**南 开 大 学数据库系统原理大作业论文**

中文题目：电影购票系统

学 号： 2013852

姓 名： 郁万祥

年 级： 2020 级

专 业： 软件工程

学 院： 软件学院

指导教师： 李旭东

完成日期： 2022 年 6 月 23 日

摘 要

随着互联网不断发展，大量数据出现在人们的眼前，为了更好的储存、管理这些数据，数据库概念应运而生，其中关系型数据库是目前应用最为广泛的类型。经过一学期对于数据库系统原理的学习后，经过总结与反思，完成了本次数据库课程报告，这次课程报告的主要任务是基于关系型数据库而设计的电影购票系统，设计过程包括了基本业务的分析，建立E-R模型，使用范式理论进行模型优化，以及对于触发器，储存过程等本学期课程所学到知识的应用与总结。

同时，考虑到本学期同时进行Java语言与应用的学习，电影购票系统在关系型数据库设计的基础上，本次项目同时完成开发了完整的应用系统，此次电影购票系统的开发具有较好的完整性。

关键字：关系型数据库；E-R模型；范式；MySQL；触发器；存储函数

Abstract

With the continuous development of the Internet, a large number of data appear in front of people. In order to better store and manage these data, the concept of database came into being, of which relational database is the most widely used type. After a semester of study on the principle of database system, after summary and reflection, this database course report has been completed. The main task of this course report is to design a movie ticket purchasing system based on relational database. The design process includes the analysis of basic business, the establishment of E-R model, the use of paradigm theory for model optimization, and the use of triggers, Application and summary of the knowledge learned in the course of this semester, such as storage process.

At the same time, considering the simultaneous learning of Java language and application in this semester, the film ticketing system has completed the development of a complete application system based on the design of relational database. The development of the film ticketing system has good integrity.

Keywords：relational database; E-R model; Paradigm; MySQL；Trigger; Storage function

目录

摘要 1

Abstract 2

1. 分析与设计..................................................................................5

1.1 业务需求.................................................................................5

1.2 E-R模型...................................................................................6

1.3 关系数据库E-R模型..............................................................7

1.4 范式理论.................................................................................8

1.5 DDL语句.................................................................................9

1. 案例分析....................................................................................10

2.1 视图.......................................................................................10

2.2 触发器...................................................................................10

2.3 存储函数、过程....................................................................11 2.4 事务处理...............................................................................12

2.5 统计分析...............................................................................13

1. 完整应用系统............................................................................14

3.1 系统介绍...............................................................................14

3.2 系统展示...............................................................................14

1. 总结与思考................................................................................18
2. 分析与设计

1.1业务需求

①电影院：能够储存电影院序号、电影院名字、电影院地址

②放映厅：能够储存放映厅序号、放映厅名字、所属电影院、座位容量

③影片：能够储存电影序号、电影名字、电影类型、电影简介、电影时长、电影封面图片

④电影场次：能够储存场次序号、电影院以及大厅序号、电影序号、开始时间、场次持续时间、价格

⑤评论：能够储存评论用户、评论影片名字、评论内容、评论时间，并且要求按时间排序

⑥电影票：能够存储票据序号、用户序号、电影序号、场次序号、座位信息

⑦用户：能够存储用户序号、用户名字、用户登陆密码、用户类型、余额、会员等级

通过创建不同的用户，实现不同的功能：

总管理员用户可以对所有电影院的信息进行增删改查以及对影评进行编辑；

各个电影院管理员可以对该电影院的大厅信息和放映电影信息以及电影票信息进行增删改查；

购买用户可以在购票系统中完成对某电影院中电影播放情况和电影的查询，同时可以在系统上购买电影票和选座操作

1.2 E-R模型

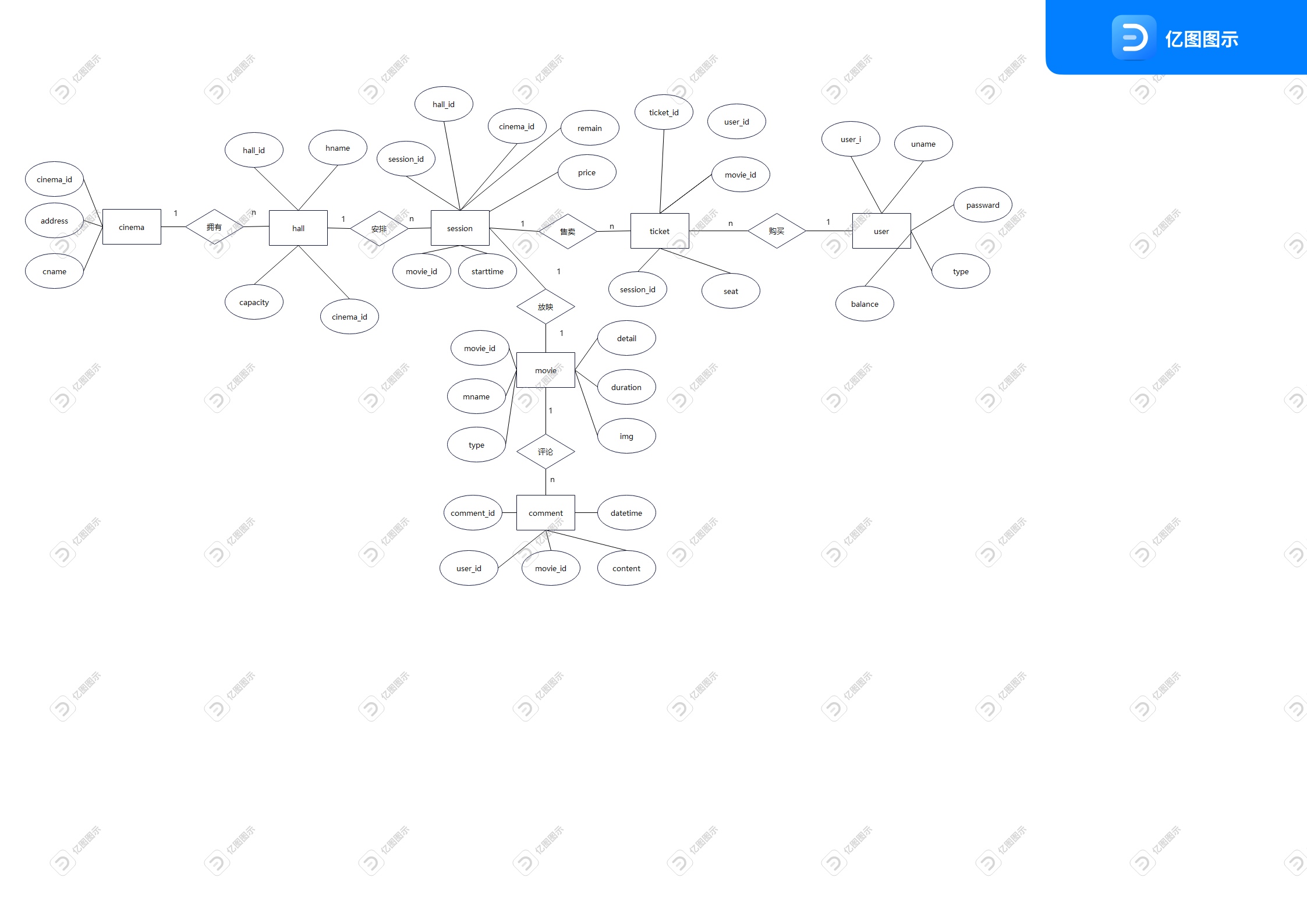


图1.1

1.3 关系数据库E-R模型

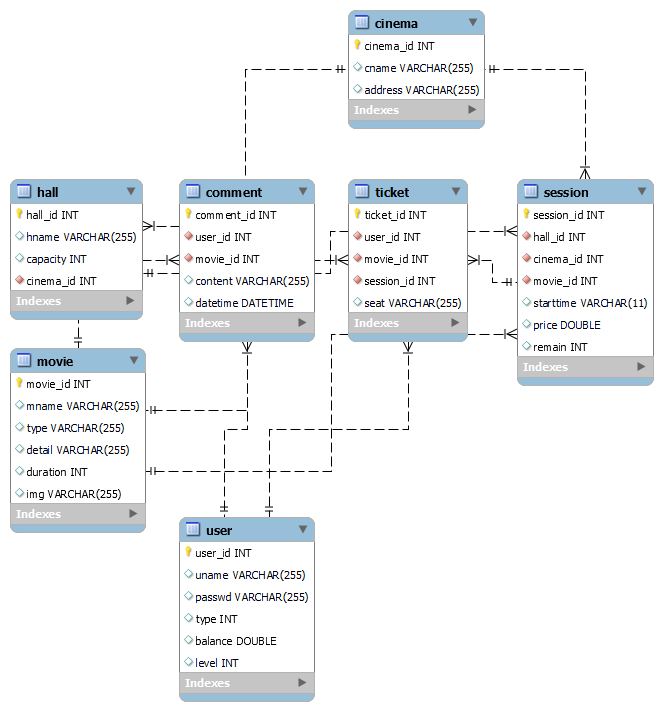


图1.2

1.4 范式理论

1NF：

数据库表的每一列都是不可分割的基本数据项

设计方案中的每一个实体集中的每一元素都不可分割，故设计方案

满足基本的1NF

2NF：

数据库中每个表必须有关键字，其他数据元素与主关键字一一对应

根据图1.2可以看出每个实体集都有关键字，并且根据实际要求，

其他数据元素据元素也应该与关键字一一对应，故设计方案满足2NF

3NF：

数据库表中的所有数据元素不但要能被唯一地被主关键字所标识，而且他

他们之间还必须相互独立，不存在其他的函数关系。

该设计方案中的每一个实体集中的元素都相互独立，故满足3NF

BCNF：

关系模式R∈1NF。若函数依赖集合F中的所有函数依赖X→Y（Y不包含于X）的左部都包含R的任一候选键，则R∈BCNF。换言之，BCNF中的所有依赖的左部都必须包含候选键。

本次数据库设计满足3NF，但是不满足BCNF，由于考虑到合理性，顾设计方案满足3NF足矣。

1.5 DDL语句

借助于MySQLworkbench，直接将关系型数据库E-R模型生成DDL语句，具体DDL语句见附录1

1. 案例分析

2.1 视图

视图是指计算机数据库中的视图，是一个虚拟表，其内容由查询定义。同真实的表一样，视图包含一系列带有名称的列和行数据。但是，视图并不在数据库中以存储的数据值集形式存在。行和列数据来自由定义视图的查询所引用的表，并且在引用视图时动态生成。

具体案例：

某店员临时需要获取正在观看某影片的用户部分信息以及此影片部分信息，生成一张视图，内容包括电影名字和用户名字。

drop view if exists users\_total

create view users\_total select user\_id,movie\_id from users natural join movie natural join session where session\_id=输入

2.2 触发器

触发器（trigger）是SQL server 提供给程序员和数据分析员来保证数据完整性的一种方法，它是与表事件相关的特殊的存储过程，它的执行不是由程序调用，也不是手工启动，而是由事件来触发，比如当对一个表进行操作（ insert，delete， update）时就会激活它执行。

具体案例：

①某用户是电影院的vip用户，不需要购票可直接进行观影，但是账户余额同时也需要扣除相应的票价，需要设计一个触发器，在观影场次中某座位如果被某vip用户占用，则触发器需要自动执行扣费操作。

②为奖励用户积极对影片进行评论，只要用户发布一条超过200字的影评，便可以得到5元的余额奖励，通过设计触发器，每次增加一条评论时，便对该评论的用户的余额进行增加五元的处理。

DELIMITER //

drop trigger if exists vip\_users;

create trigger vip\_users after insert

on ticket for each row

begin

update balance set users.balance=users.balance-ticket.price where users.user\_id=now.user\_id;

end

//

DELIMITER //

drop trigger if exists award;

create trigger award after insert

on comment for each row

begin

update users set users.balance=users.balance+5 where users.user\_id=now.user\_id;

end

//

2.3 存储函数、过程

存储函数和存储过程一样，都是在数据库中定义一些 SQL 语句的集合。存储函数可以通过 return 语句返回函数值，主要用于计算并返回一个值。而存储过程没有直接返回值，主要用于执行操作。

具体案例：将用户购票过程封装成一个存储过程，不需要返回值，传入用户信息和购票信息，在存储过程中实现购票过程。亦或者可以返回用户的余额，封装成存储函数。

DELIMITER //

drop procedure if exists buyticket;

create procedure butticket(user\_id varchar(45),move\_id varchar(45),seat,session\_id,hall\_id,cinema\_id)

begin

Delacre price int;

Select ticket.price into price from ticket where ticket.movie\_id=move\_id;

Update user set user.balance = user.balance-price where user.user\_id=user\_id;

Insert ticket values(user\_id,movie\_id,session\_id,seat)

end;

//

DELIMITER //

drop function if exists buyticket;

create function butticket(user\_id varchar(45),move\_id varchar(45),seat,session\_id,hall\_id,cinema\_id)

Returns int;

begin

Delacre price int;

Select ticket.price into price from ticket where ticket.movie\_id=move\_id;

Update user set user.balance = user.balance-price where user.user\_id=user\_id;

Insert ticket values(user\_id,movie\_id,session\_id,seat)

Return balance;

end;

//

2.4 事务处理

数据库事务( transaction)是访问并可能操作各种数据项的一个数据库操作序列，这些操作要么全部执行,要么全部不执行，是一个不可分割的工作单位。事务由事务开始与事务结束之间执行的全部数据库操作组成。

具体案例：用户之间的余额赠送功能，假如张三要给李四转账200元，这个转账会涉及到两个关键操作就是：将张三的余额减少200元，将李四的余额增加200元。如果两个操作之间突然出现错误，例如银行系统崩溃导致张三余额减少，而李四的余额没有增加，这样的系统是有问题的。事务就是保证这两个关键操作要么都成功，要么都要失败。

2.5 统计分析

统计分析是一种数据处理方式，有时我们需要对数据的总数进行统计，或者对于满足某种条件的数据进行总数分析，这时候我们需要进行统计分析。

具体案例：

①为了了解电影院的经营情况，需要每天对观影人数进行总数统计。此使只需要简单统计所有场次的人数之和即可。

②为了调查电影院的不同消费人群的状况，需要对不同余额区间的用户进行总数统计，亦或者统计余额为某区间的总用户人数，例如统计余额为100~200之间的总人数，此时需要对统计条件进行限制。

Select count(\*) as num from ticket;

Select count(\*) as num2 from user

where user.balance >100 and user.balance <200;

1. 完整应用系统

3.1 系统介绍

本次完整应用系统的开发借助于java语言，基于Spring可视化界面的开发工具,使用开发软件eclipse进行开发，最终实现了电影购票系统的完整实现，其中功能包括：

用户的登陆界面（管理员和普通购票用户）

普通用户的注册，用户修改密码

管理员对于影院和电影场次的编辑操作

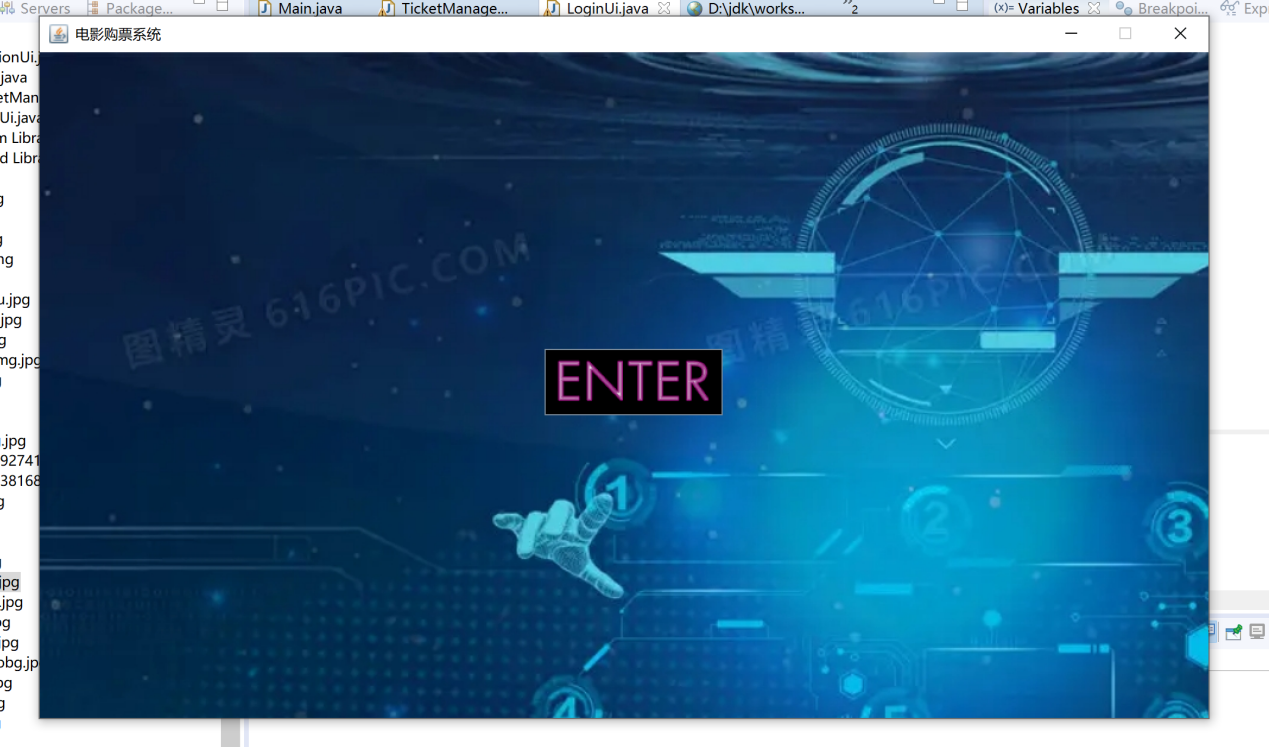
用户充值余额

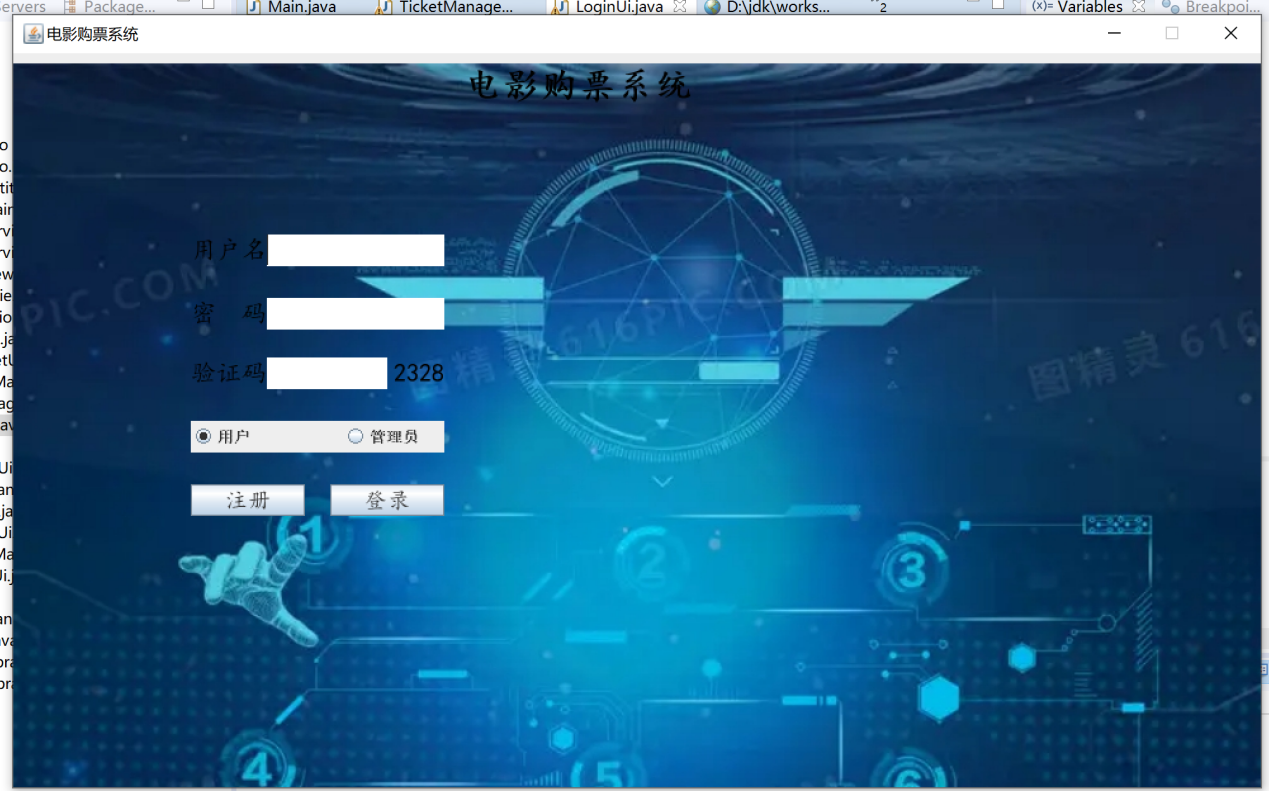
用户对于已添加影院信息和电影信息的查阅

用户在线选择电影和场次以及作为进行购票

用户观影后进行电影的评论以及可以查看其他用户影片的评论

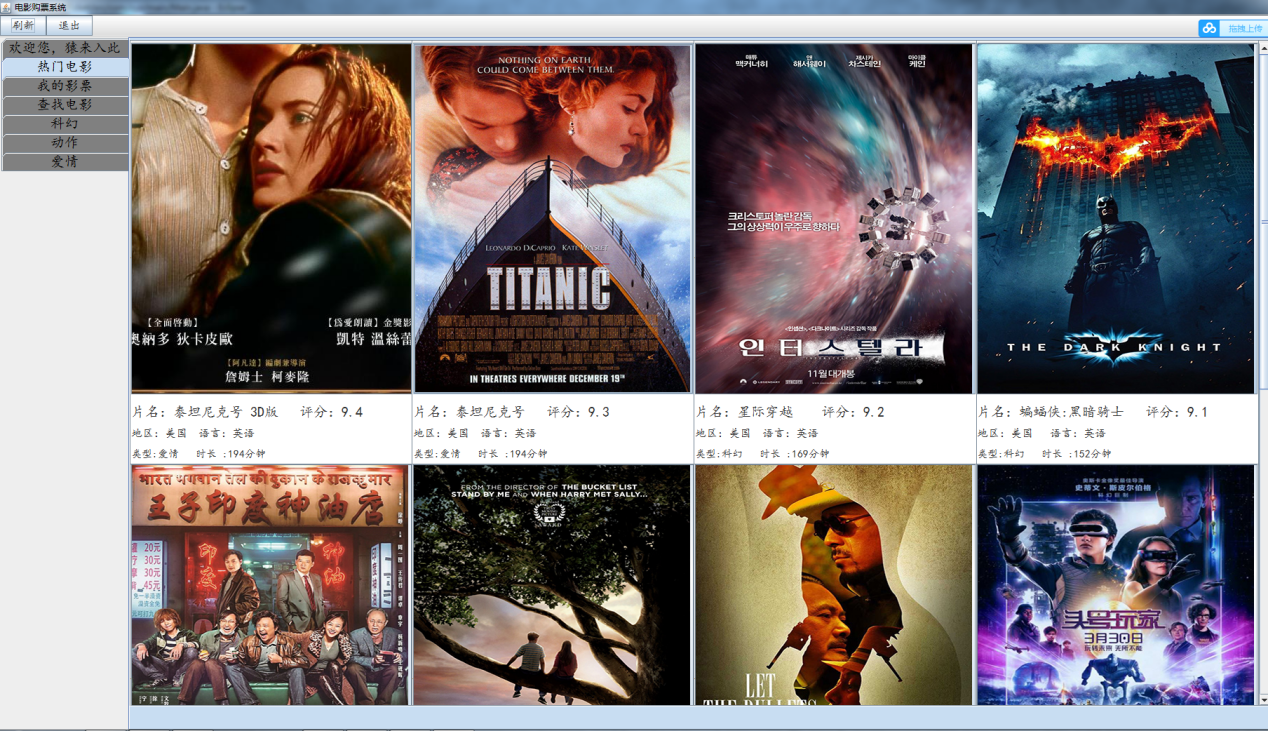
3.2 系统展示部分截图

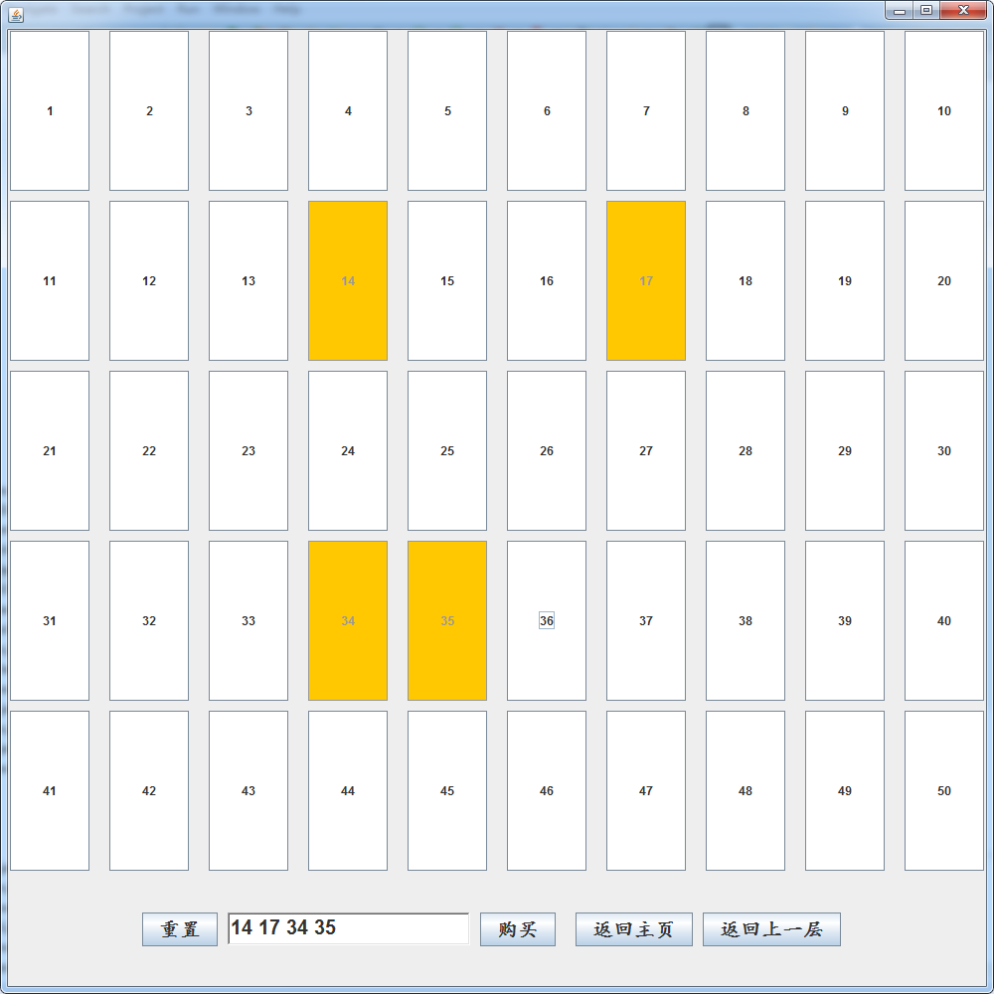


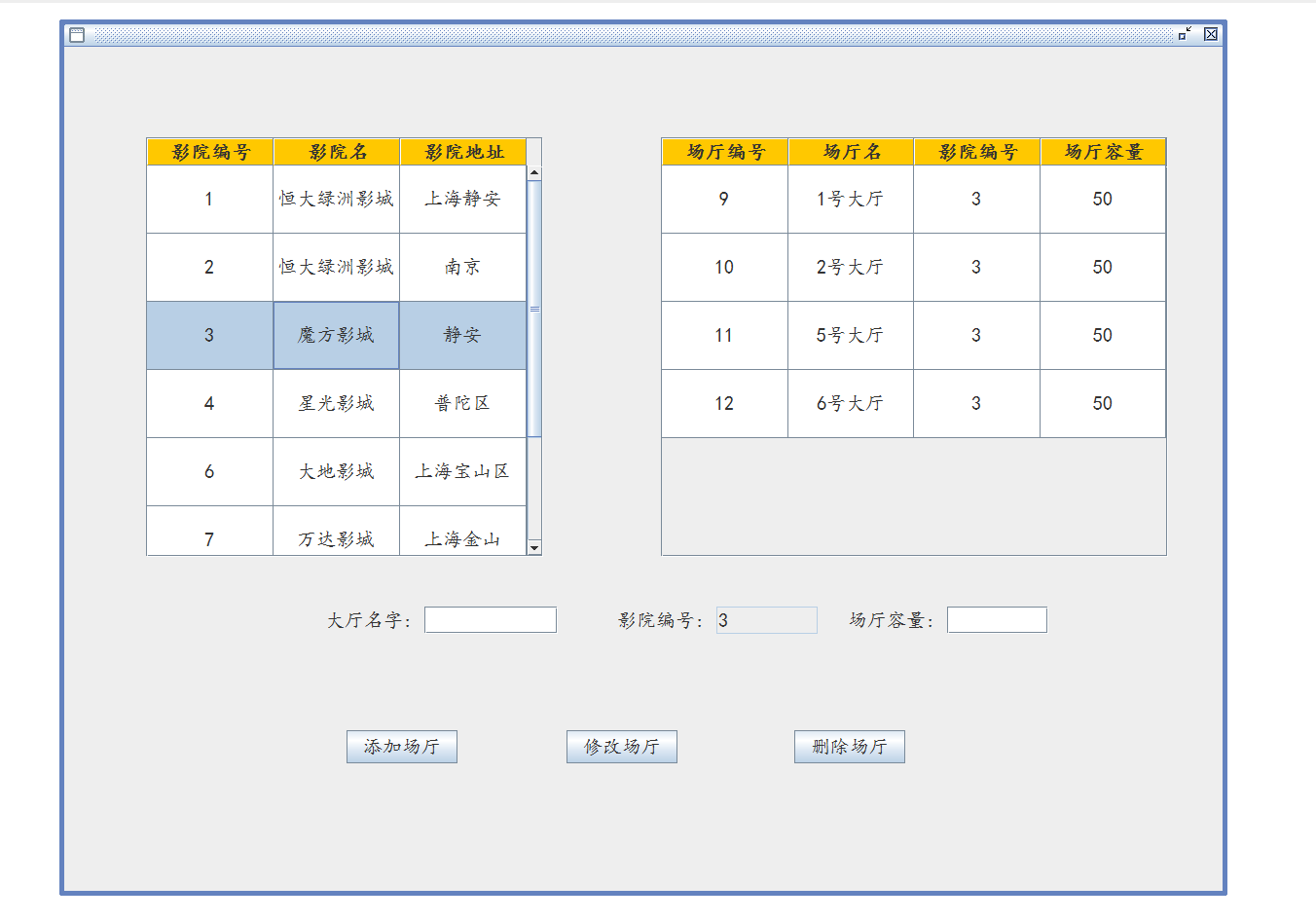












系统完整源码见附录2

1. 总结与思考

通过完整的实现电影票购票系统，从一开始的数据库ER模型构建，到使用mysqlworkbench软件构造数据库模型，以及对初步得到的数据库模型进行不断的改善，也算是基本完整的熟悉了一遍能够应用于应用系统项目中的关系型数据库的开发流程，在进行应用系统开发的过程中，我发现，其中作为数据的载体，数据库的构造直接关系到整个开发过程，只有数据库建立的合理，开发过程所用到的数据才能更好更合理方便的调取。

其中，我发现，对于类似于管理员身份的创建的涉及到安全性问题的数据，由于这类数据我们一般不允许外人进行访问，所以在这次的应用系统的创建过程中为了保障管理员的安全性，以及实现管理员界面的不同，users表中我增加了utype属性，目的就是区分管理员和普通用户。而在应用系统的开发过程中，通过创建用户界面的设计，使得管理员不可以被创建，这样可以从一定程度上保障系统的关键数据不会被随意修改。然而，如果有心之人获得了数据库的操作权限，那么可以增加管理员用户，从而对一些关键信息进行修改，那么特别是对于远程操作数据库的时候，为了保证数据安全，我们一般情况下只给出查看权限，并且通过查阅资料，我们通常使用结合访问ip白名单的控制，可以有效防止恶意修改数据甚至是删除数据。