

# 华侨大学

## 信息科学与工程学院

### 课程设计报告

课程名称 数据库技术课程设计

---

题 目 图书馆管理信息系统

---

院(系)别 信息科学与工程学院

---

专 业 自动化

---

级 别 2016

---

学 号 1615322013

---

姓 名 李玮超

---

指导老师 黄彩虹、郑义民

---

2019 年 11 月 28 日

# 目 录

引言.....	1
第一章 概述.....	2
1.1 数据库设计背景.....	2
1.2 数据库设计目的.....	2
1.3 数据库设计内容.....	2
第二章 图书管理系统设计.....	3
2.1 需求分析.....	3
2.2 概念模式的设计.....	4
2.2.1 局部 E-R 图设计 .....	5
2.2.3 全局 E-R 图设计 .....	8
2.3 逻辑模式设计.....	9
2.3.1 逻辑模式的转换 .....	9
2.3.2 逻辑模式的规范化 .....	11
第三章 数据库的实施和维护.....	12
3.1 界面开发.....	12
3.2 读者部分代码的讲解及运行效果.....	17
3.3 管理员部分代码的讲解及运行效果.....	18
3.4 借阅部分代码的讲解及运行效果.....	23
3.5 排行榜部分代码的讲解及运行效果.....	26
3.6 报表部分实现的讲解及运行效果.....	28
总结.....	31
参考文献.....	32
附录.....	33

## 引言

《数据库课程设计》是自动化专业的重要实践课程。在系统地学习《数据库技术及应用》专业课程后，通过具体的课程设计，可以了解到数据库的基本开发流程。通过自己动手代码实践整个过程，培养自己分析问题和解决问题的能力。

关系型数据库的设计过程主要分为五个阶段：需求分析、概念模式设计、逻辑模式设计、数据库实施和数据库运行维护。在充分了解题目要求且结合现实情况后，需要合理地做出分析，然后抽象出实体各个属性与各自之间的关系。之后将其转换成机器语言并进行范式分析，最后进行数据库的实施和运行。

本次课程设计是对图书管理系统的设计，设计的过程中也严格遵循以上的五个阶段。但在实施阶段会视实际情况作相应调整。在设计完成本次数据库中，主要具备有两大功能：对于读者，数据库的设计可以满足其个人信息的查询和修改、图书信息的查询、个人借阅情况的查询和排行榜查看。对于管理员，数据库的设计可以满足其管理员个人信息的查询和修改、图书信息的维护、读者信息的维护、图书的借阅服务、读者借阅信息的维护和报表查看功能。

第一部分，将介绍图书管理系统的设计背景、设计目的和设计内容。通过简单的比较，对于图书管理数据库的整体有个基本的了解。

第二部分是本报告的重点，图书管理系统将按照关系型数据库设计的五个阶段一步一步执行去把整个数据库给建立出来。在这其中会包含一些自己对于整个图书管理数据库搭建的想法和理解。

第三部分是侧重于整个图书管理系统的相关代码的讲解和效果的演示，分成读者和管理员两部分。

最后是自身对于整个图书管理数据库开发过程的心得体会和开发总结。

## 第一章 概述

### 1.1 数据库设计背景

随着科技发展和社会进步,尤其是计算机大范围的普及,计算机应用逐渐由大规模科学计算的海量数据处理转向大规模的事务处理和对工作流的管理,以数据库管理系统为开发环境的管理信息系统在大规模的事务处理和对工作流的管理等方面的应用,特别是在图书馆图书管理系统中。

### 1.2 数据库设计目的

在早期图书馆的借阅管理系统中,经常以卡片登记的形式进行借阅信息的记录。图书相关信息的存储通常都是纸质版的。在这样的情况下,卡片、纸质材料容易遗失,经常造成重要信息丢失的情况,并且随着时间的累积所需要的空间越来越大,对于图书和读者信息分类管理和更新就显得更加困难了。

除此之外,图书馆对于图书和读者信息的处理涉及到各种不同的数据类型,在这之中数据关联复杂,统计和查询的方式各不相同。并且由于人工的原因,会出现信息的重复、错误和遗漏。而图书和读者的管理涉及到的信息相关性很强,微小的错误会导致巨大的工作量。所以开发一个实用、高效的图书管理系统是十分有必要的。这样的数据库系统不仅能占用更小的空间存储更多的数据,而且能够更加有效的对图书馆信息进行管理,节约了人力和物力,提高了工作效率。

### 1.3 数据库设计内容

运用所学的数据库技术和理论知识,对图书管理系统进行数据库设计。设计包含了需求分析、概念模式设计、逻辑模式设计、数据库实施和数据库运行维护五个阶段。数据库设计完成后,应该包含以下几个功能:

- 1、图书、读者及借阅信息的维护(增加、删除、修改和查询)
- 2、借书和还书功能(根据书号和借书证号)
- 3、图书的分类(展示具体的类型)
- 4、图书的报表功能(可选三种不同的主要信息报表)
- 5、图书的排行图表展示功能(包含图书和读者信息的排行榜)

## 第二章 图书管理系统设计

本次对于图书管理系统的设计，应该包含以下五个阶段：需求分析、概念模式式设计、逻辑模式式设计、数据库实施、数据库运行维护。如下所示：

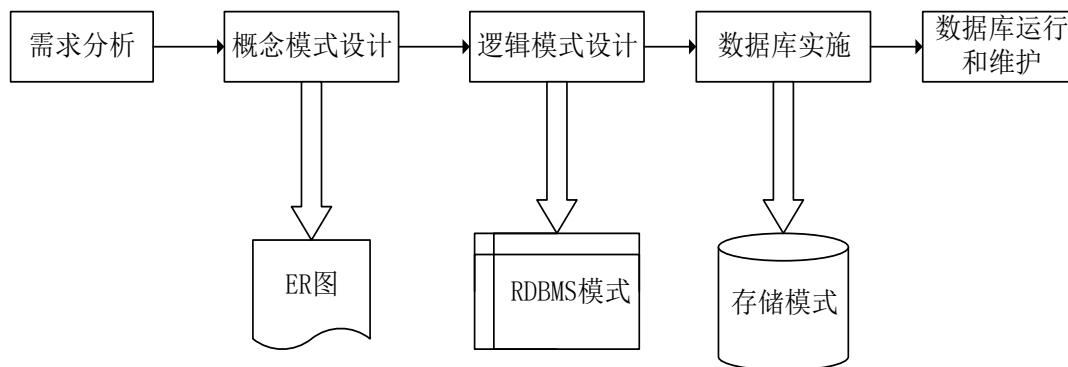


图 2.1 阶段流程图

### 2.1 需求分析

在这次《数据库课程设计》中，根据课程内容的安排，所选题目为：图书管理系统。在充分理解题目提出的功能和需求后，结合网上查阅的相关资料，设计了表的基本结构如下：

- 1、图书信息表（书号，书名，作者，出版社，出版时间，价格，总数量，剩余数量，位置，被借次数，类型）
- 2、读者信息表（借书证号，密码，姓名，性别，工作单位，借阅次数）
- 3、类型表（类型名称，类型编号）
- 4、管理员表（工号，密码，姓名，性别，电话）

其中对于密码的设置是为了方便使用同一登录界面能够登录到不同的导航页面而设置。在这里直接把密码作为实体的一个属性，而不是再创建一张用户表了。

根据上述几个表的属性，我们可以总结出来设计的图书管理系统应该同时面向管理员和读者这两类用户。对于读者，应该具有个人用户信息查询、个人信息的修改、馆藏图书信息查询、个人借阅情况的查询和排行榜显示功能。对于管理员，则需要包含对图书信息、读者信息和借阅信息的维护，即包含基本的增加、

删除、修改和查询功能。并且管理员能够根据数据库中现存储的数据生成对应期望的报表。

上面表的基本结构比较混乱，如果没有进行必要的修改，将导致出现以下几个问题：（1）数据冗余；（2）修改异常；（3）删除异常（4）插入异常。所以接下来将根据需求分析所需要的功能做后续的分析 and 设计。

下面给出简易的程序流程图：

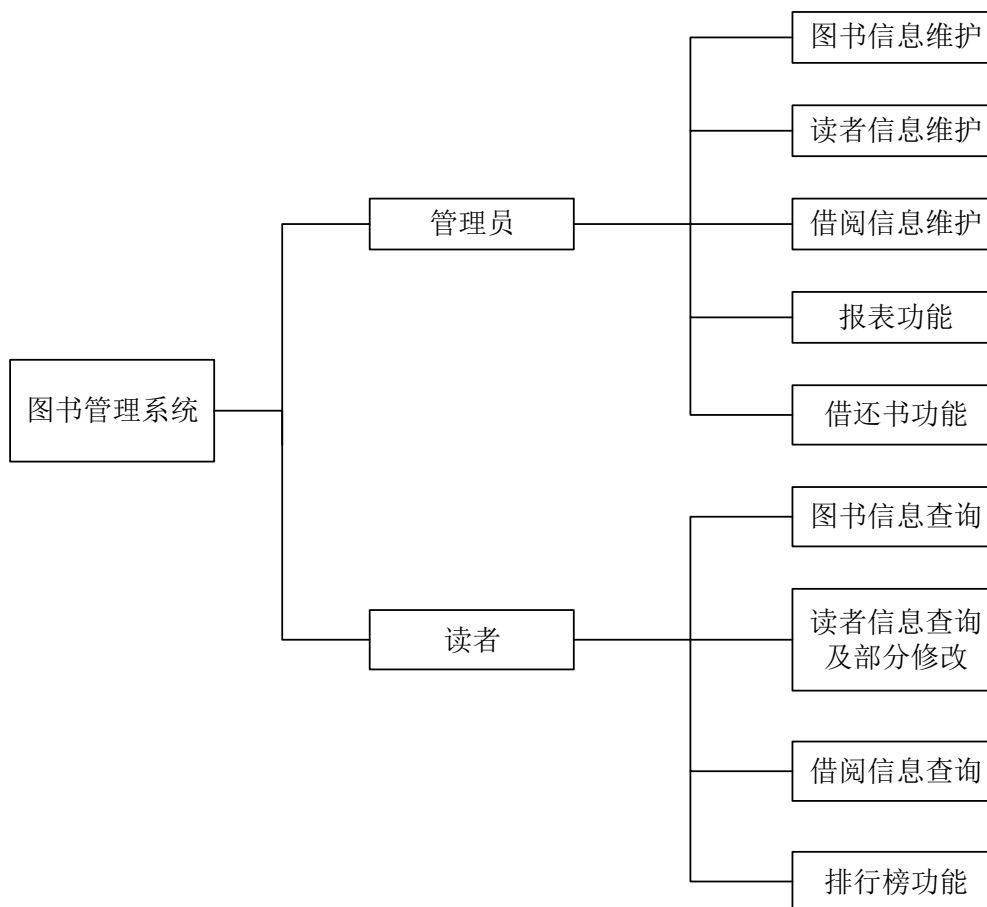


图 2.1.1 图书管理流程示意图

## 2.2 概念模式的设计

概念模式设计是完成对需求分析报告中描述的现实世界的建模，即用一种数据模型来实现对现实世界的抽象表达。这种建模与具体的机器世界、DBMS 无关，是现实世界到信息世界的第一层抽象。因此用于表达概念模型的数据模型一方面应该具有较强的语义表达能力，能够方便、直接地表达实际应用中的各种语义知识，所以我们选择实体-联系模型。

通常是先将现实世界的实物及其联系抽象为信息世界实体及联系（概念模型），然后再将其转换为计算机世界的数据库模型（关系数据库模式）。

实体-联系模型（E-R 模型）是由美籍华人陈平山提出的。这个模型直接将现实世界中的实物及其之间的联系抽象为实体类型和实体间联系，然后用实体联系图表示数据模型。

E-R 模型是用 E-R 图表示的。E-R 图中有下面四个基本成分：①矩形框：表示实体型；②菱形框：表示联系；③椭圆形框：表示实体或联系类型的属性；④直线：联系类型与其所涉及的实体之间用直线连接，实体与实体之间用直线连接。

E-R 模型的基本元素有以下几个：（1）实体：客观存在并可相互区别的事物称为实体。（2）属性：实体所具有的某一特征称为实体的属性。（3）键：也称关键字或码。唯一标识实体的最小的属性集称为实体的键。（4）联系：现实世界的事物彼此是有联系的，反映在信息世界就是实体之间的联系。

实体间的相互关系称为联系。联系在数据库中的反映是实体集之间存在着这样和那样的联系，这种联系实际上表示了实体集之间的某种函数映射关系。可分为以下三类：

①1:1 联系：已知实体集 A 和 B，若其中每个实体集中任意实体至多与另一实体集中的一个实体有联系，则称 A 和 B 的联系为“1 对 1 联系”，简记为 1:1 联系。

②1:n 联系：已知实体集 A 和 B，若 A 中每个实体可与 B 中任意个实体有联系，而 B 中每个实体至多与 A 中的一个实体有联系，则称 A 和 B 的联系为“1 对多联系”，简记为 1:n 联系。

③m:n 联系：已知实体集 A 和 B，若其中每个实体集中任一实体可与另一实体集中的多个实体有联系，则称 A 和 B 的联系为“多对多联系”，简记为 m:n 联系。

### 2.2.1 局部 E-R 图设计

针对本次实验的题目图书管理系统，我通过前期查阅相关的图书管理的案例，并且结合自身在校图书馆的体验，总结出了几个实体之间的联系。同时，我也考虑到自己数据库实践过程中的一些需求和方便，有意识地去调整了一些实体的属

性，具体各个局部 E-R 图绘制和分析如下。

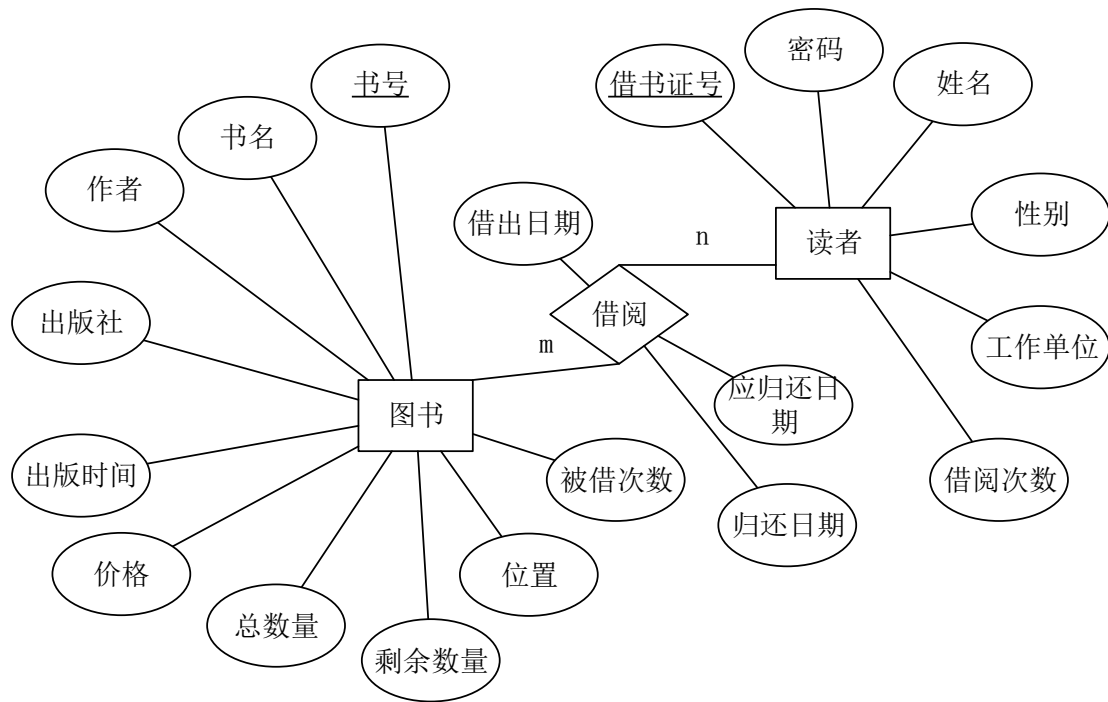


图 2.2.1 读者和图书局部 E-R 图

**分析：**从以上的读者和图书的局部 E-R 图 2.2.1 中，我们可以看到一名读者可以借多本图书，一种图书可以被多名读者借走。所以是  $n:n$  的关系。其中读者主键：借书证号；图书主键：书号。

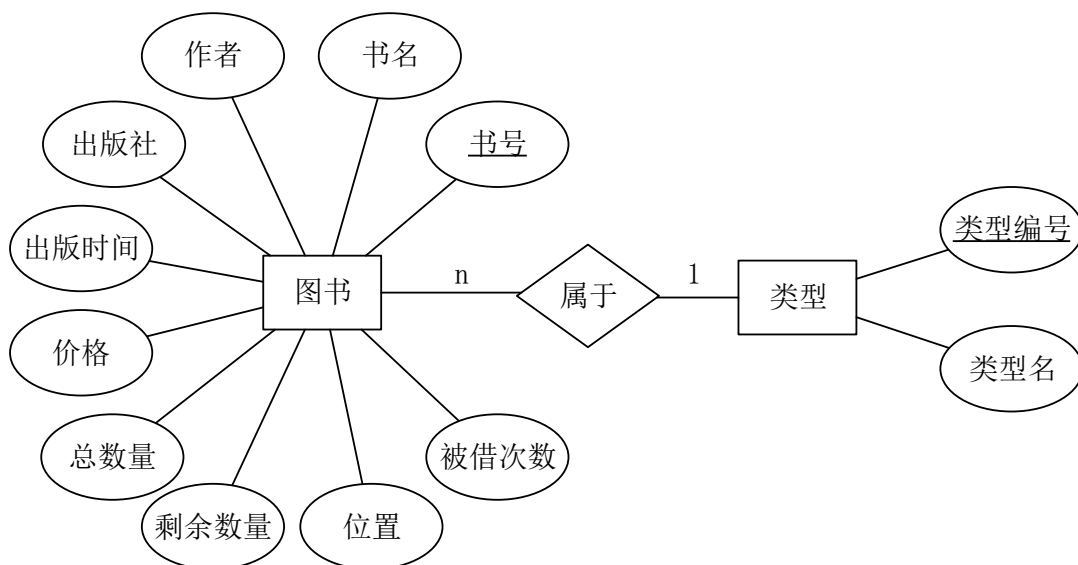


图 2.2.2 图书和类型局部 E-R 图

**分析：**从以上的图书和类型的局部 E-R 图 2.2.2 中，我们可以看到一种图书



属于一种类型，一种类型可以包含多种图书。所以是 1:n 的关系。其中图书主键：书号；类型主键：类型编号。

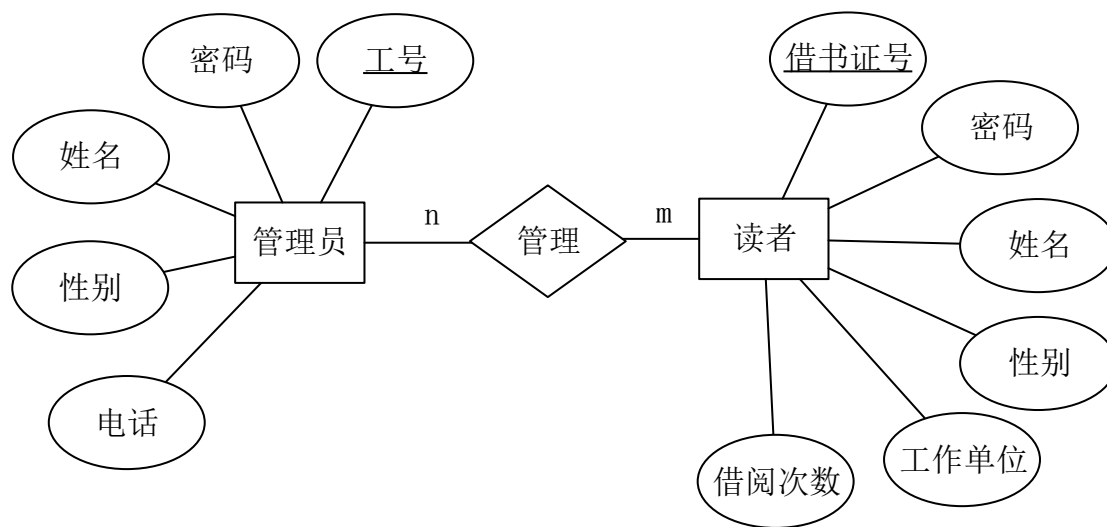


图 2.2.3 管理员和读者局部 E-R 图

**分析：**从以上的管理员和读者的局部 E-R 图 2.2.3 中，我们可以看到一个管理员可以管理多个读者，一个读者可以被多个管理员所管理。所以是 n:n 的关系。其中管理员主键：工号；读者主键：借书证号。

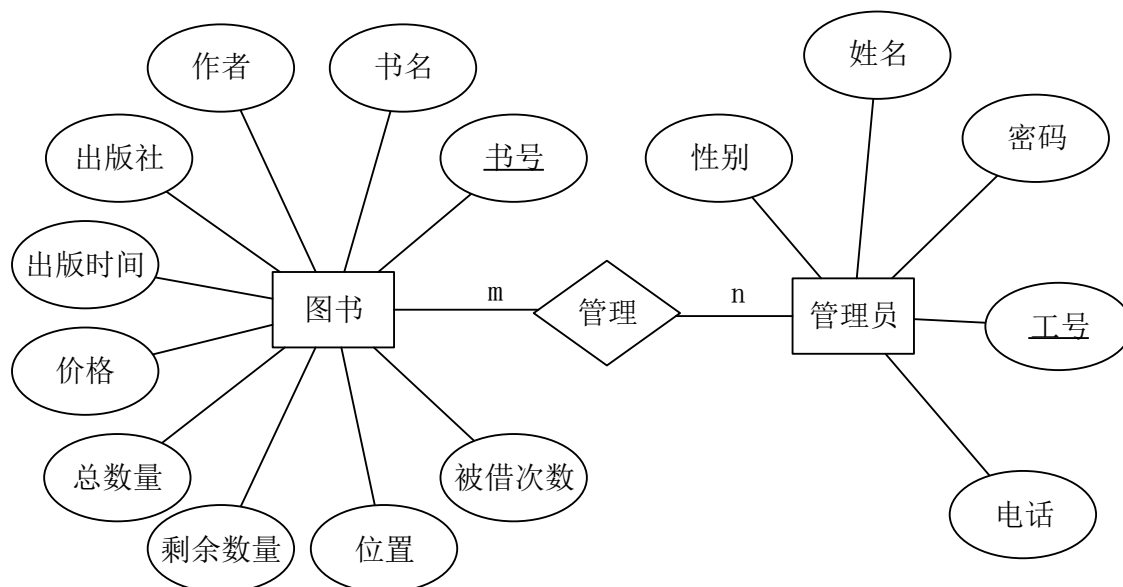


图 2.2.4 管理员和图书局部 E-R 图

**分析：**从以上的管理员和图书的局部 E-R 图 2.2.4 中，我们可以看到一个管

理员可以管理不同的图书，一种图书课可以被多个管理员所管理。所以是 n:n 的关系。其中管理员主键：工号；图书主键：书号。

### 2.2.3 全局 E-R 图设计

将局部的 E-R 图进行组合将变成全局 E-R 图，通过全局 E-R 图可以清晰地看出各个实体之间的关系。

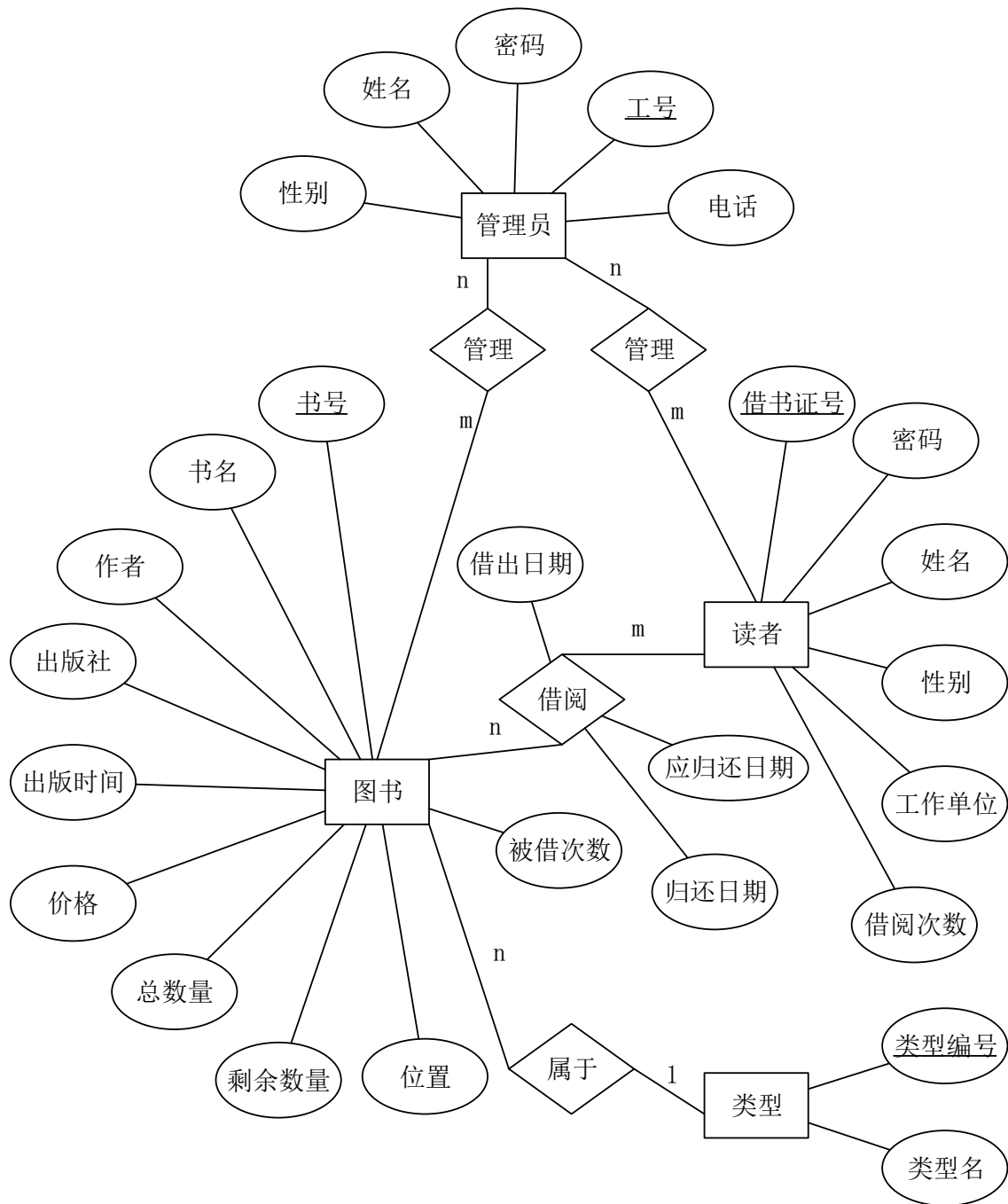


图 2.2.5 图书管理系统全局 E-R 图

## 2.3 逻辑模式设计

在数据库的概念模式设计完成后,数据库的设计进入就到了逻辑模式设计阶段。此时的数据库设计与具体的机器世界是相关联的。逻辑设计阶段有两大任务:

1、按一定的规则将 E-R 图转换成关系模式; 2、关系模式的规范化处理。所谓关系模式的规范化就是对关系模式进行某种处理,使它满足一定的规范化要求。

### 2.3.1 逻辑模式的转换

根据 E-R 模型向关系模型转换规则,对上述的 E-R 图进行转换,转换的结果如下:

1、管理员 (工号, 密码, 姓名, 性别, 电话)

**分析:** 管理员以各自的工号为区别,所以工号是该表的唯一主键,且通过 E-R 图分析不存在外键。

2、读者 (借书证号, 密码, 姓名, 性别, 工作单位, 借阅次数)

**分析:** 不同读者之间的借书证号是不同的,是唯一区别他们的标识,所以设置借书证号为该表的主键。

3、类型 (类型编号, 类型名)

**分析:** 类型名由唯一的类型编号决定,所以类型编号为其主键。

4、图书 (书号, 书名, 作者, 出版社, 出版时间, 价格, 总数量, 剩余数量, 位置, 被借次数, 类型编号)

**分析:** 由图 2.2.2 可知,图书和类型之间是 1:n 的关系。由 E-R 图的转换原则我们知道,需要在 n 端的实体加上 1 端实体的主键构成关系模型,并且加上的该主键变成对应表的外键。

5、借阅 (书号, 借书证号, 借出日期, 归还日期, 应归还日期)

**分析:** 由图 2.2.1 可知,读者和图书之间通过借阅产生关系,并且他们之间是 n:m 的关系。由 E-R 图的转换原则我们知道,需要把关系单独独立出来成为一张借阅的表。此时该表的主键是由图书的主键书号、读者的主键借书证号和自身属性的借出日期所组成的联合主键。并且再加上借阅关系所对应剩下的属性。

**补充说明:** 在图 2.2.5 的全局 E-R 图上有管理员和图书形成 n:n 的关系,有

管理员和读者形成 n:n 的关系。但在实际的设计过程中，我们采取的方案是参照图 2.1.1 的程序来设计。通过对应账户和密码形成不同的界面，从而在上层去实现管理员对图书信息和读者信息的管理，而不需要单独再形成两张关系表，使得数据库更加简洁明了。

至此，成功地把图书概念模式转换成对应的逻辑模式。这里需要特别再强调的是，在上述的 E-R 图中管理员对读者和图书进行管理的关系需要转换对应的数据库表，但在实际前台的设计过程中，由于我们可以在 VB 上直接根据不同的登录账号进入不同的登录页面，从而可以实现管理员的功能需求。所以，在实际过程中，可将产生冗余的转换表删除，但是对实现我们的功能并不影响。

对应的数据库关系图如图 2.3.1 所示：



图 2.3.1 数据库关系图

### 2.3.2 逻辑模式的规范化

我们设计的最终目的是使关系模式规范化，最低应该达到第三范式。关系模式的规范化可以消除异常的情况。

首先，上述关系中的管理员、读者、类型、图书和借阅中的每一个分量都是不可再分的数据，所以符合第一范式的要求。

其次，符合第二范式的关系必须要属于第一范式，且每个非主属性都完全依赖于主键。

在上述的关系中，在“管理员”、“读者”、“类型”和“图书”中主键有且只有一个属性，不存在部分依赖关系，属于第二范式。在关系“借阅”中由三个属性构成联合主键，其中“书号”和“借出证号”是外键，但是非主属性“归还日期”和“应归还日期”必须由“书号”、“借书证号”和“借出日期”组成的联合主键来决定，所以借阅关系属于第二范式。

最后，如果要符合第三范式，必须达到第一范式的要求，且每一个非主属性都不传递依赖于主键。在“管理员”、“读者”、“类型”、“图书”和“借阅”中不存在有传递函数依赖于主键，所以上述关系达到第三范式。

## 第三章 数据库的实施和维护

### 3.1 界面开发

图书管理系统选择在 VB.NET 环境下进行开发。对于读者，具备个人信息的查询和修改、借阅查询、图书查询和排行榜显示的功能。对于管理员来说应具备个人信息查询和修改、图书和读者信息的维护、图书的借书和还书等功能。

首先是登录界面，我们通过不同的账号识别出其管理员或者读者的身份，很方便的实现登录功能。登录界面如图 3.1.1 所示：



图 3.1.1 用户登录界面图

对读者，我选取一名读者的借书证号和密码进入个人信息页面图 3.1.2 示：



图 3.1.2 读者登录个人信息页面

可以对读者借阅情况进行查询，如图 3.1.3 所示：



图 3.1.3 借阅情况查询界面

读者可以对图书信息进行查询，如图 3.1.4 所示：



图 3.1.4 图书信息查询界面

查看使用图表的借书人和图书的排行榜，如图 3.1.5 所示：



图 3.1.5 排行榜界面

其次是管理员，对图书管理系统展示如下。选取一名管理员的工号和密码进入个人信息页面如图 3.1.6 所示：



图 3.1.6 管理员登录界面

管理员可以对个人信息进行相应的修改并保存，如图 3.1.7 所示：



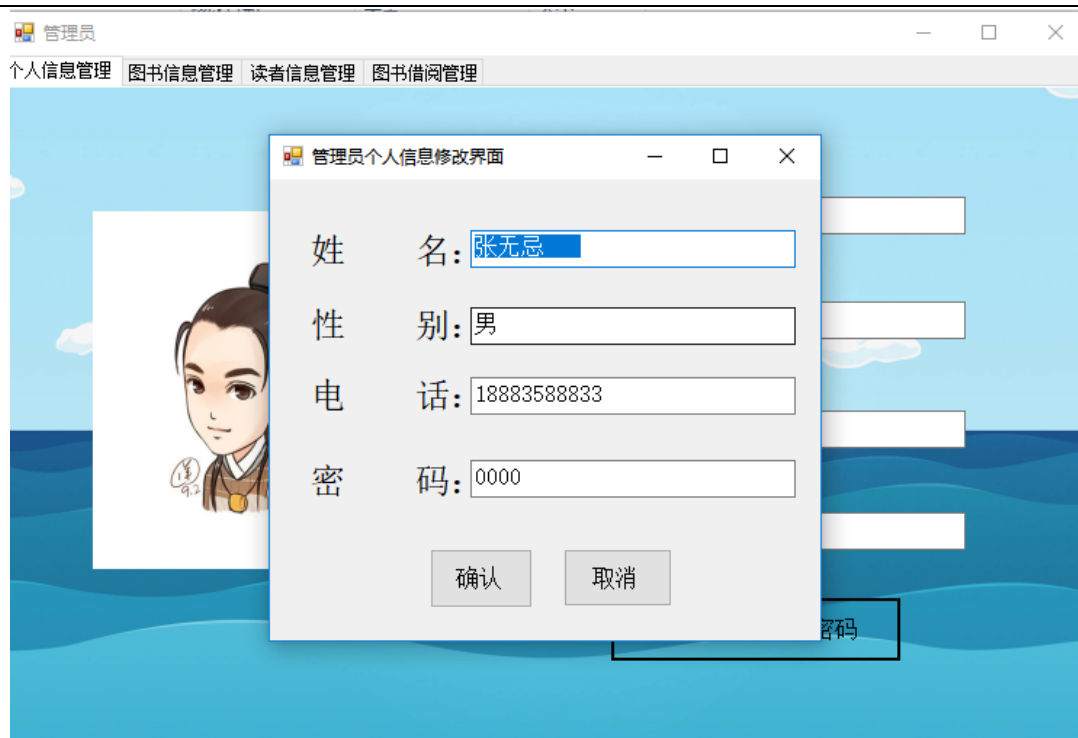


图 3.1.7 管理员个人信息修改界面

管理员能对图书信息进行维护（增、删、改、查），如图 3.1.8 所示：



图 3.1.8 管理员图书维护界面

管理员对读者信息进行维护（增、删、改、查），如图 3.1.9 所示：

图 3.1.9 读者信息管理界面

管理员具的图书借阅管理，包含读者借阅情况管理、图书的借还服务、报表的自动生成的功能，如图 3.1.10 所示：

图 3.1.10 图书借阅管理和报表功能

### 3.2 读者部分代码的讲解及运行效果

首先为了从 VB.NET 中访问 SQL 数据库，在程序的最开始要引入 SQL 访问的库，之后涉及到访问 SQL 中的数据都是类似如此进行处理的：

```
Imports System.Data.SqlClient
```

第一部分程序是读者对图书信息进行的模糊查找：

```
Dim strConnection As String = "Data Source = .;Initial Catalog = 图书管理系统;Integrated Security = True;" '定义与数据库的连接
```

```
Dim strSq2 As String = "SELECT 图书.*, 类型. 类型名 FROM 图书, 类型 WHERE 图书. 类型编号=类型. 类型编号 AND 书号 like ' " & TextBox5.Text & "%' " & "and 书名 like ' " & TextBox6.Text & "%' " & "and 作者 like ' " & TextBox7.Text & "%' " & "and 类型名 like ' " & TextBox9.Text & "%' " '定义SQL语句模糊查找，其中%表示后任意的字符串
```

```
Dim myConnection As SqlConnection = New SqlConnection(strConnection) 'SQL的连接
```

```
Dim dt As New DataSet '定义一个数据集dt, '连接和执行语句构成SQL适配器
```

```
Dim myCommand As SqlDataAdapter = New SqlDataAdapter(strSq2, myConnection)
```

```
dt.Clear() '清空数据集dt
```

```
Try
```

```
    myCommand.Fill(dt, "图书") '根据适配器连接对应数据库查询执行的结果填写
```

```
    Catch ex As Exception '数据集Dt
```

```
    MessageBox.Show(ex.Message)
```

```
End Try
```

```
dt = dt.Tables("图书") '把图书这张表赋给dt, 并且表dt和DataGridView1绑定在一起
```

```
DataGridView1.DataSource = dt
```

程序运行后获得效果如图 3.2.1 所示：

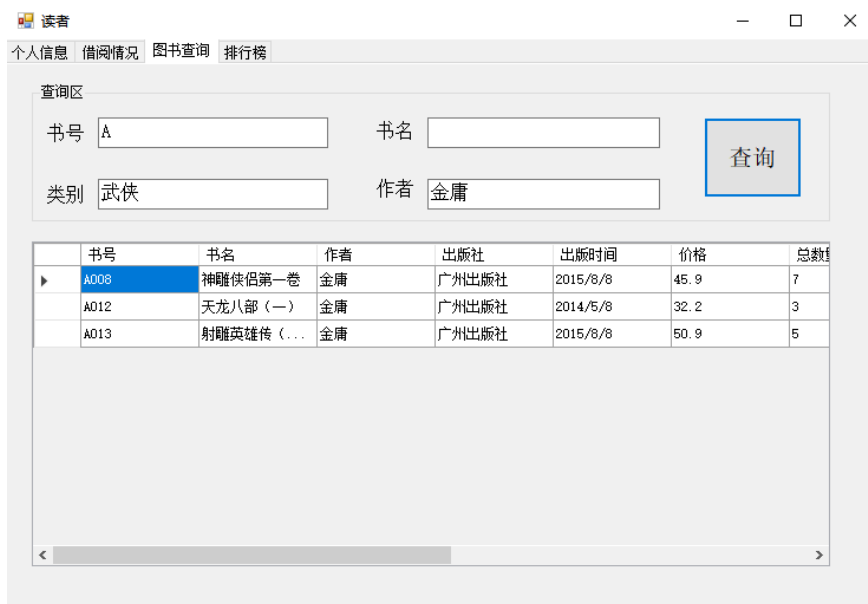


图 3.2.1 图书查询界面

### 3.3 管理员部分代码的讲解及运行效果

第三部分是管理员对图书和读者信息进行维护的相关操作(增、删、改、查)。其中关于图书信息和读者信息的查询其原理同样是使用模糊搜索,代码的主体部分与读者对图书信息的查询大同小异,不再阐述。在这里主要进行介绍的是增加、修改和删除部分的代码及运行效果图。

修改部分的代码主要是使用了 SqlCommandBuilder 的方法,通过将修改与 DataGridView 绑定在一起的表的相关数据对连接的表进行实时的更新,这种方法 的优点在于其简单高效,但弊端在于数据源应该是同一张表,不能实现多次跨表 连接。在程序的其他部分采用了其他方法。具体的代码如下所示:

```
Public dt2 As DataTable
Public sda2 As SqlDataAdapter
Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
    ' 定义一个整型intResult用于寸法判断的返回值
    Dim intResult As Integer
    ' 定义连接的语句
    Dim strConnection = New SqlConnection("Data Source = .;Initial Catalog = 图书管理系统;Integrated Security = True;")
    ' 用SqlDataAdapter定义数据适配器的对应
    sda2 = New SqlDataAdapter("SELECT * FROM 读者", strConnection)
    ' 把SqlCommandBuilder的方法建立适配器
    Dim SCB2 = New SqlCommandBuilder(sda2)
    intResult = MessageBox.Show("确认修改", "取消", MessageBoxButtons.OKCancel,
    MessageBoxIcon.Asterisk, MessageBoxDefaultButton.Button1)
    If intResult = DialogResult.OK Then
        Try
            ' 适配器把与DataGridView绑定的表更新到数据库中
            sda2.Update(dt2)
            ' 接收数据的最新变化更新到数据库中
            dt2.AcceptChanges()
        Catch ex As Exception
            MessageBox.Show(ex.Message)
            MsgBox("修改成功")
        End Try
    End If
End Sub
```

效果示意如图 3.3.1 所示:



图 3.3.1 图书信息实时更改界面示意

对于从 DataGridView 中删除对应数据的思路是,通过选中对应行,得到行的 RowIndex,就可以得到对应 Item 中书号的值,使用 VB.NET 中与数据库连接的方法执行对应的 Sql 语句,从而删除对应的某个数据。具体的代码如下所示:

```
Private Sub Button5_Click_1(sender As Object, e As EventArgs) Handles Button5.Click
```

```
    ' 定义用于存储书号的字符串
```

```
    Dim BookNumber As String
```

```
    Try
```

```
        ' 尝试获取DataGridView中的书号并且赋值给BookNumber
```

```
        BookNumber = DataGridView1.Item("书号", DataGridView1.CurrentCell.RowIndex).Value
```

```
        Dim intResult As Integer
```

```
        Dim strConnection As String = "Data Source = .;Initial Catalog = 图书管理系统;Integrated Security = True;"
```

```
        Dim mycmd As New SqlCommand
```

```
        Dim myconn As New SqlConnection(strConnection)
```

```
        ' 定义查询删除的SQL语句
```

```
        Dim strSql As String = "DELETE FROM 图书 WHERE 书号 = '" & BookNumber & "''"
```

```
        Dim myconnStr As New SqlCommand(strSql, myconn)
```

```
        myconn.Open()
```

```
        intResult = MessageBox.Show("确认删除", "取消", MessageBoxButtons.OKCancel, MessageBoxIcon.Asterisk, MessageBoxDefaultButton.Button1)
```

```
        If intResult = DialogResult.OK Then
```

```
            Try
```

```
                myconnStr.ExecuteScalar()
```

```
            Catch ex As Exception
```



```
Catch ex As Exception
```

```
MsgBox("添加失败，请确认是否输入完整准确")
```

```
End Try
```

```
strConnection.Close()
```

```
Me.Close()
```

……效果示意如图 3.3.3 所示：



图 3.3.3 图书信息添加界面

除此之外，管理员很重要的一个功能时对于读者借阅信息的查询和进行相关维护。在本图书管理系统中，我们查询的方法依然是模糊查找借书证号，但是查找表换成借阅这张表。并且在处理的过程中，有意识地分开成当前在借情况的统计和历史借阅情况统计两个部分。两个都具有修改和删除的功能，并且有相应的删除触发器。

查找部分和在前述代码的基础上进行修改，主要区别在于这里有日期时间的转换操作：

```
……DataGridView2.Columns("借出日期").DefaultCellStyle.Format = "yyyy/MM/dd HH:mm:ss"……
```

对于修改部分，由于考虑到信息的完整性，并且已经在其他页面实现管理图书读者信息的功能，这里的修改主要是针对借出日期、还书日期和应归还日期这三项。主要部分代码如下：

```

Dim BookNumber As String
Dim BorrowDate As DateTime
Dim ShouldReturnDate As DateTime
myconn.Open()
If intResult = DialogResult.OK Then
    Try
        ' 利用for循环和结合DataGridView2获得对应栏的内容对原数据库进行更新
        For i = 0 To DataGridView2.RowCount - 1
            BookNumber = DataGridView2.Rows(i).Cells(0).Value
            BorrowDate = DataGridView2.Rows(i).Cells(5).Value
            ShouldReturnDate = DataGridView2.Rows(i).Cells(6).Value
            ' SQL语句的定义
            Dim strSql As String = "UPDATE 借阅 SET 借出日期 = '" & BorrowDate & "
                ', 应归还日期 = '" & ShouldReturnDate & "' WHERE 借书证号 = '"
                & TextBox5.Text & "' AND 书号 = '" & BookNumber & "'"
            Dim myconnStr As New SqlCommand(strSql, myconn)
            myconnStr.ExecuteNonQuery()
        Next i
        MsgBox("修改成功")
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
    myconn.Close()
End If

```

效果示意如图 3.3.4 所示：

图 3.3.4 借阅信息查询效果示意图



上述主要是针对的是图书信息的维护,在管理员中对读者信息的维护只要在其基础再进行相应的更改就可以了,大同小异,这里不再重复描述。

### 3.4 借阅部分代码的讲解及运行效果

第四部分是关于借还书部分功能的实现及相关主要触发器的介绍。在我设计的这个图书管理系统中,我结合自己自身实际的日常生活把借还书的权限设定给了管理员。

实现的思路是,我通过获取输入对应的借书证号和书号,实现对借阅这张表的插入和更新。当要实现借书功能时,我会往借阅这张表中插入一条新的记录,并且获得系统的当前时间的基础上加上 30 天得到应归还日期,但是归还日期为 null,同时会调用相关的插入触发器。当要实现还书功能时,我会把当前日期更新到对应记录的归还日期上,同样会调用相关的更新触发器。

VB.NET 中借书部分具体的实现代码如下:

```
' 定义DateTime类型的借书日期和归还日期
Dim BorrowDate As New DateTime
Dim ShouldReturnDate As New DateTime
BorrowDate = Now
ShouldReturnDate = DateAdd(DateInterval.Day, 30, BorrowDate)
Dim strSql As String = "INSERT INTO 借阅(书号,借书证号,借出日期,应归还日期) VAL
UES ('" & TextBox10.Text & "', '" & TextBox11.Text & "', '" & BorrowDate & "', '" &
ShouldReturnDate & "')" ' 实现插入操作
Dim myconnStr As New SqlCommand(strSql, myconn)
myconn.Open()
intResult = MessageBox.Show("确认借书", "取消", MessageBoxButtons.OKCancel, Mes
sageBoxIcon.Asterisk, MessageBoxDefaultButton.Button1)
If intResult = DialogResult.OK Then
    Try
        myconnStr.ExecuteScalar()
        MsgBox("借书成功")
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
    myconn.Close()
End If
Catch ex As Exception
End Try
```

VB.NET 中还书部分区别的主要代码如下所示：

.....

```
Dim strSql As String = "UPDATE 借阅 SET 归还日期 = '" & ReturnDate & "' " & "WHE  
RE 书号 = '" & TextBox14.Text & "' " & "And 借书证号 = '" & TextBox12.Text & "' "
```

.....

实现效果借书效果如图3.4.1所示，还书效果如图3.4.2所示：



图 3.4.1 借书效果示意图



图 3.4.2 还书效果示意图

接下来主要介绍的部分是关于借阅时候的插入触发器、更新触发器和删除触发器。

首先是对于思路的讲解。当我们插入一条新的借阅数据时，对应的读者的借阅次数会增加 1 次，图书的被借次数会增加 1 次，图书的剩余数量减少一本。还书的时候，会对借阅表中的归还日期 null 值进行归还日期的写入，图书的剩余数量会增加 1 本。当管理员对借阅记录进行管理时，删除已有归还日期的历史借阅记录不会产生影响，删除未有归还日期的借阅记录会使图书的剩余数量增加一本。三个触发器的代码如下所示：

### 1、插入触发器：

```
CREATE TRIGGER [dbo].[BorrowBook_Handle]
ON [dbo].[借阅] FOR INSERT
AS
    DECLARE @BookNumber char(10)
    DECLARE @BorrowNumber char(10)
    DECLARE @BookRest int

    SELECT @BookNumber = i.书号 FROM inserted i
    SELECT @BorrowNumber = i.借书证号 FROM inserted i
    UPDATE 图书 SET 剩余数量 = 剩余数量-1, 被借次数 = 被借次数+1 WHERE 书号 = @BookNumber

    UPDATE 读者 SET 借阅次数 = 借阅次数+1 WHERE 借书证号 = @BorrowNumber
    SELECT @BookRest = 剩余数量 FROM 图书 WHERE 书号 = @BookNumber
    if (@BookRest=-1) //这里是对图书的剩余数量所做的判断，有库存才能借，没有回滚
    BEGIN
        PRINT '图书已被借完'
        ROLLBACK TRAN
    END
    Else
        print '插入成功'
```

### 2、更新触发器：

```
CREATE TRIGGER [dbo].[ReturnHandle]
ON [dbo].[借阅] FOR UPDATE
AS
    DECLARE @BookNumber char(10)
    SELECT @BookNumber = i.书号 FROM inserted i

    UPDATE 图书 SET 剩余数量 = 剩余数量+1 WHERE 书号 = @BookNumber
```

### 3、删除触发器：

```

CREATE TRIGGER [dbo].[DeleteBook]
on [dbo].[借阅] FOR DELETE
AS
    DECLARE @BookNumber char(10)
    DECLARE @ReturnDay datetime

    SELECT @BookNumber = d.书号 FROM deleted d
    SELECT @ReturnDay = d.归还日期 FROM deleted d

    UPDATE 图书 SET 剩余数量 = 剩余数量+1 WHERE 书号 = @BookNumber AND @ReturnDay is null

```

### 3.5 排行榜部分代码的讲解及运行效果

第五部分是关于排行榜查询的图表显示内容，在本图书管理数据库中我在读者客户端设置了排行榜的功能。除了包括图书排行榜和读者排行榜的表格显示外，还设置更为直观的柱形图显示，丰富了功能。重要的代码如下所示：

```

.....Dim strSq2 As String = "SELECT 图书.*,类型.类型名 FROM 图书,类型 WHERE 图书.类型编号=类型.类型编号 ORDER BY 被借次数 DESC" '按照被借次数降序排列显示
Dim myConnection As SqlConnection = New SqlConnection(strConnection)
.....

Try
    myCommand.Fill(dt, "图书")
Catch ex As Exception
    MessageBox.Show(ex.Message)
End Try

'设置DataGrid1的数据源
bookrank = dt.Tables("图书")
DataGridView3.DataSource = bookrank

bookrank = dt.Tables("图书") '设置DataGrid3的数据源来自的表
'把设置的表书本排行赋给了DataGridView3

DataGridView3.DataSource = bookrank

.....Dim strSq3 As String = "SELECT 读者.姓名,读者.性别,读者.工作单位,读者.借阅次数 FROM 读者 ORDER BY 借阅次数 DESC".....

peoplerank = dt3.Tables("读者")
'把设置的读者排行赋给了DataGridView3

DataGridView4.DataSource = peoplerank

```

实现的效果如图 3.5.1 所示：



图 3.5.1 排行榜表格显示

图表部分的代码使用了 VB.NET 中的 CHART 控件，关键部分的代码如下：

```

.....Dim strSq2 As String = "SELECT 图书. 书名, 图书. 被借次数 FROM 图书, 类型 WHERE
图书. 类型编号=类型. 类型编号 ORDER BY 被借次数 DESC"
.....Dim strSq3 As String = "SELECT 读者. 姓名, 读者. 借阅次数 FROM 读者 ORDER BY
借阅次数 DESC" Dim dt As New DataSet.....
Dim myCommand As SqlDataAdapter = New SqlDataAdapter(strSq2, myConnection)
rankbook = dt.Tables("图书")
Chart1.DataSource = rankbook
Chart1.DataBind()
'对chart1的横纵坐标进行设置，下面的chart2是一样的
Chart1.Series(0).XValueMember = "书名"
Chart1.Series(0).YValueMembers = "被借次数"

rankpeople = dt3.Tables("读者")
Chart2.DataSource = rankpeople
Chart2.DataBind()
Chart2.Series(0).XValueMember = "姓名"
Chart2.Series(0).YValueMembers = "借阅次数"

```

效果如图 3.5.2 所示：

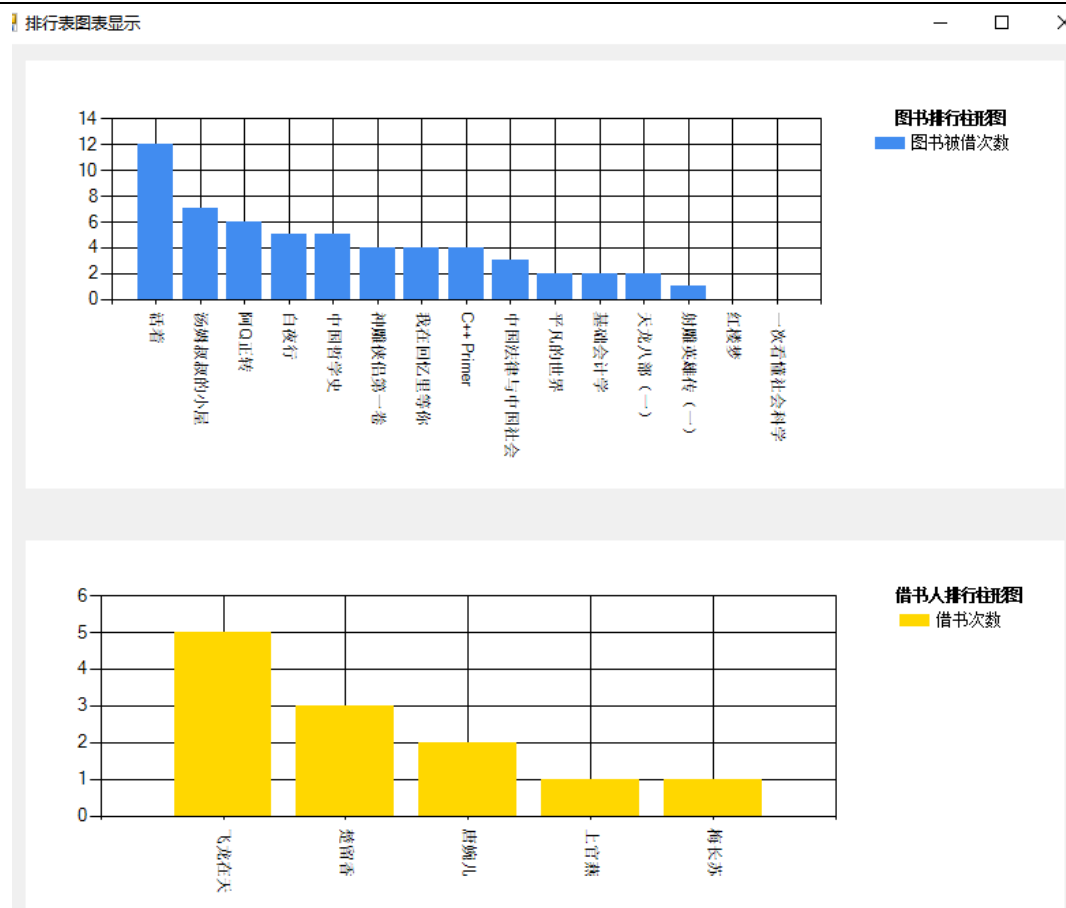


图 3.5.2 排行榜柱形图显示

### 3.6 报表部分实现的讲解及运行效果

报表可以用来显示动态的数据，在数据库开发中具有很重要的意义。在本课设的图书管理系统中，我在管理员客户端设置了报表的功能，主要使用的 RDLC 报表文件、ReportViewer 控件来实现数据报表。一般设计步骤如下。

①定义视图。在 SQL Server 端，根据前台要显示的报表信息定义相应的视图，为报表文件准备数据源。

②定义 RDLC 报表文件。报表的输出数据格式及显示的数据项都是在 RDLC 报表文件中设定的。

③拖放 ReportViewer 控件并设置其数据源。在工具箱中将控件 ReportViewer 拖放到窗体上，指定其加载的报表文件及数据源即可完成报表设计。

定义的三张视图的代码如下所示：

1、图书视图：

```
CREATE VIEW [dbo].[BookReports]
```

```
AS
```

```
SELECT 图书.书号, 图书.书名, 图书.作者, 图书.出版社, 图书.出版时间, 类型.类型名,
图书.总数量, 图书.价格
```

```
FROM 图书, 类型 WHERE 图书.类型编号 = 类型.类型编号
```

## 2、读者视图:

```
CREATE VIEW [dbo].[ReaderReports]
```

```
AS
```

```
SELECT 读者.借书证号, 读者.姓名, 读者.性别, 读者.工作单位, 读者.借阅次数
```

```
FROM 读者
```

## 3、借阅视图:

```
CREATE VIEW [dbo].[BorrowReports]
```

```
AS
```

```
SELECT 借阅.借书证号, 读者.姓名, 图书.书名, 图书.作者, 图书.出版社, 图书.出版时间,
借阅.借出日期, 借阅.应归还日期, 借阅.归还日期
```

```
FROM 借阅, 读者, 图书 WHERE 借阅.借书证号 = 读者.借书证号 AND 借阅.书号 = 图
书.书号
```

关于报表的设计还有一些执行时间、报表名称和统计的聚合函数,限于篇幅,这里不再赘述。在报表设计的过程中,通过相应的视图进行对应连接显示。具体的效果如图 3.6.1、图 3.6.2 和图 3.6.3 所示。



书号	书名	作者	出版社	出版时间	类型名	总数量	价格
A001	平凡的世界	路遥	长江出版社	1998/1/2 0:00:00	经典著作	10	40
A002	基础会计学	彭恒	华南理工大学出版社	2001/1/1 0:00:00	财经管理	5	32.4
A003	阿Q正传	鲁迅	花城出版社	2003/2/3 0:00:00	中国文学	5	25
A004	活着	余华	上海文艺出版社	2010/9/18 0:00:00	中国文学	5	32
A005	我在回忆里等你	辛夷坞	江苏文艺出版社	2012/12/1 0:00:00	言情小说	5	35.9
A006	C++ Primer	李普曼	电子工业出版社	2014/12/12 0:00:00	工具书	5	78.2
A007	中国哲学史	冯友兰	商务出版社	2017/7/3 0:00:00	哲学类	6	50.5
A008	神雕侠侣第一卷	金庸	广州出版社	2015/8/8 0:00:00	武侠小说	7	45.9
A009	汤姆叔叔的小屋	斯陀夫人	人民文学出版社	2018/9/9 0:00:00	外国文学	2	23.3
A010	中国法律与中国社会	瞿同祖	中华书局	2019/9/1 0:00:00	政治法律	4	15.5
A011	一次看懂社会科学	刘炯朗	山西人民出版社	2012/12/2 0:00:00	社会科学	5	28

图 3.6.1 图书信息汇总报表

读者信息报表

2019/11/12 17:36:58

借书证号	姓名	性别	工作单位	借阅次数
1615311032	上官燕	女	腾讯科技公司	1
1615322013	飞龙在天	男	华侨大学	5
1615322020	梅长苏	男	百度技术	1
1615322030	唐婉儿	女	大疆科技	2
1615322049	楚留香	男	中国农业银行	3

读者人数统计: 5

图 3.6.2 读者信息汇总表

借阅信息报表

2019/11/12 17:38:30

借书证号	姓名	书名	作者	出版社	出版时间	借出日期
1615311032	上官燕	平凡的世界	路遥	长江出版社	1998/1/2 0:00:00	2019/11/10 9:00:00
1615322013	飞龙在天	平凡的世界	路遥	长江出版社	1998/1/2 0:00:00	2019/11/9 9:20:00
1615322013	飞龙在天	平凡的世界	路遥	长江出版社	1998/1/2 0:00:00	2019/11/9 21:20:00
1615322013	飞龙在天	基础会计学	彭恒	华南理工大学出版社	2001/1/1 0:00:00	2019/11/9 17:00:00
1615322013	飞龙在天	阿Q正传	鲁迅	花城出版社	2003/2/3 0:00:00	2019/11/11 18:00:00
1615322013	飞龙在天	阿Q正传	鲁迅	花城出版社	2003/2/3 0:00:00	2019/11/11 18:00:00
1615322013	飞龙在天	活着	余华	上海文艺出版社	2010/9/18 0:00:00	2019/11/9 17:00:00
1615322013	飞龙在天	我在回忆里等你	辛夷坞	江苏文艺出版社	2012/12/1 0:00:00	2019/11/12 10:00:00
1615322013	飞龙在天	神雕侠侣第一卷	金庸	广州出版社	2015/8/8 0:00:00	2019/1/8 0:00:00

图 3.6.3 借阅信息汇总表



## 总结

通过本次课程设计,我对关系型数据库系统的基础及应用有了比较全面的了解。在实际数据库的设计过程中应严格按照需求分析、概念模式设计、逻辑模式设计、数据库实施阶段和数据库运行和维护阶段五个方面来执行。特别是关于前面三个阶段的设计。需求分析是基础,是对整个题目要求的分析。只有把需求分析做好,接下来的概念模式设计和逻辑模式设计就水到渠成了。

在本次课程设计的过程中,从最刚开始的采用绑定方式连接数据库再到后面的利用适配器和界面的连接方式,充分使用了各种连接的方法以获得各自最好的功能。其中,在概念模式阶段对于管理员与图书和读者之间的关系,最开始设计时比价混乱。之后,自己通过查询有关资料并且结合自己的想法合理地处理好这部分的问题。除此之外,在实际的开发过程中,经常遇到多次执行失败的各种错误。自己通过不断的调试程序和通过网上查阅相关资料,自己把问题一个一个地解决掉。

在本次关系型数据库的实践过程中,自己收获颇丰。掌握了基本软件的使用和代码的编写,学习了关系型数据库的基本思想和基本开发流程,学会如何高效地查找资料和独立解决过程中出现的问题。感谢老师的指导和同学们的交流,让整个过程充满趣味。

## 参考文献

- [1] 袁科萍, 杨志强, 龚沛增. 数据库技术及应用[M]. 北京: 高等教育出版社, 2015.

## 附录

调试图书管理系统时的图片如下所示：

