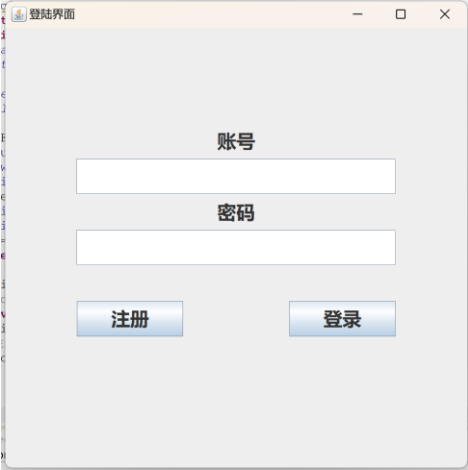

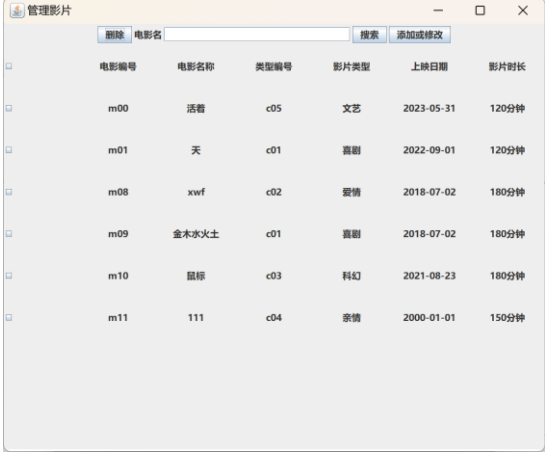


# 工程作业报告

## 1. 项目信息（10分）

学号	2110803	姓名	谢雯菲	专业	信息安全、法学双学位
项目名称	电影院售票系统				
必备环境	mySQL 5.7 java 1.7				
系统主要功能简介（4分）	实现了一个影院工作人员对于影片及其排片的管理系统。系统工作人员拥有自己的账户，可以对影片进行增删改查的操作，也可以对排片进行增删查，并且可以清晰地看到影片类型和影厅的情况。在进行这些增删改的操作时，如果有外键冲突或者不满足实际的情况发生，系统会报错。				
系统主要页面截图（6分）	<div><div></div><div></div><div></div><div></div></div>				



2. 系统配置（10 分）

说明		(2 分) 请说明系统配置情况（后台数据库，高级语言）； (8 分) 请使用连接串连接高级语言和数据库，并分析字符串的各个部分。			
配置 步骤 2 分	DBMS	1. MySQL			
	高级 语言	1. SQL			
		2. Java			
连接串 分析 (6 分)		序 号	名 称	功能说明	取 值
		1	驱动	注册驱动	Jdbc
		2	连接目 标地址	连接目标地址	localhost
		3	用户名	MySQL 用户名	root
		4	密码	MySQL 密码	
		5	目 标 数 据 库	要连接的数据库名字	cinema
		6	端口号	使用的端口号	3306

连接串代码 (截屏) (2分)	<pre>sqlHelper(){     try {         Class.forName("com.mysql.jdbc.Driver");         System.out.println("加载驱动成功");     } catch (ClassNotFoundException e) {         // TODO Auto-generated catch block         e.printStackTrace();         System.out.println("加载驱动失败");     }     try {         conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/cinema?useUnicode=true&amp;characterEncoding=UTF-8","root","");         System.out.println("连接数据库成功");     } catch (SQLException e) {         // TODO Auto-generated catch block         e.printStackTrace();         System.out.println("连接数据库失败");     } }</pre>
备注	

3. 数据库设计 (14分)

说明	(10 分) 按照数据表的创建顺序，依次给出所涉及数据表的信息，其中参照字段以“(字段 1, 字段 2, ……，字段 n)”的形式给出，被参照字段以“表名(字段 1, 字段 2, ……，字段 n)”的形式给出； (4 分) 一般 DBMS 都可以为数据库生成关系图，请将该图片截屏并粘贴到表格中。				
数据表 (10)	创建顺序	数据表名称	主键	参照属性	被参照表及属性
	1	hall	hallid	hallsize	
	2	viphall	hallid	vhstyle	Hall(hallid)
	3	category	caid	cname	
	4	users	account	Pw	
	5	movie	movid	movname	
				cid	Category(cid)
				movdate	
				Movtime	
	6	arrange		Movid	Movie.movid
Hallid				Hall.hallid	
stime					
关系图 (4)					
备注					

4. 含有事务应用的删除操作 (13分)

说明	(1分) 简要说明该操作所要完成的功能; (2分) 该操作会涉及的表(必须含有两张或两张以上的关系表, 同时以“表名”的形式给出) (1分) 表连接涉及字段描述(描述方式为“表 1. 属性=表 2. 属性”) (1分) 删除条件涉及的字段描述(以“表名. 属性=? ”形式给出)
----	--

	(4 分) 实现该操作的关键代码 (高级语言、SQL), 截图即可; (其中如果删除语句中不包含任何形式的事务应用将扣除 3 分) (4 分) 如何执行该操作, 按所述方法能够正常演示程序则给分。	
功能描述 (1 分)	可以对影片信息以及相关的排片信息进行批量删除操作。	
涉及的表 (2 分)	Movie, arrange	
表连接涉及字段 (1 分)	Movie.movid=arrange.movid	
删除条件 字段描述 (1 分)	字段	规则
	Movie.movid	如果 movie.movid 为所选择的 movid, 就删除 movie 表中有关的记录
	Arrange.movid	如果 arrange.movid 为所选择的 movid, 就删除 arrange 表中有关的记录
代码 (4 分)	<pre> //删除影片函数 boolean delMovie() throws SQLException{     conn.setAutoCommit(false);     boolean flag=false;     String sql1="delete from arrange where movid=?";     String sql2="delete from movie where movid=?";     PreparedStatement pstmt1 = null;     PreparedStatement pstmt2=null;     try {         pstmt1=conn.prepareStatement(sql1);         pstmt2=conn.prepareStatement(sql2);         for(int i=0;i&lt;MovieInfor.inforNum;i++){             if(MovieInfor.movList[i].isSel.isSelected()){                 pstmt1.setString(1, MovieInfor.movList[i].lMovid.getText());                 pstmt2.setString(1, MovieInfor.movList[i].lMovid.getText());                 System.out.println(MovieInfor.movList[i].lMovid.getText());                 pstmt1.addBatch();                 pstmt2.addBatch();             }         }         pstmt1.executeBatch();//批量执行         pstmt2.executeBatch();//批量执行         conn.commit();//提交事务         flag=true;     } catch (SQLException e) {         try {             e.printStackTrace();             conn.rollback();//失败就进行回滚操作             System.out.println("失败");         } catch (SQLException e1) {             // TODO Auto-generated catch block             e1.printStackTrace();         }     }     finally{         pstmt1.close();         pstmt2.close();     }     conn.setAutoCommit(true);     return flag; } </pre>	
程序演示 (4 分)	选择管理影片界面, 勾选想要删除的影片, 点击删除按钮, 则删除有关影片及排片记录。	
备注		

## 5. 触发器控制下的添加操作（20 分）

说明	（1 分）简要说明该操作所要完成的功能； （2 分）简要说明该触发器所要完成的功能 （1 分）该操作会涉及的表（以“表名”的形式给出）。 （2 分）该操作输入数据以及输入数据应该满足的条件，如：数值范围、是否为空； （6 分）实现该操作的关键代码（高级语言、SQL），截图即可； （8 分）如何执行该操作，按所述方法能够正常演示程序则给分。	
功能描述 （1 分）	向排片表中插入新的行。	
触发器描述 （2 分）	在向排片表中插入新的行之前，先对外键 movid 和 hallid 以及排片的时间进行检查，如果外键 movid 在 movie 表中存在，外键 hallid 在 hall 表中存在，且排片时间不冲突，则正常插入，否则触发器报错。	
涉及的表 （1 分）	Arrange, movie, hall	
输入数据 （2 分）	字段	规则
	Arrange.movid	Movid 的值在 movie 表中存在
	Arrange.Hallid	Hallid 的值在 hall 表中存在
	Arrange.stime	在影厅相同时，排片时间不能冲突
插入操作 源码 （3 分）	<pre>//添加排片函数 boolean addArranges(String movid,String hallid,String stime){     String sql="insert into arrange values(?,?,?)";     PreparedStatement pstmt=null;     int row=0;     try {         pstmt = conn.prepareStatement(sql);         pstmt.setString(1, movid);         pstmt.setString(2, hallid);         pstmt.setString(3, stime);         row=pstmt.executeUpdate();         addArrange.getInstance().lTips.setText("添加排片成功");     } catch (SQLException e) {         addArrange.getInstance().lTips.setText("错误: 不存在该影片或影厅排片冲突");         // TODO Auto-generated catch block         e.printStackTrace();     }     try {         pstmt.close();     } catch (SQLException e) {         // TODO Auto-generated catch block         e.printStackTrace();     }     if(row==1){         return true;     }     return false; }</pre>	

触发器源码 (3分)	<pre>1 drop trigger addarr; 2 create trigger addarr 3 before insert on arrange for each ROW 4 BEGIN 5 if NEW.movid not in (select movid from movie) 6 or NEW.hallid not in (select hallid from hall) 7 or EXISTS (select * from (select arrange.movid,hallid,stime,( DATE_ADD(stime,INTERVAL movtime MINUTE)) as etime from arrange left join movie on arrange.movid=movie.movid) t1 where new.stime BETWEEN t1.stime and t1.etime) THEN SIGNAL SQLSTATE 'HY000'; 8 end if; 9 end;</pre>
程序演示 (4分)	正确执行：打开管理排片界面，点击添加按钮，输入影片编号、排片厅号、排片时间，如果影片厅号及排片厅号符合外键约束，且排片时间不冲突，则触发器通过，正确插入。
程序演示 (4分)	报错：打开管理排片界面，点击添加按钮，输入影片编号、排片厅号、排片时间，如果影片厅号及排片厅号不符合外键约束，或排片时间冲突，则触发器报错。
备注	

## 6. 存储过程控制下的更新操作（18 分）

说明	(1 分) 简要说明该操作所要完成的功能; (1 分) 简要说明该存储过程所要完成的功能; (2 分) 说明该操作涉及操作的表(必须包含两张或两张以上的关系表,以“表名形式”描述) (1 分) 表连接涉及字段描述(描述方式为“表 1. 属性=表 2. 属性”) (2 分) 该操作会修改字段(以“表名. 字段名”的形式给出),以及修改规则,如新数值的计算方法、在何种条件下予以修改等; (6 分) 实现该操作的关键代码(高级语言、SQL),截图即可; (5 分) 如何执行该操作,按所述方法能够正常演示程序则给分。	
功能描述 (1 分)	实现对于影片的增加或修改。	
存储过程 功能描述 (1 分)	分析增加或修改的内容是否符合要求:如果电影存在,则其电影编号和电影名称应该一致,如果电影类型存在,则其类型编号和类型应该一致,否则将不能修改;满足上述条件可以对影片的其他字段进行更新。如果不存在,则添加新的影片和电影类型。	
涉及的关系表 (2 分)	Movie, category	
表连接涉及 及字段 (1)	Movie.cid=category.cid	
更改字段 (2 分)	字段	规则
	Movie.cid	如果电影编号与电影名称一致,则更新 cid
	Movie.movdate	如果电影编号与电影名称一致,则更新 movdate
	Movie.movtime	如果电影编号与电影名称一致,则更新 movtime
	Movie.movid	如果电影不存在,添加新的 movid
	Movie.movname	如果电影不存在,添加新的 movname
	Category.cid	如果类型不存在,添加新的 cid
	Category.cname	如果类型不存在,添加新的 cname
更新代码 (3 分)	<pre>BEGIN   UPDATE movie set cid=mycid,movdate=mymovdate,movtime=mymovtime where movid=mymovid;   set rtn=1; END;</pre>	

创建存储  
过程源码  
(3 分)

```
1 delimiter //
2 create procedure addmov
3 (in mymovid varchar(5),
4 in mymovname varchar(20),
5 in mycid varchar(5),
6 in mymovdate DATE,
7 in mymovtime SMALLINT,
8 in mycname varchar(10),
9 out rtn int)
10 BEGIN
11 if EXISTS (select * from category where cid=mycid) then
12 BEGIN
13 if mycname NOT in (select cname from category where cid=mycid) then
14 BEGIN
15 set rtn=0;
16 END;
17 ELSE
18 BEGIN
19 if EXISTS (select * from movie where movid=mymovid) then
20 BEGIN
21 if mymovname in (select movname from movie where movid=mymovid) then
22 BEGIN
23 UPDATE movie set cid=mycid,movdate=mymovdate,movtime=mymovtime where movid=mymovid;
24 set rtn=1;
25 END;
26 ELSE
27 BEGIN
28 set rtn=2;
29 END;
30 end if;
31 END;
32 ELSE
33 BEGIN
34 INSERT into movie values (mymovid,mymovname,mycid,mymovdate,mymovtime);
35 set rtn=3;
36 END;
37 end if;
38 END;
39 END if;
40 END;
41 ELSE
42 BEGIN
43 insert into category values (mycid,mycname);
44 if EXISTS (select * from movie where movid=mymovid) then
45 BEGIN
46 if mymovname in (select movname from movie where movid=mymovid) then
47 BEGIN
48 UPDATE movie set cid=mycid,movdate=mymovdate,movtime=mymovtime where movid=mymovid;
49 set rtn=1;
50 END;
51 ELSE
52 BEGIN
53 set rtn=2;
54 END;
55 end if;
56 END;
57 ELSE
58 BEGIN
59 INSERT into movie values (mymovid,mymovname,mycid,mymovdate,mymovtime);
60 set rtn=3;
61 END;
62 end if;
63 END;
64 end if;
65 END;
66 //
67 delimiter ;
```

存储过程  
执行源码  
(1 分)

```
//添加或修改影片函数
int addMovies(String movid,String movname,String movc,String movcid,String movdate,String movtime) throws SQLException{
String sql0="set @movid='"+movid+"'";
String sql1="set @movname='"+movname+"'";
String sql2="set @cid='"+movcid+"'";
String sql3="set @movdate='"+movdate+"'";
String sql4="set @movtime='"+movtime+"'";
String sql5="set @cname='"+movc+"'";
String sql6="call addmov(@movid,@movname,@cid,@movdate,@movtime,@cname,@rtn) ";
String sql7="select @rtn";
Statement stmt=conn.createStatement();
stmt.executeQuery(sql0);
stmt.executeQuery(sql1);
stmt.executeQuery(sql2);
stmt.executeQuery(sql3);
stmt.executeQuery(sql4);
stmt.executeQuery(sql5);
stmt.executeQuery(sql6);
ResultSet rst=stmt.executeQuery(sql7);
int rtn=0;
if(rst.next()){
rtn=rst.getInt(1);
}
stmt.close();
return rtn;
}
```



程序演示 (2 分)	选择管理影片界面，点击修改或添加按钮，输入电影编号与电影名称、电影类型编号和电影类型对应或全新的电影类型的影片，更新内容成功。
程序演示 (2 分)	选择管理影片界面，点击修改或添加按钮，输入电影编号与电影名称相悖，或电影类型编号与电影类型相悖的影片，更新失败，系统报错。
备注	

## 7. 含有视图的查询操作（15 分）

说明	<p>（1 分）简要说明该操作所要完成的功能；</p> <p>（1 分）简要说明建立的该视图的功能；</p> <p>（2 分）简要说明该操作涉及的关系数据表（以“表名”的形式给出）</p> <p>（1 分）简要说明表连接涉及的字段（以“表 1. 属性=表 2. 属性”）</p> <p>（6 分）实现该操作的关键代码（高级语言、SQL），截图即可；</p> <p>（4 分）如何执行该操作，按所述方法能够正常演示程序则给分。</p>
操作功能描述(1分)	根据输入字段查找具有该关键词的影片的排片记录。
视图功能描述(1分)	查询需要将 arrange 表、movie 表和 category 表连接,并取出 movid, movname, cname, hallid, stime 的字段, 此视图实现了三张表的连接和该功能, 简化了查询。
涉及的关系表(2分)	Arrange, movie, category
表连接字段（1 分）	<p>Arrange.movid=movie.movid</p> <p>Movie.cid=category.cid</p>
创建视图代码(3分)	<pre>1 create view arr as (select movname,cname,hallid,stime, arrange.movid from arrange left join movie on arrange.movid=movie.movid left join category on movie.cid=category.cid)</pre>
查询代码（3 分）	<pre>//搜索排片函数 int searchArrange(String key) throws Exception{     if(key.equals("")){         return getArrange();     }     else{         String sql="select * from arr where movname like '%" +key+"%";         Statement stmt=conn.createStatement();         ResultSet rst=stmt.executeQuery(sql);         System.out.println(sql);         int num=0;         while(rst.next()){             String movname=rst.getString(1);             String cname=rst.getString(2);             String hallid=rst.getString(3);             String stime=rst.getString(4);             String movid=rst.getString(5);             ArrangeInfor.setArrange(num++, movname, cname, hallid, stime,movid);         }         stmt.close();         ArrangeInfor.inforNum=num;         return num;     } }</pre>
程序演示（4 分）	打开管理排片界面,在搜索框输入关键词,点击搜索按钮,如果已经排片的电影中有电影名称中包含该关键词,则会显示所有包含该关键词的电影的排片。
备注	