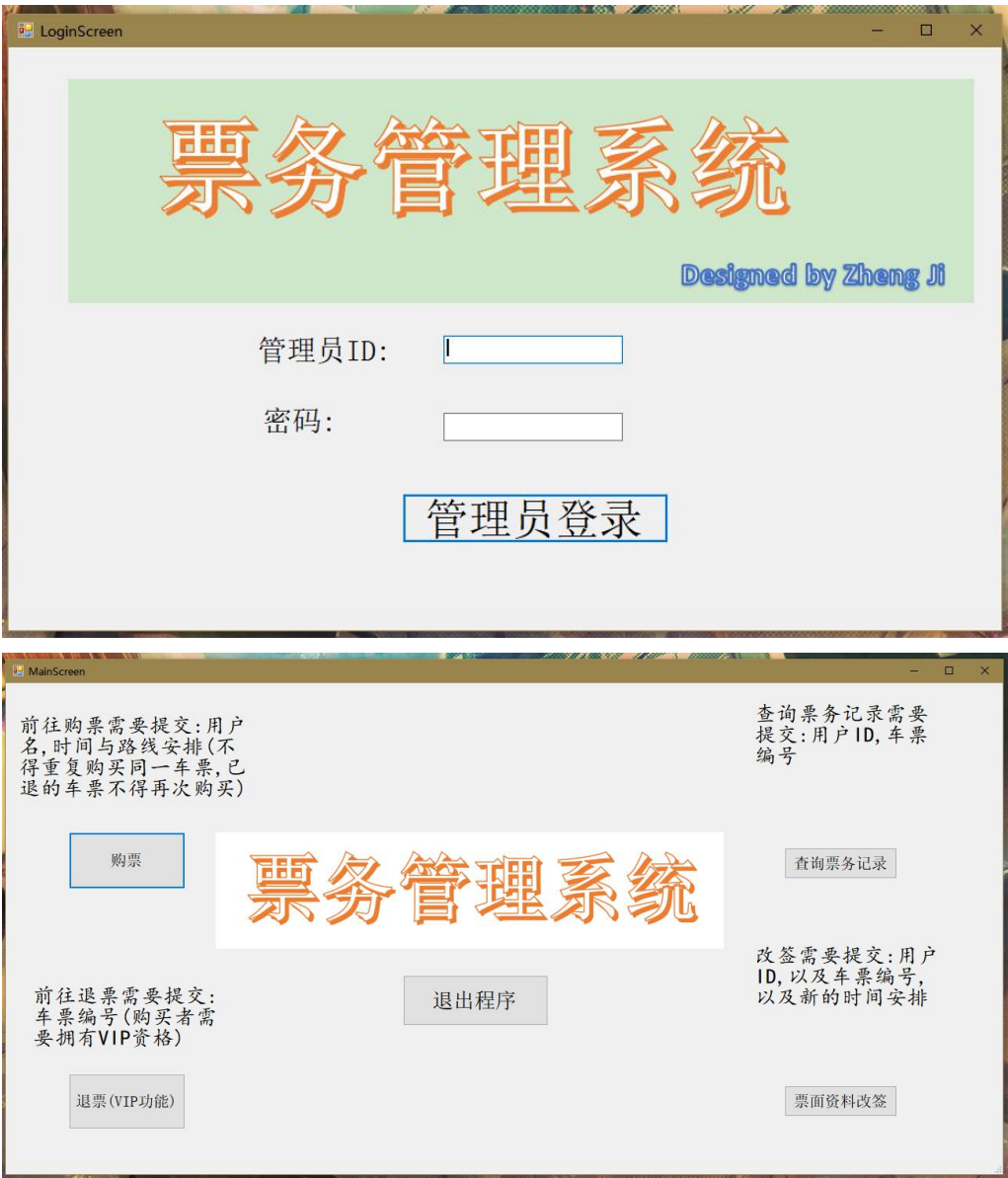


# 工程作业报告

## 1. 项目信息

学号	1811464	姓名	郑佶	专业	信息安全
项目名称	车站票务交易系统				
必备环境	windows 10 Mysql server 8.0 visual studio 2019				
系统主要功能简介(4分)	管理火车站火车站的票务信息， 实现其购票、改签、查询、退票等操作				
系统主要页面截图(6分)	 <p>The image shows two screenshots of a '票务管理系统' (Ticket Management System). The top screenshot is the 'LoginScreen' with a title bar, a large title '票务管理系统' in orange, 'Designed by Zheng Ji' in blue, and fields for '管理员ID:' and '密码:' with a '管理员登录' button. The bottom screenshot is the 'MainScreen' with a title bar, the same title '票务管理系统', and several buttons: '购票' (Purchase Ticket), '查询票务记录' (Query Ticket Record), '退票(VIP功能)' (Refund (VIP Function)), and '退出程序' (Exit Program). Text instructions are provided for each function: '购票' requires user name, time, and route; '退票' requires ticket number and VIP status; '查询' requires user ID and ticket number; '改签' requires user ID, ticket number, and new time.</p>				

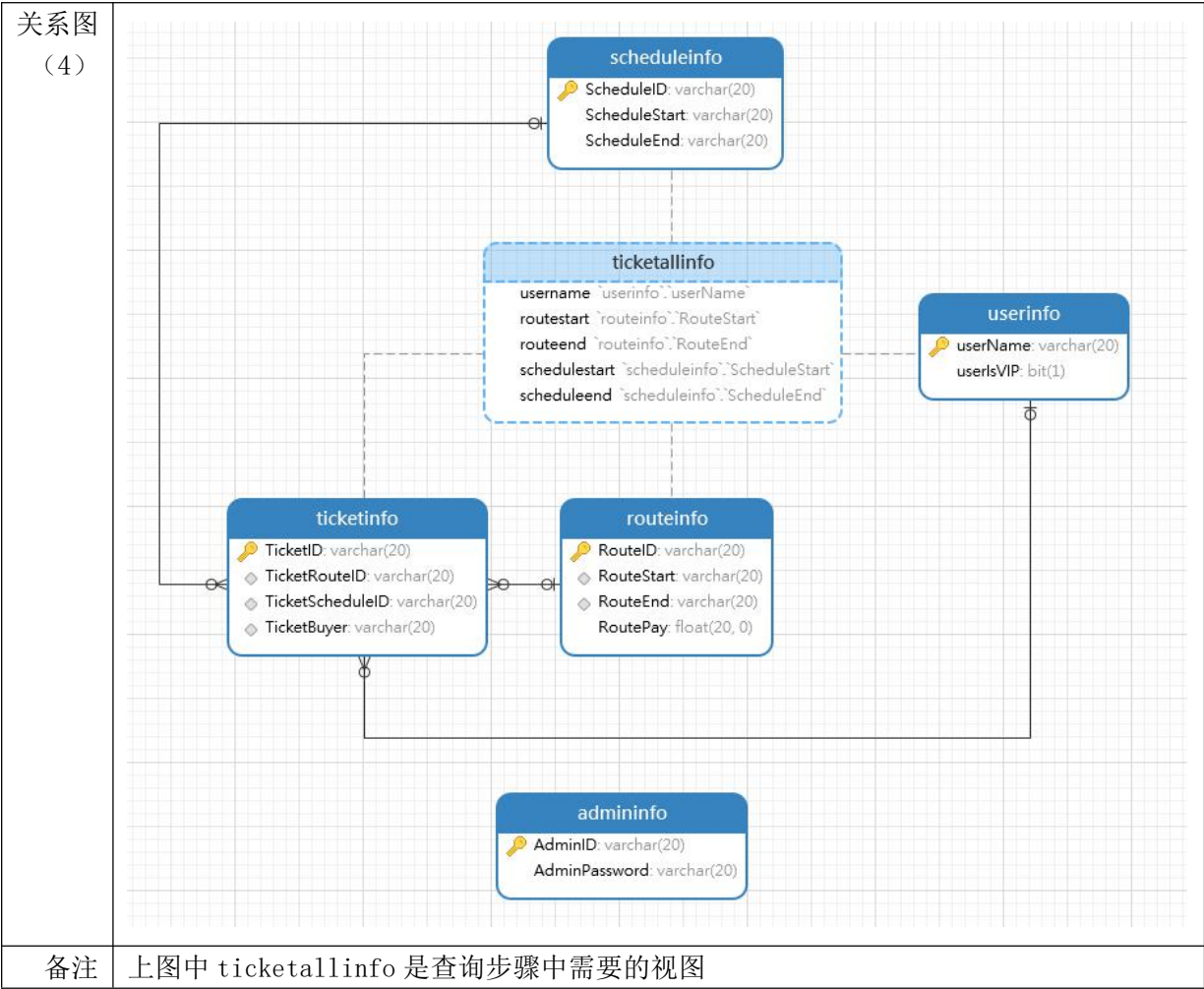
## 2. 系统配置

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	后台数据库使用 Mysql Server 8			
	高级 语言	C#			
连接串 分析 (6分)		序号	名称	功能说明	取值
		1	server	表示服务器名	localhost
		2	userid	表示用户名	root
		3	password	表示用户密码	123456
		4	database	表示数据库名	ticket_db
		5	allowuservariables	表示可以使用”@”开头的用户自定变量	True
连接串代码 (截屏) (2分)		"server=localhost;user id=root;password=123456;database=ticket_db;allowuservariables=True;"			
备注		数据库在本机, 所以服务器为 localhost; 使用 root 用户, 所以用户 id 为 root			

### 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	ticketinfo	ticketid	ticketRouteid	Routeinfo(routeid)
				ticketScheduleid	Scheduleinfo(scheduleid)
				ticketBuyer	Userinfo(username)
	2	Routeinfo	routeid		
	3	Scheduleinfo	scheduleid		



4. 含有事务应用的删除操作（13 分）

说明	<p>(1 分) 简要说明该操作所要完成的功能;</p> <p>(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)</p> <p>(1 分) 表连接涉及字段描述 (描述方式为“表 1. 属性=表 2. 属性”)</p> <p>(1 分) 删除条件涉及的字段描述 (以“表名. 属性=?”形式给出)</p> <p>(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)</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>
功能描述 (1 分)	根据给出的车票编码, 删除车票信息记录 (vip 用户可以使用)
涉及的表 (2 分)	Ticketinfo, Userinfo
表连接涉及字段 (1 分)	ticketinfo.ticketbuyer=userinfo.username

删除条件 字段描述 (1 分)	字段	规则
	ticketid	选择车票编号值为指定编号的那一行数据为删除对象
	Userisvip	在用户为 VIP 用户时才执行删除为指定编号的那一行数据
代码 (4 分)	<pre> private void BtnRefund_Click(object sender, EventArgs e) {     string refund_ticketid = ""+TxtRefund.Text.Trim()+"";     string refund_sql1 = "select ticketinfo.ticketid,userinfo.userisvip"         + " from ticketinfo,userinfo where "         + "ticketinfo.ticketid=" + refund_ticketid         + " and ticketinfo.ticketbuyer=userinfo.username;";      MySqlConnection con1 = getmysqlcon();     MySqlCommand refund_1 = new MySqlCommand(refund_sql1, con1);     con1.Open();     MySqlDataReader refund1 = refund_1.ExecuteReader();     if(!refund1.Read())     {         MessageBox.Show("该票不存在");         con1.Close();         System.Environment.Exit(0);     }     else     {         if(refund1[1].ToString()=="0")         {             MessageBox.Show("该票购买者无VIP用户资格,无法退票");             con1.Close();             System.Environment.Exit(0);         }     } } </pre>	

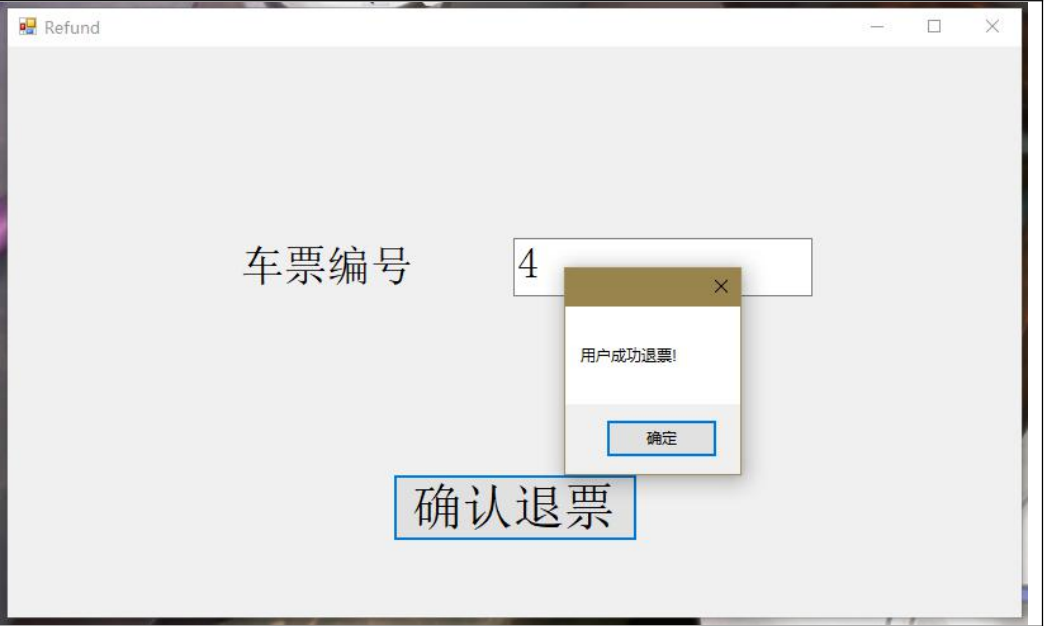
```

{
    MessageBox.Show("该票购买者无VIP用户资格,无法退票");
    con1.Close();
    System.Environment.Exit(0);
}
else
{
    con1.Close();
    string refund_sql2 =
        "begin;" +
        "DELETE FROM ticketinfo " +
        "WHERE ticketid =" + refund_ticketid + ";" +
        "commit;";
    MySqlConnection con2 = getmysqlcon();
    MySqlCommand refund2 = new MySqlCommand(refund_sql2, con2);
    con2.Open();
    int flag = refund2.ExecuteNonQuery();
    if (flag == 1)
    {
        MessageBox.Show("用户成功退票!");
        con2.Close();
        System.Environment.Exit(0);
    }
    else
    {
        MessageBox.Show("用户退票失败!");
        con2.Close();
        System.Environment.Exit(0);
    }
}
}

```

程序演示  
(4分)



	 <p>The screenshot shows a window titled "Refund". Inside, there is a label "车票编号" (Ticket Number) followed by an input field containing the number "4". A small dialog box is open in the foreground, displaying the message "用户成功退票!" (User successfully refunded!) and a "确定" (Confirm) button. Below the input field, the text "确认退票" (Confirm Refund) is highlighted with a blue box.</p>
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)	根据给出的用户名, 时间规划和路线规划购票	
触发器描述 (2 分)	当用户表中不存在该购票用户名时, 在用户表中插入该新用户资料, 默认为非 VIP 用户	
涉及的表 (1 分)	Userinfo, Ticketinfo, Scheduleinfo, Routeinfo,	
输入数据 (2 分)	字段	规则
	Username=ticketbuyer	Ticketbuyer 需要满足外键约束, 存在于 user info 表中, 若不存在, 则通过触发器在更新 ticketinfo 表前先更新 user info 表, 创建一个新用户 (默认为非 VIP 用户)
插入操作 源码 (3 分)	<pre> private void BtnExecuteBuy_Click(object sender, EventArgs e) {     Random ran = new Random();     string random_ticketid = ran.Next(100, 1000).ToString();     string _a = TxtUserName_Buy.Text.ToString();     string _b = TxtUserStartTime_Buy.Text.ToString();     string _c = TxtUserEndTime_Buy.Text.ToString();     string _d = TxtUserStartSta_Buy.Text.ToString();     string _f = TxtUserEndSta_Buy.Text.ToString();     string a = "" + _a + "";     string b = "" + _b + "";     string c = "" + _c + "";     string d = "" + _d + "";     string f = "" + _f + "";      if (TxtUserEndSta_Buy.Text == ""    TxtUserEndTime_Buy.Text == ""            TxtUserName_Buy.Text == ""    TxtUserStartSta_Buy.Text == ""            TxtUserStartTime_Buy.Text == "")     {         MessageBox.Show("任意一项都不能为空!");         return;     }     else     { </pre>	

```

string f1 = "select scheduleid from scheduleinfo where"
+ " schedulestart = " + b
+ " and scheduleend = " + c + ",";
 MySqlConnection con1 = getmysqlcon();
 MySqlCommand f__1 = new MySqlCommand(f1, con1);
 con1.Open();
 MySqlDataReader f_1 = f__1.ExecuteReader();
 if (!f_1.Read())
 {
     MessageBox.Show("时间安排不存在");
     System.Environment.Exit(0);
 }
 string the_schedule = "" + f_1[0].ToString() + "";
 con1.Close();

 string f2 = "select routeid from routeinfo where routestart = " + d
+ " and routeend = " + f + ",";
 MySqlConnection con2 = getmysqlcon();
 MySqlCommand f__2 = new MySqlCommand(f2, con2);
 con2.Open();
 MySqlDataReader f_2 = f__2.ExecuteReader();
 if (!f_2.Read())

```

```

{
    MessageBox.Show("路线安排不存在");
    System.Environment.Exit(0);
}
string the_route = "" + f_2[0].ToString() + "";
con2.Close();

string Sql1 =
    " select ticketid from ticketinfo where ticketBuyer = " + _a
+ " and ticketscheduleid= " + the_schedule
+ " and ticketrouteid= " + the_route + ",";

 MySqlConnection ms1 = getmysqlcon();
 MySqlCommand mySqlCommand1 = new MySqlCommand(Sql1, ms1);
 ms1.Open();
 MySqlDataReader reader1 = mySqlCommand1.ExecuteReader();

 if (reader1.Read())
 {
     MessageBox.Show("重复购票失败!");
     ms1.Close();
     System.Environment.Exit(0);
     return;
 }

```



	<pre>else {     ms1.Close();     string Sql2 =     " insert into ticketinfo (ticketid,ticketscheduleid,ticketrouteid,ticketbuyer) "     + "VALUES (" + random_ticketid + ","     + "(" + the_schedule + ","     + "(" + the_route + ","     + "(" + _a + "));";      MySqlConnection ms2 = getmysqlcon();     MySqlCommand mySqlCommand2 = new MySqlCommand(Sql2, ms2);     ms2.Open();     int k=mySqlCommand2.ExecuteNonQuery();     ms2.Close();     string news = "用户" + _a + "成功购买该车票!\n"     + "车辆运行时间:" + _b + "--&gt;" + _c + "\n" +     "车辆运行区间" + _d + "--&gt;" + _f + "\n" + "车票编号为:" + random_ticketid;     MessageBox.Show(news);     System.Environment.Exit(0); }</pre>
触发器源 码 (3分)	<pre>drop TRIGGER if exists `nameadd`; delimiter // create TRIGGER NameAdd before insert on ticketinfo for EACH row begin if NEW.TicketBuyer not in (select username from userinfo) then insert into userinfo(username,userisvip) values(NEW.TicketBuyer,b'0'); end if; end;// delimiter ;</pre>
程序演示 (4分)	<p>说明：不违背触发器能够执行插入操作。</p>  <p>The screenshot shows a Windows application titled 'Buy'. It contains a form with the following fields: '用户名:' (Username) with the value '李明' (Li Ming), '始发站:' (Origin Station) with '北京站' (Beijing Station), '终点站:' (Destination Station) with '天津站' (Tianjin Station), '始发时间:' (Start Time) with '2020-6-1 9:00', and '到站时间:' (Arrival Time) with '2020-6-1 12:00'. At the bottom are two buttons: '购买车票' (Buy Ticket) and '退出' (Exit). A small '确定' (Confirm) dialog box is overlaid on the right, displaying the message: '用户'李明'成功购买该车票!' (User 'Li Ming' successfully purchased this ticket!), followed by the train details: '车辆运行时间:2020-6-1 9:00--&gt;2020-6-1 12:00', '车辆运行区间:北京站--&gt;天津站' (Train route: Beijing Station --&gt; Tianjin Station), and '车票编号为:272' (Ticket number: 272).</p>

<p>程序演示 (4分)</p>	<p>说明：违背存储过程，系统报错；</p> <p>由于在更新 ticketinfo 表中，出现的不满足外键约束的问题，已经通过触发器对 userinfo 表的更新得到解决，所以即使触发了触发器条件，ticketinfo 表插入过程中不会发生报错</p> <div data-bbox="384 358 1433 855"><p>The screenshot shows a 'Buy' window with the following fields: '用户名:' (Username) with value '王大锤', '始发站:' (Origin Station) with value '北京站', '终点站:' (Destination Station) with value '天津站', '始发时间:' (Start Time) with value '2020-6-1 9:00', and '到站时间:' (Arrival Time) with value '2020-6-1 12:00'. There are two buttons: '购买车票' (Buy Ticket) and '退出' (Exit). A small dialog box is overlaid on the right, displaying the message: '用户'王大锤' 成功购买该车票! 车辆运行时间:2020-6-1 9:00--&gt; 2020-6-1 12:00 车辆运行区间:北京站--&gt;天津站 车票编号为:900'. A '确定' (Confirm) button is at the bottom of the dialog.</p></div> <p>原用户表中不存在“王大锤”这个用户名</p> <p>这个 ticketinfo 表的更新成功依赖于触发器对 userinfo 表的更新</p>
<p>备注</p>	

## 6. 存储过程控制下的更新操作（18分）

说明	<p>（1分）简要说明该操作所要完成的功能；</p> <p>（1分）简要说明该存储过程所要完成的功能；</p> <p>（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）</p> <p>（1分）表连接涉及字段描述（描述方式为“表 1. 属性=表 2. 属性”）</p> <p>（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；</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>（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。</p>	
功能描述 （1分）	根据给定的车票编号和新的时间规划, 更新票面信息	
存储过程功能描述 （1分）	若新时间规划和原规划不同, 则更新, 反之不更新	
涉及的关系表 （2分）	Scheduleinfo, ticketinfo	
表连接涉及字段（1）	Scheduleinfo.scheduleid=ticketinfo.ticketscheduleid;	
更改字段 （2分）	字段	规则
	TicketScheduleid	当新的时间规划存在于规划表中, 且于旧规划变化不同, 则更新
更新代码 （3分）	<pre>update ticketinfo set ticketscheduleid=newscheduleid where ticketid=theticketid;</pre>	

创建  
存储  
过程  
源码  
(3  
分)

```
public MySqlConnection getmysqlcon() - Sublime Text
文件(F) 编辑(E) 选择(S) 查找(F) 视图(V) 跳转(G) 工具(T) 项目(P) 首选项(N) 帮助(H)
select projectno(select budget from project where
public MySqlConnection getmysqlcon()
80
81
82 delimiter //
83 drop procedure if exists `changeinfo`;
84 create Procedure changeinfo
85 (in theticketid varchar(20),
86 in newscheduleid varchar(20),
87 out alert varchar(20))
88 begin
89 declare a varchar(20);
90 set alert= '给出新值,更新成功';
91 set a=(select ticketscheduleid from ticketinfo where ticketid=theticketid);
92 if a>newscheduleid or a<newscheduleid
93 then
94 update ticketinfo set ticketscheduleid=newscheduleid where
   ticketid=theticketid;
95 else
96 set alert='未给出新值,更新失败';
97 end if;
98 end;//
99 delimiter ;
100
Line 99, Column 12 Spaces: 4 Plain Text
```

存储  
过程  
执行  
源码  
(1  
分)

```
private void button1_Click(object sender, EventArgs e)
{
    string change_ticketid=""+TxtChange_Ticketid.Text.Trim()+"";
    string change_sql1 =
        "select * from ticketinfo where ticketid= " + change_ticketid+"";

    MySqlConnection con1 = getmysqlcon();
    MySqlCommand change_1 = new MySqlCommand(change_sql1, con1);
    con1.Open();
    MySqlDataReader change_flag1 = change_1.ExecuteReader();
    if(!change_flag1.Read())
    {
        MessageBox.Show("该车票不存在!");
        con1.Close();
        System.Environment.Exit(0);
    }
    con1.Close();

    string ScheduleStart = " ScheduleStart=" + TxtChange_StartTime.Text.Trim() + "";
    string ScheduleEnd = " ScheduleEnd=" + TxtChange_EndTime.Text.Trim() + "";
    string change_sql3 = " select scheduleid from scheduleinfo where "
        + ScheduleStart + " and " + ScheduleEnd + "";;

    MySqlConnection con3 = getmysqlcon();
    MySqlCommand change_3 = new MySqlCommand(change_sql3, con3);
    con3.Open();
    MySqlDataReader change_flag3 = change_3.ExecuteReader();
```

	<pre>if (!change_flag3.Read()) {     MessageBox.Show("该时间安排不存在!");     con3.Close();     System.Environment.Exit(0); } string find_scheduleid = "" + change_flag3[0].ToString() + ""; con3.Close();  MySQLConnection con4 = getmysqlcon(); string change_sql4 =     "call changeinfo(" + change_ticketid + "," + find_scheduleid + ",@p);" +     "select@p;"; MySQLCommand change_4 = new MySQLCommand(change_sql4, con4); con4.Open(); MySQLDataReader change_flag4 = change_4.ExecuteReader(); string a = ""; if (change_flag4.Read()) {     a = a+change_flag4[0].ToString(); } MessageBox.Show(a); con4.Close(); System.Environment.Exit(0);</pre>
程序演示 (2分)	<p>说明：不违背存储过程，能够执行更新操作</p> 
程序演示 (2分)	<p>说明：违背存储过程，系统报错； 由于之前成功插入和新值一样的旧值, 所以更新失败</p>

	<div><div>TicketChange</div><div><div>车票编号: 900</div><div>出发时间: 2020-6-1 9:00</div><div>到站时间: 2020-6-1 10:00</div><div>改签该车票</div><div>未给出新值,更新失败</div><div>确定</div></div></div>
备注	



## 7. 含有视图的查询操作（15 分）

说明	(1 分) 简要说明该操作所要完成的功能; (1 分) 简要说明建立的该视图的功能; (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) (1 分) 简要说明表连接涉及的字段 (以 “表 1. 属性=表 2. 属性”) (6 分) 实现该操作的关键代码 (高级语言、SQL), 截图即可; (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。
操作功能描述(1分)	给定车票编号, 在视图中展示车票相关信息
视图功能描述(1分)	存储车票所有相关信息
涉及的关系表(2分)	Ticketinfo, scheduleinfo, routeinfo
表连接字段 (1 分)	routeid = ticketrouteid and scheduleid = ticketscheduleid
创建视图代码(3分)	<pre>string search_sql1 =     "create view ticketallinfo as " +     "select ticketbuyer,routestart,routeend,schedulestart,scheduleend " +     "from routeinfo,scheduleinfo,ticketinfo " +     "where routeid = ticketrouteid and scheduleid = ticketscheduleid" +     " and ticketid =" + search_ticketid + " ;select * from ticketallinfo;";</pre> <p>其中, search_ticket 为要查询的目标车票编号</p>
查询代码 (3 分)	

```

private void BtnSearch_Click(object sender, EventArgs e)
{
    string search_ticketid = "" + TxtSearch.Text.Trim() + "";
    string find =
        "select * from ticketinfo where ticketid=" + search_ticketid + ",";
    MySqlConnection con_find = getmysqlcon();
    MySqlCommand _find= new MySqlCommand(find, con_find);
    con_find.Open();
    MySqlDataReader reader1= _find.ExecuteReader();
    if(!reader1.Read())
    {
        con_find.Close();
        MessageBox.Show("该车票不存在!");
        System.Environment.Exit(0);
    }

    string drop = "drop view if exists `ticketallinfo`; ";

    MySqlConnection con_drop = getmysqlcon();
    MySqlCommand _drop= new MySqlCommand(drop, con_drop);
    con_drop.Open();
    int s=_drop.ExecuteNonQuery();
    con_drop.Close();

    string search_sql1 =
        "create view ticketallinfo as " +
        "select ticketbuyer,routestart,routeend,schedulestart,scheduleend " +
        "from routeinfo,scheduleinfo,ticketinfo " +
        "where routeid = ticketrouteid and scheduleid = ticketscheduleid" +
        " and ticketid =" + search_ticketid + ";select * from ticketallinfo;";
    MySqlConnection con1 = getmysqlcon();
    con1.Open();
    MySqlCommand cmd_search = new MySqlCommand(search_sql1, con1);
    MySqlDataReader search1 = cmd_search.ExecuteReader();
    while (search1.Read())
    {
        int index = this.View1.Rows.Add();
        this.View1.Rows[index].Cells[0].Value = search1[0].ToString();
        this.View1.Rows[index].Cells[1].Value = search1[1].ToString();
        this.View1.Rows[index].Cells[2].Value = search1[2].ToString();
        this.View1.Rows[index].Cells[3].Value = search1[3].ToString();
        this.View1.Rows[index].Cells[4].Value = search1[4].ToString();
    }
}

```

程序演示  
(4 分)

车票编号:

	购买者	始发站	终点站	出发时间	到站时间
*					

查询该票信息

车票编号:

	购买者	始发站	终点站	出发时间	到站时间
▶	王大锤	北京站	天津站	2020-6-1 9:00	2020-6-1 10:00
*					

查询该票信息

备注