

脱口秀演出票务系统-设计文档

1120190699 关晓宇

3.1	作业声明.....	2
3.2	需求分析	
3.2.1	开发系统需求.....	2
3.2.2	软件功能设计.....	2
3.2.3	软件运行环境.....	2
3.3	数据库设计	
3.3.1	整体框架.....	2
3.3.2	ER 图.....	4
3.3.3	关系模式规范.....	5
3.3.4	数据表设计.....	6
3.4	软件编码	
3.4.1	数据库创建和表单建立.....	7
3.4.2	预置信息插入.....	8
3.4.3	界面级联.....	9
3.4.4	权限管理、触发器与存储过程.....	10
3.4.5	窗口设计与功能实现.....	11
3.5	附件说明.....	12

3.1 作业声明

我承诺本次作业全部由我个人完成。

3.2 需求分析

3.2.1 开发系统需求

操作系统：windows

数据库：sql server

SSH 工具：sql server management studio 18

应用程序开发工具：Delphi 2010

3.2.2 软件功能设计

软件需要实现三个大模块功能。

身份验证模块：登录脱口秀演出票务系统共有两种身份，管理员和普通用户。两种身份都能够注册账号、设置密码、修改密码、查询演出信息。

管理员权限模块：管理员能够修改、删除演出信息、查询某场演出的售卖情况。管理员应该能够录入新演员并删除已有演员，应该能够查询已有演员的年龄，应该能够按月份查询已有演员的生日。管理员应该能够查询并安排某场演出的演员上场次序。

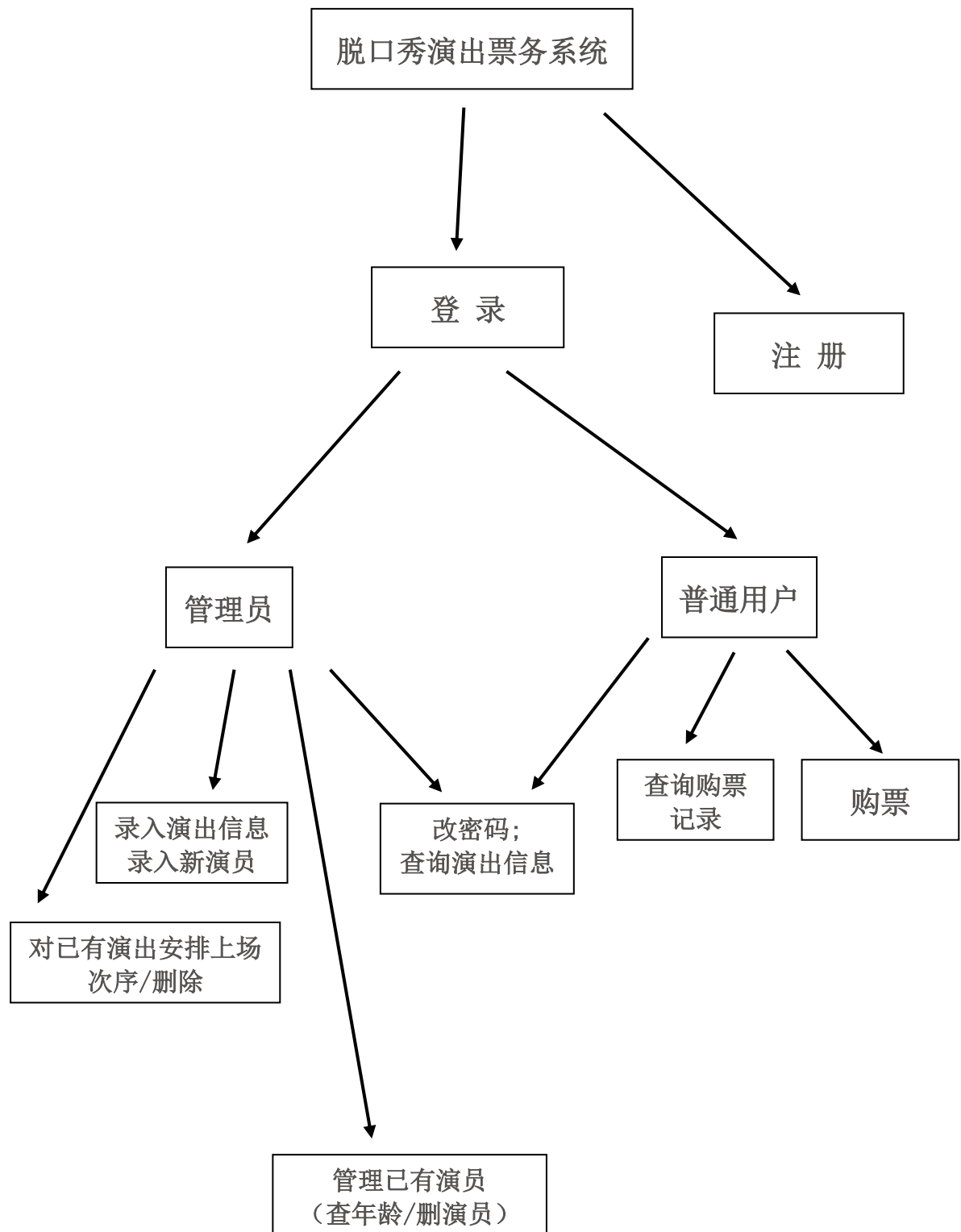
观众购票模块：管理员和普通用户均可以查询演出时间、地点、票价、余票，用户可以购票，管理员不能购票。用户能够以任意数量购买正在售卖的演出票，由于脱口秀演出座位为演出当日先到先得，购买演出票时不需要选择座位。

3.2.3 软件运行环境

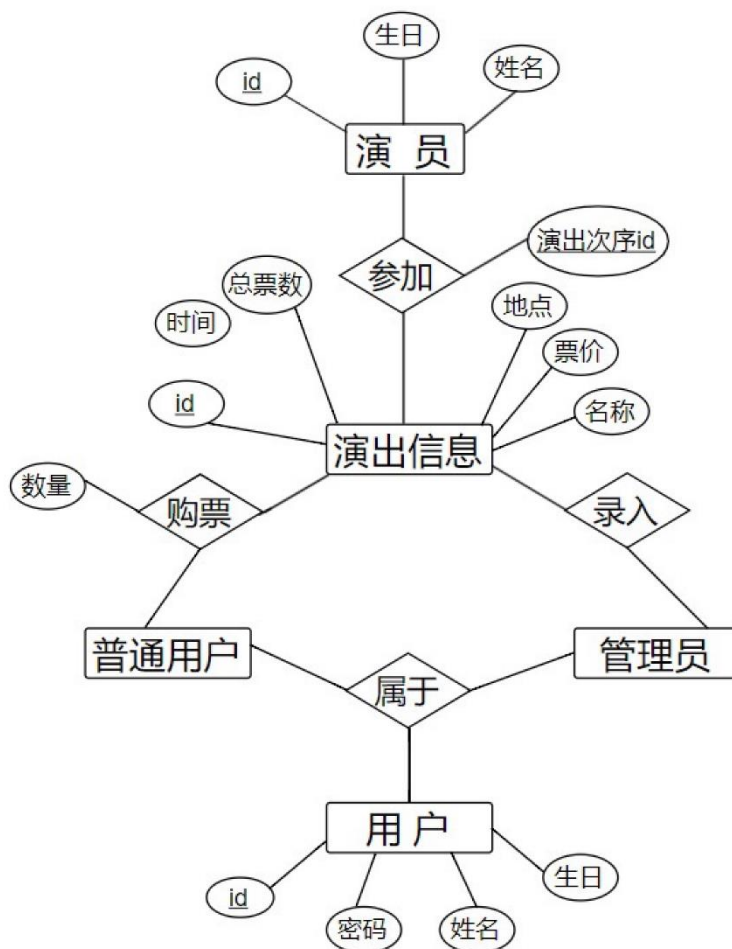
由于最后生成的是可执行 exe 文件，使用 Windows 操作系统即可运行。

3.3 数据库设计

3.3.1 整体框架



3.3.2 ER 图



3.3.3 关系模式规范

由 ER 图中抽象出关系模型，经合并冗余信息，结合功能需求优化得下述关系模式：

用户类型（类型 id，类型名）

用户信息（用户 id，用户类型编号，账号密码，用户姓名，用户生日）

演出信息（演出场次 id，演出时间，演出地点，演出票价，演出票总数）

演员信息（演员 id，演员生日，演员姓名）

参演信息（演员 id，演出场次 id，演出次序编号）

购票信息（用户 id，演出场次 id，购票数量，购票是否成功）

下面分析六个关系模式的函数依赖，证明它们符合 BCNF 或第三范式：

用户类型关系函数没有函数依赖，符合 BCNF。

用户信息关系中 **key** 为用户 **id**，所有其他属性都决定于用户 **id**，符合 BCNF。

演出信息关系中 **key** 为演出场次 **id**，所有其他属性都决定于演出场次 **id**，符合 BCNF。

演员信息关系中 **key** 为演员 **id**，所有其他属性都决定于演员 **id**，符合 BCNF。

参演信息关系中 **key** 为演出次序编号+演出场次 **id**，存在函数依赖演出次序编号 **id**、演出场次 **id**→演员 **id**，左部是 **superkey**，符合 BCNF。

购票信息关系中没有函数依赖，符合 BCNF。

3.3.4 数据表设计

用户类型表 **user_type**

字段名	字段含义	字段类型	主外键	NULL	备注
Id	类型标识编号	int	PK	NOT NULL	
Typename	类型名	varchar(12)		NOT NULL	观众/普通用户、主办方/管理员

用户信息表 **users**

字段名	字段含义	字段类型	主外键	NULL	备注
ID	用户账号	int	PK	NOT NULL	
Typeid	用户类型标识编号	int	FK	NOT NULL	引用 user_type 表中的 id
Username	用户姓名	varchar(12)			
pw	密码	varchar(12)		NOT NULL	
birthday	用户出生日期	date		NOT NULL	

演出信息表 shows

字段名	字段含义	字段类型	主外键	NULL	备注
Id	演出场次标识 id	int	PK	NOT NULL	
showtime	演出时间	date		NOT NULL	
place	演出地点	varchar(50)		NOT NULL	
price	演出票价	int		NOT NULL	
title	演出名称	varchar(50)			
ticket	演出票总数	int		NOT NULL	

演出人员信息表 performer

字段名	字段含义	字段类型	主外键	NULL	备注
Id	演出人员编号	int	PK	NOT NULL	
pName	演出人员姓名	varchar(12)		NOT NULL	
birthday	演出人员出生日期	date			

参演表 who_onshow

字段名	字段含义	字段类型	主外键	NULL	备注
Pid	演出场次标识 id	int	FK	NOT NULL	引用 shows 表的 id
onid	演出人员编号	int	FK	NOT NULL	引用 performer 表的 id
onsequence	上场顺序	int		NOT NULL	

购票信息表 sells

字段名	字段含义	字段类型	主外键	NULL	备注
Aid	购票用户账号	int	FK	NOT NULL	引用 users 表的 id
Pid	演出场次 id	int	FK	NOT NULL	引用 shows 表的 id
number	购票数量	int		NOT NULL	
buystatus	购票是否成功	bit		NOT NULL	成功/失败

3.4 软件编码

软件所有详细编码均附在源码压缩包中，此处只对部分重要过程进行说明与展示。

3.4.1 数据库创建和表单建立

创建数据库，命名为 bit_pc，依照上部分设计的数据表创建表单共六张，设置主外键和非空约束、添加字段说明。对建立好的数据库查询其数据字典，结果如下：

	表名	表说明	字段序号	字段名	字段说明	主键	外键	类型	长度	允许空
1	user_type	用户类型表	1	id	用户类型标识编号	✓		int	10	NO
2			2	typename	用户类型类型名(观众/普通用户、主办方/管理员)			varchar	12	NO
3	users	用户信息表	1	id	用户账号	✓		int	10	NO
4			2	typeid	用户类型标识编号		✓	int	10	NO
5			3	username	用户姓名			varchar	12	YES
6			4	pw	账号密码			varchar	12	NO
7			5	birthday	用户出生日期			date	10	NO
8	shows	演出信息表	1	id	演出场次标识id	✓		int	10	NO
9			2	showtime	演出时间			date	10	NO
10			3	place	演出地点			varchar	12	NO
11			4	price	演出票价			int	10	NO
12			5	title	演出名称			varchar	50	YES
13			6	ticket	演出票总数			int	10	NO
14	performer	演出人...	1	id	演出人员编号	✓		int	10	NO
15			2	pname	演出人员姓名			varchar	12	NO
16			3	birthday	演出人员出生日期			date	10	YES
17	who_on...	参演表	1	pid	演出场次标识id		✓	int	10	NO
18			2	onid	演出人员编号		✓	int	10	NO
19			3	onseq...	上场顺序			int	10	NO
20	sells	购票信息表	1	aid	用户账号		✓	int	10	NO
21			2	pid	演出场次标识id		✓	int	10	NO
22			3	number	购票数量			int	10	NO
23			4	buyst...	购票是否成功			bit	1	NO

3.4.2 预置信息插入

在表单中插入一些预置数据，方便后续进行效果展示和编码正确性检验，执行插入后，单独查询六张表单数据大致效果如下：

用户类型表 user_type

	用户类型标识编号	用户类型类型名
1	1	管理员
2	2	普通用户

用户信息表 users

	用户账号	用户类型标识编号	用户姓名	密码	用户出生日期
1	1213389134	2	桶同	tongtong!	2002-06-01
2	1520511410	1	关晓宇	1520511410	2001-06-23
3	1523006999	2	树Y	tree1062	2001-01-29

演出信息表 shows

	演出场次标识id	演出时间	演出地点	演出名称	演出票价	演出票总数
1	1	2021-05-05	北京	五月五号北京	55	100
2	2	2021-07-04	北京	七月四号北京	88	100
3	3	2021-07-04	广州	七月四号广州	88	100
4	4	2021-07-04	南京	七月四号南京	88	100
5	5	2021-06-23	北京	六月二十三北京	66	100
6	6	2021-05-05	广州	五月五号广州	55	100
7	7	2021-05-07	北京	五月七号北京	57	100

演出人员信息表 performer

	演出人员编号	演出人员姓名	出生日期
1	1	庞博	1989-02-22
2	2	豆豆	1996-01-22
3	3	House	1992-03-20
4	4	王大哥	1988-08-26
5	5	CL	1985-05-08
6	6	思文	1986-04-30
7	7	橘莉娜	2000-01-01

参演表 who_onshow

	演出场次标识id	演出人员编号	上场顺序
1	5	1	1
2	5	2	2
3	5	3	3
4	5	4	4
5	5	5	5
6	5	6	6
7	5	7	7
8	1	2	1
9	1	2	1
10	2	4	1
11	2	5	2
12	3	6	1
13	3	1	2
14	4	2	1
15	4	7	2
16	6	7	1
17	6	1	2
18	6	4	3
19	7	1	2
20	7	5	1
21	7	7	3

购票信息表 sells

	购票用户账号	演出场次标识id	购票数量	购票是否成功
1	1523006999	2	2	1
2	1523006999	1	10	1
3	1213389134	6	10	1
4	1213389134	6	10	0
5	1213389134	5	1	1

3.4.3 界面级联

对于所有含有外键的表格，建立界面级联，演出人员信息表中的演出人员的编号被修改时，参演表中的演出人员编号也要对应修改；演出信息表中的演出场次标识 id 被修改时，参演表、购票信息表中的演出场次标识 id 也要对应修改；用户类型表中的用户类型表示编号被修改时，用户信息表中的用户类型表示编号也要对应修改。除此之外，虽然用户账号不能被修改（没有用户有权限修改用户账号），但是为防止错误操作，同样对其建立级联修改操作。

由于建表时已经设置好了外键，这里我们可以直接在 **sql server** 的对象资源管理器中找出对应表单的外键属性，设置更新与修改规则为层叠 **cascade** 即可；或者也可以通过 **sql** 语句更改外键属性。对于建表时没有设置外键的，我们也可以通过 **sql** 语句重新设置外键。

示例：在演出信息表中将 id=1 修改 id=10，对应地参演表、购票信息表中 id=1 处也被修改为了 id=10。修改前后对比图如下。

演出场次标识id	演出时间	演出地点	演出名称	演出票价	演出票总数
1	2021-05-05	北京	五月五号北京	55	100
2	2021-07-04	北京	七月四号北京	88	100
3	2021-07-04	广州	七月四号广州	88	100
4	2021-07-04	南京	七月四号南京	88	100
5	2021-06-23	北京	六月二十三北京	66	100
6	2021-05-05	广州	五月五号广州	55	100
7	2021-05-07	北京	五月七号北京	57	100

购票用户账号	演出场次标识id	购票数量	购票是否成功(1成功, 0失败)
1523006999	2	2	1
1523006999	1	10	1
1213389134	6	10	1
1213389134	6	10	0
1213389134	5	1	1

演出场次标识id	演出时间	演出地点	演出名称	演出票价	演出票总数
2	2021-07-04	北京	七月四号北京	88	100
3	2021-07-04	广州	七月四号广州	88	100
4	2021-07-04	南京	七月四号南京	88	100
5	2021-06-23	北京	六月二十三北京	66	100
6	2021-05-05	广州	五月五号广州	55	100
7	2021-05-07	北京	五月七号北京	57	100
10	2021-05-05	北京	五月五号北京	55	100

购票用户账号	演出场次标识id	购票数量	购票是否成功(1成功, 0失败)
1523006999	2	2	1
1523006999	10	10	1
1213389134	6	10	1
1213389134	6	10	0
1213389134	5	1	1

示例：在演出人员信息表中删除 4 号演员王大哥，对应地参演表中该演出人员也要被删除。删除前后对比图如下。

演出人员编号	演出人员姓名	出生日期
1	庞博	1989-02-22
2	豆豆	1996-01-22
3	House	1992-03-20
4	王大哥	1988-08-26
5	CL	1985-05-08
6	思文	1986-04-30
7	橘莉娜	2000-01-01

演出场次标识id	演出人员编号	演出人员姓名	上场顺序
1	5	庞博	1
2	5	2	2
3	5	3	3
4	5	4	4
5	5	5	5
6	5	6	6
7	5	7	7
8	10	2	1
9	10	2	1
10	4	2	1
11	4	7	2
12	6	7	1
13	6	1	2
14	6	4	3
15	7	1	2
16	7	5	1
17	7	7	3

演出人员编号	演出人员姓名	出生日期
1	庞博	1989-02-22
2	豆豆	1996-01-22
3	House	1992-03-20
4	王大哥	1988-08-26
5	CL	1985-05-08
6	思文	1986-04-30

演出场次标识id	演出人员编号	演出人员姓名	上场顺序
5	1	庞博	1
5	2	豆豆	2
5	3	House	3
5	4	王大哥	4
5	5	CL	5
5	6	思文	6
10	2	豆豆	1
10	2	豆豆	1
4	2	豆豆	1
6	1	庞博	2
6	4	王大哥	3
7	1	庞博	2
7	5	CL	1

通过实验可以看出，对于所有含有外键的表格，界面级联操作能够顺利完成。

3.4.4 权限管理、触发器与存储过程

由于只有管理员能够录入、修改演出信息，我们在 sql server 中对演出信息表 shows、参演表 who_onshow、演出人员信息表 performer 建立 update，delete 和 insert 的触发器，限制只有当用户的 typeid 为管理员类型时，才有权限在这三张表中修改、插入或删除信息。

同时，为了简化一些将来要用到的查询过程，方便后续 delphi 中的窗口功能实现，建立了一些带参存储过程，下对其中几项进行举例。

示例：管理员查询售票情况的存储过程，查询在 2021 年 6 月 1 日之后的售票情况，正确显示余票，按演出时间排序，结果如下。

演出地点	演出时间	演出票价	总票数	余票	演出名称	标识id
北京	2021-06-23	66	100	99	六月二十三北京	5
北京	2021-07-04	88	100	98	七月四号北京	2
广州	2021-07-04	88	100	100	七月四号广州	3

示例：管理员依照选定月份，准确查询该月生日的演员们和演员们年龄的存储过程，要求获取系统当前时间，如果过了生日则年龄加一，如果没过生日年龄为当前年与生日年份的差值。要允许不带参数进行查询，查询在 8 月生日的演员，结果如下。

	演员姓名	生日	年龄	标识id
1	王大哥	1988-08-26	32	4

示例：管理员准确查询某场演出的演员上场顺序表的存储过程。查询结果示例如下。

上场顺序	演出时间	演员姓名
1	2021-07-04	思文
2	2021-07-04	庞博

3.4.5 窗口设计与功能实现

本部分是关于软件界面的设计及实现，主要在 Delphi 中完成，具体实现方式可在附件的压缩包中找到详细编码，此处只对大致过程进行说明与展示。

Delphi 是一个功能强大的快速应用程序开发工具，一个项目由许多窗口(unit)组成，窗口之间可以通过窗口属性设置或者控件属性设置实现相互跳转。实现每个窗口功能的主要工作过程为：选中合适的空间拖放到窗体中——使用 Delphi 自带的图形化设置页或者使用 pas 文件中的代码设定控件的相关属性。

本软件使 Delphi 与数据库相连接的方式是通过 Delphi 中的 ADOConnection 控件，配置好服务器名称 (bit-pc)、登录用户名 (sa) 和密码 (supreme,)，自动生成连接字符串，这样集成的软件每次运行时就会自动连接指定的数据库。Delphi 中的 ADOTable、ADOQuery、DATASource 控件可以执行 sql 命令或者

显示数据库的表单信息，完成软件设计要求。Delphi 自带的 `messagebox`、`inputquery` 等控件可以与用户鼠标点击或键盘输入操作进行交互。在项目-option 中可以设置软件名称及软件图标。

3.5 附件说明

A3 数据库编码含有本软件的 Delphi 项目源代码、创建数据库及数据的 SQL 脚本，可执行文件为 `stand_up_comedy.exe`，由 Delphi 直接生成。数据库备份文件为 `backup`。