


项目结题验收单

1 专家验收表（主持人所在单位组织 3-5 名专家对项目进行验收、自评。）

项目名称	VBA 技术在图书馆数字化管理系统中的应用实践			
主持人	艾里亚尔·阿不都克里木	职务/职称	馆员	
所在单位	新疆农业大学图书馆（加盖公章）			
专 家 意 见	<div style="text-align: center;">  </div> <p>该研究深入探讨了 VBA 技术在图书馆数字化管理系统中的应用，取得了显著成果。特别是在自动化数据处理、查询统计、报表生成和信息互通等方面的创新应用，对提高图书馆管理效率和服务质量具有重要意义。其重要意义在于为图书馆管理提供了先进技术支持，有助于提升其管理水平和服务水平。建议在实施过程中，充分考虑图书馆的实际需求，定制化系统设计，以确保其有效支持和优化。</p> <p style="text-align: center;">研究结果已达到课题目标，可顺利结题。</p> <p style="text-align: right;">（如需要可增加页数）</p>			
专家签字	陈英杰	冯军	白克力	
职务/职称	副教授	副研究员/副馆员	副研究员	



项目编号:

CALIS 全国农学文献信息中心研究项目 结题报告

项目名称: VBA 技术在图书馆数字化管理系统中的应用实践

项目关键词: VBA, 图书馆, 数字化管理

项目单位: 新疆农业大学图书馆

通信地址: 新疆乌鲁木齐市沙依巴克区农大东路 311 号

项目主持人: 艾里亚尔·阿不都克里木

联系电话: 18999859876

电子邮件: xndtsg@126.com

提交日期: 2024 年 5 月 18 日

目 录

1 研究背景、目的及意义	1
1.1 研究背景	1
1.2 研究目的	1
1.3 研究意义	1
2 研究内容、思路及方法	1
2.1 关键技术	1
2.2 系统设计与实现	2
3 结论与建议	14
3.1 结论	18
3.2 建议	18
4 项目成果	19
5 参考文献	20

VBA 技术在图书馆数字化管理系统中的应用实践

关键词：VBA，图书馆，数字化管理

1 研究背景、目的及意义

1.1 研究背景

随着数字化时代的到来，图书馆管理方式也在不断演进。传统的手工管理已经不能满足现代图书馆的需求，数字化管理系统成为了图书馆管理的必备工具。在这样的背景下，VBA 技术作为一种高效、灵活的编程语言，被广泛应用于 Excel 和 Access 等软件中，为图书馆数字化管理系统的开发提供了新的思路和可能性。

1.2 研究目的

本研究旨在探讨 VBA 技术在图书馆数字化管理系统中的应用实践，分析其在提高图书馆管理效率、优化服务质量等方面的优势和局限性，为图书馆数字化管理系统的开发和优化提供参考。

1.3 研究意义

通过对 VBA 技术在图书馆数字化管理系统中的应用进行深入研究和分析，可以：

- 提高图书馆管理效率和服务质量，简化管理流程，提高工作效率；
- 开发出定制化的图书馆数字化管理系统，更好地满足图书馆管理需求；
- 减少图书馆管理人员的工作量，更好地利用人力资源；
- 提高用户的使用体验，为读者提供更多多样化的服务；
- 促进 VBA 技术的研究和应用，提高图书馆数字化管理系统的开发水平。

2 研究内容与方法

2.1 关键技术

2.1.1 VBA 技术

VBA 是一种强大的编程语言，能够与 Microsoft Office 套件（如 Excel、Word 和 Access）实现无缝集成，为用户提供了高度定制化和自动化处理的能力。在该项目中，

充分利用了 VBA 技术，开发了一系列功能，包括自动化数据处理、统计查询、报表生成、提醒预警以及信息互通等方面的功能。

2.1.2 Microsoft Office 应用和对象模型

Microsoft Office 套件涵盖了诸如 Excel、Access 等应用程序，提供了丰富的功能和对象模型，可以通过 VBA 编程实现个性化操作和处理。在项目中，利用了 Microsoft Office 的对象模型来操作和控制 Office 应用程序中的各种元素，例如工作表、单元格、文档和段落等。通过 VBA 编写的代码，可以直接操纵这些对象，实现各种定制功能和自动化处理任务。

2.1.3 ORACLE 数据库技术

Oracle 数据库是一款业界领先的关系型数据库管理系统，以其卓越的性能、高度可靠性和丰富的功能特性而著称。其强大的数据管理和查询功能使其成为许多企业和组织首选的数据库解决方案之一。在图书馆集成管理系统中，Oracle 数据库扮演着至关重要的角色。通过 Oracle 数据库技术，图书馆可以轻松地存储、管理和查询大量的图书信息、借阅记录和用户数据，确保图书馆管理工作的高效运作。

本项目选择采用 Oracle 数据库技术，是基于其在业界的广泛应用和可靠性。Oracle 数据库不仅能够处理大规模的数据存储需求，还能够支持复杂的数据查询和分析操作。在图书馆集成管理系统中，Oracle 数据库不仅可以存储各类图书馆资源的信息，还可以记录用户的借阅情况、借阅历史等重要数据，为图书馆管理者提供全面的数据支持和管理工具。

2.2 系统设计与实现

2.2.1 系统架构

系统的基本结构包括前端界面层、业务逻辑层、数据访问层和 Oracle 数据库层四个主要组成部分。前端界面层采用 Excel 和 VBA，为用户提供了与系统交互的界面和功能。业务逻辑层负责处理用户请求并执行相应的业务逻辑，其中包括与 Oracle 数据库的交互，确保系统功能的正常运行。数据访问层通过 Oracle Instant Client 与 Oracle 数据库进行交互，执行 SQL 命令以满足业务逻辑层的需求。而 Oracle 数据库层则负责存储系统的数据，并通过 Oracle 数据库管理系统进行管理和维护，确保数据的安全性和完整性。这四个层级的协同工作，构成了系统的基本框架，为图书馆集成管理系统提供了稳定、高效的运行环境。图 1 展示了系统架构图。

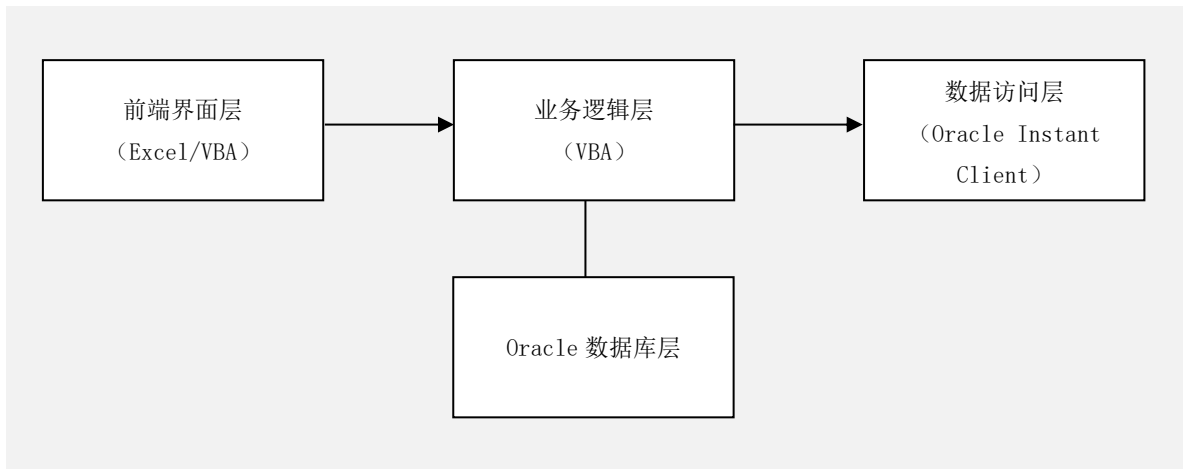


图1 系统架构图

2.2.2 功能模块

系统通过自动化处理模块简化了借书、还书、续借等操作，利用 VBA 编写的程序与管理系统交互，提高了效率。数据处理与分析模块能清洗、提取和分析借阅记录等数据，为管理者提供支持。定制功能模块满足了特定需求，提升了效率。系统维护模块提高了稳定性，确保系统长期稳定运行。

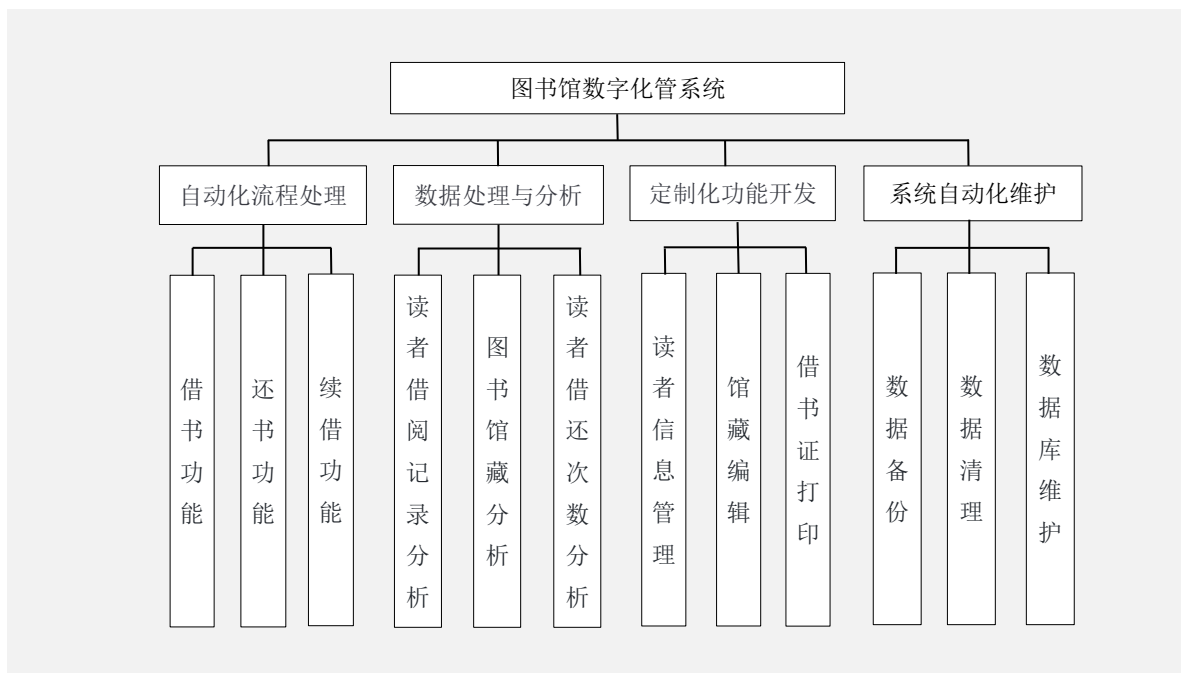


图2 系统功能结构图

如图 2 所示，该系统应该具有以下功能模块：

(1) 自动化流程处理

实现自动化借书、还书、续借等功能，通过 VBA 编写宏代码与图书馆管理系统进行交互，简化操作流程。

(2) 数据处理与分析

利用 VBA 与 Excel 等软件配合，进行读者借阅记录分析、图书馆藏分析、读者借还次数分析等，提供数据清洗、提取、分析功能。

(3) 定制化功能开发

根据图书馆需求，利用 VBA 编写程序实现定制化功能的开发，如读者信息管理、馆藏编辑、借书证打印等，满足特定管理需求。

(4) 系统自动化维护

利用 VBA 编写程序实现系统自动化维护和排错功能，包括数据备份、清理、数据库维护等，确保系统运行的稳定性和可靠性。

2.2.3 系统开发环境

在本项目中，为了实现与 Oracle 数据库的连接和操作，采用了 Oracle 官方提供的轻量级客户端软件 Oracle Instant Client。该软件允许在本地部署环境中轻松连接和操作 Oracle 数据库，而无需安装完整的 Oracle 客户端。这一选择既减少了系统资源的占用，又简化了部署过程，提高了系统的灵活性和可维护性。

开发工具方面，本项目使用了 Excel 和 VBA 作为主要开发工具，结合 Oracle Instant Client，为系统开发和数据库交互提供了便利。通过 Excel 和 VBA，开发人员可以方便地进行数据处理、报表生成等操作，并利用 Oracle Instant Client 实现与 Oracle 数据库的数据交换和管理。这一组合使得开发过程更加高效，并为图书馆集成管理系统提供了强大的数据处理和管理功能。

此外，在本项目还采用了 DBeaver 数据库查询软件作为辅助工具。DBeaver 是一款功能强大的跨平台数据库工具，提供直观友好的用户界面，能够帮助开发人员进行数据库查询、管理和监控。DBeaver 的主要用途包括数据查询、检查和更新，以及数据库状态监控。通过结合 Oracle Instant Client 和 DBeaver，能够更加高效地进行数据库操作和管理，提高了系统开发和维护的效率，同时也为系统提供了强大的数据处理和管理功能。

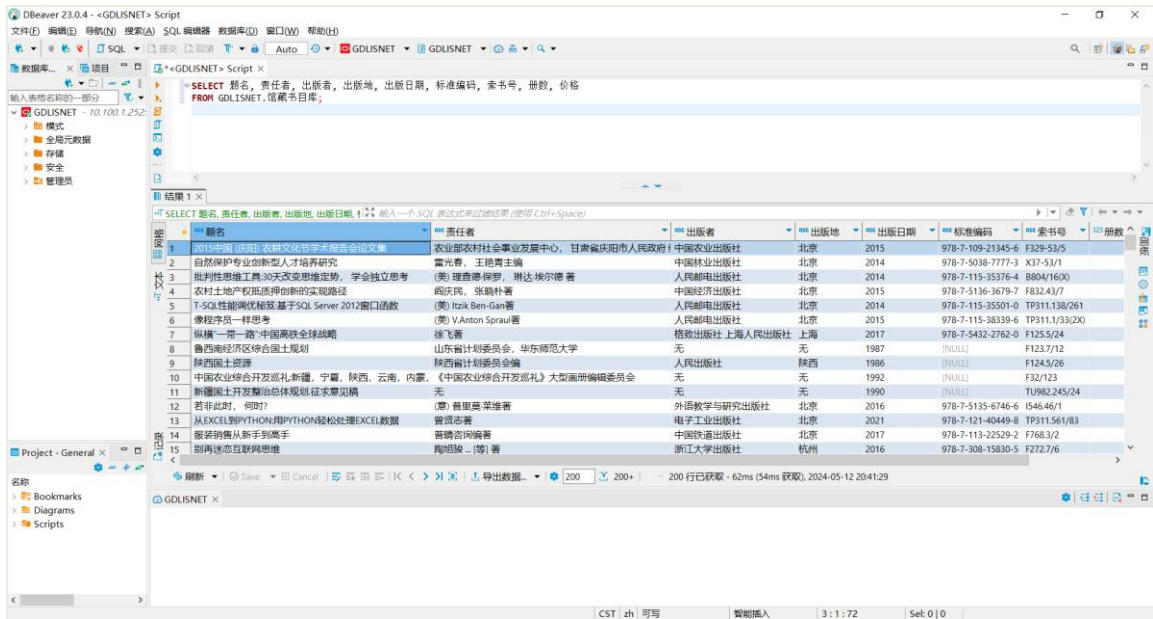


图3 DBeaver 数据库远程管理软件

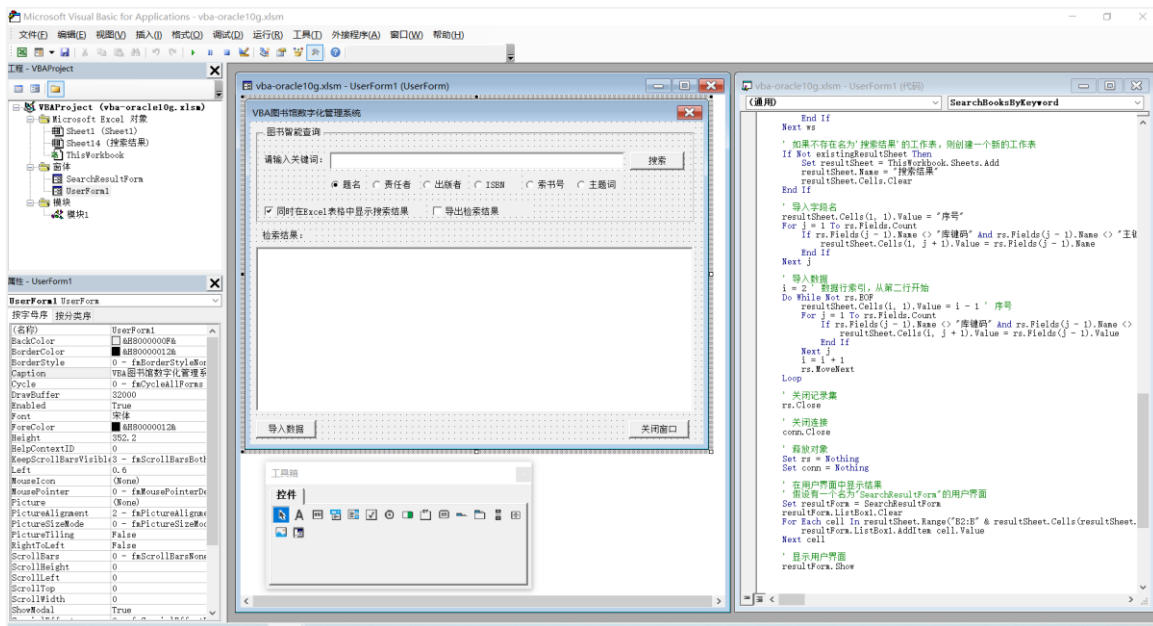


图4 Microsoft Visual Basic for Applications 开发环境

2.2.4 系统实现

首先, 针对开发环境的各项参数需进行适当设置。在确保与数据库系统有效对接的前提下, 需要在开发环境中安装 Oracle Instant Client 软件, 并在 ODBC 数据源中配置相应的驱动程序和用户 DSN。这样做可以保证在 VBA 环境中能够引用到 Microsoft ActiveX Data Objects 和 Microsoft ActiveX Data Objects Recordset 等两个关键库。

接下来，在编程过程中，需要通过创建 ADOBE.Connection 对象来建立与数据库的连接。通过使用之前配置的 ODBC 用户 DSN、数据库用户和密码等信息，可以实现连接数据库的操作。以下是具体的示例代码：

```

Dim conn As Object
Dim rs As Object

' 创建连接对象
Set conn = CreateObject("ADODB.Connection")
' 设置连接字符串，使用之前创建的ODBC 用户 DSN
conn.ConnectionString = "DSN=YourDSNName;UID=YourUsername;PWD=YourPassword;"
' 打开连接
conn.Open
Set rs = CreateObject("ADODB.Recordset")

```

这样设置后，系统便能够在开发环境中顺利与数据库进行连接，为后续功能模块的实现奠定了必要的基础。

(1) 程序交互界面设计

在程序交互界面的设计方面，使用了 VBA 提供的窗体 (Form)、框架 (Frame)、标签 (Label)、列表框 (ListBox) 和按钮 (CommandButton) 等常用控件，并设置它们的属性和功能。通过窗体组件的 Initialize 事件和按钮组件的 Click 事件中调用窗体 (Form) 的 Show 和 Hide 等方法，实现不同窗体之间的切换。



图 5 VBA 程序主界面



图 6 图书检索模块

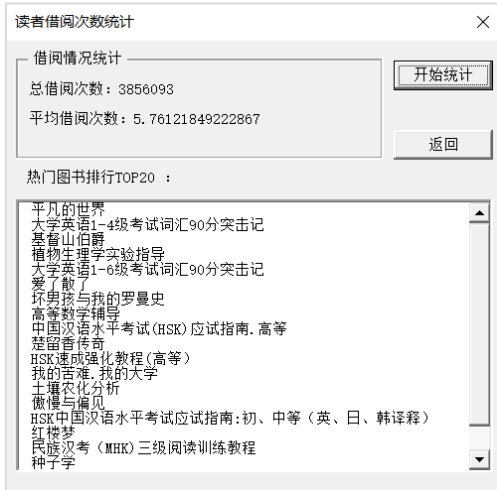


图 7 借阅情况分析模块

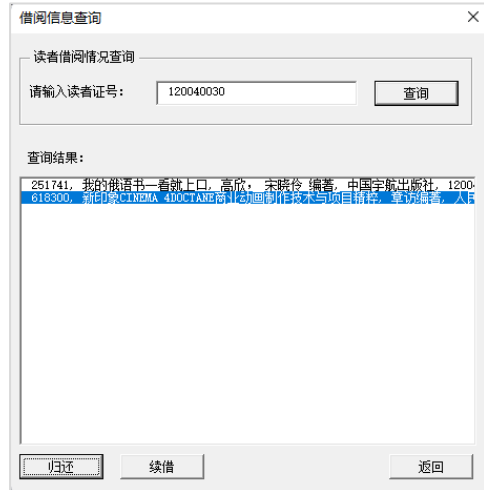


图 8 借阅信息查询模块



图 9 用户信息管理模块



图 10 图书馆藏分析模块

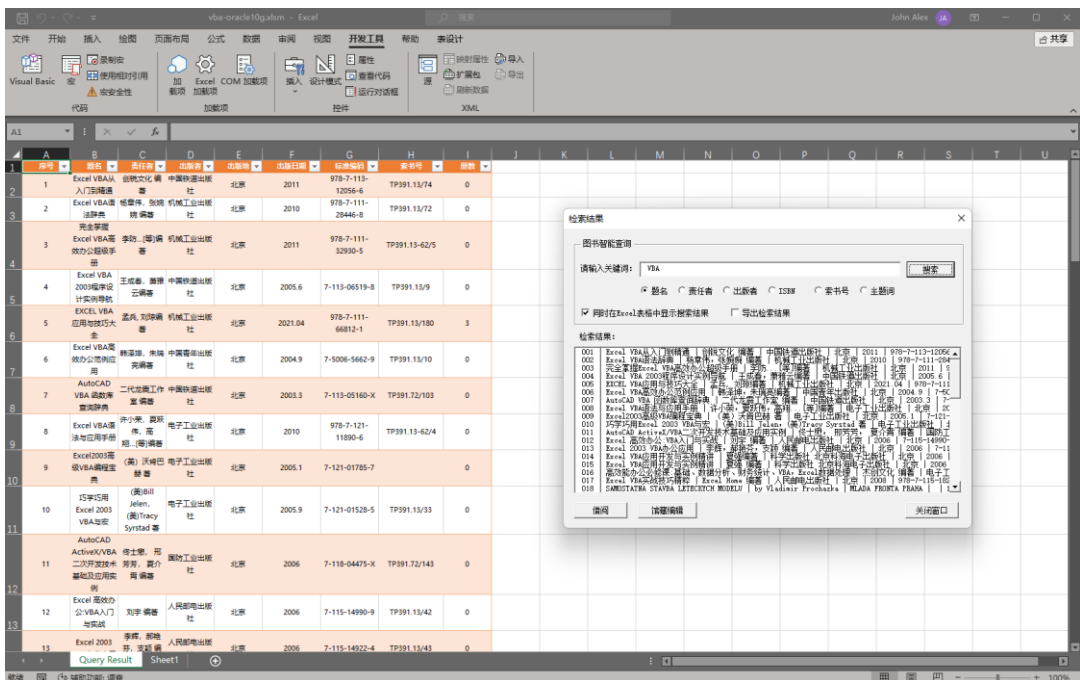


图 11 自动报表生成模块

(2) 系统功能模块的实现

通过 VBA 编写宏代码与图书馆管理系统进行交互,实现了一个简易的图书管理系统。系统功能包括搜索图书、馆藏编辑、图书的借还、馆藏书目数据的分析、读者借阅数据分析以及展示搜索结果等,旨在简化用户操作流程。

在程序设计方面,采用了模块化设计。例如,数据库查询、数据展示等功能通过独立的函数来完成具体的任务。以下结合部分独立函数和子过程,实现了数据库的查询和在 Excel 中展示结果。同时还提供了一些定制化的功能,如创建新工作表、设置表格样式和列宽度等。这些功能可以帮助用户更有效地处理数据库查询结果并展示数据。具体的代码段请参见附件中的 VBA 文件。

以下为部分核心函数,是指在图书管理系统中实现各项功能的关键代码段,包括数据库查询、数据展示以及表格样式设置等。通过调用和组合这些函数,用户可以完成系统中的各项操作。核心函数的设计和编写是系统的基础,负责实现主要功能并提供友好的操作界面和结果展示。优化和完善核心函数将直接影响系统的性能和用户体验,因此在开发过程中需要特别重视核心函数的设计和实现。

● QueryDatabase 函数:

```
Function QueryDatabase(tableName As String, keywordField As String, keyword As String,
returnFields As String, newSheetName As String) As Boolean
    On Error GoTo ErrorHandler
    Dim conn As Object, rs As Object
    Dim strSQL As String
    Dim resultSheet As Worksheet
    ' 创建数据库连接对象
    Set conn = CreateObject("ADODB.Connection")
    ' 设置连接字符串,使用之前创建的ODBC数据源名称
    conn.ConnectionString = "DSN=YourDSN;UID=YourDatabase;PWD=YourPassword;"
    ' 打开连接
    conn.Open
    ' 构建查询语句
    strSQL = "SELECT " & returnFields & " FROM " & tableName & " WHERE " &
keywordField & " LIKE '%" & keyword & "%'"
    ' 创建记录集对象
    Set rs = CreateObject("ADODB.Recordset")
    ' 执行查询
    rs.Open strSQL, conn
```

```

' 创建新工作表
Set resultSheet = createNewSheet(newSheetName)
' 导入字段名
resultSheet.Cells(1, 1).Value = "序号"
For j = 0 To rs.Fields.Count - 1
    resultSheet.Cells(1, j + 2).Value = rs.Fields(j).Name
Next j
' 导入数据
Dim i As Long
i = 2 ' 数据行索引, 从第二行开始
Do While Not rs.EOF
    resultSheet.Cells(i, 1).Value = i - 1 ' 序号
    For j = 0 To rs.Fields.Count - 1
        resultSheet.Cells(i, j + 2).Value = rs.Fields(j).Value
    Next j
    i = i + 1
    rs.MoveNext
Loop
' 关闭连接
conn.Close
' 设置函数返回值为 True, 表示查询成功
QueryDatabase = True
Exit Function
ErrorHandler:
' 错误处理
MsgBox "查询数据库时发生错误: " & Err.Description, vbExclamation
If Not rs Is Nothing Then rs.Close
If conn.State = 1 Then conn.Close
QueryDatabase = False
End Function

```

该函数用于执行数据库查询并将结果显示在新的 Excel 工作表中。参数包括表名 (tableName)、关键字段(keywordField)、关键词(keyword)、返回字段(returnFields) 和新工作表名称 (newSheetName)。首先创建数据库连接对象, 然后构建 SQL 查询语句, 执行查询并将结果导入新的工作表中。如果查询成功, 函数返回 True; 如果发生错误, 将显示错误消息并返回 False。

- QueryDatabaseToUserForm 函数:

```

Function QueryDatabaseToUserForm(tableName As String, keywordField As String,
keyword As String, returnFields As String, BooksInfo As String) As Boolean
    On Error GoTo ErrorHandler

```

```

Dim conn As Object, rs As Object
Dim strSQL As String
Dim UserForm As Object
Dim listBox As Object
' 创建数据库连接对象
Set conn = CreateObject("ADODB.Connection")
' 设置连接字符串, 使用之前创建的ODBC 数据源名称
conn.ConnectionString = "DSN=YourDSN;UID=YourDatabase;PWD=YourPassword;"
' 打开连接
conn.Open
' 构建查询语句
strSQL = "SELECT " & returnFields & " FROM " & tableName & " WHERE " &
keywordField & " LIKE '%" & keyword & "%'"
' 创建记录集对象
Set rs = CreateObject("ADODB.Recordset")
' 执行查询
rs.Open strSQL, conn
' 清空ListBox 内容
SearchResultForm.DuplicateBooks.Clear
' 添加数据
Do While Not rs.EOF
Dim rowValues As String
rowValues = rs.Fields("条形码").Value & "," & BooksInfo & "," &
rs.Fields("索书号").Value & ","
' 判断外借状态字段的值
If rs.Fields("外借状态").Value = 0 Then
rowValues = rowValues & "在库"
ElseIf rs.Fields("外借状态").Value = 1 Then
rowValues = rowValues & "已借出"
End If
' 添加到ListBox
SearchResultForm.DuplicateBooks.AddItem rowValues
rs.MoveNext
Loop
' 关闭连接
conn.Close
' 设置函数返回值为True, 表示查询成功
QueryDatabaseToUserForm = True
Exit Function
ErrorHandler:
' 错误处理
MsgBox "查询数据库时发生错误: " & Err.Description, vbExclamation
If Not rs Is Nothing Then rs.Close
If conn.State = 1 Then conn.Close

```

```
QueryDatabaseToUserForm = False
End Function
```

该函数用于查询图书单册信息以及单册图书的借阅情况。参数包括表名 (tableName)、关键字段(keywordField)、关键词(keyword)、返回字段(returnFields) 和图书信息 (BooksInfo)。该函数与 QueryDatabase 函数配合使用, 根据图书主键码信息构建 SQL 查询语句执行查询, 并将结果以规定的格式显示在指定的列表框 (ListBox) 中。如果查询成功, 函数返回 True; 如果发生错误, 将显示错误消息并返回 False。

- createNewSheet 函数:

```
Function createNewSheet(resultSheetName As String) As Worksheet
    Dim resultSheet As Worksheet
    Dim existingResultSheet As Boolean
    Dim userResponse As VbMsgBoxResult
    Dim ws As Worksheet ' 声明工作表变量
    ' 检查是否存在指定名称的工作表
    existingResultSheet = False
    For Each ws In ThisWorkbook.Sheets
        If ws.Name = resultSheetName Then
            existingResultSheet = True
            Set resultSheet = ws
            Exit For
        End If
    Next ws
    ' 如果不存在指定名称的工作表, 则创建一个新的工作表
    If Not existingResultSheet Then
        userResponse = MsgBox("工作表 '" & resultSheetName & "' 不存在, 是否创建?", vbYesNo)
        If userResponse = vbYes Then
            Set resultSheet = ThisWorkbook.Sheets.Add
            resultSheet.Name = resultSheetName
        Else
            MsgBox "操作已取消。"
            Exit Function
        End If
    Else
        ' 如果工作表已存在, 提示用户是否替换内容
        userResponse = MsgBox("工作表 '" & resultSheetName & "' 已存在, 是否替换?", vbYesNo)
        If userResponse = vbYes Then
            resultSheet.Cells.Clear
        End If
    End If
End Function
```

```

Else
    MsgBox "操作已取消。"
    Exit Function
End If
End If
Set createNewSheet = resultSheet
End Function

```

该函数用于创建一个新的 Excel 工作表，或者清空已存在的工作表。参数是要创建或操作的工作表名称（resultSheetName）。函数会检查工作簿中是否存在指定名称的工作表，如果不存在则询问用户是否创建，存在则询问是否替换内容。

- DisplayDataWithListObject 函数：

```

Sub DisplayDataWithListObject(tableName As String)
    Dim ws As Worksheet
    Dim lo As ListObject
    Dim lastRow As Long, lastCol As Long
    Dim rng As Range
    Dim columnWidths As Variant
    ' 获取指定名称的工作表
    On Error Resume Next
    Set ws = ThisWorkbook.Sheets(tableName)
    On Error GoTo 0
    If ws Is Nothing Then
        MsgBox "工作表 " & tableName & " 不存在。", vbExclamation, "提示"
    Else
        ' 删除之前的 ListObject (如果有)
        For Each lo In ws.ListObjects
            lo.Delete
        Next lo
        ' 获取数据范围
        lastRow = ws.Cells(ws.Rows.Count, 1).End(xlUp).row
        lastCol = ws.Cells(1, ws.Columns.Count).End(xlToLeft).Column
        Set rng = ws.Range("A1:" & ws.Cells(lastRow, lastCol).Address)
        ' 将数据范围转换为 ListObject
        Set lo = ws.ListObjects.Add(xlSrcRange, rng, , xlYes)
        ' 设置表格样式
        lo.TableStyle = "TableStyleMedium3"
        ' 字体设置
        With lo.HeaderRowRange.Font
            .Name = "微软雅黑"

```

```

        .Size = 8
    End With
    With lo.DataBodyRange.Font
        .Name = "微软雅黑"
        .Size = 8
    End With
    ' 设置表格对齐方式为居中对齐
    lo.Range.HorizontalAlignment = xlCenter
    ' 设置自动换行
    lo.DataBodyRange.WrapText = True
    ' 初始化列宽度数组
    columnWidths = Array(5, 10, 10, 10, 8, 8, 8, 8, 5)
    ' 设置列宽度
    SetColumnWidths lo, columnWidths
End If
End Sub

```

该函数用于在指定的工作表中显示数据，并将数据以 ListObject（Excel 表格）的形式呈现。首先获取指定名称的工作表，然后删除之前的 ListObject（如果有），获取数据范围，将数据范围转换为 ListObject。设置表格样式、字体、对齐方式，并可以设置列宽度和自动换行。

- SetColumnWidths 子过程:

```

Sub SetColumnWidths(lo As ListObject, columnWidths As Variant)
    Dim i As Integer
    ' 检查列宽度数组和列表对象列数是否匹配
    If UBound(columnWidths) - LBound(columnWidths) + 1 <> lo.ListColumns.Count Then
        MsgBox "列宽度数组与列表列数不匹配!", vbExclamation
        Exit Sub
    End If
    ' 循环赋值列宽度
    For i = LBound(columnWidths) To UBound(columnWidths)
        lo.ListColumns(i + 1).Range.columnwidth = columnWidths(i)
    Next i
End Sub

```

这个子过程用于设置 ListObject 的列宽度。参数包括 ListObject 对象 (lo) 和列宽度数组 (columnWidths)。子过程会检查列宽度数组和列表对象列数是否匹配，

然后循环设置列宽度。

- BorrowBook 函数:

```
Function BorrowBook(barcode As String, loginID As String, readerCode As String,
operatorID As String, operatorIP As String) As Boolean
    On Error GoTo ErrorHandler
    Dim conn As Object
    Dim strSQL As String
    ' 创建数据库连接对象 (省略)
    Set conn = CreateObject("ADODB.Connection")
    ' 设置连接字符串, 使用之前创建的 ODBC 数据源名称
    conn.ConnectionString = "DSN=YourDSN;UID=YourDatabase;PWD=YourPassword;"
    ' 打开连接
    conn.Open
    ' 构建插入借阅信息的 SQL 语句
    strSQL = "INSERT INTO GDLISNET.流通日志 (操作类型, 条形码, 登录号, 读者条码,
处理时间, 操作员, 操作员 IP) VALUES ('J', '" & barcode & "', '" & loginID & "',
'" & readerCode
& "', SYSDATE, '" & operatorID & "', '" & operatorIP & "')"
    ' 执行 SQL 语句
    conn.Execute strSQL
    ' 关闭连接
    conn.Close
    ' 设置函数返回值为 True, 表示借阅成功
    BorrowBook = True
Exit Function
ErrorHandler:
    ' 错误处理
    MsgBox "借阅图书时发生错误: " & Err.Description, vbExclamation
    If conn.State = 1 Then conn.Close
    BorrowBook = False
End Function
```

该函数用于将图书借阅信息写入到数据库中。接受参数包括条形码 (barcode)、读者条码 (readerCode)、操作员 ID (operatorID) 和操作员 IP 地址 (operatorIP)。函数首先建立数据库连接对象, 然后构建插入借阅信息的 SQL 语句, 最后执行这条 SQL 语句。如果操作成功, 函数返回 True; 如果发生错误, 则显示错误消息并返回 False。

- ReturnBook 函数:

```
Function ReturnBook(barcode As String, readerCode As String, operatorID As String,
```

```

operatorIP As String) As Boolean
    On Error GoTo ErrorHandler
    Dim conn As Object
    Dim strSQL As String
    ' 创建数据库连接对象
    Set conn = CreateObject("ADODB.Connection")
    ' 设置连接字符串, 使用之前创建的 ODBC 数据源名称
    conn.ConnectionString = "DSN=YourDSN;UID=YourDatabase;PWD=YourPassword;"
    ' 打开连接
    conn.Open
    ' 构建插入归还信息的 SQL 语句和删除流通库中记录的 SQL 语句
    strSQL = "INSERT INTO GDLISNET.流通日志 (操作类型, 条形码, 读者条码, 处理时间,
    操作员, 操作员 IP) VALUES ('H', '" & barcode & "', '" & readerCode & "', SYSDATE,
    '" & operatorID & "', '" & operatorIP & "')"
    ' 执行 SQL 语句
    conn.Execute strSQL
    strSQL = "DELETE FROM GDLISNET.流通库 WHERE 条形码 = '" & barcode & "' AND
    读者条码 = '" & readerCode & "'"
    conn.Execute strSQL
    ' 关闭连接
    conn.Close
    ' 设置函数返回值为 True, 表示归还成功
    ReturnBook = True
Exit Function
ErrorHandler:
    ' 错误处理
    MsgBox "归还图书时发生错误: " & Err.Description, vbExclamation
    If conn.State = 1 Then conn.Close
    ReturnBook = False
End Function

```

该函数用于将图书归还信息插入到数据库中，并删除流通库中对应的记录。接受参数包括条形码（barcode）、读者条码（readerCode）、操作员 ID（operatorID）和操作员 IP 地址（operatorIP）。函数首先建立数据库连接对象，然后构建插入归还信息的 SQL 语句和删除流通库记录的 SQL 语句，最后分别执行这两条 SQL 语句。如果操作成功，函数返回 True；如果发生错误，则显示错误消息并返回 False。

- RenewBook 函数:

```

Function RenewBook(barcode As String, readerCode As String, operatorID As String,
operatorIP As String) As Boolean
    On Error GoTo ErrorHandler

```

```

Dim conn As Object
Dim strSQL As String
' 创建数据库连接对象
Set conn = CreateObject("ADODB.Connection")
' 设置连接字符串, 使用之前创建的 ODBC 数据源名称
conn.ConnectionString = "DSN=YourDSN;UID=YourDatabase;PWD=YourPassword;"
' 打开连接
conn.Open
' 构建续借信息的 SQL 语句
strSQL = "INSERT INTO GD LISNET.流通日志 (操作类型, 条形码, 读者条码, 处理时间,
操作人, 操作人 IP) VALUES ('X', '" & barcode & "', '" & readerCode & "', SYSDATE,
'" & operatorID & "', '" & operatorIP & "')"
' 执行 SQL 语句
conn.Execute strSQL
' 关闭连接
conn.Close
' 设置函数返回值为 True, 表示续借成功
RenewBook = True
Exit Function
ErrorHandler:
' 错误处理
MsgBox "续借图书时发生错误: " & Err.Description, vbExclamation
If conn.State = 1 Then conn.Close
RenewBook = False
End Function

```

该函数用于将图书续借信息插入到数据库中。接受参数包括条形码 (barcode)、读者条码 (readerCode)、操作员 ID (operatorID) 和操作员 IP 地址 (operatorIP)。函数首先建立数据库连接对象, 然后构建续借信息的 SQL 语句, 最后执行这条 SQL 语句。如果操作成功, 函数返回 True; 如果发生错误, 则显示错误消息并返回 False。

(2) 数据库查询及算法设计

通过 VBA 实现与 Oracle 数据库的数据交换和管理, 首先需要对现有图书馆集成管理系统的底层数据库表结构进行深入分析。在此基础上, 设计并编写各种 SQL 查询代码, 以操作系统底层表格和视图, 并有效利用查询结果的数据。通过设计合适的算法实现以下功能: 图书信息的查询、图书的借阅、归还和续借操作、对读者借阅信息的分析、馆藏图书的统计、读者信息的修改、数据库的维护以及数据备份等多种操作。这些功能的实现将有助于提升图书馆管理系统的效率和准确性, 同时为用户提供更好的服务体验。以下是在系统功能模块中设计的核心算法及 SQL 代码:

' 查询总借阅次数

```
SELECT Count(*) AS TotalBorrow FROM 流通日志 WHERE 操作类型 = 'J'
```

' 查询平均借阅次数

```
SELECT Count(*) / Count(DISTINCT 登录号) AS AvgBorrow FROM GDLISNET.流通日志 WHERE 操作类型 = 'J'
```

' 查询借阅热门图书 (TOP20)

```
SELECT 馆藏书目库.主键码, 馆藏书目库.题名, 馆藏书目库.责任者, 馆藏书目库.出版者, 馆藏书目库.出版地, 馆藏书目库.出版日期, 馆藏书目库.索书号 FROM ( SELECT 主键码, COUNT(*) AS 出现次数 FROM GDLISNET.流通日志 WHERE 操作类型 = 'J' GROUP BY 主键码 ORDER BY COUNT(*) DESC) Top_20 INNER JOIN GDLISNET.馆藏书目库 馆藏书目库 ON Top_20.主键码 = 馆藏书目库.主键码 WHERE ROWNUM <= 20
```

' 查询图书总量 (种数)

```
SELECT COUNT(*) FROM GDLISNET.馆藏典藏库;
```

' 查询图书总量 (册数)

```
SELECT COUNT(*) FROM GDLISNET.馆藏书目库;
```

' 查询图书类型分布

```
SELECT SUBSTR(索书号, 1, 1) AS category_letter, COUNT(*) AS total_books FROM GDLISNET.馆藏典藏库 WHERE 索书号 IS NOT NULL AND REGEXP_LIKE(索书号, '^[A-Z]') GROUP BY SUBSTR(索书号, 1, 1) ORDER BY category_letter;
```

' 查询馆藏年代分布

```
SELECT TO_CHAR(入藏日期, 'YYYY') AS year, COUNT(*) AS total_books FROM GDLISNET.馆藏典藏库 WHERE 入藏日期 IS NOT NULL GROUP BY TO_CHAR(入藏日期, 'YYYY') ORDER BY year;
```

' 读者信息查询

```
SELECT 读者条码, 姓名, 性别, 身份证号, 读者级别, 单位, 电话, 联系地址, 邮政编码, EMAIL, 密码, 照片 FROM GDLISNET.读者库;
```

' 查询数据库日志文件情况

```
SELECT group#, member FROM v$logfile;
```

' 备份数据库表数据:

```
INSERT INTO backup_table SELECT * FROM original_table;
```

' 查看数据库空间使用情况:

```
SELECT tablespace_name, SUM(bytes) / 1024 / 1024 AS "Size (MB)", SUM(maxbytes) / 1024 / 1024 AS "Max Size (MB)" FROM dba_data_files GROUP BY tablespace_name;
```

' 优化数据库性能:

```
ANALYZE TABLE table_name COMPUTE STATISTICS;
```

3 结论与建议

3.1 结论

本研究以 VBA 技术在图书馆数字化管理系统中的应用为切入点，探讨了其在提高图书馆管理效率、优化服务质量等方面的作用和意义。通过系统的设计与实现，验证了 VBA 技术在图书馆数字化管理中的可行性和有效性，为图书馆管理的现代化和数字化提供了新的思路和方法。

在应用 VBA 技术的过程中，系统实现了以下关键功能：

(1) 通过自动化流程处理功能，简化了图书馆的借书、还书、续借等操作流程，从而提高了工作效率和用户体验。

(2) 数据处理与分析模块充分利用 VBA 与 Excel 等软件的协作，对读者借阅记录、图书馆藏分析等数据进行清洗、提取和分析，为图书馆管理决策提供了有力支持。

(3) 定制化功能开发模块通过 VBA 编程实现了各种定制功能，例如读者信息管理、馆藏编辑等，更好地满足了图书馆特定管理需求，提升了系统的灵活性和适用性。

(4) 系统自动化维护模块运用 VBA 编写程序，实现了系统的自动化维护和排错功能，包括数据备份、清理、数据库维护等，确保了系统的稳定性和可靠性，为持续运行提供了保障。

综上所述，VBA 技术在图书馆数字化管理系统中的应用为图书馆管理工作带来了显著的改进和便利，为实现图书馆的现代化转型和数字化发展提供了重要支持和指导。

3.2 建议

基于对 VBA 技术在图书馆数字化管理系统中的研究和实践，提出以下建议：

(1) 持续优化 VBA 应用：不断学习和掌握 VBA 技术的最新发展，及时更新和优化图书馆数字化管理系统中的 VBA 程序，以确保系统的稳定性和高效性。

(2) 加强人才培养：培养图书馆管理人员和技术人员对 VBA 技术的应用能力，提升他们在系统开发和维护中的专业水平，从而更好地发挥 VBA 技术在图书馆管理中的作用。

(3) 拓展 VBA 应用领域：探索将 VBA 技术应用于更多的图书馆管理领域，如数据分析、信息检索、用户服务等，进一步提升图书馆数字化管理系统的功能和效益。

(4) 加强与用户沟通：与图书馆用户保持密切沟通，了解他们的需求和反馈，

根据用户反馈不断改进和优化图书馆数字化管理系统，提升用户体验和满意度。

(5) 推动行业合作：与其他图书馆或相关机构开展合作，共享 VBA 技术在图书馆数字化管理中的经验和成果，促进行业间的交流与合作，推动整个行业的发展和进步。

4 项目成果

完成了 1 套基于 VBA 的图书馆数字化管理系统（详见电子版材料）。

发表学术论文 1 篇（待刊）。

5 参考文献

- [1]关智远. ExcelVBA 在图书馆中文新书复本推荐中的应用[J]. 电脑知识与技术, 2020, 16(05):55-57+107. DOI:10.14004/j.cnki.ckt.2020.0518.
- [2]郑珊霞. 基于 VBA 的电子图书采访辅助系统构建及实现[D]. 扬州大学, 2019. DOI:10.27441/d.cnki.gyzdu.2019.001941.
- [3]王朋, 付正芳, 张静, 等. 基于 VBA 编程的高校图书馆图书在线荐购设计[J]. 中华医学图书情报杂志, 2018, 27(10):65-68.
- [4]王娜, 马永娟. 利用 VBA 实现图书到货的一键统计[J]. 电脑编程技巧与维护, 2015, (22):8-9. DOI:10.16184/j.cnki.comprg.2015.22.003.
- [5]江均勇. 基于 ExcelVBA 创建高效的图书验收系统[J]. 电脑编程技巧与维护, 2013, (16):42-43. DOI:10.16184/j.cnki.comprg.2013.16.038.
- [6]徐海, 姚翔. 通用查重比对模板系统的 VBA 实现[J]. 现代图书情报技术, 2010, (02):95-99.
- [7]马静. 基于 Access 的图书借阅管理系统[J]. 装备制造技术, 2009, (07):90-91.