 where s2.FID=f.FID and s2.YEAR in
(select min(s1.YEAR) from dbo.SHOW s1 where s1.FID=f.FID)
)and f.GRADE>90
12)
select f.FNAME,s.YEAR,s.MONTH,s.TID from dbo.SHOW s,dbo.FILM f
where s.YEAR in (select min(s1.YEAR) from dbo.SHOW s1 where s1.FID=f.FID)
and s.MONTH in (select min(s2.MONTH) from dbo.SHOW s2 where s2.FID=f.FID and s2.YEAR in
(select min(s1.YEAR) from dbo.SHOW s1 where s1.FID=f.FID)
)and f.GRADE>90
13)
select f.FNAME, count(f.FID) from dbo.FILM f,dbo.SHOW s
where f.FID=s.FID
group by f.FID,f.FNAME
14)
select f.DNAME from dbo.FILM f
where f.FTYPE in ('警匪','枪战','动作')
15)
select f.FID,f.FNAME,t.TNAME,s.YEAR,s.MONTH from dbo.FILM f,dbo.SHOW s,dbo.THEATER t
where f.FID=s.FID and s.TID=t.TID and f.FNAME like '战狼%'
order by f.FNAME
16)
select t.TID,t.TNAME from dbo.THEATER t
where exists
(select * from dbo.SHOW s where s.YEAR in(select s2.YEAR from dbo.SHOW s2 where s2.FID=2 and
s2.TID=t.TID)
and s.MONTH in (select s3.MONTH from dbo.SHOW s3 where s3.FID=2 and s3.TID=t.TID)
and s.FID=1
)
group by t.TID,t.TNAME
17)
select a.ACTID,a.ANAME from dbo.ACTIN ai,dbo.ACTOR a
where 85<=all
(select f.GRADE from dbo.FILM f where f.FID in
(select FID from dbo.ACTIN ai where ai.ACTID=a.ACTID)
)
group by a.ACTID,a.ANAME
18)
select a.ANAME from dbo.ACTOR a

```

```

where not exists
( select * from dbo.FILM f where f.DNAME='吴宇森' and f.FID not in
( select ai.FID from dbo.ACTIN ai where ai.ACTID=a.ACTID )
)
19)
select a.ACTID,a.ANAME,f.FNAME from (dbo.ACTOR a left join dbo.ACTIN ai
on a.ACTID=ai.ACTID) left join dbo.FILM f on f.FID=ai.FID
group by a.ACTID,a.ANAME,f.FNAME
20)
select f.FID,f.FNAME from dbo.FILM f,dbo.SHOW s
where f.GRADE is NULL and s.FID=f.FID
group by f.FID,f.FNAME
having count(*)>3

```